# GVE'99

## COMPUTER GRAPHICS
## AND
## VISUALIZATION EDUCATION '99

Charting the Future for Computer Graphics Education

July 3-5, Coimbra, Portugal

EUROGRAPHICS Workshop
co-sponsored by ACM SIGGRAPH

PROCEEDINGS

Editors

José Carlos Teixeira
Werner Hansmann
Michael B. McGrath

This work is supported by the National Science Foundation, under AWARD 9906124. All opinions, findings, conclusions and recommendations in any material resulting from this work are those of the principal investigators, and do not necessarily reflect the views of the National Science Foundation.

# PREFACE

Welcome to GVE'99! Welcome to Coimbra!

Following several Eurographics education workshops held in different European cities, the Computer Graphics and Visualization Education '99, under the motto "*Charting the Future for Computer Graphics Education*", will be held in Coimbra, Portugal, from 3th to 5th July. Eurographics organized this workshop in cooperation with ACM SIGGRAPH and with the support of the National Science Foundation. Computer graphics and visualization education specialists will meet to share their experiences and their visions and to make projections and recommendations for the next ten years of computer graphics education.
43 position papers were submitted from 15 countries. The excellent position papers we received promise a very interesting event. People were invited to participate in the workshop according to their contributions. Based on the contributions and the workshop goals, the IPC decided that the workshop should be organised so that participants spend significative time working in group. Group discussion will be one of the most important parts in the workshop, addressing different educational issues. The participants will be free to choose the working groups they are most interested in or they feel they can best contribute to. The main goal of the few formal presentations we will have is to give broad overviews and start the discussions.
Considerable time will be spent discussing future directions for computer graphics and visualization education. The recommendations and predictions from this workshop will be distributed at both SIGGRAPH'99 and Eurographics'99 conferences as part of their joint celebration of SIGGRAPH's 30th anniversary and Eurographics' 20th anniversary.
Based on the position papers quality, some of them will be selected for revision and later published in the journal Computers & Graphics.

The workshop co-chairmen would like to thank all the speakers for their valuable effort in preparing their formal presentations; the participants for their contributions; the IPC members for the effort in promoting and structuring this event; the GVE'99 Committee for its continuing and active support during all the preparation phase; and the workshop organisation committee for its committed and creative effort to make the GVE'99 a pleasant workshop.

We welcome you and hope you enjoy GVE'99.

Coimbra, July 1999.

*José Carlos Teixeira, Werner Hansmann, Michael B. McGrath*

## INTERNATIONAL PROGRAMME COMMITTEE

Ana Luisa Solis, UNAM, Mexico City, Mexico
Anne Mumford, Loughborough University, UK
Beatriz Sousa Santos, University of Aveiro, Portugal
Bianca Falcidieno, CNR, Italy
Charles Wuethrich, Bauhaus University Weimar, Germany
Danail Dochev, IIT-BAS, Bulgaria
Dena Eber, Bowling Green State University, USA
Dieter Fellner, Braunschweig University of Technology, Germany
Gary Bertoline, Purdue University, USA
Gitta Domik, University of Paderborn, Germany
Jack Bresenham, Winthrop University, USA
Jacquelyn Morie, Rhythm & Hues, USA
Jiaoying Shi, Zhejiang University, Hangzhou, China
John Patterson, University of Glasgow, UK
José Encarnação, Technical University of Darmstadt, Germany
Judith R. Brown, The University of Iowa, USA
Ken Brodlie, University of Leeds, UK
Lars Kjelldahl, Royal Institute of Technology, Sweden
Lewis Hitchner, California Polytechnic State Univ., San Luis Obispo, USA
Mário Rui Gomes, IST - Instituto Superior Técnico, Portugal
Michael Joseph Zyda, Naval Postgraduate School, USA
Pere Brunet, Polytechnic University of Catalunya, Spain
Peter Stockinger, MSH, France
Roger Hubbold, Manchester University, UK
Rosalee Wolfe, DePaul University, USA
Scott Owen, Georgia State University, USA
Stavros Christodoulakis, TUC/MUSIC, Greece
Steve Cunningham, California State University Stanislaus, USA
Vaclav Skala, University of West Bohemia, Czech Republic
Yuri Bayakovski, Moscow State University, Russia

## ORGANIZATION

CCG – Computer Graphics Center – Coimbra

## WORKSHOP CO-CHAIRMEN

José Carlos Teixeira
Dep. Matemática - F.C.T. - Universidade de Coimbra

Werner Hansmann
FB Informatik/TIS - Universität Hamburg

Michael B. McGrath (ACM SIGGRAPH Contact)
Engineering Department - Colorado School of Mines

## WORKSHOP ORGANISATION COMMITTEE

Joaquim Madeira (FCTUC)

César Páris

Sónia Branco

Paula Santos (Secretary)

Vítor Varalonga

# Index

## 6 - Human-Computer Interaction

### Presentation



### Position Paper



## 7 - Media Education

### Presentation



### Position Paper



## 8 - Courseware

### Presentation



### Position Papers

# Educational Trojan H o rses

**Mike Bailey**
University of California at San Diego
San Diego Supercomputer Center
mjb@sdsc.edu

I have always found that it is difficult to teach _at_ people. It is very much like throwing a bucket of water on your plants: you can't be sure how much has soaked in and how much has evaporated or run off somewhere else. You can't even be sure if water is what they really need right now.

Rather, I believe it is much more productive to "trick" people into teaching themselves. As educators, our job is less to throw information at our students than to motivate and excite them into wanting to reach out and pull in the information.

The question then is how to do this. How do we instil educational motivation in someone so that it blossoms into intrigue and curiosity? How do we make it strong enough for them to want to reach out and figure out more? I call this an "Educational Trojan Horse" approach because you slip some education by the students when their guard is down. After all, the best education happens when people think they are exploring, not being educated.

I believe that computer graphics is _the_ key element of an Educational Trojan Horse. Graphics is such a perfect mix of technology and fantasy that it can be used to excite students as no other medium can. The video game appearance causes students to let down their guard and allow themselves to be drawn in. The ability to control, design, and simulate leads them into learning. And, they not only develop their own explanations, they are motivated to discuss it with each other.

This talk will highlight some key technologies and technology trends that can be used to create Educational Trojan Horses. It will include 3D graphics, hardware, software, web-based graphics, digital libraries, and common formats.

# Strengthening Visual Skills by Recognising Rendering Algorithms

**Rosalee Wolfe**
DePaul University
Chicago, Illinois
wolfe@cs.depaul.edu

**Abstract**

*Computer science students contemplating a career in graphics need to develop a visual sense, but traditional course topics do not meet this need. Visual analysis is a teaching technique developed for computer science instructors that helps impart this ability. Through the use of a few visual cues, students learn to visually identify surface algorithms, shaders and lighting techniques. An interactive software package called TERA (Tool for Exploring Rendering Algorithms) provides nearly a million image combinations that students can use to practice their visual identification skills.*
**Keywords:** *algorithm visualisation, computer graphics education.*

## 1. Background

For a successful career in computer graphics, computer science majors need to develop a visual sense in addition to their technical knowledge. Computer science graduates hired by the graphics industry will work side-by-side with artists, and it is essential for effective communication that they have an appreciation and enthusiasm for the visual aspects of the field. In addition to programming skills and knowledge of operating systems, recent labor market analyses list "strong visual style sense" and "understanding how artistic elements work with technical elements," as desirable traits for programmers and technical directors [1,2]. An increasing number of graphics houses want to see a demo reel or a portfolio from computer science graduates who count Unix shell scripting, C++ programming and the mathematical derivation of NURBS in their skill set.

There are many exemplary resources [3, 4] and learning tools that allow students to examine aspects of geometry [5, 6, 7, 8, 9, 10], rasterisation [6, 11], illumination [6, 12, 13, 14] and the entire 3D rendering pipeline [13]. These present visualisations that a student can use to explore and study an algorithm in isolation. This is a sound pedagogical approach for a first introduction, but eventually a student will need to understand an algorithm's behavior in co-operation with other algorithms and in the context of a finished image.

Unlike other specialities in computer science, the choice of a graphics algorithm usually cannot be based solely upon space, speed and implementation considerations. Equal in importance to time and memory requirements is the visual effect that a graphics algorithm produces. In fact, visual appearance will be the overriding factor for some applications. Visual understanding is essential to developing and debugging new algorithms. For example, when writing a shader, a programmer looks at a rendered sample, analyses how its appearance differs from the ideal and uses the information thus gained to refine the shader. An essential aspect to developing a visual sensibility is the ability to identify and compare rendering algorithms. Visual knowledge can prepare students to answer such real-life questions as, "Algorithm X gives us exactly the effect we want, but it is prohibitively expensive. Can we get a similar effect by tweaking our implementation of Algorithm Y?"

Although the graphics industry desires a developed visual sensibility in new computer science graduates, there is no place in the traditional curriculum where students can gain this knowledge. The discipline of computer science draws on mathematics and emphasises algorithm study and development, which are almost entirely text-based and involve little visualisation. Computer science instructors often view the prospect of discussing the visual aspects of computer graphics as a daunting, if not overwhelming endeavor. Typical reactions include

I'm not an artist! How am I supposed to gain the requisite background to teach this?

I don't have time for this in class – I have too many algorithms to cover as it is.

Both reactions are understandable and justifiable. Computer science instructors are not artists and their goal does not include producing artists. Further, there is an enormous amount of technical knowledge in the computer graphics discipline, and attempting to decide

what topics belong in an introductory course has been the topic of many papers. Any approach to the problem of inculcating a visual sensibility has to be one that does not take up much lecture time and constitutes a teaching method that computer science instructors are willing to try. This paper proposes that *visual analysis* is one approach that fits these constraints.

## 2. Visual Analysis

Visual analysis is a teaching technique developed specifically for computer science instructors that imparts this knowledge [15]. The technique stems from critical analysis, which establishes a structure for examining works of art [16]. Students in the visual arts learn to describe and compare works in terms of design, concept and media.

Instead of design, concept and media, visual analysis teaches students to recognise a small number of visual cues, including:

- visibility of polygon faces or polygon edges
- transition from light to dark on diffuse-reflecting surfaces
- color and shape of specular highlights
- presence of transparency, reflection, refraction, patterns and textures

| | |
|---|---|
| 1. First, determine the surface algorithm. | |
| *Appearance* | *Algorithm* |
| Outlined polygons. Object's far sides are visible. Uninterrupted horizon lines. | Wireframe |
| Outlined polygons. An object's far sides are not visible. Occluded objects are invisible. | Hidden-line removal |
| Filled-in polygons. Presence of refraction in transparent objects. (Presence of reflection, shadows are also helpful) | Ray tracing |
| Filled-in polygons. No refraction, reflection or shadows. | Z-buffer |
| 2. Determine the shader. | |
| For Z-buffer | |
| One color per object. Objects appear flat, as if cut from paper. | Constant |
| One color per polygon. Objects now appear to have a shaded contour. | Faceted |
| Smooth transitions from light to dark. Specular highlights follow polygon edges. | Gouraud |
| Smooth transitions from light to dark. Specular highlights are white, compact and elliptical. "Shiny plastic" look. | Phong |
| For ray tracing | |
| Opaque, colored surface. No highlight. | Diffuse |
| Opaque, colored surface with highlight. | Diffuse and Specular |
| Transparent object that appears to bend light. | Transparency |
| Part of scene is visible in the surface of the object. | Reflection |
| 3. Determine additional surface interest (both z-buffer and ray tracing) | |
| Image appears to have been pasted or glued onto object. | 2-D texture mapping |
| Object appears to have been carved from a solid substance like stone or wood. | 3-D texture mapping |
| Object surface appears rough or wrinkled, but its profile is smooth. | Bump mapping |
| 4. Light sources | |
| Harsh shadow / lighter shadow. | Low/high ambient |
| Sharply defined shadow. | Point light |
| Presence of a penumbra, but degree of shade is constant within penumbra. | Area light |
| Color bleeding; deepening of shadow in corners, under furniture. | Radiosity |

Table 1: Visual cues and algorithms (visual equivalencies omitted).

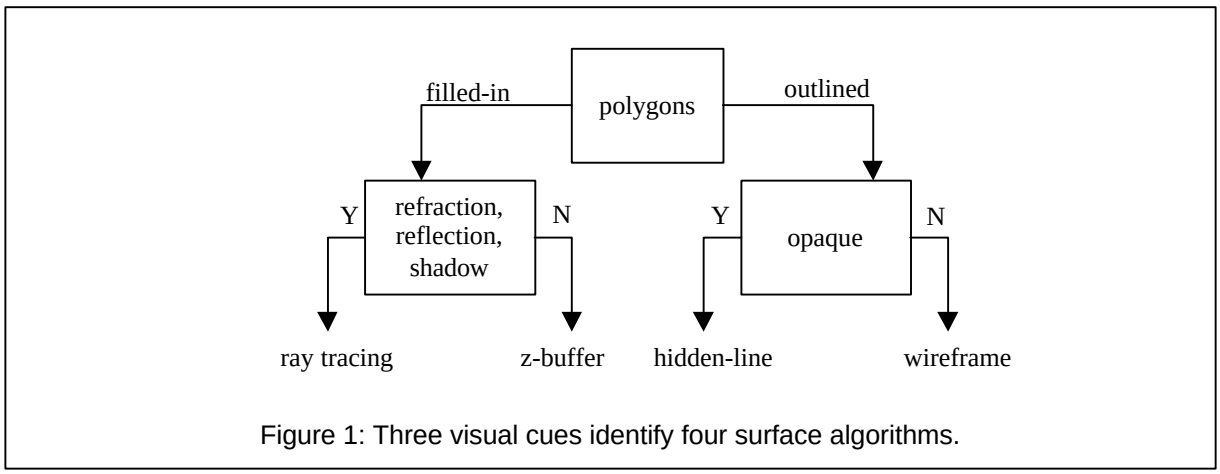



Figure 1: Three visual cues identify four surface algorithms.

- sharpness of shadow
- interactions between adjacent diffuse reflectors (color bleeding)

These cues are usually sufficient to identify a rendering algorithm as Table 1 demonstrates. Informal studies have shown that both instructors and students find this list of cues non-threatening and easy to spot in an image. By learning to observe and describe these cues, students are able to identify rendering algorithms.

## 3. Classroom Presentation

In the first lecture, the teacher discusses images portraying three or four algorithms that produce starkly different effects. For example, the four surface algorithms of wireframe, hidden-line, ray tracing and z-buffer create distinctive effects and can be distinguished by the use of only three visual cues. See Figure 1. Each week, the teacher adds more algorithms, and by the midterm, the class has examined all commonly used rendering algorithms. Until the midterm, the teacher emphasises the characteristic visual behavior of the algorithms. After the midterm, the teacher shows how multiple algorithms can achieve equivalent effects. By the end of the course, students learn to identify the surface algorithm, shaders and types of light sources.

While discussing a visual cue, the teacher presents two or three images that demonstrate it. After explaining the visual cue, the teacher gives the name of the algorithm that creates the cue. The teacher then shows a short series of images and invites students to identify the visual cues and then suggest a rendering algorithm. Students should specify the cues first so that they learn to spot them in the context of an image. Any premature guesses as to the identity of the rendering algorithm are met with the response, “Cues first!”.

After the first class meeting, discussions on visual analysis begin with a short series of images that review the rendering algorithms from the previous meeting. Students first name the visual cues and then suggest a rendering algorithm.

The methodology requires only five minutes of an hour-long lecture. Such a small amount of time will not significantly impact the presentation of other topics, especially if the teacher covers visual analysis during the last five minutes, when student focus is beginning to wander. In fact, there is a way of presenting this material without using any lecture time at all, as will be discussed in the next section.

## 4. Study Materials

Students need to practice visual analysis outside of the classroom. While posting images on the Web or pointing out examples in textbooks is a start, students say that they benefit more from the question-and-answer sessions held at the end of a lecture. Simply looking at images does not promote active learning.

Having students learn a rendering package helps a bit because students can choose parameters and view the resulting images. However it takes a significant amount of time to learn a package, and since the students have a priori knowledge of the surface and shading algorithms, this approach does not provide a student with a means of self assessment. A better approach is to provide students with an easy-to-use interactive tool that can demonstrate any rendering or shading algorithm while encouraging active learning.

TERA (Tool for Exploring Rendering Algorithms) is an interactive program that facilitates comparative study of the visual effects of rendering algorithms [17]. See Figure 2 for an annotated screen dump of TERA.

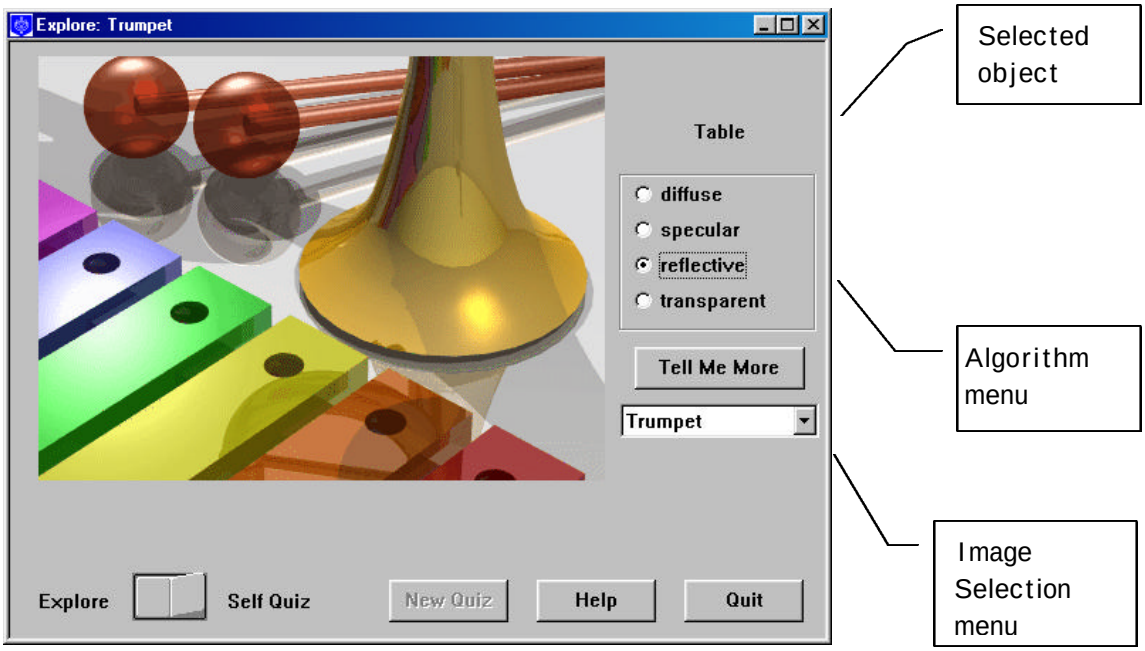



Figure 2: A sample TERA session.

A student can choose a scene and select a rendering algorithm for any object in the scene. Students can practice visual analysis using TERA. In "Self Quiz" mode, students select a scene, and TERA presents it with each object rendered by a random algorithm. Students then guess the rendering algorithm for each object in the scene. TERA responds with "Correct", "Try Again" or "Close Enough".

The "Close Enough" response is for those cases where multiple algorithms produce similar visual effects. For instance, Gouraud and Phong shading produce similar effects when no highlight is present.

Always available is the "Tell me more" button, which provides specific feedback about a student's last algorithm selection. When a student is in "Explore" mode, the "Tell Me More" button will activate a popup window describing the relevant visual cues. In "Self Quiz" mode, when students get a response of "Close Enough", they can click on "Tell Me More" for an explanation. A pop-up window will list the algorithm they picked, the actual algorithm and a specific explanation of why the two algorithms produced effects that are visually equivalent. For example, if a flat surface is very evenly lit, then Gouraud shading may produce variations in shade that are so subtle that the result looks like constant shading. In this situation, if a student picks "constant" and receives the "Close Enough" response, the "Tell Me More" button will activate a pop-up window containing the relevant explanation. See Figure 3.

The new version of TERA is capable of creating nearly a million images for students to analyse. The images cover surface algorithms, z-buffer shaders, ray tracing shaders, texture mapping, bump mapping, lighting and visual equivalencies.

Some teachers do not discuss visual analysis in the classroom, but give a short demonstration of TERA and tell students that TERA is available in the lab. TERA has enough appeal that students are drawn to it, and they spend enough time with it that they can score reasonably well on a visual identification test [18].

## 5. Results and Feedback

In my experience, visual analysis adds excitement and a sense of the "big picture" to introductory graphics courses. Students may not be able to implement every rendering algorithm when they leave the course, but they will be able to recognise them and know their names, which provides a starting point for further investigation.

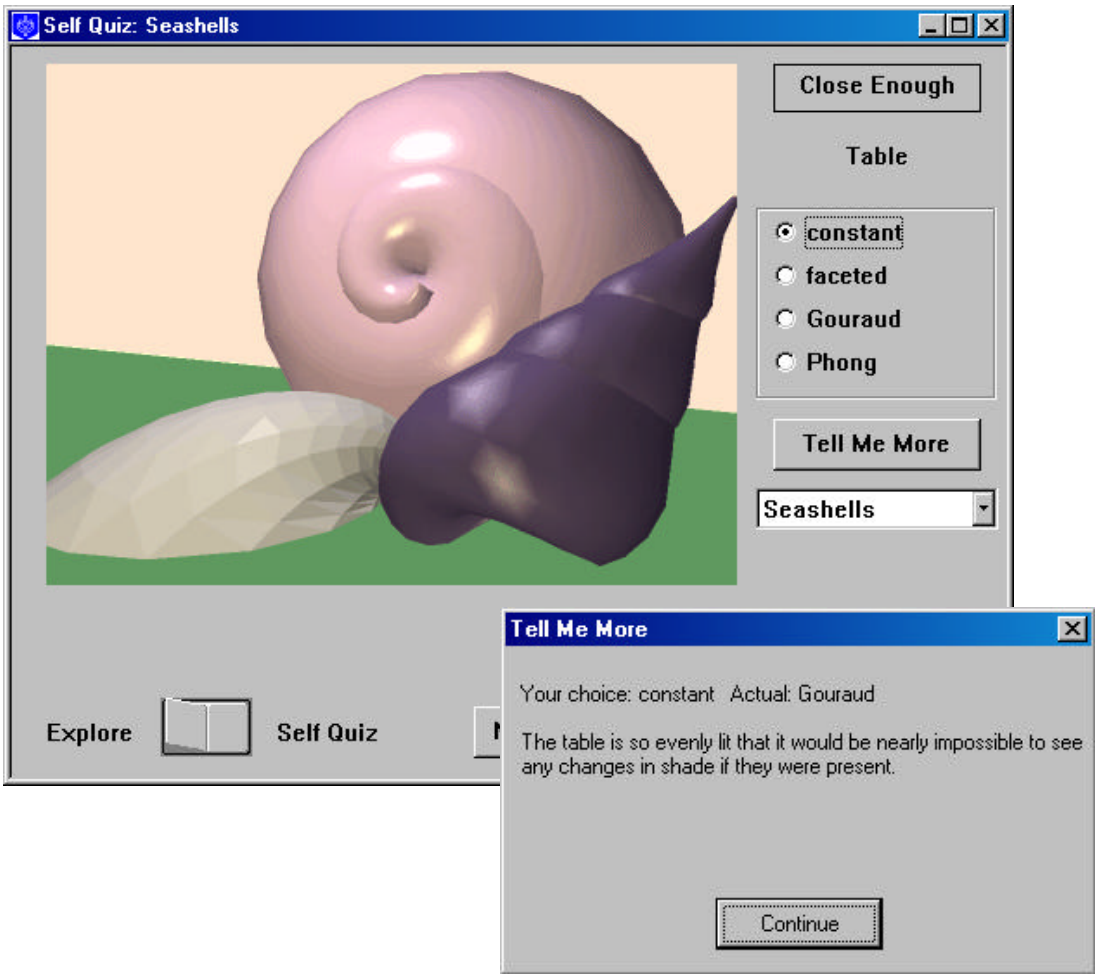



Figure 3: TERA in "Self Quiz" mode.

A visual sensibility, fostered by visual analysis, enhances a depth of knowledge of those rendering algorithms that students do implement. Because they are already familiar with the visual behavior of the algorithm, they have expectations of how an image should appear and can perform more of their debugging on their own without outside help. Students no longer ask, "Is it right?" They state, "There's something wrong with my highlight."

A very exciting development is the favorable feedback from digital artists. This approach promises to help art students "grasp the differences between light models, surface algorithms and shaders" [19]. Further work includes the exploration of developing the approach to expand the possibilities of better communication between digital artists and computer scientists.

## References


[1] Regan and Associates. A Labor Market Analysis for the interactive Digital Media Industry: *Opportunities in Multimedia*. Sunnyvale, CA: North Valley Private Industry Council (NOVA), 1997.

[2] Public Affairs Coalition of the Alliance of Motion Picture and Television Producers, *Making Digits Dance: Visual Effects and Animation Careers in the Entertainment Industry*. Sunnyvale, CA: NOVA Private Industry Council, 1997.

[3] Owen, G. Using Hypermedia to Teach Computer Graphics. Presented at the Graphics and Visualization Workshop in *Eurographics '93* Barcelona, September, 1993.

[4] G. S. Owen. Integrating World Wide Web

Technology into Courses in Computer Graphics and Scientific Visualization. *Computer Graphics* 29(3) pages 12-14, August 1995.

[5] D. Schweitzer. Designing Interactive Visualization Tools for the Graphics Classroom. In *Proceedings of the Twenty-third SIGCSE Technical Symposium on Computer Science Education.* pages 299-303, 1992.

[6] P. Min. www.cs.princeton.edu/~min/cs426/applets.html

[7] A. Shabo, M. Guzdial and J. Stasko. Addressing Student Problems in Learning Computer Graphics. *Computer Graphics* 29 (3) pages 38-40, August 1996.

[8] R. Klein, F. Hanisch, W. Strasser. Web-Based Teaching of Computer Graphics: Concepts and Realization of an Interactive Online Course. *Computer Graphics Annual conference Series, Conference Abstracts and Applications.* pages 88-93, 1998.

[9] Y. Zhao, J. Lowther and C. Shene. A Tool for Teaching Curve Design. In *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education.* pages 97-101, 1998.

[10] Y. Zhou, Y. Zhao, J. Lowther and C. Shene. Teaching Surface Design Made Easy. In *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education.* pages 222-226, 1999.

[11] A. C. Naiman. Interactive Teaching Modules for Computer Graphics. *Computer Graphics* 30 (3) pages 33-35, August 1996.

[12] Ansari, M. Simple Illumination Visualization. Department of Computer Science, DePaul University, Chicago, 1993.

[13] E. Wernert. A Unified Environment for Presenting, Developing and Analyzing Graphics Algorithms. *Computer Graphics* 31 (3) pages 26-28, August 1997.

[14] D. Gould, R. Simpson and A. van Dam. Granularity in the Design of Interactive Illustrations. In *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education.* pages 306-310, 1999.

[15] A. Sears and R. Wolfe, Visual Analysis: Adding breadth to a computer graphics course. In *Proceedings of the Twenty-sixth SIGCSE Technical Symposium of Computer Science Education.* pages 195-198, March 1995.

[16] R. Arnheim. *Art and Visual Perception: A Psychology of the Creative Eye.* 2nd ed. Berkeley: University of California Press, 1974.

[17] R. Wolfe, A. Sears. TERA: an interactive Tool for Exploring Rendering Algorithms. *Journal of Computing in Small Colleges* 10 (4) pages 41-46, February 1995.

[18] R. Wolfe, S. Grissom, T. Naps and A. Sears. A Tested Tool for Teaching Computer Graphics. *Journal of Computing in Small Colleges* 12 (2) pages 70-77, November 1996.

[19] D. Eber. Personal communication, December 1997.

# An Introductory Cou rse on Visualization

**Beatriz Sousa Santos**
Departamento de Electrónica e Telecomunicações, Universidade de Aveiro
Aveiro, Portugal.
bss@inesca.pt

**Abstract**

*A Visualization course offered twice (1997/98 and 98/99) as an elective in the MSc degree on Electronics and Telecommunications at the University of Aveiro is presented. Its contents, bibliography and teaching methods are described. Some difficulties encountered during the preparation and lecturing of this course are identified.*

## 1. Introduction

Taking into consideration that Visualization is becoming very important and useful in many areas, an introductory course on Vizualization seems a valid contribution to the curriculum of any post-graduation in science or technology and thus it was considered adequate as an elective course of the MSc in Electronics and Telecommunications offered at the University of Aveiro. This post-graduation program includes several courses and a thesis and aims to be a large spectrum degree encompassing mainly one of four areas of Electrical Engineering (Electronics, Telecommunications, Signal Analysis and Processing and Computer Science); this means that it can give formation either to future consumers or facilitators of visualization techniques and systems. This is the context where the referred introductory course was meant to exist; thus its general objective is to introduce the students to a new area which evolves rapidly, addressing the fundamental concepts, providing a basic foundation, good enough to allow and encourage them to apply or proceed work in that area. Stating this objective more specifically, this course should introduce the students to what Visualization is and can do, as well as it should make them appreciate its benefits and how current tools can be exploited in many application areas; it intends to give a consumer's perspective as well as a facilitator's perspective. However such a course should also serve other more general objectives related to training students to be able to perform the kind of work a post-graduated is expected to do; this involves training them on doing bibliographic research, using adequate working methods and correct technical communication (i.e. writing and speaking).

In the following sections a brief description of the general contents, bibliography and teaching methods used twice (1997/98 and 98/99) on the introductory course on Visualization will be presented. Some difficulties encountered by the author during the preparation and lecturing of this course are also identified.

## 2. General Contents

This course was designed to have 36 hours (24 lectures) spread along a semester; some of these lectures are dedicated to other activities as justified in section 3. The list of addressed topics, presented bellow, indicates the duration, focus and specific bibliographic references (briefly described in section 4) which have been used.

1- Introduction: definitions, history, goals and principles of Visualization (1 lecture)
2- Overview of Visualization applications (2 lectures)
3- Human Visual System (1 lecture)
4- Visual Representation of Quantitative Information (1 lecture)
5- Framework (1 lecture)
6- Visualization Techniques (4 lectures)
7- Foundations on the enabling technologies: Computer Graphics, Digital Image Processing and Human Computer Interaction (7 lectures)
8- Data Characteristics (1 lecture)
9- Visualization Products (1 lecture)

In the first lecture, corresponding to topic n.1, a general introduction to the area is made. Besides presenting the definitions and goals of Visualization, a focused approach to visualizing data is presented as a means to select adequate techniques to allow exploration, analysis and then presentation of data. The main references used are the books by Earnshaw and Wiseman, and Keller and Keller.

This first lecture is of great importance, it is intended to motivate the students and establish the focus and "philosophy" of the course. Two fundamental ideas are conveyed: a) Visualization is concerned with exploring data and information graphically as a means of gaining understanding and insight into the data (not just presentation graphics!) b) how to do

visualization? There is a need of a methodology for selecting visualization techniques, however no "recipes" exist. The focused approach described by Keller and Keller in the first section of their text is presented. To conclude, a set of images produced from seismic data (also from the book by Keller and Keller) is used to illustrate the incremental process one may expect while seeking the best way of visualizing data.

Topic n.2, is meant to give an idea of the large quantity of applications that Visualization finds in a great diversity of scientific and engineering disciplines. This should make the students aware of the importance and usefulness of Visualization, as well as allow them to better understand all the other topics of the course. This overview is given through the presentation of a large collection of visualization images obtained in a variety of disciplines and emphasising the kinds of information revealed rather than the details of the visualization techniques. A number of examples extracted from several bibliographic references is used (as the books by Brown et al., Keller and Keller, Brodlie et al., Earnshaw and Wiseman).

The third topic is concerned with the "human part" of the visualization process, which is essential. The capabilities and limitations of the human visual and information processing systems are briefly described, stressing that they have important implications for design and that users, in spite of sharing common capabilities and limitations, are individuals with differences which should not be ignored. The human visual system is the main subject and the lecture is based on a document prepared by the author which is included in HyperVis (an on-line tutorial on Visualization) and indicates specific bibliography.

In Topic n.4, the fact that visualization is not new in concept is stressed; most principles that have been used to produce good maps, scientific drawings and data plots apply to computer visualization. This lecture is based on the interesting work of E. Tufte (mainly the 1983 text but also the 1990 text, to support colour usage).

Topic n.5 presents a framework model describing Visualization systems in abstract terms; this framework will be used to present techniques, data characteristics, products and applications in the remaining lectures. As an introduction to this topic, models of scientific investigation and visualization process are presented. This lecture is based on the corresponding chapter of the reference by Blodlie et al.

The concept of visualization technique, introduced in the previous topic as the responsible for generating and manipulating a graphic representation from a set of data and allowing investigation through user interaction, is developed in topic n.6. The three main elements of a visualization technique are described accordingly to the model used by Brodlie at al.; a classification and a notation for visualization techniques introduced by the same authors is used to support the study of a range of techniques, generic in nature and which can be tailored to different applications. These techniques are illustrated through several examples (extracted also from the references by Brown et al., Keller and Keller, among others).

Topic n.7 addresses the main aspects of Computer Graphics, Digital Imaging Processing and Human Computer Interactions needed to be able to understand the visualization techniques. Addressed issues are: a 2D S/W package, geometrical transformations and projections, specifying a view in 3D, colour, overview of the problems and methods to achieve realism, visibility, illumination and shading, histogram operations, filters and image transforms and general principles for user interface design. The explanation of these subjects has always in mind its use in Visualization and examples including Visualization techniques already introduced are used as much as possible.

Topic n.8 is concerned with data. A classification that seems suitable for describing different types of data identified within the used reference model is presented. An overview of data formats and data compression techniques is also presented; this overview can be supported by the reference on graphics file formats by Brown and Shepard.

A Visualization system is presented, ideally, as an integrated whole providing means to support the effective exploration of complex data. Topic n.9 presents a classification of visualization software and a variety of existing software products. The adopted classification is an hybrid between the classification used by Brodlie et al. and the one proposed by Brown et al.. The software buyer's guide included in Brown et al. is used to illustrate the great variety of existing commercial products and relevant characteristics which need to be considered in the choice of a S/W product. The study of this subject is complemented by demonstrations some of the students have to make on several of these products (as part of their assignments).

This is the general contents of the curriculum for the presented course as it has been applied on both academic years of 1997/98 and 1998/99. From the initial proposal of this course [1], the main change corresponded to a substantial increase on the number of lectures devoted to the enabling disciplines. The author had initially planned to spend a lesser amount of time on these subjects since elective courses on those subjects (Digital Image Processing, Computer Graphics and Human Computer Interaction) are

currently offered at the Department and it seemed reasonable to expect that students taking the Visualization course would have already some background on those subjects. The fact that none of the students, in both academic years, had that background came as a surprise; in fact it seems that, most of the students had made this choice in order to get some formation in those areas. For this reason, the author decided to increase substantially the amount of time devoted to enabling technologies; this option, although controversial from the point of view of a visualization curriculum, appeared to be the most reasonable choice for those students, possibly allowing them to benefit the most from the course.

The current contents of the course follows, in essence, what has been done by the community of Visualization educators [2,3,4]. It is expected to evolve, according to the experience obtained, each time it will be taught; however the author expects evolution to occur mostly in the specific way each topic is addressed, bibliography, sequence or duration of different topics, rather than on the overall structure.

## 3. Teaching Methods

As referred, the course is organised in 1h 30min lectures along a semester, i.e., two lectures a week; its distribution by the topics is indicated in the previous section. There are no practical classes; students are supposed to perform their assignments, under supervision of the teacher, but on their own time and with a high degree of autonomy. These students usually work full time (at the University or at enterprises), which makes very difficult to organise working sessions involving all the students; that is the reason why only 19 lectures are assigned to the topics, i.e., 5 are left open and are "sacrificed" to activities that could, otherwise, be performed in extra hours. These sessions have been devoted to:

- presenting some selected SIGGRAPH videos[1] with examples of applications (1 lecture)
- visiting facilities where people use visualization regularly in their work (1 lecture)
- presenting student's assignments (2 lectures)
- lectures by people that use or make visualization (1 lecture)

[1] from SIGGRAPH Video Review Issue 108 - Scientific Visualization 95 which, in spite of having some years, still contains interesting examples; however a more recent issue already exists: SIGGRAPH Video Review Issue 124 - SIGGRAPH 97: Visualization Program, that could be interesting.

Until now two visits were organised, both related to Earth Sciences and inside the Campus, one to a Laboratory using Remote Sensing data and another to a Laboratory of Geophysics including some field demonstration on the acquisition of Ground Penetrating Radar data. Both visits where guided by researchers from those laboratories, showing the students which kind of work they are involved in and how they use at the moment, and they would like to be able to use, visualization techniques and tools. Some other laboratories exist at the University that can be visited in years to come, as in Chemistry and Environment.

The two lectures by external people that have been organised were concerned with visualization of Archaeological data and Geographical Information Systems; however a great variety of other subjects might be interesting.

As referred, general objectives of this course include training students on doing bibliographic search, using adequate working methods and correct technical communication (i.e. writing and speaking). Since these objectives are usually easier to achieve when the evaluation is based on practical assignments and the expected number of students in elective courses on this MSc is low, that was the type of evaluation used in both academic years (the course had 5 students in 1997/98 and 6 in 1998/99).

Each student has two different assignments, the first consists in giving an oral presentation of ≈ 40 minutes on some subject and the second, a more demanding assignment, consists in a work which in general involves some implementation (either developing code in some language as C++ or using a visualization S/W as IDL). Both assignments are proposed by the teacher taking into consideration the profile of each student (background, interests, etc.); however the students are told that they can propose their own assignment by presenting an extended abstract; these proposals may, of course, be rejected. On other courses, offered by the author of this work, some interesting proposals have been presented and accepted; however, on the Visualization course none of the students decided to do it, until now.

The choice of the subjects for the oral presentations as well as for the second assignment is related to the current interests of both the teacher and the students as well as to interests of other people (from the University or outside) that ask for collaboration in some specific visualization tasks. Along these two academic years the oral presentations have consisted mainly in S/W demonstrations (Matlab, IDL, Iris Explorer and OpenGL) or have addressed volume rendering algorithms. Second assignments have been related to the following interest areas:

- visualization of archaeological data from a XV century shipwreck existing in the Aveiro Lagoon
- visualization of geophysical data obtained from Ground Penetrating Radar
- visualization of data related to the use of Location-Routing models of obnoxious facilities
- development of an hardware processor for volume rendering implemented with FPGAs
- development of evaluation methodologies for visualizations used in the study of epilepsy and involving data obtained from different modalities as surface electroencephalogram (EEG), depth electro-encephalogram (SEEG), tomography (CAT) or magnetic resonance imaging (MRI)
- development of evaluation methods for visualizations obtained using raycasting
- development of evaluation methods involving models of Human Visual System

For the supervision of many of these assignments the author had the invaluable collaboration of colleagues working in enabling technologies and application areas.

These assignments consist, in general, of a bibliographic research on the subject, a proposal of what work could be interesting to develop, an implementation of part of this work, a small technical report and a document written as a scientific paper. Most students put a lot of work and enthusiasm into these assignments and supporting them has been very interesting, however students typically take long time to finish their tasks. In spite of the fact that this course is not a research oriented course such as the one described by Banks [5], many of the recommendations and difficulties presented by this author are also applicable; the choice of subjects for the second assignment, its supervision and finally the preparation of the paper corresponds to a significant burden associated to the course; this solution is only viable for a very small number of students.

This type of assignments have resulted very stimulating for the more interested students, however for less motivated students it is perhaps too demanding and may result frustrating both for the student and the teacher. For these reasons, the author is considering the possibility of maintaining this second assignment for volunteer students and give the alternative of an examination for the ones who are not so committed to visualization.

One of the main difficulties has to do with the writing of the "scientific paper", many students just don't like to write (and don't have enough experience) and a special attention is needed to this part of the work (as well as in the preparation of the oral presentation). Difficulty in writing is a general problem of our students, so some extra effort has been devoted to it, in this course. During a lecture, the teacher usually stresses the importance of communicating correctly any work and generally instructs the students how to do it before they start preparing their oral presentations or their papers; some bibliography on the subject of writing and speaking is also recommended [6,7]. The teacher also encourages students to write in English since it is the "lingua franca" in technology. The process of preparing the papers is iterative and some of them need as many as 3 or 4 iterations. Since the teacher and, in some cases, other people also have to invest effort, not only in the supervision but also in reviewing the paper they become co-authors; when the final version of a paper is ready it can be published as an internal note (in the Department's Journal), submitted to a conference or not used at all (accordingly to the nature and quality of the developed work and achieved results).

From the 5 papers of 1997/98, four were published as internal notes; modified versions of two were accepted at conferences organised in Portugal and one will be submitted to another conference. One of the assignments involved porting to a different platform an already existing S/W developed by someone who was far away and no longer working on the subject; in spite of the fact that the student has done enough work to succeed in the course, the obtained results did not justify their publication. This is always a risk of this kind of assignments and care should be taken to minimise it, however the author feels it is impossible to completely eliminate the possibility of "something going wrong".

Papers accepted in conferences should be presented by the students whenever it is possible; this opportunity should be viewed by students as a "reward" for having done a good work. From the institution perspective this should be considered as a valuable contribution to the overall quality and good name of the degree. For this reason it was possible, until now, to find institutional financial support for this activity, with the reasonable constraint of submitting papers only to conferences which imply low travelling expenses.

## 4. Bibliography

Knowing how to search for and use bibliography is an important part of any course, however at MSc level it is fundamental. At this level lectures are mainly meant to give the basic underlying theory and point out important issues, not to present them in detail; for these reasons a good bibliography is in fact fundamental, thus a select bibliography is given to

the students and the importance of having the capacity of obtaining new and adequate bibliography is stressed.

Due to the fact that Visualization is a relatively new discipline, no text books seem to be available (at least in the sense as they exist in other longer established disciplines as Computer Graphics or Digital Image Processing). To overcome this type of difficulty usually two alternatives exist: use several books and papers or write some notes to support the course. The second alternative, in the view of the author, is not interesting at MSc level, the students ought to consult several bibliographic references; however an experienced teacher may write some notes conveying his/her perspective on some of the topics. Since the author does not have enough experience in lecturing this course, even writing some notes seemed out of the question. Thus the book by Brodlie et al. was used as general support for the course. As it is stated in its preface:

" this book proposes a framework through which scientific visualization systems may be understood and their capabilities described. It provides overviews of the techniques, data facilities and human computer interface, that are required in a scientific visualization system.... the ways in which scientific visualization has been applied to a wide range of applications are reviewed"; and it seemed comprehensive enough to be used as a "guideline".

However since it is not a recent and detailed text, some other more up to dated or more detailed references were used as support for several topics as already mentioned in section II. Attempting also to provide the students with a bibliography that can be useful in their future activities, the author selected a small set of bibliographic references that covers the subjects addressed in the lectures. In the next sections the general usage of these bibliographic references is described and a commented bibliography is provided. A list of some other references which have been useful, for instance to support practical assignments, is also provided along with brief comments.

During the lectures, but mainly during the preparation of their assignments, students are strongly advised to become familiar with other possible sources of interesting references such as journals and conference proceedings as well as the different methods of getting them (as in libraries, throughout data bases, Internet). In fact, the University Library provides an information system which allows the remote bibliographic search in its databases, other portuguese databases and in services as INSPEC, which students are urged to use for their bibliographic research.

### 4.1. General usage of the selected bibliography

The first bibliographic reference (of the list presented in 4.2) is, as already mentioned, the one that will provide general guidance for the course. The following three references of the same list can also be used as support for several topics, as definition and goals of Visualization, overview of visualization applications, techniques, data characteristics, visualization products and case studies.

The most interesting reference by Tufte can be used to support the study of the history and principles of Visualization (independently from using computers).

The reference on graphics file formats can be used for the overview on the relevant issues to file formats and data compression.

The next five references can all be considered as text books of the so called enabling technologies of Visualization: Computer Graphics, Human Computer Interaction and Digital Image Processing; thus they can be used by the students to obtain a background on those technologies.

Finally, the last reference can be used to support the introduction to the Human Visual System.

### 4.2. List of main bibliography

- Brodlie, K., L. Carpenter, R. Earnshaw, J. Gallop, R. Hubbold, A. Mumford, C. Osland, P. Quarendon, *Scientific Visualization, Techniques and Applications*, Springer Verlag, 1992

This book was written to be a reference guide for the Visualization community on the technical aspects. A framework is described and used to present techniques, data characteristics, products and applications. In spite of not being up to dated in certain aspects it still gives a good overview of the main issues that should be addressed in an introductory course on Visualization.

- Brown, J., R. Earnshaw, M. Jern, J. Vince, *Visualization, Using Computer Graphics to Explore Data and Present Information*, John Wiley, 1995

Provides background on the field of visualization giving an overview of design issues, visualization market and various visualization products. It illustrates a wide variety of real-world applications through case studies.

- Keller, P., M. Keller, *Visual Cues*, IEEE Computer Society Press, 1993

It is intended to people confronted with the problem of discovering the meaning of their data sets. Using practical examples from many disciplines, it illustrates visualization techniques, tips and rules of thumb, that help to produce informative images.

- Scott Owen, G. et al., *HiperVis-Teaching Scientific Visualization Using Hypermedia*, (on-

line), ACM SIGGRAPH Education Committee, http://www.education .siggraph.org, 1996

It is a hypermedia document under development which addresses the fundamental topics of Scientific Visualisation.

- Tufte, E., *The Visual Display of Quantitative Information*, Graphics Press, 1983

It is concerned with the design of statistical graphics as well as with how to communicate information through the simultaneous presentation of words, numbers and pictures. It reviews the graphical practice in the last two centuries and seeks to account for the differences in quality of graphical designs.

- Brown, C., *Graphics File Formats: reference guide*, Manning Publications, Prentice Hall, 1995

It is a comprehensive guide to file formats used in computer graphics and related areas (including visualization). It discusses implementation and design of file formats focusing on the basic issues for its evaluation and development (as data types, organization and compression).

- Foley, J., A. van Dam, S. Feiner, J, Hughes, *Computer Graphics: Principles and Practice*, 2nd ed., 1990

This is considered the standard reference in Computer Graphics. It deals with the fundamental topics of this area in adequate depth, as well as with many others.

- Watt, A., F. Policarpo, *The Computer Image*, Addison Wesley, 1998

An updated coverage of subjects from the three fields of computer imagery which have previously only appeared in separate texts from Computer Graphics, Image Processing and Computer Vision in a coherent overview.

- Shneiderman, B., *Designing the User Interface-Strategies for Effective Human Computer Interaction*, 3rd ed. Addison Wesley, 1998

Provides a complete and current introduction to user interface design. It offers practical techniques and guidelines taking also great care to discuss underlying issues and to support conclusion with empirical results.

- Dix, A., J. Finlay, G. Abowd, B. Russell, *Human Computer Interaction*, 2nd ed., Prentice Hall, 1998

It is a text book in its area, providing a multidisciplinary perspective of the subject. It covers the basic psychology and computer technology involved and the interface between them, as well as usability and more advanced topics.

- Gonzalez, R. C., R. E. Woods, *Digital Image Processing*, Addison Wesley, 1992

It is commonly used as a text book in its area; it covers the fundamental concepts and methodologies for Digital Image Processing.

- Fishler, M., O. Firschein, *Intelligence, The Eye, the Brain and the Computer*, Addison Wesley, 1987

Uses an integrated approach on human and machine intelligence, using knowledge from several areas as computer science, cognitive science, linguistics, biology anthropology and psychology.

### 4.3. Other bibliography

Other bibliographic references have been used to support either specific topics of the lectures or practical assignments. These references include papers published in journals and proceedings of conferences as well as books. A list of some of the books (the ones the author considers may be most useful to a reader of this work) is included.

- Tufte, E., *Envisioning Information*, Graphics Press, 1990

Presents a collection of exemplary designs representing all types of information, widely distributed in time and space, through which design excellence in complex data representation is identified and explained. Includes an interesting chapter on how to use colour.

- Earnshaw, R., N. Wiseman, *An Introductory Guide to Scientific Visualization*, Springer Verlag, 1992

A book intended for readers new in the field, gives a quick summary of what Scientific Visualization is and can do. Can be indicated as an easy to read introduction to the subject.

- Travis, D., *Effective Color Displays: Theory and Practice*, Academic Press, 1991

Addresses perception, displays and colour models. Provides practical guidelines to the effective use of colour, without disregarding theoretical foundation. It is a useful text for designers wanting to use colour.

- Kaufman, A., *Recent Trends in Volume Visualization*, 2nd IEEE EMBS International Summer School, June 1996, Berder Island, France, 1996

Provides a survey of volume visualization and its trends with a focus on biomedical applications.

- Rosenblum, L., R. Earnshaw, J. Encarnação, H. Hagen, A. Kaufman, Sklinenko, G. Nielson, F. Post, D. Thalman, *Scientific Visualization Advances and Challenges*, IEEE Computer Society, Academic Press, 1994

Demonstrates techniques, examines diverse application areas and addresses relevant issues in Scientific Visualization.

- Grinstein, G., H. Levkowitz (eds.), *Perceptual Issues in Visualization*, Springer Verlag, 1995

Addresses issues in the field of applied perception, provides a portrayal of the problems that can be

addressed towards increasing the effectiveness of information displays.

- Lichenbelt, B., R. Crane, S. Naqvi, *Introduction to Volume Rendering*, Prentice Hall, 1998

Provides an introduction to volume rendering concepts and presents an organised logical progression through the volume rendering pipeline; it also contains a CD-ROM with source code which results very useful for practical assignments in the subject.

- Murray, J., W. VanRyper, *Encyclopedia of Graphics File Formats*, O'Reilly Associates, Inc., 1994

Provides detailed technical information on nearly 100 file formats; it also includes chapters on graphics and file format basics.

## 5. Conclusions

Offering this course has not been an easy task for the author, which have encountered several difficulties; she hopes this work may give a positive contribution to whom it may be interested in preparing a course on visualization.

Perhaps the first and main difficulty encountered in the preparation of the course is related to the lack of text books offering comprehensive approaches to the field; this implies not only a great effort in searching for adequate bibliography but also (and perhaps more important) a great effort of organisation and systemisation of the topics to address. This problem can be tackled with the (direct or indirect) help of other more experienced educators and working-groups in the area which publish papers, organise workshops and maintain on-line sites containing useful information (as SIGGRAPH, EUROGRAPHICS, AGOCG). After the initial effort of putting together a curriculum that makes sense, it is necessary to decide on the best approach for the kind of students one is expecting to encounter, to collect and choose the specific materials to the lectures and last (but not least) to choose the subjects and type of practical assignments and the evaluation to be used. All these choices have to be re-evaluated every time the course is offered based on the accumulated experience of the teacher and feedback from the students. The search for new up-to-dated bibliography as well as the choice and preparation of practical assignments which could result interesting have been the greatest concerns of the author of this work. These concerns are also expected to be the main concerns in the years to come since the proposed curriculum is based on the curricula of similar courses offered in several European and American Universities and it is expected to evolve mostly in the specific way each topic is addressed, bibliography, sequence or duration of different topics, rather than on its overall structure.

## References


[1] Beatriz Sousa Santos, "An Introductory course on Visualization", *Revista do Departamento de Electrónica e Telecomunicações (DETUA)*, Universidade de Aveiro, Vol. 2, N.1, Setembro 1997, pp.13, 16

[2] Domik, G.,"Education for Visualization - Activities of the acm-siggraph Education Committee", *Proceedings of the Third Eurographics Workshop on Graphics and Visualization*, Maastricht, August, 1995

[3] Scott Owen, G. S., M. Bailey, K. Brodlie, G. Domik, "Issues in Education for Visualization", *Proceedings of IEEE Visualization 95*, Atlanta, October, November, 1995, pp. 427-430

[4] Scott Owen, G. et al., HiperVis, ACM SIGGRAPH Education Committee (on line), http://www.education. siggraph.org, 1996

[5] Banks, D. , "Including Graphics and Visualization Research in a Master's-level Course", *Computer & Graphics*, Vol. 21, n. 3, 1997, pp.379-387

[6] Beer, D (ed.)., *Writing and Speaking in the Technology Professions*, a practical guide, IEEE Press, 1991

[7] *Referencing*- (In-text-referencing) (on line), http://www.canningcollege.wa.ed.au/reference (Jan, 1999)


## Acknowledgement


The author wishes to thank Prof. Gitta Domik, chair of the ACM-SIGGRAPH Education for Visualization Committee, for her great help in the initial definition of the curriculum and bibliography of the presented course, as proposed in 1996.

# Restructuring Comp u ter Graphics and Visualisation Curricula at IST

**J. M. Brisson Lopes, M. R. Gomes, J. C. Bernardo, J. Jorge** and **J. M. Pereira**
Department of Informatics Engineering, Instituto Superior Técnico,
Av. Rovisco Pais, 1049-001, Portugal
brisson@ist.utl.pt, {mrg, jcb, Joaquim.Jorge, jap}@inesc.pt

**Abstract**

*This paper addresses the restructuring of the undergraduate and graduate programs in Computer Graphics, Visualisation and Multimedia that was recently carried out at Instituto Superior Técnico. The paper details the rationale for the changes in both curricula and course content and describes the new structure. The result is a coherent and complete series of courses spanning the undergraduate and M.Sc. degrees in Informatics Engineering. It further discusses the structure of the basic course on Computer Graphics within this context and draws conclusions to contribute to the ongoing discussion on Computer Graphics, Visualisation and Multimedia curricula.*
**Keywords:** *course curricula, Computer Graphics, multimedia, undergraduate programs, graduate programs.*

## 1. Introduction

The Informatics and Computer Science undergraduate program at Instituto Superior Técnico (IST) started in the academic year of 1989/90 under the aegis of three departments, Electrical Engineering, Mathematics and Mechanical Engineering, which shared the co-ordination of the new degree program. Roughly at the same time the M.Sc. program in Electrical Engineering and Computers offered a specialisation in Computing. Courses on Computer Graphics were taught within the undergraduate and graduate programs.

This scenario lasted until 1998 when IST decided to create a new department, the Department of Informatics Engineering (DEI). The decision was taken given the need for better co-ordination of the undergraduate program which had come of age and the number of teaching staff and undergraduate students involved. The new department was officially established September 1st, 1998.

DEI´s creation started a new and extremely dynamic process. The novel Department's professors engaged in an overall revision of the department's teaching activities with three purposes in mind. First, previous experience had shown an effective need for adjustments in the undergraduate program syllabi. Second, the new department needed to assess in detail and demonstrate its requirements in terms of teaching staff. Finally, the revision was necessary because of the continued changes in Information Technology.

At the same time this dynamic process had implications in the graduate curricula. The result was the decision to create a new autonomous M.Sc. program addressing the needs of graduate student in the scientific area of the new department and related areas.

By and large, this constituted a unique opportunity to address all these matters from a global perspective. Instead of taking the classical approach and address undergraduate and graduate levels separately, this was the time when the two programs could be changed together to make the graduate program the logical sequence of the undergraduate program.

The outcome was the adoption of a model currently adopted in other universities, notably in the United States. This model allows undergraduate students intending to continue their studies through the graduate program to take advanced courses from the M.Sc. program while in their senior year. This concept became the Integrated M.Sc. Program.

After restructuring the undergraduate program, all scientific areas in the Department of Informatics Engineering were asked to set-up their particular graduate (M.Sc.) curricula. Later on, the integrated undergraduate-graduate scenario was finally revised and approved. In this process the Computer Graphics and Multimedia professors of DEI took the opportunity to assess and reprogram all courses within their responsibility.

This paper reports the results of that work. Section 2 lists the objectives set at the beginning while section 3 presents the rationale and describes the new program structure for the area of Computer Graphics and Multimedia. Section 4 addresses in detail one of the basic components of the program, the Computer Graphics course. The final section presents some remarks and general conclusions.

## 2. Objectives

In Portugal, as in most other countries, the supply of information engineering professionals has not kept pace with job market needs. Besides the hiring frenzy driven by the arrival of the Euro and Y2K endeavours, many new Portuguese companies have been created in recent years, with some of them involved in international activities such as ParaRede and EasySoft, to name a few. This shows clearly that there is a need for new and highly qualified professionals at all levels. Furthermore, the continuous development of information related technologies, of which the WWW is a good example, requires professional skills that were not taught a few years back. Furthermore, currently employed professionals are required to continuous update their qualifications in order to be on par with technological development.

Therefore, the first goal of the new curricular structure was to produce an undergraduate program that would satisfy the above needs. The second goal was to create a specialisation area in Interactive Systems and Multimedia suited to the needs of future professionals. Another important goal was to set the groundwork for a doctoral program in Interactive Systems and Multimedia.

Since the topics to cover are interrelated, we decided to develop an integrated approach spanning both undergraduate and graduate programs, in order to:

- offer students a coherent global program in Computer Graphics, Visualisation, Multimedia and Interactive Systems.
- establish a complete and logical sequence of subjects and courses extending from early undergraduate to advanced graduate studies.
- allow students intending to obtain a M.Sc. degree to proceed smoothly and quickly to graduate studies by shortening the intermediate steps required.
- empower students not intending to pursue graduate studies to achieve a high degree of proficiency.
- enable students from other undergraduate programs to enrol in our graduate program.

## 3. The M.Sc. Program in Interactive Systems and Multimedia

This section presents the Master's program in Interactive Systems and Multimedia, its guidelines and structure, together with the outline of each course's syllabus.

### 3.1. Rationale

This following addresses the general guidelines for the new program organisation. We will discuss the requirements at undergraduate level before moving on to the graduate program in Interactive Systems and Multimedia.

All undergraduate students must become familiar with the basic principles of Computer Graphics. These minimum requirements comprehend both two-, three-dimensional, surface and hierarchic modelling and use of colour. Furthermore, most undergraduate students will take courses on Computer Aided Design and Human-Computer Interaction. These complete the knowledge requirements at the undergraduate level and are the basis for graduate level studies. Any student applying to the master's program must have the above basic Computer Graphics background. If not, they will have to complete it before moving to the master program by taking the missing courses.

Students at the end of the fourth year of graduation may elect to proceed directly to the master's program. Instead of moving on to the fifth and final year of the undergraduate degree, they can take advanced courses from the master's program first year.

The master's curriculum in Interactive Systems and Multimedia includes courses from two major areas, Computer Graphics and Information Systems. Students intending to specialise on the design, implementation and use of Computer Graphics Systems opt for a major on Computer Graphics and a minor on Information Systems. Conversely, students whose interest lies on Interactive Systems can opt for a major in Information Systems and a minor in Computer Graphics.

Students earn credits from courses in these two areas. Each major begins with a fundamental course offered during the first semester of graduate studies. After that students must take at least four courses in their major area. Of these four courses, two intermediate curses on the second semester are required studies for the two advanced courses on the third semester.

A master's thesis to be done during the second year completes the graduate master program. The thesis must be supervised by a professor of DEI's Computer Graphics & Multimedia scientific area.

Students choosing a major in Computer Graphics Systems must take the Three-dimensional Modelling and Visualisation course in the first semester. During the second semester, students will take two more specialised courses, one on Visualisation and the other on Modelling.

Students choosing a major in Interactive Systems must take the course on Human-Computer Interaction in the first semester. These students will then have to take the two more specialised courses on Hypermedia and Intelligent Interfaces in the second semester.

In the third semester, students can choose two courses from the five advanced courses available. These are Visual Languages, Virtual Reality, Scientific Visualisation, Geographic Information Systems and Advanced Topics on Computer Graphics and Multimedia.

### 3.2. Program Structure

The courses in the new structure were classified into four categories: G, L, M and A.

G courses are typical undergraduate courses and constitute pre-requisites for enrolling into the master program. The Computer Graphics course is part of the Informatics and Computer Science undergraduate program. Other related undergraduate programs have the option for the Computational Graphics Methods course, which is a version of the Computer Graphics course adapted to such other undergraduate programs.
L courses are offered both as part of the undergraduate program, and of the first semester of the graduate master program

M courses take place in the second semester and are typical master program courses. Completion of the corresponding L courses is necessary to take any M type course. M courses are pre-requisites for all A type courses.

Type A courses are final master's program courses addressing advanced topics and take place during the third semester.

Figure 1 presents a diagram showing the courses and their pre-requisites. Table 1 lists the undergraduate and graduate courses according to type, curricular semester and year.

### 3.3. Syllabi

This section presents the abridged syllabi of the courses listed above. The **Computer Graphics** course is presented in more detail in the next section. The reason for this is the fact that this course is the introductory fundamental course for the whole area of Computer Graphics and, therefore, deserves special attention.

The first semester course on **Three-dimensional Modelling and Visualisation** aims at teaching students how to create and use different 3D representation models, model scenes and animate objects. The curriculum also addresses visualisation techniques to cope with specific requirements like games. This course is project oriented.

| Course Designation | Type | Yr. | Sem. |
|---|---|---|---|
| Computer Graphics | G | 2 | 2 |
| Computational Graphics Methods | G | 5 | 1 |
| 3-D Modelling and Visualisation | L | 5 | 1 |
| Human-Computer Interfaces | L | 4 | 1 |
| Intelligent Interfaces | M | | 2 |
| Hypermedia | M | | 2 |
| Graphical Modelling Complements | M | | 2 |
| Visualisation Complements | M | | 2 |
| Visual Languages | A | | 2 |
| Virtual Reality | A | | 3 |
| Scientific Visualisation | A | | 3 |
| Graphical Information Systems | A | | 3 |
| Adv. Topics in CG and Multimedia | A | | 3 |

Table 1. Undergraduate and graduate courses according to type, curricular semester and year.

The course on **Human Computer Interaction** (first semester) is intended to teach students how to create and evaluate user interfaces. To achieve these goals, students learn the concepts of Human Factors (cognitive and motor-sensorial), and Usability and how to use and apply different software methodologies and tools for interface creation. This course is project oriented.

The second semester course on **Intelligent Interfaces** explores new paradigms in user interaction that will replace present-day approaches based on direct manipulation and the desktop metaphor. The course discusses multimedia input analysis; recognition of speech, gestures, gaze and calligraphic sketches; synergetic combination of multiple interaction modalities; the architecture and construction of novel user interfaces; presentation of multimedia information; agent-based interfaces and natural language.

The course on **Graphical Modelling Complements** course ($2^{nd}$ semester) addresses a large set of topics related to the design and implementation of generic geometric algorithms in both 2D and 3D. Starting with a comparative study of several solutions to some classical and elementary geometric problems, it then progresses to the description of more complex geometric modelling techniques. The course then presents suggestions for the application of those results to solving practical problems in different domains such

as Virtual Reality, Computer Aided Design, Computer Vision, Robotics, etc.

The second semester course in **Visualisation Complements** is an exposition of state of the art techniques in 3D rendering such as antialiasing theory, texture mapping, ray-tracing advanced strategies, radiosity methods or volume rendering. The motivation of this course is to explain the many techniques that have evolved, with sufficient theoretical rigour and in enough detail to enable their use in a project on realistic image synthesis.

In the third semester, the course on **Visual Languages** focuses on a class of formal languages based on the non-linear organisation of information. The relevance of such languages is reflected on the proliferation of systems such as spreadsheets, CAD and GIS systems, which structure information in an essentially visual manner. The course establishes the connection and contrasts visual to conventional formal languages, balancing a discussion of theory with the study of practical applications.

The objective of the third semester **Virtual Reality** course is to provide a comprehensive technical review of the engineering details underlying the development of virtual environment displays. Students should not only become familiar with concepts like immersion and stereo rendering but also understand existing virtual reality systems and their associated technologies. Special focus is given to the Web visualisation. Students will then explore this topic through a project where they must apply VRML and Java technologies.

The advanced course on **Scientific Visualisation** (third semester) addresses the architectural solutions found in such systems and the means and ways of achieving realistic scientific data visualisation applied by both traditional and immersive visualisation systems, including real-time visualisation and user interaction. This course is project oriented.

In third semester advanced course on **Geographic Information Systems** the students will apply previous knowledge on Organisation Models, Information Systems, Interactive Systems and Computer Graphics to study not only the technology (information access, information publication, specific methodologies) but also to study the impact of technology on organisations. Case studies will be used.

The third semester advanced course on **Advanced Topics on Computer Graphics & Multimedia** program will be defined each year and will include the latest scientific subjects in the field.

## 4. The Basic Course: Computer Graphics

The **Computer Graphics** course is the entry point to the Computer Graphics world to most students enrolled in the Informatics and Computer Science undergraduate program. For this reason, this paper addresses it in more detail in the following.

Note that the **Computational Graphics Methods** course, that is offered at IST to students enrolled in the Electronics and Computer Engineering and Applied Mathematics undergraduate programs, runs parallel to the Computer Graphics course and addresses roughly the same subjects. However, it follows a more traditional pedagogical approach with adaptations to those programs and in line with those programs' nature.

### 4.1. Objectives and Guidelines

The Computer Graphics course is the basic course at undergraduate level in its scientific area. After this course, students are expected to understand and apply the fundamentals and algorithms in Computer Graphics.

At the same time, this course plays the very important role of attracting students to the area of Computer Graphics. Students are familiar with the latest advances in the area since they have most certainly experienced it in computer games and while browsing the World Wide Web (e.g., VRML). Moreover, what used to be expensive hardware and heavy applications can be found today in multimedia PCs at a low cost. Most basic algorithms come already embedded in low cost hardware.

From the above, it is evident that the traditional pedagogical approach to a basic course in Computer Graphics is no longer on par with the today's reality and can be a negative factor in keeping students interested in the subject. This is why we decided to follow a different pedagogical approach that looks at Computer Graphics from a global structured perspective using today's typical applications and then proceed to more detailed levels.

### 4.2. Course Contents

A typical semester consists of thirteen weeks with three hours of lecture time per week at IST. In line with was previously said we propose to develop the course curriculum of the introductory Computer Graphics course as in the following course outline (the numbers in parentheses indicate the number of classes dedicated to each subject).

1. Introduction and General Concepts [2]
2. Geometric Transformations [3]
3. 3D Visualisation [7]
   3.1. Projections and the Rendering Pipeline
   3.2. Geometric Modelling and Object Hierarchy
   3.3. Visualisation Algorithms
   3.4. Clipping, Scan Conversion, Hidden Line and Surface Removal
4. Parametric Surfaces [2]
   4.1. Representation
   4.2. Tessellation
5. Colour models, halftoning, reproduction [2]
6. Shading and Illumination [2]
7. Ray-Tracing and Radiosity [2]
8. Animation [2]

We will use the classic textbook "Computer Graphics and Applications" by Foley et. al.[1]. This comprehensive classical textbook has been traditionally used at IST although it leaves much to be desired from the pedagogical point of view. We are currently looking for alternatives.

### 4.3. Other Features of the Curriculum

The practical pragmatic aspects of programming and algorithm implementation are relegated to the weekly laboratory and practice classes (two hours per week) instead of recitation sessions.

The study and description of 2D windowing systems and raster primitive operations, already reduced in previous editions of the course, disappears as autonomous curriculum item. The discussion of two-dimensional primitives appears at its logical place in the discussion of the visualisation pipeline, i.e., after the discussion of the scan-line algorithm conversion.

We introduce students to modelling languages such as VRML to construct a fairly complex scene as their first assignment. The second laboratory project focuses on the study, implementation and experimentation of three-dimensional visualisation algorithms, building on the enthusiasm students traditionally vow to such activities.

## 5. Conclusion and Final Notes

This paper presented the new program structure in Computer Graphics and Multimedia of the integrated undergraduate and graduate (M.Sc.) programs in Informatics and Computer Science at Instituto Superior Técnico, Technical University of Lisbon. We addressed the aims of the novel integrated program and described the curricula of both programs' courses, paying special attention to the introductory course in Computer Graphics.

The new curricula are the outcome of collaboration work carried out by the professors of the DEI's Computer Graphics and Multimedia scientific area and follow the experience gained by similar initiatives already in place in other countries, especially in the United States. In our view, the program and curricula proposed here constitute a well balanced and comprehensive set, covering both the fundamentals and advanced topics in this scientific area in a progressive manner. Furthermore, the advanced topics proposed are well in line with current research trends in our field.

The proposed program is also clearly in tune with modern requirements for competent Computer Graphics professionals at both undergraduate and graduate (M.Sc.) levels. Moreover, the program ensures a smooth transition from undergraduate to graduate level with minimum delay.

The program will be set in place at the beginning of the 1999/2000 academic year and is expected to be in full operation by the year 2001.

## References

[1] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, *Computer Graphics, Principles and Practice*, 2nd. Edition, Addison-Wesley, 1990.

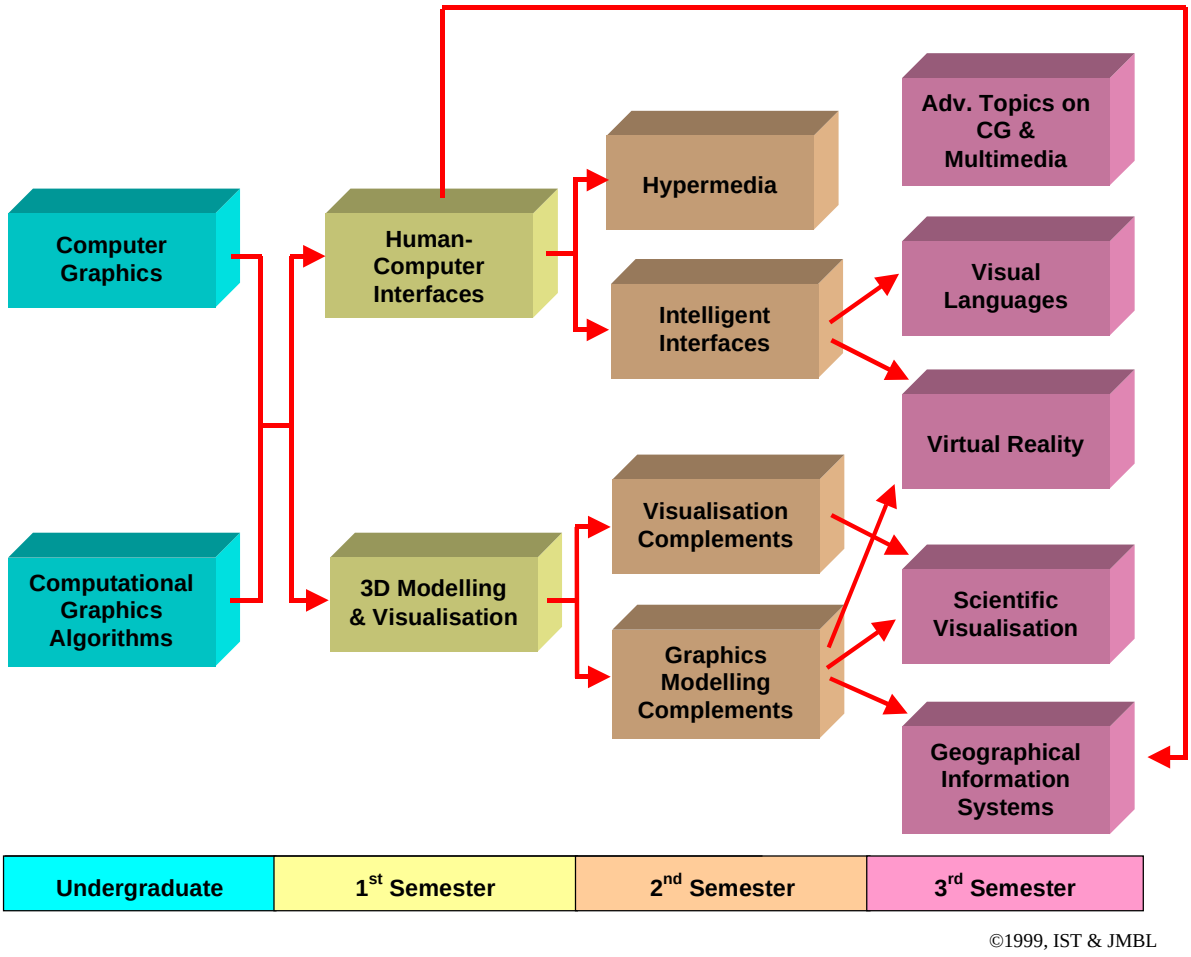



Figure 1. Course diagram showing course dependencies.

# Adapting Computer Graphics Curricula to Changes in Graphics

**Lewis E. Hitchner**
Dept. of Computer Science
California Polytechnic State Univ.
San Luis Obispo, CA U.S.A.
hitchner@csc.calpoly.edu

**Henry A. Sowizral**
Graphics Division
Sun Microsystems, Inc.
Menlo Park, CA U.S.A.
henry.sowizral@eng.sun.com

**Abstract**

*Introductory computer graphics courses are changing their focus and learning environments. Improvements in hardware and software technology coupled with changes in preparation, interest, and abilities of incoming students are driving the need for curriculum change. Past courses focussed on low- and intermediate-level rendering principles, algorithms, and software development tools. Many of these algorithms have migrated into hardware. Though important knowledge for advanced graphics programmers, most graphics applications programmers have no need to study at this level, much as application programmers have no need to study hardware systems or assembly level programming. Courses need to focus on intermediate- and high-level principles, algorithms, and tools. A fundamental need in modern graphics curricula is integration of a 3D graphics API into the instruction. This paper presents experiences teaching this focus with both low and high level graphics programming API's. The experiences were gained in courses at an undergraduate university and in multi-day industrial courses for experienced professional programmers.*
**Keywords:** *computer graphics, curriculum.*

## 1. The Changes in Directions of Introductory Computer Graphics Courses

The curriculum of courses for the study of computer graphics has changed and continues to change in response to technology and student needs. We first review graphics technology, students, and curricula, as related to computer graphics instruction during the three stages of acceptance of a new technology (ignoring the outlier stages): that of the "Early Adopters", that of "Early Majority", and that of "Late Majority"[**1**]. Next a set of principles and goals for a new curriculum for introductory computer courses is presented. We follow this with examples of practical experience in teaching such a course at a polytechnic state university and in multi-day short courses for practising professionals.

### 1.1. Graphics Technology, Students, and Course Curricula in the "Early Adopters" Era

From the 1970's through early 1980's computer graphics courses existed primarily at large universities. Computer Science education was still in its infancy and courses in computer graphics were rare. They were usually specialty courses at a few schools that could afford expensive graphics hardware, usually funded by government research grants. During that time the curriculum emphasis was on recent developments in research that had not yet entered the public domain or commercial software worlds. Research focussed on developing new techniques, more functionality, and algorithms that performed more efficiently.

Courses taught during this era assumed the following about students and graphics systems:

- students were highly motivated technologists with advanced science and mathematics preparation (computer science, mathematics, and engineering majors),
- most students had little or no experience with the new field of 2D and 3D graphics,
- students should learn all the fundamental principles, algorithms, and techniques that researchers had recently developed,
- graphics hardware devices were not widely available, and they depended upon special purpose software interfaces.

The curriculum of courses during the pioneer days reflected these assumptions. Typical topics included:

- an overview of graphics technology and applications,
- introduction to low level graphics hardware: CRT's, input devices, processors, peripheral devices,
- algorithms for performing basic rendering pipeline operations:
  - line and circle drawing (e.g., Bresenham's Algorithm),
  - line and polygon clipping,

    - matrix transformations (primarily for 3D viewing, though also for modeling),
    - lighting and shading algorithms, and
    - image generation algorithms (polygon scan conversion, visible surface computation, Z-buffer)
- introduction to 3D modeling: parametric curve and surface generation, some procedural techniques,
- software technology for interfacing graphics processors and displays to mini-computers and mainframes using locally developed (one-of-a-kind), or proprietary commercial, software function libraries.

Most graphics textbooks used during this era reflected these assumptions. University courses followed these assumptions and covered topics with the same emphasis as stressed in popular textbooks. During this era there was also somewhat of an elitism. Only computer scientist-technologists with advanced university degrees were capable of developing (or even using) complex graphics display programs.

### 1.2. Graphics Technology, Students, and Course Curricula in the "Early Majority" Era

By the early 1980's, graphics technology became lower priced and more widely available. Graphics workstations, such as those made by Evans and Sutherland, SGI and other vendors, provided high-powered processors for those schools able to afford them. Graphics peripheral processors (e.g., the AED 512) and early versions of PC graphics systems also became available at somewhat more reasonable prices.

Early attempts at graphics software standardization occurred during this era. Some API's were short lived or never widely used (e.g., the "original" graphics standard, CORE), or were extended and revised as new needs and features evolved (e.g., GKS to GKS-3D, PHIGS to PHIGS+). Such moving targets were not very successful at becoming widely adopted. Proprietary graphics software packages dominated the industry.

Students during this era were somewhat similar to those of the earlier era. However, the popularity of computer science as a career attracted many more students from more educationally diverse backgrounds. Many graphics students during this era had weaker math and science knowledge than earlier students. Fundamental topic areas of graphics (geometry, physics of color and lighting, graphics hardware device operation) were unfamiliar to them, and had to be covered in course curricula. More students had some familiarity with using 2D (and some 3D) graphics in video games.

Although graphics software was more standardized, educators and textbook authors were wary of teaching students the "API du jour". They avoided teaching standards that were likely to be soon out of date. Instead, most courseware developers designed their own generic, "standard-like" software packages to use in accompaniment with a particular text or graphics hardware system. It was also common for a university or individual instructor to write his or her own graphics package (often publicly shared with other universities via anonymous ftp in the pre-WWW days). Unfortunately, these "teaching API's" were often minimal sets of functions. They were sufficient for learning the principals of graphics software development, but were not representative of real-world, industrial-strength packages. And, of course, knowledge of them was not portable to another school or to a job. However, since most student assignments were small, one-shot programs and never lived long enough to get to the maintenance phase of their life cycle, use of a "toy" API was acceptable. That met the goal of providing the necessary learning experience. Of course, after graduation, the employers of these students had to train them to use "real world" graphics software systems.

In response to the availability of and interest in graphics technology, many computer science departments adopted graphics as a part of their curriculum. The ACM Curriculum '91 [**2**] specified a graphics course as a technical elective. The use and development of graphics applications had now become accessible to any computer, math, engineering, or science student who enrolled in a university graphics course.

### 1.3. Graphics Technology, Students, and Course Curricula in the "Late Majority" Era

Current, late 1990's graphics hardware is orders of magnitude faster and cheaper, and it is much more robust and fully functioned, than earlier technology. The prevalence of personal computers with low cost, high-performance 3D graphics accelerators, is the dominant technology. However, extreme increases in performance to price ratios have occurred for products at all pricing levels.

Furthermore, graphics hardware technology has evolved to a state where rendering algorithms are

specialized for high performance. Certain aspects of geometry representation, lighting and shading, and interaction handling are primarily implemented within restricted preconditions rather than general purpose ones. For example, most modern graphics rendering hardware either requires all geometry to be represented using triangles only or else is tuned for highest performance using triangles. Also, some current 3D graphics hardware systems implement the entire OpenGL rendering pipeline in hardware.

Nearly as great an increase in performance and functionality has occurred for graphics software. The most significant aspect of software evolution has been the emergence and acceptance of standardized graphics software API's (Application Programmer Interfaces). Today there are several standard 3D graphics API's used in industry. The three leading, nearly universally used "low level" API's are OpenGL (evolved from SGI's proprietary GL and now available on nearly all platforms), Microsoft's Direct3D (Windows platforms), and Apple's QuickDraw 3D (Apple platforms). There are also many high level 3D API's, though currently there is not the dominance by one or two as there is at the low level.

In addition to graphics API's there are four other significant factors in graphics software development:

1. Powerful, low cost IDEs (Integrated Development Environments) for rapid and easy code development and debugging. For example: Microsoft and Imprise (formerly Borland) products.
2. Extensive libraries of general purpose support functions that provide higher level graphics capabilities and simplified GUI development, most of which are either free or very reasonably priced. For example: GLUT (GL Utility Library), MFC (Microsoft Foundation Classes).
3. A number of powerful software packages for high level, application independent development such as ray tracing and animation systems (many of which are public domain). For example: POV Ray.
4. The World Wide Web and its wealth of free and instantly accessible demos, software tools, data sets, and human resources (via email, newsgroups, and chat sessions).

Graphics students of today are quite different from those in earlier years. Some characteristics are:

1. Most technical university students (i.e., computer science, engineering, math, or science students) enter a first course in computer graphics with significant prior exposure to graphics concepts, such as lighting, perspective, and 3D viewing and navigation. Many students have had significant programming experience, and it is not uncommon for them to have already written 2D and 3D graphics programs.
2. Although most students come to school with greater computer skills and graphics experiences than their predecessors, on the other hand, many arrive with greater handicaps. Even among engineering and computer science students, their mathematical, problem solving, and logical reasoning skills appear weaker than in prior years.

In the opinion of these authors, contemporary curricula of most university level graphics courses and textbooks do not appropriately emphasize the most critical aspects of graphics hardware technology, nor have they appropriately adapted to the backgrounds of their students.

It is no longer necessary to introduce today's students to such topics as a pixel, a bit-mapped image, a color palette, or basic RGB color systems. For many, even more advanced topics such as 3D viewing, 3D navigation, and texture mapping are familiar. Avid, even casual, web-surfers or PC gamers of today are familiar with using these terms and features.

Many fundamental algorithms and procedures of graphics are no longer relevant, even though they are pedagogically valuable. For example, Bresenham's line and circle algorithm and the polygon scan conversion algorithm are not relevant. Such operations are done in very low level hardware (e.g., microcode or ASIC's) and even there, traditional algorithms are not always used. Clipping operations are also buried deep within the hardware. Cohen-Sutherland line clipping and re-entrant polygon clipping algorithms are good intellectual exercises for today's students, but have less relevance to helping them learn useful graphics software development techniques.

Now that small set of graphics API's are nearly universally used (and also becoming more similar), students should learn to apply the fundamental principles of graphics within these environments, not within make-believe, "toy" software environments. A few recent textbooks integrate modern API's, and one of them [**3**] has been adopted by a number of universities. However, to our knowledge no textbooks fully meet the criteria of presenting

graphics technology in the context of today's hardware and software systems.

Regardless of criticisms of graphics education weaknesses, the accessibility and usability of computer graphics technology today is truly phenomenal. No longer are those who study and use graphics members of an elite club. Nor are they restricted to university engineering and science students. We now have experienced the ”democratization” of computer graphics.

## 2. New Models for Introductory Graphics Courses: SIGCSE and SIGGRAPH Panels

In response to the need for modernizing the curricula of graphics instruction, a panel at the recent ACM SIGCSE (Assoc. for Computing Machinery, SIG Computer Science Education) Technical Symposium [**4**] chaired by one of this paper's co-authors, discussed and presented a proposal for contemporary computer graphics curricula. In addition to the panel, a Birds-of-the-Feather session co-sponsored by the SIGGRAPH Education Comm. at the same conference held discussions following the panel. A similar panel with three of the four same panelists has been submitted and accepted for presentation at the ACM SIGGRAPH '99 conference (Los Angeles, August 1999).
This panel proposed the following philosophy of the first graphics course:

- Computer graphics is inherently 3D and courses should be also.
- The fundamental subject of a computer graphics course is geometry and how it is expressed in computational terms. Thus, geometry is a major part of the introductory course. Geometry is expressed in terms appropriate to the field, such as coordinate systems, transformations, and surface normals. The basic shape is the triangle.
- Computer graphics is intrinsically visual, and even the most technically oriented graphics practitioner must be aware of the visual effects of algorithms. Unlike other areas of computer science, algorithms must be considered not only for time and memory usage, but also for their visual effect.
- Besides geometry, computer graphics is about light and surfaces, and about developing algorithms to simulate their interplay. Courses need to include material about light and surface properties and about the distinction between the ways various algorithms present light and surfaces visually.
- Computer graphics has matured to a state in which there are a small number of high-level API's that support all the fundamental concepts needed for early work. Courses should be built upon this kind of high-level approach.
- Computer graphics should be interactive. Courses should include interactive projects and cover event-driven programming.

## 3. Experiences in Teaching Introductory Graphics Courses

One co-author has taught undergraduate university graphics courses from 1984-1988 at the Univ. of Calif., Santa Cruz, and from 1995 to present at California Polytechnic State University (Cal Poly). The syllabus and topics emphasis of courses has shifted significantly.

The original syllabi were similar to that of the "Early Majority" era model, based upon the same assumptions about available graphics technology and students as stated above. The topic focus was on fundamentals (hardware devices, geometry, viewing, lighting, and shading) and algorithms for implementing the rendering pipeline. Interaction methods were limited to fairly simple event handling, and they used console text i/o without a GUI. Some hierarchical modeling was taught using modeling transforms implemented by the programmer. Early versions of the course used a "home grown" API (implemented on top of a device dependent 2D graphics API). Later versions migrated to proprietary versions of GL (SGI and HP). Typical programming lab assignments included implementation of Bresenham's line drawing, Cohen-Sutherland line clipping, re-entrant polygon clipping, and simple hierarchical modeling applications.
The 1996-98 Cal Poly graphics courses use modern API's. The co-author teaches a version that uses OpenGL for the first half of the course, and uses Open Inventor for the second half [**5**]. In this course the topic focus is also on fundamentals. But, instruction in implementation techniques has shifted to higher level aspects of rendering, event-driven interaction handling with GUI's, complex hierarchical modeling including extensive study of scene graphs, and software design using the two API's. The first lab assignment draws complex 3D primitive shapes with interactive control of color, orientation, and line style. A second lab assignment draws chairs, a table, and a floor and requires interactive control of position and orientation for

each object and for the viewpoint. A third lab uses Open Inventor to model a 16-jointed robot with interactive control for joint manipulation. This lab also requires model development using a graphical scene graph editor without writing program code.

## 4. The Future: Java 3D as a Learning Environment for Introductory Graphics Courses

### 1.4. Overview of Java 3D Design Philosophy and Features

Java 3D is an API used for writing 3D graphics applications and applets. It is a library of basic and utility classes written in the Java language. Java 3D is platform independent and extends Java's "write once, run anywhere" benefits for application developers. It also integrates well with the Internet because applets written using Java 3D have access to the entire set of Java classes. A complete description of the Java 3D design philosophy and its features is available in published texts and online web documents [**6,7,8**].

The Java 3D programming paradigm includes these principles:

- Fully object-oriented implementation
- Classes are extensible and compatible with all other Java 2 Platform libraries
- Scene Graph based for both geometry and behaviors
- High performance a primary design and implementation philosophy:
  - Layered Implementation: Native code based, aiming at hardware acceleration,
  - Application programmer can specify what will change so that system can perform optimization,
  - Supports multiple rendering Modes: Immediate, Retained, Compiled-retained

### 1.5. Justification for Java 3D As A Learning Environment for Beginning Students

Although at first glance Java 3D may seem like an API most applicable for experienced professionals developing production quality software, several aspects make it desirable as an API for beginners.

- **Platform independence**
  Students usually prefer to work in labs at school, on their home computers, and, even at their place of work (for those employed full or part-time). Java and Java 3D's platform independence greatly simplifies portability, not only for 3D graphics but also for 2D GUI's.
- **Cost**
  The purchase price is zero, and there is no software maintenance cost. This is a significant factor for schools, as well as for students.
- **Programming Paradigm**
  The fully object-oriented environment is consistent with training students receive in their prerequisite courses. There is no need to kludge together OO with non-OO procedures.
- **High Performance**
  Students are accustomed to high performance -- or at least what they perceive to be high performance. The many PC games and Web applets that present apparent high performance rendering raise expectations. However, they can cause frustration for students if they are limited to programming lab examples that appear simplistic by comparison. A significant aspect of graphics education is the motivation provided by its appealing real-time, interactive, visual results. If that motivation is frustrated by low performance, the learning that occurs will be diminished.

## 5. Experiences in Adapting Course Curricula to Changes in Graphics Technology

We have put our recommendations into practice. In Spring 1999 one co-author taught a revised version of the Cal Poly undergraduate Introduction to Graphics course using OpenGL and Java 3D APIs. During Winter and Spring 1999 the other co-author taught three courses, varying in duration from one to three days, on Introduction to Java 3D for experienced professionals.

In the current university environment, students are entering graphics courses with less foundational knowledge than in the past. Although most students come to school with much greater exposure to computers and graphics, they have less exposure to mathematics, problem solving, and less of an idea where they want to focus their careers. Furthermore, the field of graphics has expanded greatly since its inception, and given the new APIs, computer graphics is no longer the exclusive province of experts. The challenge for educators is to provide enough information so students can learn graphics without requiring them to becomes masters of a particular sub-discipline. After all, twelve-year-olds are producing beautifully ray-traced images without having any concept of how ray tracing works.

A high level graphics API provides educators with the ability to introduce a broad range of fundamental concepts without detouring into the details important only to graphics professionals. Despite the higher level presentation of concepts, students can quickly learn how to write useful and compelling graphic applications. In the process, they are exposed to enough graphics concepts that they can decide whether to devote more of their student and professional career to graphics.

In the Cal Poly CSC 471 Introduction to Graphics Spring 1999 course [**9**] students used OpenGL and Java 3D APIs, a low level and high level API respectively. These APIs allow topics to be covered quickly in both breadth and depth. Programming assignments using a high level API (Java 3D) allow students to produce quite substantial results within a short period of time. In one programming assignment students wrote a Java 3D program to display a humanoid-like robot with 16 rotational joints and interactive behaviors that allow run-time modification of the joint angles. This project was completed within 3 weeks of their introduction to Java 3D (and, for some students, within 3 weeks of their first exposure to Java after having been trained in C++). At the same time students were learning the fundamental concepts of a scene graph and were solidifying their knowledge of complex transform operations, they also learned the details of Java 3D scene graph implementation.

Experienced computer professionals have significantly more exposure to mathematics, problem solving and logical thinking. However, when they decide to study graphics, they often revert to "student mode". Presentation order, thus, remains important. Fundamental material must be introduced in a constructive order rather than deconstructive order. Otherwise students become confused and frustrated. Although experienced professionals may be able to solve the most complex problems without hand holding -- outside the classroom -- when they become students they require the smallest details to be completely specified. This is not surprising, since much like university students they also cannot distinguish between fundamental concepts and unimportant details.

This becomes more of an issue given the breadth and depth of topics within the graphics field today and the short time frame of a typical professional short course. Despite this challenge, higher level APIs allow educators to introduce breadth as well as to delve into selected topics in depth within short time frame courses. By emphasizing fundamental ideas an educator can start such students in the necessary direction and leave them with sufficient landmarks to allow the students to explore the details further on their own.

## 6. Summary and Conclusions

Graphics technology has changed vastly in the nearly three decades since researcher-educators first started passing their knowledge on to students. Students and computing environments have also changed significantly. Course curricula and textbooks have been slow to adapt and have not kept up with these changes. Proposals for adapting curricula to match modern technological requirements have been presented here and elsewhere. Personal experience with such adaptation in university courses and in industrial short courses has demonstrated the effectiveness of using modern, high-level API's and emphasis upon student-centered learning.

## References:


[1] Geoffrey A. Moore, *Crossing the Chasm*, HarperBusiness, 1991.

[2] Alan B. Tucker and Bruce H. Barnes, editors, *Computing Curricula 1991: Report of the ACM/IEEE/CS Curriculum Task Force*, ACM Press/IEEE Computer Society Press, 1991.

[3] Edward Angel, *Interactive Computer Graphics: A top-down approach with OpenGL*, Addison-Wesley Longman, 1997.

[4] Lewis Hitchner, Steve Cunningham, Scott Grissom, and Rosalee Wolfe, *Computer Graphics: The Introductory Course Grows Up*, Panel session, Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '99), New Orleans, LA, USA, March 24-26, 1999.

[5] Lewis E. Hitchner, CSC 455 course web page, Spring 1998, http://www.csc.calpoly.edu/~hitchner/CSC455.S98

[6] Henry Sowizral, Kevin Rushforth, and Michael Deering, *The Java 3D API Specification*, Addison-Wesley Longman, 1995.

[7] Henry Sowizral, Kevin Rushforth, and Michael Deering, *The Java 3D API Specification* http://java.sun.com/products/java-media/3D/forDevelopers/j3dguide/j3dTOC.doc.html

[8] *Java 3D White Paper*, http://java.sun.com/marketing/collateral/3d_api.html

[9] Lewis E. Hitchner, CSC 471 course web page, Spring 1999, http://www.csc.calpoly.edu/~hitchner/CSC471

# Computer Graphics in the Scope of Informatics Engineering Education

**Manuel Próspero dos Santos**
Computer Science Department
Faculty of Science and Technology
New University of Lisbon
P-2825-144 Caparica
ps@di.fct.unl.pt

**Abstract**

*At the Faculty of Science and Technology of the New University of Lisbon, the education in Informatics Engineering has been supported by a B.Sc. course and a M.Sc. course. Both of them include Computer Graphics lectures, being the corresponding curricula adapted as much as possible to the other basic scientific matters. On the other hand, the tools and teaching methods have also been accommodated according to the global resources available at the Faculty for those courses, namely the computational ones. The paper presents the methodologies used today for teaching Computer Graphics at the New University of Lisbon, gives the results of some succeeded experiments, refers the most relevant relationships with other subjects and put important questions to the future of Computer Graphics education in the scope of Informatics Engineering.*
**Keywords:** *Computer Graphics, Teaching, Education, Informatics Engineering.*

## 1. Introduction

This position paper is about computer graphics education at the Faculty of Science and Technology of the New University of Lisbon (UNL), which was the first University conferring a B.Sc. degree in Informatics Engineering in Portugal. The way teaching has been done and the corresponding curricula have been chosen can be the object of a generalized discussion that will eventually lead to some improvements not only in these courses but also in similar ones. As it is well known, the computer graphics field has a multidisciplinary character and is still evolving nowadays. The impact of new products and different techniques that can be used today, such as the Internet, clearly shows another orientation a teacher has to give in order to solve some arising problems.

Although the main focus of the present paper is related with the teaching of computer graphics in B.Sc. and M.Sc. courses, the impact of this subject of study in other areas of Informatics Engineering at the UNL is also emphasized. Therefore, computer graphics education has been oriented, not only in the perspective of the end-user of a graphical application, but specially from the point of view of the underlying design and implementation. Thus, both the API and the basic programming level of a graphics system are to be covered in the courses.

## 2. Curricula of Computer Graphics

The next subsections of the paper describe the main contents of the different courses in computer graphics at the UNL. In addition to the indispensable textbooks, teacher's own notes are also available for all these courses. On the other hand, since the impact of the Internet has been enormous, not only this tool is filling the common lack of books and journals in our academic libraries but it is also a way to easily find very useful software products.

### 2.1. The Graduation Course Level

The existence of distinct modules at different levels of interest has shown to be very convenient for teaching computer graphics in the graduation course for Informatics Engineering, which is five years long. In the subsections below one can see the choices made in the real situation.

#### *2.1.1. Fundamentals*

Obviously, the first of several modules has always to have a more introductory programme. In the case of the B.Sc. course at the UNL, this module is called Computer Graphics I. It is one semester long and a mandatory subject. The topics corresponding to the covered matters are classified into some relatively diversified and well-known main areas:

- Computer Graphics definition, history and applications
- Main algorithms for output primitive raster conversion and corresponding attributes
- Clipping of lines and polygons
- Geometric primitives in graphical systems
- 2D viewing pipeline: Coordinate Systems and Window-Viewport transformations
- 2D and 3D Geometric Transformations

- Utility functions for geometric transformation composition
- Parallel and Perspective Projections
- Interactive methods for the design of 2D and 3D curves, including evaluation and representation techniques
- Interactive methods for the design of free-form surfaces, including evaluation and representation techniques
- Input logical devices and interaction techniques
- Implementation techniques for the Locator e Pick logical devices
- Main control functions and paradigms in both PHIGS and OpenGL™
- Concepts of hierarchical modeling and edition using PHIGS, OpenGL™ and VRML

#### *2.1.2. An Advanced Course*

Based on the previous knowledge of the students about graphics programming and concepts, a second module – called Computer Graphics II – covers a more restricted domain, although the main textbook is the same in both modules [1]. Due to the recent importance of 3D graphics, the chosen topics are mainly associated with photorealism and can be classified into the following groups:

- Introduction to the problem of photorealistic images
- Integration of geometric projections
- View volumes for arbitrary parallel and perspective projections
- The synthetic camera paradigm
- Normalizing Transformation and Perspective Transformation
- 3D clipping in homogeneous coordinates
- Algorithms for Hidden Line and Hidden Surface Removal
- Achromatic color and halftoning techniques
- Fundamentals of Colorimetry
- Color models
- Rules for the use and harmony of colors
- Basic notions of Fotometry
- Illumination models
- Special effects
- Shading methods for polygon meshes
- Texture mapping
- Ray-Tracing
- Radiosity
- Anti-aliasing techniques

The comparison of the curricula of these undergraduation courses, as they were stated above, may lead to some erroneous conclusions. For example, 3D graphics are seemingly postponed to the advanced course, causing an important lack in the basic one. However, this is not entirely true. As a matter of fact, what are really the significant characteristics of 3D graphics that are not covered by the topics of the first course? Note that geometric projections, 3D curves and the design of free-form surfaces, among others, were already included there. For this reason, we think the missing topics are the following ones: HLHSR algorithms, solid modeling, shading and illumination models. But these issues are presented to the students during the semester from an end-user point of view, sometimes with the help of pictures produced by well-known commercial products or extracted from recent movies. This kind of information is easily found in the Internet nowadays. If this is so, why don't we dispense with the second level course? Just because, in that way, the consequent level of students' education would not at all be adequate to an Informatics Engineering graduation. Thus, the specific topics about 3D appearing explicitly in the advanced course are treated in a programmer's point of view and not only by informing the students about the right application product to quickly produce the final result.

### 2.2. A Post-Graduation Course

One important goal of the M.Sc. course in Informatics Engineering is the specialization in computer science of graduated students. Due to the importance of computer graphics nowadays, this field of knowledge could not stay away.

The actual curriculum is based on the following items, whose main emphasis will be explained in section 4.:

- The role of Computer Graphics
- Main graphical representation methods
- Elements of Computational Geometry
- Geometric Modeling supporting the construction of synthetic images
- Solid modeling and representation schemes, namely B-rep, BSP, Octree and CSG
- Reference Models and API standards
- Concepts, paradigms and programming methods using PHIGS/PHIGS+, OpenGL™, Java and VRML

· Data Visualization, including isosurface generation, particle systems and volume rendering techniques

## 3. The Training at the Laboratory

An education in computer graphics is far from being complete without any laboratory component. Students are asked to develop some tasks whose level of difficulty depends on the kind of course they are attending: basic or advanced. The pictures included in this section are intended to provide the reader with a pictorial overview of some significant tasks.

Currently, the problems proposed to the students are a mixture between algorithmic and project oriented tasks. An effort is made in order to have all main chapters of the courses present in the final programs developed by the students. Additionally, the theoretical classes have to be synchronized with the explanation of the matters that are needed to find the most convenient solution for each problem.

All the pictures shown in the included Figures are screenshots taken from some programs developed by students at the UNL very recently. The programming activity in computer graphics has clearly benefited from the general purpose platform given by the Intel-based PC nowadays. For this reason, showing already a great evolution from the early recommendations stated in [2], at the UNL there is no longer any visible difference between a computer graphics laboratory and another one equipped for unspecified programming training.

The Microsoft's Visual C++ has been used in a Windows 95 environment [3]. Also the OpenGL™ programs were entirely created having the support of this platform [4], most of them interfacing the window system through the GLUT library [5].

Figure 1 is just one example of window-viewport transformations and planar geometric projections, with the possibility of an interactive control of the parameters.

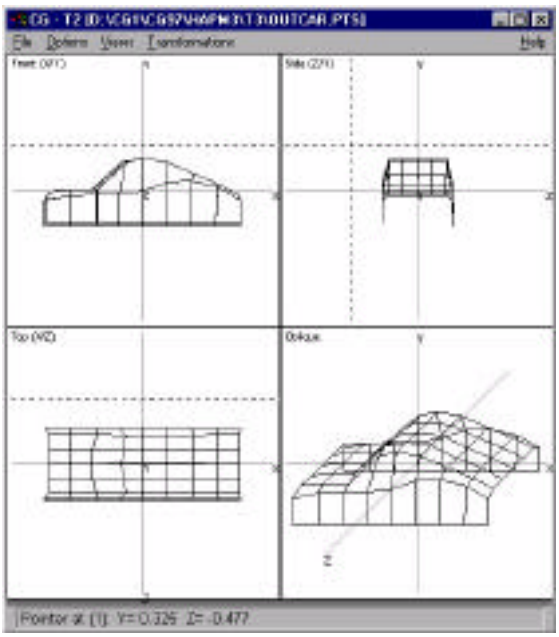


Figure 1. Window-Viewport Transforma tions and Planar Geometric Projections.

Different methods for designing curves are integrated in a simple editor to be implemented by the students and illustrated in Figure 2.

The fire-fighting truck depicted in Figure 3 can be moved forward and backwards under the user control. Additionally, the stairs have two kinds of independent movements: an elevation produced by a rotation transformation and a variable length achieved by shifting the three components accordingly.

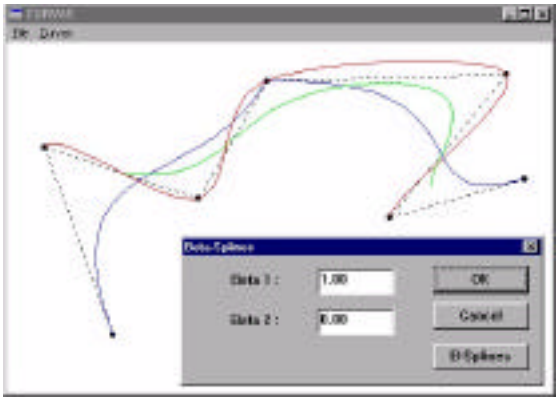


Figure 2. Interactive free-form curve editor.

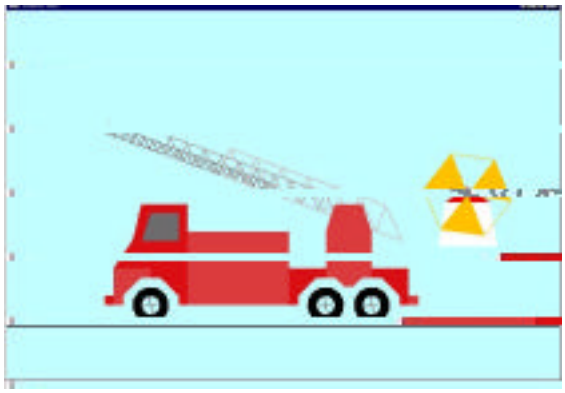


Figure 3. 2D hierarchical modeling with interactivity.

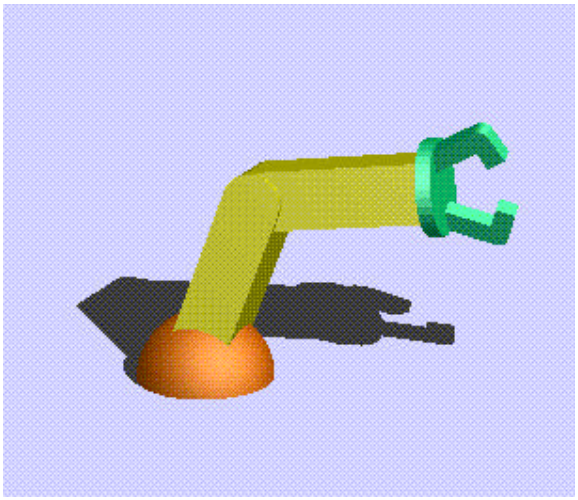


Figure 4. 3D hierarchical modeling with interactivity.

The ability of the students to create the underlying hierarchical model has shown us that not only all important concepts can be learned and practised in a 2D environment but also with additional advantages: normally the results in a 2D space have an easier interpretation and are less time consuming for debugging. The corresponding 3D situation is merely a generalization. This is illustrated in Figure 4. Therefore, the main difficult here is related to the use of illumination and projected shadows. Both Figures 3 and 4 resulted from OpenGL™ programs.

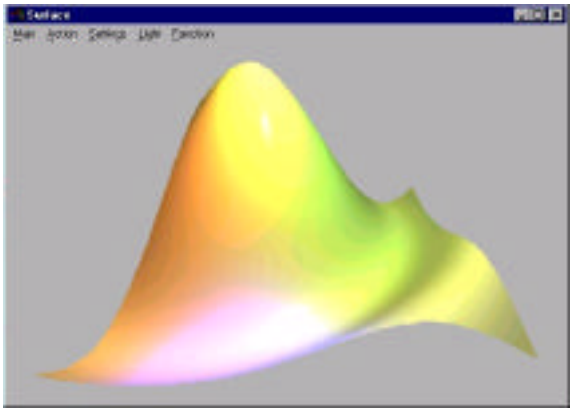


Figure 5. Surface rendering.

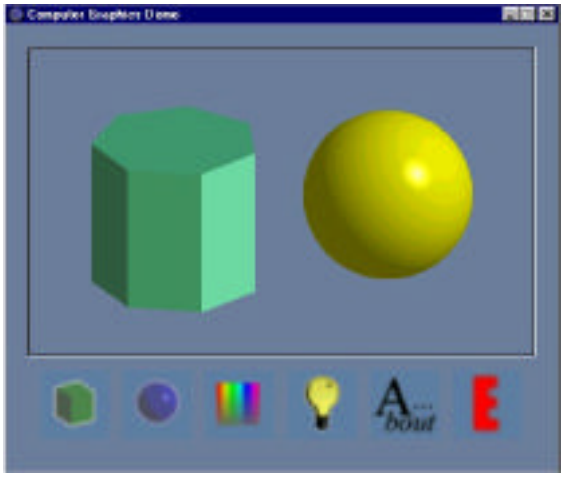


Figure 6. Illumination models and Shading.

The generation of the pictures shown in the Figures 5 and 6 was achieved by writing directly in the frame buffer, thus dispensing with the high-level functions usually found at the API level of a graphical system. This kind of exception to the use of OpenGL™ (see [6]) gives the students a good understanding of the rendering issues and allows them to program any special technique, such as the Phong shading. However, to be a feasible task in a short period of time, some restrictions had to be made: for instance, the output pipeline was simplified by fixing the viewing parameters.

Figure 7 shows us one single frame from an OpenGL™ animation program using B-splines.


Figure 7. 3D Modeling and Animation.

Finally, it is important to refer that an event-driven programming style is present in all applications written by the students.

## 4. Problems and Suggested Solutions

In what concerns Computer Graphics I, the topics dealing exclusively with 2D issues are taking nearly 25% of the total time of the actual theoretical classes, being very difficult to make this value lower. One cannot forget that even what we call 3D graphics become 2D at some stage of the complete output pipeline. Nevertheless, it is obvious that the referred percentage would be much more reduced at the cost of sacrificing the explanation of several algorithms and techniques if there were no further courses available, such as Computer Graphics II.

But, in our opinion, that reduction can be achieved by other means, which are compatible with a graduation course having a strong basic kernel of subjects. Let us put the following question: should we teach a programming language (e.g. C or C++) within a computer graphics course? The answer is clearly negative, because the knowledge about programming languages is an attendance pre-requisite for any computer science student. So, why don't we try to take a step forward by moving part of the traditional computer graphics topics into other basic subjects? The main problem here is that a serious compromise must be agreed with the instructors responsible for those subjects. But the foreseen benefits are great for all parts. Consider, for instance, the case where the student is learning the concepts of recursion. What would be the difference if those studies were complemented with a comparison between recursive and iterative algorithms for filling regions of pixels? Computationally speaking, this would be a much more valuable approach than the traditional numerical examples such as the factorial function or the Fibonacci numbers. And there is a lot of additional examples from the area of computer graphics that could be useful to other fields of computer science and don't need long explanations to the new generation of students.

This solution can lead to a natural integration of different subjects of the graduation in Informatics Engineering, but requires the staff responsible for the corresponding lectures to have also a very good background knowledge about graphics. To achieve this goal, the necessary teacher's education is being started at the UNL in what concerns the preliminary programming language courses. We hope that, with the functionality present in Delphi, some basic computer graphics concepts and techniques will be easily introduced at the same time students start to learn Pascal and basic data structures.

There are also several general problems that became visible during the past experience in computer graphics education at the post-graduation level in Portugal. The most important is related with the dissimilarity of the background knowledge of the students, since they usually come from very different Universities in the country. This fact would not be serious if a suitable set of different curricula could be implemented in order to configure the course to the profile of each student. However, this methodology is only feasible when the expected number of students is considerably high. This is not the present case at the UNL and no change is foreseen at short or long term. As a matter of fact, we are talking about a global number between 15 and 20 attendants per year at this M.Sc. course.

As a consequence, the suggested strategy is to elaborate a curriculum where the following compromise is assumed. On the one hand, some basic and general issues in computer graphics have to be quickly addressed in order to uniformize the background knowledge of the students. On the other hand, the choice of the most convenient topics must include the state of the art techniques and the methodologies that can meet the wishes of a student having already a good understanding of computer graphics fundamentals.

However, no intensive programming is normally needed at the curricular part of the M.Sc. courses, since they consist of theoretical lectures and the laboratory is free to be used by the students whenever they want to see the practical aspects of some particular matters. Computational geometry topics and geometric modeling issues are considered to be a good choice from this point of view, because they can be treated at a level as high as we want. Nevertheless, a programming activity is always welcome in Informatics Engineering. Due to its declarative character, VRML was elected as the most

privileged candidate to the main tool for developing skill in graphics programming either at a basic or advanced level [7, 8].

## 5. Conclusions

In general, the evolution of educational methods greatly results from the analysis of past experiences. But these systems are very complex and the staff responsible for teaching is not always having the knowledge about all the relevant parameters. Although many teachers conscientiously assume this fact, that is not the general rule. It thus becomes clear that the lack of helping mechanisms to solve the problem also affects those teachers who want to improve their methods based on the experience from another one.

When trying to find an adequate answer to a specific problem one must obviously consider each methodology that proved to succeed in similar situations before. However, the existence of a small difference between them, even not very significant apparently, may lead sometimes to a completely different and unexpected result. It also happens that the education and learning processes can be highly improved if some details are taken into account. For instance, following an observation by Jansen and Nieuwenhuizen in [9], the first work given to the students attending the laboratory classes in Computer Graphics I set to be a pre-programmed task instead of a simple statement about the problem to be solved. And the gratifying result was an extraordinary grow in the motivation of the students.

Computer graphics is far from being restricted to 3D graphics, no matter their increasing importance today. The way the basic undergraduate course is implemented at the UNL, as described in the paper, does not punish the student who has more difficulties to picture 3D entities and to realize, for instance, their shape and behavior related with movements, intersections, derivatives, volumes and so on.

The paper addresses other important issues. One of them is related with the direct impact the computer graphics education has had in other subjects of Informatics Engineering courses, specially in those matters connected to programming languages, algorithms and data structures. But that is not all, as we can see next. At the UNL there is no computational geometry course yet. The corresponding most significant issues are only referred in the computer graphics courses. However, we are trying to move part of them into a recently created course on complementary mathematics, named Discrete Mathematics, in the second year of the graduation studies. We hope it will be possible to extend this kind of collaboration and integration to other subjects in the near future.

Due to the rapid progress on the computer graphics field, the methods and contents of each course change every edition at the UNL. This is true even for the most basic one. The actual domain of computer graphics is so wide that it is impossible to completely embrace it in any course, especially when a small set of classes is provided. Comparing the UNL courses with the HyperGraph project [10], for example, we can observe that the major set of uncovered items concerns the hardware descriptions and not the software issues. In summary, each educational institution must choose the most relevant topics and the way they can be taught in order to contribute to the specific objectives of the graduation and post-graduation courses to be offered. Lacking for adequate evaluation methods, this is usually a very subjective and difficult task for the teachers.

In 1999, at the UNL, the introductory course for undergraduate students moved from the fourth to the third year of the graduation in Informatics Engineering, reflecting the fact that computer graphics is no longer a complementary subject. As a matter of fact, that course gives such a complete preparation to the students that it can be very useful to several subjects taught in the last two years of the graduation time period. One must note that this kind of change is only possible when the required extensive amount of background in mathematics and programming is guaranteed at the end of the second year of the computer science studies.

## Acknowledgements

The following students gave their contribution to the Figures of section 3: Hugo Abreu, Paulo Matos, Telmo Cruz, Paulo Santos, Sérgio Estêvão, João Feliciano, João Canhita, Alfredo Pereira, Cristina Garcia, Pedro Valente, Pedro Lopes and José Domingos.

## References

[1] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes. *Computer Graphics — Principles and Practice*. Second Edition in C. Addison-Wesley (1996).

[2] ACM/IEEE-CS Joint Curriculum Task Force, *Computing Curricula 1991*. (February 1991).

[3] C. Petzold. *Programming Windows*. 5th edition, Microsoft Press (1998).

[4] R. Fosner, *OpenGL Programming for Windows 95 and Windows NT*. Addison-Wesley (1996).

[5] J. Neider, T. Davis, M. Woo (OpenGL ARB). *OpenGL Programming Guide*. Second Edition, Version 1.1. Addison-

Wesley (1997).

[6] R. Wolfe. OpenGL: Agent of Change or Sign of the Times?. *Computer Graphics*. 29-31 (November 1998).

[7] A.L. Ames, D.R. Nadeau, J.L. Moreland, *The VRML 2.0 Sourcebook*. 2nd edition. John Willey & Sons Inc (1997).

[8] B. Roehl, J. Couch, C. Reed-Ballreich, T. Rohaly, G. Brown. *Late Night VRML 2.0 with Java*. Ziff-Davis Press (1997).

[9] F.W. Jansen, P.R. Nieuwenhuizen. Computer Graphics Education at Delft University of Technology. *Proceedings of the Eurographics Workshop on Graphics and Visualization Education*, Oslo (1994).

[10] G. Scott Owen (project director). *HyperGraph — Teaching Computer Graphics*: A project of the ACM SIGGRAPH Education Committee, the Hypermedia and Visualization Laboratory, Georgia State University, and the National Science Foundation. http://www.education.siggraph.org

# A Course in Topology-based Geometric Modeling

**Pascal Lienhardt**, **Laurent Fuchs** and **Yves Bertrand**
IRCOM-SIC,
Poitiers, France
{lienhardt, fuchs, bertrand}@sic.sp2mi.univ-poitiers.fr

**Abstract**

*The content of a course in Topology-based Geometric Modeling is presented. The key ideas are based upon simplicial notions issued from combinatorial topology. The well adequacy of these notions to geometric patches (such as Bézier patches) is pointed out. Topology and geometric notions are simultaneously presented in order to show their complementarity.*
***Keywords:*** *Geometric modeling, combinatorial topology, subdivisions, geometric patches.*

## 1. Introduction

The goals and the content of a course in Topology-based Geometric Modeling are described in this paper. Since several years, it is addressed to Computer Science students (Master Degree, ninth and tenth semester) who have acquired basic knowledge in computer graphics, but who did not thoroughly approach Geometric Modeling. Other courses specialized in other fields of Geometric Modeling can supplement this course, e.g. C.S.G. and implicit surfaces.

The purpose is the acquisition of solid basic concepts: so, we insist here about the fundamental concepts in mathematics and computer science, and mainly about :

- (combinatorial) topological notions related to subdivisions of geometric spaces,
- parametric curves and surfaces, and more generally curved shapes,
- the links that can be made between these two notions, especially combinatorial ones.

The design of data structures is considered as well as that of basic construction operations and algorithms for computing topological properties.

The course contains three main parts :

- simplicial notions, mainly simplicial sets and triangular patches ;
- simploidal notions, where cells are cartesian products of simplices ;
- cellular notions, where cellular structures are deduced from “numbered” simplicial sets and embedding is based upon parametric trimmed patches.

We insist here about the complementarity between the topological aspects (subdivision, etc) and the embedding ones (curves and surfaces). Highlighting the common combinatorial concepts facilitates this. Hence a global and homogeneous outline of the field is given.

## 2. Course content

### 2.1. Introduction (2h)

Basic concepts are introduced: subdivision of geometric space (i.e. partition into cells of various dimensions on which boundary relations are defined), various "classes" of subdivisions (simplicial or cellular ones, complexes, manifolds, ...), homeomorphism, etc.

This initiation is carried out by the means of a constructive approach of the classification of surface subdivisions into topological surfaces, as it can be found in [8]: the basic objects are polygons (topological discs) and surface subdivisions are built by identification of edges. The classification of surfaces (and the related characteristics: number of boundaries, orientability factor, genus) is approached by distinguishing the various cases of identification (the proof that any subdivision belongs to a unique topological surface is not tackled).

This part can by supplemented by a practical session during which the students, provided with paper, scissors and tape, build and classify paper surfaces.

### 2.2. Simplicial : Fundamentals (6h).

A first part shows how to deduce combinatorial notions from geometrical ones. We define successively:

- simplicial complexes, defined as convex hulls of linearly independent points,

- abstract simplicial complexes, where a simplex is a set of abstract vertices ; the *geometric realization* of an abstract simplicial complex is a simplicial complex (vertices are associated with points, ...),
- semi-simplicial sets [12], where a simplex is an abstract object and boundary operators associate its boundary simplices with the simplex (cf. the following definition and Figure 1). The geometric realization of a semi-simplicial set is a CW-complex.

**Definition** [12] **:** A $d$-dimensional *semi-simplicial set* $K$, is a union $K = \bigcup_{l=0}^{d} K_l$ of sets $K_l$ of abstract simplices of dimension $l \le d$, together with boundary operators $\partial_i$ such that:

$$\partial_i : K_l \to K_{l-1} \text{ with } l \ge 1 \text{ and } 0 \le i \le l,$$
$$\sigma\partial_i\partial_k = \sigma\partial_k\partial_{i-1} \text{ for } i > k \text{ and } \sigma \in K.$$

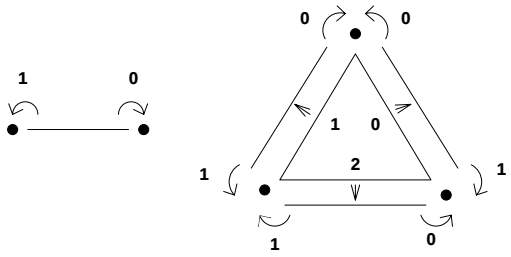



Figure 1. 1- and 2-dimensional simplices, and their boundaries.

Basic additional notions and construction operations are defined (e.g. geometric realization, isomorphism, simplex creation, identification) and traversal algorithms are described in order to compute topological properties (e.g. connectivity, boundary, star).

In a second part, the embedding notion is introduced in order to define the object shape. First, we show that a subclass of semi-simplicial sets is equivalent to abstract simplicial complexes ; so, they can be embedded as simplicial complexes, by associating points with $0$-simplices.

Embedding is then generalized by taking curved shapes into account (i.e. curves, triangular surfaces, tetrahedral volumes, etc). These shapes are introduced by the "Bézier simplex" notion and described using the de Casteljau algorithm in any dimension. Bernstein polynomials are defined. Joining and continuity problems are taken into consideration. Figure 2 gives us an example of a semi-simplicial set.

Combinatorial links between an abstract simplex and a Bezier simplex are pointed out. Mainly it is possible to extract a semi-simplicial set structure from the control point structure (cf. Figure 3) : it is thus possible to associate subsets of control points with abstract simplices, and this can be used in order to design data structures.

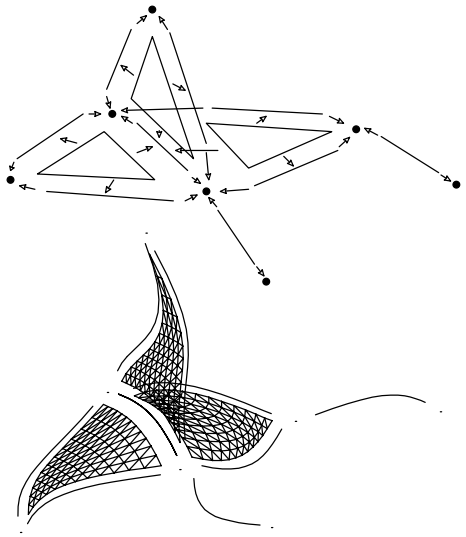


Figure 2. Example of semi-simplicial set, up the topological structure, down the embedding.

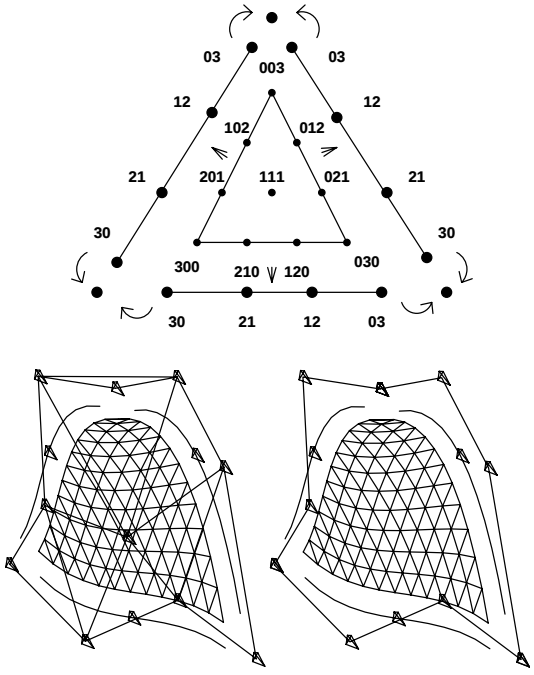



Figure 3. Semi-simplicial set structure and control point structure.

The last part is devoted to geometric operations (e.g. splitting, extrusion and cone). They are compositions of operations acting on semi-simplicial sets and on Bézier simplices.

Implementation is also addressed from abstract data types to concrete data structures using records and pointers. A basic concrete data structure can be directly deduced from the combinatorial definition of $n$-dimensional semi-simplicial sets.

```
struct cpl {
       (I+1)_SIMPLEX * s ;
       int b_op ;
       struct cpl * next, * prev ;
} * CPL ;

struct i_simplex {
       struct simplex * delta[i+1] ;
       CPL ante ;
} I_SIMPLEX ;
```

The record (written in pseudo-code) `I_IMPLEX` represents an $i$-simplex : the first field (`delta`) implements the boundary operators ; the second field implements the "inverse" operators : a doubly-linked list contains the simplices which have the considered simplex as image by a boundary operator.

From an abstract data type point of view, the simplicial sets are defined by the operations working on the data structure. Basic operations such as

```
set_border :
  nat simplex simplex simplexSet -> simplexSet

border : simplex nat -> simplex
```

which respectively define boundary operators and apply boundary operator to a simplex are defined by equations. Then fundamental properties are expressed, e.g.

```
border(k, border(i, s)) =
   border(i-1, border(k, s)) if i > k
```

Similar implementation mechanisms are used for the other simploidal and cellular notions presented in the next sections.

### 2.3. Simplicial (4h).

This part has three complementary goals:

- Extend semi-simplicial sets by defining simplicial sets [12] ; thus cartesian product can be defined. This operation is a generalization of the extrusion one; it is the combinatorial basis for the computation of Minkowski sums. Moreover, notions presented in section 2.4 are based upon cartesian product.
- Define basic mechanisms which make it possible to deduce optimized structures for some classes of simplicial sets (e.g. homogeneous, regular, quasi-manifolds), and related operations (i.e. constrained operations that are internal to sub-classes). Using these mechanisms, relations can be established with other structures defined in literature, and with cellular structures (see section 2.5) [4, 9, 11].
- Notions related to curved shapes are abstracted and generalized by introducing the general simplicial algorithms [6, 7]. This provides a uniform approach for the parametric curves, surfaces, volumes, etc.

### 2.4. Simploidal (4h).

Previous simplicial notions are generalized by defining structures that can represent subdivisions in which cells are cartesian products of simplices.

First, cubic sets are defined [13] : a cube is the cartesian product of 1-dimensional simplices.

**Definition:** A $d$-dimensional *cubic set* $K$ is the union $K = \bigcup_{l=0}^{d} K_l$ of sets $K_l$, which elements are abstract cubes of dimension $l \le d$, together with boundary operators $\partial_j^i$ such that:

$$\partial_j^i : K_l \to K_{l-1} \text{ with } l \ge 1,\ j = 0,1 \text{ and } i \le l,$$
$$\sigma\partial_j^i\partial_l^k = \sigma\partial_l^k\partial_j^{i-1} \text{ for } i > k \text{ and } \sigma \in K.$$

Combinatorial links between Bézier patches (tensor product of Bézier curves) and abstract cubes are pointed out, leading to the embedding of cubic sets using cubic shapes.

Abstract simplices and abstract cubes are generalized by abstract simploids (the corresponding geometric notion is the simploid notion [3]). They are cartesian products of abstract simplices. An abstract simploid is fully defined by the $k$-tuple $(d_1, \ldots, d_k)$ of the dimensions of the $k$ simplices involved in the product (cf. Figure 4).

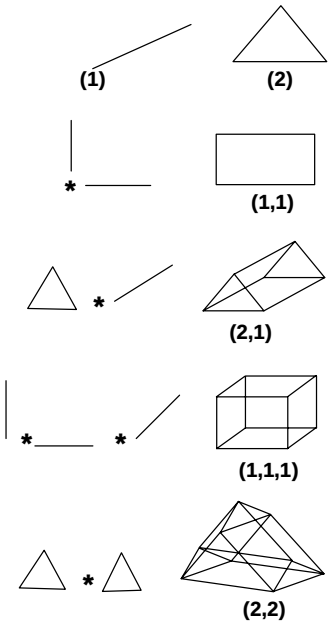



Figure 4. Example of simploids. The cartesian product of at least two simplices is graphically represented.

Then, simploidal sets are defined by (cf. Figure 5) :

**Definition** [7] **:** A $d$ -dimensional simploidal set $K$ is the union $K = \bigcup_{l=0}^{d} K_l$ of sets $K_l$, of abstract simploids of dimension $l \le d$, together with boundary operators $\partial_j^i$ such that[1]:

$$(\ldots,d_i,\ldots)\partial_j^i = \begin{cases} (\ldots,d_i-1,\ldots) \text{ if } d_i > 1 \\ (\ldots,\hat{d}_i,\ldots) \quad \text{else} \end{cases}$$

$$(\ldots,d_i,\ldots)\partial_k^i\partial_l^i = (\ldots,d_i,\ldots)\partial_l^i\partial_{k-1}^i \text{ with } k > l \text{ and } d_i > 1$$

$$(\ldots,d_i,\ldots,d_j,\ldots)\partial_k^j\partial_l^i = \begin{cases} (\ldots,d_i,\ldots,d_j,\ldots)\partial_l^i\partial_k^j & \text{if } d_i > 1 \\ (\ldots,d_i,\ldots,d_j,\ldots)\partial_l^i\partial_k^{j-1} & \text{else} \end{cases}$$

Products of general simplicial algorithms [6], generalizing cubic shapes, are also defined, and used for embedding simploidal sets. Once again, the complementarity of the embedding and the topological aspects are pointed out and used. For instance for the design of data structures, we also explain how common combinatorial notions simplify the implementation. Simplicial operations (and implementation aspects) are extended to the simploidal framework. Lastly, open research problems are highlighted.

[1] The hat notation means that the element is removed from the tuple.

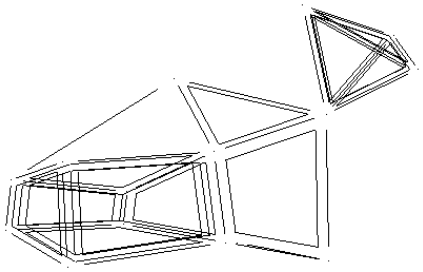


Figure 5. Example of simploidal set.

**2.5. Cellular Structures (8h)**

A first part shows how to deduce a first cellular structure from a subclass of simplicial sets [11]. Simplicial sets are structured into cells by associating numbers with $0$-simplices (this is quite close to the classical notion of barycentric triangulation : numbers associated with the vertices of an *i*-simplex are {0, 1, ..., i}). Then, we define cellular quasi-manifolds as a sub-class of such "numbered" simplicial sets, and the equivalent notion of combinatorial generalized map is deduced by applying conversion mechanisms presented in section 2.2 (cf. Figure 6).

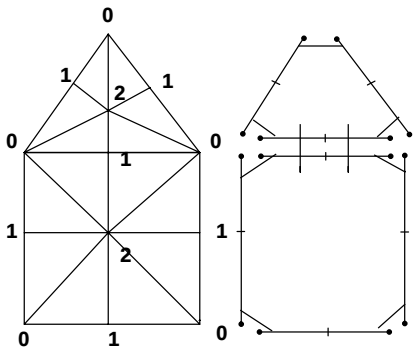



Figure 6. Numbered simplicial set and generalized map. A dart corresponds to a 2-dimensional simplex {0,1,2}. Application $\alpha_i$ joins simplices which share a 1-dimensional simplex {0,1,2}-{i}.

**Definition:** A $d$ -dimensional *generalized map* (or G-map) is a set $D$ of *darts* together with applications (cf. Figures 7,8,9) :

$$\alpha_k : D \to D \text{ with } 0 \le k \le d,$$
$$b\alpha_k^2 = b \quad \text{with } b \in D,$$
$$b\alpha_k\alpha_l = b\alpha_l\alpha_k \text{ with } k+1 < l \le d.$$

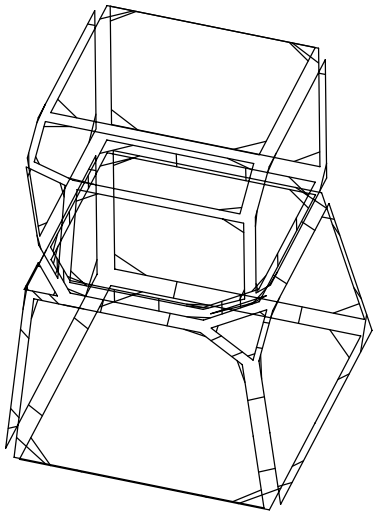


Figure 7. Example of a generalized map.

Construction operations are defined for handling generalized maps, and also algorithms for computing topological properties [1, 11] (e.g. connectivity, orientability; characteristics can be computed on 2-dimensional generalized maps providing their classification in the meaning of section 2.1).

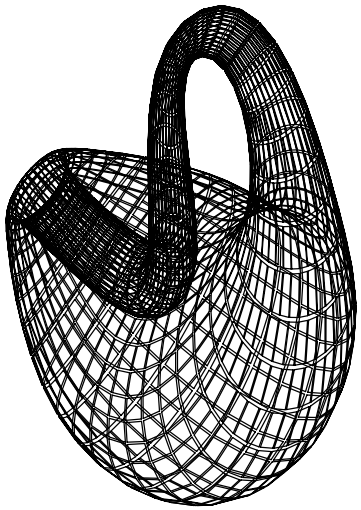


Figure 8. A subdivision of a Klein bottle whose topology is represented by a G-map ; its topological characteristics can be computed by simple traversal operations (i.e. number of boundaries = 0, orientability factor = 2, genus = 0).

A second part shows how to deduce cellular structures for other classes of "cellular objects" [4] (from combinatorial map chains for representing cellular complexes, to oriented combinatorial maps for representing oriented quasi-manifolds without boundaries). These conversions are obtained mainly by applying mechanisms presented in section 2.2. Relations with equivalent structures are also highlighted (e.g. winged-edge, quad-edge, radial-edge, facet-edge, cell-tuple structures, incidence graphs, etc. See [10] for instance).

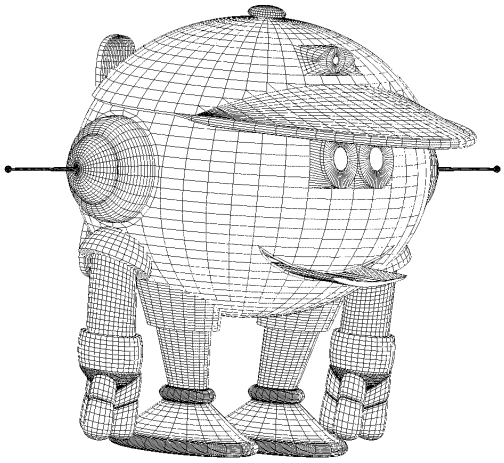


Figure 9. Complex objects can be modeled with G-maps

The last part is devoted to embedding. First linear embedding is described. After, parametric trimmed shapes are introduced. They generalize the notion of trimmed patches [5]. Hence, the link between the topological structure and embedding becomes more sophisticated: the space of parameters must be topologically structured.

Again, several open research problems are pointed out.

## 3. Remarks and Conclusion

Our experience shows that a practical work is useful : all notions can be perceived as theoretical ones and intrinsically interesting ; a practical work, for instance the conception of the kernel of a geometric modeler, shows the usefulness of these notions for the design of data structures and algorithms.

Note that sections 2.1, 2.2 and/or 2.5 can be extracted and taught in other curricula. Parts 2.1 and 2.2 have been taught to undergraduate Computer Science students : so, they can experiment simplicial structures which are graph generalizations. It is also possible to study only cellular structures, by removing sections 2.2, 2.3, 2.4. It is thus necessary to give an intuitive introduction, e.g. "how to deduce a generalized map from a cellular manifold?" (for instance using barycentric triangulation, splitting cell boundaries and/or cell-tuples [2]). But some notions can not be completely explained (e.g. definition of cartesian

product of cellular structures). At last, the course can be improved by studying other important notions, e.g. algorithms for computing homology groups, according to the background of students and the pedagogical context.

To sum up, this course is based upon simplicial notions (topology: simplicial sets; embedding: curved shapes) and two mechanisms:

- cartesian product providing spaces with simploidal cells,
- numbering 0-simplices, providing cellular structures embedded with parametric trimmed shapes.

This approach presents mathematical and computational aspects. Fundamentals are explained. Indications are given about higher-level aspects, some of them still being subjects of active research.

Topology and embedding are simultaneously presented in order to show their complementarity and to highlight their common combinatorial notions, which are important for the conception of data structures and algorithms.

All topological notions are based upon simplicial ones issued from combinatorial topology. This discussion thread is useful for the study of deduced structures, and simplifies the explanations about topological and embedding concepts, operations and algorithms. Moreover, this approach leads coherently to many other works in geometric modeling, and mainly in Boundary Representation.

## References


[1] Y. Bertrand, J.-F. Dufourd, J. Françon and P. Lienhardt. Algebraic Specification and Development In Geometric Modeling. *Conf. Tapsoft'93*, LNCS n°668, pages 75-89, Springer-Verlag, April 1993.

[2] E. Brisson. Representing Geometric Structures in d-dimensions : Topology and Order. *Discrete and Computational Geometry*, 9, pages 387-426, 1993.

[3] W. Dahmen and C. A. Micchelli. On the Linear independence of Multi-Variate B-splines I. Triangulation of Simploids. *SIAM J. Numer. Anal.* 19, pages 993-1012, 1982.

[4] H. Elter and P. Lienhardt. Cellular Complexes as Structured Semi-simplicial sets. *International Journal of Shape Modeling*, 1(2) pages 191-217, 1994.

[5] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide.* Academic Press, San Diego, 1988.

[6] L. Fuchs. Spécification formelle des modèles de courbes et de surfaces de formes libres. PhD Thesis, Université Louis Pasteur, Strasbourg, France, 1997.

[7] L. Fuchs and P. Lienhardt, Topological Structures and Free-Form Spaces. *Journées de Géométrie Algorithmique*, Barcelone, Spain, September 1997.

[8] H.-B. Griffiths. *Surfaces.* Cambridge University Press, 2nd edition, 1981.

[9] V. Lang and P. Lienhardt. Geometric Modeling with Simplicial Sets. *Proc. of C.G.I.'96*, Pohang, Korea, 1996.

[10] P. Lienhardt. Topological Models for Boundary Representation : a Comparison with N-Dimensional Generalized Maps. *Computer-Aided Design*, 23(1), pages 59-82, 1991.

[11] P. Lienhardt, N-Dimensional Generalized Combinatorial Maps and Cellular Quasi-Manifolds. *International Journal of Computational Geometry and Applications*, 4(3), pages 275-324, 1994.

[12] J. P. May. *Simplicial Objects in Algebraic Topology.* Van Nostrand, Princeton, 1967.

[13] J. P. Serre. Homologie singulière des espaces fibrés. *Ann. of Mathematics*, 54, pages 425-505, 1951.

# Re-Inventing the Introductory Computer Graphics Course: Providing Tools for a Wider Audience

**Steve Cunningham**
Computer Science
California State University Stanislaus
Turlock, CA 95382 USA
rsc@csustan.edu

**Abstract**

In ACM/IEEE Curriculum 91 [TUC 91] and in earlier curriculum statements on computer science, the computer graphics course in computer science focused on fundamental algorithms and techniques for creating images and animations. Forthcoming curriculum statements are going to find a need to expand the role of courses such as computer graphics in the undergraduate computer science curriculum [CUN 98]. In the last few years the approach to teaching this course has changed to take advantage of more powerful graphics tools. By using these tools, however, we are also able to make the graphics class accessible to students from other areas. This paper describes a set of presentations, examples, and projects that supports the introductory computer graphics course as a tool for the student in the sciences, mathematics, or engineering as well as for the computer science student.

## 1. Introduction

Computer graphics is an important tool for the working scientist, mathematician, and engineer, and has become a commodity item in the sciences just as it has in so many other areas of life. Its value is expanding as the scientific culture embraces scientific visualization and as powerful computers become more and more commonly available. However, the traditional computer graphics course focuses on fundamental graphical algorithms and processes, not high-level graphics programming. This is oriented to the computer science student who wants to pursue advanced work in the field, not the more general student in science or engineering.

This offers computer science an opportunity to serve this wider audience by creating an introductory computer graphics course that has dual goals. One goal is to introduce computer science students to basic geometric processes and to graphics programming, and the other is to give students in science, mathematics, and engineering a basic background in thinking and programming in 3D geometric terms.

## 2. The Course Philosophy and Outline

One fundamental goal of the revised introductory computer graphics course sets it apart from the traditional course. This goal is to create a sound course for science, mathematics, and engineering students (hereafter abbreviated as "science") that will give them the graphics and visualization tools that they need for their future [BRO 90]. In doing this, we must avoid creating a course that poses significant programming challenges for students, because many students from these fields will have less programming experience than a computer science student would. Thus the course must provide a sound graphics experience for a student who has had only a two-semester sequence in programming.

At the same time, another course goal is to provide computer science students with an introduction to the field of computer graphics. Thus the course must be useful to the general computer science student and provide him or her sound skills in graphics programming, because many applications will need to add graphical content in the future.

All the work in the course is done entirely in 3D, with no reference at all to 2D grapics except perhaps in a few specific examples. This supports both goals for the course and is consistent with current directions in API development and in graphics applications. The heterogeneous nature of the class does offer a challenge, but the expected benefits should outweigh this problem. Certainly computer science students should benefit from working with others outside our field, and science students should benefit from seeing a computer science approach that supports their fields.

The course outline is straightforward. It begins with an introduction to the fundamental notions of geometry as used by

OpenGL, and with simple examples of OpenGL programs to help students see how to build their first programming projects. It goes on to cover fundamentals of polygon-based modeling, including the use of instancing transformations; to discuss rendering including projections, viewing, lighting, and shading; and to work with event-handling to provide user controls and animation. It ends with some of the features that students rarely actually use in a traditional fundamentals course, such as alpha blending, texture maps, and fog.

This course is largely possible because of advances in low-cost and high-function graphics systems. Computers with these capabilities are now available for even budget-strapped universities. Graphics programming APIs such as OpenGL and Java3D are readily useable by students who have a programming background but are not computing specialists. Further tools such as the Fahrenheit API being developed by SGI and Microsoft [FAR 98] will also soon be widely available.

## 3. The Presentation Sequence and Examples

In order to introduce students to the subject, we begin with geometric and programming fundamentals and proceed to more sophisticated topics in modeling and rendering. The list below summarizes the contents, but it should not be taken as an indication that we proceed in a linear way through this material. Instead we do a little modeling, then a little rendering, then come back to more modeling, then more rendering, and so on until we have covered all this content in one semester.

In order to be valuable for the broad audience we noted above, the course material needs to be backed up with a collection of examples and projects that come from an equally broad set of areas. We need some examples and projects from mathematics, some from physics, some from chemistry, and some from various branches of engineering. Creating a set of materials with this broad coverage will take some time, and the author has begun to work on such a project. These materials will be developed and tested in collaboration with other faculty having many different kinds of experience and with students from other disciplines. The goal is to have a draft set of materials available by mid-2000 and a completed set a year from then. The materials will be distributed on the Web and will continue to evolve as our understanding of the needs of science and engineering students grows.

An outline of the overall set of topics covered is:

I. Introduction to geometry and to graphics programming
   a. Coordinate systems and geometry fundamentals in 3-space. Coordinates, normals, vectors, dot and cross products, polygons, etc.
   b. What is an OpenGL program? Example or two of some simple code in template programs to introduce the idea of using the OpenGL system.

II. Modeling
   a. Modeling primitives. Points, lines, triangles, quads, triangle strips, triangle fans, quad strips, etc. Using built-in GLUT primitives; modeling using them. Frames as transformed modeling coordinate systems.
   b. Modeling objects using transformations. Creating items for a scene by starting with primitive objects and using scaling, rotation, and translation to shape them and place them in a scene. Handling transformation stacks. How the stack of modeling transformations is used to handle several objects. LIFO behavior of modeling transformations and the distinction between function composition and stack behavior.
   c. Evaluators and Bézier splines. Defining control points; specifying evaluators; using evaluator to generate points in the line or surface.
   d. Scene graphs. Concept, examples, applications

III. Rendering
   a. Projections and viewing. Orthogonal and perspective projections, standard view volume. Setting the viewpoint; relation to both kinds of projections
   b. Hidden surfaces. Enabling and specifying; what effects do you get
   c. Lighting and materials. Lights in a scene; ambient, diffuse, and specular lighting. More than one light. Colored lights. RGB and other color models. How materials are specified; application to objects in a scene.

d. Shading. Specifying shading; use of normal vectors. The difference between flat and Gouraud shading.

IV. Interaction

a. Event and event handling concepts. Event queues, event types, concept of callbacks, registering callbacks. Keyboard, special keys, menu, mouse, idle

V. Additional features

a. Color blending (alpha channel). Specifying, effects of blends
b. Fog. Enabling; effect of fog color; effect of density; specifying front and back
c. Textures. Creating texture maps, specifying maps, 2D and 3D texture maps
d. Text as a graphic element. Labels, in-space menus, billboards.

As each part of this material is presented to the class, the presentation first focuses on what the feature means in creating or interacting with an image by presenting demonstrations of the feature's effect on an image and pointing out how the feature is invoked in code by showing code examples and experimenting with parameters. When the student has some idea of what the feature is, the presentation will go on to give some background on what the feature means in terms of the actual graphics system work, and on occasion to include a brief description of the algorithms involved. To put these in balance, if we consider the OpenGL Programmers Guide [WOO 99] on one hand and a standard textbook that uses OpenGL on the other [ANG 97], the presentation is in based 80% on the programmers guide and 20% on the textbook.

## 4. The Projects

One key to making this course effective for students in science, mathematics, and engineering is creating a good set of projects that are meaningful to the student. We cannot use a single set of projects or projects that ask a student to create good-looking but meaningless images. The author is working on a set of projects that will fit this orientation for the course, and hopes to have a set ready to distribute in mid-2000. Some initial projects that were used in the first instance of the course included:

a) 3D histograms that can be rotated around one axis; only ambient light
b) physical object with one light source, where both object and light source can be rotated
c) pseudo-fractal landscape with semi-transparent water and both snow and tree lines
d) N-body gravity problem with each body showing a short-term trace of its recent location, with time-step animation
e) a Bézier surface with selectable and moveable control points

All of these projects included user-controlled motions or some form of animation, and some examples of student work on these projects is shown below. Because we had not yet done any promotion of this course to students outside computer science, we did not have any student input into interesting project topics for non-computer science students. This will be one approach to developing such projects in future course offerings.

The language used for projects, either C or C++, is not an issue in the course or projects. It is possible to call any of the OpenGL functions through either language. Thus students can choose the language they are most familiar with and use whatever level of the language one can handle. In the future we expect to consider offering the course with Java3D or perhaps Fahrenheit, depending on the availability of the systems and the amount of programming background needed to use them. A relatively modest independence on the programming language is a very important part of making the course work for non-computer science students and cannot be overemphasized: students can create interesting and useful images without using sophisticated programming techniques.

## 5. The Sequel to This Course

At the end of the introductory computer graphics course, students have experience in thinking geometrically and in creating programs that express that geometric thinking and use some modest animation and interaction. They have seen a very modest description of how some graphics algorithms are used to create images, but have done no work directly with those algorithms. This gives my computer science students an excellent background for tackling a second graphics course containing the traditional fundamental algorithms content.

In the second course, then, we include familiar fundamental techniques. These include interpolation material such as scan-converting

polygons, shading models, Z-buffering, bump and environment mapping, as well as techniques such as antialiasing and ray tracing. We have found that straightforward modifications to the Bresenham algorithm [CUN 99] provide a general and straightforward approach to the whole range of interpolation algorithms on polygons, and between this approach and the experience students get from the first course, they are able to develop a good understanding of advanced graphics techniques.

## 6. Results so Far

The author has taught this course and its sequel only once, so we cannot give a good evaluation of this work. My computer science students are enthusiastic about this approach to the course and several took the fundamental graphics principles course that was taught as a second course in sequence. The results of the second course showed that students did learn many key concepts in the subject from the first course, and we were able to move quite quickly through material that had been fairly painful.

The value of the traditional computer graphics principles course does not seem to be reduced by beginning computer graphics studies with this new kind of course. Instead the new first course makes a very good introduction to the principles course by providing students with a background in realizing images from geometric concepts. Starting with the programming course allows a traditional principles course to move more quickly and to address more advanced concepts than a course that assumes no graphics background. The combination of a first course based on graphics API programming and a fundamental principles course seems to be an excellent way to introduce computer graphics content for computer science students. More advanced courses can be developed as needed when additional topics need to be covered.

In order to understand the level of this graphics programming approach, it is useful to see the kind of work done by students in the new first course. Below are some images from the last three of the course's five projects. Each project also involves animation and/or user interaction in addition to the image synthesis.

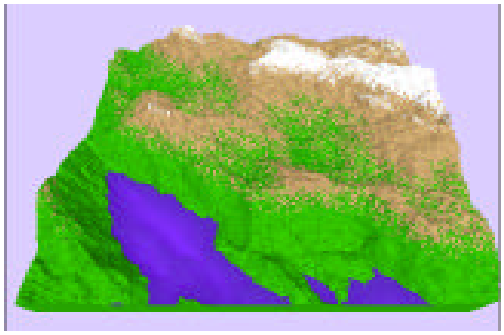


Figure 1. pseudo-fractal landscape with alpha blending in water

Figure 2. n-body problem with fade-out trails on bodies

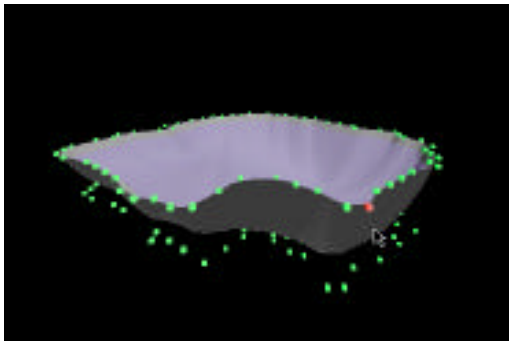


Figure 3. Bézier spline surface with control points shown; one is selected for movement

In the first example, students use keyboard callbacks to move around the landscape and see it from different viewpoints, and use mouse callbacks to work with a menu to change various features of the landscape. In the second example, the bodies move in real time driven by the idle callback, using simulated gravitational forces between the multiple bodies. In the third example, students move around the shape with keyboard control and can select and manipulate a

control point with the mouse, as shown; the shape of the surface changes as the point is moved with keyboard control.

## 7. Summary

One of the challenges of computer science is to find ways to serve our universities and still maintain the quality of our courses and programs. Because computer graphics is so useful to so many different fields, it is a natural course to take on a service role, but because the course has traditionally made significant mathematical and programming demands on its students, it has rarely been used in this way.

The approach outlined in this note offers us a way to provide a course that benefits both the computer science student and the student from mathematics, science, or engineering. It is accessible to faculty, is useful to computer science students, opens the door to studies of more advanced computer graphics topics, and is a valuable service to other students whose professional lives will require continual use of computer graphics.

And — let's say this only here at the end, and let's say it quietly, though it makes a real difference to the nature of the course — the course is a lot of fun to teach and students really enjoy the high-quality work they produce in the course.

## References


[ANG 97] Angel, Ed, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*, Addison-Wesley, 1997

[BRO 90] Brown, Judith R., Steve Cunningham, and Mike McGrath, "Visualization in Science and Engineering Education," in *IEEE Tutorial: Scientific Visualization*, Nielson, Gregory M. and Bruce Shriver, eds., IEEE Computer Society, 1990

[CUN 98] Cunningham, Steve, "Outside the Box — The Changing Shape of the Computing World," invited editorial, *SIGCSE Bulletin* 30(4), December 1998, 4a-7a

[CUN 99] Cunningham, Steve, "The Ubiquitous Bresenham Algorithm and Its Applications Across Many Graphics Processes," in preparation

[FAR 98] SGI Fahrenheit Web site, `http://www.sgi.com/fahrenheit/`

[NAD 98] Nadeau, David R. and Henry Sowritzal, "Introduction to Programming with Java3d," Eurographics 98 tutorial, also presented at SIGGRAPH 98

[TUC 91] Tucker, Allen B. and Bruce H. Barnes, eds., *Computing Curricula 1991: Report of the ACM/IEEE/CS Curriculum Task Force*, ACM Press/IEEE Computer Society Press, 1991

[WOO 98] Woo, Mason and Dave Schreiner, *A Visual Introduction to OpenGL Programming*, SIGGRAPH 98 course notes, 1998

[WOO 99] Woo, Neider, and Davis, *OpenGL Programming Guide*, third edition, Addison-Wesley, 1999

# A Mathematica Package for CAGD and Computer Graphics

**Andrés Iglesias**[1], **Flabio Gutiérrez**[1,2] and **Akemi Gálvez**[1]
[1]Departament of Applied Mathematics and Computational Sciences
University of Cantabria, Santander, Spain
[2]Departament of Mathematics, Sciences Faculty
National University of Piura, Peru
iglesias@ccaix3.unican.es, flabio@tallan.unp.edu.pe, uc8031@cclx1.unican.es

**Abstract**

*This paper introduces a Mathematica package for learning and teaching Computer-Aided Geometric Design (CAGD). The package incorporates commands for the treatment of the most usual curves and surfaces in CAGD (Bézier, B-spline, rationals, etc.). The powerful symbolic and graphical Mathematica capabilities, the functional and pattern recognition programming and the Mathematica visualization environment have been extensively applied to get a user-friendly, didactic and powerful tool for education, with applications in areas as CAGD and Computer Graphics. Some illustrative examples showing the performance of the package are also given.*
**Keywords:** *Symbolic programming, visualization, computer graphics, CAGD.*

## 1. Introduction

Computer graphics and geometric modeling constitute fundamental parts of the engineering design process. Geometric modeling methods are applied to construct a precise mathematical description of the shape of a real object and its efficient computer representation. On the other hand, computer graphics capture the visual and quantitative aspects of this object. They are also essential topics in the engineering education process (see [1]).

1. In the last few years, engineering teachers have witnessed an extraordinary change in the way of teaching and developing students' skills for their professional future. The extraordinary advances in hardware and software have made possible the appearance of a new generation of scientific Symbolic Computation Programs (SCPs), as Mathematica or Maple. In comparison with the traditional program-ming languages, (Fortran, Pascal, C, etc.) the SCPs incorporate a great number of symbolic, graphical and programming facilities. Today, teachers can take advantage of these graphical capabilities when introducing students to computer graphics and/or visualizing the geometry of the objects under analysis. In addition of this, as we show below, SCP symbolic tools allow to get not only a graphical output but also a mathematical expression for curves and surfaces.

Mathematica [2] has emerged as the most popular and widely used symbolic computation program. In this paper, we present a Mathematica package, `CAGD.m`, for learning and teaching CAGD and Computer Graphics. The package incorporates many commands for the treatment of the most usual curves and surfaces in CAGD (Bézier, B-spline, rationals, etc...). Some classical geometric operations with interesting applications to real-world problems have also been performed. In the following paragraphs, we describe the main features of the package and show some interesting and illustrative examples of its application.

## 2. Description of the Package

Tha package `CAGD.m` to be introduced in this paper has been implemented in the Mathematica programming language. We think that Mathematica is an ideal environment for developing programs for computer graphics, because of its powerful programming language (supporting procedural, functional, based-on rules, based-on pattern recognition and object-oriented programming), and because it is easy to understand, it reproduces many C language structures, it has several remarkable graphical capabilities and, mainly, it works *in a symbolic way*. This important feature will be used later to obtain the symbolic expression of some curves and surfaces.

### 2.1. How Does the Package Works?

Perhaps the best way to understand how the package has been built is to take a look to some of the routines it incorporates. For example, let $\mathbf{P}=\{\mathbf{P}_0,\mathbf{P}_1,...,\mathbf{P}_n\}$ be a set of points in $\mathbf{R}^d$, d=2,3 ($\mathbf{R}$ meaning the set of real numbers). As already know (see [3]) the Bézier curve associated with the set $\mathbf{P}$ is defined by:

$$\sum_{i=0}^{n} \mathbf{P}_i B_{i,n}(t) \qquad (1)$$

where $B_{i,n}(t)$ represents the Bernstein polynomials, which are given by:

$$B_{i,n}(t) = \binom{n}{i}(1-t)^{n-i}t^i \quad i = 0, \ldots, n \qquad (2)$$

$n$ being the polynomial degree. The points $\mathbf{P}_i$ are called *control points* or *Bézier points* of the curve. These expressions can be easily translated to Mathematica. For example, Equation 2 becomes:

```
Bernstein[i_,n_,v_]:=
 Binomial[n,i]*(1-v)^(n-i)*v^i
```

Table 1. Code for the Bernstein polynomials.

while Equation 1 takes the simple form:

```
BezierCurve[pt:{{_,_}..}|{{_,_,_}..},v_]:=
 Module[{n=Length[pts]-1},
 Simplify[
 Table[Bernstein[i,n,v],{i,0,n}].pt
 ]
 ];
```

Table 2. Code for the BézierCurve command.

Let us analyze the previous code: in line 1 we introduce the variables, namely `pt` indicating the (two- or three-dimensional) points, and `v` the variable of the curve. After obtaining the curve degree (line 2), we translate Eq. 1 to Mathematica. Thus, line 4 corresponds to the scalar product of two vectors: the table with the Bernstein functions and the vector of points. Finally, the result is simplified.

From the previous codes, it becomes clear that:

1. Mathematical expressions (as Eqs. 1 and 2) can be easily implemented in Mathematica; in most cases, this process consists of a simple translation of these expressions to its programming language.
2. Note also that the previous code allows us to work in a symbolic way. For example, after loading the package:

   ```
   In[1]: <<CAGD.m
   ```

   we can obtain the following mathematical expressions for the Bernstein polynomial and the Bézier curve:

   ```
   In[2]: Bernstein[1,3,t]
   ```
   *Out[2]:* $3(1-t)^2 t$

   Fig. 1 shows the Bernstein functions of degree 4:

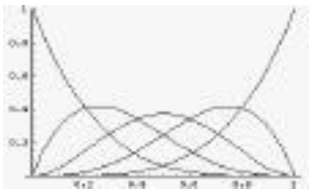


Figure 1. Bernstein polynomials for n=4.

```
In[3]: pts={{1,1},{2,4},{3,0},
 {4,3},{5,-1},{6,0}};
In[4]: BezierCurve[pts,t]
```
*Out[4]:* $\{1+5t,\ 1+15t-70t^2+140t^3-140t^4+54t^5\}$

We remark here that we have obtained the symbolic formula for the Bézier curve. This fact represents a clear advantage with respect to the traditional programming languages. These mathematical formulas are exact, because they can be obtained by, more or less complex, symbolic operations. Since many computer graphics methods are very susceptible to round-off errors, this capability is specially valuable.

Plotting the corresponding curve reduces to this simple input:

```
In[5]: PlotBezierCurve[pts,t]
```
*Out[5]: Figure 2(left).*

Figure 2(left) shows a Bézier curve, its Bézier points and the control polygon (a set of segments joining the control points of the curve).

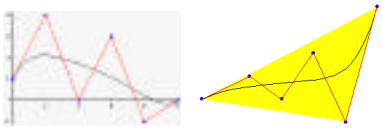


Figure 2. (left) A Bézier curve; (right) Convex hull property of a Bézier curve

An interesting property of the Bézier curves is the *convex hull property*: the curve lies entirely within the convex figure set by the extreme points of the polygon determined by the control points (see [1, 4, 6]). This fact is illustrated in Figure 2(right), where the grey area indicates such a convex figure. To get this figure, we introduce the control points and obtain the plot of the corresponding Bézier curve:

```
In[6]: p1={{1/2,1},{2,3/2},
 {3,1},{4,2},{5,1/2},{6,3}};
In[7]: cur=PlotBezierCurve[p1,t];
```

Then, we load the Mathematica package for calculating the convex hull of the given points. Finally, we display the resulting area (in grey) and the curve:

```
In[8]: <<DiscreteMath`
 ComputationalGeometry`
In[9]: ConvexHull[puntos];
In[10]: Show[
 {Graphics[
 {RGBColor[1,1,0],
 Polygon[%]}],cur}
 ]
```
*Out[10]: Figure 2(right)*

In computer graphics, curves and surfaces are usually dealt with by using numerical routines, which are required to be as short, fast, elegant, numerically stable and general as possible. For example, the problem to evaluate a Bézier curve at a given parameter value $t=t_0 \quad [0,1]$ (say t=0.3) is solved by applying the de Casteljau algorithm (see [3], Chapter 4). However, in our case this problem just reduces to compute:

```
In[11]: BezierCurve[pts,0.3]
```
*Out[11]: {2.5,1.97722}*

3. We have used pattern recognition to identify the dimension of the points through the pattern `{_,_}` or `{_,_,_}`. Moreover, the pattern `{{_,_}..}` (`{{_,_,_}..}` resp.) indicates that one or more pairs (triplets resp.) are expected. Otherwise, no evaluation is performed:

```
In[12]:r={{1,1},{2,4,3,2},{4,3}};
In[13]: BezierCurve[r,t]
```
*Out[13]: BezierCurve[{{1,1},{2,4,3,2},{4,3}},t]*

Finally, the operator "`|`" (line 1 in Table 2) is used to indicate "this pattern" or "this another one". This means that the same commands are able to work with two- and three-dimensional curves and surfaces:

```
In[14]: PlotBezierCurve[{{1,4,4},
 {2,2,3},{3,0,3},{3,4,0},
 {4,5,2}},t]
```
*Out[14]:*

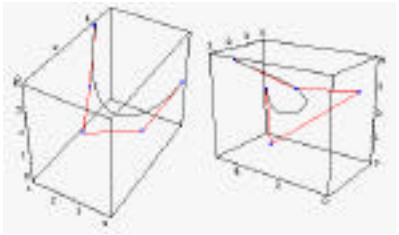


Figure 3. Two different viewpoints of a three-dimensional Bézier curve.

4. The graphical Mathematica capabilities can be successfully applied to visualization. They include the rendering of parametric, implicit and explicit curves and surfaces, colors, grids, different points of view (see Figure 3), shading, shadows, intersection algorithms, etc. In addition of this, the package incorporates many other options, as two- and three-dimensional transformations, projections, perspectives, visible lines and surfaces techniques, etc.

We remark that the previous examples have been chosen primarily for simplicity and illustrative purposes, rather than to correspond to valuable codes or very complicated algorithms. However, very similar discussions can be applied to many other commands. As an illustration, Table 3 shows the corresponding code for a Bézier surface. In this code, we have used the functional programming, a powerful alternative to the standard procedural programming. Note that the pattern recognition and functional Mathematica programming avoid conditionals and loops, respectively. Motivated by the different kinds of programming that Mathematica can support, the final syntax of our commands is very short, elegant and easy to understand.

```
BezierSurf[pts:{{{_,_,_}..}..},v1_,v2_]:=
 Module[{m=Length[pts]-1,U,V,
 n=Length[First[pts]]-1},
 {U,V}=MapThread[Table[
 Bernstein[i,#1,#2],{i,0,#1}]&,
  {{m,n},{var1,var2}}
  ];
        Plus @@ (U.pts*V) //Simplify
 ]
```

Table 3. Code for the Bézier surface.

| x | y | z | x | y | z | x | y | z | x | y | z | x | y | z | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 3 | 0 | 3 | 3 | 0 | 4 | 2 | 0 | 5 | 1 |
| 1 | 0 | 2 | 1 | 1 | 3 | 1 | 2 | 4 | 1 | 3 | 4 | 1 | 4 | 3 | 1 | 5 | 2 |
| 2 | 0 | 3 | 2 | 1 | 4 | 2 | 2 | 5 | 2 | 3 | 5 | 2 | 4 | 4 | 2 | 5 | 3 |
| 3 | 0 | 3 | 3 | 1 | 4 | 3 | 2 | 5 | 3 | 3 | 5 | 3 | 4 | 4 | 3 | 5 | 3 |
| 4 | 0 | 2 | 4 | 1 | 3 | 4 | 2 | 4 | 4 | 3 | 4 | 4 | 4 | 3 | 4 | 5 | 2 |
| 5 | 0 | 1 | 5 | 1 | 2 | 5 | 2 | 3 | 5 | 3 | 3 | 5 | 4 | 2 | 5 | 5 | 1 |

Table 4. Set of 3-D points of control.

The following picture shows a typical output for the `BezierSurf` command with the control points given in Table 4 and stored in the variable `pts3d`:

```
In[15]: BezierSurf[pts3d,{u,v}]
```
*Out[15]: {4 v,2 u,2 v (3+(-3+u) v)}*

The corresponding patch surface is shown in Figure 4(left):

```
In[16]:PlotBezierSurf[ptos3d,{u,v}]
```
*Out[16]: Figure 4(left)*

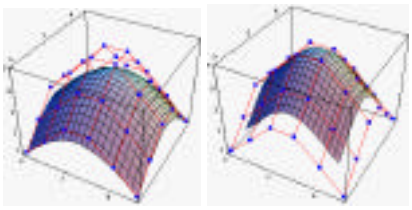


Figure 4. (left) Bézier patch; (right) B-spline patch.

Of course, many other commands (which are not described here for limitations of space) for dealing with the most usual curves and surfaces in CAGD have also been implemented. Figure 5 shows two Mathematica palettes with the most important commands included in the package.

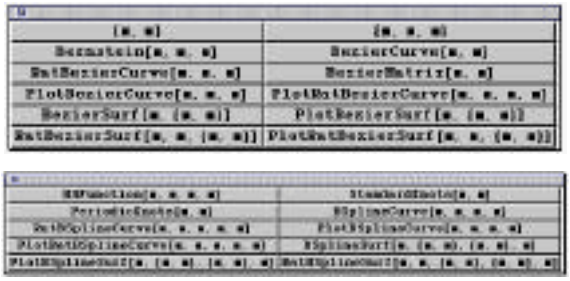


Figure 5. Some CAGD package commands.

In the next section these commands will be applied to solve some interesting and illustrative examples.

## 3. Some Examples of Application

This section is devoted to show some examples of application of the CAGD Mathematica package described in the previous paragraphs.

### 3.1. Two-dimensional transformations

In this example we start with an initial square, given by joining the list of points:

```
In[17]: square={{-5.,-5.},{5.,-5.},
 {5.,5.},{-5.,5.},{-5.,-5.}};
```

Now, some two-dimensional transformations are applied to the previous list (see [4], Chapters 2 and 3, for a survey on two- and three-transformations). First of all, we consider a proportional 2D-scaling (given by the `Scaling2D` command) of factor `Sqrt[68]/10`. Then, a rotation (`Rotation2D` command) of angle *arctan(5/3)*- *π/4* is applied. The `NestList` command is finally applied to produce a 40-steps iteration of this process, as follows:

```
In[18]:s=NestList[Rotation2D[
 Scaling2D[#,
 {Sqrt[68]/10,Sqrt[68]/10}],
 ArcTan[5/3]-Pi/4]&,
 square,40];
```

obtaining the Figure 6 by using the graphical Mathematica commands:

```
In[19]:Show[Graphics[ListPlot[#,
 PlotJoined->True,
 Axes->False,PlotRange->All]
 ]& /@ s,AspectRatio->1]
```

*Out[19]:*

Figure 6. Iterative process of scaling and rotation two-dimensional transformations.

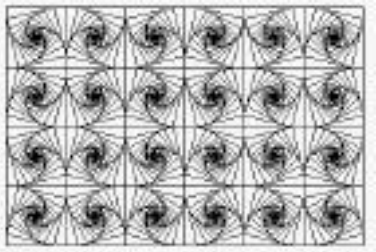


Figure 7. Mosaic obtained by the application of some simple (rotations, scalings and reflections) two-dimensional transformations.

Finally, reflecting and traslating Figure 6 (by using the `Reflection2D` and `Traslation2D` commands) Figure 7 is obtained. As the reader may appreciate, the previous transformations can be achieved in a very easy and intuitive way, by the simple application of the Mathematica and `CAGD` package facilities.

### 3.2. The Spinning Top Example

Given the list of points:

```
In[20]:spitop={{0,16},{1,16},
 {1.5,16},{1.5,16},{1.5,16},
 {1.5,16},{1.5,14.5},{1.5,14.5},
 {1.5,14.5},{1.5,14.5},{1.5,14.5},
 {2,14.5},{3.5,14},{5,13},{5.5,11},
 {4.5,8.5},{3,7},{1.5,6},{1,4.5},
 {0.5,3},{0,1.5},{0,0}};
```

the mathematical expression for its Bézier curve is given by:

```
In[21]:BezierCurve[spitop,t]
```

*Out[21]:* {21 t-105 $t^2$+2992.5 $t^4$-20349 $t^5$+81396 $t^6$-232560 $t^7$+508725 $t^8$- 881790 $t^9$+ 1.2345 $10^6$ $t^{10}$-1.23451 $10^6$ $t^{11}$+146965 $t^{12}$+ 2.34013 $10^6$ $t^{13}$-4.593 $10^6$ $t^{14}$+ 4.17833 $10^6$ $t^{15}$-1.30234 $10^6$ $t^{16}$- 1.21196 $10^6$ $t^{17}$+ 1.67114 $10^6$ $t^{18}$-915600 $t^{19}$+259717 $t^{20}$-31662.5

$t^{21}$, 16-81396 $t^6$+1.04652 $10^6$ $t^7$- 6.40993 $10^6$ $t^8$ + 2.46901 $10^7$ $t^9$- 6.66633 $10^7$ $t^{10}$+ 1.33327 $10^8$ $t^{11}$- 2.0384 $10^8$ $t^{12}$+2.42764 $10^8$ $t^{13}$-2.27734 $10^8$ $t^{14}$ + 1.69277 $10^8$ $t^{15}$-9.98322 $10^8$ $t^{16}$+4.6437 $10^7$ $t^{17}$-1.666 $10^7$ $t^{18}$+4.36432 $10^6$ $t^{19}$-737383 $t^{20}$+59270.5 $t^{21}$}

which is a polynomial of degree 21, obtained from the 22 control points (see Figure 8(left)). A 2D-reflection is then applied in order to obtain a whole spinning top, as shown in Figure 8(middle). Finally, several 2D-rotations can be applied for simulating the movement of the spinning top in a plane (see Figure 8(right)). Finally, Figure 9 shows four different frames of a movie generated from the 3-D transformation commands implemented in the package.

Figure 8. Simulation of the spinning top movement: three different steps.

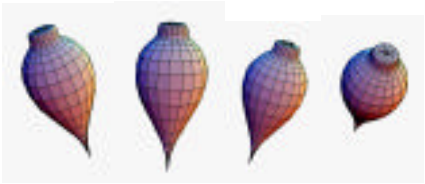


Figure 9. 3-D movie of the spinning top.

### 3.3. B-spline Curves

In general, Bézier curves are not appropriate for computer design because no local control can be performed, that is, moving the position of one control point alters the shape of the whole curve (see Figure 10(left)). Moreover, the resulting curve degree, *n*, is totally determined by the number of the control points (see [1,3-6]). Hence, it is more interesting for computer graphics to consider B-spline curves, whose degree can be chosen and that exhibit local control, that is, if a control point is moved only the closest parts of the curve to this point are affected, as shown in Figure 10(right).

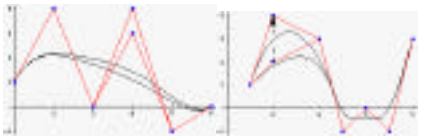


Figure 10. Different control for the curves: (left) Bézier: global control; (right) B-spline: local control.

As another example of B-spline curves, Figure 11 shows three different pictures of a squirrel obtained by using the `PlotBSplineCurve` command. They differ in the options, `ControlPoints` and `ControlLines`, used to display a given picture, which can take the values `True` or `False`, indicating whether or not the picture is displayed with the control points and/or lines.

Figure 11. Example of B-spline applied to design.

In addition of this, different choices of the knots vectors, control points, and orders can be taken, for both curves and surfaces. This will be illustrated in the case of a surface in the next example.

### 3.4. B-spline Surfaces

Figure 4(right) showed a typical example of a B-spline patch, which has been built as a tensor product of two B-spline curves (see [1, 3, 4, 5] for more details). In this case, a periodic knot vector has been chosen for both axes. Compare with the Bézier patch defined by the same control points (Figure 4(left)).

The implemented commands allow us to choose many different options. As an illustration, Figure 13 shows two B-spline surfaces: on the left, a (3,3)-order with two nonperiodic knot vectors surface; on the right, a (5,2)-order with a nonperiodic and a periodic knot vectors. Differences between them can be visually stablished.

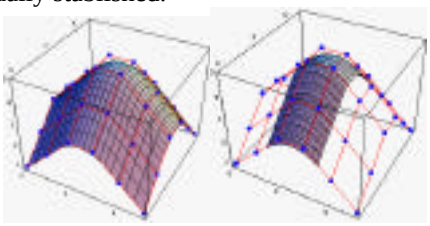


Figure 13. Two different choices of orders and knot vectors for a B-spline surface.

### 3.5. Blending Surfaces

If a single parametric surface is not able to approximate a given set of points sufficiently well, then we may use several patches which are joined together. Figure 14 illustrates this blending process. Two Bézier surface patches $\mathbf{F}_0$ and $\mathbf{F}_2$ (see Figure 14 (left)) are connected using a Bézier $\mathbf{F}_1$ with $C^1$ continuity. Following [3], pag. 269, this connection can be achieved by a careful choice of the control points for the Bézier surface $\mathbf{F}_1$. Figure 14(right) shows the resulting blending surface.

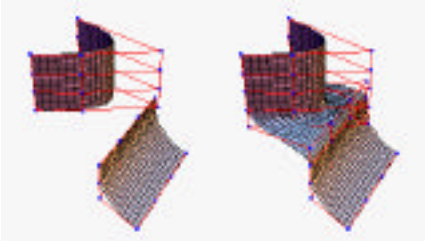


Figure 14. $C^1$-continuous connection of two Bézier surfaces.

Once again following [3], Figure 15 shows another example of blending surface: in this case, it is given by two different curves, a circle and a ellipse.

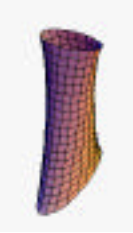


Figure 15. Blending an ellipse and a circle.

### 3.6. The CAGD Package in Industry

In many industrial areas, geometric information is stored, transmitted and manipulated in different standard formats, like IGES, VDA, CATIA, etc. Each of these standard formats support a different mathematical representation of the objects. For example, VDA uses the monomial representation, whereas IGES supports the B-spline one (see [7] for more details on the IGES format). Since the CAGD package is able to deal with this last description, we have developed an IGES-Mathematica converter. This program allows us to have access to some geometric entities of the electronic files from the automotive or aerospace industries. After reading the files, we can apply all the above described commands to these geometric entities and introduce the students to the industrial world.

## 4. Interaction with New Computer Graphics Education Resources

Nowadays, the new technologies are changing the usual teachers' resources for computer graphics and visualization education. Among them, we consider the use of the virtual reality languages and Internet.

### 4.1. Conversion to VRML language

Perhaps one of the most recent innovations in computer graphics education is the appearance of the virtual reality languages. By using these languages, like VRML, you can navigate around and through the graphic and manipulate it in a very easy and intuitive way. In virtual reality, the user "lives" in the scene. Evidently, this opens many possibilities, some of them still unexplored, not only technical but also in terms of students' motivation. E. Donley, at IUP, has developed a Mathematica package that converts Mathematica 3D graphics into VRML format and writes it to a .wrl file. This file can be viewed using a variety of VRML browsers, like Netscape. For further information, we refer the reader to [8]. This program allows the students a better manipulation of the graphics, adding many options not available in the Mathematica outputs.

### 4.2. Use of Internet

Today, many teachers are concerned about the possibility to take advantage of Internet as a vehicle for communication and education. Indeed, there is a large collection of Java programs and Web pages for computer graphics education. In this sense, Mathematica 3.0 and higher also allow to convert any standard Mathematica notebook to HTML. Unfortunately, pictures are fixed as they are converted to GIF and JPEG formats. However, there is a free Java applet, `LiveGraphics3D`, by M. Kraus [9], for incorporating some kind of interactivity (display and rotation) to these pictures in your Web pages. Some examples can be found in [9].

## 5. Conclusions

In this paper we have introduced a Mathematica package for learning and teaching CAGD and Computer Graphics. This package constitutes, as far as we know, the first attempt to bring SCPs, and specially Mathematica, to the Computer Graphics community in an extensive way. This work arose from our feeling that SCPs could be successfully applied in computer graphics and visualization education. After using the packages in some regular courses, our experience was so satisfactory that we are convinced now that this is one of the best tools for introducing students to CAGD and computer graphics.

The weakest part to be appreciated in this work is related with the requirements in memory and computation time that Mathematica requires, in comparison with other programming languages. On the contrary, the Mathematica code is shorter and easier to understand, and many mathematical functions are already incorporated. For example, the students can immediately obtain a graphical

representation of the problem under analysis and its solution without using any external visualization program.

## Acknowledgements

Authors would like to thank the CICYT (Comisión Interministerial de Ciencia y Tecnología) of the Spanish Ministry of Education (project TAP98-0640), the European FEDER funds (project 1FD97-0409) and University of Cantabria for finnancial support of this work.

## References

[1] V. B. Anand. *Computer Graphics and Geometric Modeling for Engineers*, John Wiley and Sons, New York, 1993.

[2] S. Wolfram. *The Mathematica Book*. 3rd. ed., Wolfram Media / Cambridge University Press, 1996.

[3] J. Hoscheck and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* A.K. Peters, Massachussetts, 1993.

[4] D.F. Rogers and J.A. Adams. *Mathematical Elements for Computer Graphics*. Second Ed., Mc Graw-Hill, New York, 1989.

[5] L. Piegl and W. Tiller. *The NURBS Book*, Second Ed., Springer-Verlag, Heidelberg, 1997.

[6] G.Farin. *Curves and Surfaces for Computer Aided Geometric Design,* Academic Press, Orlando Fl, 1988.

[7] IGES/PDES Organization. *The Initial Graphics Exchange Specification (IGES). Version 5.1*. National Computer Graphics Association. Virginia, USA, 1991.

[8] E. Donley. *VRMLConvert program*. Web Page: http://www.ma.iup.edu/people/ hedonley. html

[9] M. Kraus. *LiveGraphics3D program*. Web Page: http://theorie3.physik.uni-erlangen.de/ ~mkraus/Live.html

# The Development of a Digital Library to Support the Teaching of Computer Graphics and Visualization

**G. Scott Owen**, **Raj Sunderraman** and **Yanqing Zhang**
Department of Computer Science
Georgia State University
Atlanta, Georgia USA
owen@siggraph.org

**Abstract**

*The ACM SIGGRAPH Education Committee web site is being evolved into a Digital Library that will be a worldwide resource for Graphics and Visualization Education. We discuss the history of the website, including a long-standing partnership with the National Science Foundation, and the current work to evolve it into a Digital Library. Finally, we discuss a proposed vision of the future of the Digital Library and how we plan to achieve that vision.*
**Keywords:** *Digital Library, XML, graphics and Visualization education.*

## 1. Introduction

The National Science Foundation (NSF) in the United States, in conjunction with other federal agencies, has started a major research initiative on the design and creation of Digital Libraries. As part of this initiative, the NSF Division of Undergraduate Education (DUE) has a vision of a national Digital Library to support education in Science, Mathematics, Engineering, and Technology (SMET). The goal is to leverage previously funded NSF projects that have developed curriculum materials, and to combine the results of these projects, as well as future such projects, with research into Digital Libraries to develop useful Digital Libraries to support education in SMET. Since these Digital Libraries will be based on the World Wide Web (WWW) they will be international in nature.

This paper addresses the effort to create and enhance a Digital Library to support education in Computer Graphics and Visualization. This is a joint effort between the NSF and the ACM SIGGRAPH Education Committee. As part of the SIGGRAPH Education Committee web site (www.education.siggraph.org) we have had a Computer Graphics Courseware Repository (CGCR) for several years. This has now been extended to Visualization, becoming the Computer Graphics and Visualization Courseware Repository (CGVCR). The CGVCR is being enhanced and is evolving into the SIGGRAPH Education Committee Digital Library (SECDL). This work builds on previous cooperative work between the NSF and the Education Committee.

There are two main tasks involved in the creation of a Digital Library to support education. The first is the assembling and/or production of relevant educational materials and placing them onto a web site. The second task is the evolution of this web site into a Digital Library.

## 2. Digital Libraries

A Digital Library differs from a database, web site, or online repository in the way it is structured and the way that users can find, access, and retrieve the information. As the amount of materials increases, the ability to locate the needed materials, or even to browse the information, becomes more difficult. New and innovative methods of navigation are needed. In addition, the materials should be flexible enough to be reused or repurposed, i.e., reconfigured and easily delivered in different sized "chunks" of information. Finally, there should be a sense of community among the developers and users of the materials. The vision of a Digital Library is that of a repository of materials wherein all of this is possible.

Now let us apply this vision to graphics and visualization educators. A professor teaching a course in global illumination wants to find material on ray tracing caustics. She goes to the SECDL, does a search, and retrieves teaching materials consisting of text, images, and videos illustrating the principles. She also finds relevant portions of SIGGRAPH Conference Course notes and links to original research papers that are housed at the main ACM SIGGRAPH site (www.siggraph.org). She decides to visit the SEC World (see later discussion) to see who might be around. There, she meets an Art professor who, as

she discusses her work, decides that he is also interested in the same topic, but from an art perspective. He goes to the SECDL and retrieves a set of materials that also cover caustics, but that are suitable for his art classes.

Another professor teaching visualization wants to find images of all visualization techniques that produce images that have an appearance similar to streamlines. So, he performs a query by image content and finds a set of images, ranked in similarity order.

Finally, a professor teaching human modeling techniques wants to illustrate proper and realistic poses, but is unsure of just how to do it. She goes to the CGVCR and finds a section on Contrapposto with images comparing Egyptian and Greek sculpture, as well as some by Donatello.

The common thread in all of the above is that the professor may have an exact idea of what he wants, or only a vague idea and may just be browsing. In all cases the SECDL is able to support their needs.

Computer Graphics and Visualization are unique in that they are of interest to a wide variety of disciplines, e.g., engineers, mathematicians, computer scientists, natural and physical scientists, artists, graphics designers, animators, etc. Thus, educators from all of these disciplines need to have access to materials to teach their own view of these topics. Ideally, the materials could be reused and tailored to the different views.

## 3. Previous and Current Efforts on Increasing the Materials in the CGVCR

Previous efforts were focused on building up the amount of educational materials available on the Education Committee web site. The Education Committee and the NSF have collaborated on several previous projects to provide faculty enhancement workshops in Computer Graphics (1990 (NSF USE-8954402), 1993, and 1994 (NSF DUE-9255489), and Scientific Visualization (1996 and 1997 (NSF DUE-9554692). While the primary thrust of these projects was the faculty workshops, some educational materials were created and placed on the SECDL (HyperGraph and HyperVis). In addition, there has been some donation of materials from other sources.

We have intensified our efforts to gather and/or produce educational materials to place on the SECDL. A recent effort, funded by the NSF (DUE-9752398), has been a project to develop instructional materials to teach concepts of Visualization. These materials are being developed in conjunction and accord with the recommendations of the Education Committee **Committee on Education for Visualization**, Chaired by Gitta Domik, University of Paderborn, Germany. In June 1996, The U.K. Advisory Group on Computer Graphics (AGOCG) lead by Anne Mumford, held a workshop on computer graphics and visualization education. Several of the recommendations of that workshop are being implemented in this project. In addition, Ken Brodlie, University of Leeds, was funded by AGOCG to develop a web site on Visualization Techniques, which has now been added to the SECDL.

We have begun to place more materials from the annual SIGGRAPH Conference on the SECDL, e.g., Educational Slide sets, and some SIGGRAPH Conference Course notes. We plan to put papers from the Educators program on the site. We also plan to place some digitized versions of SIGGRAPH Video Reviews on the SECDL. Of course, all of the above depends upon getting proper copyright permission from the original authors. We also have been in contact with the organizers of the annual IEEE Visualization conference and will try to place some of these tutorials, etc. on the SECDL.

In summary, the SECDL has a rich set of multimedia learning materials. However, the very richness and size of the SECDL, and the different viewpoints and needs of the educators that want to access it, makes the material not as accessible and flexible as desired, which is the motivation for these projects.

## 4. Digital Library Project

We have just begun another set of projects, in collaboration with the NSF to evolve the CGVCR into the SIGGRAPH Education Committee Digital Library (SECDL). As discussed above, this primarily means improving the navigation and retrieval capabilities and the flexibility of the materials. Some of this work is currently funded by the NSF DUE-9816443), while some has been proposed and we have not yet begun work on it.

### 4.1. Current SECDL Projects

These projects involve improving the navigation and information retrieval aspects of the SECDL. We are approaching this in several ways. We have implemented a conventional search engine (InfoSeek). However, this type of technology is limited so we are performing research on two other types of navigation and retrieval.

It is the opinion of many people that database technology will play a critical role in Digital Libraries. However, there is one important difference between the information available on the Web and the information usually stored in a traditional database: the lack of "structure" in the Web information and the strong "structure" to the information stored in a database. Recently, the term "semi-structured" data has been coined to refer to the data and information such as those stored on the Web. The SECDL falls under this category of "semi-structured" information. We are investigating two different data base systems that deal with semi-structured data on the Web. We plan to use these systems in conjunction with the SECDL to provide ad-hoc querying and search capabilities.

One system is Lore (Lightweight Object Repository, a new database system, developed at Stanford University, designed to support storage and queries of semistructured data. In Lore, data is self-describing, so it does not need to adhere to a schema fixed in advance. Lore is particularly well suited for document data, including HTML documents available on the World Wide Web.

The second approach is Virtual database technology, a radically different approach to manage and query semi-structured data on the Web. In this approach, textual data is converted to structured relational database data and queries are performed on the relational database. All this is transparent to the user who is interacting with a Web agent which receives search and query requests and processes them by invoking data converter modules (adaptors) and then querying a relational database.

Virtual database technology transforms the Web into a database, using adapters that analyze the HTML from Web sites to present them as relational data sources. The limited descriptive capability of HTML necessitates manual analysis for the creation of adapters. However, with XML (see later discussion), the new standard for Web publishing, the creation of these adaptors can be automated.

We are also investigating Visual Information Retrieval techniques, such as the Query by Image Content (QBIC) system from IBM and the Oracle based VIR system. These systems will allow users to use graphical queries, e.g., either start with an existing image or sketch a new image and request similar images. A user will also be able to search for videos that have parts that match a certain type of motion, as drawn by the requester.

### 4.2. Proposed SECDL Projects

These projects have been submitted as proposals to the NSF Digital Libraries Initiative and, if funded, will commence in January, 2000. We have done, and are continuing, some preliminary work on the projects. These projects extend the above efforts to increase the navigation and information retrieval aspects of the SECDL and also directly address the reusability problem. They also include the creation of SEC World. These projects involve XML its associated technologies.

The eXtensible Markup Language (XML) [1] is a markup language for documents containing structured information, i.e., information that contains both content (words, pictures, etc.) and some indication of what role that content plays. It is different than HTML, which is a markup language for the display of the document information but which specifies nothing about the information content. The XML specification defines a standard way to add markup to documents. Unlike HTML, XML is extensible, i.e., you can create your own tags.

XML can be applied to any application area by creating a Document Type Definition (DTD) for that topic. The DTD is the set of XML tags for the particular area and is specific to that area. DTDs are the technology that allow standard XML tags to be defined, and that permit different platforms and machines to understand XML descriptions and applications. DTDs for specific disciplines have been previously created, e.g., for mathematics [2], chemistry [3], and biology [4].

#### *4.2.1. Material Reuse*

For this project we will create a DTD for computer graphics and visualization. This DTD will be based on the results of the SIGGRAPH Education Committee Graphics Taxonomy Project, led by

Jacquelyn Ford Morie. The goals of the Taxonomy Project are to create a complete Taxonomy for Computer Graphics and Visualization and to have educators from different disciplines create different curricular views of the topics. This DTD will allow us to semantically markup any document in these areas.

Another important part of XML technology is the eXtensible Style Language, or XSL. An XSL style sheet is an XML document containing a set of template rules. These template rules describe how the original XML element node is converted into an XSL element node that can be formatted, styled, and displayed. By having different XSL style sheets, a set of XML documents can be tailored to different views. We will create different XSL style sheets in our project that correspond to the different curricular views of educators wanting to use the resources in the SECDL.

There is a huge effort to produce XML tools in industry and we plan to leverage this effort. These tools are all new and in a state of rapid flux. Over the next six months many more such tools will become available. An example is the IBM site (http://ibm.alphaworks.com) which has a set of XML tools that are freely available.

Even though the Taxonomy Project will not be completed until August 2000, we can begin to develop and test prototype DTD's based on the current version of the Taxonomy Project at the anticipated start of this project (January 2000). As a feasibility study, we have already begun work on this using the current version of the Taxonomy. We have taken part of the section on Rendering (Lighting Models and Texture Mapping) and created a prototype DTD from this. We have found that this is a straightforward process, enough to give us the confidence that, given the final, full Taxonomy, we will be able to create a full DTD.

#### *4.2.2. 3D Navigation of the Information Space*

Navigation in a complex 3D space is a difficult problem. The user needs to be able to have an overview of the complete space and be able to dive down into the details of a particular subspace. The user should also have a trail and know which parts of the space they have already visited. This is now possible with the 3D Graphics technologies associated with XML. There has been some interest in relating XML and VRML technology. There has been a draft DTD created for VRML created by Bullard [5]. Lipkin [6] discusses the use of XSL Stylesheets to dynamically generate VRML worlds to display the results of database queries. Ancona [7] discusses 3DXML as a way of describing websites and other structured information spaces in XML and publishing them in VRML. This would allow for 3D navigation of the information space. For the relationship between XML and 3D Graphics, see also [8].

The objective of this portion of our project is to be able to generate a 3D VRML world that will represent the particular view of the information space generated in the first part of the project. Assume a user in computer science has requested a particular view of a topic area. From the first part of the project, this query will have generated a set of XML/HTML pages and associated materials. This query will also generate a 3D VRML representation of all of the materials in the form of a hierarchical graph structure.

In this VRML world each node might be a sphere that could contain a sub-graph structure. Sub topic areas could be represented as translucent spheres that the user could move into as they traveled. There would be a nested graph structure that could represent an information space of considerable complexity. For example, a top-level node might be *Rendering*. This might then contain a sub-graph with one of the nodes being *Lighting Models*. This could contain another sub-graph with nodes representing *Local Illumination* and *Global Illumination* models, each with additional sub-graphs. The user can navigate in this VRML world to find the different types of materials. The user can pull back and look at the entire space or traverse along the links to look at different volumes of the space.

When the user visits a portion of the space that node/sphere would be marked, e.g., a color change, as having been visited. Since a user can have more than one instance of a VRML browser, they could simultaneously have multiple views of the information space world.

#### *4.2.3. Sense of Community*

Developing a sense of community among Digital Library users and developers was one of the important issues identified at the NSF January, 1999 workshop on Digital Library research issues. There are several ways we have addressed this in the SECDL. The first is a simple listserve that is used for announcements and has some discussion. But a listserve or web discussion page is

asynchronous and not real time. A real time chat room engenders more sense of community, but is text based and not visual. We propose to develop a 3D virtual world that our community will be more interested in using.

Imagine the existence of a fully 3D interactive virtual world (SEC World) and the following possible scenarios:

A computer science professor from California logs in to SEC World. As she wanders around she encounters another avatar who is an art professor from Ohio and also a professor from the University of Leeds, in the U.K., who teaches visualization. They engage in a 3-way discussion about the possibilities of incorporating aspects of each others' disciplines into their own courses. The art professor mentions a book on drawing that illustrates how certain techniques can help a person focus in on what is important in an image and the Leeds professor decides he will look at it and incorporate parts of it into his class. After further discussion the three of them decide to get together later and submit proposals to their respective Universities for interdisciplinary courses that they will jointly develop.

An engineering professor from Colorado enters SEC World for the first time. He is just beginning to teach courses on Engineering Graphics and he is not sure what he is looking for and would like to find out what all this is about. He sees an avatar with a baseball cap that identifies her as an SEC World Guide and goes and asks her some questions. (He knows that she is a guide because the Student Volunteers at the annual SIGGRAPH Conference wear baseball caps to identify them.) She also gives him brief descriptions of the parts of SEC World and answers some of his questions. He thanks her and goes to another room where another engineer is talking with some artists.

The people in the above cases might not be faculty but could just as easily be students who are looking for some interdisciplinary help that might not be readily available at their own institution.

The above is possible by the use of commercially available Multiple User Shared Environment (MUSE) worlds. SEC World could be created and downloaded the first time from the central SECDL. After that, the users could invoke this world and enter it without downloading a large amount of information. A complex world needs a powerful processor and graphics card, but these are becoming very inexpensive. The required bandwidth is not very high since the only information that is being transmitted is the changing coordinates of the avatars and the text chat. If the users use microphones, then the bandwidth requirements are higher, but still not excessive. These commercial worlds are designed for users who communicate to the Internet via 28/56K modem.

Users could choose an avatar in the database or even design a new one and upload it to the server. The special guide mentioned above would not be a person but an intelligent agent who would have access to a large amount of information about the SEC World as well as the SECDL. We are not proposing a full natural language discourse but only a subset.

We have experimented with both Sony Community Place and the Blaxxun [9] community and found that they are similar. If we create a VRML world for one then it is trivial to convert it to the other format.

#### *4.2.4. EduCause IMS*

A final proposed project relates again to navigation and retrieval issues. We have been in contact with the developers of the EduCause Instructional Management System (IMS), an XML based system of metadata for general instructional navigation and retrieval. EduCause is proposing that IMS become a standard for all educational sites. We are planning on using the SECDL to act as a multimedia testbed for IMS version 1.0. We will extend the general IMS metadata to be more specific for topics in graphics and visualization. This project is a collateral project with those discussed above and has also been submitted to NSF as part of a consortium effort led by the Collegis Research Institute.

## 5. Conclusion

The evolution of the current CGVCR into a full Digital Library, the SECDL, will provide much greater functionality for graphics and visualization educators. Using new XML and related technologies, we plan on increasing the flexibility and navigation capabilities of the SECDL and enhancing the sense of community among its users. . The SECDL has the potential to be an international resource that can service educators worldwide. There are many research problems that must be investigated and overcome before the

ideal can be realized, however once achieved, the SECDL will provide a prototype of the educational Digital Library of tomorrow.

## References


[1] See for example Natanya Pitts-Moulis and Cheryl Kirk, **XML Black Book,** Coriolis Technology Press, 1999; See also *http://www.xml.com;* *http://www.xmlinfo.com*

[2] http://www.w3.org/Math

[3] http://www.venus.co.uk/omf/cml/ also see Peter Murray-Rust "Chemical Markup Language: A Simple introduction to Structured Documents" **http://www.xml.com/xml/pub/w3j/s3.rustintro.html**

[4] http://www.visualgenomics.com/

[5] Len Bullard, "Draft VRML DTD", http://ariadne.iz.net/~jeffs/vrmLab/Documentation/vrmlDTD.html

[6] Daniel Lipkin "Integrating XML and VRML: A Technical Discussion" http://www.vrml.org/WorkingGroups/dbwork/vrmlxml.html July 30, 1998

[7] Dan Ancona, "3DXML Authoring, Publishing and Viewing Structured Information", http://www.vrml.org/WorkingGroups/dbwork/ancona/home.html, August 17, 1998

[8] Web3D Consortium http://www.vrml.org/fs_specifications.htm

[9] Blaxxun Corporation http://www.blaxxun.com

# Learning 3D Visualis a tion with MALUDA

**João A. Madeiras Pereira** and **Mário Rui Gomes**
Instituto Superior Técnico/Instituto Engenharia Sistemas e Computadores
Lisboa, Portugal.
{jap, Mrg}@inesc.pt

**Abstract**

*This paper describes MALUDA system, which constitutes an important tool for teaching 3D-visualisation pipeline at the laboratory classes of the Computer Graphics course at Instituto Superior Técnico.*
**Keywords:** *application tool, 3D visualisation pipeline, modelling, rasterization algorithms.*

## 1. Objectives

The traditional 3D-visualisation pipeline is one of the main topics to be taught in any introductory Computer Graphics course. The MALUDA system constitutes itself as an important complementary tool to learn about 3D visualisation. In fact, it is used at the laboratory classes of the Computer Graphics course at Instituto Superior Técnico, not only as a way to practice the use of a virtual camera model based-viewing pipeline but also as a test environment for rasterization algorithms developed by the students.

## 2. MALUDA main features

There are three main approaches a novice student in Computer Graphics can take to make effective use of MALUDA:

- Learning to create 3-dimensional scenes
- Understanding and observing different realism levels of image rendering
- Testing his own scan conversion strategy.
-

### 2.1. Three-dimensional modelling

A scene can be built up by using the NFF (Neutral File Format) language and/or the SIPP (Simple Polygon Processor) scene description library.

NFF [1] is a low level scene description language in the sense that it provides a minimal interface which allows a programmer to quickly create a scene by describing the geometry and basic surface characteristics of objects, the placement of lights, and the viewing frustum for the eye. There is no hierarchical modelling. An example of a NFF file is illustrated in Appendix A.1.
In addition to support the NFF scene description language, MALUDA also supports SIPP scene description library [2]. A SIPP scene is built up of objects (described in their local coordinates systems) which can be transformed with rotation, translation and scaling. The objects form hierarchies where each object can have arbitrarily many sub-objects. It allows also a scene to be illuminated by an arbitrary number of light sources of different types. In a SIPP scene is possible to create several virtual cameras, and then specify one of them to use when rendering the scene.

An important feature of MALUDA modelling capability is the possibility of merging a SIPP scene with one described in NFF language. Then, a user can ask either to render the resulting scene or to export it in NFF format (containing optionally only triangles).

### 2.2. Image Rendering

MALUDA provides the ability for creating 3-dimensional scenes and rendering them using either the internal scan-line z-buffer algorithm or an external rasterization algorithm provided by the user. By default, the internal rasterization engine is used in the rendering operation. This engine, through the use of the public domain SIPP library, renders images at different realism levels. Scene objects are rendered with either Phong, Gouraud or flat shading. An image can also be rendered as a line drawing of the polygon edges without any shading at all. The SIPP library also provides 3-dimensional texture mapping with automatic interpolation of texture coordinates. Simple anti-aliasing can be performed through oversampling. Transparencies are allowed and several shading functions (“shaders”) are available. Some examples generated by MALUDA by using the internal SIPP rasterization engine are illustrated in Appendix B.

### 2.3. Rasterization support

A major feature of this tool is the ability for a student to provide his own rasterization function. This makes it easy to experiment with various scan conversion schemes and to do special effects.

Two procedures can be used accordingly to the platform to execute the rasterization algorithm is relevant or not for the test. If execution platform is not important, the routine containing the rasterization code developed by the student can be integrated into MALUDA system to be invoked later. This procedure is meant to check the strategy and its performance whatever the machine and operating system. Or, the platform is important, like a parallel machine where we are interested to experiment several parallel schemes for the rasterization stage. In this case, MALUDA should be instructed to output an ASCII file containing device dependent coordinates polygonal database, which will be then used to feed the target rasterization algorithm. An example of this "flat" file is shown in Appendix A.2

## 3. MALUDA architecture

In figure 1, is depicted the MALUDA activity flow diagram

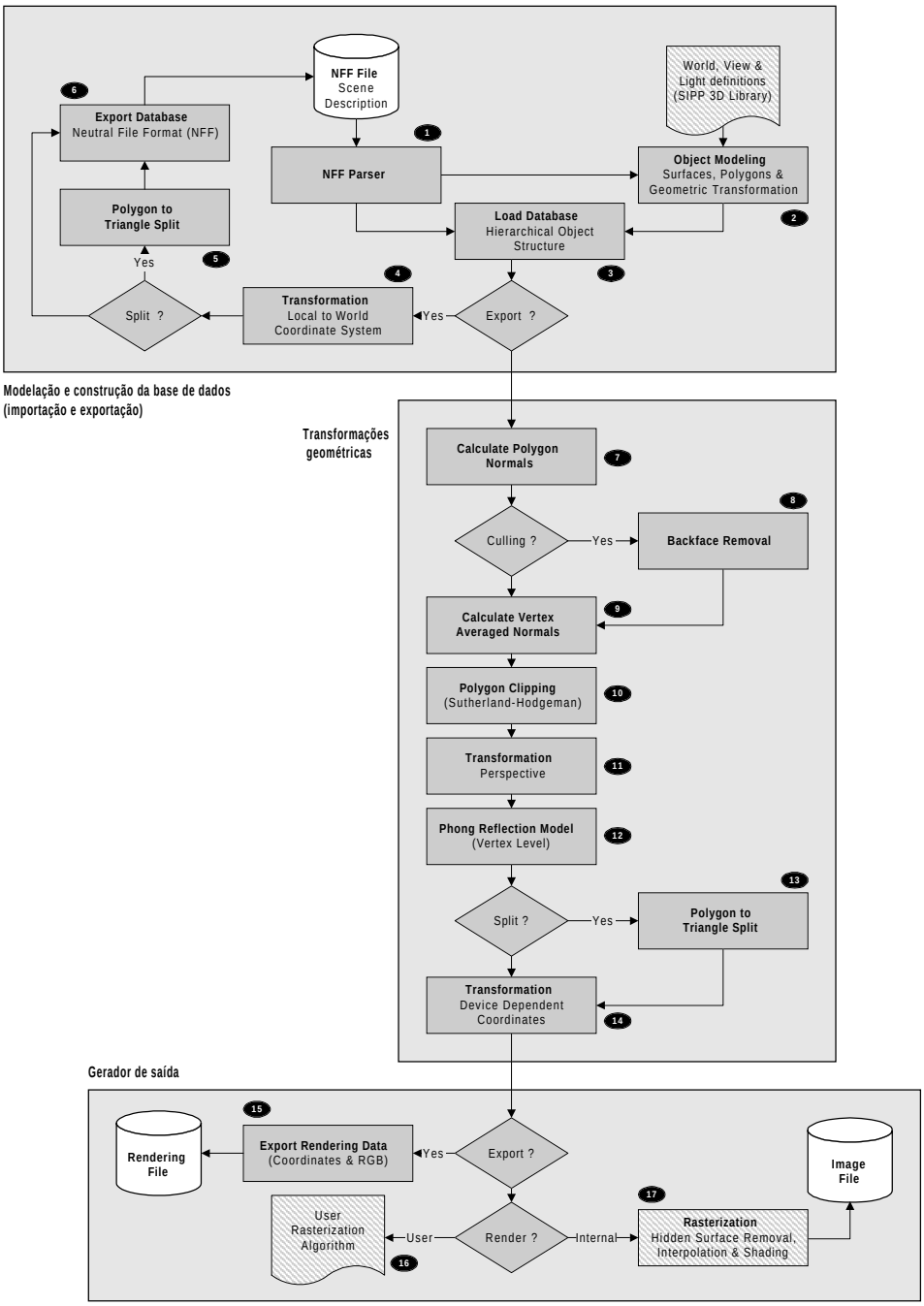



Figure 1 - MALUDA activity flow diagram

An analysis of it shows to a reader the most important modules and their associated functionality. Thus, accordingly to the MALUDA system description made above, we can distinguish the following functional blocks:

- Reader and parser of scenes created by NFF and/or SIPP languages
- Geometric transformer
- Output generator

At this point, is convenient to remind some important features of MALUDA and relate them with the numbered operations illustrated in Figure 1.

MALUDA can export the input scene as a NFF file format (World Coordinates) –operation 6– or as an ASCII polygonal "flat" file format (Device Dependent Coordinates) –operation 15. The system provides also the ability of image rendering by using either the internal SIPP rasterization engine –operation 17- or an external rasterization algorithm provided by the student –operation 16.

It is interesting to mention that when MALUDA is configured to act as image renderer, diagnostic and statistics informations are provided by the system. Appendix A.3 shows the information provided by the system when the default SIPP rasterization engine is being used.

## References


[1] Eric Haines, "A Proposal for Standard Graphic Environments", in *IEEE Computer Graphics and Applications*, Vol 17, nº11, November 1987, pp 3-5

[2] J. Yngvesson, I. Wallin; "User's Guide to SIPP –a 3D rendering library", version 3.1, Public Domain Software via anonymous ftp from isy.liu.se (IP nº 130.236.1.3) in the directory /pub/sipp; March 1993


## Appendix A – MALUDA Input/Output information

### A.1 NFF File Format

```
v                Viewpoint definition;
from  1.02285  -3.17715  -2.1745
at  -0.004103  -0.004103  0.216539
up  -0.816497  -0.816497  0.816497
angle  45
hither  1
resolution 1024  1024
l  1.87607 -18.1239 -5.00042      light source position;
f  1 0.2 0.2 1 0 100000 0 0       surface parameters;
p  3          Polygon with three vertices;
-1  -1  1
-1  -0.9375  0.9375
-0.9375  -1  0.9375
p  3
-0.9375  -0.9375  1
-0.9375  -1  0.9375
-1  -0.9375  0.9375
p  4
12  12  -0.5
-12 12  -0.5
-12  -12  -0.5
12  -12  -0.5
f  1  0.9  0.7  0.5  0.5  45.2776  0  0          New surface;
s  0  0  0  0. 5          a sphere;
s 0.272166  0.272166  0.544331  0.166667
…
```

### A.2 Device Dependent Coordinates Output File

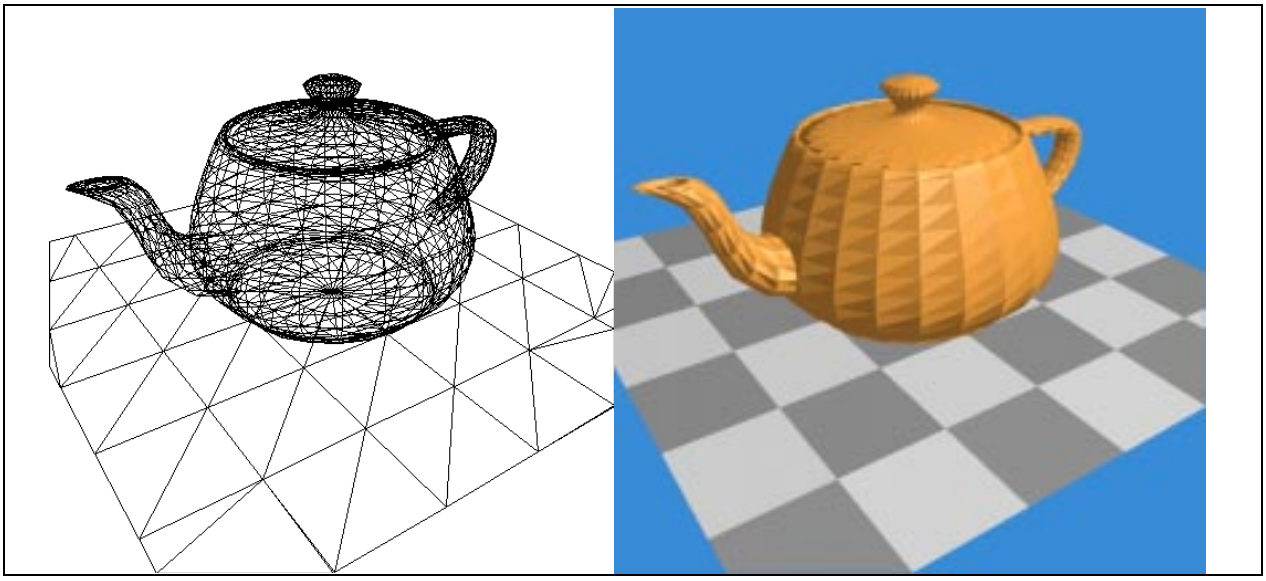
```
512 512 65535 255          Maximum resolution (x,y,z and colour);
120838                     Total number of polygons;
1                          Separator;
3 100 50                   Polygon with 3 vertices and Ymax/Ymin;
40 100 89 30 128 30        (x,y,z) and (RGB)
50 80 89 30 128 30
60 50 89 30 128 30
1
4 110 40                   Polygon with 4 vertices and Ymax/Ymin;
40 40 89 30 128 30
50 110 89 30 128 30
60 50 89 30 128 30
55 60 89 30 128 30
1
127                        End of file;
```

### A.3 Diagnostic and Statistics information

```
NFF scene will be imported!
Pipeline: Parsing NFF file, wait...
Pipeline: Be aware that some SIPP and user defined view parameters
          will be override by NFF parameters!
Pipeline: NFF background loaded
Pipeline: NFF view parameters loaded
Pipeline: NFF light loaded (6.00 6.00 6.00) (1.00 1.00 1.00)
Pipeline: NFF surface with 1 polygons
(0.30,0.60,0.30,5,1.00,0.85,0.70)
Pipeline: NFF surface with 1 polygons
(0.30,0.00,0.20,5,1.00,0.61,0.41)
Pipeline: NFF surface with 145 polygons
(0.30,0.00,0.20,5,1.00,0.61,0.41)
Pipeline: NFF surface with 1 polygons
(0.30,0.00,1.00,5,1.00,1.00,1.00)
Pipeline: NFF surface with 145 polygons
(0.30,0.00,1.00,5,1.00,1.00,1.00)
Pipeline: NFF surface with 1 polygons
(0.30,0.00,1.00,5,0.02,0.67,1.00)
Pipeline: NFF surface with 145 polygons
(0.30,0.00,1.00,5,0.02,0.67,1.00)
Pipeline: NFF surface with 1 polygons
(0.30,0.00,1.00,5,0.21,1.00,0.01)
…
Pipeline: NFF surface with 145 polygons
(0.30,0.00,1.00,5,1.00,0.26,0.75)
Pipeline: Imported 129 surfaces with 9345 polygons, 0 spheres and 0
cones.
Pipeline: NFF parser done
Pipeline: Hither:1.00,Yon:15.00,
Resolution(X,Y,Z,Color):1024,1024,1024,512
Pipeline: Pipeline creating render data, wait...
Timer started           Geometric Transformations start;
Pipeline: Image file to be created: 'pipe.ppm'
Pipeline: Objects loaded: 3
Pipeline: Surfaces loaded: 129
Pipeline: Polygons loaded: 9345
Pipeline: Polygons ready to be rendered by SIPP!          Geometric
Transformations end;
Timer Stopped (Elapsed: 2.108s, User: 2.023s, System: 0.066s)
Pipeline: SIPP rendering, wait...
Timer started           Rasterization stage;
Pipeline: Render Mode: Gouraud Método de sombreamento;
Pipeline: SIPP rasterization done!
Timer Stopped (Elapsed: 112.693s, User: 108.058s, System: 1.053s)

Pipeline: Successful End
Pipeline: Polygons backfacing: 4864
Pipeline: Polygons totally clipped: 39
Pipeline: Polygons splited to triangles: 0
Pipeline: Polygons rendered: 4442
Pipeline: Releasing resources...!
Pipeline: Release complete
```

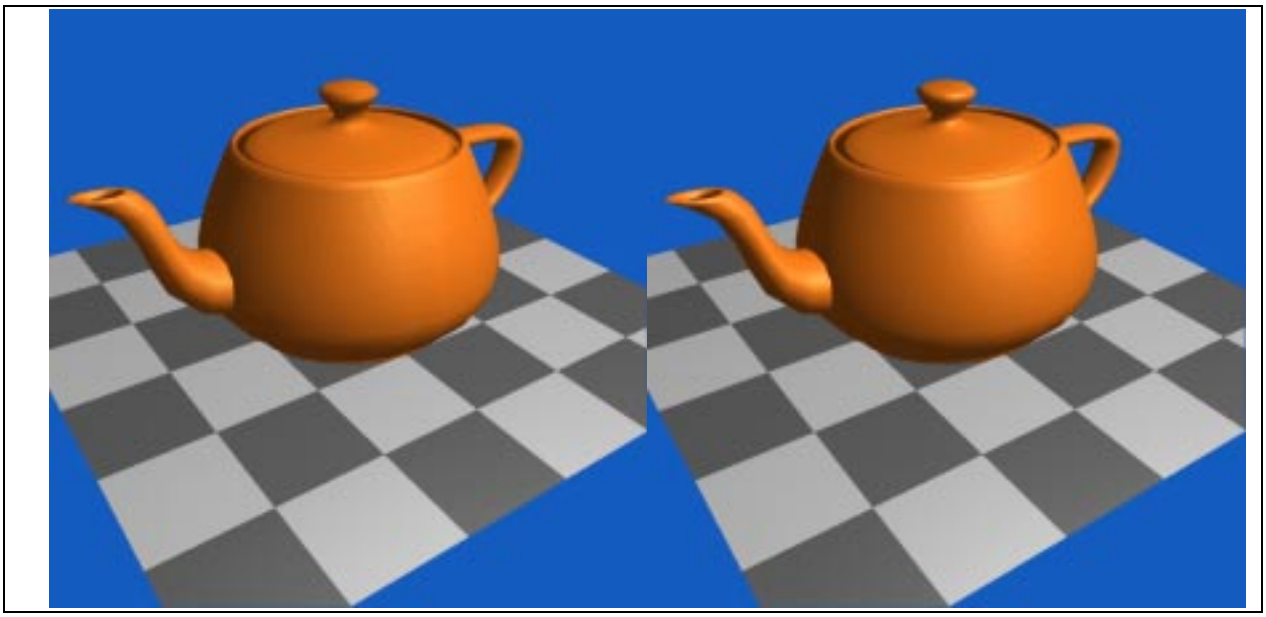
## Appendix B – Different types of image rendering with SIPP rasterization



**Teapot** - *Wire-Frame* model

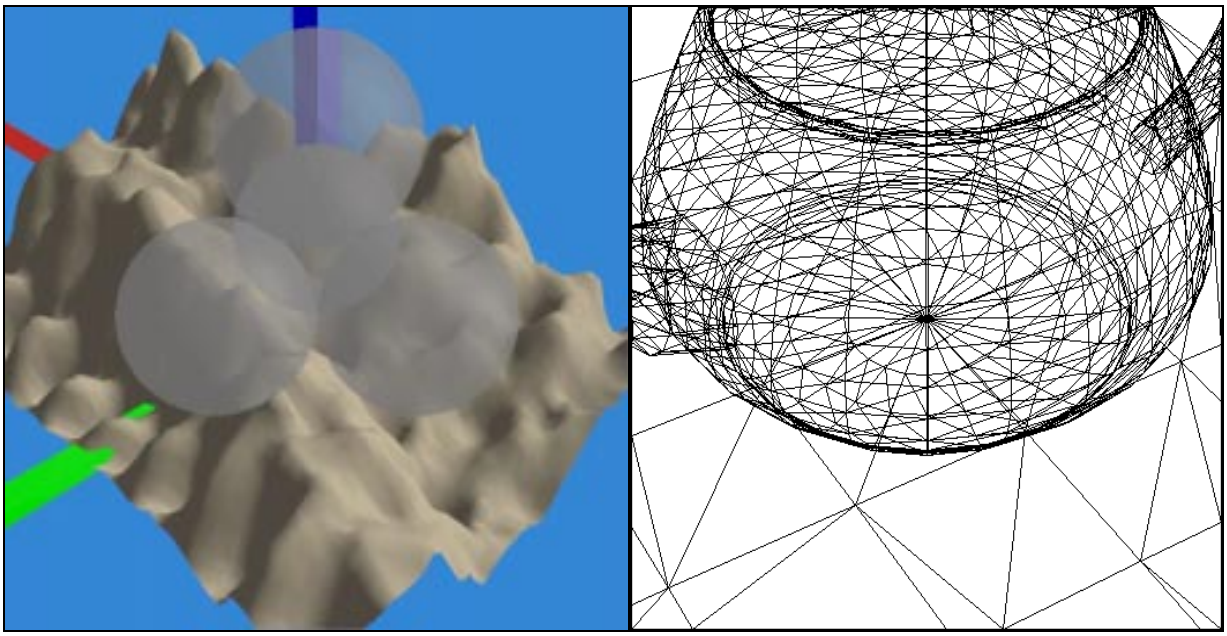
**Teapot** – Flat shading

## Appendix B – Different types of image rendering with SIPP rasterization (cont.)



**Teapot** –Gouraud shading

**Teapot** – Phong Shading



**Mount** with transparent spheres

**Teapot** clipped by Sutherland-Hodgman

# Teaching the Graphics Processing Pipeline: Cosmetic and Geometric Attribute Implications

**Jack Bresenham, Ph.D.**
Winthrop University
Rock Hill, SC 29733 USA
bresenhamj@winthrop.edu

**Abstract**

*The pipeline processing model for computer graphics typically includes stages for interpreting the display list <object entity descriptions, picture abstraction>, geometric manipulation <scaling, rotational, skewing transformations>, raster rendering <scanline conversion, dot image approximation>, and finally presentation <scanline refresh, image display>. At what point in the graphics processing pipeline do attributes actually become associated with picture entities? The answer depends upon the type of attribute and the specification intent for the attribute. I believe it is instructive for students to overtly and consciously become aware attributes such as line width and line style can have various interpretations. Cosmetic attributes are applied post-transformation; geometric attributes are applied pre-transformation. Two different types of clipping also can be considered as scissor clipping in post-rasterization discrete pixel space or as analytic clipping in pre-rasterization continuous space.*

**Keywords:** *computer graphics attributes: cosmetic & geometric*

## 1. Introduction

Is line width a purely cosmetic effect meant to mimic a finely sharpened 'thin' pencil or widely worn 'fat' pencil line thickness? In this case, line width is associated with geometric entities post-transformation. Engineering line weights are cosmetic attributes applied by a draftsman. Cosmetic attributes do not scale, skew, or otherwise distort. Asking for an architectural drawing to be redone on a paper twice as long and twice as wide treats line width as a cosmetic attribute since pens or pencils used will be the same as those used in the smaller original rendering. Pen nib selection or choice of pencil lead width determines line thickness in this cosmetic attribute instance.

Is line width a geometric property inherent in the original display list abstraction meant to mimic photographic enlargement? In this case, line width will be associated with geometric entities pre-transformation. Having created areas for formerly zero-width lines, those areas must be processed through the full transformation stage with the effect that line width area can be scaled, skewed, or otherwise distorted. Pressing the enlargement by 2 button on a xerographic copier applies line width as a geometric attribute. Use of an overhead projector to produce large lines on a viewing screen some distance away from the projector is another example of a geometric interpretation of the attribute line width. Drawing or writing on an inflated balloon then letting much of the air escape is a 3-D example of a shrinking figure size, geometric width reduction.

Do dashes in a line style transform from rectangles to parallelograms with sharp points? Yes, they can, if one elects to apply line style as a geometric or pre-transformation attribute. Treated as a cosmetic or post-transformation attribute dashes will keep their rectangular shape. The basic question to resolve in one's **reference model** is whether a picture is first conceptually rendered, then has attributes applied, and is subsequently transformed and displayed or whether a picture is first transformed, then has attributes applied, and is subsequently displayed.

Both approaches are valid. Students should appreciate the subtleties and interactions involved in each reference model approach for attribute application. I prefer the more mnemonic nomenclature I first heard used by Dr. Brian Middleton: '**cosmetic**' and '**geometric**' attributes.

## 2. Motivation

Presenting simple line style implementation alternatives offers a good opportunity to reinforce transformation concepts. I use rounded, semicircular end caps that transform into elliptical caps for dashed lines as a simple example. One can readily derive the equation for the resulting ellipses …but it can be messy to implement in

graphics hardware or software. If one approximates an original semicircle by chords along its perimeter, then, since lines always transform into lines, implementation is facilitated at reduced cost with concomitant reduced accuracy <fidelity>. The chords do introduce an early approximation; all approximation errors then subsequently propagate with the possibility that recognizable, visual degradation can be the result.

Clipping is another processing stage that can have alternative reference models. Should one first raster, at least conceptually raster, the full picture in infinite raster space then cut out the small viewing window to be actually displayed? Or, should one first clip the picture abstraction entities to an appropriately small viewing window and then raster that post-clip result. The latter is commonplace but it, too, introduces approximation error earlier than need be the case.

A fairly easy demonstration of the two clipping effects can be to first raster a full screen display of a picture. Blank out a small rectangle, perhaps for a pop up menu, then clip the abstract picture to retain just the portion for this small blank rectangle. Finally, render that newly clipped small part of the abstract picture to heal and restore the original full display picture. Discontinuities, likely, and undesirable mismatches off by a few pixels will be visible in analytic clipping.

Alternatively, consider blanking the same area for the small rectangular menu. This time, though, let's assume a fully rastered copy of the original picture has been retained in a second, nondisplayed buffer. Bit-Blit <copy> the pixels temporarily subsumed by the menu back into the displayed picture to restore the area overwritten by the small menu. Likely the result is different from the previously described restoration process; the picture will seamlessly be restored with no discontinuities in scissor clipping.

Here I like the term used by Dr. Bryan Roberts: **scissor clipping** to describe the model in which one first conceptually renders a pixel space picture then clips as is the case with the Bit-Blit approach. The alternative reference model in which one first clips the picture abstraction in continuous coordinates before rendering, then paints the clipped entity in discrete pixel space I refer to as **analytic clipping**.

The graphics pipeline can be used to demonstrate different visible results one can obtain even for the simple attributes line width, line style, and clipping. Students can try a few examples to see for themselves how important a good understanding of one's underlying reference model is. The alternative models also offer opportunity to discuss introduction of early approximations with attendant error accumulation build-up.

## 3. Attribute Reference Model

Before implementing any graphics interface, it is essential to have a clear conceptual model in mind. Otherwise, well-meaning individuals may interpret a common specification differently owing to preconceived notions of where in the graphics processing pipeline an individual assumes attribute associations operations take place.

Clipping produces different visual results depending upon whether one performs a mathematical clip of theoretical, zero width loci of geometric entity boundaries or whether one first rasters then scissor clips the realized picture in raster space. A different picture will be drawn by those who consider a logic operation such as exclusive or to be applied after a polyline is drawn conceptually in an off-screen space then exclusive or'd to the displayed picture frame buffer. Those who consider logic operations to apply in an on-screen pixel by pixel manner dependent upon the algorithmic sequence by which the graphics system generates geometric entities during rendering will typically achieve a different visual effect.

A proper reference model should trivially let us answer questions such as: Do circular dots and rectangular dashes in a line style transform from circles to ellipses and from rectangles to parallelograms? Do wide lines join smoothly or will mitering be appropriate even though sharp spikes may result at the joining point? With a good reference model and agreed upon stages at which attribute actions are to be implemented in the graphics processing pipeline, if we don't like the answer, we should be able to change the reference model to achieve the result we really intended.

## 4. Example: Line Width & Line Style

Let's examine a few implications of two possible reference models that can be used in pipeline processing of Line Width and Line Style and interaction with general transformation and drawing mode. I'll call my alternatives:

1. Geometric <Stretched Rubber Sheet>
2. Cosmetic <Multiple Pen Nib Sizes>

Lines are specified by two end points such as LINE($x_0$,$y_0$,$x_1$,$y_1$) for the line from ($x_0$,$y_0$) to ($x_1$,$y_1$). Lines and any other geometric entity can be transformed by rotation, scaling, translation and shearing by use of a concatenated general 2-D transformation of the form:

$$\begin{bmatrix} u & v & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{matrix} a & c & 0 \\ b & d & 0 \\ e & f & 1 \end{matrix}$$

The result for any affine transformation of a line is another line so, abstractly at least, we would have a line from ($u_0$,$v_0$) to ($u_1$,$v_1$) as the result of transformation. What we don't explicitly know is the effect expected from transformation of either line width or line style. The visual effect achieved by applying attributes before transformation can produce distinctly different pictures from that which would result from applying attributes only after all transformation is completed.

Line width typically is taken to be perpendicular to the defining line. Often specifications for line width fail to specify end point shape for area closure. What, then, shall we transform when dealing with a line width since, in reality, we must treat areas? What other effects interact with the process; what shapes are employed in line style?

Graphic primitives <lines, rectangles, elliptical arcs, ...> enter the pipeline with defining points in an abstract world coordinate space. The defining points imply a complete geometric locus which represents the infinitely thin, geometric abstraction of the graphic primitive form <contour or outline> in original, pre-transformation space. The essence of the pipeline processing is to collect associated attributes <such declarations as line width, line style, drawing mode, and clipping values> then to render conceptually the geometric form. The first conceptual rendering of the full locus in abstract model space is then subjected to an affine transformation <which can be the cumulative result of concatenation of multiple transformations> for composite scaling, rotation, translation, and shearing. The resulting post-transformation, conceptually abstract, picture obtained from the locus transformation potentially is clipped before then being mapped to device coordinate space.

When line width is implied in a geometric sense, graphic primitives such as a line, rectangle or ellipse inherently have an area definition. In the absence of closure of the area owing to lack of line end shape specification, we could consider various possibilities for width processing. An implementation can define an end shape or end cap such as axially sheared, perpendicularly butted, semicircular, or triangular. For a perpendicular butt, we may want to use either a true end point reference or an extended end point reference in anticipation of joining line segments to form, say, a right angle corner.

With an anisotropic transformation, semicircular end caps can be seen ultimately on the display surface as partial ellipses capping ends of lines. As an example, try a reference line segment from (0,0) to (4,0) with a line width of two units on each side of the reference line. Let transformation matrix parameters be: a=2, b=2, c=3, d=1, e=0, and f=0. A pre-transformation semicircle centered at the origin could be the leftmost end cap joining the 'wide line' end points (0,2) and (0,-2); a comparable semicircle centered at (4,0) could join the rightmost end cap points from (4,-2) and (4,2). Notice the two semicircles are drawn anticlockwise (CCW).

Applying the transformation yields boundary wide line segments from (4,2) to (12,14) and from (-4,-2) to (4,10). Should you sketch the result of connecting the post-transformation end points from (4,2) to (-4,-2) with an origin centered semicircle having radius $2\sqrt{5}$ and the other two endpoints from (4,10) to (12,14) with a comparable circle centered at (8,12) you can observe the effect is aesthetically lacking. With dot-dash line styles, it is important to keep the proper sense of directional movement consistent with the effect of a transformation. Notice here that the two semicircles would now properly be drawn clockwise (CW). Line style can cause a viewer to be aware of the direction in which a line is drawn. Attention must be paid to sense of movement after transformation.

A more consistent visual effect is to transform the appropriate semicircle from $x^2 + y^2 - 4 = 0$ into the appropriate portion of the ellipse $10u^2 - 16uv + 8v^2 - 64 = 0$ for the transformed 'leftmost' end cap to close an end of the wide line. With geometric transformation of graphic entities, a question implementers must answer is what shape is a dot or dash in a line invoking line style as an attribute.

If the *graphics system* is to provide inherent support for such attributes as width of a line or dot-dash style of a line as a geometric attribute, then reproduction of effects such as would be the case for stretching a rubber sheet or a 150% copier enlargement can guide our decision regarding

geometric attributes. The rubber sheet or magnifying glass model for functionality encourages a geometric interpretation of the attributes line width and line style.

By contrast, consider a different physical model to be the motivating reference for a computer graphics functional capability. Architectural and engineering drawings have a number of standard paper sizes. An 'A' size and an 'E' size drawing can be produced for the same picture content. Typically a draftsman will use the same mix of pen nib sizes in both drawings. Within any drawing, a draftsman likely will use several different pen nibs for line width to represent different meanings. For example, a map will use very thin lines for unimproved, secondary roads and will use much thicker lines for major superhighways or a dual carriageway.

A quill pen writing instrument can provide an analogy for cosmetic line width. Suppose a draftsman keeps a collection of ostrich, turkey, chicken, pigeon, sparrow, and hummingbird feathers. The choice of feather quill <pen nib> thus determines line width <weight>.

Patterns within an area may have symbolic meaning. Civil engineering applications need to represent concrete and other materials. Line patterns are one way of representing the material. If such a drawing is replicated on larger size paper, the lines representing the pattern for concrete should not have width enlarged. The effect to be reproduced by a computer graphics package is provision of a selection of line widths for cosmetic visual effect. In this situation line width and end cap shapes and dot-dashes should not be subject to transformation per se. The multiple pen nib size model for functionality ties a cosmetic interpretation to the attributes Line Width and Line Style.

From my development days as a graphics system implementer, I'll mention in passing that line style needs careful attention to consistency. Even in drawing a full circle, care must be taken to have the starting point on the circumference change so that rotated full circles in a pseudo-animation sequence have cosmetic line style properly rotated.

## 5. Clipping

Clipping is another task to which multiple interpretations can be applied. Many textbooks deal only with mathematical clipping, that is, traditional geometric intersection calculations. In traditional geometry, parallel lines never meet or share a common point. Non-parallel lines do intersect and share one and only one common point. Such is not the case in raster graphics. Parallel lines can share common pixels and non-parallel lines can intersect in more than one common pixel in a raster environment. The line segment from (1,1) to (6,4) is parallel to the line segment from (0,0) to (5,3) yet their rastered representations share three pixels in common. The line segment from (0,0) to (5,4) and the line segment from (0,0) to (5,3) intersect geometrically only at the origin yet their rastered representations share three pixels in common.

With raster graphics, it makes a noticeable, visible difference whether one analytically clips a line segment locus then rasters the shortened line segment or whether one just straightforwardly rasters the full line segment onto a much larger canvas then cuts out the smaller 'clip window' of interest <scissor clipping>. I consider a pre-rastering clip to be an algebraic or mathematical clip or, that is, a traditional geometric clip. I consider a post-raster clip to be a scissor clip.

Raster pixels, conceptually, are small, zero-area dots centered at the integer grid intersection points <lattice points> of a rectangular, Cartesian coordinate system in which both the vertical and horizontal grid line spacing between pixels is unity. A clip border will be a line segment specified by its beginning and ending pixel coordinates. Pixels falling exactly on a border line segment are, by definition here, within the picture.

**Analytical clipping** examines the mathematical locus of each graphic entity without regard to its attributes. Any portion of an infinitely thin line outside the active clip window is discarded. Only that portion of the idealized line segment, now with new, likely fractional, end points is retained. If any portion of the line segment does lie within the clip window, then attributes such as width and style are applied. When a line segment's mathematical locus lies wholly outside the clip window, it is discarded in its entirety and no portion of its widened representation is drawn even if the widened portion would fall inside the clip window.

**Cosmetic clipping** can be thought of as first rendering an entire picture on an infinite area canvas then subsequently using a pair of scissors to cut out the clipped window of interest. I think of scissor clipping as 'raster first then clip'. I think of analytic clipping as 'mathematically clip first' then raster.

Suppose a rectangular clip area is specified to have sides parallel to the axes. Let the area be, border inclusive, defined by the area surrounded by the four lines y=0, x=10, y=6, x=1. Seven-pixel wide lines having three pixels on each side of the respective reference line segments could be the line segment from (9,0) to (9,8) {the line x=9} and the line segment from (0,7) to (10,7) {the line y=7}. Analytic clipping would reject completely the line y=7 while the line from (9,0) to (9,8) would be revised to a shorter line segment from (9,0) to (9,6). Scissor clipping would retain both lines for conceptual rastering followed by a scissor clip.

Assume an identity matrix transformation. Analytic clipping would display for the line x=9 the 35 pixels inside the clip area. The line y=7 would contribute zero pixels.

Scissor clipping would display 50 pixels derived from each of the two original lines x=9 and y=7. While the reference line y=7 per se and its 'upper wide pixels' and 'leftmost pixels' would be discarded, y=7 would contribute 30 pixels bounded by the rectangle formed from lines y=4, x=10, y=6 and x=1. The line x=9 would need to contribute its 35 pixels within the clip area. The resulting picture is that which would come from rastering the full lines then using a pair of scissors to cut out the clip area from the post-rastered picture. With scissor clipping, a large picture could be generated in successive swathes. For any form of raster monitor or dot-matrix printer using a bit map image and having limited memory, a picture drawn as successive swathes should be identical to the same picture drawn from a single rastering of the full picture into a very large memory.

For future reference, I'll note here that I've consciously double counted in individual Scissor clipping line segments the pixels in a rectangle bounded, edge inclusive, by y=4, x=10, y=6 and x=6. In a replacement pixel drawing mode it won't affect the final picture seen on a display. In an ‘Exclusive OR’ drawing mode, what should happen to doubly drawn pixels?

Now let's consider some specific rastering situations which can arise. I'll contrive my example to offer integer intersections of lines and shall not cover additional considerations arising from non-integer intersections. Let the lower clip boundary be y=1 and the rightmost boundary be x=9. The leftmost clip boundary can be x=0 and the upper boundary can be y=3. For reference here, I'll claim a line segment from (0,0) to (27,3) will raster to:

5 pixels on scanline y=0 from x=00 to 04
9 pixels on scanline y=1 from x=05 to 13
9 pixels on scanline y=2 from x=14 to 22
5 pixels on scanline y=3 from x=23 to 27

The line segment from (0,0) to (27,3) intersects the boundary line y=1 at (9,1). It also intersects the boundary line x=9 at (9,1). Mathematical intersection leaves a single point. Even a single width, scissored line should draw the 5 pixels at (5,1) (6,1) (7,1) (8,1) and (9,1); they clearly are within the boundary inclusive clip area. With wide lines, some sense of slope would be needed to determine how to do widening for this analytic clip instance of a single point (9,1)..

To continue the example, suppose the rightmost clip boundary were x=8. The line segment from (0,0) to (27,3) mathematically has its theoretical locus wholly outside the clip area.. Scissor clipping would present the four pixels (5,1) (6,1) (7,1) (8,1).

Next let the clip boundaries be given by y=0, x=23, y=3 and x=4. A textbook clip would produce the line segment from (4,4/9) to (23,23/9). Rounding to raster a perturbed line segment from (4,0) to (23,3) is not particularly accurate.

Actually rastering a line with a repetitive algorithm is not necessary to determine starting or terminating run lengths. The pixels too can be simply a direct mathematical evaluation or a run length slice evaluation. Cognizance also must be taken of the boundary line and its crossing direction. In the above example, in which an 'integer' boundary line y=K is examined for intersection with an x-major line $y=y_0+((y_1-y_0)/(x_1-x_0))(x-x_0)$, one can use y=K-_ to calculate one extreme of the run along the scanline y=K and use y=K+_ to calculate the other end of the run along the lower boundary. Clipping for raster devices also will need to process consistently equal error measure instances such as described by Boothroyd and Hamilton.

## 6. Drawing Mode

Drawing modes <such as Replace, Or, And, Exclusi ve Or , ...> are not uncommon as a computer graphics attribute. The question for a user can be: OK, I know what the dichotomous logic operation AND is; what are the operands for this binary operation on pixels? Does drawing mode affect a graphics entity such as a line as a whole or is it a combinatorial specification merely for pixels or is it meant to affect only bits within an implementation-specific, position dependent, ordered bit sequence internal

representation of a pixel before color table expansion? What's correct depends upon one's reference model.

For simplicity, I shall restrict my brief discussion of drawing modes to a monochrome or bi-level raster display environment in which a bit plane is used to refresh a display device. Pixels can be ON or OFF.

Exclusive Or (EXOR) is a drawing mode which usually can be relied upon to illustrate different picture presentation depending upon one's choice of reference model. It also is a good test to verify quirks in an algorithm; incautious use of 8-way symmetry in an incremental circle generation is exposed when drawing in EXOR mode if octant end points are doubly dotted.

In a Polyline from (0,0) to (3,0) to (7,0), how many times is the pixel at (3,0) drawn? If it is both the last pixel in the first line and the first pixel of the second line, it disappears owing to being drawn twice. A convention such as the first line segment in a polyline owns both its first and last pixel while all other line segments do not own their first pixel would cause a single drawing of the pixel at (3,0). What would it do for contiguous lines used to outline a closed simple area? Would pixels still be 'double dotted'?

Joined line segments with a small angle between them often share many pixels in common. Raster the polyline from (0,0) to (10,0) to (0,1). What should be expected using an EXOR drawing mode? The convention described in the preceding paragraph could leave the pixel at (10,0) visible but rather lonesome as its immediate neighbors would have disappeared owing to double dotting.

## 7. Conclusion

At the rendering level and for the display processing pipeline it is key to define clearly what objective or viewable action is intended. Parts of a graphics system can not be specified in a vacuum. If a reference model is established first, then decisions regarding implementation details can be made more consistently.

## 8. Acknowledgements

Thanks are owed to my colleagues on the IBM 3270 PC/GX graphics workstation at the development laboratory in Hursley, England, especially to Nancy Bull, Mike Davis, Adrian Gay, Brian Middleton, Bryan Roberts, and Norm Sheen. I appreciate the support and encouragement from Steve Cunningham, Joaquim Madeira, and Mike McGrath for this GVE workshop. The specific numerical examples here were a part of an invited presentation at Ausgraph'90 International Conference on Computer Graphics; Michael Gigante did a great job with the conference.

## References

Arnold, D.B. and Bono, P.R. CGM and CGI: Metafile and interface standards for computer graphics. Springer-Verlag. 1988.

Boothroyd, J. and Hamilton, P.A. Exactly reversible plotter paths. Australian Computer Journal 2(No.1) p20-21. 1970.

Bresenham, J. E. Pixel processing fundamentals. IEEE Computer Graphics and Applications, 16(No15), p74-82. (January, 1996)

Bresenham, J. E. Computer graphics attributes and reference model alternatives. Proceedings: Ausgraph'90 International Conference on Computer Graphics. p413-424. (September, 1990)

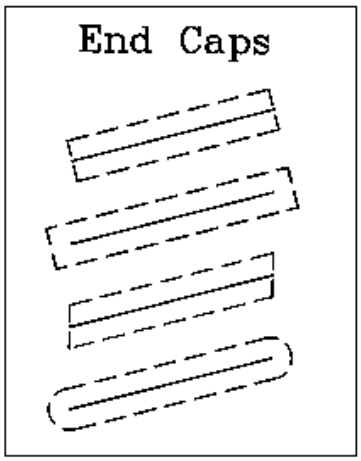
Bresenham, J. E. Ambiguities in incremental line rastering. IEEE Computer Graphics and Applications, 7(No.5), p31-43. (May , 1987)

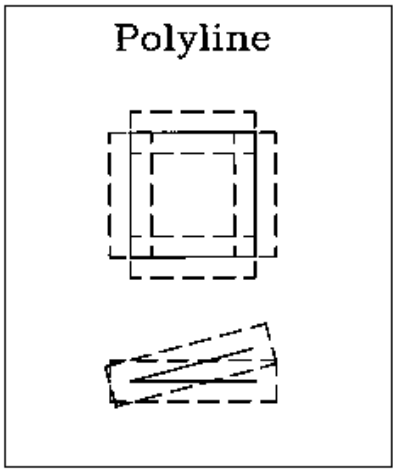
Gay, A. C. Experience in practical implementation of boundary-defined area fill. Fundamental algorithms for Computer Graphics, edited by R. A. Earnshaw. Springer-Verlag. p153-160. (1985)

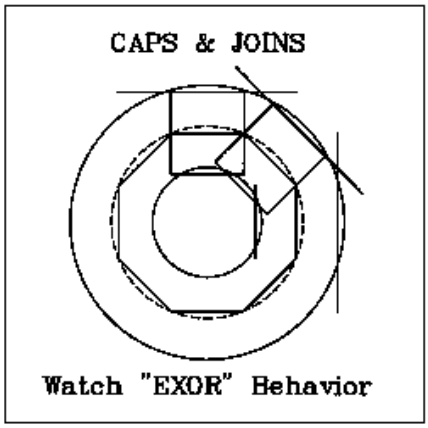
Henderson, L.R. and Mumford, A.M. The computer graphics metafile. Butterworths. 1990.

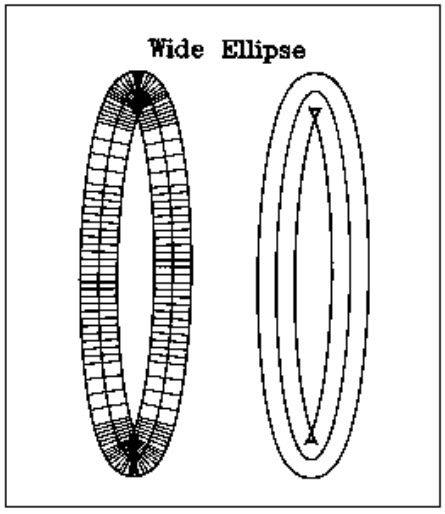
Posch, K. C. and Fellner, W. D. The circle-brush algorithm. Theoretical Foundations of Computer Graphics and CAD, edited by R. A. Earnshaw. Springer-Verlag. p857-878. (1988)

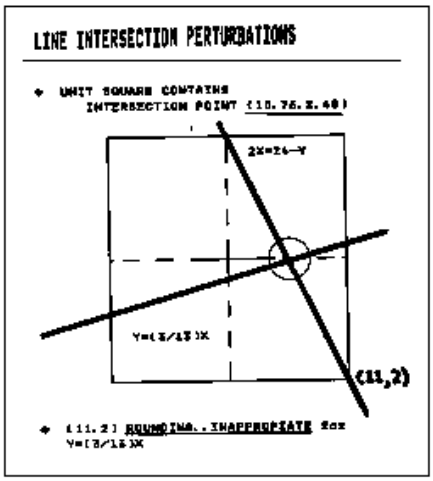
**Appendix**


End Caps


Polyline


CAPS & JOINS
Watch "EXOR" Behavior


Wide Ellipse


LINE INTERSECTION PERTURBATIONS
(11,2)

# Strengthening Visual Skills by Recognising Rendering Algorithms

**Rosalee Wolfe**
DePaul University
Chicago, Illinois
wolfe@cs.depaul.edu

**Abstract**

*Computer science students contemplating a career in graphics need to develop a visual sense, but traditional course topics do not meet this need. Visual analysis is a teaching technique developed for computer science instructors that helps impart this ability. Through the use of a few visual cues, students learn to visually identify surface algorithms, shaders and lighting techniques. An interactive software package called TERA (Tool for Exploring Rendering Algorithms) provides nearly a million image combinations that students can use to practice their visual identification skills.*
**Keywords:** *algorithm visualisation, computer graphics education.*

## 1. Background

For a successful career in computer graphics, computer science majors need to develop a visual sense in addition to their technical knowledge. Computer science graduates hired by the graphics industry will work side-by-side with artists, and it is essential for effective communication that they have an appreciation and enthusiasm for the visual aspects of the field. In addition to programming skills and knowledge of operating systems, recent labor market analyses list "strong visual style sense" and "understanding how artistic elements work with technical elements," as desirable traits for programmers and technical directors [1,2]. An increasing number of graphics houses want to see a demo reel or a portfolio from computer science graduates who count Unix shell scripting, C++ programming and the mathematical derivation of NURBS in their skill set.

There are many exemplary resources [3, 4] and learning tools that allow students to examine aspects of geometry [5, 6, 7, 8, 9, 10], rasterisation [6, 11], illumination [6, 12, 13, 14] and the entire 3D rendering pipeline [13]. These present visualisations that a student can use to explore and study an algorithm in isolation. This is a sound pedagogical approach for a first introduction, but eventually a student will need to understand an algorithm's behavior in co-operation with other algorithms and in the context of a finished image.

Unlike other specialities in computer science, the choice of a graphics algorithm usually cannot be based solely upon space, speed and implementation considerations. Equal in importance to time and memory requirements is the visual effect that a graphics algorithm produces. In fact, visual appearance will be the overriding factor for some applications. Visual understanding is essential to developing and debugging new algorithms. For example, when writing a shader, a programmer looks at a rendered sample, analyses how its appearance differs from the ideal and uses the information thus gained to refine the shader. An essential aspect to developing a visual sensibility is the ability to identify and compare rendering algorithms. Visual knowledge can prepare students to answer such real-life questions as, "Algorithm X gives us exactly the effect we want, but it is prohibitively expensive. Can we get a similar effect by tweaking our implementation of Algorithm Y?"

Although the graphics industry desires a developed visual sensibility in new computer science graduates, there is no place in the traditional curriculum where students can gain this knowledge. The discipline of computer science draws on mathematics and emphasises algorithm study and development, which are almost entirely text-based and involve little visualisation. Computer science instructors often view the prospect of discussing the visual aspects of computer graphics as a daunting, if not overwhelming endeavor. Typical reactions include

I'm not an artist! How am I supposed to gain the requisite background to teach this?

I don't have time for this in class – I have too many algorithms to cover as it is.

Both reactions are understandable and justifiable. Computer science instructors are not artists and their goal does not include producing artists. Further, there is an enormous amount of technical knowledge in the computer graphics discipline, and attempting to decide

what topics belong in an introductory course has been the topic of many papers. Any approach to the problem of inculcating a visual sensibility has to be one that does not take up much lecture time and constitutes a teaching method that computer science instructors are willing to try. This paper proposes that *visual analysis* is one approach that fits these constraints.

## 2. Visual Analysis

Visual analysis is a teaching technique developed specifically for computer science instructors that imparts this knowledge [15]. The technique stems from critical analysis, which establishes a structure for examining works of art [16]. Students in the visual arts learn to describe and compare works in terms of design, concept and media.

Instead of design, concept and media, visual analysis teaches students to recognise a small number of visual cues, including:

- visibility of polygon faces or polygon edges
- transition from light to dark on diffuse-reflecting surfaces
- color and shape of specular highlights
- presence of transparency, reflection, refraction, patterns and textures

| | |
|---|---|
| 1. First, determine the surface algorithm. | |
| *Appearance* | *Algorithm* |
| Outlined polygons. Object's far sides are visible. Uninterrupted horizon lines. | Wireframe |
| Outlined polygons. An object's far sides are not visible. Occluded objects are invisible. | Hidden-line removal |
| Filled-in polygons. Presence of refraction in transparent objects. (Presence of reflection, shadows are also helpful) | Ray tracing |
| Filled-in polygons. No refraction, reflection or shadows. | Z-buffer |
| 2. Determine the shader. | |
| For Z-buffer | |
| One color per object. Objects appear flat, as if cut from paper. | Constant |
| One color per polygon. Objects now appear to have a shaded contour. | Faceted |
| Smooth transitions from light to dark. Specular highlights follow polygon edges. | Gouraud |
| Smooth transitions from light to dark. Specular highlights are white, compact and elliptical. "Shiny plastic" look. | Phong |
| For ray tracing | |
| Opaque, colored surface. No highlight. | Diffuse |
| Opaque, colored surface with highlight. | Diffuse and Specular |
| Transparent object that appears to bend light. | Transparency |
| Part of scene is visible in the surface of the object. | Reflection |
| 3. Determine additional surface interest (both z-buffer and ray tracing) | |
| Image appears to have been pasted or glued onto object. | 2-D texture mapping |
| Object appears to have been carved from a solid substance like stone or wood. | 3-D texture mapping |
| Object surface appears rough or wrinkled, but its profile is smooth. | Bump mapping |
| 4. Light sources | |
| Harsh shadow / lighter shadow. | Low/high ambient |
| Sharply defined shadow. | Point light |
| Presence of a penumbra, but degree of shade is constant within penumbra. | Area light |
| Color bleeding; deepening of shadow in corners, under furniture. | Radiosity |

Table 1: Visual cues and algorithms (visual equivalencies omitted).

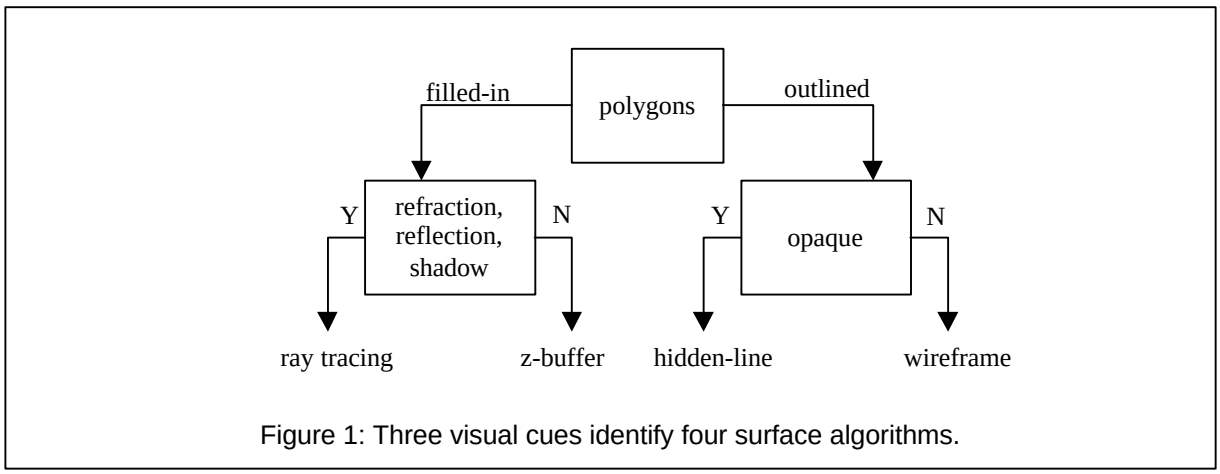



Figure 1: Three visual cues identify four surface algorithms.

- sharpness of shadow
- interactions between adjacent diffuse reflectors (color bleeding)

These cues are usually sufficient to identify a rendering algorithm as Table 1 demonstrates. Informal studies have shown that both instructors and students find this list of cues non-threatening and easy to spot in an image. By learning to observe and describe these cues, students are able to identify rendering algorithms.

## 3. Classroom Presentation

In the first lecture, the teacher discusses images portraying three or four algorithms that produce starkly different effects. For example, the four surface algorithms of wireframe, hidden-line, ray tracing and z-buffer create distinctive effects and can be distinguished by the use of only three visual cues. See Figure 1. Each week, the teacher adds more algorithms, and by the midterm, the class has examined all commonly used rendering algorithms. Until the midterm, the teacher emphasises the characteristic visual behavior of the algorithms. After the midterm, the teacher shows how multiple algorithms can achieve equivalent effects. By the end of the course, students learn to identify the surface algorithm, shaders and types of light sources.

While discussing a visual cue, the teacher presents two or three images that demonstrate it. After explaining the visual cue, the teacher gives the name of the algorithm that creates the cue. The teacher then shows a short series of images and invites students to identify the visual cues and then suggest a rendering algorithm. Students should specify the cues first so that they learn to spot them in the context of an image. Any premature guesses as to the identity of the rendering algorithm are met with the response, “Cues first!”.

After the first class meeting, discussions on visual analysis begin with a short series of images that review the rendering algorithms from the previous meeting. Students first name the visual cues and then suggest a rendering algorithm.

The methodology requires only five minutes of an hour-long lecture. Such a small amount of time will not significantly impact the presentation of other topics, especially if the teacher covers visual analysis during the last five minutes, when student focus is beginning to wander. In fact, there is a way of presenting this material without using any lecture time at all, as will be discussed in the next section.

## 4. Study Materials

Students need to practice visual analysis outside of the classroom. While posting images on the Web or pointing out examples in textbooks is a start, students say that they benefit more from the question-and-answer sessions held at the end of a lecture. Simply looking at images does not promote active learning.

Having students learn a rendering package helps a bit because students can choose parameters and view the resulting images. However it takes a significant amount of time to learn a package, and since the students have a priori knowledge of the surface and shading algorithms, this approach does not provide a student with a means of self assessment. A better approach is to provide students with an easy-to-use interactive tool that can demonstrate any rendering or shading algorithm while encouraging active learning.

TERA (Tool for Exploring Rendering Algorithms) is an interactive program that facilitates comparative study of the visual effects of rendering algorithms [17]. See Figure 2 for an annotated screen dump of TERA.

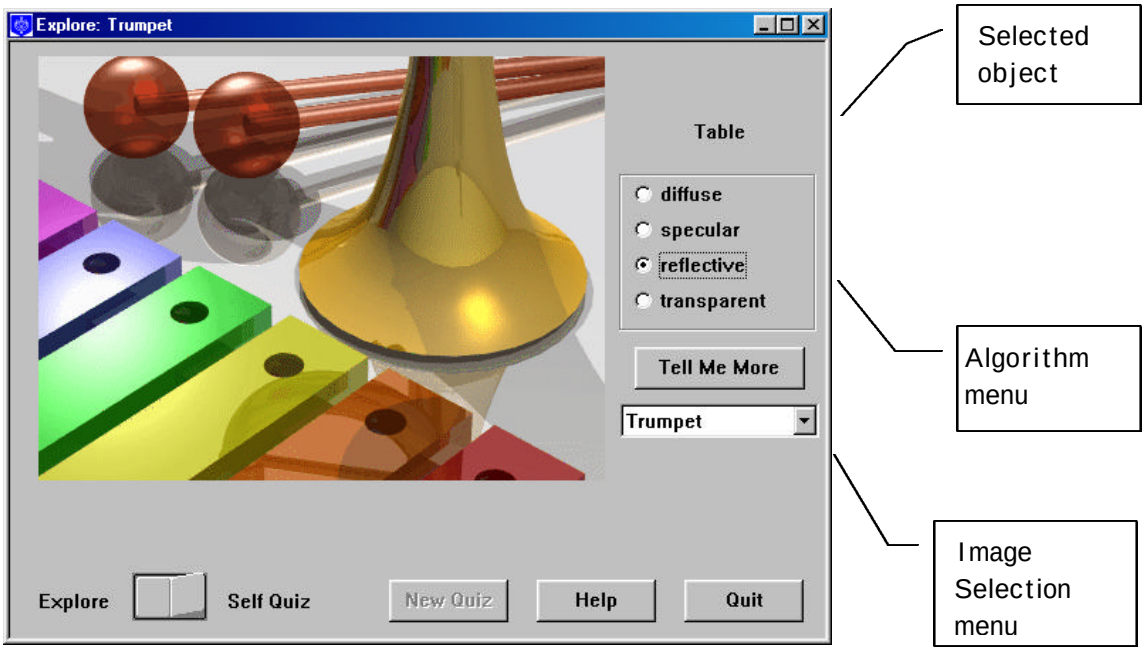



Figure 2: A sample TERA session.

A student can choose a scene and select a rendering algorithm for any object in the scene. Students can practice visual analysis using TERA. In "Self Quiz" mode, students select a scene, and TERA presents it with each object rendered by a random algorithm. Students then guess the rendering algorithm for each object in the scene. TERA responds with "Correct", "Try Again" or "Close Enough".

The "Close Enough" response is for those cases where multiple algorithms produce similar visual effects. For instance, Gouraud and Phong shading produce similar effects when no highlight is present.

Always available is the "Tell me more" button, which provides specific feedback about a student's last algorithm selection. When a student is in "Explore" mode, the "Tell Me More" button will activate a popup window describing the relevant visual cues. In "Self Quiz" mode, when students get a response of "Close Enough", they can click on "Tell Me More" for an explanation. A pop-up window will list the algorithm they picked, the actual algorithm and a specific explanation of why the two algorithms produced effects that are visually equivalent. For example, if a flat surface is very evenly lit, then Gouraud shading may produce variations in shade that are so subtle that the result looks like constant shading. In this situation, if a student picks "constant" and receives the "Close Enough" response, the "Tell Me More" button will activate a pop-up window containing the relevant explanation. See Figure 3.

The new version of TERA is capable of creating nearly a million images for students to analyse. The images cover surface algorithms, z-buffer shaders, ray tracing shaders, texture mapping, bump mapping, lighting and visual equivalencies.

Some teachers do not discuss visual analysis in the classroom, but give a short demonstration of TERA and tell students that TERA is available in the lab. TERA has enough appeal that students are drawn to it, and they spend enough time with it that they can score reasonably well on a visual identification test [18].

## 5. Results and Feedback

In my experience, visual analysis adds excitement and a sense of the "big picture" to introductory graphics courses. Students may not be able to implement every rendering algorithm when they leave the course, but they will be able to recognise them and know their names, which provides a starting point for further investigation.

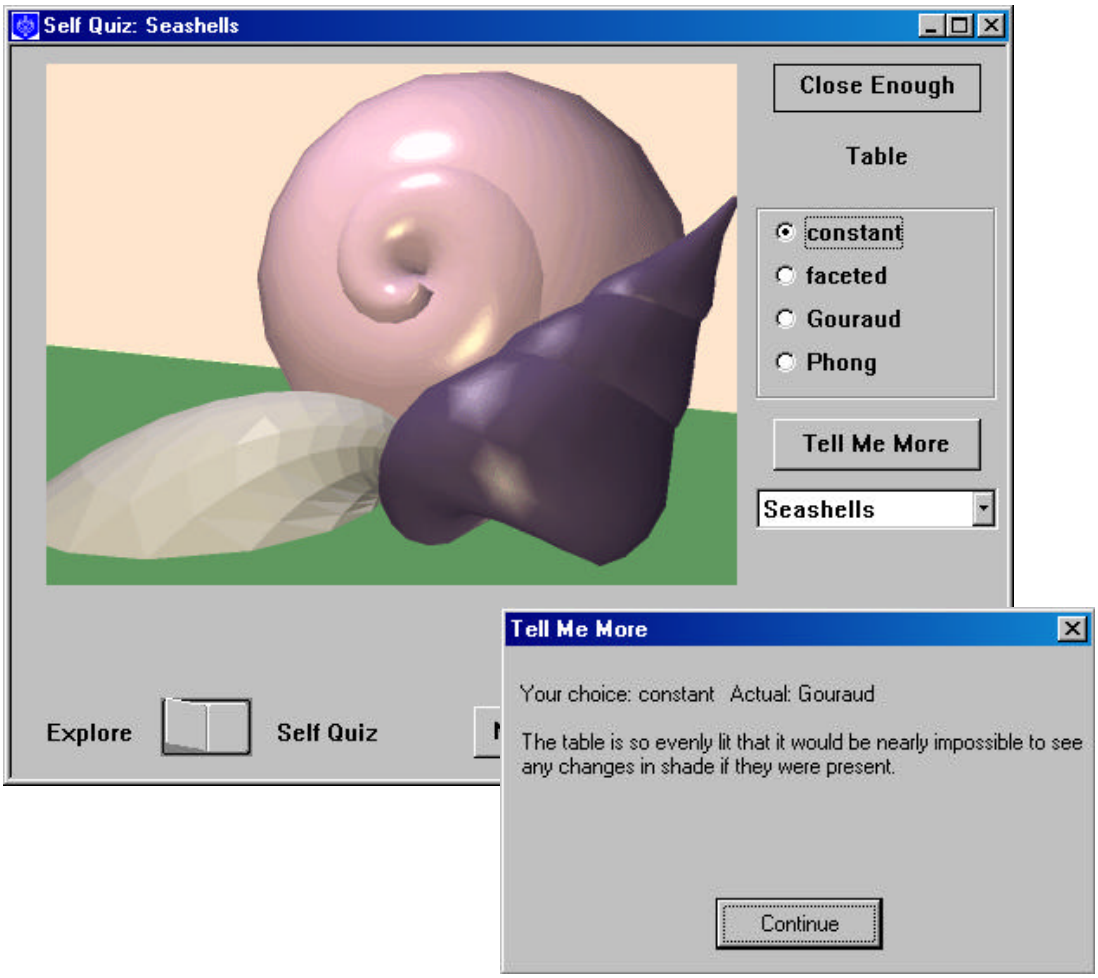



Figure 3: TERA in "Self Quiz" mode.

A visual sensibility, fostered by visual analysis, enhances a depth of knowledge of those rendering algorithms that students do implement. Because they are already familiar with the visual behavior of the algorithm, they have expectations of how an image should appear and can perform more of their debugging on their own without outside help. Students no longer ask, "Is it right?" They state, "There's something wrong with my highlight."

A very exciting development is the favorable feedback from digital artists. This approach promises to help art students "grasp the differences between light models, surface algorithms and shaders" [19]. Further work includes the exploration of developing the approach to expand the possibilities of better communication between digital artists and computer scientists.

## References


[1] Regan and Associates. A Labor Market Analysis for the interactive Digital Media Industry: *Opportunities in Multimedia*. Sunnyvale, CA: North Valley Private Industry Council (NOVA), 1997.

[2] Public Affairs Coalition of the Alliance of Motion Picture and Television Producers, *Making Digits Dance: Visual Effects and Animation Careers in the Entertainment Industry*. Sunnyvale, CA: NOVA Private Industry Council, 1997.

[3] Owen, G. Using Hypermedia to Teach Computer Graphics. Presented at the Graphics and Visualization Workshop in *Eurographics '93* Barcelona, September, 1993.

[4] G. S. Owen. Integrating World Wide Web

Technology into Courses in Computer Graphics and Scientific Visualization. *Computer Graphics* 29(3) pages 12-14, August 1995.

[5] D. Schweitzer. Designing Interactive Visualization Tools for the Graphics Classroom. In *Proceedings of the Twenty-third SIGCSE Technical Symposium on Computer Science Education.* pages 299-303, 1992.

[6] P. Min. www.cs.princeton.edu/~min/cs426/applets.html

[7] A. Shabo, M. Guzdial and J. Stasko. Addressing Student Problems in Learning Computer Graphics. *Computer Graphics* 29 (3) pages 38-40, August 1996.

[8] R. Klein, F. Hanisch, W. Strasser. Web-Based Teaching of Computer Graphics: Concepts and Realization of an Interactive Online Course. *Computer Graphics Annual conference Series, Conference Abstracts and Applications.* pages 88-93, 1998.

[9] Y. Zhao, J. Lowther and C. Shene. A Tool for Teaching Curve Design. In *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education.* pages 97-101, 1998.

[10] Y. Zhou, Y. Zhao, J. Lowther and C. Shene. Teaching Surface Design Made Easy. In *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education.* pages 222-226, 1999.

[11] A. C. Naiman. Interactive Teaching Modules for Computer Graphics. *Computer Graphics* 30 (3) pages 33-35, August 1996.

[12] Ansari, M. Simple Illumination Visualization. Department of Computer Science, DePaul University, Chicago, 1993.

[13] E. Wernert. A Unified Environment for Presenting, Developing and Analyzing Graphics Algorithms. *Computer Graphics* 31 (3) pages 26-28, August 1997.

[14] D. Gould, R. Simpson and A. van Dam. Granularity in the Design of Interactive Illustrations. In *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education.* pages 306-310, 1999.

[15] A. Sears and R. Wolfe, Visual Analysis: Adding breadth to a computer graphics course. In *Proceedings of the Twenty-sixth SIGCSE Technical Symposium of Computer Science Education.* pages 195-198, March 1995.

[16] R. Arnheim. *Art and Visual Perception: A Psychology of the Creative Eye.* 2nd ed. Berkeley: University of California Press, 1974.

[17] R. Wolfe, A. Sears. TERA: an interactive Tool for Exploring Rendering Algorithms. *Journal of Computing in Small Colleges* 10 (4) pages 41-46, February 1995.

[18] R. Wolfe, S. Grissom, T. Naps and A. Sears. A Tested Tool for Teaching Computer Graphics. *Journal of Computing in Small Colleges* 12 (2) pages 70-77, November 1996.

[19] D. Eber. Personal communication, December 1997.

# MAVERIK: A Virtual Reality System for Research and Teaching

**Toby Howard**, **Roger Hubbold** and **Alan Murta**
Advanced Interfaces Group, Department of Computer Science
University of Manchester, United Kingdom
toby@cs.man.ac.uk

**Abstract**

*This paper describes some experiences with the use of the MAVERIK system for supporting undergraduate and postgraduate teaching and research. MAVERIK is a high-level system for creating and managing interactive virtual environments. It is available free under the GNU General Public Licence. MAVERIK is modular and extensible, and its use of Mesa, the free OpenGL-like graphics system, means that it can be run on low-cost PCs, making it especially suitable for use in computer graphics and visualization education. We discuss the novel architecture and main features of MAVERIK, and illustrate its use by presenting case studies of projects undertaken by our students.*
**Keywords:** *Computer graphics education, MAVERIK, Mesa, Virtual Reality.*

## 1. An Introduction to MAVERIK

MAVERIK is a system for managing graphics and interaction in Virtual Reality applications [1, 2]. It is specifically designed to address the challenges of highly interactive virtual environments containing many objects with complex geometry. MAVERIK runs on PCs under the GNU/Linux operating system, using the free Mesa OpenGL-like graphics API [3], and can take advantage of 3D acceleration hardware if present. It also runs on Silicon Graphics workstations using Irix/OpenGL [4].

MAVERIK is free software, released under the GNU General Public Licence; the distribution includes all of the C source code, for both MAVERIK, and also a number of example applications (with data). This sets it apart from most commercial virtual reality systems, for which the source code is typically not available – or may be available, but at a cost prohibitive to educators. MAVERIK has a highly modular structure, and comprises a **micro-kernel** and a collection of **supporting modules**. The micro-kernel implements a set of core services, and a framework that applications can use to build complete virtual environments and virtual reality interfaces. The supporting modules contain default methods for optimised display management including spatial management, culling, interaction and navigation, and control of conventional and VR input and output devices. MAVERIK's structure allows these default methods to be customised to operate directly on application data, so that optimal representations and algorithms can be employed.

The MAVERIK micro-kernel uses a simple object-oriented **class structure**. Following normal OO philosophy, each class has an associated set of **methods**. For example, for geometric primitives, default methods are provided for displaying them, and for selecting them with a mouse. To keep the implementation simple, methods are implemented as callback functions. Changing a method is as simple as writing a new (or modified) callback function and registering it with the MAVERIK kernel. The kernel also provides a mechanism for defining new classes, making the system extensible. This approach was a deliberate design decision – we wanted the system to be easy to understand and install. In a student context, we have found that using C and callbacks has meant that those with basic programming skills can quickly become proficient at using the system.

A key difference between MAVERIK and many other VR systems is that MAVERIK does not use its own data structure for storing application data. Of course, the kernel does have data structures for managing classes and their methods, but as far as possible we have tried to avoid imposing data structures on applications. Instead, classes of object types can be created which suit the needs of particular applications. Methods for displaying and interacting with these objects are then defined and registered with the kernel. This means that only a **single representation** of the application data needs to be maintained. Any changes to this data are automatically reflected as soon as the next frame is displayed, because MAVERIK uses immediate mode rendering. The default supporting modules take much of the work out of writing the display and interaction methods.

MAVERIK is distributed with nineteen default object classes, and standard methods for navigation and for picking objects using a standard mouse and keyboard. In our research laboratory we also employ 3D mice based on Polhemus magnetic trackers, and code for handling these is included in the distribution. We also use head-mounted displays and large-screen projection systems and MAVERIK includes the code needed to compute stereo projections, and to synchronise frame buffer updates. Another part of the distribution is a set of tutorial examples, which are described in the documentation (in PostScript, HTML, and on-line man page formats), and some more advanced demonstrations which illustrate the use of MAVERIK for some of our own research applications.

## 2. Using MAVERIK for Teaching

There are several reasons we believe MAVERIK is useful for teaching computer graphics and visualization:

1. MAVERIK is free. Students can run it on their own PCs, allowing them to study in their own time as well as on campus. Similarly, other educators have free access to the system. MAVERIK runs on relatively low-cost PC hardware, relying only on other software which is also available free.

2. MAVERIK is a fully featured, professional-level system, not a small subset developed specifically for teaching. We use it in our own research to explore very large, complex virtual environments. For example, polygonal representations of some of our industrial CAD models would amount to more than 100 million polygons. Examples can be found on our Web pages [5].

3. MAVERIK implements a range of algorithms – such as culling, using a hierarchy of bounding volumes or occlusion techniques, object tessellation, computing projections, and stereoscopic viewing – which students can study to see how they work. The availability of source code means they can learn by example and by experimenting for themselves by changing default methods.

4. MAVERIK is distributed with default methods which handle the most common cases – such as primitives like boxes, cones, spheres, cylinders, tori, polygon meshes. These serve as examples of how to build other types of objects: NURBS perhaps, or even Jell-O! For example, as we describe in one of the case studies, it proved straightforward to add a new class of object to implement superquadric shapes.

## 3. Case Studies

We use MAVERIK extensively in our own teaching –for final year six-month undergraduate projects, and for postgraduate work – as well as our own research. In this section we present short accounts of a selection of recent and on-going student projects which illustrate the diverse range of applications of MAVERIK in an educational context.

### 3.1. Builder – Direct 3D Interactive Construction

Builder is a 3D object manipulation program (see Figure 1). It was designed to explore techniques for selecting and manipulating hierarchically structured objects in 3D space, using two-handed manipulation. The manipulation is carried out using a pair of 3D mice, and the resulting images are shown on a stereo projection screen, a head-mounted display, or a normal monitor. Builder uses voice recognition to issue commands, because both hands use the 3D mice.

Builder is a very dynamic program. Much as in a 2D drawing program, objects can be selected and stretched or squashed, using handles attached to key points on their bounding boxes. Objects can be created, deleted, copied, pulled apart, joined together, and positioned in space. Each operation may be reinforced using sound cues, via a MIDI interface. It is possible to enter a virtual environment and to completely rearrange it using these techniques.

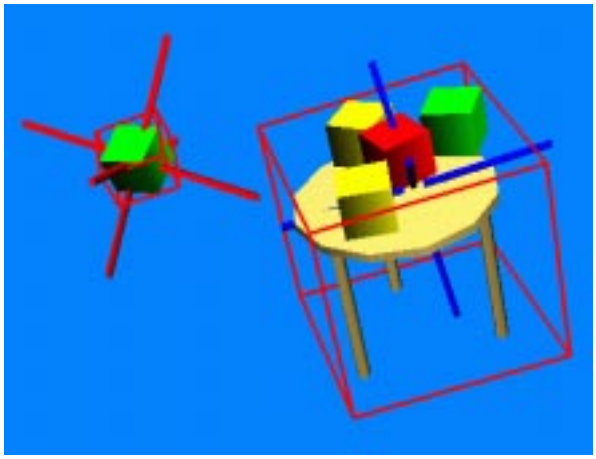


Figure 1. Builder: an interactive 3D modeller.

Buttons on the mice differentiate between moving around the environment and picking up objects. The two mice can be used interchangeably, so that it is possible to pick up an object in one hand and to carry the object around, using the other mouse to control navigation.

Builder relies on a key feature of MAVERIK – that it uses only the application's data and a set of callback functions to update the display. There is no separate scene graph or other data structure that needs to be synchronised with changes to the application's data. It also uses MAVERIK's culling methods to permit quite large environments to be displayed at fast frame rates, which is essential when using an HMD.

### 3.2. Virtual Lego

Another approach to interactive model building is illustrated in the Virtual Lego project (see Figure 2). Here we consider the joining and breaking apart of models consisting of a set of bricks. It differs from Builder in that constraints are applied to only allow the joining of shapes in a small number of known arrangements, and the system applies continuous object clash detection during the assembly process. Interaction is achieved using a combination of 2D mouse and keyboard.

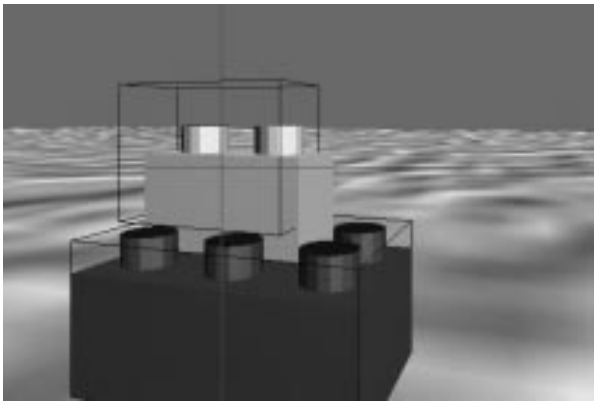


Figure 2. Constrained Lego brick manipulation.

### 3.3. Professor Dijkstra goes Walkabout

One of the demonstration programs distributed with MAVERIK is a virtual city ("cityscape"), with buildings, roads, trees, pavements, grassy areas, monuments, and animated avatars. This project explored extensions to the cityscape program to include a route planning mechanism. The initial method used Dijkstra's algorithm for computing the minimum cost path between any two places. To provide a more efficient implementation, a second version was implemented based on the Floyd-Walshall algorithm. Both of these algorithms are used to find the minimum cost for traversing a graph structure, and are standard algorithms taught to Computer Science students. The project used the Xforms interface builder [6] to provide a GUI. A speech recognition system can also be used to instruct the program. The project is quite interesting from two angles.

First, it is an interactive city guide. Using the interface, you can ask Professor Dijkstra – represented by an avatar – to guide you around the city (see Figure 3). For example, you may ask to be taken to the railway station, or the public library. The program builds a graph structure for all of the connected streets, and you are guided along the pavements, traversing roads at junctions or pedestrian crossings, using the shortest route. The result is displayed as an animated walk-through. At any time, you can stop or move away from your guide, and he will recognise this and wait for you.

Second, it is a nice illustration of the Dijkstra and Floyd-Walshall algorithms in practice. While the program is running, obstacles can be introduced, interactively, to block streets – to simulate road works, or perhaps an incident like a fire, for example – and the algorithms then search for the best way around the obstacles via alternative streets.

Figure 3. Professor Dijkstra goes walkabout.

This program exploits several features of MAVERIK. First, it uses occlusion culling algorithms to achieve good frame rates, even for large cities (e.g. 40 blocks square). Second, the navigation methods can be employed to wander around the city at will. Third, MAVERIK's default avatar class is used to display representations of

actions of people in the city. The avatars are animated, and walk around the city in real-time.

### 3.4. Head-Driven Navigation

Specifying scene navigation within interactive 3D environments is often based on 2D mouse movements and key-presses, or uses 3D mouse or dataglove gestures. This project considered the use of head motion (tracked using a 3D Polhemus sensor) to trigger user movement within a virtual environment. This may be useful when both hands are already involved in a manipulation task, for example. Navigation behaviours were devised which map head motion onto view rotations and translations. Alternative mappings were tried for both head-mounted displays and fixed large-screen projection screens. MAVERIK proved to be a good platform for investigations of this kind, since the user is free to customise navigation functions for specific applications. User tests were performed to compare the usability of the head-driven approach with more traditional hand-driven navigation schemes. These included a timed fly-through of a twisting 3D tube, and the navigation of a slalom course.

### 3.5. Out-of-Body Experiences

Navigation in virtual environments is a difficult task. One of the main hindrances is the limited field of view afforded by head-mounted displays and projection systems, as this is typically much narrower than we experience in the real world. It has been described as like looking at the world through a pair of toilet rolls! This makes life difficult, because it means that one tends to blunder into objects that are just out of view.

An important aspect of virtual reality is the notion of presence – that is, of being present in a virtual environment and of being surrounded by it. Thus, when navigating around, one is normally presented with a view corresponding directly to that of one's virtual body – an "in-body" or first person view. However, an alternative way to navigate is to follow a few paces behind one's virtual body – a kind of "out-of-body experience". In this case, one still controls one's body, but sees it as a separate representation, or avatar. This type of interface can be found in some PC games, such as Tomb Raider [7].

The purpose of this project was to investigate whether navigation is more easily controlled using an in-body or an out-of-body paradigm. One reason why the latter might be easier is that it affords a much wider field of view – you can see obstacles surrounding your virtual body that would not normally be within your field of view using the in-body approach. The project entailed devising some simple virtual environments containing obstacles and then monitoring the performance of volunteers in negotiating their way along prescribed routes. The code was instrumented to measure results, such as the number of collisions with objects, the length of the path followed, and the time taken to complete a route.

### 3.6. Visualization of Nanotechnology

This project investigated how computer graphics and Virtual Reality techniques might be used to visualize and control the construction and activity of nanomachines. The project involved the simulation of molecular interactions, and the modelling of collections of molecules organised into mechanical structures (see Figure 4).

This was an ambitious project, and MAVERIK proved especially useful – its provision of high-level virtual environment management enabled the student to concentrate on the application problem, rather than the infrastructure needed to manage the virtual environment.

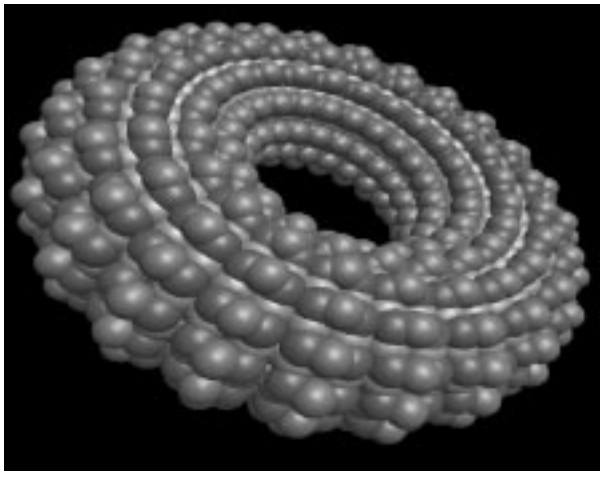


Figure 4. Visualizing nanotechnology.

### 3.7. A Clutterizer for Virtual Environments

A virtual environment is typically constructed by creating accurate geometrical architecture, and populating the scene with specific objects (and textures) at specific locations. Such environments are usually quite tidy. Real environments, however, are rarely tidy, having accumulated clutter over time.

The purpose of this project was to investigate methods for automatically "clutterizing" a virtual environment, by semi-automatically adding "junk" geometry and texture.

MAVERIK provides default methods for testing for collision detection between objects in the virtual environment, and this feature facilitated the development of algorithms for rapidly populating the environment with large numbers of "clutter" objects, distributed statistically.

### 3.8. Interactive City Generation

This undergraduate project investigated methods for automatically generating realistic cities (see Figure 5). The project implemented a suite of three separate programs; the first generated a plausible layout of streets within the city; the second then populated the streets with buildings, monuments, parks and other urban features. The final program was an interactive viewer, which used MAVERIK for visualization, and navigation within the scene.

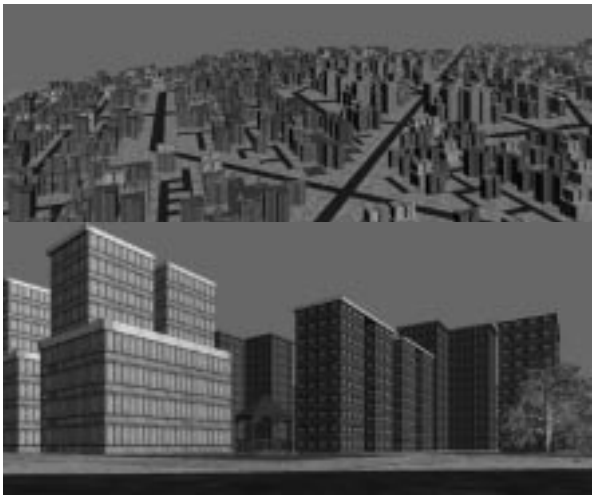


Figure 5. Interactive city generation.

### 3.9. Line of Sight and Trajectory Analysis for Scene of Crime Reconstruction

Working in collaboration with Greater Manchester Police, we have constructed a prototype virtual environment corresponding to a real scene of crime [8]. Subsequently, this undergraduate project undertook an investigation of two areas: first, methods for testing "line of sight" scenarios, to accurately determine which parts of a scene are visible from certain vantage points; second, for interactively tracing and modelling bullet trajectories within the scene.

MAVERIK provided the virtual environment infrastructure, and its default methods for computing vector-object intersections assisted the modelling of bullet trajectories, and multiple ricochets from rigid surfaces. The project implemented two kinds of trajectory simulation: a simple Newtonian model, and a "point mass" model which takes into account aerodynamic drag on a bullet. Figure 6 shows a simulated bullet trajectory and ricochet.

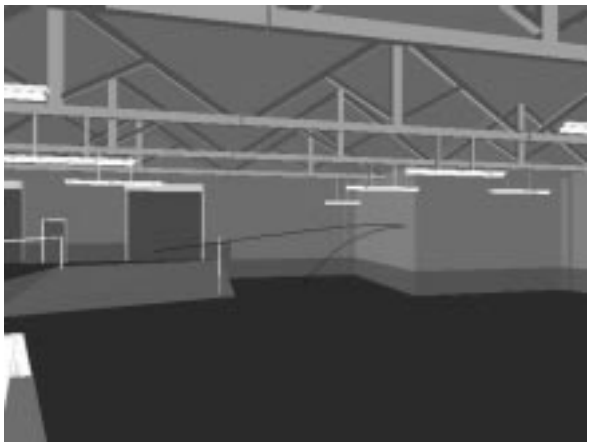


Figure 6. Visualization of bullet trajectories.

### 3.10. Simulating Undersea Viewing Conditions

We are interested in using VR as a means of rehearsing assembly and maintenance procedures on large industrial structures, such as offshore gas and oil platforms. This project investigated the graphical portrayal of the kind of environments encountered during such undersea operations (see Figure 7). In particular, the incorporation of visual phenomena such as animated caustics, lighting effects within participating media, limited visibility (fogging) and the presence of particulate matter in the environment were considered. This project used MAVERIK for navigation, interaction and scene management, with additional use of Mesa/OpenGL to achieve specific graphical effects. Real time undersea effects (over 20 frames per second) were achieved on a modest PC with a 3D graphics card.

Figure 7. Shallow undersea view featuring animated lighting effects.

### 3.11. Interactive Modelling Using Superquadrics

Superquadrics are a family of solids derived from the parametric forms of the basic quadric surfaces – ellipsoids, tori and hyperboloids. Superquadrics are, however, much more flexible than standard quadrics and can represent a huge variety of shapes. This postgraduate project is investigating the use of interactively deformed superquadrics for object modelling in a virtual environment (see Figure 8).

This involved adding a new, quite complex, object class to MAVERIK. In practice this proved to be straightforward: the application defines a data structure for the object, and a minimal number of methods – which it registers with MAVERIK – to operate upon it.

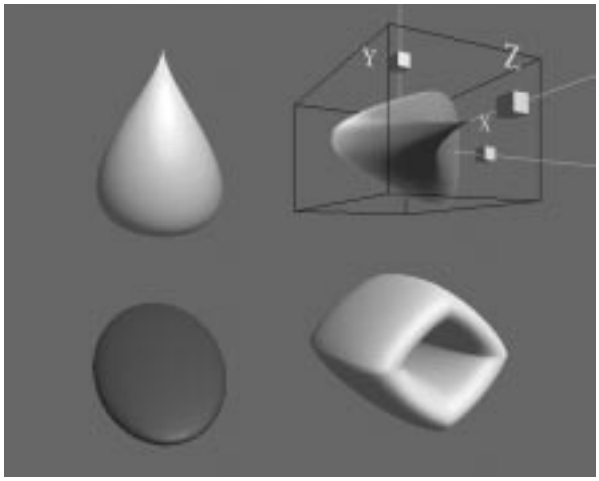



Figure 8. Interactive deformation of superquadrics.

## 4. Conclusions

We have been using MAVERIK as the basis for our teaching and project work since 1996, and have been extremely encouraged by its success. Because the source code is freely available, students have the opportunity to look inside and see exactly how a large professional-level software system is structured, and they can examine the algorithms in detail. And because MAVERIK uses Mesa as its rendering engine, computer graphics students can go further, and see exactly how the rendering pipeline is implemented. As for project work, because MAVERIK manages the complexity of the virtual environment, students are able to pursue more ambitious projects than have previously been realistic.

It is our hope that sharing our positive experiences of teaching using MAVERIK will be of benefit to other educators. We would be delighted if others would consider evaluating MAVERIK as a suitable tool for computer graphics and visualization education.

## Acknowledgements

We would like to thank our colleagues in the Advanced Interfaces Group: Jon Cook, Simon Gibson, Martin Keates, Steve Pettifer, Adrian West – all of whom have contributed substantially to the MAVERIK system. And we'd especially like to thank our students for their enthusiasm, and for implementing a series of fun and interesting projects for us to report: Osian Ap Garth, Tim Davis, Sinem Guven, Chris Kirk, James Pearce, Jamie Pratt, Ahmed Rahali, Chris Redburn, James Sinnott, John Taylor and Irina Titovich.

MAVERIK was developed as part of the VRLSA project, funded by the Engineering and Physical Sciences Research Council (GR/K99701), with additional support from our industrial partners, CADCentre Ltd, Brown & Root Ltd, and Sharp Laboratories of Europe Ltd.

MAVERIK is available in tar and RPM formats from http://aig.cs.man.ac.uk/systems/Maverik/.

## References

[1] Roger Hubbold, Dongbo Xiao and Simon Gibson. MAVERIK: The Manchester Virtual Environment Interface Kernel. In *Virtual Environments and Scientific Visualization 1996*, M. Gobel, J. David, P. Slavik, J.J. van

Wijk (Editors).
[2] Jon Cook, Roger Hubbold and Martin Keates. Virtual Reality for Large-Scale Industrial Applications. In *Future Generation Computer Systems*, **14**, pp. 157–166, 1998.
[3] http://www.mesa3d.org.
[4] http://www.opengl.org.
[5] http://aig.cs.man.ac.uk/systems/Maverik/.
[6] http://bragg.phys.uwm.edu/xforms/.
[7] http://www.tombraider.com.
[8] Alan Murta, Simon Gibson, Toby Howard, Roger Hubbold and Adrian West. Modelling and Rendering for Scene of Crime Reconstruction: A Case Study. In *Proceedings Eurographics UK*, pp. 169–173, 1998.

# Computer Graphics Curricula in the Visual Arts

**Dena Elisabeth Eber**
Bowling Green State University
Bowling Green, Ohio, USA
deber@christo.bgsu.edu

**Abstract**

*Computers have recently emerged as commonplace on the scene in a record number of university art departments. This rapid change has created the need for new curriculum in the computer art discipline. In designing these courses I see three ideas that are paramount to any computer graphics curriculum in the visual arts. First, the courses and course work should focus on individual expression and creativity, second, the instruction should include an awareness of the "wow" factor, and third, the courses should include a balance of computer art, traditional art, computer science, and academics such as theory, art history, and criticism.*
**Keywords:** *computer graphics curriculum, visual arts, higher education*

## 1. Introduction

Against many odds and tight budgets, computers have recently emerged as commonplace on the scene in a record number of traditional college and university art departments. As part of this rapid change, artists are not only using computers as an extension of many traditional art practices, but they are also using them as a medium to create unique digital art forms. From imaging, digital painting, and 3D modeling, to interactivity, video, and 3D animation, graphics technologies give artists the power to use the computer as an expressive art form, thus forging new educational and aesthetic concerns in curricula for computer graphics educators. This paper will address a subset of these concerns in a general way with the intention of highlighting areas that individual programs and researchers can further expand.

### 1.1. Scope of Paper

This paper will articulate some curricular issues related to teaching computer graphics in the visual arts at the higher education level. The 1990 *Guidelines for Curricula in Computer Graphics in the Visual Arts*, also know as *The Dream Curriculum*, addresses similar issues, however some have changed and others have emerged since its publication [7]. I will touch on a small number of the emergent and dated topics. Other areas of art such as music and theater may also be affected by the changes in computer graphics, but these domains are not covered in this paper. Also of great importance but outside the scope of this paper is computer graphics in kindergarten through twelfth grade (K-12). This document is concerned with curricula for students at the college level who have chosen computer graphics as a field of study and as a career in the visual arts.

Curriculum for computer graphics is quite broad with applications ranging from graphics and textile design to studio art. This paper is modeled after a studio art approach, however, it also encompasses the commercial arts such as graphics design, interior design, textile design, architecture, interior, industrial and packaging design. Both the studio artist and the commercial artist focus on expression through visual media, however, the key difference is the application of that focus. In graphics design and commercial arts, the focus is on a client or a customer and the goal is to use personal creativity to visually express, design, or illustrate a product or idea. In studio art, the focus is most often on the artist's personal expression, creativity, or idea through visual media.

The suggestions in this paper do not go into enough detail to differentiate the nuances between these two general approaches, especially as the line between them becomes more and more indistinguishable. However, it is my intention to highlight that I am writing from a studio art frame of reference and to leave the application of my suggestions up to the reader.

### 1.2. What is Art?

> *Art is a human activity consisting in this, that one man [or woman] consciously by means of certain external signs, hands on to others feelings he [or she] has lived through, and that others are infected by these feelings and also experience them.*
>
> Tolstoy, from *What is Art?*, 1897

A work of art, according to Tolstoy [8], is sincere, and it transmits information, feelings, and experiences through lines, colors, sound, or words. Actualization of the elements embedded in the imagery start with the creator, the creative process, and the ability to see and to know through that seeing. The work may take varied forms, but to be art, the object, idea, or presentation goes beyond the physical and contains some aspect of human experience. Art may be created with any tool, as long as the artist can use that tool to put some of their true self into the art.

A common approach to teaching studio art is to stress expression of the individual, a focus that goes far beyond the tool or medium. Artists have always used technology and many have learned to express themselves in a way that emphasizes the message, not the medium. There are many other dissenting views of media, some of which are based on Marshall McLuhan's idea of electronic media, that the medium is the message [6]. Bits that underlie the varied digital representations are all the same, thus setting expressive boundaries upon artists [4].

The balancing of the message and the medium in digital technologies is similar to a problem that artists have always had when working with new media of the time, however, digital technologies add a new twist. Computer hardware and software technologies are being replaced at the rate of every eighteen months, if not faster. Rising beyond the medium takes time, but an ever-changing paintbrush that is highly technological hinders the time artists have with it. New tools require them to re-learn a skill set, which in turn, forces them to constantly shift between artistic expression and skill acquisition, two different ways of thinking. Teaching computer graphics in an art setting demands finding a delicate equilibrium between artistic expression and technological proficiency, something the curriculum must address.

## 2. A General Approach

### 2.1. Philosophical underpinnings

No matter the focus of a digital art program, be it a specialization within a traditional university art department, an extension of an established studio area, a commercial major, or other related areas, the curriculum must balance digital technology with other concerns. I see three ideas that are paramount to any computer graphics curriculum in the visual arts:

1) The courses and course work should focus on individual expression and creativity,
2) The instruction should include an awareness of the "wow" factor
3) The courses should include a balance of computer art, traditional art, computer science, and academics such as theory, art history, and criticism.

For students to make expressive art, they must pay attention to the message it portrays. This means that instructors need to make a pedagogical shift from technique oriented curricular goals to exercises that limit the tool set and leave room for experimentation, play, and theme based works. The tool set should be expanded only after students address content and show a level of comfort with the medium. In this way the digital techniques will be one important concern, among others. Further, the students then have time to figure out what they want to express, thus shaping the medium rather than the medium shaping them. The critiques and assignments should focus around an idea, which stimulates self reflection and critical thinking. The premise of this approach is if an instructor stimulates critical reflection and dialogue within and between the students, then they will have a better understanding of how to shape digital media to express their ideas. This will in turn dramatically improve the outcome of their work [3].

Related to individual expression is an awareness of the "wow" factor, a condition that is especially abundant when working with digital media. This characteristic takes many forms, but in most cases it is a desire on the part of the student to create works solely for the application of a cliché or often overused special effect. Not only does this kind of work look canned or have an "off-the shelf" aesthetic, but it lacks personal expression and creativity and shifts the focus away from critical reflection and back to surface level technique. It is for this reason that instructors must point out the "wow" factor and de-emphasize it.

Beyond cliché and content, it is my view that the curriculum should include art theory, criticism, and history at least within the studio courses, but most preferably in addition to them. Not only does this help artists identify their work with other art in computer graphics and traditional media, but it helps them critique it and understand how their ideas can expand on those of others. Schools such as Keel University stress a 50/50 split between studio and theory, which provides the student with discourse and data to situate their work in the digital and art world [2].

Keel University also requires the computer graphics student to work in another studio area such as photography, print, painting, or sculpture. This allows students to see different approaches to expression that computer graphics can extend. It is my view that this approach also helps the student understand and contextualize their work with digital media, a new and multifaceted medium.

Not only should the curriculum address computer art, other studio art areas, and theory, but it should also introduce the student to programming. *The Dream Curriculum* [7] points out that teaching artists how to program is a controversial notion. At the time it was written, artists who wanted to use three dimensional (3D) imagery and other image effects in their work had to write many of their own programs. Since then, powerful image processing and 3D modeling packages have been written and the artist no longer has to, but often still does, program.

Also since *The Dream Curriculum*, use of the world wide web (WWW) has emerged and many other kinds of scriptable interactive packages have become available. There will be times when most artists will need to write scripts, however what kind, in what language and for what purpose will prove to be dynamic. Underlying any specific programming application is the basic understanding of programming logic, data structures, functions, input and output, and variables. At minimum, artists should understand these ideas in whatever flavor is available, such as C or visual basic. This should be a class most preferably in computer science because the cross disciplinary setting gives both artists and computer scientists each a taste of the other discipline and how people in that field think. This setting is increasingly important as industry heads towards interdisciplinary art and science teams.

Some artists may choose to go beyond the basic programming level and make it a larger part of how they create art. Some artists and educators believe that programming is one way to break the "pre-set" or "off-the-shelf" aesthetic that makes some work appear unoriginal [4]. It is my view that this approach will work for some, but not all. Programming requires a technical, spatial, and linear way of thinking, which some artists employ in addition to non-linear and emergent approaches. However, there are many art students who do not think in this way and have talents or intelligences in many other styles such as personal or bodily-kinesthetic forms [1]. These methods and more can be integral to making successful works of digital art. For these students, intuitive software provides the tools they need to make rich and expressive works of art. There is a lot that focused and creative students can do with a pre-programmed digital paint brush.

I believe that the learning objectives for students in computer graphics and the visual arts should include not only technical proficiency, but also an understanding of the significance of their work and how they can manipulate the medium to express their message. They can accomplish this through a curriculum balanced in complimentary areas as well through instructors that stress ideas as well as technique.

### 2.2. Foundations Core

A common approach to university level art programs is to begin with what is called a foundations core, or a set of classes that address basic art principles. Many programs break this up into the two dimensional (2D) and three dimensional (3D) areas. 2D course work often includes drawing, composition, color theory, figure drawing, and painting, while 3D course work focuses on different forms of sculpture. Some programs are also incorporating sections that cover the concept of time, performance, criticism, theory in contemporary art, interactivity, and installation art. The computer in foundations is threefold, it can be used as an educational tool to teach concepts such as color theory, it can be taught as an extension of other media such as printmaking or painting, or it can be addressed as a medium in and of itself.

It is my view that computer graphics as an art medium should in some way be part of the foundations core. It is important that students using all media understand the role that computer graphics plays in their art making, be it directly or indirectly. In other words, it is meaningful for all artists to have an understanding of digital media as a basic core requirement, much like drawing is treated as a core for all art students, which brings up another point of contention.

Converse to the argument that non-computer art students should be exposed to digital media is the argument that computer art students should take drawing. Since *The Dream Curriculum* was written, computer art has become a commonplace major or specialization. As a result of this change, may contend that not all the standard core requirements apply to computer art majors. It is my view that drawing and other basic core requirements are absolutely essential for computer graphics majors.

Recall I stated that art is about seeing and core requirements, if nothing else, help students learn to see.

Others say that drawing helps those in computer graphics, a discipline that stresses vertical thinking, to solve problems in a lateral way [5]. Vertical thinking is characterized by a stepwise logical process, while lateral thinking is a non-linear process. A vertical approach to finding a solution might be to search one entire plane of solutions when in fact the solution is on another search plane. The lateral approach will throw the search into a new space at which time vertical thinking may take over. Most people can think in both ways and according to some, drawing is one way to exercise lateral thinking in computer graphics, which is otherwise a vertical discipline.

In designing the core curriculum, I think one very important and often overlooked notion, especially for computer art majors, is that of time, or 4D. 2D or 3D computer animation, digital video, and interactive works of art such as WWW pieces or virtual environments (VE) all require an understanding of how ideas unfold, change, and take shape over time. Whether it is dealt with in a separate course or addressed in a section of a universal art core class, the concept of 4D should be included because it is basic to many computer graphic art forms.

### 2.3. Enter Computer Graphics

Whether computer graphics is part of the core curriculum or not, computer art majors must begin work in their discipline at some time. A point of contention is when to start computer art majors in computer art classes. In some programs, computer art majors begin work with computer graphics at the upper-class level, only after the core classes are complete. In a four year program, this means that they do not start with computer graphics until the end of their sophomore or the beginning of their junior year.

I believe that computer graphics majors in the arts must start on the computer as soon as they can, preferably in their first year. The first course should be a sampling of the computer art focus areas offered at the institution. From this course, the ones that follow should be a higher competency level in a particular focus area. See table 1 in section 3 for some of these areas.

As I have already mentioned, art is a form of expression and it is up to the artist to learn how to shape the medium to do so. This takes time, much more than two final years of college can provide. Further, the students need to be aware of the capabilities that the computer graphics medium provides while they form their artistic ideas.

## 3. Beyond the Core: Focus Areas in Computer Art

*The Dream Curriculum* mentions that art educators need to decide if computer graphics will be a support area for other media or a major unto itself. Since 1990, it is more common to find a computer art major or specialization in an art program than a computer lab which supports other disciplines. It is for this reason that the following suggestions are intended for computer art majors. I did not indicate competency levels or what courses should come in what order following the core classes. That, I believe, should be tailored to individual programs and resources. However, I feel students should have some understanding about *the still* and *time* before they can fully grasp the potential of interactivity.

The purpose of the following list is to highlight areas or directions that programs can follow in computer graphics. I chose to break this expansive discipline into four major areas; the still, time based works, interactive works, and computer graphics with other media. This same list could be viewed in many other ways, such as 2D, 3D, and time. The four suggested areas are not mutually exclusive, as many crossover in inextricable ways. Table 1 is a list of the four areas and some topics that fall within each.

| The Still | Time | Interactive | Cross Computer Art |
|---|---|---|---|
| 2D Imaging/Digital Paint | 2D animation | Web based art | Digital Photography |
| 3D modeling | 3D animation | Interactive CDs | Printmaking & Computer |
| Installation or Sculpture | Digital video | Interactive installations | Fibers & Computer |
| Algorithmic | Digital Performance | Virtual Environments | Drawing/Painting & Computer |

Table 1. Some suggested areas of focus in computer graphics for the visual arts.

### 3.1. The Still

The still is a form of computer art that does not include time and can range from a print of a digitally manipulated photographic image to an installation or a sculpture. The still can be wholly generated on the screen, such as a digital painting or a 3D model, or can include scanned or acquired imagery or algorithms. Especially at issue with the still is "objecthood," or the still as a physical material, such as a print rather than an image on a screen. "Objecthood" is not a concern within the scope of this paper, however it is one that should be addressed in the course work.

The still is often thought of as a 2D or 3D image, however I have included installation and sculptural works because they deal with a single instance of time. An understanding of 4D is so crucial to some areas in computer art that I used it as an organizing factor that surmounts the difference between imagery and sculpture. A digital installation could be time based, especially if it includes digital video, but the sculpture I refer to in the still category is one that deals with a single instance in time. This includes anything from the latest resin based stereolithographic sculpture to still images that have been realized in some sculptural form such as digital images wrapped around a cube.

### 3.2. Time

It is one thing to understand how the composition of a single image works formally and expressively, but it is quite another issue to understand how that aesthetic changes over time. What is important for expression when the viewer cannot contemplate each frame? How do ideas change as the still image moves? How does time factor into the overall success of the art? These questions and more are ones that time based course work must address.

Courses in this area include 2D and 3D animation, digital video, and performance using computer graphics technologies. Students who focus in this area should take support courses in theater, film, and drama.

### 3.3. Interactive Works

Since *The Dream Curriculum* the notion of interactivity has grown profoundly more complex, especially in light of developments such as the WWW and multi-user installations. These courses include web based art, interactive CDs, interactive installations, and VEs.

Students who work with interactivity use both time and imagery as well as human response. This course work should include an understanding of human computer interface design as well as concepts such as the aesthetic experience for viewers as they unfold layers of the work.

### 3.4. Computer Graphics and other Media

The ways in which computer art crosses into other art areas is abundant and numbers far beyond the four that I have listed. Computer graphics as an extension of other media is helping artists approach their studio area in a new way by enlarging the scope of artistic possibilities. For example a painter can scan a slide of her painting, manipulate it, reproduce it as an Iris print, then transfer it to wet plaster to create a sculpture with a fresco aesthetic. This process surpasses the act of painting and perhaps defies a label, but it is a unique process made possible by computer graphics. It is this type of thing that would fall within the Computer graphics and other media focus area.

In addition to computer art, students in this area should have an equally strong focus in at least one other studio area.

## 4. Conclusion

In a time when our culture is becoming much more visually literate, those who create visual media must pay more attention to clear, meaningful, and sophisticated messages in their work. Visual media for the sake of effect stops at the surface of the work, and the audience is lost after the initial "wow." This is much like movies that sport fascinating special effects but fall short of a meaningful story line. On the other hand, movies that weave monumental visual effects into purposeful narratives are the works of art that will be significant in years to come. This means that the curriculum that educates some of the artists who will make these films must address technical proficiency along with creativity and meaningful expression.

To this end I have highlighted a few concerns in core curriculum, listed some computer art focus areas, and suggested key issues to address in the general approach to designing curriculum for computer graphics in the visual arts. These are only suggestions and individual programs and goals should be the ultimate guiding force.

My hope with this document is to continue the dialogue about curriculum so computer graphics

educators in the visual arts can address it in a meaningful and vital way.

## References


[1] H. Gardner. *Frames of Mind: The Theory of Multiple Intelligences*. Basic Books, Inc., New York, 1983.

[2] J. Harris. Art Education and Cyber-Idealogy: Beyond Individualism and Technological Determinism. In *Art Journal*, pages 39-45, Fall, 1997.

[3] T. A. Jackson. Ontological Shifts in Studio Art Education: Emergent pedagogical Models. In *Art Journal*, pages 69-73, Spring 1999.

[4] P. Klein. Structuring Digital Media in a Fine Arts Institution. Paper presented to the annual meeting of the College Art Association, February, 1999.

[5] M. Kurlansky. Educating Designers in the Digital Age. In *Digital Creativity*. Lisse, the Netherlands, Swets and Zeitlinger, pages 218-222, Vol. 9, No. 4, 1998.

[6] M. Mcluhan. Myth and Mass Media. In M. Moos *Media Research: Technology, Art, Communication*. OPO, Amsterdam, pages 5-15, 1997.

[7] B. Mones-Hattal, K. O'Connell and D. Sokolove. Guidelines for Curricula in Computer Graphics in the Visual Arts. In *Computer Graphics*, pages 78-76, June 1990.

[8] L. Tolstoy. Extracts from *What is art?* In E. Simmons, *Leo Tolstoy, VII*. Vintage Books, New York, page 239, 1960. (Reprinted from *What is art?*, (Aylmer Maude Trans.). by L. Tolstoy, Oxford University Press, Oxford, 1898)

# An Interdisciplinary Approach to Teaching Computer Animation to Artists and Computer Scientists

**David S. Ebert** and **Dan Bailey**
University of Maryland, Baltimore County
{ebert, bailey}@umbc.edu

**Abstract**

*Animation has always required a close collaboration between artists and scientists, poets and engineers. Current trends in computer animation have made successful and effective teamwork a necessity. To address these issues, we have developed an interdisciplinary computer animation course for artists and scientists that focuses on contemporary issues in computer animation. This course emphasizes collaborative teams for practical experience, cross-mixing of student expertise, and group-based education: the students learn from each other, as well as the instructors. Student teams produce a professional animation that extends the capabilities of a commercial animation package to gain experience in and exposure to the state-of-the-art research in computer animation and rendering, the complete animation process, and the artistic and aesthetics of computer animation. We describe our approach to teaching this course, the structure of the course, the results, and lessons learned from our experience with the first offering of this course (Spring 1999).*

**Keywords:** *interdisciplinary, computer animation, education, collaborative education.*

## 1. Introduction

Computer animation has always required a close collaboration between artists and computer scientists. We have developed an interdisciplinary computer animation course that focuses on contemporary issues in computer animation and requires the skills of animators and programmers working in teams. The goals of this course are the following:

- Increase the technical graphics and animation knowledge of computer science students.
- Increase the animation skills and knowledge of advanced computer animation techniques of art students.
- Introduce art students to the technical aspects of rendering and animation, and expose them to research issues in computer animation.
- Introduce computer science students to traditional and computer animation techniques.
- Introduce art students to the creative potential of writing procedural shaders, models, and animation expressions.
- Provide practical animation production experience, using and extending commercial animation software.
- Provide a collaborative learning environment where students will learn from each other, as well as the course instructors.

A key aspect of this course is that students gain experience in participating in interdisciplinary teams. Teams of 4 - 5 visual arts and computer science students work together to produce animations that utilize each member's skills and interests, in a manner similar to commercial animation environments. The computer animation industry requires employees to work in teams on large projects. Traditional educational environments do not teach skills to make students successful in such environments. Computer animation environments pose a further challenge: teams are composed of members from quite disjoint backgrounds. We have structured this course to help students learn how to communicate, work, and even thrive in this environment.

## 2. Background and Motivation

Animation's history, from its origins in the 1880s to contemporary time, is a continuous line of technological inventions that have allowed animators the ability to achieve higher quality effects with greater ease: Bray/Hurd patented cels, Fleischer Brothers invented and patented rotoscoping, Disney developed the multi-plane camera, and the list goes on [1]. At the heart of each of these developments has been the successful synthesis of artistic and scientific talents. At times, these skills have come together in the unique individual. Normally, however, they have been the result of creative collaborations, especially in today's highly technical computer animation arena.

Equally as important, large scale animation production has always required large teams of variously talented individuals. The Warner Brothers animators of the 1930s through 1950s arguably produced some of the most successful cartoons of this century and historians note that effective collaboration between its directors, animators, writers, technicians, artists, and musicians was one of the prime reasons for its success [1].

Both of these issues point to the fact that a successful and contemporary animation curriculum should not only be interdisciplinary, but also encourage students to develop effective team skills.

Schools, universities and institutions are usually divided into departments to better serve students needs. Crossing the boundaries between areas and departments has always been difficult. A particular school may not be reluctant to develop interdisciplinary courses, but usually mechanisms and incentives to do so are not in place. Thus, many schools are slow to address the industry trend to teach and encourage effective teamwork and collaboration between animators and computer scientists. Currently many animation, special effects, and computer graphics houses are creating their own in-house workshops and programs for addressing these issues.

## 3. Our Pedagogical Approach

Our pedagogical approach to this course has two key themes: **interdisciplinary work** and **collaborative education**. This is true even in the instruction and design of the course, which is team taught by a visual arts faculty member and a computer science faculty member. Most of our lectures are designed to have sections that both faculty members present, highlighting the technical computer graphics aspects and the art/animation aspects of the material. In every aspect of the course, we encourage students to collaborate and help each other. Initial assignments, described below, are designed to have segments that are easy and segments that are difficult for students from each background, thus encouraging students to begin interacting with their counterparts.

As mentioned above, we are presenting material in the course that extends the backgrounds of both students, while making them familiar with (and even extending) the state-of-the-art in computer animation. Computer science students lead the discussion of computer animation research papers, helping the art students understand the new material. Conversely, art students present profiles and critiques of computer animators and animation techniques that expand the computer science students' appreciation of computer animation as art.

## 4. Structure and Implementation

We have structured this course to take advantage of a classroom equipped with SGI workstations for interactive instruction and demonstrations, and the open architecture and procedural flexibility of the Maya software package from Alias/Wavefront[1]. The course is 15 weeks in length and the students start working in teams (4-5 students) during the second week of the class. The students may propose the composition of their team, with the restriction that each team must consist of at least two art students and two computer science students. The students have an icebreaker team project to perform in Weeks 3-5 of the class: create a 10-second animation to be composited with the blue-screen filmed sequence of a student performing for 10 seconds. After this initial icebreaker project, we allow the teams to be revised to accommodate any poor group dynamics that were discovered. We also have some initial assignments that expose artists to working with vectors, angles, and simple illumination, expose computer scientists to key-frame animation, and both to the procedural, extensible aspects of the Maya modeling, animation, and rendering package.

To expose students to the power of procedural shading techniques [2] and the challenge of photorealistic image generation, the students' second major project is to generate a photorealistic image/animation with the following specification: it must contain a specific type of light and a specific object element each chosen randomly from a hat. Some example lighting types used in Spring 1999, included sunset light, light from a lava lamp, light from a neon sign, and light from a candle flame. Some example object elements included a cup of cappuccino with froth, icy or snowy patches on a sidewalk, and fur. Students were given three weeks to complete this assignment.

---

[1] This software was made available through a generous grant from Alias/Wavefront, Inc.

The main component of the course is a 12-week team project to produce an interesting, professional animation that includes extending the Maya package in a new way. Some proposed extensions include the following:

- Particle dynamics for cloud formation
- Volume rendered implicit primitives for accurate cloud rendering
- Volumetric light sources to model fire
- Volumetric simulations of fire, taking into account simple physical simulations
- Texturing of implicit surfaces with deformations
- Extending Maya's implicit animation capabilities
- Hydrologic simulations of ocean waves
- Underwater simulation of light rays and dispersion
- Simulations of rich, organic, evolving environments
- Flocking simulations for crowd animation

Regardless of the technical and graphic effects that are achieved, a successful animation is only as good as its story, premise, or content. Students must also consider the subject matter of the project. Suggested formats and structures for the animation include the following:

- Narrative (comedy, drama, adventure. etc.)
- Character driven
- Animating directly to sound or music
- Environmental or architectural (a precursor to a VR or interactive space)
- Poetic or text driven piece (using a quotation, journal entry, voice over, or written text)
- Commentary (political, technological, social, theoretical)
- Adaptation of existing works (homage, contrast, contradiction, satire)

Early in the semester each team is required to "pitch" their animation to the instructors and the entire class in a professional presentation with storyboards, charts, and slides. This project also requires a presentation of progress to the class after five weeks of work and a final presentation of results to the class. These are graded both on the presentation, and the artistic and technical merit of the work.

## 5. Initial Results

The initial offering of the course was very successful and educational for the students and instructors. The students were very enthused about the class and gained valuable experience in computer animation, as well as working in teams. Most of the teams worked successfully together and, to our surprise, each team decided to name itself. The team dynamics varied widely among the 5 teams and also changed greatly from the beginning of the semester to the end. One of the teams that worked the best during the semester started out with very poor group dynamics between the artists and computer scientists. By the end of the semester, they learned how to communicate effectively, to appreciate each other's skills, and work as a team. Only one team (20%) suffered significantly from group dynamics, including one team member dropping the class. Spending more class time on effective team techniques and team building may help eliminate these problems.

### 5.1. Blue-screen Project

Our icebreaker team project required each team to learn how to composite live-action video (10 seconds of a student using exercise equipment) with computer generated effects for a 10 second animation. They were required to generate both CG foreground and background elements. An example still of the blue-screen video is shown in Figure 1. Two example stills from this project are shown in Figures 2 and 3.

After viewing and critiquing the students' initial animations, we gave them an additional week to improve them. This produced a better product, but also affected the amount of time available for the rest of the semester work. Students spent 5 weeks completing this project. After viewing the final animations, the importance of teaching anti-aliasing techniques became apparent.

Figure 2: A still from the blue-screen animation by the Screaming Nixons (Eun Baek, Jon Feibelman, Costas Kleopa, Vlad Korolev, Steve Matuszek).

Figure 3: A still from the blue-screen animation by CSmART Allstars (Ava Collins, Alex Eller, Jason Lubawski, Marlin Rowley, Christian Valiente).

**5.2 Artistic Shader Assignment**

As mentioned above, the goal of this assignment was to create a photorealistic image or animation that contained a specific type of lighting and a specific element. After each team chose a type of light and a scene element from a "hat", the following became the scene specification for each team:

| | | |
|---|---|---|
| Team1 | Kiwi and peach in a bowl | Candle light |
| Team2 | Fur | Mirrored disco ball and strobe light |
| Team3 | Hot tub with steam and bubbles | Neon light with neon tube showing |
| Team4 | Ice patches on a sidewalk | Outdoor sunset or sunrise lighting |
| Team5 | Cappuccino with froth and a cinnamon swirl | Lava lamp light |

Figures 4 and 5 show two example stills from this project. Figure 4 contains an image showing snowy patches on a sidewalk at sunrise and Figure 5 contains an image of a cup of cappuccino illuminated by a lava lamp.

Figure 4: Snowy sidewalk at sunrise by the Screaming Nixons (Eun Baek, Jon Feibelman, Costas Kleopa, Vlad Korolev, Steve Matuszek).

Figure 5: "Cappuccino by lava lamp light," by CsmArt Allstars (Ava Collins, Alex Eller, Jason Lubawski, Marlin Rowley, Christian Valiente).

**5.3 Semester Animation Project**

The semester animation projects contained a wide variety of technical and artistic styles. Below are summaries of the projects for each team:

- *The Autonomous Chicken Farm* - This team explored the practical application of behavioral animation by developing procedurally animated "Creatures" which are controlled by a customized user interface in Maya. The example "Creatures" chosen were chickens with controllable personality attributes of hunger, beauty, and aggression (based on the 7 deadly sins). The final product of this project was an extension of Maya for creating autonomous "smart" creatures and

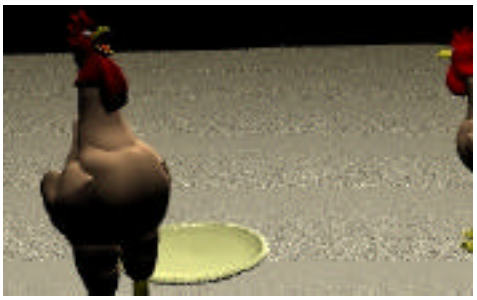
controlling their movement. This generated input for the inverse-kinematic controls and expressions that the animators used for controlling the motion of the creatures. A short demonstration of the Maya extension was also produced.

- *Gelegant Jammin'* - Animated short in which character movements (blendshapes, joints, and lattices) of a drop of gel were driven by sound file information (wav file). A graphical user interface was created to control which dance movements were driven by particular sound data.
- *Infinity* - This animation was a symbolic projection of life and astral time represented through the juxtaposition of an angel contemplating notions of time, history and evolution, with an icon of modern technology. Plugins for the rendering and modeling of volumetric clouds were developed to set the scene for this animation.
- *Martyr's Playtime Theatre Presents: Joan d'Arc* - The story of Joan of Arc was presented using models resembling the children's toys, the Fisher Price[tm] People. Procedural effects were created for fire, melting, and crowd animation.
- *A Buck Ninety-Nine and Someone Else's Wall* - This project explored the ambiguity of environment by moving between cave formation space and the urban realm of the subway station. The main character was a spelunker who wears a headlight to illuminate the scenes. Initially, he seems to be located alone in a cave somewhere, but as he explores the environment he finds himself to really be in the tunnels of the subway system. In order to create the cave space, a MEL (Maya Embedded Language) script was created for creating turbulent cave walls with varying fractal parameters.

Some example resulting images from these projects can be seen in Figures 6, 7 and 8. Most teams completed a significant portion of their project, but didn't complete it. We worked with the teams to complete their animations after the semester so that they had produced a finished animation from the class that they could be satisfied with and include in a demo reel. Teams had difficulties both in terms of the technical programming aspects, as well as animation and production.

Figure 6: A still from ***Infinity*** by the CSmART Allstars (Ava Collins, Alex Eller, Jason Lubawski, Marlin Rowley, Christian Valiente) showing an angel in a volumetric procedural clouds.

Figure 7: A still from ***Martyr's Playtime Theatre Presents: Joan D'Arc*** by the Screaming Nixons (Eun Baek, Jon Feibelman, Costas Kleopa, Vlad Korolev, Steve Matuszek).



## 6. Conclusions

We have developed a successful interdisciplinary course to teach computer animation to computer scientists and artists based on interdisciplinary collaborative work. This approach for education is very powerful and rewarding; however, it does require a significant amount of effort in teaching

not only computer animation, but successful team work techniques and group dynamics.

Our first offering of this course was very successful and we have gained valuable experience to help us improve the course in the future. For the next offering of the course, we will decrease the amount of time students have to complete their first icebreaker assignment so that they can concentrate their efforts on their semester projects. We also will allow for more individual credit in the team projects. Most of a student's grade in Spring 1999 was based on the grade their team received. We made a few exception in extreme cases. We plan on still having most of the work in the class be done with collaborative teams, but will be adjusting the grading scheme to also reflect each individual's contribution. The homepage for the Spring 1999 edition of this class can be found at http://research.umbc.edu/~bailey/courses/CS_ART_Anim/

From our experience, we believe that an interdisciplinary computer animation course provides both types of students with an increased appreciation of, and an improved ability to communicate with, the other community. Students also gain valuable experience in working in collaborative teams and an increased sense of the history and state-of-the-art of computer animation.

We would like to thank the student teams who helped us develop and improve this course:

- The A-Team (Tracy Corder, Will Gee, Mike Keesey, Joe Romano)
- CSmART Allstars (Ava Collins, Alex Eller, Jason Lubawski, Marlin Rowley. Christian Valiente)
- Beasts (Kim Harrington, Mike Madison, Chris Morris, Brian Resurreccion, Shawn Wood)
- Screaming Nixons (Eun Baek, Jon Feibelman, Costas Kleopa, Vlad Korolev, Steve Matuszek)
- Wookie Pimps (Michelle Hunt, Steve Jacobs, Chris Slingluff, Joy Saunders)

## References

[1] Maltin, L., *Of Mice and Men*, McGraw-Hill, 1980.

[2] Ebert, D., Musgrave, F., Peachey, D., Perlin, P., Worley, S., *Texturing and Modeling: A Procedural Approach, second edition*, AP Professional, 1998.

# Combining Art and Mathematics in Computer Graphics Education

**Mark Ollila**
University of Gävle
Gävle, Sweden
molly@hig.se

**Abstract**

*This paper presents an innovative education program at the University of Gävle, Sweden. The program, Creative Programming, combines traditional methods of teaching computer graphics, with other cross-disciplinary areas such as art, cognition, and film. The uniqueness of the program is represented by the students that are accepted and the teaching environment they are put in. A judicial process is used to screen applicants, with and without portfolios to the program. This is followed by an examination of intellectual capacity, practical examination, and finally an interview. Together, these are used as guidelines to determine if the student is a suitable candidate for the program. The final group selected has varying backgrounds, ranging from the professional artist to the traditional computer scientist. They are then immersed in a lab environment for the whole duration of the education on the campus. The remainder of the education involves a period in industry. The combination of selective students, a lab environment has produced very satisfactory result, and can be used as basis for further developing the education.*
**Keywords:** *Computer Graphics Education, Art, Industry*

## Introduction

Universities are under pressure to develop new, dynamic education curricula. The area of digital media, for instance, is becoming increasingly important in this world. There is a shortage of people with the required educational background who can design systems in this new medium, in research terms, as well as in visual development and implementation. The interfaces to computers are designed in a much more visual and intuitive manner today, so the skills involved in creating software applications have expanded from traditional programming to include creative skills, such as graphic design, writing interactive narratives and creating story boards. Research also plays a very important role in the design of systems and implementation of new techniques yet to be utilized in current technologies. Universities with traditional education of graduates in computer science must now examine the need for these new multidisciplinary skills. One common solution is the creation of cross-disciplinary degrees such as Computer Science/Media Studies. If we examine the skill set required in developing these new media applications we find that they are essentially different depending on the desired end result. Several applications will require a programming/logical approach as a solution, whereas others will require a creative, visual solution to the problem. If we examine the various types of graduates from a typical University, we will find different skill sets appropriate to this new medium. An arts graduate will have developed skills in written communication, an artist will have an appreciation of aesthetics, psychology graduates appreciate the difficulties of human-computer interaction, and computer science graduates will have an understanding of the technical issues required and how current technologies might not be adequate for the problem at hand.

The Creative Programming program was created in 1995 with the goal of combining the previously mentioned skill sets together in an innovative way. The initial program is an eighteen month program which can be used in conjunction with computer science credits (or similar) to obtain a Bachelors degree. The program looks at both the technical and aesthetic issues involved in the digital media of today. Education is in the form of lectures and practical sessions with skilled instructors from industry or academia. This makes the students up to date with what is happening in industry, as well as understanding what the latest developments in research are. The program helps students to combine existing skills with the technological skills needed in this area, enhancing their knowledge, both technically and analytically.

## Screening Process

The process of screening applicants is very complex. The work is examined by a jury of experts in the area of digital media, both from academia and from industry. A previous Creative Programming student is also included in the jury to provide a students perspective. The jury reduces all the initial submissions to thirty possible candidates that are then invited for an interview at the University. The criteria used for selection include the portfolio, academic qualifications,

and industry experience. When judging the portfolio, issues of creativity, originality and quality are most important. It is often the case that the portfolios examined are varied in nature, presenting very different kinds of work, in either creativity and originality, but with not necessarily high technical art skill.

The interview generally involves an examination of the applicants intellectual ability, creative ability, skill at handling a computer, and evidence of a creative mind as well as of sociability. One element of the interview stage is the use of Raven's Advanced Progressive Matrices (APM) [1] to indicate the analytical and logical ability of the applicant. The APM is designed to assess a person's intellectual and reasoning ability. It is one of the most well examined tests of general intelligence, it is nonverbal, and has very good psychometric properties, normated against lots of populations with different cultural and educational backgrounds. The advanced version of the test (APM) used here is designed for people of above average intellectual capability. The test-retest reliability is over to 0.93 for age 30 and under. The test is used for evaluating the logical intellectual capacity of those very artistic applicants that are lacking traditional computer science. The social and emotional competence is also judged in the interview, to make sure that an open and dynamic group process is promoted by the students accepted. The creation of this group dynamics is also the rationale for selecting a good mix of gender, age and artistic and computational profiles into the group. With this the most crucial conditions for merging the artist with the programmer are met with.

The program is unique in its approach to education and selection of students. It aims to give an opportunity to highly creative and intelligent people to experiment and enhance themselves in the world of digital media by combining their skills from other areas, such as art or programming.

## Curriculum

After the selection process, fifteen students are invited to participate in the Creative Programming program. The first twelve months of the program the students are in a lab environment at the University. The final six months the students are in industry. The program is taught in competition to the Computer Science program and the Computer Engineering Program at the University. The structure of the curriculum is taught in four terms over two semesters, with the final industry project in the third semester. The first semester concentrates on aesthetics of the image, new media technologies and human computer interaction. The subjects taught in that semester include -

*Multimedia Techniques* – introduction to the theoretical concepts of multimedia. The practical sessions involve developing multimedia applications, which are both technically and aesthetically challenging. This is taught by Computer Science staff.

*Aesthetics* – introduction to the important concepts in aesthetics. This looks at both traditional and current theories in aesthetics. This is taught by an expert in art from a department at the University. There is also some interaction from industry art experts.

*Multimedia Management* – this subject examines how to create complex multimedia applications. This is combined with dealing with real clients and deadlines. The clients are usually from industry or the local community. The lectures involve seminars about topics such as contract writing, copyright issues, management of a team and client handling. These seminars are given by experts from the digital media industry.

*Program Design and Cognition* – examines the human computer interaction (HCI) issues when designing software. There is a strong cognitive psychology background presented here. The course is taught by lecturers in the Computer Science department.

The second semester of the course concentrates on computer graphics, animation, film aesthetics and future research. The subjects include -

*Computer Graphics, Modelling and Animation* – this course is actually two courses in one. It examines the theoretical aspects of computer graphics in detail. Even though the majority of students do not have the traditional linear algebra background, the course still delivers complex mathematical concepts. The contents include, but are not limited to, 2D and 3D transformations, lighting, realism, rendering techniques, shading models, colour models, hidden surface algorithms, 2D algorithms, 3D algorithms, radiosity, surface representations, facial animation, facial modelling, volume rendering, virtual reality, future research. The practical sessions are taught by computer animators from the industry. This will often include someone to handle the creative direction of an animation as well as someone to deal with the technical issues. This is to reinforce what the students have been taught in lectures. The lectures and programming oriented practicals are taught by Computer Science staff. The modelling and animation, creative direction aspects are taught by appropriate experts in the industry.

*Film Aesthetics* – this is a course taught by the Media department at the University. It examines concepts of what is film, the types of film, criticism in film. It is taught by various media professors and improves the students' ability to analyse work. The course also helps the students in designing storyboards and narratives.

*Industry Seminars* – this course is the last before the students have their industry practice. Approximately ten industry experts are invited to give seminars on how digital media is affecting them. These lecturers have included people from film, government, post-production, computer gaming, research, small companies, large companies and so forth. The main goal is to avoid presenting a narrow view of the digital media industry. The students also have a free project of their choice to deliver.

In all the above courses, the students have access to their own lab of equipment. The lab usually consists of fifteen personal computers, six SGI computers, one scanning machine, two printers, a video editing machine, TV's, video equipment, virtual reality equipment and so forth. This equipment is accessible 24 hours of the day to both staff and students in the program. The benefit of the lab environment is that it acts as a place where the students can learn from each other and to develop.

## Practice versus Theory

The Creative Programming program examines many issues of what is important in content in today's digital media. This involves highly theoretical lectures in areas such as HCI, modelling and animation, and computer graphics. The general idea is that the theory does not change remarkably, but the tools do. Therefore, it is important to stress the theory more than the tools. This has become obvious in several instances such as the 3D animation industry, where software updates and competition is fierce. The important thing that is stressed in the education is that the tools are irrelevant. With good support of industry we are able to achieve this high standard of education, without compromising the principles of quality with quantity.

HCI is a very important part of computer graphics, and hence the program involves a heavy cognitive theory section. As we see in Figure 1, simple design tasks are given to the students to help build up an understanding of simple, easy to use interfaces. When educating both programmers and artists in computer graphics, a pragmatic approach is required. We stress the theory aspects of computer graphics, modelling and animation to both students, as these principles generally will remain the same. The practical aspects of the course are performed by industry, and as of such, we do not stress the tools that we use, only how the theory affects the tools. Looking at Figures 2 and 3, we can see some experiments with lighting, reflection maps, and radiosity concepts using the tools available in the labs. This is stressed in the lectures, and the students are encouraged to experiment with the theoretical ideas presented.

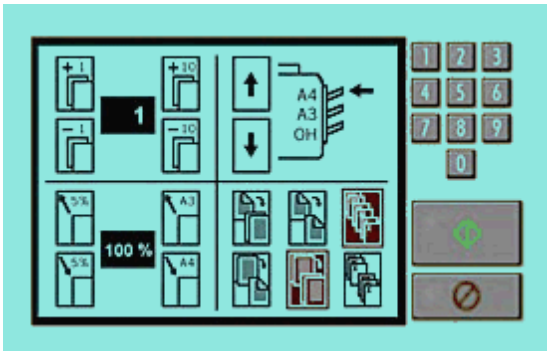



Figure 1. The theoretical concepts of Design and Cognition are used to help students to design simple and language independent interfaces. In this particular example, a new photocopier interface is designed.

Group projects are often encouraged to educate the students about collaboration. There is also plenty of time for students to perform individual projects. During the Computer Graphics, Modelling and Animation course, the students are strongly encouraged to submit their work to various international competitions. These include the Rhythm and Hues student competition, the Alias student competition for instance.

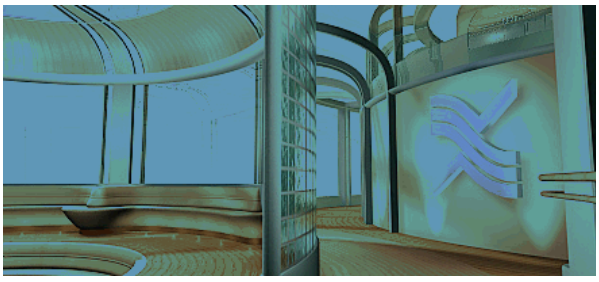


Figure 2: A still from an animation to show the future dental clinic. This is a project with real deadlines and real clients.

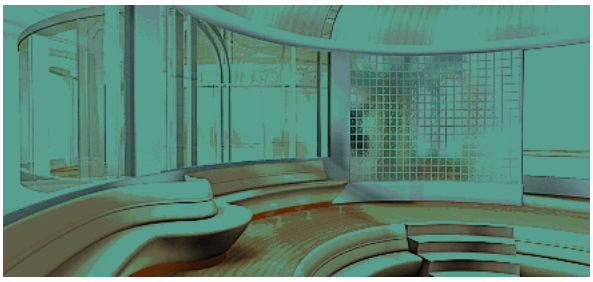


Figure 3: A still from a two minute animation presenting the future dental clinic. This was produced after a highly theoretical lecture on radiosity.

## Industry Involvement

The final stages of the students' education is spent in industry, where the student specializes in some area. Generally, this has included programming in the gaming industry, 3D industry (animation and modelling), and also the web industry. The process of matching the student to industry is often a complex task. This usually involves various visits by the student to different companies that they like. This is complemented by visits by companies to the lab where they are able to see the student at work. When a successful company has been found, the student then can begin their practice there. Some conditions are stressed by the University about the practice session. These include that the student be guaranteed a computer and desk, that they are working on some project, and that they are given one day a week to prepare a report and presentation. At the end of practice (twenty weeks), the student then presents a major presentation, which is either research related or practice related. To ensure that both the student and the client are satisfied constant communication by the subject controller to both parties is maintained. It is in this process that feedback about both client and student is received. This is important as it allows us to review the selection process for both the student and the company. At the time of writing, we are about to send out our fourth group of students into industry. This process and relationship with industry was discussed in [2]. In Figure 4 we can see the result of one student's work whilst in industry. The student worked in the special effects industry and produced over five productions that were shown on television. Figure 5 and 6 demonstrate how the movement from traditional art to computer generated art can produce very good results. The art work in Figure 5 was in part of course work. The model in Figure 6 was individual free time work that was applied after various theoretical lectures on facial modelling. Figure 7 is a very complex model that has been submitted to the Rhythm and Hues international student competition.


Figure 4: A still from the industry project where a student worked on a commercial that was presented on television.

Figure 5: Traditional artwork.

Figure 6: Facial modelling by same student as in figure 5.

Figure 7: A rendered model with advanced textures.

## Conclusion

In this paper a new digital media program is described, one that attempts to merge the computer programmer and artist together. The program aims to provide the opportunity for the programmer that has artistic skills to combine them in an education, and for the artist that has never approached a computer, to learn and improve themselves. As APM is a very good predictor of academic success, it can also be used to rise the confidence of this group of applicants in that they will manage the theoretical and technical courses in the program. An examination into the methods used in combining theory of Computer Graphics with practice is also explored. A lab environment is used, where the different cultures of people, experiment with new techniques, and use each other as reference of information, both technical and aesthetic wise. This means, those that are familiar with computer programming, can educate the artists in some aspects, and in return receive aesthetic criticism and tutorage. The education as such, concentrates on theoretical aspects of multimedia, design and cognition, computer graphics, modeling and animation, aesthetics, and media issues. The practical side generally involves reinforcing the theory, but also acts as an avenue for the students to enhance their creativity. Industry involvement is also crucial, and hence the students often find their practical work has a real end purpose.

## Acknowledgements

Students Torsten Edwinson (figure 1), Mattias Malmer (figures 2,3 and 7), Hanne Drossel (figures 2,3), Mikael Håkansson (figure 4), Ulf Lundgren (figure 5 and 6). Eva Carling for her comments regarding the Ravens test.

## References

[1] Raven, J.C. (1993). Advanced Progressive Matrices, Sets I & II. 1994 edition, with new norms for UK and USA adults and other national groups. Oxford Psychologists Press.

[2] M. Ollila, J. Kempff, and J. Ljungman. Where Industry and Academia meet : an international perspective. SIGGRAPH 98 Panel Presentation, Orlando Florida.

# Computers in Art an d Design Education — Past, Present and Future.

**Tony Longson**
California State University
Los Angeles, California.
tony@design.calstatela.edu

**Abstract**

*A backward look at the use of computers in Art and Design Education is used to speculate about the future in an attempt to anticipate and determine events. The author sites practical examples, and raises more questions than he knows how to answer.*

**Keywords:** *Art and Design education, technology, software-packages, Object-Oriented, creativity, problem-solving, browser, collaboration, tailored environment.*

## 1. Introduction

Design consists largely of problem solving. In a 1972 interview [1] Charles Eames answers the question "What are the boundaries of design?" with another question - "What are the boundaries of problems?"

Art too involves problem solving. What distinguishes it from design and promotes it to one of Mankind's highest endeavors is that an Artist also has to invent the problem to be solved.

Computers are problem solving tools and, if any justification were needed, this alone is sufficient for using them in the visual arts.

In order to solve a problem the computer needs to be programmed. This provides a creative potential for the artist or designer - visual problems need to be expressed in a structured way through a language with its control structures and variables, algorithms need to be invented. The process requires one to think of the creative problem in new ways and in doing so innovative creative solutions will inevitably show up.

## 2. PAST (1960's - 1986)

I started to use computers in my art in 1973 and was annoyed to find that they could make better drawings in seconds than I could make in weeks. There was a precious quality to my drawings that the computer had no regard for. Taking the skill out of my work made me look more closely at the content. My second thought was a feeling of excitement that I could define rules that would condition what the image would be. My arts heritage was the English Constructivist movement, the Systems Group and the Constructionists. And my third thought (all this within an hour or so) was that I had to teach my students how to use computers in their work.

The tools at our disposal were limited: BASIC or FORTRAN with simple device control such as moveto and lineto, output was a small green CRT or, when the ink didn't run dry, a pen plotter.

Despite, or perhaps because of, the limitations, the work that people produced was extraordinary both in its originality and its integrity. Examples of this work were shown in the Siggraph '86 Art Show curated by Patric Prince[2].

I was visiting lecturer to London's Slade School Postgraduate Experimental Area lead by Malcolm Hughes. A handful of students working with an eviscerated Data General mini-computer included Chris Briscoe, who pioneered commercial computer animation in England, and Paul Brown who has been a major influence in education. Brown's early work can be seen at his web site.

The tools settled out into three categories: 2D graphics packages, 3D graphics packages and 'Paint Programs'. The first two were really the domain of scientists and engineers as they worked in conjunction with a programming language. The Paint Programs were something of an anomaly. Developed by scientists who presumably had some idea that they would be useful it was up to artists to demonstrate the fact.

While people were asking whether computers could be used creatively (Yes), artists like David Em (Jet Propulsion Laboratory) and Darcy Gerbarg (New York Institute of Technology) were quietly getting on with it. The Paint Program at JPL was written by Jim Blinn who used it to make texture maps for the moons and planets in his simulation of the solar system. As I recall, he usually got his brother (an artist) to do the work.

The next question up for discussion was “should artists and designers know how to program?” I said yes, and was derided for it! In fact I had already established two successful educational initiatives in Computer Graphics for Artists and Designers based on that premise. One was at West Coast University in Los Angeles (now defunct), and the other - where I still teach - is in the Art Department at California State University, Los Angeles. I think they were the first of their kind. People were lining up to get in [3].

Frankly it wasn’t too hard to make that decision. I had taught myself to program and enjoyed the unique feeling of control over creative questions that it offered; I could see excellent work from other artists; and... there wasn’t anything else to teach! Software packages didn’t exist and the use of computers in Design was just beginning. In 1984 the Macintosh suggested a potential for design but didn’t deliver until 1986.

## 3. PRESENT (1986-now)

It was the invention of the Laser Printer and its page description language, Postscript, that delivered the promise of computers in Design -_Output.

Desktop Publishing became a practicality. Commercial success lead to cheaper computers. Cheaper memory lead to more colors (one of the rare examples in life where you get more out than you put in,) more colors created a need for paint programs, 2D and then 3D modeling and animation packages became common, and HyperCard established a practical basis for interactive design. Specialized software packages proliferated - and that is what people began to teach.

Why not teach software packages? They offer a well-defined body of information, usually come with documentation and tutorials, and are easy to justify to administration. Furthermore there are hundreds of job adverts which say “must know PageMaker”, and hundreds of students who want to get that job. As a consequence you’ll see academic courses in Universities across the world with titles like “Advanced Photoshop”. It sounds reasonable but it is a mistake.

Academic courses in Art and Design are about creativity and problem solving, the tools are secondary. Also the tools are out of date in a few months, so that a student who know menus and commands and keyboard shortcuts for one program will be at a loss when the new version comes out.
An alternative is to teach the principles behind the software and to set this in a creative context where practical projects encourage an exploration and understanding of the concepts. The principles behind the software will survive. Teaching a student the principles of computer graphics means that she will be able to transfer her knowledge to whatever environment of software and hardware she finds herself in.

Computer Graphics software seems to separate out into categories: drawing programs such as Illustrator or Freehand characterized by a 2D coordinate system with transformations operating on object-oriented lines and shapes; raster based graphics such as Photoshop characterized by pixel-level manipulation, layers, and image-processing techniques; 3D modeling and animation programs such as CINEMA4D or Maya characterized by 3D coordinate systems, 3D objects with transformations applied to them, rendering, and the notion of change over time; and Interactive Design which may include any of the other categories but is characterized by non-linear presentation of ideas and information in a context which is easy to negotiate.

It is not surprising that these software packages had their origins in the 2D, 3D and paint programs of the early days. Even Interactive Design, proposed by Ted Nelson in the 1960’s, had most of its design principles raised and codified before the Internet made it a reality.

I try to tailor projects to encourage an exploration and understanding of the principles behind the software. I make the projects “self-starting”. Creativity is a cycle of activity and review. Often students find it difficult to start the process so I invent projects which will have them producing objects and images without much effort. Usually the projects are rule-based. When they have an image in front of them that has not required too much effort on their part students are usually more open to critically evaluating it and making changes, and so they quickly become involved in the creative process. Once the image or object has a life of its own the students can respond to it creatively.

I also try to invent projects which will allow the students to juxtapose their work with someone (or everyone) else in the class.

You might agree that the idea of teaching principles (not packages) is good but argue that at some point the student needs to be shown details of the software package such as which menu item to select or window to open. I would counter that if the student can’t figure it out for herself then there is something wrong with the software package. As consumers we wouldn’t dream of buying a domestic appliance such as a washing machine or a car whose operation was not

self-evident. Software should be the same. I usually take a naive approach to teaching software explaining what they might expect the package to do and putting responsibility on to the student to find out how to do it. This encourages self-reliance, and a problem-solving approach.

As artists and designers we tend to push software and hardware to its limits which means that things can go wrong, so problem-solving is an important talent. I encourage it.

### 3.1. Trends

Paul Brown spoke of the future as the invisible places beyond an "event horizon" [4]. Extending his analogy I suggest that we build a platform of current knowledge and use that vantage point to get a glimpse of what might lay ahead.

Programs have become larger as a result of cheaper memory. Developers wish to provide more functionality and so there's an overlap between what used to be distinct software categories. As a trend this is good but the consequence is an unweildy and expensive software package. The concept of 'plugins' has gained popularity as a way of extending the functionality of a software package without making it monolithic. Many software developers allow for and encourage them. Browsers thrive on third-party plugins to extend their scope.

Object Oriented Programming is becoming the standard, and implies a flexibility that will allow for problems and solutions of which we can't yet conceive of. CINEMA 4D is an example of object-oriented animation software. The manual apologizes at one point:

> "This is not meant to challenge your imagination but is simply the result of CINEMA 4D being a totally open system. Nobody knows the type of animation effects the future might bring and (what) you may wish to assign them to."

Web-based Objects are taking off.

## 4. FUTURE (now to the next Millennium)

### 4.1. A Disclaimer.

Predicting what Art and Design will look like in the future is oxymoronic. As soon as you describe it - it exists (in the present.) Witness the failure of Disney's Tomorrowland, and Mr. Spock's three-dee chess board. It's safer to invent a retro-future such as Kubrick's Louis Quinze interior in 2001 A.D.

However we can anticipate a working (and teaching) environment, and possibly even help to make it happen.

Software packages will cease to exist as we now know them. Applications will consist of objects and messaging. You won't necessarily know what computer they are on. For any particular task you'll be assembling (without much effort) the appropriate modules. You won't own software any more, you'll rent it, paying a fraction of a penny for each object each time you use it. Equally your own objects may be sought after items and provide some revenue. Forget about software upgrades, or new versions of the operating system. Any one of the millions of the components of which they consist could be updated transparently at any moment in time.

Browsers will probably be the only software on your computer. They will be the portal to your web-based software modules, and they won't look like they do now (no clunky windows and that anachronism known as scrolling).

The result of this will be a trend to tailored environments. As a designer you'll have your own set of tools (like craftsmen of old) which you will hone and refine. These tools will result from the kind of work that you do and will reflect your creative sensibilities. They will be an analog of your distinct style.

Think of the potential for teaching. Principles will still abide, but now you can design a tailored environment appropriate to who and what you are teaching.

As students mature creatively you'll teach them how to build their own tools (if they haven't already found out!) Is this back to programming? Yes, in the sense that you'll have control over your own solutions to problems. But not in the sense of sequential programming with a few simple control structures and variables. Object-Oriented Programming demands a different mental agility and the need to know details of huge libraries of existing objects and their syntax. Good user interfaces will facilitate this.

### 4.2. Output - put out.

More of the things designed on the computer will use the computer as a means of delivery. Physical output, will diminish for several years until the Rapid Prototyping Systems become consumer appliances. The Prestigious Prix Ars Electronica Competition had a category called "sculpture" for several years

eventually to replace it with “Web-based media”. Conincidentally this was the year I planned to submit!

### 4.3. Collaboration - the Good, Bad, and the Ugly

Early Siggraph conferences often showcased collaborative paint programs. You could add to an image on a monitor at the same time as someone in an adjacent booth or across the world. Somehow the results were always disappointing (the Bad). Perhaps a lack of clear goals lead to what was a competitive rather than collaborative exercise.

Algorithmically generated images have been posted on the Net which one can rank (the Ugly.) Subsequent generations of the art work spawn from images that were most popular - Aesthetic Darwinism, perhaps, or more likely the cyber equivalent of Velvet Elvis.

As part of the Electronic Theater at Siggraph some years ago, Rachel and Loren Carpenter set up a collaborative visual venture (the Good) which continues to astonish me. Each member of the audience had a wand with reflective stickers on each side, green and red, which they could hold up to signal a binary decision. The collective audience response was grabbed by a camera at the front of the auditorium processed in some way and revealed on the cinema screen.

In the first exercise a large circle was drawn on the screen. With the whole screen representing a map of our seats in the auditorium we were asked to signal one color if we thought we were inside the circle, and another if we were outside. Within a second or so the circle filled in. In a couple of seconds more the edges of the circle were well-defined and stable. How did this happen with such authority? Each person made a single binary decision. The decision was based on some knowledge of their location in space. They could perhaps see what decisions their immediate neighbors had made. The image on the screen provided some general feedback but not specific to an individual’s decision.

Though in its consequence this example may seem trivial I think the implications are profound. The notion could be extended to creative tasks using the Internet both as the equivalent of the auditorium and of the screen. Collaborators would provide far more sophisticated decisions than in our example, and their role in the task would be correspondingly significant.

The nature of creative problems that could be addressed and the design answers we may see are boundless.

## References


[1] Paul Goldberger. The Eames Team. In *The New Yorker*, pages 90-99, New York, May 1999.

[2] *Art Show Catalog*, Siggraph ACM, Dallas, 1986.

[3] Judy Brown. *Teaching Computer Graphics: An Interdisciplinary Approach.*, pages 189-190 Siggraph ACM, Anaheim, 1987.

[4] *Computer Animation Workshop.*,California State Summer School for the Arts, Humboldt, California, 1990.


## Acknowledgements


To Bob Holzman for enabling the Present and Future in my career and to my Stepfather, Ken Dixon.

# Online Education

**Anne Mumford**
Audio Visual Services, Loughborough University
Loughborough, UK
a.m.mumford@lboro.ac.uk

***Abstract***

New technologies for distance learning through the Web, video conferencing, collaborative working tools have the potential to enhance and extend learning. This requires a large leap for many institutions, teachers and students as well as an investment in technology and training. Many of those involved in graphics and visualization have in some way contributed to the development of the technologies being used and to their potential application use. These technologies have potential for widescale use at a time when there is pressure, at least in the UK, to widen access to higher education and to consider learning to be a life-long process.

This paper challenges those involved in Graphics and Visualization Education to look at the following areas:

*Distance Learning*
For collaborative ventures to share materials, bring in guest presentations from international speakers and to develop and enhance collaborations between those involved in Graphics and Visualization Education.

*Collaborative Working*
Research has been presented to the Working Group about the possibilities offered by collaborative working for visualization. The group should promote such work and build on it widening the application areas and involving other subject areas.

*Widening Access – Promoting Participation*
The technical and pedagogic developments of those involved in Graphics, Visualization and Virtual Environment education have the potential to open up education to those needing to study remotely from a University. The potential of such techniques should be promoted within a range of application areas.

*Getting Online With More Disciplines*
Much of the development of graphics and visualization tools has occurred within Computing Science, Engineering and the Physical Sciences. The benefits have great potential in other areas, particularly in the social sciences. Data mining and understanding large and complex datasets, such as longitudinal datasets, is of considerable importance in the social sciences. The potential for the use of visualization techniques within research and teaching seems to be thus far unrealised.

# On the Functionality of Distributed Multimedia-System Architecture for Educational Applications

**D. Dochev[1], R. Pavlov[2], R. Yoshinov[3]**
[1]Institute of Information Technologies – Bulgarian Academy of Sciences
[2]Institute of Mathematics and Informatics – Bulgarian Academy of Sciences
[3]Computing Center - Bulgarian Academy of Sciences
dochev@iinf.acad.bg, radoslav@bgcict.acad.bg, yoshinov@bgcict.acad.bg

**Abstract**

*The paper discusses the functionality of distributed multimedia architectures for educational applications. It reflects some developments made under the international project ARCHIMED "Advanced Multimedia-System Architectures and Applications for Educational Telematics" (part of the EC RTD program INCO-COPERNICUS). The aim of this project is to explore and implement technological methods and tools for creation and distribution of multimedia educational materials. The main elements of the architecture implementation are the project Local Internet Support Centres. The functions and structure of the Bulgarian Support Centre are presented with the adopted hardware and software platforms, the initial repository content and the pilot courses to be developed during the project.*
**Keywords:** *educational multimedia, distributed architectures, virtual campus*

## 1. Introduction

The educational multimedia data management has to deal with several specific features:

- The necessity of interactive use of the multimedia materials, allowing individualised feedback and discussion.
- Extensive use of distributed multimedia resources available on WWW.
- The necessity of effective student navigation to ensure a balance between the two extremes of total constraint and total absence of constraint, according to the student's needs and current state of knowledge.
- The necessity for versatility of composition for fast update and modernisation of educational content.
- The necessity of modular and open-system organisation.
- The need of intelligent assistance in information handling.
- The availability of a big pool of small educational content providers.

All real implementations of systems, accessing multimedia documents for a given class of tasks, have to compromise between the requirements for full functionality and the constraints imposed by the available resources. Therefore in order to create a viable pragmatically and technologically sound architecture the solutions to be applied have to consider:

- the modern tendencies in organisation of multimedia systems;
- the educational needs of the project end-users;
- the developments in the modern computer-supported learning environments;
- the current computer and communication structures of the university partners in the project and their possible development in the near future.

## 2. Conceptual Base of the Distributed Multimedia Educational Architectures

The architectures under discussion are build upon the following types of conceptual models:

1/ Multimedia data models, specifying the structure of multimedia data management systems and the methods applied for storing, composing and retrieval of multimedia documents. The multimedia data models have to provide useful abstractions and related mechanisms to support:

- Generalisation/specialisation hierarchy
- Attribute specification for describing the properties of a document
- Specification of operations, performed on a multimedia document
- Manipulation of composite objects
- Objects sharing
- User-oriented presentation.

The multimedia data management requires mechanisms, common for classical databases to ensure integration and integrity control, query support, privacy, version control etc. However some features are of special interest as due to huge data volumes and variety of information their achievement is more difficult - efficient data capture and access; data availability; data persistence and recovery; presentation of delay-sensitive objects etc.

2/ Pedagogical model, specifying both the structure of multimedia educational documents and various ways of their use (for teacher-centered or learner-centered organisation of the educational processes). Many current educational projects for distributed computer environments apply the principles of constructivist pedagogy and use models, based on the following leading ideas [1, 2]:

- Learning should be context based, i.e. learning experiences should be contextualised in authentic activities; learning is acquired through making links with existing knowledge.
- Conceptual learning is through active involvement: a task is understood through participation in it; learning involves creating personal meaning and understanding; experience becomes part of the meaning.
- Learning is through collaboration with others: sharing knowledge resolves misunderstandings; interaction leads to new knowledge; understanding evolves from shared knowledge constructing;
- Learner should have personal autonomy and control over learning: modern learning involves personal decision making, formulating own learning strategies and own goals; teacher mediation depends on needs and skills of the learners.
- Learning is personal growth - argument leading to reflection helps refine concepts; outcomes are unique to the learner;
- Learning outcome is a perspective and an understanding: specific content and learning outcome should not be prescribed; multiple perspectives of the learning task and different approaches to understanding are needed.

3/ Model of communications, organizing the information exchange of multimedia information in distributed computer environment. The model of communications has to ensure proper multimedia environment in networking inside and between the partners. The partners networks normally include Proxy servers, Cache servers, WWW servers, Name servers (DNS, WINS and DHCP servers), Mail servers, FTP servers, Terminal servers, Application servers, Database (SQL) servers. The new quality of the network model is achieved through appropriate mechanisms for cooperation of these servers and the extension of their capabilities using protocols and standards (e.g. ActiveX, specific plugins etc.) recommended for exchange of multimedia information.

## 3. End-User Groups and Generic Services

The ARCHIMED architecture is meant to serve two different groups of end-users:

A. Learners, accessing and learning with multimedia educational materials: university students, students in specialised schools, participants in vocational training educational forms, non-formal and life-long learners. The architecture to be developed has to ensure sufficiently rich environment, relevant to the modern pedagogical approaches and tendencies for teacher-assisted and learner-centered education in local and global computer networks with some collaborative learning possibilities.

B. Authors of multimedia educational materials: university lecturers, high-school teachers, lecturers in specialised training courses. As a rule the authors are not IT specialists, so it is not real to expect that they will be willing to use authoring systems and tools with high level of complexity to produce courseware. The majority of these authors are also meant to use the created multimedia educational materials directly in their teaching practice.

These two groups of end-users require two classes of generic services to be supported by ARCHIMED architecture. These services may be organised using the modern computer-assisted education metaphor of "virtual campus". In order to develop a specific detailisation of this metaphor, suitable for the needs of the project, the participants in the educational process and their main roles has to be revealed. Similarly to [3] we consider the following main actors in the educational process in the “virtual campus”:

1/ Learner - transfers information into knowledge. S/he uses the courseware from the distributed repository (“virtual library”) acting in the following roles: explorer of repository resources; navigator through learning scripts; self-evaluator of personal assignments; participant in collaborative learning (e.g. peer review, debates in telediscussions etc.).

2/ Trainer/advisor – assists/advises the learner in the educational process acting in the following roles: produces diagnosis; advisor; assignment evaluator; coach.

3/ Manager - manages actors and events, acting in the following roles: planner; decision-maker;

supervisor and controller; team or group organiser; director of learning assignment.

4/ Author – supplier of educational content. S/he produces courseware using the MM materials from the repository by means of "author studio" tools, acting in the following roles: document analyst; content expert; teaching materials designer; instructional scriptwriter.

5/ Mediator ("information broker") – facilitates the navigation of the other participants through the "virtual campus" acting in the following roles: information communicator; user profiles production and maintenance; help in environment use; intelligent help in document retrieval.

Each actor is expected to assume various roles; a given role may be assigned to more than one actor. The roles are related with specific information subprocesses, which have to be supported by the educational architecture under development.

These groups of end-users require two classes of generic services to be supported by ARCHIMED architecture:

A. Services for the learners

In order to ensure interactive use of multimedia courseware by the learners the ARCHIMED architecture has provide the following components:

- Access to a distributed repository ("virtual library"), containing multimedia documents and interactive courseware with different levels of complexity;
- Tools, organising the processes of supervised learning or self-learning using multimedia courseware in a distributed computer environment;
- Tools, facilitating the navigation through the repository according to the user's needs and profile ("information brokers");
- Tools, organising asynchronous/synchronous group educational activities in distributed environment (e.g. link to human tutors, bulletin boards, peer reviews, discussion forums, conferencing etc.).

B. Services for the authors

These services provide means for the authors to store, maintain, archive and finally produce their educational multimedia materials. In order to cover this service class the ARCHIMED architecture has to support the following elements:

- A document directory system and a database for storing the digitized pedagogical materials;
- A web interface for uploading materials on a web server;
- A library of documents templates (generic models or scenarios) to be used by the author to create pedagogical materials;
- Secure discipline for accessing the web server with the document templates;
- An instructional methodology to guide the author in his authoring process.

## 4. Implementation of ARCHIMED architecture

Though the ARCHIMED architecture doesn't address directly the whole set of distant learning activities, it covers an important part of the popular in the last years "virtual campus" metaphor. The virtual campus is structured in virtual spaces, organising the realisation of the abovementioned information subprocesses. A possible set of virtual spaces may include [3]:

- An information space - repository ("virtual library") containing various types of documents or data required by the actors to fulfil their function. It contains multimedia materials (text, graphic, audio and video objects) selected thematically for the needs of the courses produced/distributed and courseware on different levels of complexity
- A learning space with the requisite tools to complete assignments and to participate to learning system.
- A communication and collaboration space with tools enabling actors to communicate, carry out group activities and participate in remote seminars and teleconferencing.
- An assistance space where actors may get support and advice or have their environment customised, by calling upon either online support personnel or computer-based help resources.
- An authors' space, containing tools permitting the authors to store, maintain, archive and finally produce their educational multimedia materials.

The models and tools in the architecture under consideration will be experimented on the following target groups - 1/ graduate students of information technologies studying courses on multimedia technologies, computer animation, web graphics and design; 2/ graduate students of journalism for courses on mass media advertising. The experimental courseware is developed considering some requirements for learner-centered and project-based learning.

Considering the current state, tendencies and traditions for hardware use in the partner's sites ARCHIMED architecture maintains the existing strong PC

orientation. The supporting operating system and **multimedia** database management system have to ensure flexibility in application interfaces, systems intercorporability, connection decisions, network protocols, routing, distributed component realization, security on all levels etc. Considering the PC domination in server and client trends, Microsoft Windows NT was chosen as an appropriate choice for system support. Windows NT server looks proper server operating system for the goals of the project and Windows NT workstation or Windows'95/ Windows'98 - a proper client operating system.

For management of multimedia documents in network environment an appropriate MMDBMS seems Microsoft SQL Server which has all the capabilities, needed for such types of databases. SQL Server offers distributed security integrated with Windows NT with central control of passwords across a domain and new encryption services for logon and data stream. SQL Server remains an open, interoperable platform that works with what customers have and know.

The virtual campus of ARCHIMED relies entirely on the Internet connectivity for the communications between the project partners. The respective transfer speeds are function of the communication policies and finances of the institutions of the project partners. Their networks will be extended by adding Cluster servers for achievement of more power and throughput running the applications and Remote Access Servers for serving remote clients (from universities, schools, SMEs etc.) who will use the developed environment (methodology, tools, templates, courseware etc.) in the frame of ARCHIMED. The connections between the Bulgarian partners from Bulgarian Academy of Sciences work at the speed of 100 Mbps, as the three institutions are situated close by. The connections are based on UTP with Switches (devices like Cisco1900, Cisco 2500, Switch-gateway Cisco3100), using TCP/IP as main routing protocol. The Local Internet Support Centres use Remote Access Servers, based on PPP and PPTP protocols for support of the remote clients.

More information about the design decisions, functions and the structure of ARCHIMED architecture may be found in [6, 7]. The overall structure of the Bulgarian Local Internet Support Centre is shown on Figure 1.

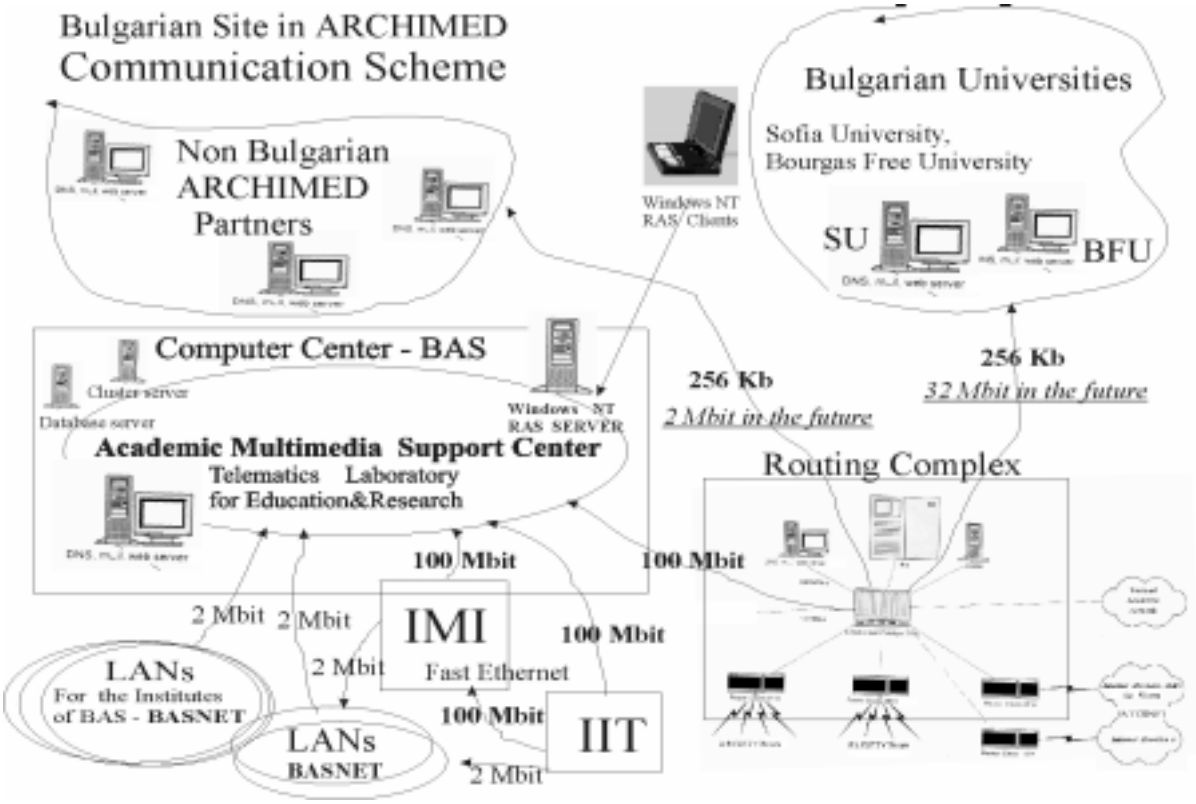



Figure 1 **Structure of the Bulgarian Local Internet Support Centre**

## Reference


[1] J. M. Ewing , J. D. Dowling, N. Courtis – Learning Using the World Wide Web: A Collaborative Learning Event. In: *Journal of Educational Multimedia and Hypermedia*, Vol. 8, No 1, 1999, pages 3-22.

[2] E. B. Susman – Coperative Learning: A Review of Factors that Increase the Effectiveness of Cooperative Computer-Based Instruction. In:*Journal of Educational Computing Research*, Vol. 18, No 4, pages 303-322.

[3] F. Henri – Designing a Virtual Learning Environment for a Graduate Course on Multimedia. In: *Davies G. (ed) Proc. of XV IFIP World Congress, Teleteaching'98 - Distance Learning, Training and Education*, Part I, pages 427-444

[4] C. Hmello, M. Guzdal, J. Tums – Computer-Supported Collaborative Learning: Learning to Support Student Engagements. In*:Journal of Interactive Learning Research*, Vol. 9, No 2, 1998, pages 107-129

[5] G. Paquette – Engineering Interactions in a Telelearning System. In: *Davies G. (ed) Proc. of XV IFIP World Congress, Teleteaching'98 - Distance Learning, Training and Education*, Part I, pages19-41.

[6] D. Dochev, R. Pavlov et al. - Modelling of Data Management. *Project ARCHIMED PL961060, Report D2.1*, 1998

[7] R, Yoshinov et al - Modelling of Communications. *Project ARCHIMED PL961060, Report D2.2*, 1998

# Three Collaborative Models for Computer Graphics Education

**Colleen Case**
Schoolcraft College
Computer Graphics Technology
Livonia, MI 48152 USA
ccase@schoolcraft.cc.mi.us

**Abstract**

*Visual communication is an area that acts as an integrator to other disciplines, business work projects and institutions. The collaborative models and innovative delivery systems described in this document provide "real world" experience, initiate accelerated learning and develop flexible structures for curriculum that meet the needs of business, industry, the learner and the educational environment. These models and samples address many of the current issues facing the computer graphics profession.*
**Keywords:** *collaboration, inter-disciplinary, distance learning, video- conferencing.*

## 1. Introduction

Schoolcraft College is a public, tax-supported community college serving the people of northwest Wayne County, in Livonia, Michigan USA. The college district is composed of five public school districts: Clarenceville, Garden City, Livonia, Northville, Plymouth-Canton and part of the Novi Community Schools. Schoolcraft College was founded October 24, 1961.

The Computer Graphics Technology program at Schoolcraft College prepares students for careers in computer graphics and graphic design through a combination of classroom and real world experiences. With this mix, students learn to apply technical skills and develop professional work skills. This document identifies some issues facing the graphics profession and describes some structures and models used to address those issues.

## 2. Computer Graphics Education Issues

- shortage of qualified graphics professionals
- keeping curriculum current with rapidly changing technologies
- student expectations for high quality applicable experience

The shortage of qualified graphics professionals should not be over simplified to numbers reflecting supply and demand. The shortage is also due to a lack of coordinated efforts between business/industry and educational systems. Educational institutions struggle to keep the curriculum current due to both rapidly changing technologies and changing industry needs. Students expect and should receive a high quality educational experience that is applicable to the needs of the market. These are not new issues but the speed of change requires a new way of dealing with them. The old model of education, work experience and curriculum development as separate activities needs to unfold into a real-time simultaneous activity. Students can learn by experiencing real projects that simultaneously identify curriculum needs and deliver learning (section 3—Sample).

- complex integrated communications, software and hardware technologies require multiple experts
- faculty development and training are an institutional issue

Departmentalized educational institutions do not reflect the integrated multi-disciplined reality of the world. Complex integrated communications, software and hardware technologies require multiple experts for project/product development. Computer graphics and visual communication is a field that acts as an integrator to other disciplines. Educational systems can provide a value-added environment by allowing interdisciplinary experimentation. Coordinated joint instruction is not just a bonus for the student but has the added value of faculty development. We need to look at how we use our time—how we can allocate

human resources to simultaneously teach and learn (section 4—Sample).

- digital media, and communications technologies such as Interactive Video-conferencing and the Web are mechanisms for developing new relationships and experiences in collaborative work

New communications technologies have opened up opportunities for collaborative relationships. We have a need to share information in order to remain current in the specialized technologies we pursue. Large complex projects require multiple expertise and the Internet and Interactive technologies provide mechanisms to share expertise and information (section 5—Sample).

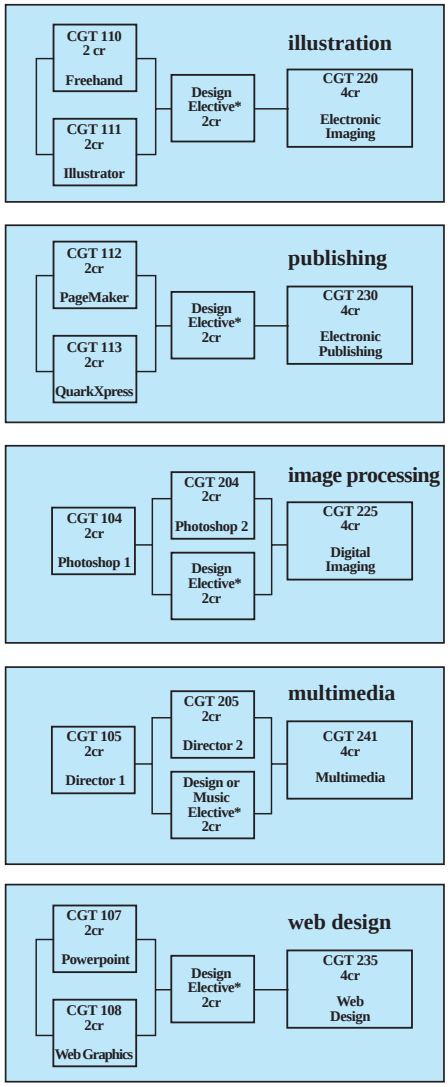
## 3. A Flexible Program Structure …enables a "Real World" Project Philosophy

The Computer Graphics Technology department has adopted a "real world" project philosophy. To implement this philosophy, Stephen Wroble developed an innovative flexible curriculum structure that meets the changing needs of students, faculty, local business and industry, and can adapt to a changing technical environment. By creating a track system with crossover electives, students can create a more specialized sequence of instruction. The curriculum structure was developed with growth and change in mind.

The need for frequent curriculum revisions and for specialization due to rapid changes in graphics technology is also built into the curriculum structure. The design elective structure provides a mechanism for adding new courses and a testing ground for new tracks (see Figure 1).

For example, the screen design elective was implemented initially for the Multimedia track and initiated the development of the newest Web Design track. The storyboarding design elective is useful for many tracks and is currently a testing ground for future animation and videography tracks.

There are currently 5 tracks in the curriculum. Publishing, Illustration, Digital Imaging, Multimedia and Web Design. Students choose 2 tracks for an associate degree or can get post associate certificates (see Figure 2). Our student population includes both the traditional students and graphic professionals returning to update their skills. This mix gives the classroom an energetic dynamic and a professional maturity.




**Design Electives Flexible Module Choices**

| | |
|---|---|
| CGT 150–Typography | Design modules in development |
| CGT 151–Survey of Design | –Color Theory |
| CGT 152–Screen Design | –Legal/Business Issues |
| CGT 153–Portfolio Development | –Interactive Writing |
| CGT 154–Music Sound Design | –Transformational Geometry |
| CGT 155–Storyboards | –Digital Video |
| CGT 156–Photography | |

Figure 1. CGT Tracks

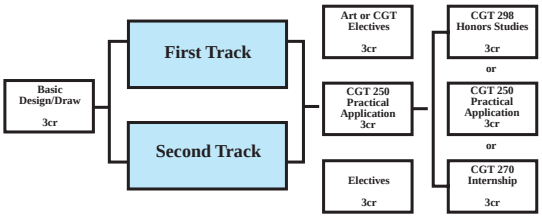



Figure 2. CGT Requirements for Associate Degree

Each track starts with foundation skills in the basic drawing and design; application software skills follow, along with design electives. Once these skills are established, they are applied in specialized projects for the track (CGT 220, 230, 225, 235, 241).

All of the tracks then converge into the CGT 250 Practical Applications. Students from all tracks are mixed into this project class. The "real world" project philosophy provides students with a career experience in a supportive environment.

In these project classes, students' partner with business and industry, non-profit organizations or on-campus marketing design groups to experience and learn through actual projects. Because all projects do not fit neatly into semester timeframes, projects can be extended if necessary into a variety of forms: honors studies, internships or additional practical applications classes (CGT 251, 270, 298).

As a result of working on a real project, students gain valuable experience in:

- Application of design techniques and processes
- Project design and time management
- Systems thinking
- Budgeting and production constraints and analysis
- Collaboration and teamwork and relationship building
- Decision making based on research and evaluation
- Communications, presentation
- Troubleshooting and problem solving
- Critical thinking, and data interpretation

Over the past 4 years, the Graphics Department faculty coordinated close to 200 large and small projects for non-profit organizations and the college. These were used in both the track classes and in the Practical Applications classes. Examples of the projects include layout design for two literary journals, the McGuffin and The Michigan Community College Journal. Logo and T-shirt design for the Walk for World Hunger and the disabled sports Wheel Chair Hockey league, building and campus maps for the Schoolcraft kiosks and Web site, graphic design training materials for Statewide Methodist Church publishers, displays for the Women's Resource Center, layout and design of the Siggraph Education catalog and CD, and cover designs for the Schoolcraft catalog and schedule.

Several projects were also done with six business partners. Each business related project was initiated through the Workforce Development Office. The Workforce Development Office at Schoolcraft College facilitates and implements the development of partnerships between business, industry, the community and education. These partnerships help contribute to the development of new teaching and learning systems for the college. This model allows the business partner to take part in the learning environment and experience the same learning objectives as the student, while producing a product needed for the industry. The Project Driven Learning System and Courses (PDC's) are an example of a new learning system for the college. The first PDC was designed in partnership with General Motors Delphi. Computer Graphic Technology students produced a series of tools (Job Aids) that were designed to help GM employees perform tasks and operations. For the CGT program business partnerships included:

1. GM-Delphi
   Job Aids and Graphic Design Training Manuals
2. Ford Motor-EEME Emerging Export Market Engineers
   Intranet based time reporting interface design
3. Johnson Controls
   CDRom product implementation process training
4. Crucam
   Logo and marketing materials
5. Advanced Communication Incorporated
   Operations manual layout
6. Environmental Technology Corporation
   Marketing Trifold

Imperative in the building of the business partnership is the focus on the priority of the

learning process and not the actual product. Unique instructional tools and methodologies have also been produced to support the delivery of PDC's.

A proposal is written to document the deliverable product, and also includes the responsibilities of the learner, facilitator and business partner in the learning process. The proposal becomes the foundation for the course and dictates both the applied technical objectives, along with the process. A framework has also been developed that identifies a design process that guides the team through the project. (e.g. Understanding the scope of the project, the business partners environment, needs inventory, team processes, applying skills, devising production plans and implementing and delivering the product, debrief and learning analysis.)

As the class proceeds the project's needs are discovered. This happens through meetings, interviews, surveys, and other needs analysis tools. The curriculum needs emerge and are developed around the project needs. In the example of the Ford EEME interface design, CGI scripting, database and programming skills were needed along with the graphic design and HTML skills that already existed in the group. A second phase of the project was established as an internship for creating the database connection to the graphical interface. The students also learned about Ford Motor Company's culture and global markets.

The students maintain a skill process log where they record and reflect on their learning and team processes. A formal learning analysis presentation is required in addition to presenting the final product.

The benefit to the student for all types of projects is a real experience in the production process with both instructor facilitators and partners. The benefit to the facilitator is current curriculum and course development that meets the needs of actual work. The benefit to the partner is a learning environment to interact with potential employees and a deliverable product that meets a need. This direct input into the educational system benefits both the partner and the educational institution.

Two areas of caution and difficulty in implementation of the business projects are the partner's priority and the project selection process.

Partners who focused on the product and were not concerned with the learning rushed the process and became a management issue for the faculty facilitator. Also, projects that were selected more closely by the instructors were inherently more successful experiences for the students. Projects that were chosen incorrectly became extremely frustrating for the faculty. Project selection must be driven by the desire to provide the student with appropriate learning opportunities and not by some other political business relationship need.

By developing flexibility in the curriculum structure, a program has been developed that meets a wider range of individual specialization needs. By developing a framework for experimentation with "real world" projects, the program can respond to the job market in the community, fulfill non-profit service needs and provide a resource for the college. Real projects provide all participants with a relevant educational experience and authentic curriculum development.

## 4. Interdisciplinary Collaborating Environments …Breaking the Barriers of Departments

Several experiments have been conducted with the Computer Graphics Technology department and other departments at Schoolcraft College. With these cross discipline experiments comes a glimpse at the possibility of infinite combinations of instructional offerings. Most recently in the business arena, three instructors are coordinating efforts in a course called Marketing and the Internet. This includes a marketing instructor, a graphic design instructor, and a language arts instructor. The students in this initial offering are marketing students, but the instructors have considered the integration of students from all three disciplines. Students are now analyzing Web marketing from multiple perspectives of language content, aesthetic look, technical and navigational information design, and marketing concepts. They have integrated graphic design and language arts skills into their marketing presentation design, merging different perspectives and blending concepts. Faculty are also broadening their knowledge base and learning from each other.

Other coordinated experiments have included a multimedia class that requires both visual graphics and the synchronization of sound and a Web design project integrating the interface designer and the programmer. An illustration class is proposed to include a graphic design instructor

and a mathematics instructor bringing in the dimension of transformational geometry.

For the student, the benefits of coordinated teaching are obvious. Coordinated teaching provides the student with the multiple expertise of several faculty members or consultants. The students become aware and understand on a multidimensional level. They also witness the collaboration and interaction as the instructors become actively involved in learning. The students were aware that this was an experimental environment and provided feedback and suggestions to all the instructors. This behavior made them more open to critiques of their own work as they learned to appreciated feedback as an opportunity for improvement.

A less obvious benefit from joint instruction is the benefit to the faculty. By working collaboratively in teams, around emerging technologies, faculty members learn together and are able to update their skills; training occurs while simultaneously teaching. The course develops out of the faculty and student knowledge bases and out of the social interaction taking place. This retraining and investment in faculty development will pay off in future offerings of the course and in the development of alternative delivery methods courseware.

Technology choices were made based on how the technology enhanced the learning. In the Marketing class, we experimented with email, discussion groups, Web analysis and Electronic Library research along with presentation technologies. These experiments quickly brought students together and formed a communications mechanism to do work. Marketing was the common language of the class, and the cross-disciplinary approaches were related to this foundational language. A decision to use Marketing as the foundation allowed this first offering to be managed and coordinated around marketing topics, with marketing students.

Future offerings will allow for better integration of language arts and graphic design principles into the planning. Future offerings should be modified to allow for more faculty integration and pre-planning time.

Several issues arose from this experimental class. Administrative support is an issue for new experimental delivery methods. The value of an innovative faculty member is reflected in that support. This is an issue that we must further address in the expansion of the course. Investment in faculty development through joint instruction and support will pay off in the future, allowing for improvement in the quality of our work. Each teaming configuration might require a different amount of overlap. A matrix compensation plan was proposed to assist the administration in dealing with the complex issues of compensation. The matrix relates class session percentages by instructor to a totaled minimum and maximum range for the course. Factoring in the benefit of instructor training can also help justify the expenses of joint instruction.

## 5. Interactive Videoconferencing ... a Shared Learning Experience with Multiple Institutions

The Internet, networks, and interactive video-conferencing systems have expanded the opportunities for collaboration across multiple institutions. After attending a NSF workshop on Instructional Computing, staff and instructors from several institutions set out to create a collaborative learning experience for students, instructors and media staff. By combining computer technologies with distance learning delivery systems, a joint interactive-TV art and design project emerged. This collaborative activity has been done several times over the past 2 years.

Various levels of graphic design students from three institutions were given a similar project. The project this year was to create a Red Cross Blood Drive poster. Students researched using the Internet and gained insight into blood types, donor populations and blood donation needs. Schoolcraft College and the University of Wisconsin-Stout graphic design instructors used email to prepare the guidelines for the project, while the media support personnel verify the compatibility and connection of the systems. Schoolcraft College agreed to act as the initiating site.

Second year students from the community college participated along with fourth year students from the University. The projects varied each year but had some common objectives:

1. Students learned the features and tools of graphic packages, while emphasizing design principles.

2. Students, instructors and support staff technicians were exposed to graphic design as applied to video conferencing equipment. This included a look at screen design and video monitor display including such topics as aspect ratio, safe title areas, interlacing, NTSC and color limitations, along with comparisons between compressed video resolution, TV resolution and computer monitor resolution. This gave the students experience in using features such as color correction techniques, to meet the technical limitations of the media.

3. Students had the opportunity to interact with students from other schools using the Internet and video conferencing equipment. This helped them to develop communications skills and introduced them to people who may be future collaborators. It also exposed them to work produced by students in other programs at other institutions. This is beneficial to the instructor as both a comparison check and for exposure to other instructors teaching approaches and critiquing comments.

4. Students were allowed to control the conferencing equipment from the touch control panel, giving them a unique new experience with a new technology.

Teams of students orally, visually and interactively presented their work for critique using the interactive classroom. Technicians assisted in training faculty and students so they could present live in the Interactive TV classroom.

The interactive TV room has a British Telecom Codec that does any connection from a single ISDN to a 1/2 T-1. The instructor/student used a control panel that is Windows NT based. Equipment is controlled using a touchscreen icon based panel. This includes audio and video switching and peripheral control. Video and audio routing is through software and interfaced to a 12x8 video matrix switcher card and a 4x4 audio matrix switcher. There is an RS232 interface to the professional VCR.

Audio functions from the control panel include microphone mute, gain, speaker volume control and mute, VCR volume control and mute, audio calibration, privacy and refresh. There are two directional classroom and instructor microphones and a speaker systems. Other functions of the system include automatic adapting to the acoustic environment; this eliminates echo, feedback and other noise.

The video functions such as camera selection, camera positioning, AV routing, VCR control and some testing are also controlled on the touchscreen panel. There are classroom cameras, an instructor camera, and a document camera; which serves the purpose of an overhead projector yet allows 3-D objects to be displayed. Twelve preset camera positions can be stored as a profile, allowing different instructors to preplan their camera positions. Panning and tilting up down and diagonally are controlled with variable speed tied to the firmness of touch. The VCR controls (play, stop, record, pause, fast forward, rewind, counter, reset, eject) are also available from the touchscreen and video recording can be of the local site or the remote classroom.

Other peripherals include; the Computer with Internet access, a video disk player, a facsimile machine, telephones for fax, voice-speaker, and remote diagnostics. There also is the capability to take control of the remotes site's cameras. One remote site demonstrated their wireless tracking system feature. The tracking system allowed the instructor to move around the room and remain on camera.

Schoolcraft College belongs to MiCTA. This is a communications organization that follows the recommendations of the (ITU) International Telecommunications Union. The ITU coordinates global telecommunications networks. (www.itu.int) MiCTA is a support mechanism for the college technicians and works to maintain interoperability specification for equipment and connections with other sites. Information about similar video conferencing systems can be found at (www.micta.org/mictamap/).

Students from Schoolcraft College and Washtenaw Community College and the University of Wisconsin-Stout have successfully participated in joint project critique sessions using the two way audio and video - computer interactive television classroom.

This project could be expanded. Schoolcraft College has agreed to act as the initiating site. The goals of this expanded project could include the following:

1. Create a 4-way link using a bridge to connect multiple schools.

2. Experiment with partners from other areas of the world. Initially in point to point connections and then in bridge connections.

3. Initiate student project teams from multiple institutions to collaborate and develop relationships across geographic boundaries, using Internet email communications with a few sessions for critique and sharing in the Interactive classroom environment.

An Internet variation of this concept could be developed for those who do not have interactive video capabilities.

## 6. Summary

The benefits of the collaborative models articulated in this paper deal with some of the issues and challenges of educating students in a rapid changing environment. They do not however come without some problems in implementation. As technologies continue to evolve and change at incredible rates, education must look at ways to integrate the tasks of education, work experience and curriculum development. The CGT program structure is very effective in providing a dynamic curriculum model. The program has grown from 2 to 5 tracks in less than 4 years and has doubled in enrollment. Employer's hiring CGT graduates have sent positive feedback to the program. Students expect a high quality applicable individualized experience. The flexible track and modular design elective structure allows students to tailor their curriculum to meet their unique interests or needs. Some students although, require extra guidance in this task, having experienced more rigid, less fluid structures in their past. Counselors also have difficulty with the choices and rapid changes in the curriculum. Faculty find themselves taking on an advisory role. Curriculum developed around actual real project requirements and employment needs has helped identify weaknesses and needs in the college's curriculum and has affirmed some of its strengths.

Faculty development must also occur quickly in a rapidly changing world. Joint coordinated instruction can promote peer learning among faculty. The course exceeded its goals for the initial offering, and identified some issues for improvement. A more integrated articulation of learning objectives will help coordinate efforts and assessment. The benefits to the future of the college along with the cost of faculty retraining must be part of the equation, when funding is re-evaluated.

We now live in a networked environment that provides us with wonderful opportunities for connections that we did not easily have in the past. Faculty from multiple institutions have opportunities to compare outcomes and collaborate in learning projects using communications technologies, such as Interactive TV and the Web. Opportunities exist to build and maintain global relationships through experiences enhanced through technologies.

Computer graphics as visual communication can act as an integrator for interdisciplinary courses, cross-institutional experiences, and "real world" projects. Improvement of learning happens with frequent interactions and active involvement, this is required in collaborative learning environments. Collaborations are waiting to be imagined.

**Tell me and I'll forget.**
**Show me and I may not remember.**
**Involve me, and I'll understand.**
**—Native American saying**

## Acknowledgements

My thanks to the GVE'99 workshop for the opportunity to initiate dialog on the models presented in this document.

My appreciation to Jeanne Bonner, Fernon Feenstra, Sam Gooden, Mark Harris, Susan Hunt, Lisa Jacobs, Ray Krawczyk, Susan Lupo, Michael Mehall, John Miativich, Lori Navalta, Christine Rejniak, Don Ryktarsyk, Faye Schuett, Robert Smiley, Gayle Stanek, Bonnie Taylor, Brett Wilson, Gordon Wilson, Frank Wiltrakis, and all the instructors and students who shared in the collaborative efforts described in this document.

My thanks to Schoolcraft College, UW-Stout, Washtenaw Community College, GM-Delphi, Ford Motor-EEME, Crucam, ETC., Johnson Controls, Advanced Communications Corporation, and all the non-profit organizations for their willingness to experiment with new teaching and learning models.

A special thanks in particular to Stephen Wroble for his patience, insight and vision in the creation of the flexible structure used in our program.

# Enabling Educational Collaboration - a New Shared Reality

**Judith R. Brown**
Manager, Advanced Research Computing and Visualization
The University of Iowa
judy-brown@uiowa.edu

## 1.

Educational collaborations, using enhanced computer graphics, high performance networks, and virtual reality technologies, will enable new and exciting educational opportunities. Tele-immersive collaborations can provide new learning experiences and insights by joining educators and students from remote locations in a shared virtual space. Bringing together students with cultural differences and diverse perceptions of reality into this shared immersive space allows them to see and hear the same objects and sounds and to share their own unique views and perceptions, creating a new, shared reality. My vision of this shared reality for learning is based on three premises regarding research and learning.

## 2. Premise 1: Methods developed for research naturally find their way into the classroom.

The visualization lab has long been the crossroads between research and education. The research component in a university is an important facet of the educational environment and cannot be viewed as separate from "education." Research is dependent upon student assistants, both undergraduate and graduate, many of whose theses will grow out of the research. A professor's research pushes student assistants' learning experiences, and the results from the research by both faculty and students find their way into national presentations and into the classroom.

Besides the fact that research labs provide the educational environment for student assistants, the classroom needs for computer graphics will also push technological limits since student labs and classrooms need the same capabilities as research labs. Interaction is essential for collaboration, and the appropriate user interface is important, lest the computer get in the way of the students' exploration and interfere with students' natural curiosity.

## 3. Premise 2: Virtual reality, as an interactive technology, is a valuable tool for instruction.

As virtual reality technologies have recently enabled scientists to better understand their data by exploring the images of the data, so are these technologies enabling students to better understand the concepts they are learning. Although these concepts may not be new to the world, they are new to the students, and the same joy of discovery takes place.

The University of Iowa is making progress with applications on three ImmersaDesks acquired through National Science Foundation funding for educational purposes and environmental research. A year ago, there was very little software for ImmersaDesks, and our programmers had a long learning curve to get started. Now some good tools have been developed and are offered free to the user community. These tools include CAVE5D, a version of University of Wisconsin's VIS5D software for environmental visualization, and Limbo, from Unversity of Illinois at Chicago that serves as a template for developing tele-immersive applications, providing avatars, pointers, sound, and other tools for working together in a shared immersive environment.

We have ImmersaDesk applications in higher-dimensional mathematics, geography, chemistry, biochemistry, and computational fluid dynamics. Some applications have begun as research and moved to the classroom. Some entire classes come over to the ImmersaDesk, so the students can experience and share the instructor's discoveries. In other cases, graduate students are sent over to work with us.

One of the key projects being developed with the NSF funds is the *ENVISAGE* Project (*En*vironmental *Vis*ualization *a*nd *G*eographic *E*xploration.) The purpose of ENVISAGE is to allow students to experience a more complete understanding of geographical processes by having

them operate in the semi-immersive 3D visualization environment provided by the ImmersaDesk and high performance networks. Immersion, for analysis of mapped data and model output, enables students to view relationships that would otherwise remain shrouded by the limitations of widely used graphics technologies. Openshaw and Fisher, for example, assert the following:

> "It is important, therefore, to develop technologies that attempt to understand the data and develop views of the information or database world in which attempts are made by our clumsy spatial analysis tools to find patterns and relationships. Can we look inside spatial databases and walk around in them via virtual reality concepts? Currently we are so blind to many aspects of the data and the data flows being analyzed. We cope by being highly selective and subjective and in the process probably fail to see and find many of the patterns and relationships of potential interest other than those we blindly stumble over by chance!"

It is precisely this type of immersive exploratory learning that we are bringing to undergraduate education so that students will be able to more readily grasp basic concepts that have a pronounced visual component

## 4. Premise 3: Enhanced computer graphics and tools for interaction, along with high performance networks, enable a shared learning environment.

We have learned a great deal in recent months by participating in tele-immersive applications of others and are now developing our applications. Just as early research applications made their way into industry, so have they also emerged as tools for instruction.

Networked, collaborative learning can begin at a very young age, as shown by the Narrative Immersive Constructionist/Collaborative Environments (NICE) project between the Electronic Visualization Laboratory and the Interactive Computing Environments Laboratory at the University of Illinois at Chicago. In NICE, children can collaborate to plant and cultivate a virtual garden and to create stories from their interactions. See Figure 1.

Figure 1. Boy in the NICE garden. Image courtesy of the Electronic Visualization Laboratory, University of Illinois at Chicago, 1997. The NICE Project: Narrative, Immersive, Constructionist/Collaborative Environments for Learning in Virtual Reality, developed by Maria Roussos, Andrew Johnson, Jason Leigh, Craig Barnes, Chris Vasilakis, and Tom Moher.

As universities have connected to the very high performance backbone network service (vBNS), they have looked for high-profile applications to show faculty and administrators the value of these new educational and research networks. Applications that show an ability to link with other universities and explore together in shared virtual space have been popular. As part of the Pennsylvania State University Internet2 Days, we explored a model of the city of Berlin with an additional proposed building designed by a Penn State professor of architecture, as shown in Figure 2. Boston University produced a wonderful, whimsical Art World, and we appeared as avatars to play in that world during the Supercomputing 98 conference. One of the 3D models from Art World is shown in Figure 3.

What we learned from these experiences, we can now apply to our applications and build new collaborations worldwide. We recently worked with the National High Performance Computing Center of Taiwan. They were celebrating the connection of their national high performance network, TAnet, to the vBNS and other national and international high performance networks. Because the time for their planned celebration was 3 a.m. our time, they recorded our tele-immersive session earlier. For this presentation, we investigated together the pressure fields on a high speed train going through a tunnel, shown in Figure 4. We are currently

working on geographic and computational fluid dynamics applications, and we hope to develop interesting collaborative opportunities with other participants at GVE99.

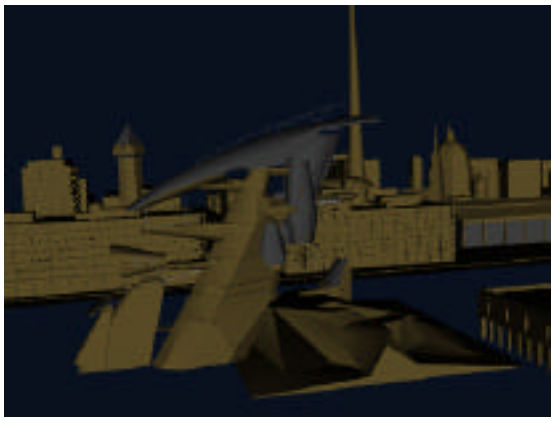


Figure 2. CAD model of Berlin with sculpture building as proposed by Lebbeus Woods. Modeled by Reggie Aviles, Pennsylvania State University (Penn State) Architecture Student.. Translated for use on the ImmersaDesk by Ray Masters, Center for Academic Computing and Affiliate Associate Professor of Architecture, and George Otto, Manager of the Center for Academic Computing Visualization Group at Penn State.

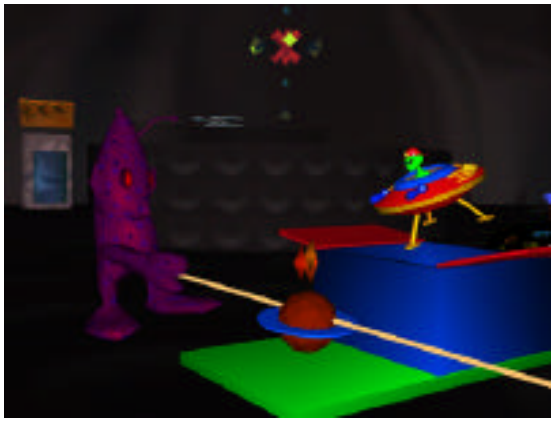


Figure 3. Image of the Martian Landscape from Art World, courtesy of the Scientific Computing and Visualization Group at Boston University. The following artists contributed to the scene: Matt Harter, Paul Haman, Tom Coffin, Jeong-Hoon Lee, and Karlo Takki.

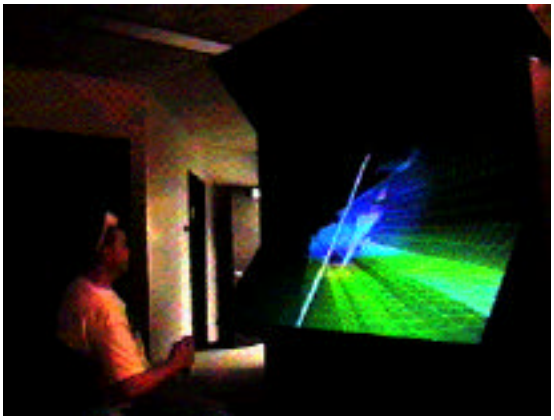


Figure 4. Image of the ImmersaDesk showing the pressure fields on a high speed train as it goes through a tunnel, the avatar of the remote collaborator, and the pointer of the University of Iowa participant. This image illustrates a tele-immersive application between the National Center for High-Performance Computing of Taiwan (NCHC) and Advanced Research Computing Services and Visualization, Information Technology Services, The University of Iowa.

There are still some issues with wide spread use of these collaborative technologies:

1. Cost - Shared immersive technology is expensive
2. Accessability - The size of the equipment requires large spaces. The interfaces are not readily accessible by persons with disabilities.
3. Distance - Time zone differences make same-time collaborations difficult. in the Taiwan example, we settled on 6 p.m. our time on a Tuesday, 8 a.m. Wednesday in Taiwan.
4. Network performance - Complicated graphics and sound require fast networks and low latency. In the Taiwan experiment, the graphics were good, but the six second time lag for the sound was disconcerting.

I look forward to meeting with other educators at GVE 99 to discuss these issues and cultural roadblocks to tele-immersive educational applications. I expect to be able to form some interesting collaborations through which we can share our tools and knowledge and make good use of the national and international high performance networks for educational advancement.

**References:**


[1] Brown, Judith R. and Steve Cunningham"Visualization in Higher Education," *Academic Computing*, March, 1990.

[2] Cunningham, Steve, Judith R. Brown, and Mike McGrath, "Visualization in Science and Engineering Education," *IEEE Tutorial: Visualization in Scientific Computing*, Gregory M. Nielson and Bruce Shriver Eds. IEEE Press, 1990.

[3] Brown, Judith R., "The multi-faceted blackboard: the future of visualization for education, *The Impact of Computer Graphics on Education: the Challenge of Interactive Learning*, S. Cunningham and R. Hubbold (Eds.), Springer-Verlag, Heidelberg, 1992.

[4] Openshaw, S. and Fisher, M.M. 1995. A framework for research on spatial analysis relevant to geo-statistical information systems in Europe. *Geographical Systems*, **2** (4):325-338.

[5] Brown, Judith R., "Computer Graphics Education and Presentation" Presentation at IGD opening, October, 1998.



This work is supported by the National Science Foundation, Division of Undergraduate Education, under the grant DUE-9750537 and by the National Science Foundation, Advanced Network Infrastructure, under the grant NR-9613871. All opinions, findings, conclusions and recommendations in any material resulting from this work are those of the principal investigators, and do not necessarily reflect the views of the National Science Foundation.

# Supporting Face to F ace Teaching

**Anne Mumford**
Audio Visual Services, Loughborough University
Loughborough, UK
a.m.mumford@lboro.ac.uk

**Abstract**

*The needs of Graphics and Visualization Education (GVE) are often more demanding of teaching facilities than are other subject areas. In the UK there is political pressure to improve the student experience through assessment of subject areas and as a result of the introduction of student fees. One area is the need to improve the teaching facilities and methods of delivery. GVE may be considered to be very specialist and not a key driver for facility improvement. The requirements, when met, can however provide improvement for all. There is also a need for widening the use of the techniques. Despite many awareness raising exercises, the techniques are still seen as being specialists and there is little penetration in some subject areas.*
**Keywords:** *teaching rooms, teaching methods, online learning, student experience.*

## 1. Introduction

When delivering Graphics and Visualization Education there is a need for the provision of appropriate teaching space and/or technology for flexible learning by students at a time and place of their choosing.

This paper focuses on the needs of the teacher and the student for face-to-face teaching and the pressures on Higher Education Institutions for the provision of suitable teaching space. The perspective given in the paper is a UK one and is based on work undertaken by the Advisory Group On Computer Graphics (AGOCG [1]) and the funders of that initiative, the Joint Information Systems Committee of the Higher Education Funding Councils (JISC [2]). Some of this work is now being taken forward by the UK's Standing Conference for Heads of Media Services [3].

The paper concludes by recognising the benefits which might accrue to many teachers and students if the requirements of GVE are met within teaching space provision,

## 2. Teaching Space Provision

Provision of teaching space in UK Higher Education Institutions tends to be a mix of centrally owned rooms and departmentally owned rooms with support provided by the centre or by the department. Equipment in rooms may be installed or may be supplied as required for particular lectures, normally through a booking system. The result of centrally owned space and insufficient funding is that use of the space is efficient but that the facilities available and the support provided tends to be at a relatively low level across all rooms. The facilities are often inconsistent. Booking of rooms is rarely based on facilities required by a teacher but simply on student numbers and, perhaps, special access requirements, e.g. wheelchair access. The AGOCG case studies showed virtually no links between room bookings and the requirements of the teacher for facilities to support their preferred teaching method. The result is a very slow move towards the use computers for face to face teaching. Lecturers tend to prefer to use low levels of technology e.g. OHPs, slides and writing boards as these can be guaranteed.

## 3. Political Pressures

There are a number of political pressures which make this an interesting time in UK Higher Education.

### 3.1. Teaching vs Research

The UK Higher Education Institutions are all subject to external reviews by Government. These cover research, through the Research Assessment Exercise and teaching and learning through the External Subject Reviews (formerly the Teaching Quality Assessment).

The pressure to deliver results in both areas is common to all institutions (about 185 in the UK), though aims and aspirations vary. The pull on resources as institutions strive to improve on research results can have an impact on the funding for teaching resources and support.

### 3.2. External Subject Reviews

The External Subject Reviews focus on the student experience and are an intensive assessment of the delivery of teaching and the learning within a subject in a University. The provision of appropriate teaching space and facilities is part of this review.

### 3.3. Flexible Learning

The current government strategy to develop life long learning and to expand the Higher Education sector means that a greater range of teaching and learning methods need to be put in place if this is to be achieved. The need to provide more flexible learning methods allowing students to learn at a time and place of their own choosing for some or perhaps most of their studies will be embraced as a concept in different ways by different institutions. Many will adopt new methods to some degree.

### 3.4. Student Fees

A further pressure to focus on the student experience comes from the very recent introduction of student fees to be paid by all in UK Higher Education.

## 4. The Perspective of the Teacher

The teacher needs to feel that they have a guaranteed availability of facilities before using different technology for their teaching. The lack of online access and data projection in teaching rooms results in caution and a lack of willingness to use new technology in teaching face to face situations. There is also a lack of confidence in many teachers which needs to be overcome with training and support. Teachers need to think about fall-back procedures if things go wrong and to be supported to develop these.

## 5. The Perspective of the Student

Students are expecting to be able to access resources online. Lecture notes and support materials should be made available online to support the wide range of students enjoying higher education today. Such students may be working their way through University (common in the US but a new feature of UK Higher Education) who may not be able to attend all lectures. The availability of materials online is also critical in supporting the education of some students with disabilities who may only be able to access materials in a way modified to support their particular disability.

Students also expect to be able to see the materials and many rooms are unfortunately not suitable for data projection. Some examples and good practice guidelines are given in the AGOCG case studies.

## 6. Teaching Space fit for GVE

Teaching space that is “fit for purpose” is the key to a successful experience for both the student and the teacher. The requirements for GVE are towards the top of the range of requirements. Online computer access, data projection, good resolution display, suitable blackout, dimming lights are all needed. Such requirements would benefit all. Yet, these facilities are far from common in UK Higher Education across the large numbers of teaching rooms.

## 7. Widening the Use of GVE and Promoting the Benefits

At the same time, there is a lack of both awareness and takeup of graphics and visualization tools in many disciplines. Many disciplines could benefit from the use of these tools in both research and in teaching. In many disciplines the lack of takeup has been a result of a number of factors including lack of inter disciplinary working and a lack of equipment funding.

The social sciences have been identified by AGOCG as disciplines which could benefit from the takeup of visualization techniques, particularly in regard to the analysis of large and complex datasets and other data mining activities. Workshop output and case studies can be seen on the AGOCG Web site.

There is a major contribution still to be made by those who have been involved in Graphics and Visualization techniques in both research and teaching for some years. The techniques are applicable to many subject areas which have thus far still been little touched by awareness of the potential of these techniques. The use of such techniques in a wider range of subjects and the demand that would inevitably cause for improved teaching facilities and the development of new teaching methods in areas beyond Computer Science would benefit many.

## References


[1] Advisory Group On Computer Graphics, http://www.agocg.ac.uk (including various reports relating to Lecture Room Services at: http://www.agocg.ac.uk/reports/mmedia/casestdy/casestud.htm

[2] Lecture Room Services, Workshop Report http://www.jisc.ac.uk/pub98/assist3.html

[3] Standing Conference For Heads of Media Services http://pari.mmu.ac.uk/msh/welcome.htm

# Web Based Multi-User Interactive Graphics Worlds for Education

**G. Scott Owen**
Department of Computer Science
Georgia State University
Atlanta, Georgia, USA
owen@siggraph.org

**Abstract**

*Several technologies are currently being developed that will have a huge impact on the future of education. Advances in computational power and graphics capability will give the future home PC the capabilities of very high-end workstations of the present. Advances in connectivity will allow everybody to be connected at all times with a very high bandwidth. Advances in computer graphics software technology, combined with advances in other fields, such as simulation and artificial life, will allow us to create realistic worlds for students to investigate. These worlds will be Multi-User Shared Environments (MUSE), i.e., students will be able to interact with instructors, other students, and intelligent agents in these worlds.*

*In this paper, I will briefly discuss the background issues, vision for the future with some specific examples, and our current efforts in this area.*

## 1. Background Issues

### 1.1. Technology

Current home computers in the $2,000 price range have a substantial amount of graphics and processing power, e.g., a 400 Mhz P II, 128 Mbytes Ram, Open GL accelerated graphics card with 8Mbytes of RAM, a 12 Gbyte disk, and a 17-inch monitor. This is rapidly increasing and within a year this class of machine will have a 700Mhz processor with enhanced instructions for 3D Graphics and MPEG-2 decoding, e.g., the Intel Katmai cpu, 256 Mbytes RAM, a 20 Gbyte hard disk drive, and a faster graphics card.

If we assume that Moore's law (computers double in power approximately every eighteen months) will continue to hold, then a conservative prediction for the home PC (with a cost of under $2,000 U.S.) of the year 2009 will be as follows:

- Processing power about one hundred times that of a Pentium 400 mhz cpu
- Main memory of about one to two gigabytes
- Disk space of about one terabyte

People will be able to carry these computers around and use portable displays such as eyeglasses, i.e., the computing will be ubiquitous. Another technology advance is in improved Internet bandwidth into the home. Currently most people connect via 56 K modems, but more other technologies are rapidly becoming available. For example, in many cities either Cable Modems or telephone based DSL lines are becoming available for about $50/month. These allow for speeds up to 1.5 mbits/sec or about as fast as a T-1 line.

By 2007 this will be of the order of fast ethernet (100 mbytes/sec) and will be wireless. The introduction of VRML 97 and Java 3D, combined with the rapid advances being made in the simulation of animals (artificial life) and of humans are allowing us to approach the ideal of the Star Trek Holadeck. We will be able to create worlds for students to inhabit such that they can change the parameters or else just live with other creatures.

### 1.2. Trends in Education

Three major trends in education are:

1. The increasing emphasis on Distance Learning, especially Web based learning
2. Life-long education
3. The increased use of technology in education

This means that web based learning technologies and interactions are becoming increasingly important. Students, especially older students, are less and less interested in coming to a central campus to listen to traditional lectures. They want to be able to learn on their own time and at home. However, there is still a role for the professor-student interaction, either one on one or with one professor and a few students, although this may become a virtual interaction.

## 2. Vision for the Future

Imagine an Art History class where the instructor is holding office hours in the class world, which is structured as a virtual Art museum. A student approaches the instructor avatar and states that she has a question on a particular painting. The instructor and student walk to the painting and the instructor answers the student's questions. The instructor goes into another room and finds three students discussing a particular sculpture. The instructor joins in the discussion and walks the students around the sculpture, pointing out different aspects of the work. Note that the participants are at their homes and are interacting via computer and conversing using microphones.

If a student logs into the art world and the instructor is not there, then the student can ask questions of her virtual companion. The companion, one for each student, is an intelligent agent that knows some information about the art works in the museum. But, perhaps more importantly, each companion is privy to the sessions between the instructor and the other students, and so can report about new interesting insights that the instructor has given to other students.

Imagine a biology class where the instructor leads a group of students on a tour of a cell. One of the students asks about the DNA replication process and the instructor leads the group into another world where this is occurring. The instructor discusses what is happening with the students while they observe the process. The instructor and/or students are able to modify the process by changing certain kinetic parameters and observe the changes.

Imagine a zoology class where the students can enter a virtual ocean and observe the interaction of different species. The students might themselves be different types of aquatic creatures, one might be a shark and others might be a dolphin, swordfish, whale, etc. They can design a new species of predator, set up its parameters and observe how it changes the inherent ecology.

All of these scenarios involve the use of a 3D Graphical MUSE. In some of the environments, e.g., the Art museum, the primary interaction would be between students and their virtual companion and the professor or other students. In other environments, e.g., the virtual ocean, there are other entities that interact with the students, i.e., the students might not know which of the other creatures were other students and which were virtual creatures that were programmed to behave in a realistic fashion.

## 3. Current Work

While the above scenarios are futuristic, we have begun to work to develop aspects of them. We are currently working with Art History faculty from the School of Art and Design at Georgia State University to design and implement a prototype of a MUSE Virtual Art Museum. Our goal is to create a generic museum that can be populated with different sets of art works depending upon the particular course. For example, a course covering Renaissance Art would require one set of works while a course focusing on the Twentieth Century would require a different set.

We are using the Sony Community Place server for this project and currently have a demo version that allows three simultaneous users (we are negotiating for a system that will allow up to fifty simultaneous users). With this system, each student can be given a CD-ROM with that contains the Virtual Art Museum and a set of avatars. Each student will be assigned to be a particular avatar. The student can log into the server, and interact with the other students who are logged in. Currently this interaction can either be via typed text or microphone. An advantage to this system is that the bandwidth requirements are kept low. Each user has the basic world plus avatars on their own machine so all the system needs to transmit is the updated viewpoint of their own avatar, the updated positions of the other avatars, and any communication (via text or audio). Therefore, students at home with only modem connections to the Internet can use it.

This project has just begun and we will report on our progress at the GVE 99 workshop. Our initial plans are to implement the Virtual Art Museum, test it out with GSU students, and then to begin implementing some aspects of the intelligent agents.

## Conclusion

The convergence of rapidly improving technology, especially the improving graphics capabilities of inexpensive computers, plus the increased need for non-traditional distance learning has given rise to the possibility of using web based interactive 3D graphics worlds to radically improve education.

These worlds will be multi-user so they will be able to facilitate student-student and student-instructor interaction at a distance. We have begun work on the design and implementation of one such world, to teach Art History courses.

# Using Web-Based Computer Graphics to Teach Surgery

**Ken Brodlie**
**Nuha El-Khalili**
**Ying Li**
**School of Computer Studies**
**University of Leeds**

*Position Paper for GVE99, Coimbra, Portugal*

---

## Surgical Training

Surgical training is a long and expensive process. Much of it is done by mentoring, where an experienced surgeon trains a junior colleague - either with the help of models, or on real patients. This is demanding on the time of experienced surgeons, and potentially hazardous for patients. Recently there has been considerable interest in the use of virtual reality approaches to surgical training: these have the potential to allow trainees to work at their own speed, in a fairly realistic way and in a safe environment.

## Virtual Environments for Surgical Training

However typical virtual environments for surgical training are:

- dedicated installations using custom-built apparatus, which can only be used in one location
- expensive - often based on Silicon Graphics high-end products
- designed for individual rather than large class teaching
- specific to one type of surgery
- in prototype form as a research exercise rather than in everyday use for training
- lacking in serious evaluation as to the positive transfer of training skills

We have been investigating a different approach. Can we create simple but effective surgical training applications that can be delivered over the Web? These will be VRML worlds which will execute within a Web browser. If we can do this, then the advantages are considerable:

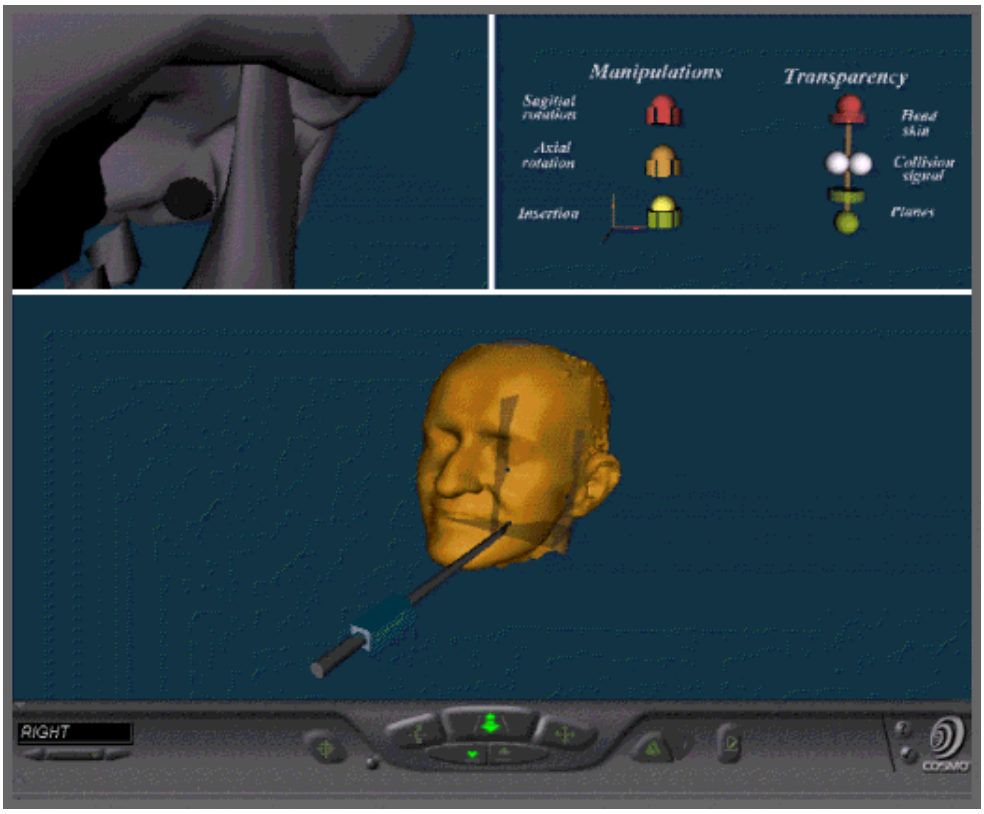
- **accessibility**: the application can be run from anywhere in the world
- **low-cost**: the application can be run from a PC without specialist software (simply a VRML plug-in to a Web browser)
- **distributed**: if powerful computation is required, this could be provided on a remote server, shared between all users
- **class size**: large numbers of trainees can be handled
- **generality**: one could imagine a family of applications, with a consistent methodology, being made available for a range of surgical procedures

Of course - there is a price to pay, namely realism. Can we provide sufficient realism in a Web-based simulation in order to provide useful training?

## Applications

We have been collaborating with three groups of surgeons and radiologists at Leeds, in order to investigate the potential of Web-based training. All three applications - while coming from quite different branches of surgery (vascular, neurosurgery, orthopaedic) - require the surgeon to manipulate a tool through the human body. Our approach likewise is common: we convert data from a body scanner into a geometric representation of the part of the body involved, storing this geometry as VRML primitives; we create a geometric model of the surgical tool, again in VRML; and finally we provide some virtual interaction where the user can manipulate the tool through the body. Two of these applications are described below.

## Neurosurgery




Neurosurgery is a very demanding area of surgery where complex operations are required. These are often composed of a sequence of simpler procedures, with critical decision-taking required in putting the sequence together. If things go wrong, they go wrong quickly and disastrously. This makes it an excellent application area for web-based virtual environment training: the simple procedures can be rehearsed safely, efficiently and inexpensively.

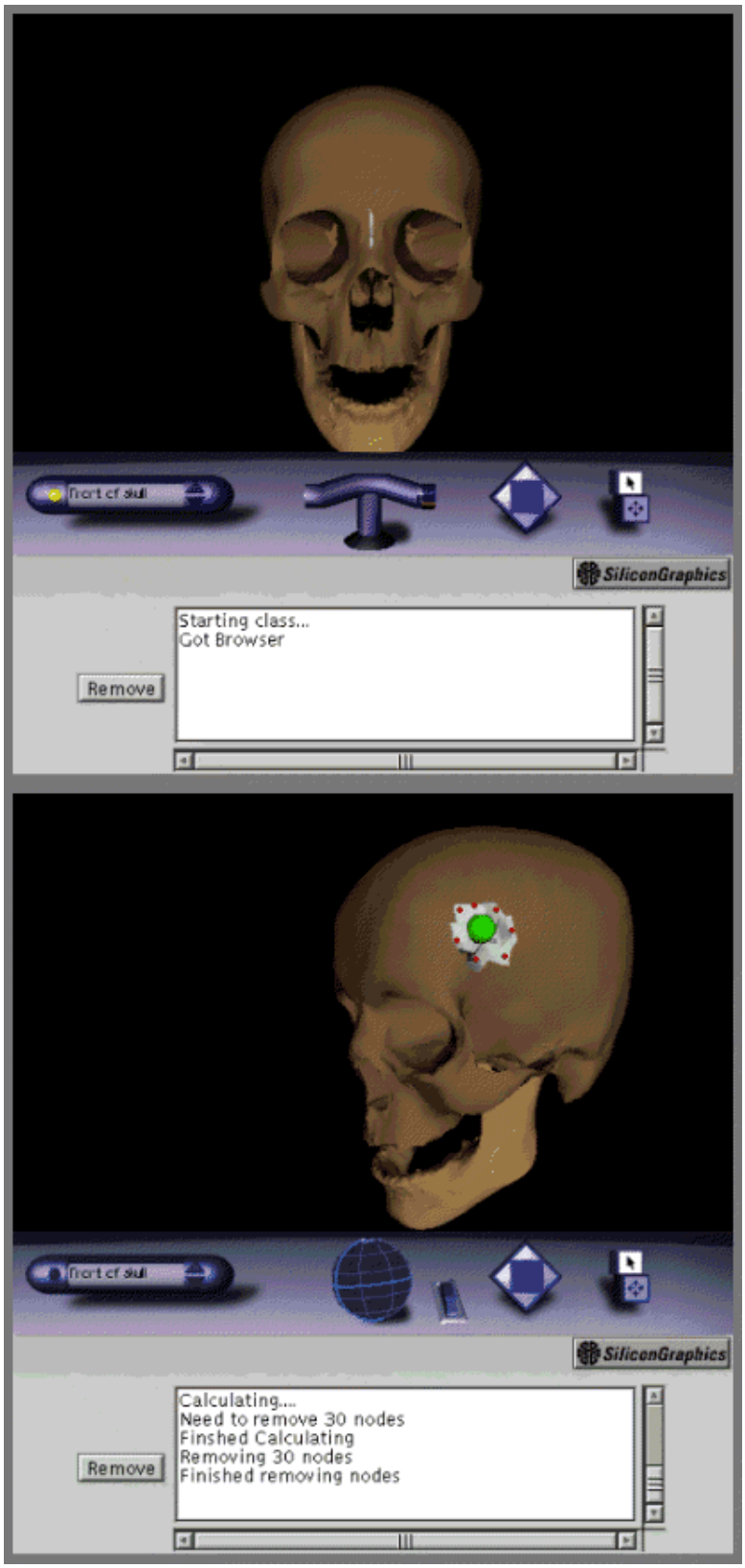
We have developed one simulator for the treatment of intractable facial pain (which if untreated can be so unbearable as to result in suicide). An accepted treatment is Percutaneous Rhizotomy: it is a safe

procedure when performed by experienced hands. Surgeons direct a needle into the medial portion of the foramen ovale, to partially destroy the sensory root for relieving the facial pain. In practice, surgeons mark three points on the patient's face: from these points, a surgeon envisages the coronal and sagittal planes through two of the points, and directs the needle from the third point to the intersection of these two planes.

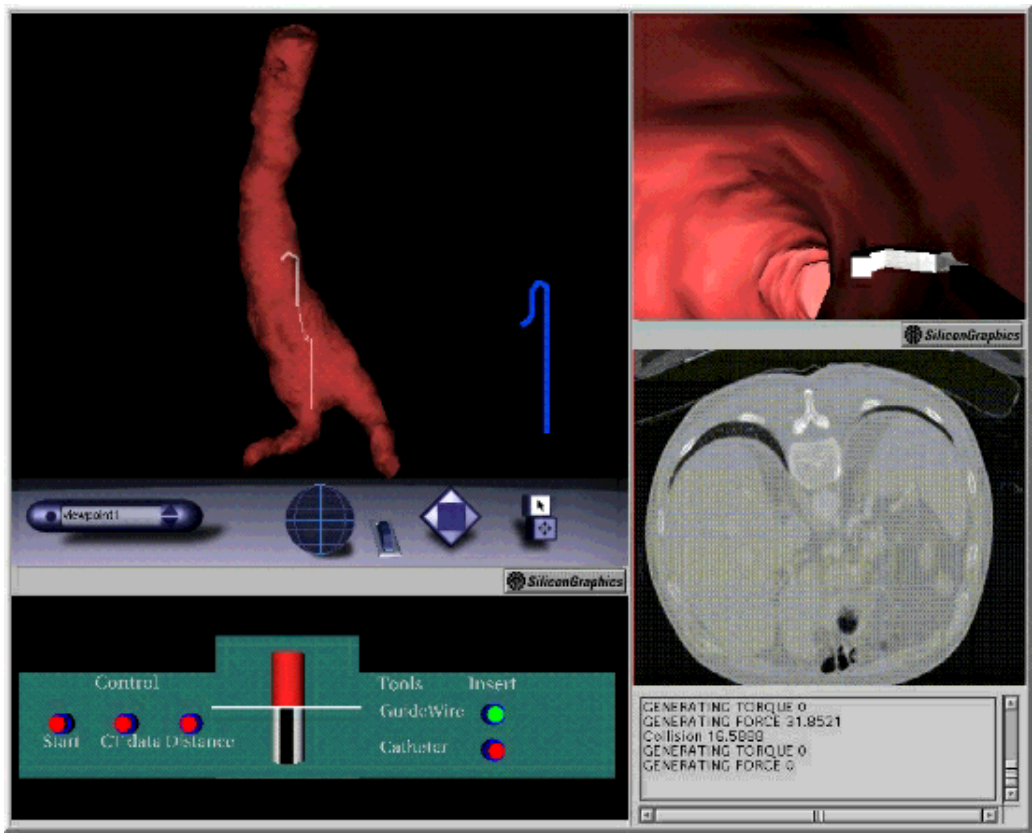
The simulator is shown above. The application is a linked combination of VRML worlds. The skull is modelled in terms of VRML (a large IndexedFaceSet)
and the needle too is a VRML object. The application allows the trainee to mark points on the skull, and insert the needle through the skin, manoeuvring it through a hole in the skull. The coronal and sagittal planes can be highlighted if the trainee needs some guidance (see the picture above). We have used three VRML worlds. One world acts as the control panel, with manipulators that simulate hand operations of the surgeon. The other two give views of the patient: the large view in the lower part of the application is an 'exocentric' view from outside; the top-left gives a more 'egocentric' view, as seen from the tip of the needle. Collision detection is important - but a problem in VRML which supports only viewer-object collision, not inter-object collision which is what is required here. However, we can use the egocentric view to capture collisions, and transmit collision information to all the linked worlds. The linking of the VRML worlds is carried out using the External Authoring Interface (EAI).

We have developed another simulator for craniotomy where part of the skull is removed so that the neurosurgeon can gain access to a brain tumour. In this procedure, the neurosurgeon drills a number of holes in the skull around the area to be removed, and then using a small saw, makes cuts between the holes.

This simulator is shown below. The application is a VRML world with a model of a human skull as a large IndexedFaceSet, with a brain model inside. The trainee can make the skull transparent in order to see the tumour. In the training exercise, the trainee marks points corresponding to the holes to be removed. The simulation then removes the triangles within the area to be cut out, so that the trainee can check the success of the virtual operation.


front of skull
SiliconGraphics
Starting class...
Got Browser
Remove
front of skull
SiliconGraphics
Calculating....
Need to remove 30 nodes
Finshed Calculating
Removing 30 nodes
Finished removing nodes
Remove

**Vascular Surgery**




Interventional radiology is another complex area where web-based virtual reality has the potential to become an important training aid. In our work we have been looking at a simulation of treatment of abdominal aortic aneurysms. This is a swelling of the aorta; if untreated, the aneurysm may burst causing a critical condition. The treatment involves placing a stent inside the artery to shield it from the anuerysm. This is achieved by inserting a series of guidewires and catheters from an incision in the groin. These instruments are manipulated through the femoral artery, and into the aorta. It is a highly skilled operation, requiring some ten years' training.

The simulation is shown above. It is altogether more complex than the neurosurgical applications because we need to model the deformation of the catheter or guidewire as it collides with the walls of the artery. Again we have a VRML world which contains a model of the aorta and the surgical instruments. Top left shows the exocentric, overall view, top right the egocentric view from the catheter. The lower left area shows an interactor that will push/pull, or twist, the catheter. The lower right shows a CT scan corresponding to a 2D slice through the aorta which can be selected by the trainee.

The system architecture for this application is quite complex. Deformable modelling requires compute-intensive calculation - which may be beyond the capability of the client machine. Thus this application uses a server process to execute the physically-based modelling code which computes the deformation of catheter and guidewire. It also tests for collision between the catheter and artery wall. Communication between client and server needs to be as efficient as possible, since the combined application needs to operate in real-time.

## Critical Issues - Interaction, Visual Realism and Haptic Feedback

This is very much work in progress but we now have three years experience and are able to identify the critical issues.

In our approach we rely only on a mouse as physical input device. Mouse movements are mapped to manipulation of virtual input devices in the VRML world - in the neurosurgery application, to simple, generic devices; in the vascular application to a virtual representation of the real catheter. The issue for training is to what extent this can replace the real input device.

We also need to understand the extent to which visual realism is important. In the vascular case, the radiologist has only a mental 3D model of the vascular structure - there is no camera on the end of the catheter. This mental model is constructed from intermittent X-ray images, and from feel - plus years of experience.

Finally, in both applications, haptic feedback is important in the real-life simulation. An expert radiologist can operate almost by touch alone. With simple Web-based tools we cannot provide force feedback (although force feedback mice are soon to be available). However in the vascular case, we have been able to simulate force feedback by colouring the virtual catheter.

## Conclusions

We are studying whether simple Web-based computer graphics can help in the training of surgeons and radiologists. We can never hope to achieve the realism of expensive, dedicated VR simulators, but we believe we can create useful tools nonetheless. Through VRML we gain a solution that is both portable and scalable. Moreover we can take advantage of new developments in VRML - such as proposed extensions to multi-user working that could allow collaboration between surgeon and trainee.

The major question to be resolved is the degree of realism that *needs* to be achieved. Perhaps too much realism can be a bad thing! In the area of rendering, there has been a powerful movement in favour of non-photorealistic rendering, where an expressionist style of rendering is used to create a more interesting final result. There may be a parallel in surgical simulation, where the key parameters in the simulation are extracted and presented to the trainee in an ’expressive’ style. We would hope to discuss this at the workshop.

## Acknowledgements

We are very grateful to the surgeons and radiologiss who have given generously of their time to help us - especially David Kessel (St James University Hospital) and Nick Phillips (Leeds General Infirmary).

*June 1999*

# Visualisation in teaching-learning mathematics: the role of the computer

**G. Chiappini** and **R.M. Bottino**
Consiglio Nazionale delle Ricerche
Istituto per la Matematica Applicata - Genova - Italy
{chiappini, bottino}@ima.ge.cnr.it

**Abstract**

*This paper is centred on the study of the role that information visualisation can have in mathematics learning. A theoretical analysis is worked out in order to give account of the complex phenomena that take place in the classroom in order to allow students to construct a mental image of the mathematical object involved in the activity. This theoretical analysis is the reference to study the different didactical functions that information visualisation can develop in learning process. This different functions are discussed making reference to the role that Cabri Gèométre can have in the teaching/learning processes of Euclidean geometry.*
**Keywords:** *Mathematics learning, Information visualisation, Visual imagery, mathematical discourse .*

## 1. Introduction

We use the term visualisation to refer to the complex phenomena of visual imagery that plays a central role in all meaning and understanding as well as in all reasoning. About the nature of images that the mind forms following an external stimulus, many studies, also on contrasting lines, were developed in the last 20 years. Here we are interested in the phenomena of visual imagery that take place in maths learning and in particular in the dialectic that develops between dynamic external visual representations mediated by the technology (information visualisation) and visual imagery.
Speaking of external representations we mainly refer to two or three-dimensional representations of some aspects of a mathematical structure. Such representations may be static or dynamic as in the case of representations mediated by the computer. Dreyfus in [2] pointed out that the dialectic between external visual representation and visual imagery implies two mappings: from the mathematical structure to the visual representation and from the visual representation to the mental image. While the first mapping can be subject to mathematical analysis (epistemological analysis of the knowledge embedded in the structure of the visual representation), the second one is much more difficult to analyse, since there is no direct access to mental images. So this latter mapping can only be postulated on the basis of interpretations of how the external representations (graphics, signs, tables, drawings,…) are produced and used in the communicative context of the teaching/learning process and on the basis of the justifications that everyone gives of this way of use.
The two mappings evidenced by Dreyfus are important while studying the conditions under which computer-supported visualisation is useful for mathematics learning.
This is true both for the design of new systems for mathematics education and for the analysis of how systems already available on the market can be used in the classroom.
In this work these two mappings will be discussed, analysing the roles assumed by information visualisation [7] in the teaching/learning processes of mathematics. This analysis will be carried out on the basis of the experience we have developed in the design, implementation and experimental evaluation of computer-based visualisation systems for mathematics learning [1].

## 2. Mathematical objects, visual representation, mental image

For the aims of our work we consider Mathematics as a domain of knowledge that is concerned with "mathematical objects", that is to say with objects with certain specific properties. Mathematical objects are abstract objects; indeed mathematics objects are not amenable to any concrete imagination or manipulation; they are immaterial, not tangible and accessible only to our thinking
In mathematics learning, differently from the physical concrete world, the learning object cannot be shown in an obstensive way, can be only conjured up by means of the use of external representations. There is not the possibility of directly accessing that "thing" that we can suppose to be the meaning of the representations.
Mathematical concepts such as numbers, functions, vectors, (which are not objects in a usual manner, but which embody relationships) are not directly accessible through everyday experience nor within intuitive perception, as for instance real or physical objects are, but they have to be represented by signs or symbols.

This is true also in the case of Euclidean geometrical learning (as we will see in section 4), where the perception involved in managing external representations (drawings) can be also an obstacle for the construction of a mental image, theoretically founded, of the correspondent geometrical object (figure).

Representations and symbols of mathematics establish a semiotic system which is of fundamental importance for any mathematical activity.

According to this epistemological position mathematical knowledge is not simply a ready made product that can be directly introduced into processes of teaching and learning. The new mathematical knowledge will only be actively constructed, in social interaction, by the student in his or her learning process within an activity.

For example, in the approach to rational numbers the relationship that takes place between representation and the mathematical object which it refers to (rational numbers), is very complex. The discourse about rational numbers suggests that these abstract objects are unique entities; on the other hand we have various representations for the rational numbers: the common fraction symbol, points on the number line, materialisation of different kinds, classes of pairs of integers; these material objects of representations do not enjoy all the properties we attribute to rational numbers [3].

It is not easy for students to understand that 2/3 and 4/6 are two different fractions which are quivalent since they refer to the same object, that is to say to the same rational number. Showing that 2/3 is obtained from 4/6 dividing the numerator and the denominator by 2 gives us a method that permits us to justify the conditions under which the two fractions are equivalent but we cannot work out an obstensive test of such an object and of its properties. We do not have imaginative access to anything which we could consider an image of the rational number represented by 2/3 or 4/6.

The image of this mathematical object emerges in the dialectic between understanding and expression relating to the mathematical activity in which the participants are involved. This image entails an orientation to negotiations with oneself about meaning, something that is outside the experience of school students [6].

The image of the mathematical object is strictly linked to the image of the mathematical activity that is negotiated by the participants in school practice. It is conditioned by social interaction; on one hand the structure of the external representations provide meaning for the mathematical discourse about the mathematical object involved in the activity and on the other hand the mathematical discourse contributes to structure an image of the mathematical activity, that is to say it contributes to structure an image of the object of this activity, that is a mathematical object. It is important to observe that mathematical discourse that emerges in social interaction within the activity is qualitatively different from any representation.

Understanding what a mathematical object is depends crucially on a very specific way of viewing and treating representations and their related mental images within a mathematical discourse.

In other words, taking into account school practice in which students and teacher are involved, the relationships between external visual representation, mathematical discourse and mental images can be sketched in this way.

Within an activity, the structure of external representations mediate the possibility to develop a meaningful mathematical discourse about the properties of the mathematical object starting from how the properties of the external representation are used in practice.

Mathematical discourse (which is always mediated by the reference to the mathematical object) allows a metaphoric use of the concrete meaning developed by working with external representations within the structure of an activity. It allows us to associate interpretations to the external representation used in the activity which can be justified on the basis of actions and goals as actually generated within an activity (in relation to some problem or task) and on the basis of cultural and historical considerations on the value and properties of the mathematical object involved in the activity. In this way the mathematical discourse contributes to transform an external representation into a mental image of a mathematical object, that is to say into an image of the abstract object which transcends the structure and the characteristics of the external representation.

From a psychological point of view the dialectic between external representation and mathematical discourse mediates the possibility for the subject to control, on the basis of external stimuli (external representations), the accordance of the mental image of the mathematical object involved in the activity with the shared ideas of the society.

Hence according to [3] grasping the properties of a mathematical object is always the result of interplay between visual and diagrammatic aspects involved in managing external representations and prepositional aspects of the discourse with respect to the mathematical object involved in the activity.

## 3. Information visualisation and mathematics learning

The design or the use of a computer-based system for mathematics learning requires careful consideration of

the conditions under which the characteristics of form and interactivity of a system can develop, within an activity, a dynamic relationship between external representation and mental image which is effective for learning.

According to [5] we use the term information visualisation to intend the use of computer supported, interactive, visual representation of abstract data to amplify cognition. This definition is particularly appropriate when it refers to the teaching and learning processes of mathematics. As previously pointed out, mathematical objects are not amenable to any visual perception or manipulation. The traditional approach to mathematics knowledge is a symbolic re-constructive approach and it is developed inside the interaction between the student and the teacher, usually according to a transmissive teaching strategy. In this approach students have few opportunities of exploring the functionality of the symbolic representation at hand and of reflecting on the properties and characteristics of its structure. The prescriptive character of the mathematical discourse that emerges in the classroom focuses the attention on the system of rules attached to the symbolic representation at hand and leaves in the background the construction of a mental image of the mathematical object involved in the activity. Information visualisation can allow the student to access mathematical knowledge integrating the symbolic re-constructive approach with a motor-perceptive one. This latter approach involves actions and perceptions and produces learning based on doing, touching, moving and seeing. As already pointed out by Kaput [4[, a visual representation system based on direct manipulation interface is an interactive medium which responds to the user's action and which offers the possibility to create new notational systems or to introduce a new dimension, movement, within traditional ones.

Information visualisation offers the concrete possibility to implement better and more easily classical visual representation mathematics formats (graphs, drawings, tables, etc.) but also to enrich them with additional features such as movement and interactivity and to integrate in the same environment (or in interconnected environments) multiple formats of representations.

Within the context of use of the system, information visualisation can support different didactical functions which are crucial in the teaching and learning processes of mathematics.

Information visualisation can offer ways which allow students to explore the knowledge domain, embedded in the system (exploratory function). Systems that present these features have been demonstrated to be very effective for learning mathematics: they are defined as microworld based systems. The pedagogical objective of microworlds is to offer students a space in which they can use visualisation supported by the computer in order to explore and manage freely an environment designed to address the construction of some mathematical knowledge. Other systems that present these features are systems for simulation.

Another important didactical function of information visualisation is to offer expressive ways to allow students to externalise their own knowledge of a domain (expressive function). This function is present when representative tools, which student can easily control both on an operative and a conceptual level, are made available within a system.

A third didactical function regards the possibility to offer ways to validate the solution strategy involved in the problem at hand (validation function). In this case information visualisation makes available verification methods that are checked on a perceptive level and are able to test the strategy developed.

Systems that present this didactical function make available tools that allow students to justify their solution procedures on the basis of verification methods that are internal to the system.

A fourth didactical function is connected to the possibility to have available tools that allow the student to store the solution steps performed in order to successively re-visualise them in sequence. In this way it is possible to re-construct the story of the solution process performed. Systems which make available this kind of tools make possible to objectify, de-personalise and de-contextualize a solution procedure in order to successively transform it into an object of discussion in the social context of the class (supporting mathematical discourse function). This discussion can have different aims such as the comparison of strategies, the analysis of the properties (and of the their relationships) involved in the procedures, the re-interpretation of such properties in relation to a theory of reference. According to the nature of the knowledge domain involved in learning, the different described didactical functions of information visualisation can be pursued in different ways. In the following, we will try to better explicit how information visualisation can affect learning mathematics, analysing these different functions in a specific domain of knowledge supported by computer.

## 4. Information visualisation and learning Euclidean geometry

Geometry is a knowledge domain in which visualisation within a microworlds has brought to impressive progress with the development of the concept of “dynamic geometry” exemplified by systems such, for example, Cabri Gèométre. The researches worked out in relation to the use of this system put in evidence that its mediation can modify

the way to enter into contact with geometrical knowledge. The changes brought into the didactical activity by the use of Cabri can be debated in terms of:

* New status assumed by the geometrical construction as mediated by the system;
* New possibilities of development of a mental image of a geometrical construction, theorem and proof, as mediated by the system.

In this framework we note that the first aspect can be put into relationship with the mapping (as reported in section 1) from the mathematical structure to the visual representation while the second aspect concerns the mapping from the visual representation to the mental image of geometrical construction, theorem and proof.

The relationship between geometrical construction and visual representation available with Cabri is particularly interesting.

By means of Cabri it is possible to perform every types of geometrical construction, within Euclidean geometry. Performing a geometrical construction by means of Cabri does not implicate aspects of measure, but it is strictly connected with the deep structure of Euclidean geometry. The fundamental element of every construction is the point. Some objects are directly defined in terms of points (i.e. a straight line is defined by means of two points). Other objects are defined as functions of some other objects already constructed (i.e. a straight line passing for a point perpendicular to a given line) [7].

Cabri makes available primitives deeply linked with the axioms of the Euclidean geometry but also a new visual direct manipulation opportunity: to drag the variable elements of the geometrical construction on the screen. In this way students can observe which properties are preserved when the construction is modified with the drag action. The movement produced by the drag action is a way to externalise the set of the relationships that define a figure.

These representation tools of Cabri can be used in both exploratory and expressive modes.

In the exploratory modes, the use of these representation tools containing domain models permit the student to examine the consequences of different model, some of which fit and other of which conflict with the student own ideas [9].

In the expressive mode student can examine their own knowledge of geometrical construction by being encouraged to experiment with their own theorizing [9].

Mariotti [8] highlights that the drag action of Cabri has a crucial importance both from a didactical and an epistemological point of view. By means of this action students can validate the constructions performed; the construction task is solved when the drawing get through the drag test, that is when the properties of the figure, under the drag action, are preserved (validation function).

Moreover the use of this new dimension, the movement, in the body of Euclidean geometry, makes available new visual tools to overcome epistemological obstacles connected with the relationship between drawing and figure in the domain of Euclidean geometry.

As stressed by Laborde [7], the drawing refers to a material entity with which one has a perceptive relation, figures refer to theoretical objects, that is to abstract geometrical object that are described by texts which define them.

In traditional didactical practice we note that several difficulties emerge in the geometrical construction tasks. Students work on the material drawing while the teacher expects they work on the figure or on the description of the figure. In other cases students do not consider that a geometrical construction task can involve the use of geometrical properties; rather they intend it as the request to produce a material visually correct drawing.

According to this framework, the opportunity of dragging the variable elements of a construction is an important tool for favouring the evolution of students' mental images, which are related, with the notion of geometrical figure and with the task itself of geometrical construction (exploratory and expressive function)

As an example the way in which student’s approach geometrical constructions with Cabri can be considered. At the beginning students work on the drawing on the basis of perceptual and figural stimuli received from the screen instead of to perform a conceptual control of the perception, exploiting the action possibilities offered by the system. Only when they realise that their empirical strategy does not get through the dragging test, students, in general, begin to approach the task by using constructive methods which allow to maintain the properties of the figure when the drawing is dragged on the screen. It is in this moment that the relationship between figure and drawing is going to be modified and the student elaborates a new mental image of geometrical constructions.

The evolution of the mental image of geometrical constructions is crucial in geometry. As the matter of fact, each geometrical construction incorporates a theoretical meaning, which goes beyond the practical task of its construction. Mariotti in [8] shows that there is always a correspondence between the specific tools and rules used for the construction and a set of axioms which structure a part of a theory. Inside this theory a valid construction can always be put in correspondence with a theorem

Geometrical learning needs to move from context-based methods of validation (e.g. the dragging action

of Cabri) to theory-based justifications (deductive method). This passage is neither simple nor automatic. It requires an educational approach able to support the acquisition of the deductive method. In this framework it is important specify the role that Cabri can assume in supporting this didactical approach.

In Cabri, for example, an important command is available at this regard: the "History command". This command permits to visualise step by step on the screen all the actions performed by the user to obtain a given geometrical construction. Such a command can develop some important function from a didactical point of view.

As put in evidence by Mariotti [8], the History command allows to develop interpretations and anticipation acts which can favour the passage from justifications based on physical actions and graphical effects to justifications based on a deductive model. In this passage there is the development of the capability to justify the validity of a procedure at the conceptual level (demonstration) on the basis of initial hypothesis, geometrical axioms and previously demonstrated theorems.

This passage cannot be spontaneously built as the result of the student-software interaction (e.g. as the result of having observed on the screen constructions that can be modified preserving some properties). The deductive method is built in the social interaction when students can experiment, under the teacher's guidance, the importance of justifying the accomplished procedure according to methods, which go beyond perceptual validation.

In the Cabri environment the History command is a fundamental tool at this regard. With this command the accomplished procedure becomes objective, de-personalised and de-contextualised.

In this way it is possible to focalise the attention on the properties of the construction and on their relationships and to consider them in the light of a geometrical theory and of previously demonstrated facts.

The History command is an important tool to maintain the mathematical discourse, which is developed in the class on the value, methods and techniques of geometrical demonstration integrated and "in focus" (supporting mathematical discourse function). This can be obtained thanks to the possibility offered of analysing and putting in relation the accomplished constructions

## References


[1] Bottino R.M., Chiappini G., "ARI-LAB: models issues and strategies in the design of a multiple-tools problem solving environment", Instructional Science, Vol. 23, n°1-3, 1995, Kluwer Academic Publishers, pp. 7-23

[2] Dreyfus T., Immagery for Diagrams in R. Sutherland and J. Mason (eds) *Exploiting mental imagery with computer in Mathematics education*, NATO ASI Series, Springer - Verlag, Berlin, 1995, Series F, Vol 138, 3-19

[3] Dorfler W., Mathematical objects, representations, and imagery in R. Sutherland and J. Mason (eds) *Exploiting mental imagery with computer* in Mathematics education, Nato Asi Series F, Vol. 138, Berlin: Springer Verlag, 82-94, 1995.

[4] Kaput J.J, Overcoming phisicality and the external present: cybernetic manipulatives, in R. Sutherland & J. Mason (eds): *Exploiting mental Imagery with Computers in Mathematics Education*, Nato Asi Series F, Vol. 138, Berlin: Springer Verlag, 161-177, 1995

[5] Card S. K., Mackinlay J. D., Shneiderman B., *Information Visualisation*, Morgan Kaufmann Publisher, 1999

[6] Thompson P. W., Imagery and the development of mathematical reasoning, in Steff, Nesher, Cobb, Goldin, Greer (eds) *Theories of mathematical learning*, LEA Publisher, 267-285, 1996

[7] Laborde C., The computer as part of the learning environment: the case of geometry, in Keitel C. & Ruthven K. (eds.), *Learning from Computers: Mathematics Education and Technology*, Nato Asi Series F, Vol. 121, Berlin: Springer Verlag, 48-67, 1993

[8] Mariotti M.A., Introduzione alla dimostrazione all'inizio della scuola secondaria superiore, *L'Insegnamento della Matematica e delle Scienze Integrate*, Vol. 21B, N 3, pp 209-252, 1998

[9] Bliss J. , Externalizing thinking though modelling: ESRC tools for exploratory learning research program, in Vosniadou, De corte, Glaser, Mandl (eds), *International perspectiveson the design of tecnology-supported learning enviroments*, LEA Publisher, 25-40, 1996

# Virtual Environment of Water Molecules for Learning and Teaching Science

**Jorge F. Trindade**
Instituto Politécnico da Guarda
Escola Superior de Tecnologia e Gestão
Guarda, Portugal.
jtrindade@ipg.pt

**Carlos Fiolhais**
Departamento de Física
Universidade de Coimbra
Coimbra, Portugal.
tcarlos@hydra.ci.uc.pt

**Victor Gil**
Departamento de Química
Universidade de Coimbra
Coimbra, Portugal.
vgil@cygnus.ci.uc.pt

**José C. Teixeira**
Centro de Computação Gráfica
Coimbra, Portugal.
teixeira@ccg.uc.pt

**Abstract**

*Virtual reality is an emerging computer visualization technology which allows educators to place their students into instructional environments heretofore difficult or impossible to achieve. In order to take full advantage of this technology, a virtual environment, "Virtual Water", is being developed. Virtual reality has been recently employed in a few educational applications, but the project presented here is the first application we know of combining aspects as atomic and molecular orbits, electron densities, water molecular dynamics for the liquid and gaseous phases and phase transitions.*
**Keywords:** *virtual reality, water, molecular dynamics, quantum mechanics.*

## Introduction

In the past fifteen years, research on physics education has brought a lot of information about the difficulties students have in learning physics. At the same time, the ongoing revolution in information technology has led to new tools for creating innovative educational environments. In response to these developments, a wide variety of new models of physics instruction are available.

People's understanding of what computers can do has shifted dramatically as the size and cost of these devices has decreased while their power has increased. At the beginning, computers were seen as number-crunching machines, but now we have tools to manipulate information, namely in the graphic form. Up to now, the use of computational means in Physics education stood mainly on the creation of 2D representations that the students could use to build more refined mental models. However, recent advances have created new possibilities and the visualization of 3D objects and data becomes increasingly important in learning several scientific subjects (in particular, atomic and molecular science, fluid dynamics, etc.). With virtual reality (VR), the visualization of complex data and the building of more adequate conceptual models is possible [1, 2].

VR is one of the most significative steps towards a natural human-computer communication, allowing an easy presentation and an intuitive grasp of complex data. Virtual environments can represent various aspects of a natural environment or even a totally artificial world. The inclusion of haptic information and direct manipulation contributes to the impression of being immersed in a real situation [3]. Perception, as learning, is an individual act. Since each person evolves a unique psychomotor approach to interacting with a physical environment, individuals have more varied responses to 3D interfaces than to the standard 2D graphical interfaces with menus, windows, and mouse.

The Physics Department of the University of Coimbra, Portugal, the Exploratory Henry the Navigator, the Computer Graphics Center (both also in Coimbra) and the High Education School for Technology and Management of Guarda, are developing the "Virtual Water" project, a virtual environment devoted to the learning of Physics and Chemistry. This work involves the know-how of different fields (computational simulation of physical systems, computer graphics and science education) in order to arrive at a useful scientific-pedagogic visualization of water.

The main goal is to enable the user to visit virtual attractive and enjoyable sceneries where he can learn the constitution and properties of water. The user will be able to interact and “change” the environment to better understand the structure of water. The virtual environment is realistic and comprehensive, allowing for exploring the atomic constituents of water as well

as macroscopic structures. With this environment, the study of the atomic units, the molecule geometry, the molecular orbitals, the electron density, and molecular normal modes is facilitated. Using simplified equations, as those of Newtonian mechanics with the Lennard-Jones potential, simulation of the molecular dynamics of the solid, liquid and gaseous phases and phase transitions, is also available. Realistic simulations use more complicated forces [4].

## Description of the Project

The study of water is motivated by the fact that a better understanding of it, in its different functions and aggregation states, is only possible if the water molecule itself and the behavior of a set of molecules in given conditions are understood. With the speed of modern computing resources, it is becoming common in research to model aqueous systems atom by atom, moving each molecule in response to the forces acting on it. These advances should be followed by education applications.

### 2.1. Our Problem

Innovative practices using computers are being tried for teaching and learning science [5]. VR has been touted as a powerful teaching and training tool because ([6], [7], [8]):

- Supports direct experience of phenomena.
- Is 3D.
- Allow for multiple frames of reference.
- Offers multisensory communication.
- Is physical immersive.

However, current immersive VR technology has several limitations that may impede usability and learning ([9]):

- Head-mounted displays and gesture devices interfere with interaction.
- Input devices have restricted fidelity and versatility.
- The display capabilities are somewhat limited.
- Multisensory inputs can interact to cause unintended sensations and perceptions.

Through the design of "Virtual Water" we have started to determine how the characteristics of VR influence usability and learning. At present, achieving VR's potential to enhance learning requires transcending the interface barriers through paying careful attention to usability issues.

### 2.2. Our Approach

To design a educational virtual world, we have adopted an iterative learner-centered design approach that focuses on usability and learning. The main goals in producing our application are:

- To provide an educational environment for students to explore some microscopic and abstract concepts, which they are teached in class but are far away from daily experience.
- To develop a practical knowledge concerning the application of VR techniques in education.
- To contribute with data on the pedagogical usefulness of VR. People in the field have an intuition that VR can have a strong impact on learning. But believing in VR is not enough: its usefulness has to be proved.

### 2.3. System Configuration and Models Development

The interface of our immersive, multisensory environment is typical of current high-end VR. "Virtual Water's" hardware architecture includes a Pentium II processor with 3D graphics accelerator. For the navigation and immersion in the virtual environment, we use the Head Mounted Display V6 from Virtual Research, as well as one Cyberglove with cybertouch (for haptic information), from Virtual Technologies, and a Polhemus Isotrack II magnetic orientation position sensor for two receptors (see Figure 1). This class of hardware will not be common in classes in the next years, but it allows us to deliver a product with minimal quality.

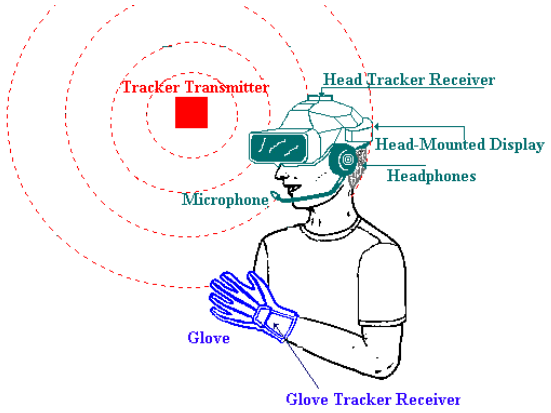



Figure 1. The virtual reality hardware interfaces.

Concerning the software, we use *WorldToolkit* (from Sense8), that serves the definition and creation of the virtual scenarios, and other packages for model development. For the design of the first part (quantum mechanics) of "Virtual Water", we use mainly the freeware *PC Gamess* [10], that performs the water molecule calculations (namely geometry optimization,

electron density, etc.) and the freeware *Molden* [11] package for molecular representation. For the second part (molecular dynamics), we use commercial software packages (*Mathcad*, *3D Studio Max* and *Autocad*) for development of models and optimization and *Visual* $C^{++}$ for implementing the molecular dynamics algorithm.

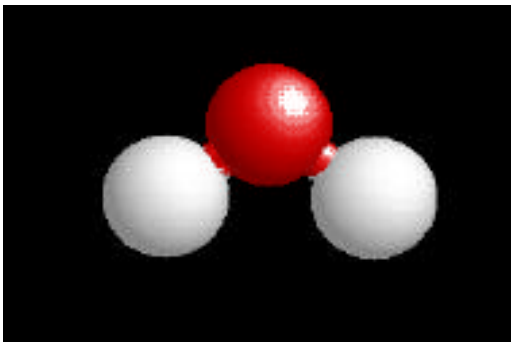
### 2.4. The Training Environment

Scenery exploration is preceded by navigation in a training environment. The first "room" which users encounter is the training environment, which contains known objects. The goal is to overcome the disorientation problems which some users experience when first using VR. This virtual scenery is designed to be a simple, familiar environment in which users can become comfortable with the VR hardware before moving on to more complex an abstract environments.

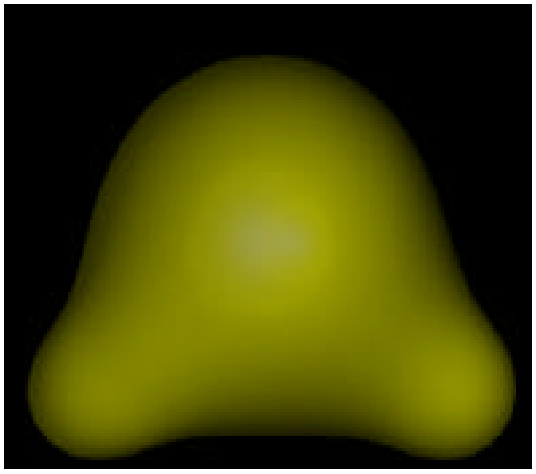
### 2.5. The Quantum Mechanics Virtual Environment

In this scenery we deal with the geometry of the water molecule (see Figure 2-a), its electron density (see Figure 2-b), and molecular orbitals (see Figure 2-c and d). The 3D models were produced with *PC Gamess* (a program that performs *ab initio* calculations to obtain the molecule geometry), and *Molden* (a program that enables the visualization).

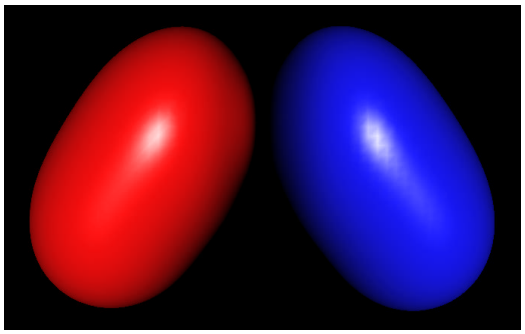
In this virtual environment students can build and travel through molecular orbitals, ad molecular density and better understand the water molecular structure in order to clarify the chemical bonding.



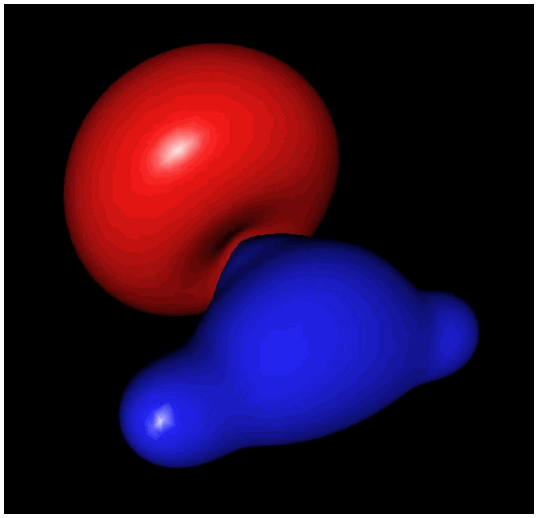
a)



b)



c)



d)

Figure 2. Some 3D models of the quantum mechanics virtual environment: a) ball-and-stick representation of water molecule; b) equidensity surface representation of the same molecule; c) the third and fourth occupied molecular orbital representation of water molecule (the total electron density is obtained adding up the occupied molecular orbits).

**2.6. The Molecular Dynamics Virtual Environment**

We aim at understanding some water properties by direct simulation. We assume that the molecular dynamics can be treated classically because realistic simulations (with quantum effects) use complicated forces [4] and are much more computational demanding. We use Newton's equations of motion

$$m\frac{d^2\vec{r}_i}{dt^2} = -\vec{\nabla}_i \sum_j u(r_{ij})$$

with the Lennard-Jones potential,

$$u(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

(the ε and σ are constants).

The repulsion at small distances is a consequence of the Pauli exclusion principle. The weak attraction at larger distances is due to the mutual polarization of each molecule.

Newton's equations are solved for each molecule starting from initial positions and velocities and using the force acting on each molecule

$$-\vec{\nabla} u(r) = \frac{24}{r}\varepsilon\left[2\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\hat{r}.$$

Then, we calculate the positions

$$x_i(t+\Delta t) = 2x_i(t) - x_i(t-\Delta t) + \frac{F_{x,i}(t)}{m}\Delta t^2$$

and velocities

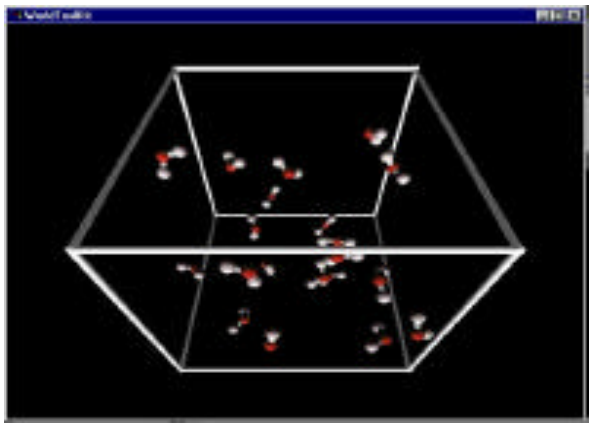
$$v_{x,i}(t) = \frac{x_i(t+\Delta t) - x_i(t-\Delta t)}{2\Delta t}$$

of each molecule at successive times.

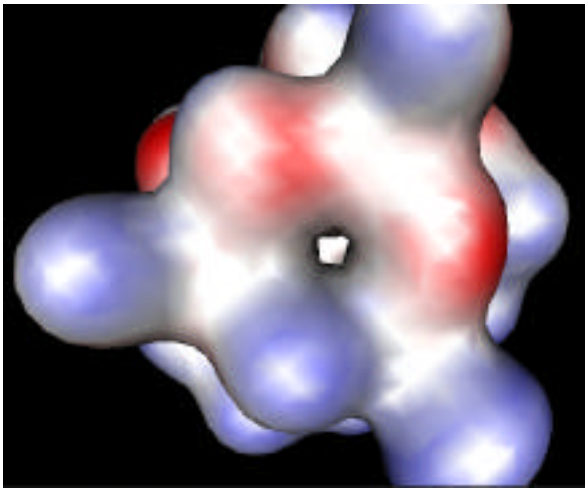
The user is able to interact and change the environment in order to study the liquid and gaseous phases and phases transitions (see Figure 3-a). The solid phase is also examined (see Figure 3-b).

One of the new aspects of this pedagogical work is the 3D representation of the ensemble of molecules instead of the usual 2D. The same algorithm has been applied to the liquid and gas phases, emphasizing their similarity.

In order to assure a real-time rendering (the level of detail which is attainable at a practical speed), the number of molecules in each phase or phase transition simulation has been carefully tested. For example, for the gaseous phase simulations we have used 20 molecules. This restriction is the compromise between the computer graphics capability, molecular models and real-time calculations. Contrary to most computer applications, VR must recalculate the user's view for each frame update, taking account considerations as light sources, shadings, distance from the user, etc. To be effective, all this must be performed several times per second in addition to any calculations that have to be made.



a)



b)

Figure 3. Images from the molecular dynamics environment: a) the gaseous phase, with the ball and stick model, showing twenty molecules in a box; b) The ice phase, with the same number of molecules, but with the electron density representation. These pictures were created using the same software as in Figure 1, being the dynamics implemented in *Visual C*$^{++}$.

## Conclusion

The use of 3D graphics seems to be a powerful tool for visualizing and understanding complex and/or abstract information. Immersion is a recent aspect which has to be better explored and evaluated. A virtual environment for teaching and learning Physics and Chemistry is being developed to study the possibilities of VR in education. The main objectives are:

- Get new means for teaching and learning the physics and chemistry of water.
- To apply VR as an educational tool, in order to determine which aspects of VR provide the most effective educational benefits, and to learn the strengths and weaknesses of this technology in an educational setting.
- To build knowledge of VR techniques and tools which can later be applied to other problems.

Feedback from students still needs to be collected and analyzed.

## Acknowledgements

We wish to acknowledge the assistance of the student André Dias, who has a developed the molecular dynamics part of the "Virtual Water".

## References

[1] J. Trindade and C. Fiolhais. A realidade virtual no ensino e na aprendizagem da Física e da Química. *Gazeta da Física* **19**, pages 11-15, 1996.

[2] C. Fiolhais and J. Trindade. Use of Computers in Physics education. *Proceedings of the "Euroconference'98 – New Technologies for Higher Education"*. Univ. Aveiro: ed. A. Ferrari, Aveiro, 1999.

[3] J. C. Teixeira, A. Côrte-Real, P. Bernardes and F. Macedo. Santa Clara-a-Velha: virtual environments and cultural heritage. *Proceedings of CAD & Graphics'97, Fifth International Conference on CAD & CG*, pages 81-88, Shenzhen, China, December, 1997.

[4] M. Laasonen, M. Sprik and M. Parrinelo. "Ab initio" liquid water. *J. Chem. Phys.* **99** (11), pages 9080-9089, 1993.

[5] J. C. Teixeira. Environments for teaching computer graphics: an experience. *Comput. & Graphics* **20** (6), pages 927-935, 1996.

[6] M. Bricken and C. Byrne. Summer students in virtual reality. In Wexelblat, A. (Ed.), *Virtual Reality: Applications and Exploration*, pages 199-218, New York: Academic Press, 1993.

[7] T. Erickson. Artificial realities as data visualization environments. In Wexelblat, A. (Ed.), *Virtual Reality: Applications and Exploration*, pages 1-22, New York: Academic Press, 1993.

[8] R. Kalawsky. *The Science of Virtual Reality and Virtual Environments. A Technical, Scientific and Engineering Reference on Virtual Environments*. Addison-Wesley Publishing Company, 1993.

[9] C. Wickens. Virtual Reality and Education. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 1,* pages 842-847, New York: IEEE Press, 1992.

[10] PC Gamess, a program for ab initio quantum chemistry, written by Alex. A. Granovski, Moscow State University.

[11] Molden, a package for displaying MOLecular DENsity, written by G. Schaftenaar, CAOS/CAM Center Nijmegen, Toernooiveld, Nijmegen, The Netherlands.

# Evaluation of Driving Education Methods in a Driving Simulator

**J. Miguel Leitão**[1,3], **Alexandra Moreira**[2], **Jorge A. Santos**[2], **A. Augusto Sousa**[3,4] and **F. Nunes Ferreira**[4]
[1] Departamento de Engenharia Electrotécnica, Instituto Superior de Engenharia do Porto, Portugal
[2] Instituto de Educação e Psicologia da Universidade do Minho, Braga, Portugal
[3] Instituto de Engenharia de Sistemas e Computadores do Porto, Portugal
[4] Faculdade de Engenharia da Universidade do Porto, Portugal
jml@ipp.pt

**Abstract**

*This paper presents the development of a realistic driving simulator and its use in the study of driving education methods. In this study, a large set of persons without driving license will be used. Each person will be submitted to a driving course that may include driving lessons in a real car and / or in the driving simulator. The realism used in the simulation can be adjusted between the classical non-interactive, film based training and the full realism achieved by our driving simulator. Each driving education method will be evaluated by comparing driver's competence before and after the applied course.*

*We expect to determine the influence of the simulation realism in the learning process, and to understand how and when the driving simulator should be used for better performance in the education process.*

## 1. Introduction

The large number of accidents and dead occurred due to traffic circulation in public ways is one of the most concerning problems of today societies. It has been commonly accepted that one method to reduce these losses is the improvement in driver's education and selection. To do so, and due to recent advances and spread of image generation systems, interactive driving simulators are being more and more used in driving education tasks.

It is commonly accepted that the use of driving simulators presents some advantages over the traditional methods of drive learning. Because of their virtual nature, the risk of damage due to incompetent driving is null. This usually allows a reduction in the cost of driving lessons or a longer and richer training but imposes a very hard limit to the cost of a driving simulator. According to professionals in the area of driving education, to explore its advantages in driving education use, a driving simulator must not be much more expensive than real vehicle.

Although the recent advances in computing and image generation systems, to keep the cost of a driving simulator compatible with the education uses, many simplifications must be done. For example, reductions in field of view, image resolution, vehicle's dynamic behavior and scenario reactivity are common in today educational driving simulators.

This paper presents the development of a realistic driving simulator and its use in the study of driving education methods. This study will be done using data acquired from a large set of persons without driving license. Each person will be submitted to a driving course that may include driving lessons in a real car and / or in the driving simulator. The realism used in the simulation will be adjusted between the classical non-interactive, film based training and the full realism that can be achieved by our driving simulator. Each driving education method will be evaluated by comparing driver's competence before and after the applied course.

As results of this study, we expect to determine the influence of the simulation realism over the learning process, and to understand how and when the driving simulator should be used to maximize the driving education's productivity.

The state of the art in the field is summarized in Section 2. Section 3 discusses the use of driving simulators in driving training and driver's evaluation. Section 4 presents a short description of DriS, its goals and applications. The study, being prepared, about driving education methods and its evaluation is detailed in section 5.

## 2. Related work

Driving simulators are hi-complex real-time, imersive systems *[1][10]*. The improvement in realism and the validation of their performance are the major concerns in development of any scientific driving simulator *[3][4][12]*.

The usability of simulators in training and in evaluation of human performance is already studied and generally accepted under several restrictions *[14][9]*. In driving education, it is today common to find the use of driving simulators either with training or with evaluation proposes. Some experiences are known that analyze discrepancies between human driving in real and in simulated environment *[11][13]*, but no studies are know that evaluate how these discrepancies affect the driving education process.

## 3. Driving Simulators

In the last years, we assisted a tremendous development in computing and image generation technology that allowed the proliferation of VR systems with acceptable quality for some education purposes. Even thought, the development and maintenance of a realistic driving simulator is still a very expensive task. So, to keep the cost of a driving simulator compatible with the education uses, many simplifications must be done.

For example, most driving simulators used for driver education present only a narrow filed of view. This reduction clearly degrades the perception of the surrounding environment by the human driver and also limits the amount of information that he must receive and process. Also, the generation of the backward images presented in mirrors is crucial, for instance, for training overtaking maneuvers.

The image resolution used in today driving simulators is always poorer than the one obtained by the direct observation of the real world. This results in a different driver behavior due to the lack of obstacle perception. It is known that, when approaching an obstacle, a driver breaks sooner in a driving simulator than in a real car *[13]*.

Reductions in vehicle's dynamic behavior and scenario reactivity are also common in today educational driving simulators.

All these, and other simplifications affect human driver reactions but, it is accepted that they do not invalidate the data we can get from the simulated driving task and the knowledge acquired from training. It is known that, when correctly used, even a simplified driving simulator can be useful either in driver performance analyses as in driving education. A big challenge that is proposed to every driving educator is the choice of how and when to use driving simulator in order to maximize the education productivity. This choice implies a selection of simulation realism and its trade off with the simulator cost. Then a selection of training exercises must be made but this selection is usually limited by the realism achieved by the possibly available driving simulator. Because a simulated training is still far from completely substitute a real experience, decisions must also be taken on how these two kind training should interact.

## 4. DriS - Driving Simulator

A description of our driving simulator (DriS) can be found in [2]. Its main core runs on a SGI Onyx Reality Engine 2 graphical workstation (Figure 1). This workstation holds the scene database, and performs the simulation and the image synthesis tasks. It is connected to the sound PC and to the cockpit PC by local area network. Communications are implemented synchronously with the rendering process, using a socket library.

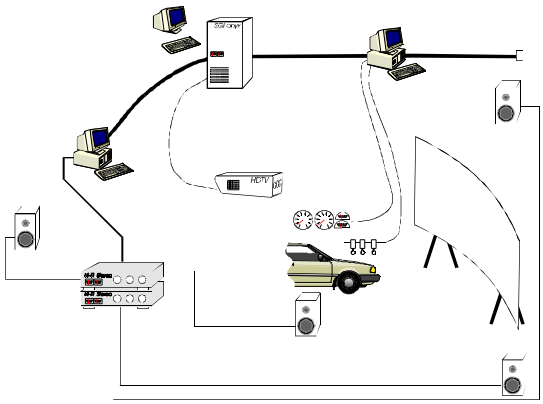


Figure 1: DriS driving simulator.

In DriS, the driver sees a realistic image either in a large screen display or in a HMD. The generated image is applied to a large flat screen (3.40mx2.00m) by a video projector. Typical experiments are usually performed with resolution of 1280x1024, with an image update of 20 images per second, and a frame rate of 60 frames per second *[15]*.

High quality audio information is also supplied in either stereo sound with headphones or in multi-channel surround sound using loudspeakers, and an additional channel for the typical sound of the vehicle *[6]*.

DriS is being developed by a large team of researchers from several institutions and different scientific areas. The development team includes experts in computer systems, image analysis, computer graphics, road and traffic engineering and psychology. Their main goals are the study of driver behavior and road analyses under conditions that are difficult or even impossible to reproduce:

- risk situations

- accident
- new roads
- new managing systems
- new road signals
- new vehicle models

### 4.1. Applications of DriS

In [7], Noriega and others presented a study of vehicle's motion detection with concurrent self motion with three kinds of road pavement:

- concrete pavement
- bituminous pavement
- bituminous pavement with chromatic bands

The DriS driving simulator allowed an easy preparation of the stimulus presented to each one of the 106 persons with driving license that participated in this study. It was concluded that the pavement with chromatic bands produces a higher number of wrong detections of vehicle's motion, while differences in image contrast were considered not relevant.

A new study is now running to evaluate the influence of external publicity panels (outdoors) in driving performance and several others are being prepared *[5]*. One of most important studies is the evaluation of driving education methods that is detailed in this paper.

## 5. Evaluation of Driving Education Methods

This section presents the study of driving education methods that is now being prepared. This study will be done using data acquired from a large set of persons without driving license. Each person will be submitted to a driving course that may include driving lessons in a real car and / or in the driving simulator. The realism used in the simulation will be adjusted between the classical non-interactive, film based training and the full realism that can be achieved by our driving simulator. Each driving education method will be evaluated by comparing driver's competence before and after the applied course.

### 5.1. Driving Environment

A driving simulator will be used (DriS), and the virtual environment will try to reproduce a typical street from Porto. For this propose, we choused some well-known streets and avenues from Porto located in the Foz do Douro area. This area is traditionally one of the most used areas in Porto for driving lesson purposes.

Roads were modeled using real field dimensions and buildings were finished using photographic textures taken from some choused streets:

- *Avenida do Brasil*
- *Avenida de Montevideu*
- Passeio Alegre.

Figure 2 presents a view of a modeled road. Several autonomous vehicles were added to compose the environment traffic. These vehicles are driven by virtual drivers that try to emulate human drivers *[8]*.

Figure 2. View of a modeled road.

### 5.2. Methodology

This study will be done using data acquired from a large set of persons (subjects) without driving license. Each person will be submitted to a driving course that may include driving lessons in a real car and / or in the driving simulator. The realism used in the simulation will be adjusted between the classical non-interactive, film based training and the full realism that can be achieved by our driving simulator. Each driving education method will be evaluated by comparing driver's competence before and after the applied course.

The subjects set is being formed using people that meet all the following requirements:

- have not driving license
- want to have driving license
- are, preferable, between 16 and 22 years old
- have not visual or physics deficiencies

The study will be performed in two identical experiments. The preliminary experiment will be used to select the course curriculums that will be used in the final experiment. The preliminary experiment is now starting and will take about 9 months long.

Each subject will first be evaluated in driving performance terms and assigned randomly into one of 4 groups (Figure 3). Some effort must be done in order to achieve groups with the same amount of subjects and with similar driving and intellectual skills. We plan

to use near 20 persons in the preliminary experiment and about 200 in the final one.

Then a different driving course will be applied to each subject group.

Course 1 will be a traditional driving course, performed at several driving schools and with different driving teachers. In Portugal the minimum curriculum of a driving course includes 25 hours practical lessons but it is known that this minimum is sometimes not needed to achieve the required driving competence. In our study, the final evaluation will take place after 15 hours of practical driving lessons.

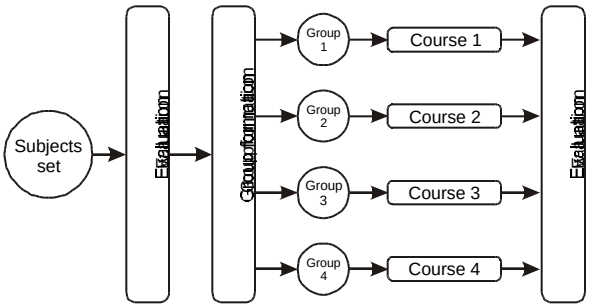



Figure 3: Experiment workflow.

Course 2 will consist of 30 hours of driving lessons using an interactive driving simulator. Lessons will be supervised by a driving teacher and will include simulated driving practice under several conditions of:

- traffic intensity,
- traffic velocity,
- own vehicle characteristics

Course 3 will consist of 40 hours of driving lessons using a non-interactive driving simulator. Lessons will be supervised by a driving teacher and will include simulated driving practice under several conditions of:

- traffic intensity,
- traffic velocity,
- own vehicle characteristics

### 5.3. Evaluations

Initial and final driving performance evaluations will be done using data acquired from the simulated driving in a realistic driving simulator.

Each subject will be asked to follow a previous route, in order to adapt to the simulator. Then, in the experimental route, the driver will follow a first road without other vehicles, a second one with low traffic intensity, and a third with high traffic intensity.

During the experimental route, several driving parameters will be measured and stored for further analyses.

Measures can be divided into three groups: measures reported by the driver, a rating scale of mental effort (RSME); physiological measures, eye movements (with a NAC eye tracking recorder); and performance measures.

The rating scale of mental effort will be monitored from several electro-magnetic sensors located in the subject's head, using dedicated specialized equipment (Figure 4) that produces a graphical output printed in paper.

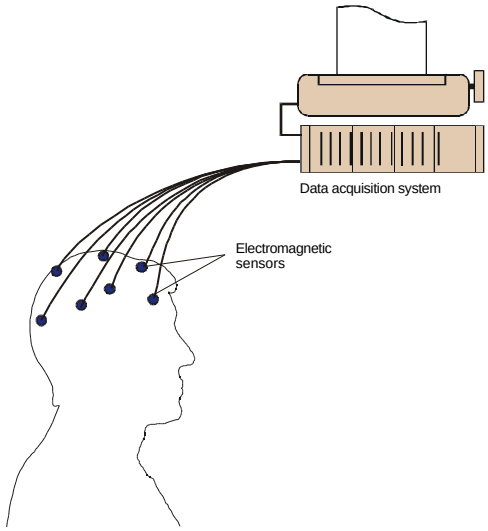



Figure 4. Mental effort acquisition system

Eye movements will be recorded in video using a dedicated optical tracking located near subject's head that detects the subject's retinal position (Figure 5). This data is related to the viewing direction and it is very important because it permits to determine attention and distraction rates.

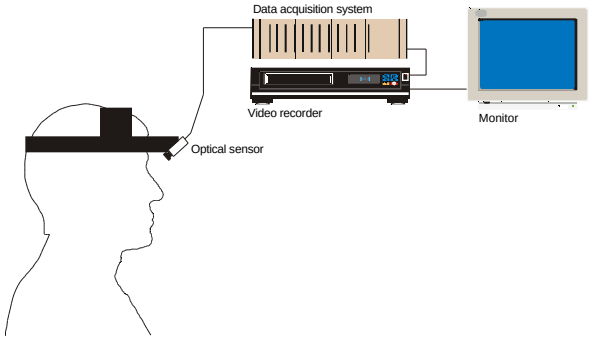



Figure 5. Eye movements recorder equipment

Both RSME and NAC equipment are synchronized with the simulation experiment by a developed interface module (Figure 6).

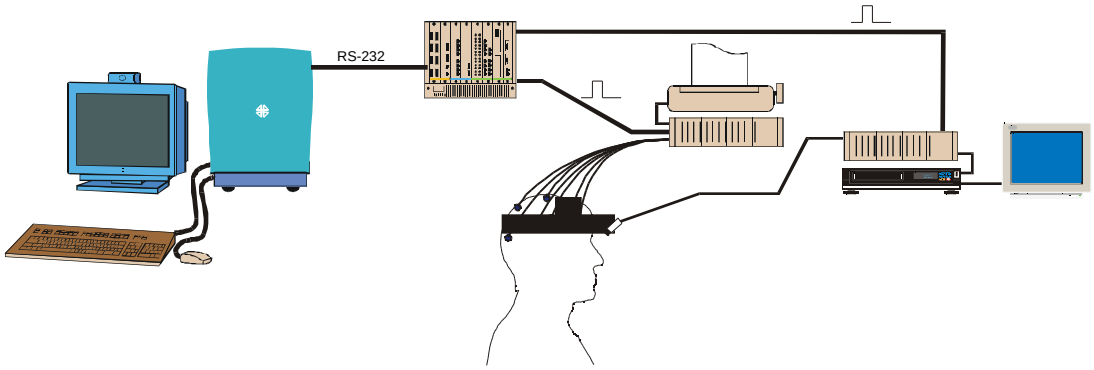



Figure 6. Synchronizing the acquisition equipment with simulation experiment

This module receives information about the simulation stage from the graphical workstation through a standard serial connection (RS-232) and outputs the required synchronization signals. The synchronization allows specialists to relate the outputted data with the simulated traffic events.

Learning measures will be the performance measures: the number of errors committed

- reaction times
- time-to-line crossing
- control of the driving path
- driving speed
- number of collisions

These measures are obtained by processing the information that the simulator reports in real-time.

This study reflects the relation between the driving simulator use and learning process in these courses. The performance results from these initial courses are presumed to be different from each other, even with the specified differences in course duration. An adaptation of each course curriculums must be done using information from the results of this preliminary experience. This adaptation is important because performance results from different courses can only be compared if they have some similarities.

The final experiment of this study will use the same methods but with differences on the number of used subjects, the number of groups and courses, and the curriculums of the driving courses.

A second study will be done on the validation of driving learning assessment. Driving teachers and experts will be asked to analyze video records from the first study (images recorded from the driver point of view), and to score the drivers performance with an evaluation sheet used at driving schools. The objective data from learning measures gathered at the first study will be then matched with the subjective evaluation from experts. This second study will allow us to analyze the value of subjective reports from driving teachers on topics such as vehicle control, skills of learning drivers in unexpected situations, learning improvement along driving sessions and agreement of experts in the assessment of driving skills.

As results of these studies, we expect to determine the influence of the simulation realism over the learning process, and to understand how and when the driving simulator should be used to maximize the driving education's productivity.

## 6. Conclusions

This paper shortly presented the architecture of DriS, a realistic driving simulator. DriS integrates a typical cockpit of a real car, a SGI Onyx Reality Engine 2 graphical workstation and a large projection screen. It also uses a PC for sound generation, and another PC to control and monitor cockpit sensors and signals. DriS is being developed to study driver behavior, road performance and driving education methods.

Experiments being prepared to the evaluation of driving education methods were also presented. These experiments will use data acquired from a large set of persons without driving license. Each person will be submitted to a driving course that may include driving lessons in a real car and / or in the driving simulator. The realism used in the simulation will be adjusted between the classical non-interactive, film based training and the full realism achieved by our driving simulator. Each driving education method will be evaluated by comparing driver's competence before and after the applied course.

We expect these experiences will allow concluding about the required level of realism that a driving simulator must achieve in order to be used in driving education processes and to understand how and when the driving simulator should be used for better performance in the education process.

## 7. References


[1] Salvador Bayarri; Fernandez M.; Martinez M., *Virtual Reality For Driving Simulation*, Communications of the ACM, May 1996

[2] J. MIGUEL LEITÃO; A. COELHO; F. N. FERREIRA; *DriS – A Virtual Driving Simulator*; Proceedings of the Second International Seminar on Human Factors in Road Traffic, ISBN 972-8098-25-1, Braga, Portugal, 1997

[3] Salvador Bayarri; M. Fernández, *Design & Validation of Traffic Models in Driving Simulation*, Symposium on the Design and Validation of Driving Simulators, Valencia, Spain, 1996

[4] Olivier Alloyer, Esmail Bonakdarian; James Cremer; Joseph Kearney; Peter Willemsen, *Embedding Scenarios in Ambient Traffic*, Proceedings of DCS'97 (Driving Simulation Conference), Lyon, France September 1997

[5] J. Miguel Leitão; A. Augusto Sousa; Carlos Rodrigues; Jorge A. Santos; F. Nunes Ferreira; A. H. Pires da Costa, *Realização de Experiências num Simulador de Condução*, Psicologia del Tráfico y la Seguridad Vial / II Congresso Iberoamericano de Psicología, Madrid, July 1998

[6] António C. Coelho; J. Miguel Leitão; F. Nunes Ferreira, *Síntese de Som 3D em Ambientes de Realidade Virtual – Aplicação a Simuladores de Condução*, Proceedings of the 8º Encontro Português de Computação Gráfica, Coimbra, Portugal, 1998

[7] Paulo Noriega; Jorge Santos; Carlos Rodrigues; Pedro Albuquerque, *Vehicle's Motion Detection: Interaction with three Kinds of Road Pavement*, Proceedings of the Second International Seminar on Human Factors in Road Traffic, ISBN 972-8098-25-1, Braga, Portugal, 1997

[8] J. Miguel Leitão; F. Nunes Ferreira, *Agentes Autónomos em Ambientes Artificiais*, Proceedings of the 8º Encontro Português de Computação Gráfica, Coimbra, Portugal, 1998

[9] A. F. Sanders, *Simulation as a Tool in the Measurement of Human Performance*, Proceedings of the Ergonomics Society Annual Conference, Leeds April 1990

[10] James Cremer, Joseph Kearney, Yannis Papelis, *Driving Simulation: Challenges for VR Technology*, IEEE Computer Graphics and Applications, September 1996, pp. 16-20

[11] Lawrence H. Frank, John G. Casali, Walter W. Wierwille, *Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in Driving Simulator*, Human Factors, 1988, 30(2), 201-217

[12] Pieter Padmos, Maarten V. Milders, *Quality Criteria for Simulator Images: A Literature Review*, Human Factors, 1992, 34(6), 727-748

[13] Steven B. Rogers, Walter W. Wierwille, *The Occurrence of Accelerator and Brake Pedal Actuation Errors during Simulated Driving*, Human Factors, 1988, 30(1), 71-81

[14] Michael L. McGinnis, George F. Stone III, *Measuring the Effectiveness of Simulation-Based Training*, Proceedings of the 1996 Winter Simulation Conference, pp 934-938

[15] J. Miguel Leitão; A. Augusto Sousa; F. Nunes Ferreira, *An Image Generation Sub-system for a Realistic Driving Simulator*, Proceedings of the 1999 International Conference on Imaging Science, Systems, and Technology, Las Vegas, June 1999

# Scientific Visualizatio n as a Tool for Power Engineering Education

**Pavel Slavik,** Department of Computer Science and Engineering
**Frantisek Hrdlicka,** Department of Fluid Dynamics and Power Engineering
Czech Technical University
Karlovo nam. 13, 121 35 Praha 2
Czech Republic
e-mail: slavik@cs.felk.cvut.cz, hrdlicka@fsid.cvut.cz

**Abstract**

*This paper deals with the application of scientific visualization methods in the field of power engineering education. Behavior of various parts of a power plant is simulated and the results of these simulations are visualized. The visualization of processes that run in a power plant gives the students an opportunity to understand better the nature of these processes. This approach solves one of the main problems in power engineering education: the students usually have no chance to see various types of power plants in practice. The existence of models and their corresponding visualization allows students to perform experiments with various devices and in such a way acquire a much deeper knowledge about the subjects taught.*

***Keywords****: power engineering, power plant, visualization, particle models.*

## 1. Introduction

Production of electrical energy plays an important role in national economies around the world. Therefore it is necessary to pay attention to the education of highly qualified professionals in the field. They should be able both to design and to run power plants of various kinds. Education of these experts is very costly and, moreover, very complicated. One of the crucial problems is the lack of practical experience the students can gain during their study. Only practical experience can help the students deeply understand the nature of physical processes that run in a power plant. The traditional way of education is mostly based on mastering the theory of these processes, which creates a good base for their later deep understanding in practice. In some cases real models of some parts of power plants are used, where the students can perform experiments. This approach has several disadvantages - first of all these models are rather costly. Also parameter settings during experiments are usually accompanied by various problems.

The solution to the problems stated above is to use computer-based models of power plants. Currently there exist commercially available models that offer the possibility to perform virtual walkthroughs in a power plant. Such a walkthrough can give students some idea about the configuration of a power plant and also about the size of single appliances in a power plant. Performing such a walkthrough, students can get a feeling about mutual relations between single parts of power plant, which allows them to understand globally the process of electricity production. Nevertheless, this approach does not say anything about the processes running inside single parts of a power plant. The solution to this problem is to create models of processes running in power plants. These models should have both the simulation part and the visualization part.

## 2. Models of power plant components used in power engineering education

At Czech Technical University in Prague a project has been running for several years where these models are developed and implemented. Their applicability to education has been extensively tested. The aim of the project is to develop a set of models (and corresponding program modules) that would allow us to simulate various modes of operation of each part of a power plant. These modules should be of a special nature as it would be important to combine these modules into a functional model. For example: it is possible to assemble complex models of various power plants types that are characterized by different types of boilers, different types of coal transport systems, etc. Such an approach can significantly improve the quality of education, as a variety of single power plant types is usually not available in one country. Only by means of these models is it possible to acquaint students with all possible aspects of design and operation of power plants in a very general way.
The project as a whole is a large scale project which means that the final state will be reached within several years. Nevertheless, several modules currently exist

which are used independently in education with a high degree of success. This paper will deal with the current state of the art of the project mentioned. The simulation and visualization modules use traditional visualization techniques (mostly based on the use of particle systems). Some simulations and visualizations are unique because of the very specific nature of the processes investigated. It was necessary to develop special simulation and visualization techniques that allowed us to investigate the behavior of these processes - professional simulation and visualization software mainly did not cover specific requirements linked up with the processes simulated.

As the system is targeted to education, there are some special features in which this system differs from traditional simulation and visualization systems used in the given field. The traditional techniques mainly stress the accuracy of modeling and visualization. This approach results in a very time consuming calculations that give results after very long time. Our approach has been slightly different in some cases: to use simulation and visualization techniques that have lower accuracy and thus lower requirements for computational time. This offers very flexible feedback to students as they can see the results immediately (mostly in real time). In such a way students can get a feeling about the dynamics of the process simulated (and visualized).

The next step in using the system developed could be the use of professional specialized software (like FLUENT) and comparing the results obtained from our system with the results obtained by a professional system. A good exercise would be to modify existing algorithms used in our system in order to improve the match with the results obtained by professional software. Such an approach is not always possible because some models in our system are quite unique and they have no counterpart in professional systems. In this situation students compare the simulated results with real situations. They can compare the model and the reality and again explain why these differences exist. The experience gained could be used in the future for system improvement.

## 3. Description of existing modules

The already existing modules cover the most important parts of the electricity production pipeline. This means that the student using all these modules can get a complex idea about mutual relations between single stages of electricity production.

The modules available are as follows:

**- Heat transport in nuclear reactor cooling system**

In this case the data obtained by simulation of nuclear fission (here a commercial software product has been used) are visualized by a special module. This module is a good example of a situation where sophisticated commercial visualization software exists. The problem is that this commercial software runs on powerful workstations which means that this software is not suitable for education on a larger scale. Our software allows us to visualize heat flow and temperature distribution. The solution was based on the use of finite elements for the reactor body description and visualization of temperature and heat flow within these elements.

**- Combustion processes in fluidized bed boilers**

The problems related to fluidized bed boilers are rather complex. In the module a particle system was used. The particles represented coal particles during the combustion process. The attributes of particles changed according to the degree of their burning out. The visualization is then quite simple. Due to the simplicity of the particle model it is possible to generate animated sequences in real time and in such a way to observe and investigate the dynamics of the combustion process (Fig.1). Comparison has been made with simulations and visualizations made by means of a mathematical model that was based on the theory of fluid dynamics.

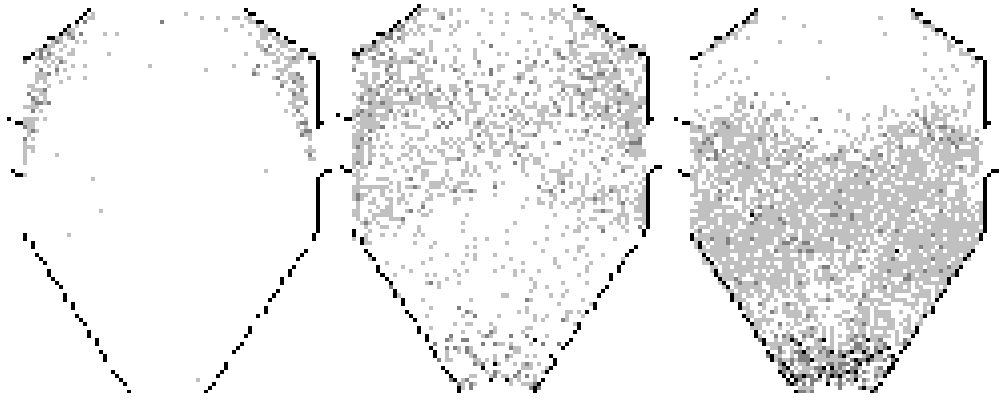


**Fig. 1 Visualization of various combustion modes**

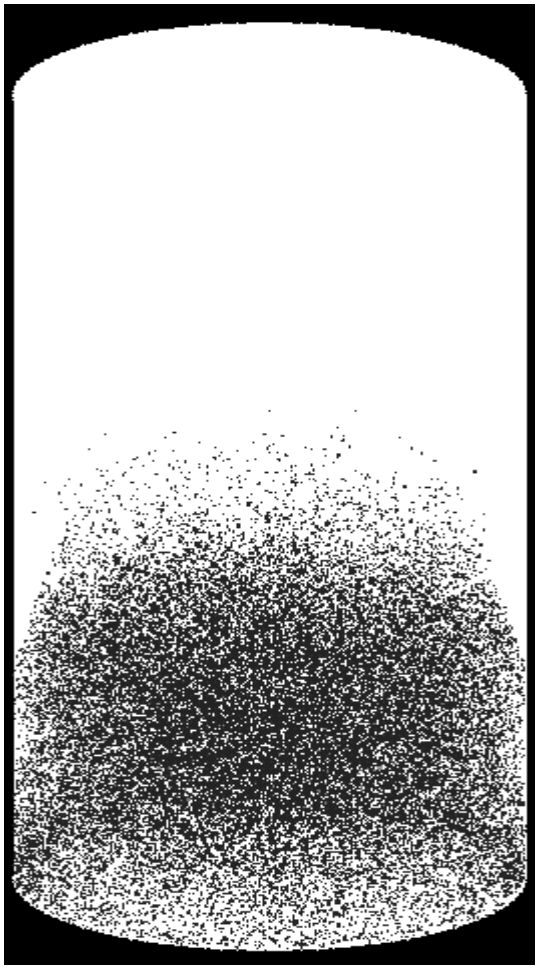


**Fig. 2 Visualization of combustion process in vortex furnace**

This model provides a more general approach to the studied phenomena. Nevertheless it is computationally expensive and each change of parameters requires it to perform new simulations (that are computationally costly). The particle model implemented allows us to perform changes of parameters dynamically with immediate visualization. Students can follow immediately the influence of the parameter changes just being made. This feature means the particle model is superior in terms of its use in education.

**- Combustion processes in vortex furnace**

The experience gained with the preceding boiler system has been used to create the model of a combustion process in a vortex furnace. One of the principal parameters of this boiler system is the configuration of an air velocity field that influences the trajectory of coal particles in the furnace. As in the previous example it is possible to set parameters of the boiler interactively by means of a properly designed user interface.

Also in this case the use of the particle model brings the same advantages as in the previous case. Due to some optimization in the program it is possible to work with several thousand particles (Fig. 2) at the same time (and visualize them). Such a large number of particles gives very extensive information about the situation in the furnace during the course of simulation.

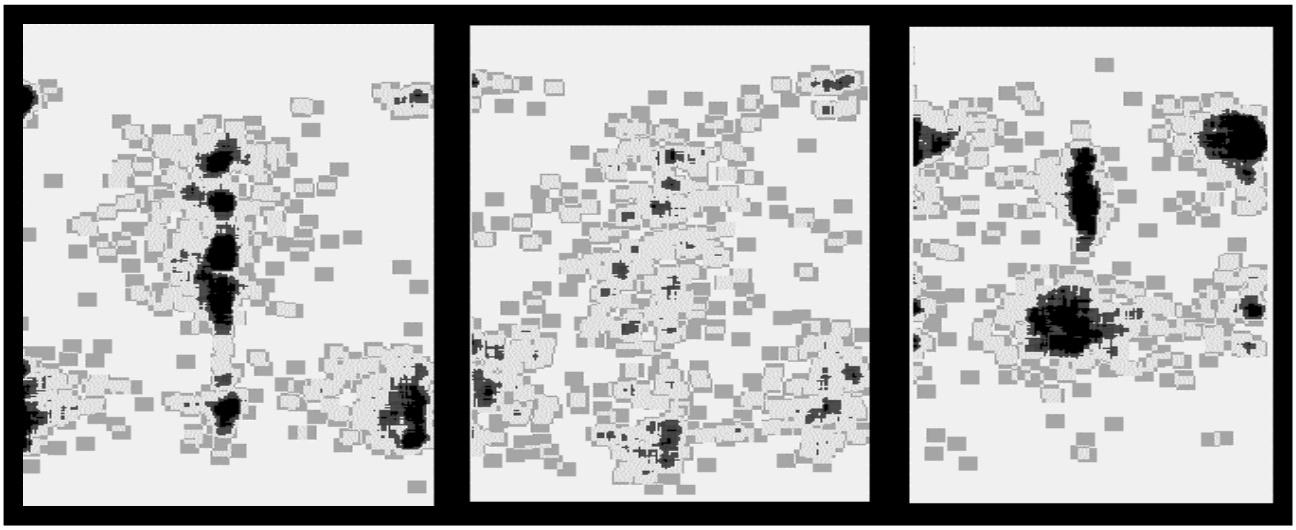


**Fig. 3 Visualization of the number of collisions**

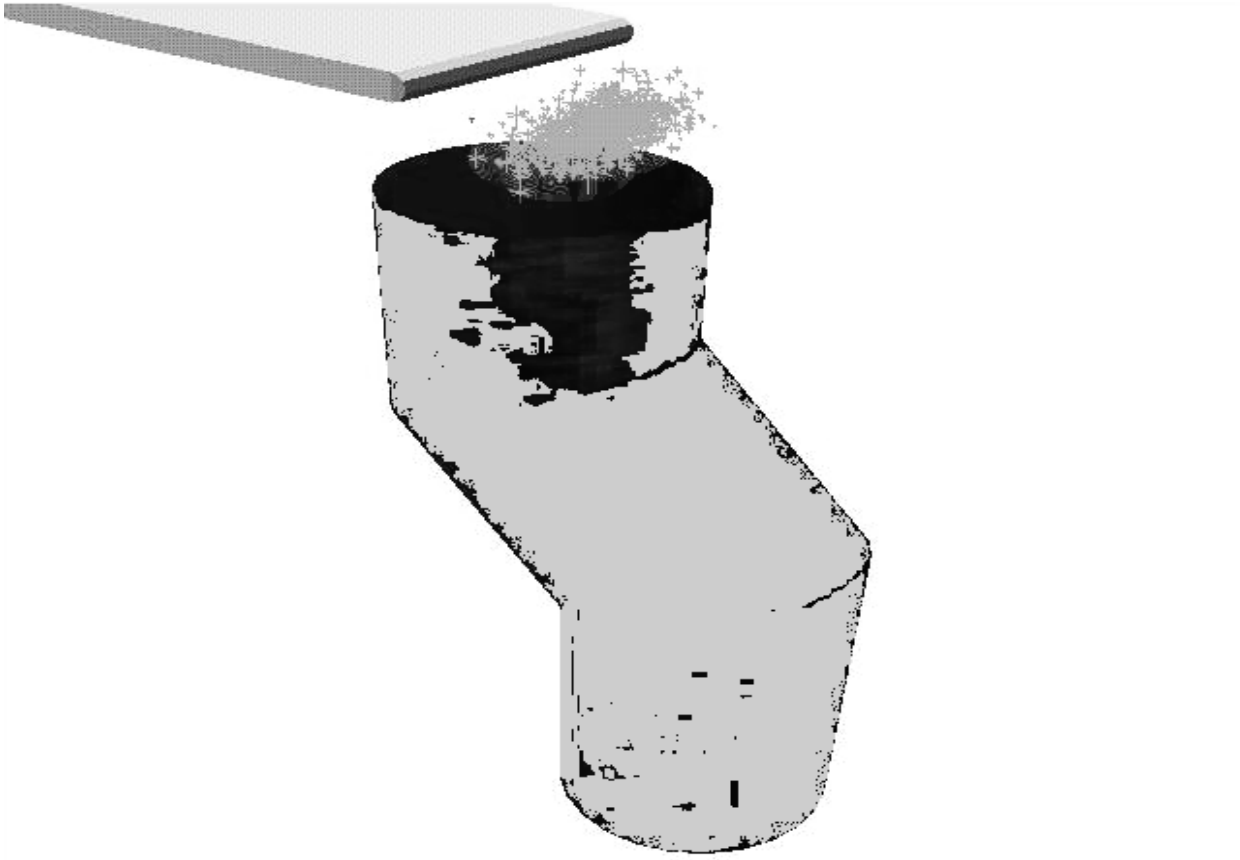


**Fig. 4 Visualization of the coal transport in the given pipe configuration (including collisions)**

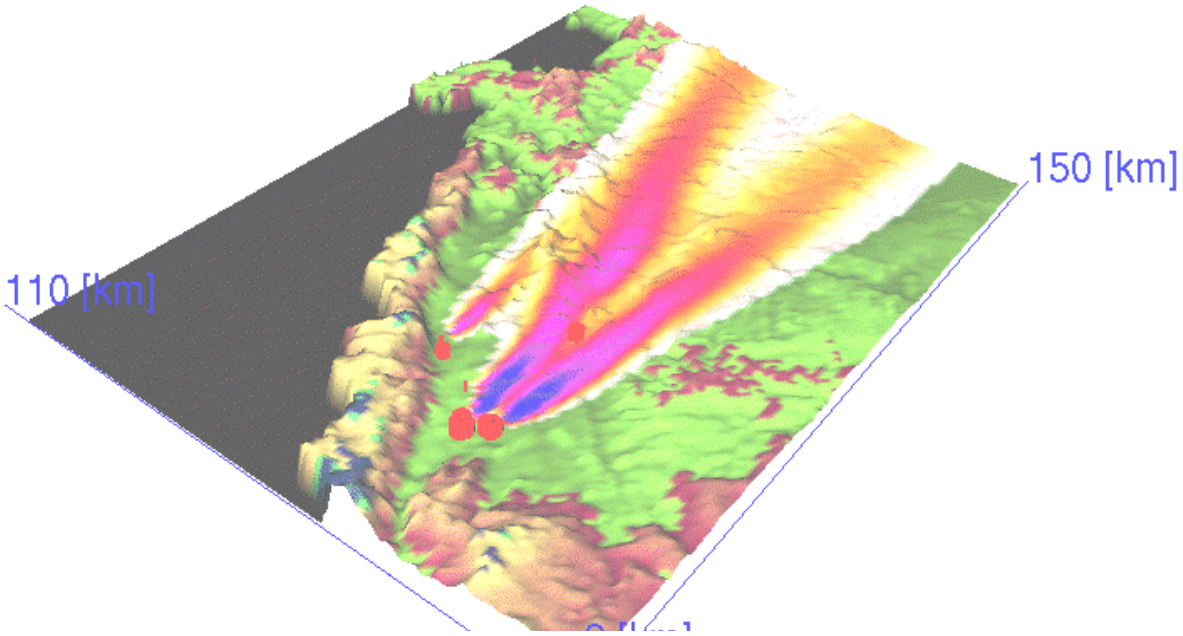



**Fig. 5 Distribution of pollution from three sources**

**- Coal transport system**

The system allows us to investigate behavior of coal particles during their transport in a coal dryer. The configuration of pipes the dryer consists of (Fig. 4), determines the extent of collisions of coal particles with the inner surface of the pipes. The number of these collisions determines the level of damage to the transport system. Another factor that influences the coal transport are parameters of coal like its humidity, granularity etc. The system allows the user to set interactively all these parameters and in such a way to simulate the coal transport process. The result of simulation and visualization is a picture where the distribution of collisions is shown.

The Figure 3. shows the visualization of the number of collisions of coal particles with the inner surface of the pipe through which the coal is being transported.

**- Moving granular bed gas cleanup filters**

The mentioned cleanup filter has been modeled. The model describes the behavior of granules (they have spherical form) - how they move in the filter during the time period of filter operation. Also here the process has several parameters that influence the efficiency of the pollutant separation. The user can experiment with various parameter settings and in such a way better understand the nature of the gas cleaning process. This problem as a whole is currently under intensive investigation and includes calculation of powers acting on single sphere in various positions inside the filter. Another feature under investigation is the calculation of friction processes between spherical granules inside the filter. The simulation of the granular bed gas filter is the most complex one of all problems investigated and mentioned in this paper.

**- Simulation and visualization of pollutants distributed from a stack in a power plant**

A model has been implemented that describes the pollutant distribution in an area near the power plant. The terrain profile is taken into account, and also the meteorological situation and characteristics of the power plant (and the stack in particular) are taken into account (Fig. 5). It is possible to simulate the influence of several power plants in a specific area where the share of pollution caused by each single power plant is calculated. The system has also been used in the course where the suitability of a location for a new power plant was tested.

## 4. Implementation

Most of the modules described were implemented in the MS Windows environment. The fact that most of the modules are available on the MS Windows platform is very important as these modules can be widely used due to wide availability of this platform. There are no specific demands on hardware. Nevertheless in the case where some dynamical effects are simulated, the appropriate computational power is of great importance. Great attention was paid to the development of a proper user interface keeping in mind that the modules could be used also by users who do not work in the field of computer science - so their knowledge about using computers could be defined as a casual user level.

## 5. Evaluation

From the examples given above it is possible to see that the modules already in use cover in cover several single stages of the electricity production. In such a way it is possible to see how the situation in one stage influences the situation in the next one - and also its global influence on the electricity production process, including the consequences for the environment.

By means of these program modules extensive teaching materials were prepared. Thanks to the flexibility of parameter settings in each of these modules, it was easy to generate a wealth of teaching material that was used both in lectures and in seminars. As an example it is possible to give the course ”Combustion and Boilers” where the visualization of the combustion process allows students to understand the mutual influence of all parameters that take part in controlling the combustion process. In such a way it is possible to determine ill-suited combinations of parameters (these combinations should be avoided in practice). The dynamics of the combustion is shown in the system by means of suitable visualization techniques.

In the framework of another course called ”Environmental Problems of Power Generation” the model of plum rise dispersion was used. Also in this case student could get a much better sense of the nature of the problem. Students could better feel the influence of each single source of pollution to the environmental situation in the given area. This means, for example, that the influence of the planned power plant can be determined in advance by means of the model implemented.

Some other models mentioned have been used to a limited extent (up to now) in the framework of seminars (e.g. for PhD students) and in specialized labs. This approach allows us to prepare the use of existing modules for regular courses. A very important

benefit is that the simulated results can be in many cases compared with the real data obtained in some cooperating power plant. This creates a very intensive feedback between the research at the university and practice. The environment created for university students in this way is, according to our experience, very stimulating and students get more involved both in research and in study.

Because the usage of these materials is quite new, there is still not enough data for the qualitative evaluation of the impact of this new approach on the quality of education. An evaluation strategy will be developed in the framework of future work. Nevertheless it is possible to say that this new approach led to a much better understanding of processes running in a power plant, which resulted in better results on exams in single courses.

## 6. Conclusion and future work

Due to the pilot nature of the program modules implemented, there is still a lot of work to be done in order to improve them. Moreover there are many parts of power plants of various types that should also be covered by corresponding modules. As an example, we can give modules for the heat transport and temperature distribution in boiler furnaces of various types, ash particle transport through boiler convection heat exchanges, etc. Some future developments should be the result of consultations with experts working in power plants in order to establish the real needs for some new models.

## References


[1] Slavik, P., Hrdlicka, F., Pavlis, T., Vlcek, P.: *Visualization of the Dispersion of Airborne Effluents*,In Proceedings of 7th Eurographics Workshop on Visualization in Scientific Computing, CTU Prague, 1996, pp.153-169

[2] Hrdlicka, F., Slavik, P.: *Dynamic Models of the Lignite Transport in Dryer of Power Plant Boiler*, To appear in Proceedings of 24th International Technical Conference on Coal Utilization & Fuel Systems, Clearwater, Florida, 1999

[3] Kubelka, F.: *Simulation of Moving Granular Bed Gas Cleanup Filter*, Student project, CTU Prague 1998

[4] Faltyn, R.: *Simulation and Visualization of Combustion Processes in Vortex Furnace*, Master Thesis, CTU Prague 1999

[5] Rais, D.: *Visualization of Combustion Processes in Fluidized Bed Boilers*, Master Thesis, CTU Prague 1997

# Training in Computer Graphics for Entertainment Production: What Future TDs Need to Know

**Jacquelyn Ford Morie**
Rhythm and Hues
jacki@rhythm.com

**Abstract**

*Training grounded in the theory and sound concepts of any discipline is widely recognized as the most appropriate educational practice which serves students well beyond their graduation. But in an increasingly competitive world students can be given a jump-start into a career by focusing their education towards an understanding of the knowledge set they will need for certain types of careers that use that discipline. Computer graphics is an especially rich area within Computer Science or Engineering departments, in that a large number of careers make use of the same basic concepts. However, careers in computer graphic areas almost always require other skills sets of its potential employees, as most are interdisciplinary professions where computer graphics forms but one aspect of the actual job. Examples of this are numerous and range from architecture to CAD/CAM to medical applications. The specific career that I will discuss as an example of training for real world applications has been a highly sought after one for many computer graphic students in the last decade — that of technical director for entertainment applications such as film, television and games. In spite of adequate computer science theory and conceptual training, many graduates are not accepted by this industry because of a lack of understanding of what is expected, and a corresponding lack of additional relevant skills. This paper explores some of these desired skills and serves as a basis for discussion for how educating students towards an understanding of skills necessary for many computer graphic careers can be included in a traditional computer science or engineering program.*

## Introduction

One of the faster growing segments of the job market for graduating Computer Science (CS) majors over the past six years has been that of the technical director (TD) or software specialist for the entertainment industry. Students to fill these needs within the industry have been in short supply throughout this time. Many companies have had to look overseas or to related industries such as aerospace or other engineering disciplines to find the talent they required during what has been an unprecedented growth period within digital entertainment. This hiring burst of recent CS graduates, at all levels, has brought with it a range of higher salaries, which, in addition to the high-profile nature of the jobs (from screen credits to the glamour of Hollywood), has left a shortage of these same students to populate positions in more traditional fields, such as defense, or CAD/CAM applications.

Traditional Computer Science education, while it provides students with the programming skills they need and the ability to approach advanced problem solving in a logical fashion, does little to prepare them totally for new and distinctive careers such as those in digital entertainment. Even with the best Computer Science education as preparation, studios and production companies have found that there is an additional level of education that these incoming hires often need, no matter what their specific job function within the company.

The types of knowledge lacking in these hires is usually not part of, nor even considered within, a traditional CS program. It consists, in part, of competence in areas of communication and management, artistic expertise, and knowledge of production itself and the way computer tools are used within the various types of production pipelines. Can these areas be folded within a traditional CS curriculum? Should they? That depends. If the task of training students is to prepare them for future jobs, then perhaps consideration should be given to methods with which to prepare them better. Several suggestions on how to accomplish this for the entertainment industry are explored, hopefully sparking a dialog leading to an enhanced computer science education.

## The Rapid Rise of Digital Entertainment Jobs

In 1992 James Cameron had a huge box-office success with a sci-fantasy film called *Terminator 2: Judgement Day*. This film was significant in that it could not have been realized without extensive use of computer generated imagery (CGI) to create what was essentially a main character in the film. More importantly, this film was a watershed in a long range economic way — it gave studio executives a level of confidence that this technology could actually be used in film production, without exceeding the economic confines of time or budget. The year before *Terminator 2*'s release (1991) there were approximately 2000 people making their living as digital animators of some sort in the entire industry. A year after T2's success (1993), there were close to 4500! And today there are well over 10,000 digital animators and artists, including technical directors employed in the business.[1] Digital entertainment has become a significant—not to mention exciting—job market for computer science graduates searching for their first job.

This rise in numbers of jobs has been an interesting phenomenon, partly because the job market increased so fast that few schools were ready to meet the challenge of training students for employment in these new jobs. This led to a need for production companies to institute their own internal training programs to help new hires bridge the gap between the skills they had and those they needed to be successful. These programs range between a few days (for someone with previous production experience who only needs to get up to speed on minimal proprietary software) to a full year for recent graduates who have much to learn. The cost of these internal training programs has not been insignificant, but companies simply had no choice. If the dollars spent over the last few years in this area were to be added up the economic impact would be seen to be profound. And since the profit margins, especially for small effects companies who do not own their own content, are often in the single digits, internal training programs are not likely to survive if companies can find any way around them. This is especially likely given the current state of the industry, which includes a number of significant layoffs and company closings, thus putting experienced people out into the job market again in large quantities.

While it is true that the previous demand for more and more hires is currently reversing, it must be realized that the entertainment industry traditionally experiences ups and downs. This cyclic nature of entertainment is another important reason to provide students with as much knowledge as possible while they are in school. Not only will those with more knowledge be the last to be let go in the slow times, they will be preferred hires when the pace picks up over other entry-level hopefuls who would need significant on-the-job training to come up to speed.

## What They Need To Know

The jobs of technical director and entertainment software developer are not widely understood. Even within the entertainment industry the same job title can encompass a wealth of different expectations, with an increasing trend towards specialization. For example, a recent recruitment ad for Industrial Light and Magic, a division of Lucasfilm, Ltd., listed the following specialized job titles, all of which might be lumped together under the "TD" category at a smaller company: Digital Matchmovers, Envelopers, CG Commercials Artist, CG Software Engineers, and Senior Level Technical Directors. The general rule is: the smaller the "house" the more hats an employee will wear.

The skill sets necessary to be a successful technical director or software developer in the entertainment industry are also fairly wide range. While any given job may not absolutely require all these skills, knowledge or understanding of them will significantly improve the efficiency with which the job is done, thus maximizing the value of the employee to the company. The following listing of additional skill sets beyond those of traditional computer science programs is drawn from the most common requests and observations received from supervisors during production.

### Communication, Social and Management Skills

This is an extremely important set of competencies and often the ones that have the most bearing on an employee's eventual success or failure. These skills include: how to work with artists, and producers, and directors; understanding the demands and rigors of production schedules and how to work within them; coping with extremely short research and development times; keeping confidentiality; and accepting criticism in a constructive way, by using it as a tool to improve

the quality of the work It is often difficult for new employees to work long hours, not talk about what they are doing with their friends, and realize that the director is always right, even as they redo a shot for the twentieth time at one o'clock in the morning. And technical directors often require strong management skills to lead those working under their direction, including knowing how to inspire a team of people facing the above pressures to push their artistic and creative limits to maximum effect.

**Artistic Skills**

Aesthetics! The best TDs are those who understand artistic fundamentals and have developed a strong sense of aesthetics. At a minimum, TDs need to know how to value the art skills of those they work with, and collaborate with artists for the best results. But understanding the art themselves is best. A fundamental comprehension of design concepts provides a significant advantage for TDs. Most all TDs would also benefit from knowing color theory, and more color theory. It is difficult to find experts in this who understand the whole color picture, as it is taught so differently within different disciplines, when it is taught at all. CS students need to understand about color psychology, color spaces that artists deal in, and color phenomena as well as gamma and wavelengths. Lighting is another area in which TDs need artistic training. Lighting TDs are in extremely short supply, typically because those who understand how to use the lighting tools, such as Renderman™, do not usually have the co-requisite skills for manipulating light in aesthetic ways.

## Learning and Research Skills

In addition, TDs will consistently be called upon to invent something that has never been done before. The best TDs are those who can make ingenious and intuitive leaps that bring unseen visions to life. They need to understand what has gone before and how to put seemingly disparate pieces together. And they will always be expected to keep up their skills, learning or writing new software on very short time frames.

**Specific Job Skills**

Some specific topics that TDs should know include: scripting and expression writing to create or control images; modeling complex, organic objects and how continuity is kept across such surfaces; enveloping, or developing the final "skin" around an animated character; rigging characters for animation, which is setting up intuitive controls for animators to use; procedural modeling using such tools as particles or implicit surface techniques; detailed shader and texture writing, which creates the final look of an object or character; pixel mathematics, which are used extensively in compositing operations; and working with 2D and 3D spaces and their interactions, especially in techniques such as 3D tracking. It is also imperative for TDs to understand how film works—how it is exposed and scanned into the computer as well as how to correctly output digital images to it. And last but not least, an understanding of the entire production process is extremely helpful. Knowing how and why certain decisions are made for a particular scene before it ever gets to the point where the digital manipulations occur can simplify the TD's task. Add to these areas an understanding of how to decide which tool is best for which job, since CG may not always be the best choice, or when to chose one CG tool over another. For example, modeling an asteroid is easier using simple geometry and a clever shader rather than a model with ten thousand polygons, and blowing up a building might be simpler to achieve with a practical, miniature model than with CGI.

## Suggestions on How To Do This

Tying the appropriate computer science knowledge to practical applications is much easier today than it was twenty years ago. However computer science education has not changed to reflect this as much as it might have. CS has a set of core classes that are certified and endorsed by various academic and computer organizations. Contrast this with the multitudes of "Computer Graphics" programs within art, design, communication, architecture, film, tv, or media departments that have been created over the past twenty years. These more artistically-inclined programs have very little in the way of standard or endorsed "core" classes. These programs have benefited, however, from a great deal of advice from studios and entertainment companies, and have modified their programs accordingly, with more or less success. They still, for the most part, lack training in the more technical areas of expertise needed for the optimum success in the entertainment industry. Both types of programs produce students who do not have all they need.

It is tough to create a truly interdisciplinary program within most university or college environments. Not only are there scarce resources which often leads to departments protecting their domains, there really isn't enough time in a typical four year program to implement the total number of classes that would create the ideal graduate for this industry. Nor is there anything remotely like an apprenticeship program for the digital entertainment field (though this is currently being promoted to the State of California). So how can we provide ways for CS students to become more prepared for jobs in this arena, without diluting the core education they need?

One obvious way is to have more input from the entertainment industry. This increased interaction might lead to additional resources such as entertainment-sponsored training materials or innovative elective classes taught on video, the internet or via video conferencing. On-line guest lectures from top TDs at a variety of production companies could be collected and distributed to a number of participating schools. (SIGGRAPH could sponsor this as a Special Project.) Online mentoring or critiquing is another option, where an industry professional nurtures, reviews or critiques student projects using student web pages.

New teaching methods within CS is another possibility, such as special topic classes that are team-taught and co-listed between departments. Dr. Dave Ebert and Dan Bailey are teaching just such a class at UMBC in Spring 1999, called "Art and Science in Computer Animation", with ten art students and ten CS students working on collaborative computer animation projects. There could also be more emphasis on available materials or publications, such as Cinefex magazine, or various special television broadcasts, which would be valuable for those students interested in these careers. This can be done though seminar classes or through career resource departments. There are many points of intersection, especially for advanced or senior level classes that allow students to bring their complementary skills together in production-like projects, thus learning from each other. Even a few such classes or techniques as described above could go a long way towards providing students better preparation for careers in the digital entertainment industry.

[1] "The Visual Effects Industry Needs a New Production Model: Why *Titanic* Is the *2001* of Our Time", by Ray Feeney, Chuck Spaulding, and Kevin Mullican, Millimeter, November 1988, pp. 65-71.

# Human-Computer Interaction and Cognitive Psychology in Visualization Education.

**Thomas T. Hewett**
Department of Psychology/Sociology/Anthropology
Drexel University
Philadelphia, PA, USA
hewett@drexel..edu

**Abstract**

*A curriculum designed to educate those who design, develop, and deploy scientific visualizations and/or visualization tools needs to contain a strong component of both human-computer interaction and human cognition. The argument centers around two observations. The first is the rapidly increasing breadth of application for computer-based visualizations and the tools for creating them. The second observation is that increasingly problems addressed using visualizations involve representing abstract objects, entities or relationships for which there is no obvious direct physical analog. Both trends point towards an increased need for understanding and incorporating visualization needs of an increasingly diverse set of users. In addition the trends towards increased diversity of users and the abstractness of representations being created for their use virtually demand an increased understanding of psychological nature of visualizations and their effects.*
**Keywords:***human-computer interaction, cognitive psychology, visualization education*

## 1. Introduction

Instances of scientific visualization currently range over many domains of human knowledge. For example, computer-based visualizations have been used to expand and contract time and space in ways that range from the modeling and rotation of entire galaxies to the enlargement and manipulation of models of molecular structure. Similarly, visualizations have made the invisible visible in ways that allow such things as viewing the dynamic movement of the valves the human heart and magnetic resonance imaging support for microsurgery. In addition, visualizations have made the difficult easy as in the case of software that graphs equations for calculus students or that enables a research mathematician to generate and manipulate complex surfaces as an aid to solving equations.

Currently, the ease with which computers track, record, and display information is leading to a flood of new uses of information visualization. Consequently, it is easy to envision even more diverse applications of computer-based visualizations in the near future as our knowledge of how to control the computer and visualization tools grows. For example, given that work is already taking place in these areas described below it is relatively safe to predict that in the next few years we will see an increasing use of and demand for information visualizations and tools for use in such application domain areas as:

1) Biomedical engineering of human health and performance through the creation of new devices for sensing and imaging of information which can be used in the design and development of new prostheses (e.g., stents to reinforce weak blood vessels, artificial limbs with improved functionalities, etc.) and new visualizations of complex patient data (e.g., combinations of vital signs and indicators being monitored on a patient in the hospital Intensive Care Unit.

2) Environmental science work engaging such diverse problems as automobile traffic control to reduce average trip time and auto-emissions; atmospheric or oceanic modeling of diffusion and chemical transformation of various pollutants; modeling the urban infrastructure to support planing and resource allocation.

3) The utilization of geoinformation systems in coping with such problems as geolocation and way-finding, public and private land use decisions, wildlife management, and police detection and control of patterns of criminal activity.

In addition, it is also easy to imagine that computer-based visualizations will play an increasingly larger role in future human use of information technology. The complex set of technological, environmental, and social problems facing the human race virtually demand that information technology (including computer-based visualizations) be used to support life-long career activities in knowledge workers from many domains. If correct, this prediction suggests that we also need to be concerned with the development of a wide range of strategies for reskilling and regeneration of professional careers. Many of these strategies for

continuing professional education will, of necessity, come to depend more and more on the development of computer based visualizations which offer us new and fruitful representations with which to think about problems.

## 2. Why Human-Computer Interaction and Cognitive Psychology?

In all of the cases described above, visualization of the abstract will come to play an increasingly greater role. In other words, the problems to be addressed using visualizations will increasingly involve representing abstract objects, entities or relationships for which there is not an obvious direct physical analog [1]. Considering the wide range of technical subjects and knowledge areas which will contribute to the solutions of the types of problems described above, how are we to sort out which blocks of knowledge should be part of the education of developers of visualization tools and visualizations? Clearly such people will need to work in multi-disciplinary teams in order to call upon the contributions which can be made by specialists from other disciplines. Given all the technical material that a student of visualization will need to learn, and given the wide range of specialists with whom the student will eventually interact, why would anyone want to argue that human-computer interaction (HCI) and cognitive psychology are critical knowledge areas to be included in a curriculum for visualization education?

### 2.1. One Answer to the Question.

One simple answer to the question of "Why HCI and cognitive psychology?" is that there is already agreement about the importance of both topics to computer science education in general. For example, the report of the ACM/IEEE-CS Joint Curriculum Task Force [2] clearly identifies human-computer communication as one of the nine core areas of computer science. In this core topic the report authors include both computer graphics and human-computer interaction as related topics. Furthermore, the authors of the report point out that cognitive psychology is an important supporting discipline for three of the nine core areas of computer science, including artificial intelligence and human-computer communication.

### 2.2. More about HCI.

Beyond the fundamental natural affinity that exists between computer graphics and HCI, a more substantive argument for the importance of an HCI component to visualization education can be found in two simple observations. First, most computer-based visualizations or visualization tools are created with the intention of supporting some sort of human interaction with the visualization and the computer which creates it. Second, there is already in existence a reasonably clear delineation of the conceptual structure and knowledge modules which are central to the topic of HCI [3]. These resources have already proven useful in guiding the organization of major textbooks in HCI [e.g., 4] and continue to be of interest (e.g., during the last 12 months the report web site recorded over 9,000 hits, even though the report was published in 1992). Consequently, this existing catalog of knowledge modules makes it possible to explore and define the component aspects of HCI which should be considered part of an education in visualization technology.

### 2.3. More about Cognitive Psychology.

Substantive arguments for inclusion of the study of cognitive psychology in a curriculum on visualization derive from the fact that computer-based visualizations and tools are typically designed to make it possible for some sort of knowledge worker to achieve a set of work related goals. Typically, these goals involve developing the solution to a complex problem, or they involve the creation of new knowledge. Thus, designers and developers of computer-based visualizations and visualization tools need to be able to ground their designs in a knowledge of the end-user task or situation for which the visualization is being designed. Increasingly these visualizations and tools will require dealing with abstract relationships that are useful in enabling user's to think about their work.

As illustrated in [5], external cognitive representations can be powerful tools with which to think. Focusing on information visualization, Card, Mackinlay and Shneiderman [6] suggest six major ways in which visualizations can be used to amplify human cognition. Visualizations can increase the external memory and processing resources available to a user. They can reduce the user's need to search for information. They can enhance the user's ability to detect patterns in data or events. They can facilitate the drawing of some inferences through direct perception of the information rather than through more complex cognitive processing. They can facilitate the monitoring of change in large numbers of potential events. Finally, visualizations can be used to encode information in a manipulable medium.

The need to understand abstract relationships which the end-user is attempting to capture necessarily entails that the designer of a visualization, or of the tools needed to work with information visually, should be able to collaborate with the end-user in identifying the demands of the situation and in developing a knowledge representation appropriate to the way the end-user thinks, or needs to think, about the information being visualized. In other words, the designer has to develop an understanding of the problem being solved, why it is a problem in the first

place, and what the constraints that determine an appropriate visualization.

However, for individuals educated only in computer science the problem of understanding the user's task oriented needs and goals can constitute a major barrier to development of appropriate tools, etc. For example, as pointed out in [7] and confirmed by the work of Harrison, Hewett and Perline [8], visualization of mathematics has some special features that require novel visualization techniques. Developers of many existing visualization packages have tended to view their task in a holistic manner which does not take these special needs into account. What has mattered from the perspective of the developers of existing visualization tools is the accuracy and speed of display of the whole model or scene. Consequently rendering is done in a buffer storage and is invisible to the user until completed. However, it turns out that in this case accuracy and speed aren't just irrelevant to many mathematicians, accuracy and speed may be completely contrary to the needs of mathematicians. For example, a visualization of the entire surface generated by an equation is not necessarily what is relevant to the mathematician exploring it. What matters even more is the evolution of the curve through another variable (e.g., time) that creates the surface. This necessitates a set of visualization tools and a visualization that can show the surface, the curves and the growth of those curves into the entire surface.

Another difficulty with traditional computer science education lies in the fact that our understanding of human behavior is, on the whole, limited to heuristic knowledge, whereas, computer science education typically deals with algorithms. Computer scientists are educated to deal with issues such as computability and completeness. Cognitive psychologists, because of the nature of their subject matter, are forced to spend much of their time working with heuristics, working with the conditionality of certain kinds of knowledge, and investigating knowledge representation. Indeed, a large measure of what has been learned in cognitive science in general is conditional knowledge which depends upon the circumstances in which information has been collected and which can only be characterized in the form of heuristics and conditional statements.

### 2.4. An Example of a Failure to Understand Cognition and Conditionality.

Although not based in an instance of scientific visualization, this example illustrates how a fundamental misconception about cognition might negatively impact the design of a visual representation of information. It also illustrates an important relationship between what has been referred to as knowledge in the head and knowledge in the world [5]. At a University which will forever remain nameless there are classes in which students are taught about HCI. In some classes, students are given the following design guideline: "The Macintosh interface is a bad interface because there are menus with more than seven items," the implication being, "Do not design interfaces such as that of the Macintosh." The logic of the argument, both explicit and implicit, runs roughly as follows. Miller [9] argued the capacity of short-term memory (STM) is $7 \pm 2$. Exceeding the capacity of short-term memory degrades recall performance. As noted, the Macintosh has menus with more than 7 items. Therefore, an interface such as the one found on the Macintosh will be a bad interface because the menus exceed the capacity of short-term memory and will lead to degraded human performance.

There are three problems that arise from applying this guideline to this design choice. First, menu items are generally related, e.g. File, Edit, Format, or View. That is, the contents of a menu tend to form a single chunk rather than a set of greater than 7 unrelated items. Second, menu titles are generally mnemonic and so provide effective recall cues for the items within. For both these reasons, menu titles aid rather than stress the user. Additionally, once the menu has been pulled down, the contained elements are visible and are now 'knowledge in the world [5].' None of the items must be stored in working memory. Using Miller's work to proscribe the design of GUIs with more than seven items per menu is misleading at best and is analogous to claiming that a coin has a bad interface because people have trouble recalling all of the information on its face [10].

Although presented in the particular, this example is not an isolated case. Several people have made this or a similar claim about the implications of Miller's work for human-computer interaction design. But what is important about this case is that it represents an attempt to get students to design from first principles. The reason that this is difficult stems from the nature of first principles and the resulting way in which they are stated. The statement "STM capacity = $7 \pm 2$" assumes but does not state the range of cases to which it applies. Nowhere, is it made explicit that items can be either atoms or chunks; and that capacity is only an issue when external memory aids are not present. The contextualizing information that tells the designer when Miller's work is appropriate is absent and so it is hard to use the abstract principle to either guide or constrain prescriptions for action.

In other words, the designers of visualizations and the tools which are used to shape them need grounding in the fundamentals of cognitive psychology in order to be able to adequately explore the user's task or situation for which the tool is being designed. They

need to be able to identify and to deal with two types of psychological knowledge. One type of knowledge is that which will help the end-user solve the task related problem, i.e., the problem of designing an appropriate representation. The other type of psychological knowledge is that which enables the designer to understand the end-user's human cognitive performance strengths and limitations. It does little good and may even be harmful to create a visualization which does not map in some relatively natural and comfortable way to the cognitive structures of the end-user.

## 3. Concluding Remarks

In conclusion, designers and developers of tools and visualizations need to have a strong understanding of human-computer interaction and cognitive psychology. So long as it is easier to learn to write code than to learn to communicate with your life partner it will be easier to adopt and maintain a machine centered focus rather than a human-centered focus. Both knowledge of the user's cognitive performance limitations and knowledge of how to design representations which will contribute to the user's task relevant goals are essential to the successful creation of new tools or visualizations which will be both useful and usable in dealing with visualizations of abstract entities and relationships. An understanding of basic HCI and human cognition will help designers and developers to remain focused on the problems that should be solved rather than on the problems which can be solved. Solving a technically challenging implementation problem may produce the immediate gratification of having accomplished something, but it represents a trap if that solution does not simultaneously contribute to the end user's task related goals.

## Acknowledgements

The author would like to thank the anonymous reviewers for their comments on an earlier draft of this paper. Work on this paper was supported in part by NSF CISE Grants CCR-9527130 and IIS-9873005.

## References

[1] T. T. Hewett. Problem solving and computer graphics. Plenary speech at EuroGraphics '92, Cambridge, UK, 1992.

[2] ACM/IEEE-CS Joint Curriculum Task Force. *Computing Curricula 1991.* The Association for Computing Machinery, New York, 1991. (ACM Order No. 201880)

[3] T. T. Hewett, R. Baecker, S. K. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong and W. Verplank. *ACM SIGCHI Curricula for Human-Computer Interaction.* The Association for Computing Machinery, New York, 1992 (ACM Order No. 608920)

[4] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland and T. Carey. *Human-Computer Interaction.* Addison-Wesley, Reading, MA, 1994.

[5] D. A. Norman. *Things That Make Us Smart.* Addison-Wesley, Reading, MA, 1993.

[6] S. K. Card, J. D. Mackinlay and B. Shneiderman. Information visualization. In S. K. Card, J. D. Mackinlay & B. Shneiderman (Eds.), *Information Visualization: Using Vision to Think.* Morgan Kaufmann, San Francisco, 1999.

[7] R. S. Palais. The visualization of mathematics: Towards a mathematical exploratorium. Unpublished manuscript

[8] A. Harrison, T. T. Hewett and R. K. Perline. Soliton viewer: a computational workbench for soliton geometry. Under review for possible presentation at the *UIST '99* conference on user interface software tools to be held in Asheville, NC, USA, November, 1999.

[9] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review. 63.* pages 81-97, 1956.

[10] R. S. Nickerson and M. J. Adams. Long-term memory for a common object. *Cognitive Psychology. 11*, pages 287-307, 1979.

# The Human-Computer Interfaces Course at IST

**Mário Rui Gomes**, **Brisson Lopes** and **Manuel João Fonseca**
Department of Informatics Engineering, Instituto Superior Técnico
Lisboa, Portugal
Mrg@inesc.pt, brisson@ist.utl.pt, mjf@asterix.ist.utl.pt

**Abstract**

*This paper presents the main and most relevant aspects of the Human-Computer Interfaces course lectured for the 4th academic years (and fifth year as an optional course) of the undergraduate program in Informatics and Computer Science at Instituto Superior Técnico, Technical University of Lisbon.*
*The paper addresses the course contents and its pedagogical approach as well as the communication facilities used in the 1998/99 academic year to support the course operation.*
**Keywords:** *Human computer interaction, teaching and learning.*

## 1. Introduction

The main topic of a Human-Machine Interfacing (HCI) course is the design, development and evaluation of Interactive Computing Systems.
Designing and developing human-machine interfaces has become the most expensive factor in the development of information applications. Such cost corresponds today to roughly between 30 and 50% of the whole development cost1.
However, even though the above is true, HCI has been one of the less studied and less understood disciplines2.
The Design and development of HCI projects requires competencies and know-how in many concerning yet different areas such as Systems Engineering, Programming Engineering, Graphical design, Industrial Design, Psychology, Ergonomics, Project Methodologies, Interfacing Engineering (hardware), and Organisation Engineering, among many others3.
These factors demonstrate how important it is the Universities and Polytechnics include HCI Courses in their curricula.

## 2. Background

In recent years, Portugal adopted the policy to foster its national software industry. Many official bodies have concurred to this with initiatives like the PEDIP Programme, The Science Programme, The Mosaico Initiative and the Nónio Programme.
Many successful showcases in the development of Portuguese products demonstrate the success of such initiatives. The Green Book on the Information Society produced by the Information Society Mission of the Science and Technology Ministry extensively reports on the results.
Software products must present the right functionality and be easy to use in order to create a stream of products and create and sustain a solid software industry.
In Portugal one can still find large competence gaps extending as far as the most successful products. This reason is more than enough to make the main Engineering Colleges in the country to include courses in HCI in their curricula. The software industry has already acknowledged the need to focus intensively on HCI too as a leverage factor to stay competitive in the international market.
The importance of HCI goes beyond developing new systems. In recent years, there has been remarkable raise in the use of integrated applications, such as SAP R3, that show a trend to dominate the market. Consultants and experts are required in order to parameterise and localise such integrated applications so that they can be efficiently used. This work includes both functional and the user interface level. The later must be adapted to the actual end users, the staff using the applications. In Portugal, there are some 60 enterprises that specialise in Interactive Computing Systems and recognise the importance of the know-how on HCI. This spans national borders and is acknowledged by both national and multinational enterprises involved in either the development or localisation of Interactive Computing Systems.
In the United States, the National Science Foundation (NSF) sponsored a study on “New Directions in Human-Computer Interaction, Research, and Practice". The report, published in 1994, stresses the importance of HCI as one of the main areas in Information Engineering. However, the report acknowledges the many difficulties that have prevented American universities from including HCI courses in their curricula.
The NSF proposes two ways to bring about the changes it recommends in its report, which are:

- Integrate lectures on the design and development of HCI in the curricula of Information Engineering programs.
- Include HCI as one of the main components of projects assigned to students in the final (senior) year of undergraduate programs.

The report further recommends the acceptance of HCI as a critical technology by universities and recognises that the development of competencies and know-how in HCI is a fundamental leverage factor in preparing students for their future professional activities in the scope of the Information Society.
The above clearly stresses the importance of HCI from an integrated social and economic point of view in those countries wishing to promote a sustained Software Industry with a future.

## 3. Objectives

The definition of the objectives of an HCI course must be on par with market needs on skilled professionals. Therefore, students should aim at:

1. Master the principles of information presentation and user interaction, as well as, acquire the competence to evaluate the quality and effectiveness of user interfaces.
2. Master the technologies to apply HCI principles to products and the operation those technologies's enabling tools.
3. Master the realisation of HCI projects in the scope of the most common environments (MS Windows, WWW).

The HCI course lectured at IST has a weekly load of three hours of lectures and two hours of laboratory activities. A project is assigned to the students at the final stages of the semester.
The lecture time is dedicated to fulfil the first objective. The syllabus includes lecturing the basic principles that must be applied in order to achieve effective information delivery to the user. Such principles stem from mental models of information processing and are a collection of methods and techniques and good practice examples. These have been known and applied since long in areas like advertising.
HCI derived its theoretical ground from a very large set of experimental data collected over the years in the laboratory and from field experiments. In the laboratory session, students examine existing interfaces and check their conformance against the theoretical principles they learn. This activity is closely linked with evaluation, either qualitative or quantitative (usability).
The second aim of the HCI course (mastering HCI enabling technologies) targets at the analysis and proficient use of the technologies applied when implementing efficient user interfaces, the evolution of such technologies and their choice. In this, the course also aims at providing students with an overview of the available technologies, their evolution and capabilities to present information and interact with the user.
The third objective is reached by using a " learn by doing" pedagogical approach with an HCI project. The project is carried out by students in the closing weeks of the semester. The project's main target is to lead students into applying the know-how they have acquired and make it mature through actual use. A rather important point is the constant stress to be put on the pivotal role played by the end users in the development of the project. This is further enhanced by the project's requirement to conduct evaluation activities at all the project's phases using real end users for that effect.

## 4. Lectures

Lectures use the active discourse method pedagogical approach with the support of transparencies and examples of good (and bad) practice to demonstrate concepts.
Students are submitted to five unannounced mini-test papers in order to further ensure learning success and provide students with feedback. These test papers take place during the last 15 minutes of a lecture and address the lecture's topics.
There is no sufficiently comprehensive textbook spanning all the subjects lectured in the HCI course. This makes it necessary to recommend several textbooks to students, with the latest edition of the textbook by Shneiderman4 as a starting basis and the textbooks by Preece5 and Mayhew6 covering the details in many areas. This dispersion stresses the need for a new a more comprehensive textbook in HCI.

## 5. Laboratory Activities

The evolution of information systems design lead to today's widely accepted client-server architecture. One of the best features of this architecture when applied to Interactive Systems is the clear separation of information processing from the user interface it provides.
On the other side, window management systems and related APIs evolved over the times. Recent years have seen the dominance of X/Motif (and its derivatives) and MS Windows. Then the World Wide Web made its first steps in 1994/95 and experienced an exponential growth in the years that followed. At the same time, the WWW increased its information presentation and user interaction

capabilities due to the emergence of new or improved standards of the HTML, Java and JavaScript languages. This, in turn, fostered the use of the enhanced capabilities by an ever-increasing number of services and users.

The classes of services available extended to services to the citizen, education, entertainment, business, home banking, etc. Business services7 became particularly important and include such categories as advertising, direct sales and employment offer. Ho8 goes even further to propose three different business categories: product and services promotion, data and information services and business transaction services.

Because of the vast world opened up by the WWW highlighted above, the WWW role in the Information Society and the WWW' programming and interactive potential, the HCI targeted its laboratory and project activities at the WWW. These activities follow two main guidelines:

- Evaluation of WWW sites with special didactic value, using case study methodologies, aiming at developing students' capacity for criticism and their proficiency in applying evaluation methods and techniques to HCI design and implementation.
- Exploration, discovery and application of the user dialogue and information displaying potential of the HTML and JavaScript languages with the aim of promoting the skills students require to carry out their project assignments.

### 5.1. Evaluation of WWW sites

The laboratory activities involving the evaluation of WWW sites were carried out in three steps.

In the first step, groups of students analysed and criticised several proposed sites against the following criteria:

- Use of colour
- Use of fonts and font sizes
- Page layout
- Look and feel
- Navigation
- Use of metaphors

The second step had each group of students choose one of the WWW sites and redesigning on light of the findings when applying the above criteria. The number of student groups allowed more than one group to redesign the same WWW site. This was important for the third step, as it will be shortly evident.

In the third step, each group of students presented its site redesign proposal to all the colleagues attending that laboratory session. Each presentation included the group's reasons for the changes proposed. A period for questions and answers followed the presentation and all students were invited to criticise the design proposals. It was extremely interesting to watch these proceedings since the degree of interactivity and commitment was very high and there was the opportunity to address almost all the relevant aspects. Discussions among students lead to synergetic results where several proposals were combined into even better solutions.

### 5.2. Technologies (HTML and JavaScript)

The laboratory sessions dedicated to HTML and JavaScript languages had the objective of promoting the students' competence in these languages, through demonstration and exploration of the languages' concepts and constructs. This would later on ease the use of these languages when carrying out the HCI course project.

The pedagogical approach to both HTML and JavaScript was the same. It started with the presentation of the basic concepts and objects of each language. That was followed by language constructs of ever increasing complexity, always with the support of illustrative examples. Students were asked to carry out some assignments at some checkpoints. These assignments had the objective of making students apply the knowledge they had just acquired and, at the same time, provide them with self-evaluation feedback.

### 5.3. Laboratory Resources

Laboratory sessions took place in an electronic classroom equipped with networked PCs with Pentium II processors (300 MHz) and running under Windows NT. The network provided students with the access to the WWW, file server services and on-line help facilities. The later included tutorial texts full with interactive examples and on-line reference manuals. The tutorial texts were also available in packed format to allow students to take them home and learn in off-line mode.

### 5.4. The HCI Project

Students were proposed several project themes to choose from. Besides the obvious aim of carrying out an HCI project assignment, the themes proposed to the students were also aimed at developing students' awareness of the societal environment for which they will be working after graduation since the themes spanned from products for disabled persons to environment protection.

**Theme A (Disabilities)**

This theme addresses the development of an Electronic Book to help parents, social workers and therapists choose products for users with special needs and disabilities. Each Electronic Book had to describe three products with similar functionality or

objectives and provide product comparison data and selection guidance. The products could be either software or hardware products.
The actual product choice was left to the students in each project team. Students had to select a type of product and look for available products, collecting literature and examples, if any. Many browsed the WWW for this. All products picked had to have been designed for users with special needs (elderly persons, children) or disabilities (visual, hearing, motor, etc.). This requirement had the objective of making students aware that HCI solutions have a vast number of needs to address and are not restricted to satisfying normal person needs (see Figure 1).

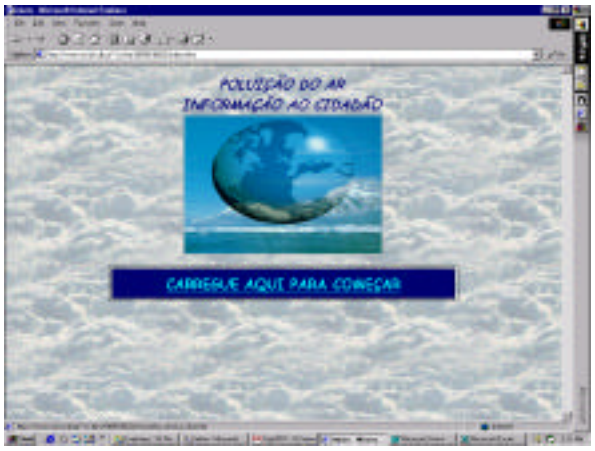



Figure 1. Example of Project A.

**Theme B (Environment Protection)**
This project theme addressed the development of services to the citizen to provide information on the environment, namely the quality of air in a region. Each project team had to select a type of end users from the following three types:

- general public
- school children
- environment experts

The service had to be designed with the provision of its deployment from information booths in mind. the service had to provide information on air pollutant measurements of sulphur dioxide (SO2), ozone (O3), carbon monoxide (CO) and nitrogenous oxides (NO/NO2) concentrations in the environment. The service also had to provide appropriate documentation.
The system had also to allow its users to select the data to be displayed, i.e., from the current day, the last week or the last month. Automatic data collection by the system was simulated with a data file containing all needed pollutant concentration values to ease system construction (see Figure 2).

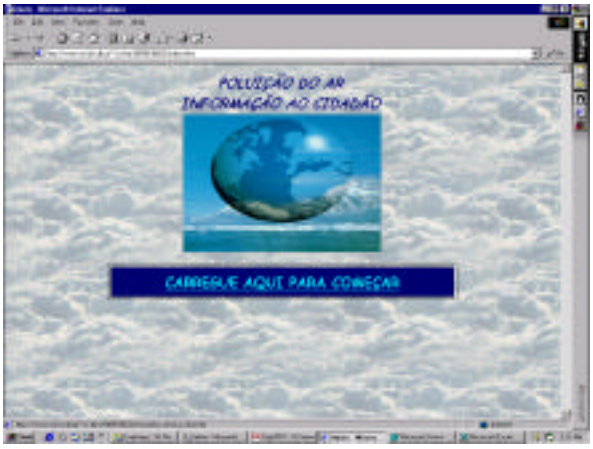



Figure 2. Example of Project B

**Others projects**
Some students proposed other type of projects some of them were accepted. The most relevant was the new site of the Informatics infrastructure that supports all the courses on Informatics, Electrical Engineering and Informatics Management, the RNL (*Rede das Novas Licenciaturas*) (see Figure 3).


Figure 3. RNL Site

## 6. Communication Between Students and Professors

One of the subjects addressed by any HCI course must be Computer Assisted Person to Person Communication. Therefore it made all the sense to address this subject with very special care and demonstrate its usage and benefits when carrying out communication between students and the course's teaching staff. This systematic pedagogical

approach made all students to have intimate contact with different technologies used today for Person to Person Communication and realise the advantages and disadvantages of each of these technologies and when and how to use them.

In the HCI course the following three Person to Person solutions were used:

- electronic mail
- newsgroups
- a WWW site

The HCI course WWW site aimed at making available to students the most complete and up to date information on the HCI course itself. At the beginning the site contained only information on the course's syllabus, its teaching staff (with e-mail and personal WWW pages addresses), how the final mark would be calculated and important dates and deadlines.

As the semester went by other information was added like:

- marks achieved by students in test papers
- themes for the course's project
- tutorial sessions support documents

The objectives of the newsgroup that was set-up were to publicise general-purpose announcements and putting and answering general questions, including questions on parts of the course contents the students might not have understood. The newsgroup aimed at providing students with the means to anticipate the answers to students' recurring questions. Students were advised to peruse through the newsgroup before coming forward with a question or remark so that they could see whether that subject had already been put and answered.

One remarkable result was the speed at which students knew of their marks on test papers. Usually, these are posted at assigned places on the walls of the administrative area. The HCI course followed this rule but, at the same time, posted the marks on its WWW site and announced the fact through the newsgroup. The result was that many students came to know of their marks well before the administrative staff had time to post them at the usual places. Moreover, the number of students looking for their marks in the later place was very small.

Finally, electronic mail was used only when the questions and matters to attend to were of a private nature and for internal communication within the teaching staff.

## 7. Conclusion and Final Remarks

The experience gained in the 1998/99 academic year allows us to say that the HCI course at IST was successful since most of the objectives aimed at were achieved. However, there is still a long way to go before an HCI course with a stable syllabus is in place and that that course answers all learning objectives set. One important point to be addressed in the future is the publication of adequate and comprehensive literature and books. The other point is the creation and use of more interactive means for demonstration and exploration of the course's contents.

## 8. Acknowledgements

The authors want express their deepest acknowledgements to Professor Lourenço Fernandes who, at the time the undergraduate program in Informatics and Computer Science was created at IST, had the perception of the necessity and the vision to create a course with the multidisciplinary characteristics the HCI course has.

## References

[1] B. A. Myers, M. B. Rosson, "Survey on user interface programming". In: Bauersfeld, P.; Bennett, J. & Lynch, G. (Eds.) CHI´92 Conference Proceedings: ACM Conference on Human Factors in Computing Systems. Addison-Wesley; pp. 195-202 (1992).

[2] G. W. Strong, "Introductory Course in Human Computer Interaction", ACM SIGCHI Bulletin, 20 (3), pp. 19-21. (1989).

[3] R. M. Baecker, et al, "Readings in Human – Computer Interaction: Towards the Year 2000", Morgan Kaufmann, (1995).

[4] B. Shneiderman, "Designing the User Interface – Strategies for Efective Human – Computer Interaction", 3rd ed., Addison Wesley Longman (1998).

[5] J. Preece, et al, "Human – Computer Interaction", Addison Wesley Longman (1994).

[6] J. D. Mayhew, "Principles and Guidelines in Software User Interface Design", Prentice-Hall (1992).

[7] J. H. Armenix, "Close Readings" of Internet Corporate Financial Reporting: Towards a More Critical Pedagogy on the Information Highway, The Internet and Higher Education, vol. 1, no. 2, pp 87-112 (1998).

[8] J. K. Ho, "Evaluating the World Wide Web: A Global Study of Commercial Sites", J. Computer Mediated Communication, vol. 3, no. 1 (1997), (http://207.201.161.120/jcmc/vol3/issue1/ho .html).

# Media Education at the Bauhaus-University Weimar

Charles A. Wüthrich

CoGVis/MMC*
Department of Media
Bauhaus University Weimar
99421 Weimar, Germany
caw@medien.uni-weimar.de

## Abstract

The rapid evolution of Media through the introduction of digital technologies, the multimedialization of computer hardware and the development of new network technologies have generated the need for new profession profiles responding to the rapid changes in this fast evolving branch. In the new Media, arts and humanities are tied to rapid technological development to create new markets and new products. This paper describes the study courses taught at the Department of Media of the Bauhaus-University in Weimar, where future Media Designers, Media Critics and Media Scientists are forged. All courses are taught in a project based way, so as to teach flexibility for a rapidly evolving sector. The new study courses represent a first attempt of definition of an education at the leading edge of technology which is entirely based on graphics, multimedia and network-based paradigms.

## 1 Introduction

In recent times, the information society is transforming from an individual or organization based archive (or repository) of information to be processed to a networked environment allowing communication between humans and within society [7]. Older broadcast-based media, which allowed essentially a one-way communication, have been completed by Internet technology, which instead allows a two way communication on the same medium. The development of computer networks, as well as the introduction of powerful graphics hardware on computers, have allowed the introduction of computer-based communication: computers have evolved from data processing units to hardware for a new medium allowing a new type of interhuman communication bringing in a new quality, and allowing multimodal location-independent communication. New powerful hardware allows the generation of shared virtual environments, where communication can happen between virtual representations of users (avatars), allowing the decoupling of the representation of the subject from his physical appearance.

The establishment of these new technologies requires new specialized study courses which differ from classical Engineering, Computer Science, Artistic and Human Sciences education [8, 4], integrating different traditional fields [5, 2]. The market requires creative and innovative graduates capable of coping with a rapidly changing industry and with the challenges of three-dimensional representation [1]. Moreover, the scientific foundations for such systems are still at their infant stage, there is no general model, and systems are developed on the stimulus of application, and are not generalizable.

The establishment of a Department of Media at the Bauhaus-University in Weimar and the introduction of two study courses in "Design of Media" and in "Media and Cultural Studies" in 1996 have awaken great interest in the German Media Community. On one side, for the first time in a German University a Department was established at the crossroad between the Humanities, the Arts and the Sciences with the purpose of forming a generation capable of researching, developing, designing and analyzing products in a fast evolving medialized world which is being created through the introduction of new graphical, acoustical, multimedia and communication technologies.

The Humanistic-Economic department component is represented by the presence of professorships in "*Perception Theory, Theory and History of Media*", "*Theory and History of Artifical Worlds*", "*Sociology*" and "*Media Management*", while Design professorships have been established for "*Multimedia Storytelling*","*Design of Media Environments*", "*Media Events*", "*Trends and Public Images*","*Interface Design*", "*Experimental Radio*" and "*Electronic Music Composition*". Instead of following the classical single medium oriented education, the design professorships have been conceived to have designers capable to utilize more than one medium for their expression. The technological component is currently represented by professorships in "*Databases and Communication Systems*", "*Networked Media*" and "*Computer Graphics*", and will expand next to include professorships in "*Virtual Reality Systems*", "*Computer Supported Cooperative Work*" and "*Computer-Based Gaming*".

The Faculty of Media understands itself first of all as a think tank and as a confrontation location. A think tank where the critics of existing and of new forms of media serve as a stimulus for the conception and for the development of new media, both as new forms of representation with traditional media as well as new types of technologies. The mission of the department is to study and criticize existing media, to research and develop new interhuman communication paradigms, and to promote the creative use of existing media.

The focus of the two study courses currently available is mainly set on the Arts and Humanities, whereby students have the possibility of receiving a strong education on the available technologies so as to be able to follow and to participate to their development as competent initiators and co-concievers. This approach is motivated by the observation that many of the technologies which were developed in the last thirty years in this field could not make a breakthrough in the market due to the lack of applications based upon them. The closeness of end users to the developers of these technologies is a key to the success of the technologies themselves. End users should be able to have a constructive exchange with the developers and conceivers, so as to both influence the development and be aware of it in time.

To complete its offer, the Department plans to introduce a third study course in 1999 on Media Systems, so as to insure the presence at the faculty also of technology developers aware of user needs. The new study course aims at the market of the rapidly expanding media, multimedia and telecommunication industry which creates technologies and systems for the new media.

*Computer Graphics, Visualization, Man-Machine Communication Group

All the study courses are organized in a project-oriented way: students do not learn a theory that in future after the diploma will be applied in practice. Instead, the study courses are organized around the "learning-by-doing" principle. This is motivated by the speed of the development of technology: the goal of studying should be here to train flexibility and the development of new paradigms, more than building the knowledge for a lifelong profession. At a very early stage, students are therefore involved in the faculty research, development and artistic activities and participate in the development, accompanying the work on the projects with traditional courses relevant for the projects themselves. Furthermore, basics are taught through introductory courses which have to be completed before gaining access to the faculty projects.

This paper is structured as follows: section 2 presents the current arts and humanities study courses in Media, section 3 presents the new study course in Media Systems which completes the offer of the department. In the last section, perspectives for graduates are traced and some conclusions are drawn.

## 2 Technological Education and Multimedia for the Arts and Humanities

In the education of artists and of human scientists working with the media, a twofold goal has to be accomplished: on one hand, traditional media, such as film, broadcasting, print media have to be taught; on the other hand, students have to be prepared to the use of new media for their projects, and to successfully cooperate to the development of new technology. The contribution of artists to the technological development is essential, since end user feed back is of extreme importance.

Since the media industry develops itself at the same time towards multinational media corporations and into tiny services deliverers, a strong component of economical foundations have to be taught. This is achieved through the teaching of basic economical courses as well as special courses on the economics of startup companies.

Both study courses are centered on projects. Under individual tutoring, groups of students are assigned a theme each semester which they have to develop. Project results are then publicly presented at the end of the semester. This way of teaching focuses the study on the production process, from conception to execution.

Artists and Humanists are known to suffer of the typical computer device fear. Moreover, most students approach the study course attracted by the contents of media, not by the technology driving them. However, in a high-tech environment, it is very difficult to avoid being confronted with the glitches and problems of the technology. This is why a comprehensive technological component, including three basic courses on the foundations of Mathematics, Computer Science and Media Systems (network and broadcast technology) are taught. Additionally, fundamentals of programming languages, of operating systems, of communication networks, of computer graphics, of multimedia, of databases, of image processing, of sampling algorithms and of interactive technologies are taught on a regular basis. Students are also confronted with the basics of hardware development.

They are allowed to learn broad and comprehensive foundations of the technologies they will use during their lives. Teaching is not application-oriented or software oriented as in most institutions, but concentrates on principles, so as to avoid an application-specific approach, which has no sustainability on the long term. This ultimately trains towards flexibility. Additionally, students are allowed to visit introductory courses on the main interaction, animation, multimedia, modeling, rendering, image processing, compositing, sound editing and web based techniques so as to understand the principles of tool based work in the media. This is done through hands-on intensive tutorials aiming at real production.

The aim of this approach is to promote the competence in students so as to allow them to dialogue with other people involved in the media industry, whatever their personal inclination is, be it towards the arts, technology or on reflection on the media, and to promote the understanding of the basic technologies available.

The job perspectives of the students are improved considerably by the in depth technological education which improves their chances in the work market. The traditional gap between research and development, and end users is minimized. Arts and humanities students cultivate in the non-technological study courses (which still constitute the main part of their study) their characteristics as artists and critics, and pursue careers in the creative or humanistic field, but are able to understand the potential development in technology and to accompany it from its very early stages. This is a decisive advantage with respect to traditionally schooled students in the arts and humanities.

## 3 Media Systems: a New Paradigm for Multimedia Education

Media Systems are systems that have an influence on the Media, i.e. that support and allow communication between people. This includes broadcast, networked and point to point communication. Setting the accent on systems underlines that focus is set on complex entities, rather than individual hardware components thereof. Media Systems are therefore complex systems used to communicate.

The study course is addressed therefore to all persons that want to acquire technical/scientific knowledge in the new media, and want to pursue a technical career in Digital Media, Gaming and Entertainment, Computer Graphics, User Interfaces, Animation, Image Processing, Visualization, Networks, Computer Supported Cooperative Work, Teleactivities, Virtual Reality and Digital and Multimedia Publication, and that want to participate to the technical conception and to the development of the expanding multimedia, entertainment and communication industries. The study course builds ultimately professionals that are creatively capable of conceiving systems that revolutionize the way of thinking and of communicating between humans, and that favour new forms of exchange between them. Furthermore, since the study course is integrated in a faculty where a wide variety of study profiles for the media industries are present at the same time, it prepares through dialogue and discussion with the creative and humanities faculty components to cooperation with all professionals of the media industry and research.

The study course global duration is four years, subdivided in two major slices of two years each.

### 3.1 The propedeutic years of the study course

The rapid developments in the multimedia and telecommunication industry require a total rethinking about how a scientific study course should be organized. Technology evolution is so fast, and new application areas and environments develop so quickly, that it is hard to predict what an optimal study curriculum is for the field. What are needed both in industry and in science are flexible persons that are capable of working in projects with minimal introduction times, and capable of working as a task force on a theme for a period of time in an innovative way, then move to new projects.

Instead of the traditional approach based on trying to cover during the study the whole spectrum of related matters until "universal knowledge" of the matter of study is reached, the study course is therefore organized around the concept of research project, and tries to teach "high-tech" on a learning-by-doing fashion. Of course, it is not possible to renounce completely to introductory courses, which

provide the tools for scientific reasoning. However, introductory courses should be kept to a minimum, so as to leave as much room as possible for project-based work.

From the point of view of content, the study course is centered on the idea of a system. Basic courses are taught during the first two years. They serve three main purposes: first they provide the mathematics and computer science basics for mathematical reasoning and for programming, second they provide an overview of system modeling which is relevant for the field and which is available in science. Third, the study of available media technology, including traditional media, is provided as a foundation. The modeling of complex systems requires the mastering of all possible modeling methods, which are then taught in the basic courses.

More in detail, the Media Systems introductory courses are constituted by the subjects listed in Table 1. Purpose of these introductory courses is to provide the student with the capability of abstracting and understanding complex systems and their components. The numbers to the right of the tables indicate the number of semester weekly hours of course during the study course, which have to be multiplied by 1.5 to obtain the credit points in the ECTS[1] given by the single units. Computer Science and Mathematics rudiments are built through the courses listed in Table 2. All the above mentioned courses have to be absolved in the first two years of the study course.

Note the centering of the subjects on the concept of systems and of mathematical modeling, and the relatively light Computer Science and Information Theory contents with respect to traditional Informatics and Electrical Engineering. The faculty of Media of the Bauhaus-University strongly distinguishes these two subjects from Media Systems, the latter involving more analysis and modeling capabilities than the first two, this in order to master the complexity of the systems to develop.

In order to prepare the students to the last two years, where the main part of the time will be spent in research projects, students are allowed to choose during the second year two labs where they can participate to the department's research activities in two chunks of 8 hours per semester (in study projects) so as to gradually familiarize with the research activities. Here, students have to bring their contribution to research, starting from the entry level of development and get acquainted with the methodology of study of the second part of the study course.

### 3.2 The second part

In the second part of the study course students have to work on the department research projects. Each semester, the student has to work in one of the laboratories of the faculty at a research project which is assigned to him at the beginning of semester. This means that students will have the possibility of working on four research projects in the last part of his study course, for a total of 16 hours per week presence at the lab each semester, which gives them 24 credit points if the project is concluded and defended successfully.

The department is currently setting up laboratories centered around the research themes listed in Table 3. Additionally to the work on projects, students have to get credits for accompanying courses in the quantity of 4 weekly hours per semester, giving them a total of 6 ECTS credit points per semester. The courses can be chosen from the advanced courses list, which are variable offers of the faculty, depending on the projects available, but which might include the course catalog listed in Table 4 The aim of extensive work on the research projects, and of the theoretical foundations which are delivered through the courses, is to train the students to be able to take up a research project having only basics knowledge and acquire during the development of the project the necessary foundations on the project themes so as to deliver research results.

In the course of their study, students enter their first projects (already in the second year) as executors, and acquire more and more experience, creativity and dialectic capabilities as they become more mature and gain broader knowledge, participating more and more to the conceptional phase and to bigger parts of the projects, until they are able with in the final thesis to take a project and develop it by themselves. This of course requires extensive individual tutoring, which has to be performed directly by faculty. In return, faculty is allowed to concentrate most of the time on its research themes and can profit from the work of the students. Through an admission exam and through a personal interview of the applicants selection of suitable students for such a non-conventional study course is achieved. The individual nature of the study course requires not only application, but also particular creativity of thought.

[1]ECTS stands for the European Credit Transfer System, which is the official exchange system among European Universities and provides transferable credits to exchange students.

## 4 Perspectives and Conclusions

According to the last German Federal Ministry of Economics report on the Information Society [3] the average rate of growth of the entertainment, telecommunication and multimedia industry in the early nineties has been over 20% yearly. This growth rate has been rapidly increasing since, and creates an immense demand on specialized personnel, in all sectors, be they in the creative, organizational or technological sector. Graduates from flexible study courses at the leading edge of research stand excellent chances for conducting an interesting work life. Moreover, their flexibility "preprograms" them for continuous education, in a world where it is less and less possible any more to learn at the university a profession which will last a whole life.

With its three study courses centered on research, on the applications and on the critics of Multimedia and Network technology, the Department of Media at the Bauhaus-University in Weimar is an unique example of cooperation among people originating from traditionally distinct academic paths. This reflects in the education, which centers, of course, on the special skills required from scientists, artists and humanists, but completes them with elements which are necessary to ensure the capability of all three components to dialogue with the other participants to the technology development and usage. Furthermore, with its transdisciplinary approach, the department follows the path traced by the interdisciplinary research group at Xerox Palo Alto Research Center (PARC) in the early seventies [6] which developed the mouse-based desktop user interface, adapting the approach to respond to the needs of a broader subject.

The applications for access to the new study courses state the interest of youth for the perspective careers involved, and are currently beyond 1000% the number of admissions available in all study courses currently available.

## Acknowledgments

Special thanks are due to the people who contributed with their support and discussions to the contents of this paper: Lorenz Engell, Peter Hupfer, Gerd Zimmermann, Grit Thürmer, Claus Pias, Bernd Schalbe, and Günther Schatter. A special thank is also due to all functionaries at the Ministry of Research, Science, and Culture of the Free State of Thuringia who supported our Department at various stages of its development.

## References

[1] E. A. Abott. *Flatland.* Dover Publications, Toronto, Ont., 1992.

[2] J. D. Foley. The convergence of graphics and imaging. *Computer Graphics Forum*, 17(3), 1998. Keynote Speech at Eurographics '98.

[3] J. M. Harms. Computertechnik, Telekommunikation, Unterhaltungselektronik und Medien wachsen zusammen. In *Die Informationsgesellschaft: Fakten, Analysen, Trends*. Bundesministerium für Wirtschaft, Bonn, 1995.

[4] W. A. Ross and M. W. McK. Bannatyne. Creating a 3-dimensional paradigm as a foundation for graphics programs in higher education. In *WSCG'99, Proceedings of the Seventh International Conference on Computer Graphics, Visualization and Interactive Digital Media '99*, pages 510–516. University of West Bohemia, Plzen, Czech Republic, 1999.

[5] Stanford University. Image system engineering. `http://www-ise.stanford.edu`.

[6] C. P. Thacker, E. M. Craig, B. W. Mapson, R. Sproull, and D. R. Boggs. Alto: A personal computer. In D. Siewiorek, G. Bell, and A. M. Newel, editors, *Computer Structures: Readings and Examples*. McGraw-Hill, New York, N.Y., second edition, 1981.

[7] D. Tsichritzis. The dynamics of innovation. In *Beyond Computation*, ACM 50th Year Anniversary Book. Copernicus Springer Verlag, 1997.

[8] D. Tsichritzis. Reengineering the university. *Communications of the ACM*, 1999. In print.

| | | |
|---|---|---|
| Media Technology | Broadcast and Network Technology,<br>Communication processes and standards. | 8 |
| Psychophysiological Systems | Perception Theory, Human Senses, Psycho-physiology | 4 |
| Physical Systems | Mechanics, Dynamics, Electromagnetism, Waves | 6 |
| Logics and Discrete Systems | Logics, Semantics, Discrete Mathematics | 4 |
| Concurrent Systems | Concurrent and Distributed Systems and Processes | 4 |
| Fuzzy Systems | Analog and Fuzzy Systems,<br>Non-quantifiable Systems (Quality, Expressivity) | 4 |
| Stochastical Systems | Statistics, Probability theory, Empirical Methods | 4 |
| Projecting Methods | Theory and Methods of Design | 4 |
| Cultural and Social Systems | Sensemaking Systems, Symbol Systems<br>Narration, Social Systems | 6 |

Table 1: Media Systems introductory courses

| | | | |
|---|---|---|---|
| Introduction to Computer Science | 4 | Higher Mathematics | 3 |
| Device Architecture | 2 | Numerical Mathematics | 3 |
| Algorithms and Data Structures | 2 | Discrete Mathematics | 3 |
| User Interfaces | 2 | Functional Analysis | 2 |
| Operating Systems | 2 | | |

Table 2: Computer Science and Mathematics courses

| Title | Description |
|---|---|
| Tomorrow's Realities | Virtual Reality, Computer graphics, Computer-augmented reality,<br>Real-time systems, Visualization, Digital sound |
| Learning and Teaching | Teaching at distance, Tele-teaching, Multimedialization of courses,<br>Continuous Education |
| Storytelling | Narrative techniques, Esthetical processes, Sensemaking systems<br>Information presentation, Computer Animation |
| Entertainment | Games, Computer supported Gaming, Media-based Entertainment |
| Work | Computer Supported Cooperative Work (CSCW), Tele-working<br>Work organization of tomorrow, Telematics |
| Interfaces | 3D-Interfaces and -I/O, Haptic Displays, Immersion,<br>Human perception |

Table 3: Research themes for the department laboratories

| | | | |
|---|---|---|---|
| Game Theory | Geometry | Computer Graphics | Multimedia |
| Storytelling | Image processing | Image Recognition | Global Illumination |
| Typography | Virtual Reality | Digital Television | Sound Processing |
| Colour Theory | CSCW | Learning at distance | Telematics |
| Ethics and Media | Empirical research | Visualization | Animation |
| Electronic Communication | Electronic publishing | Interaction | Creative Conception |
| Ergonomy and work science | Database Management Systems | Operations Research | Media Cultural Studies |

Table 4: Optional courses for the second part of the study course

# Defining Interactive Multimedia Design Education: Expanding the Boundaries

**Bonnie Mitchell**
Bowling Green State University
Bowling Green, Ohio, USA
bonniem@creativity.bgsu.edu

**Abstract**

The future of interactive multimedia design education depends upon the disintegration of discipline-based boundaries within the higher education system. Disciplines outside of computer science and computer art contribute significantly to the understanding of the cultural and social implications of the medium. A cross-disciplinary approach enables students to explore new ideas that inform their understanding of the implications of multimedia production on the individual and society. The evolution of communication design has reinforced the need for an understanding of human behavior and representation. Because functionality no longer takes precedence over appearance in WWW and CD-ROM design, students must have a strong understanding of the principles of art and design as well as programming. The boundary between content provider, visual designer, and software designer is blurring therefore reinforcing the need to explore the relationship between art, psychology, computer science, popular culture, software engineering, and the social sciences in a meaningful way and integrate these connections into the curriculum. Only then will higher education begin offering interactive multimedia design education that will enable students to influence the direction computer graphics and interactive techniques in new and profound ways.
**Keywords:** curriculum, interactive multimedia design, cross-disciplinary.

## 1. Introduction

In the past decade, academic institutions have found themselves inadequately prepared to cope with the rapid changes brought about by the cultural appeal of computer graphics and interactive multimedia. Ten years ago, who would have thought that URLs would be printed on cereal boxes, and teenagers would prefer online chat sessions to TV? A life enriched by video games, special effects in film, and the World Wide Web often entices students to seek out academic institutions that are able to prepare them to become interactive communication designers and/or technicians. To provide a well-rounded education in interactive multimedia design, discipline-based degrees should be rethought and new multi-discipline fields developed. This idea transcends the common practice of cross-disciplinary curriculum requirements and focuses on the development of hybrid courses and initiation of academic de-categorization. The development of a new interdisciplinary structure could enable flexible adaptation thus encompass the changing needs of students entering the interactive multimedia design field.

## 2. Collapsing the Structure

We are currently immersed in the technological and ideological transformation of global cultures. Not only is information on virtually any topic available to anyone with access to a networked computer, but the hierarchical structure controlling information production and publication is collapsing. In a world where scholarly texts and student productions co-exist side-by-side, we find that our expectations of and experiences with information acquisition have changed. According to Margot Lovejoy, "In the Internet, we have the commingling within a vast global community of international resources of both high and low cultural appeal. Some see the screen as a linguistic leveling device which flattens and devalues language. Others see a rich evolution of language occurring on the Web, not as it evolves in novels but as an experience of communication beyond the usual hierarchies of mediated publishing." [1] Not surprisingly, students often rely on the Web for information and we have seen URLs replace print references in their research papers. They realize the value of easy access to this cornucopia of diverse information and want to be involved in the production of information as well a consumer of it.

## 3. Revisiting the Past

Along with the collapse of the information production hierarchical structure, the method and means of conveying information have begun to take divergent paths. Whereas we see a resurgence of the dependence upon visual communication, the means of communication has evolved in a direction that defies the foundation upon which communication was founded. The earliest tangible examples of communication were often found on cave walls in the

form of depictive images. In these ancient cultures information was exchanged directly by means of visual, auditory and gesture-based reciprocation. This interactive interchange was confined to the physical locale. Over the years, cultures have become less and less dependent on images and have relied on the abstraction of symbols such as text to convey thought. With the advent of the book, information became tied to a tangible material that enabled the dissemination of ideas to a wider audience. Books became permanent records of knowledge and were often considered the authority. Cultures developed elaborate campaigns to maturate text-based literacy, while de-emphasizing the visual image. Interaction, inherent in pre-text based information exchange, was severed, leaving the receiver of information alone without recourse. In "Being On Line: Net Subjectivity", Doctress Neutopia states "There is something dead about the word. After it is written, like a photograph, it is there as a reflection of past thought, there to remind us of a moment in time when the inspiration possessed the writer"[2]. The transient nature of the World Wide Web suggests there is no authority and information is constantly in flux and very much alive. Knowledge becomes a process of exploration and evolution, not a product.

## 4. Crossing the Boundaries

The re-emergence of an emphasis on visual, interactive exchange of information in the late 20th century generates the need to expand our interpretation of what it means to communicate. The Internet has become a fertile ground for experimental forms of social exchange. The enabling of immediate reciprocation that transcends geographic space challenges the formative structure of exchange yet broadens the appeal. Abraham, Meehan, and Samual, in the book "The Future Ain't What it Used to Be", stated "Online culture is, more than anything else, about satisfying our fundamental needs for dialogue, discourse, and debate"[3]. In the past, personal interaction within local communities nourished the culture. Today the interchange within on-line culture supplements and often surpasses local exchange enabling not only dialogue and debate but also cross-cultural transmutation of ideas. Understanding the visual image plays an important role as students begin to understand cultural differences and explore new modes of exchange.

## 5. Education's Responsibility

With the responsibility of preparing today's youth to function effectively within a cross cultural, interactive, visual environment, educators have an obligation to address the concerns of the changing state of information consumption and production. In many countries, we are seeing new academic programs emerge and existing structures expand to emphasize the production of visual information using computer graphics. The technical aspects involved in the production of images and interaction using computer technology typically consume academic attention. Little, if any, time is devoted to how these images are transforming our culture. The essence of interaction and information design is overlooked in favor of the technical means of production. We teach programming which focus on software development and art classes where students design Web pages, but where do we teach courses that integrate the mathematical, artistic, and cultural aspects of new media and communication design? A cross-disciplinary approach to interactive multimedia design education will enable students to produce works that not only stimulate our senses, but also challenge us intellectually.

## 6. The More We Know, the More We Can Say

Whether we design interfaces that empower or entertain, we are attempting to create screens that convey information to the user. Visual communication, whether employed in scientific visualization or in art, is often more successful when knowledge of human perception is applied. Developing an understanding of the structure and function of an image will enhance the student's ability to communicate the essence of a concept effectively. One of the strengths of computer graphics is its ability to render obscure and abstracted data. It is able to do so only through detailed analysis of the data that makes up the image. Knowledge is power. As stated in the classic computer science proverb, *garbage in – garbage out*, the quality of the input heavily influences the quality of the output. Students with a deeper and broader base of knowledge are able to produce more sophisticated and engaging interactive computer graphic products.

## 7. Communicating Ideas

In the Computer Art curriculum at Bowling Green State University students were given a JavaScript assignment to develop WWW pages that investigated the concept of time as it relates to human perception (see Figure 1). The idea was discussed as a class, thus generating initial approaches to the subject. Sub-topics were identified and students selected which theme they wanted to research further (see Figure 2). Students were given time to search the Internet, go to the library, find images, and think about a specific approach they wanted to take with the project. The following

week, the students met with a small group of other students that selected a similar theme and each student described the specific idea they wanted to convey visually and interactively in their JavaScript project. Often, the topic involved abstract content such as Mat Brandeberry's JavaScript project that explored the idea of duplicity, doldrums, and decisions (see Figure 3). Rudolph Arnheim in "Art and Visual Perception" states that "The relationship between intellectual knowledge and visual representation is frequently misunderstood. Some theorists talk as though an abstract concept could be directly rendered in a picture; others deny that theoretical knowledge can do anything but disturb a pictorial conception" [4]. Students grappled with the idea of visually communicating these abstract ideas within an interactive environment. They also considered how interaction could effectively enhance the concept they were trying to convey. The technical aspect of how to develop an interactive exploration of their idea using JavaScript was not highlighted until the research was well underway and the idea was solidified.

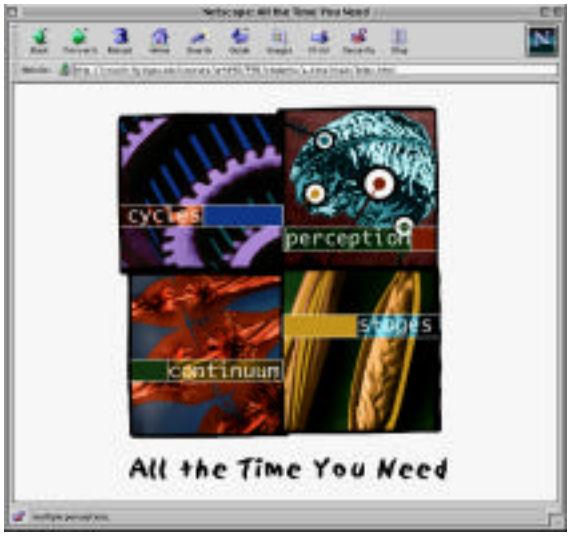



Figure 1. This is the start of the JavaScript Project.

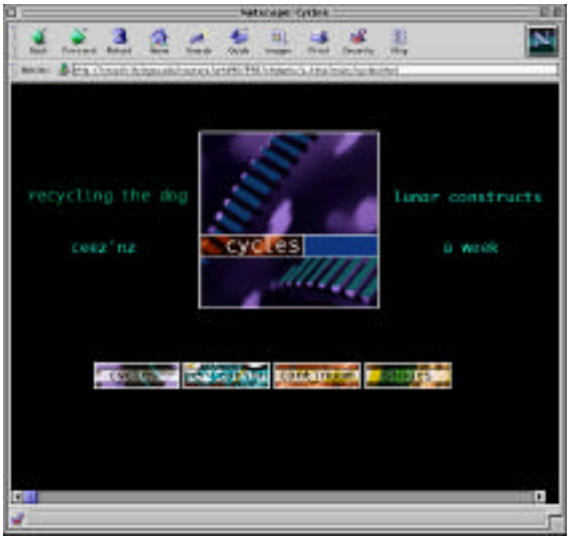



Figure 2. The topic was subdivided into themes.

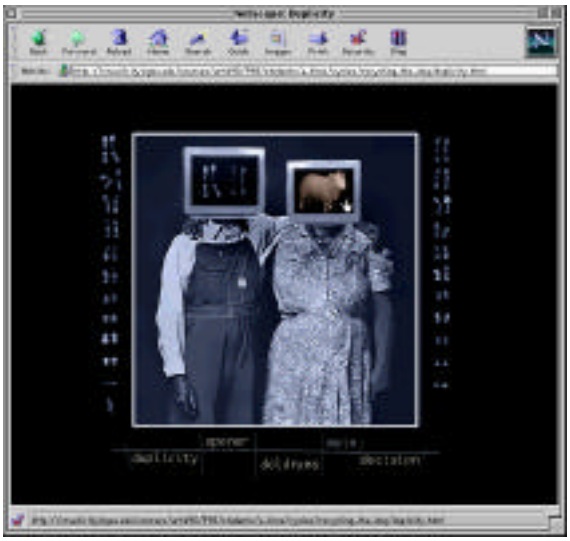


Figure 3. This approach explored the concept of duplicity, doldrums, and decisions.

## 8. Interface Design - Human Factors

As the students moved from idea to digital image production to interface development, issues related to human factors and effective visual layout arose. According to Brenda Laurel, "The working definition the interface has settled down to a relatively simple one - how humans and computers interact - but it avoids the central issue of what this all means in terms of reality and representation" [5]. Students needed to decide whether an exploratory or intuitive interface would be more effective in communicating their idea. They also discussed how the experience they were providing for the viewer translated into meaning. If a student decided to explore the concept of time, technology, and

intimidation, an interfaced designed to confuse and belittle the user would communicate more than any collection of images. Yet, if the student wished to empower the user, an interface that translated technology into human terms would result in a more relevant experience. The design of the interaction, interface, and visual communication should be seamlessly interwoven into the expression of the idea. Interdisciplinary educational experiences prepare students to make meaningful connections between disparate fields of study.

## 9. Interaction Production Education, or Technology is Never Enough

One of the obstacles that prevent the development of an integrated interactive multimedia design curriculum is that it does not fit nicely into any currently defined field. In an essay entitled, "The Design of Interaction", Terry Winograd states, "Although computers are at the center of interaction design, it is not a subfield of computer science... As well as being distinct from engineering, interaction design does not fit into any of the existing design fields. If software were something that the computer user just looked at, rather than operated, traditional visual design would be at the center of software design. If spaces were actually physical, rather than virtual, then traditional product and architectural design would suffice"[6]. Often we see academic programs such as Computer Art, Computer Science, Information Technology, and Public Communication offering classes in interactive multimedia, each with its own unique approach to the subject. Since interface and interaction design relies heavily on technical proficiency, artistic talent, and an understanding of human behavior, all aspects of the field are rarely explored in any depth in any single discipline. Nicholas Negroponte believes, that "...we are moving away from a hard-line mode of teaching, which has catered primarily to compulsive serialist children, toward one that is more porous and draws no clear lines between art and science or right brain and left" [7]. At the university level, this method of teaching is slow in developing. Occasionally computer art professors teach basic programming and the mathematical terminology necessary to understand 3D modeling, but it is rare to find courses in computer science or information technology that also focus on the principles of art and design. It is even more rare to find interactive multimedia courses that address issues of human perception, behavioral science, global studies, and popular culture.

## 10. Technical Proficiency

Within the Computer Art classroom, the students learning JavaScript to convey their ideas in relationship to time were forced to rely on a small subset of the programming language with very little knowledge of the concept of object-oriented programming. Because time was devoted to critique of the artwork and interface design as well as discussion, very little time was left to investigate the language. Without prerequisite knowledge of programming languages, the art students were stumped by concepts as fundamental as variables. Ten different JavaScript examples were presented and thoroughly discussed. These examples were made available to the students via the Internet. Needless to say, very few student projects code that significantly differed from the example code distributed. Did they understand the code? Yes, to a very limited extent. Did they understand the potential of the language and how to manipulate it to solve unique problems? No. As a beginning level assignment, the project was successful in revealing the artistic potential using computer graphics, interface design, interaction, and JavaScript, but to fully understand the medium, more programming experiences would be necessary. The students also were not given the opportunity to study the impact their work had on the global viewer nor how it fit into popular culture.

## 11. A New Generation

After completing the project, the three student volunteers developed the interface that linked the individual projects together under one unified structure. The project was published on the WWW. Although this was a student project, it was available for the world to peruse. For many students, this was the first time they had published a work of art. It is important to note that the average university student has a very different relationship to technology than their professors. Most of the students in the Computer Art course were between the age of 19 and 21. On the average, they were 5 years old when the microcomputer became widely available. Most had grown up with computers in their home. They do not remember a time when video games were not popular. Fax machines, photocopiers, and answering machines have always existed according to their recollection. They were in middle school when the World Wide Web hit the mainstream. Email, chat rooms, and URLs are as much a part of their lives as TV was a part of ours. The immediacy of real-time interaction with remote parties via the Internet is not new to this generation; it is an integral part of their life. Yet, communication that transcends cultural boundaries still has its novel appeal. As educators

we need to be aware that as consumers of information on the Internet, our students are often quite adept. Yet as producers of information, they are often very inexperienced.

## 12. Global Communication

Although publishing interactive information that reaches a global audience is exciting for the student, the lack of physicality, and the cross-cultural implications should be addressed in the curriculum. Creating interactive work that accepts or monitors responses from global visitors enables the student to begin to understand how the work reads in a cross-cultural virtual world. Oftentimes a student's vision of the world is based on their personal experiences. Lectures, books, and TV may present alternative views, but reality still exists within individual empiricism. Using the Internet to initiate communication and solicit responses to student interactive multimedia works, enable the student to begin to understand the implications and interpretations of their work. The World Wide Web enables the student to build pages and publish their work. By also including feedback forms on the page, the work not only becomes public but also is subject to global critique. A narrow, uninformed view of a topic or a design that does not compliment the content will quickly have its weaknesses revealed by the global community. It is often the case that critiques come from many different cultures and from people with various professional interests. Within a single discipline based course, this global feedback enables the student to integrate ideas that transcend geographic and discipline-based boundaries.

## 13. New Solutions

Within the academic community we are searching for a home for Internet-based interactive multimedia within existing departments and schools. Yet the very nature of the medium is cross-disciplinary. To be successful, an educational experience in interactive multimedia should span across academic boundaries and possibly include sub-topics within computer science, computer graphics or computer art, traditional art, information science, social science, psychology, business, and philosophy. Students also need to be more aware of the global economy and cultural similarities and differences. Currently, in most universities, students must select a major in one of the above areas and hope to take courses in the other areas. Contemporary solutions to providing a well-rounded experience in interactive multimedia have often fallen into four categories: new programs, experimental courses, advising, and prerequisite courses. Academic institutions are often reluctant to establish new programs therefore faculty initiatives are essential to the educational success of students focusing in this area. Creative solutions come from the realization that new media has changed the way we, as a culture, process information. Broadening our own experiences to encompass new multi-disciplinary approaches to our own understanding of the issues will inevitably translate into attainable solutions.

## 14. The Team Teaching Dilemma

One of the answers to the problem is to provide team-taught cross-disciplinary courses that integrate design, programming, and human factors as well as information from other supporting fields. For numerous reasons, this solution is often unattainable. Although many universities pay lip service to the benefits of interdiscipline studies and team-teaching, faculty are often discouraged by the lack of support in terms of course load reductions, salary compensation, and access to information and facilities. Faculty eager to cross boundaries often find themselves without the support of the university and unable to initiate new course development without compensation and approval. Supportive departments often find that budgetary structures are not flexible enough to compensate two professors fully when together they teach only one course. Also, it is common that team taught courses require more preparation because of the ongoing negotiation involved between professors. Despite these obstacles, faculty occasionally sacrifice summers and adequate pay to develop experimental courses and integrate new cross-disciplinary content into their interactive multimedia courses. These heroic efforts have not gone unnoticed. The introduction of multi-disciplinary study, complimented by multiple viewpoints, increases the students understanding of the medium in which they work.

## 15. New Approaches

Although team teaching enhances the educational experience of both the student and the faculty, the discipline-based boundaries within the university still prevail. Art and Computer Science departments operate independent of one another and proudly educate students by delivering the content they know best. Courses that do not fit the idealistic vision of what art or computer science should be are often short lived or never approved. Many universities have developed departments or programs that attempt to merge the various interests involved in communication technology education. With titles such as Multimedia Technology, Information Science Technology, and Visual Communication Technology one would think that an aspiring interactive multimedia producer would be able to

obtain a well rounded education in the field. This is sometimes true, yet not always. These new disciplines also tend to become exclusive rather than inclusive and course content follows a prescribed direction that may or may not include the multi-faceted field of interactive multimedia development. Imagine the possibility of developing a department that employed art and design faculty, computer scientists, popular culture researchers, architects, social science faculty and experts from other supporting fields. The goal would be to make connections between the various disciplines and tie it all together using computer graphics and interactive multimedia development as the catalyst. This cross-disciplinary approach to education would pose challenges as the various fields seek out common ground and merge interests. The rewards would reveal themselves not only in the educational outcomes of the students but also in the intellectual growth of the faculty as they broaden their experiences and modify their single discipline approach to education. In an actively evolving curriculum such as this, it is likely that the students would not only be the recipient of ideas but also engage in the process of creating the connections between disciplines.

## 16. Conclusion

As more and more students enter academic institutions with the desire to work with interactive multimedia, the demand for a cross-disciplinary approach will increase. The preparation necessary to enter the field will necessitate new disciplines or possibly a destruction of the academic walls that prevent students from focusing in a multi-disciplinary direction. There will be more demand for a team teaching approach and a broad-based curriculum. Students have been, and will continue to challenge existing degree requirements to accommodate their desire to integrate computer science, art and other supporting areas. The development of a diverse educational approach promises to reveal exciting new opportunities not only for students but also for faculty and the institutions themselves. The knowledge and experience that comes from multiple viewpoints and various fields of study will challenge the students to become independent and critical thinkers. Students involved in a fully integrated cross-disciplinary interactive multimedia design educational experience will gain the understanding and power to take interactive multimedia to new exciting directions. This new generation of communication engineers and artists will inevitably alter the course of this important medium thus forcing academic institutions to again re-assess and refine their approach to providing interactive multimedia education.

## References


[1] Lovejoy, Margot, *Postmodern Currents: Art and Artists in the Age of Electronic Media*, Second Edition, Prentice Hall, New Jersey, 1997.

[2] Avillez, Martim, *Being On Line: Net Subjectivity*, Lusitania Press, New York, 1996.

[3] Abrahamson, Vickie, Meehan, Mary, Samual, Larry, *The Future Ain't What it Used to Be*, Riverhead Books, New York, 1997.

[4] Arnheim, Rudolph, *Art and Visual Perception: A Psychology of the Creative Eye*, The New Version, University of California Press, Berkeley, 1974.

[5] Laurel, Brenda, "*Computers as Theater*", Addison-Wesley Publishing, New York, 1993.

[6] Winograd, Terry, "The Design of Interaction", Beyond Calculation: The Next Fifty Years of Computing, Copernicus, Springer-Verlag, ACM, New York, 1997.

[7] Negropone, Nicholas, *Being Digital*, Vintage Books, New York, 1995.

# MODULARITY: TH E KEY TO COURSEWARE RE-USE

**José Carlos Teixeira**
Dep. Matemática – Fac. Ciências e Tecnologia da Univ. Coimbra – Portugal
teixeira@mat.uc.pt

## ABSTRACT

By courseware modularity we mean the possibility to build complete courses from many small course modules fitting together. In contrast to puzzle pieces, course modules should be designed to fit into many different situations, allowing different courses to be produced. This goal introduces a lot of problems to be solved, like organisation (hierarchical) of the material, formats' compatibility, clear separation between contents, structure and presentation, …

To allow an effective *multiple-use of courseware* (use of the same courseware entities by many users), as well as its *re-use* (use of the same courseware modules in the production of different courseware entities), the courseware production should be performed in small learning units that keep courseware highly modular and self-contained. Problems with the didactic design should not omitted.

The *adaptability* (meeting individual learning characteristics and learning needs) is also a reason to support the modular approach. Nevertheless, development is needed to ensure, in a practicable way, the adaptability of courseware.

The storage and management of the learning material is also an issue. The use of distributed database systems and the integration and inter-operation of several services, allowing the access to the best fitting learning material for the user and learning target, have to be achieved to allow a real multiple use and re-use of learning material.

Last, but not the least, the production cost: the courseware production is expensive and requires large teams of persons with different skills and expertise, and it is very time and resource consuming. One way to lower production costs without affecting the material itself, could be the material re-use.

# COURSEWARE RE USE OR COURSEWARE TO REUSE?

**José Carlos Teixeira[♣], César Páris[♦], Joaquim Madeira[♣], J. Brisson Lopes[♠]**
[♣] Dep. Matemática – Fac. Ciências e Tecnologia da Univ. Coimbra - Portugal
[♦] Centro de Computação Gráfica – Coimbra - Portugal
[♠] Instituto Superior Técnico – Lisboa – Portugal
teixeira@ccg.uc.pt; cparis@ccg.uc.pt; jmadeira@mat.uc.pt; brisson@ist.utl.pt

## 1. Introduction

Teaching and training scenarios have changed over the years. This does not come as a surprise, but one must study what are the corresponding implications in current teaching and learning methods and processes.
There are three major differences. First, starter learners are more receptive to learning if they are able to visualize different objects, models and concepts — they are growing up in a TV and computer world —, and by actively manipulating them — (learning is thus a hands-on process, which stimulates the curiosity of learners and their interest in acquiring additional knowledge. Furthermore, there has been a huge advance in technological support (computers and networks), and, finally, learning is no more confined to the first 25 years of life, but has turned into a lifelong process.
Today's learning scenarios extend beyond the traditional classrooms: students can access courseware materials from home and teachers feel the need to use new media in their classes. Both scenarios require multimedia material and add interactivity to the traditional chalk and board scenario.
In this position paper we want to open the discussion about the main issues that must be addressed to enable courseware reuse and the feasibility of possible solutions, based on some experiences we already have

## 2. Production of Courseware Material

Multimedia courseware production is not a one-person job, but requires large teams of persons with different skills and expertise, and is very time and resource consuming. There are two ways of lowering costs: increase the number of people using each course, or make use of the material in different contexts, that is, reuse it.
The best way of guaranteeing that the courseware produced fits its author's idea is that he/she will produce it all. In fact, no deviations will be made concerning the focus and original design, and the author will always be up to date with the current state of production. With some tools to keep track of the courseware produced he/she will easily maintain and validate it. However, one can immediately perceive the drawbacks of this option: time constrains, tools and methods used might not be the best, but always the ones that the author can access. The best option will be then to have separate teams, according to specific functions.

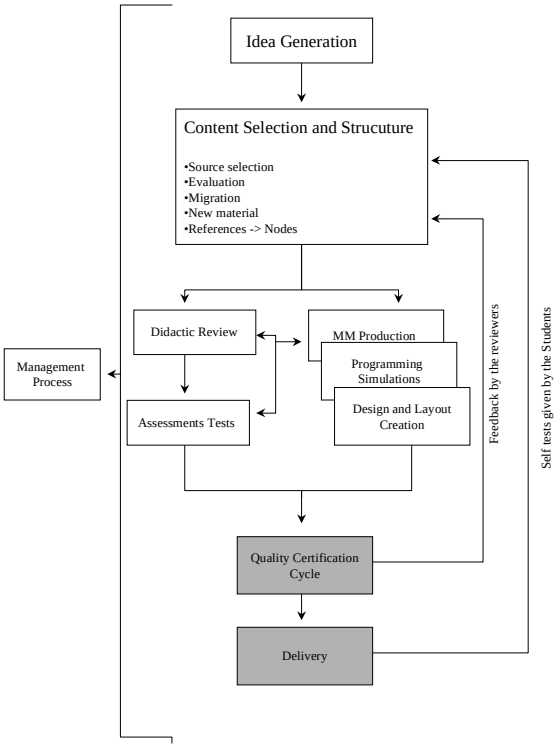



Figure 1. The courseware production process.

As depicted in the above diagram, the same person can perform different functions. For example a teacher, or a group of teachers, can generate one idea, put it on paper and take the responsibility of the didactic and assessment aspects, leaving the most technical aspects - MM Production,

Programming Simulation and Design, as well as Layout Design - to experts in those fields. Of course the authors and the members of the production team will have to interact, and that can be done in real time or in deferred time. This last option could be very helpful if thinking globally, i.e., if a scenario is planed where authors are scattered throughout the world and the production team fixed in one specific place.

The drawback of this scenario is that there are many extra variables that are input into the process. There is now the need for a Management Process ensuring that there is a correct workflow. Furthermore, since the authors are not always directly involved in the production, there must be control mechanisms allowing them to validate the successive production steps. Nevertheless, at the end of the process there should also be a Quality Certification Cycle, which guarantees that the material to be released is still valid.

Although this additional production structure, is associated with an increase in overall costs, it will now be possible to generate high-quality courseware with short-production cycles - two key factors when it comes to courseware production. One way to lower production costs without affecting the structure that produces them, could be the re-use of material already produced. The numbers in the IDEALS project [1], an European project with 13 partners, from industries to universities, from 6 different countries, show that it is possible to lower production costs making re-use of existing courseware.

In that project the conclusions showed that the production time for just one study hour was 164.3 hours [2]. This ratio depends too much on the type of media involved (pictures, video, sound) and the demanded quality. Using templates for some parts of the course the number of hours required to produce one hour of course material was lowered to 91.8 h, with savings of about 44%. That is only the amount of time associated with courseware creation and does not include the time consumed by management tasks.

More important than the above is the possibility to further reduce the production time of course materials by re-using previously developed material. This allows saving about 27 working hours per one hour of courseware. However, it was obvious to all partners involved in the field tests, that re-using implied almost always additional adaptation times which were strongly related to the complexity of the material being re-used. The average time for this was about 6.0 hours per one hour of courseware produced [3]. All of those tasks took more time than expected, mainly because there were no real, effective guidelines.

The numbers presented above just give an idea about the effectiveness that can be achieved by re-using courseware. It is obvious that the results will vary depending on the quality of courseware and the level of re-use that it is planned - just media, or chunks of knowledge.

General-purpose courseware will lower production costs and can be distributed to a larger number of people. On the other hand, knowledge is getting more and more specific today, and courseware must follow this tendency. Besides developing courseware from the scratch, one needs to be able to adapt general-purpose courseware to specific groups.

Note that generic software should be highly modular to ensure its use by large, heterogeneous groups of users, and has to be designed and produced with great pedagogic and didactic care, as well as carefully designed screen layout and user interface. Thus making it easy to use, as well as stimulating, and ensuring its success regarding the learners. The cost associated with ensuring a high-degree of courseware quality is usually offset by the number of learners that can use such generic courseware.

On the other hand, specific courseware can easily incorporate higher-level knowledge units and might be less modular. This reduced degree of modularity and higher specificity usually restrict the number of possible learners, and the usual high design and production costs might not be as easily offset by the actual number of learners.

## 3. Courseware Reuse

Nowadays we see images being reused (e.g., clipart), but this is not enough. There should be a way for teachers to reuse compact units of knowledge (modules), which focus on a specific subject. It will be necessary to build courses bearing one word in mind: modularity. When content is stored apart from sequence and structure, one can start thinking in reusing the structure itself.

Three hierarchical levels that can be defined when discussing reuse: the most atomic level - pictures, movies; modules -, a collection of atomic entities presented at the same time to a learner, and learning sequences - structures referencing the modules and their relations.

The easiest and obvious thing to do is reusing atomic entities: you just need to replace one picture for another or one movie for the other. There is no secret here: if you have a database

with some indexation you can ease your work. In terms of presentation, maybe you need to do some adjustments/conversions, but in fact this is not a critical issue.

When you want to re-use modules the first problem appears: you must present these modules to an audience. So if you want to re-use them, you must have coherence, and for that the use of templates is inevitable. Having different templates built according to guidelines will make this possible. This factor is indeed the most critical one. There must be tools and processes that can easily map atomic objects into modules according to the template elected.

Finally the learning sequences, i.e., a collection of references to modules and to other learning sequences that together build one knowledge unit. It is important that these learning sequences fully describe all knowledge that has to be presented regarding specific subjects, allowing their re-use in different contexts, without any dependencies to other modules or sequences. Once more, this must be stated in the design guidelines and followed by all authors.

If you are thinking about a book, the modules will be the sub-chapters and the learning sequences will define which and how the sub-chapters constitute one chapter. For the whole book, there is a learning sequence that references other learning sequences (chapters).

Courseware developers must pay close attention in developing an appropriate set of guidelines for courseware design. Those must be as rigid as possible, in order to achieve global coherence for the course material to be produced. Changes to the guidelines will generate new course material that might not be compatible with that which has been previously developed. When this situation happens, probably previous courseware must be re-produced, and then you start wondering if re-use is effective or not.

Based on the many experiences of re-using courseware, and despite the level of re-usability that is in question, it is known that authoring guidelines cannot remain fixed, or at least without significant changes, for a long period. The same is commonly accepted as necessary to produce a good and stable Resource Bank of courseware, which will allow effective cost reduction in Courseware Production. So, what is the answer?

The method of developing a courseware reusable library must take into account that the guidelines might change, and then treat all courseware components as entities that need to be stored and classified in an effective way. The same goes for the learning sequences, modules and presentation templates. All of what is stored in the Resource Bank can be modified/replaced individually without affecting others.

We can easily foresee teachers having available several course templates, choosing the ones which are most suitable for their specific courses, and, after that, filling in the empty spaces with previously produced modules. If no suitable modules can be found, then they will produce them and make them available.

The courseware developer must structure its course, looking up what he can reuse from the Resource Bank, and create or modify the missing entities, using the guidelines pre-defined. Then the courseware information entities are accessed and integrated inside the presentation template choose and accordingly with the learning strategy to be followed. Courseware modules are integrated, like a jigsaw puzzle, in order to fulfil courseware authors' needs. It is not expected that they create their own courseware just reusing previously produced modules, but rather that they will reorganize it, adapt some parts and create others.

Each information system that is worthwhile has the same problem: it grows unchecked until it is too hard to manage. An approach to support the previous ideas in a community of producers demands that it must be possible to create and maintain a Knowledge Centre.

It must also be ensured that everybody uses compatible technologies, and it must be guaranteed that there are common rules, procedures and standards. The use of databases is inevitable, due to the expected large amounts of information produced. These databases must reflect the following key issues: cross-platform access, modularity, scalability and, of course, different access permissions. Classification and indexing are other important problems that must be addressed.

Multimedia courseware production requires multidisciplinary production teams, while publication requires clear business models. Copyright ownership has to be transferred, but author rights remain with the creators. This shows the need for adequate business models. Furthermore, courseware delivery releases materials that can be copied over and over again with no quality loss because they are in digital format. This issue can only be solved with appropriate and secure copyright marking tools, as well as commonly accepted copyright policies.

## 4. Conclusion

This position paper raise some thought inspiring main issues that have to be addressed when considering the design and production re-usable courseware modules.
The authors hope that these ideas might contribute to fruitful discussions during the workshop, as well as possible future chances for collaboration between workshop participants.

## References


[1] IDEALS – Integration of DEDICATED for Advanced Training Linked to Small and Medium Enterprises and Institutes of Higher Education, Project Programme EU Telematics Applications Programme, Project nr. ET-1012, 1995, (http://ideals.zgdv.de)

[2] IDEALS, Report on the Restricted Demo Course. IDEALS – project of the EU Telematics Applications Programme, Project nr. ET-1012, 1998, (http://ideals.zgdv.de)

[3] Paaso, J, Computer Based Teaching Technology for Software Engineering Education, Acta Universitatis Ouluensis C123, 1998, Department of Electrical Engineering, Univeristy of Oulu, Finland

# Framework for Deve l opment of Course Material for Teaching Computer Graphics

**V. V. Kamat**
Department of Computer Science & Technology,
Goa University, Goa 403206, India
vvkamat@unigoa.ernet.in

**Abstract**

*In this paper we define the framework for developing course material to teach Computer Graphics courses. The paper will mainly discuss what support material is desired for effective teaching/learning the subject rather than implementation of a teaching tool. It lays emphasis on three important components of the courseware, mainly the design of course content, an experimenter's workbench and an assessment tool. The framework envisages the need for an integrated approach to courseware development for teaching / learning Computer Graphics. This approach would clearly separate the role of courseware developer from that of the delivery person (traditional teacher). The main theme of the paper is to develop learner-centric course content that will depend less on the teacher and more on the learner. Such a courseware would support three independent perspectives. Firstly, the developer's or author's perspective, which would view the course content as data or media objects that could be easily shared, maintained and reused in other courses. Secondly, the learner's perspective, which would accord the learner a platform-independent experimental workbench to learn by hands-on experience. Finally, the administrator's perspective which would collect feedback on each learner's progress and performance.*

**Keywords:** *Web-based education, Instructional design, Computer graphics courseware.*

## 1. Introduction

Computer Graphics as a discipline has matured over the last three decades. Today many universities teach more than one course on Computer Graphics at different levels to students with background in science, engineering and fine arts. Often the objective and the pre-requisites for each of these courses are different although there is fair amount of overlap in its content. It is visualized that good course material for teaching these courses would contain high volume of multimedia content such as visuals, animations, audio, video and simulations besides the text. Therefore to design the course material for computer graphics course has always been a challenge to both the educationist and the technologist. Kayssi et al. have discussed similar issues in a recent publication on the application of web-based tutoring as applied to computer networks courses [1].

There is no doubt that for an effective delivery of the course, a lot of efforts go into the preparation of course material to be used in the class room by the instructor. The additional support material such as student notes, laboratory assignments, reading material and self-tests, further enhances the quality of instructions. For a highly developed networked infrastructure, which is available today, it is unfortunate that each instructor has to develop this material from scratch. It is an accepted fact that good experienced teachers are difficult to come by and this approach will have limited applicability and will not scale-up.

One important aspect of any course on Computer Graphics is experimentation. Typically the instructor gives a series of assignments and students solve these assignments and learn the concept by actual implementation. The common difficulty faced by the student during implementation is that they have to build the application from scratch. This can be very time consuming and frustrating exercise.

It is also noticed that students who like programming otherwise are some times intimidated with matrix notation, vector algebra, 3D geometry and Calculus. Often these subjects are taught in the first few years of college education and by the time student opts for computer graphics course, these concepts are forgotten. As a result, students associate computer graphics to mathematics and develop an unnecessary fear for the subject. This makes it

mandatory for the instructor to review some of these mathematical preliminaries in the class, before proceeding to teach Computer Graphics concepts and algorithms. Further, the approaches taken to teach a fundamental course and an advance course may be different. For example, in a fundamental course on Computer Graphics, more emphasis may be laid on data structures and algorithms and a student may be expected to implement these concepts from ground-zero. On the other hand, for an advance course, the emphasis may be more towards making use of canned code in order to build reasonably complex visualization programs.

## 2. Graphics Courseware Components

In order to create resource material to be shared among a group of instructors and the students, adequate measures should be taken at the design and development stage. It is a matter of fact that the teachers often prefer to improve upon their own course material after each delivery of the course. Therefore, it is important that such a course material is highly modular in content, reusable, easy to maintain, store and distribute. Further it should provide the scope for the instructor to customize it to keep the learner motivated.

For quite some time now, a need has been felt by the Computer Graphics community to define a Graphics Workbench, a tool to carry out experimentation in visualization. Such a workbench would allow the students to experiment with Computer Graphics concepts without having to write a lot of additional code. It would require to have easy to learn easy to use API that would allow anyone to implement a graphics module that can be designed at a higher level, independent of any specific platform and tested on a modest system with requirements commonly available at work and home.

Assessment is an essential part of any training. An assessment tool should not only collect the statistics about usage, evaluation and progress of a particular learner but also provide assistance to the author to prepare questions and answers for an on-line test.

The framework suggests that all three components are essential to give an integrated view of the courseware to the learner.

## 3. Instructional Design Strategy

The development of course content is an extremely labor intensive and time-consuming process. Therefore it is important that proper care is taken at the instructional design stage so that the course content can be easily modified and maintained. One method to do this is to design course material in a modular way with a hierarchical structure of concepts and separation of course structure from individual concepts [2]. Each concept will be an independent entity in itself with well-defined objectives, pre-requisites and evaluation criteria. Course material organized in this fashion can be modified easily without much effort. There can be more than one structure to organize the courseware. This will allow two learners with different learning profiles and taking the same course, to be presented with different learning material. Further, it is observed that if the students do pre-work before arriving at the training session, the learning process then becomes proactive rather than reactive resulting in better comprehension and retention.

Important processes in courseware design would involve the definition of a set of concepts. Each concept would define the following:

- Concept objectives
- Any pre-requisites
- Relation to other concepts
- Questions & answers
- Testing material
- Trainer's guide
- Student's material (pre-work)
- Classroom slides
- Lab Assignments & Solution
- Reading material

The courseware design team would involve a set of subject experts, an instructional designer, a couple of programmers, and graphic artists. It is very important to realize that the development of good instructional material lies much beyond the expertise of subject experts. It definitely involves a team effort and cross-disciplinary approach. The required skills could be best compared to that of web page design if not to making a film. Since several subject experts could be involved, a template has to be prepared in order to provide a uniform structure for developing course content. Such a template can also help in identifying the properties of the content, which can be used later for managing the content. [3]

## 4. Architecture

The suggested architecture for development of courseware is based on client/server model and would use a relational database as a central repository for storing different media objects. The vast popularity of the Internet and the World Wide Web has made the Web browser as the most consistent cross-platform solution for browsing multimedia data. This would also provide a common user interface to developers, learners and administrators, a platform independent client from any location on the Internet/Intranet. The interactivity can be achieved through cross-platform languages such as Java. For example, to communicate the concept of interactive curve design, a Java applet can be written to input the points interactively and drawing a Bézier curve and then displacing one of the points and redrawing the curve. Any amount of hand waving in a classroom cannot communicate this concept effectively but a simple demo will do the trick.

## 5. Graphics Workbench

The experimentation Graphics Workbench (GW) will basically have a visual experimentation environment that can allow a learner to easily set up and test customized sequences of a graphics pipeline. It can not only help the learner to create and edit complex geometrical models but also test ones own module by plugging it along with other modules pipelined in a complex data flow graph called module network. The workbench will perform automatic module scheduling in a module network. This will allow the learner to modify parameter for each module in the network, automatically re-scheduling all modules affected by a parameter value change. It will support rich set of graphic file formats and have large set of graphics modules arranged in a hierarchical library for the fast and task oriented retrieval.

Typically a computer graphics application can be modeled as a set of cooperating processes of three types as shown in Figure 1.

- Input processes, responsible inputting the geometric model into the system
- Computational processes responsible for performing one or more graphic operations on the geometric model
- Presentation processes and display processes, responsible for presenting and storing the output result.

When designing a stand-alone application, one typically would describe the application in a top-down fashion. The description would begin with the overall structure of the application and would end with a detailed description of each process including its inputs, outputs and function. This can help the learner to rapidly prototype an application using the GW.

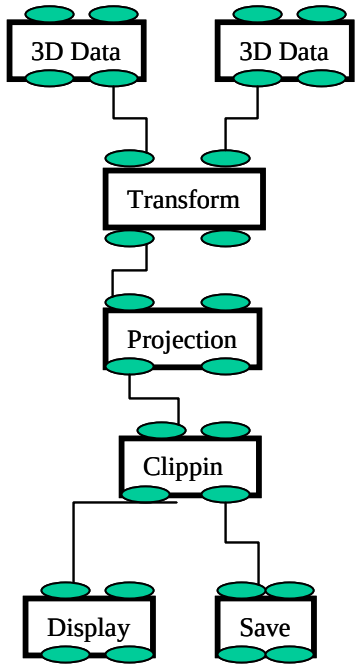



Figure 1. Data flow graph showing a network.

## 6. Evaluation Tool

This tool would help the teacher and the learner to track the progress and performance of an individual. In addition, it would keep a statistics about the difficulty of the question by its usage. It will consist of set of forms and wizards that would allow the author to create variety of objective questions of the type:

- True and false questions
- Multiple choice questions
- Match the columns
- Questions using interactive simulations

Each question and its answer are stored in the database to form a question bank. Each time a student wants to take the test, it would select the question randomly form the question bank, set the number of questions and the time. The wrong answer to any of the question would link back to concepts in the courseware. It may also include

hints, instructions and advise to a particular question using different media types such as text, audio or a video.

## 7. Conclusions

The concept paper outlines the need, objectives and suggests an integrated approach to courseware development for teaching / learning computer graphics. It envisages that such a courseware would be able to support three independent perspectives each for the author, learner and the administrator so that effective learning can be monitored closely.

## References


[1] Kayssi A., El-Haji A., El Assir, M., Sayyid, R., Web-based tutoring and testing in a computer networks course, Computer Applications in Engineering Education Vol. 7 No.1, 1999

[2] Teixeira J. C., Paris C., Brisson Lopes J.M., Paiva, A.M., Authoring Courseware for Distributed Environments Proceedings ICVC'99 held in Goa, India, February 1999

[3] Connolly C. Developing Learner Centric Content, White paper Empower Corporation, 1998 (http://www.empower-co.com)

# List of Participants

- *Anand, Vera* – Clemson University – USA
- *Bailey, Dan* – UMBC – USA
- *Bailey, Michael J.* – SDSC – USA
- *Bresenham, Jack* – Winthrop University – USA
- *Brodlie, Ken* – University of Leeds – UK
- *Brown, Judith R.* – University of Iowa – USA
- *Case, Colleen* – Schoolcraft College – USA
- *Chiappini, Giampolo* – Consiglio Nazionale delle Ricierche – Istituto per la Matematica Applicata – Italy
- *Cunningham, Steve* – California State University Stanislaus – USA
- *Dochev, Danail* – Institut of Mathematics and Informatics, Bulgarian Academy of Science – Bulgary
- *Eber, Dena E.* – Bowling Green State University – USA
- *Ebert, David S.* – Univesity of Maryland – USA
- *Fuchs, Laurent* – IRCOM-SIC, Université de Poitiers – France
- *Gomes, Mário Rui* – IST/INESC – Portugal
- *Hansmann, Werner* – University Hamburg – Germany
- *Hewett, Thomas T.* – Department of Psychology, Drexel University – USA
- *Hitchner, Lewis E.* – California Polytechnic State University – USA
- *Howard, Toby* – University of Manchester – UK
- *Kamat, Venkatesh* – Goa University – India
- *Kjelldall, Lars*– KTH – Sweden
- *Leitão, J. Miguel* – Instituto Superior de Engenharia do Porto – Portugal
- *Longson, Tony* – California State University – USA
- *Lopes, João Brisson* – Instituto Superior Técnico – Portugal
- *Madeira, Joaquim* – Universidade de Coimbra – Portugal
- *McGrath, Michael* – Colorado School of Mines – USA
- *Mitchell, Bonnie* – Bowling Green State University – USA
- *Morie, Jacquelyn F.* – Rhythm and Hue Studios – USA
- *Mumford, Anne* – Loughborough University – UK
- *Ollila, Mark* – University of Gavle – Sweden
- *Owen, G. Scott* – Georgia State University – USA
- *Páris, César* – Centro de Computação Gráfica – Portugal
- *Pavlov, Radoslav* – Institut of Mathematics and Informatics, Bulgarian Academy of Science – Bulgary
- *Prieto, Andres I.* – University of Cantabria – Spain
- *Santos, Beatriz Sousa* – Dept. Electronica e Telec., Universidade de Aveiro – Portugal
- *Santos, Manuel Próspero dos* – Departamento de Informática da Universidade Nova de Lisboa – Portugal
- *Slavik, Pavel* – Czech Technical University – Czech Republic
- *Sowizral, Henry* – Sun Microsystems Inc. – USA
- *Teixeira, José Carlos* – Universidade de Coimbra / Centro de Computação Gráfica – Portugal
- *Tomida, Akemi* – University of Cantabria – Spain
- *Trindade, Jorge F.* – IPG-ESTG – Portugal
- *Wolfe, Rosalee* – DePaul University – USA
- *Wüthrich, Charles* – Bauhaus - University of Weimar – Germany