# Document Understanding: Research Directions

Sargur Srihari, Stephen Lam, Venu Govindaraju,
Rohini Srihari and Jonathan Hull

CEDAR-TR-92-1 May 1992

**Center of Excellence for Document Analysis and Recognition**
**State University of New York at Buffalo**
**226 Bell Hall**
**Buffalo, New York 14260-0001**

# Document Understanding: Research Directions[1]

Sargur Srihari, Stephen Lam, Venu Govindaraju,
Rohini Srihari and Jonathan Hull

Center of Excellence for Document Analysis and Recognition
State University of New York at Buffalo
226 Bell Hall
Buffalo, New York 14260-0001

## Abstract

A document image is a visual representation of a printed page such as a journal article page, a facsimile cover page, a technical document, an office letter, *etc.* Document understanding as a research endeavor consists of studying all processes involved in taking a document through various representations: from a scanned physical document to high-level semantic descriptions of the document. Some of the types of representation that are useful are: editable descriptions, descriptions that enable exact reproductions and high-level semantic descriptions about document content. This report is a definition of five research subdomains within document understanding as pertaining to predominantly printed documents. The topics described are: modular architectures for document understanding; decomposition and structural analysis of documents; model-based OCR; table, diagram and image understanding; and performance evaluation under distortion and noise.

[1] Each of the main sections of this paper were individually prepared as position papers for the DARPA Document Understanding Workshop, Xerox PARC, Palo Alto, CA, May 6-8, 1992.

# Contents


**1 Document Understanding** **1**

**2 Modular Architectures for Document Understanding** **2**

2.1 Introduction . . . . . 2

2.2 Functional Architecture . . . . . 2

2.3 Representation Levels . . . . . 3

2.4 System Architecture . . . . . 4

2.5 Document Descriptions . . . . . 6

2.6 Future Directions . . . . . 6

**3 Decomposition and Structural Analysis** **7**

3.1 Introduction . . . . . 7

3.2 Block segmentation . . . . . 8

3.2.1 Top-down methods . . . . . 8

3.2.2 Bottom-up methods . . . . . 9

3.3 Block Classification . . . . . 9

3.4 Logical Grouping . . . . . 10

3.5 Output representation . . . . . 11

3.6 Research Priorities . . . . . 11

**4 Model-Based OCR** **13**

4.1 Current Limits of OCR Technology . . . . . 13

4.2 Promising Technologies for Improving Robustness . . . . . 13

4.2.1 Recognition without context . . . . . 14

4.2.2 Recognition with context . . . . . 14

4.3 Word Models . . . . . 14

4.3.1 Character-based word recognition . . . . . 15

4.3.2 Segmentation-based word recognition . . . . . 15

4.3.3 Word-shape recognition . . . . . 15
4.3.4 Classifier combination . . . . . 16
4.4 Use of linguistic constraints . . . . . 16
4.4.1 Syntactic methods . . . . . 17
4.4.2 Statistical methods . . . . . 17
4.4.3 Hybrid method . . . . . 17
4.5 Research Priorities . . . . . 18

**5 Table, Diagram and Image Understanding 19**
5.1 Introduction . . . . . 19
5.2 Table Understanding . . . . . 19
5.3 Diagram Understanding . . . . . 20
5.4 Image Understanding . . . . . 22
5.5 Relationship to other Topics . . . . . 23
5.6 Guidelines for Focusing Research . . . . . 23

**6 Performance Evaluation under Distortion and Noise 24**
6.1 Introduction and Background . . . . . 24
6.2 Performance Evaluation for Document Analysis . . . . . 25
6.3 Conclusions and Future Directions . . . . . 28

**Bibliography 29**

# 1 Document Understanding

The goal of a document understanding [DU] system is to encode the contents of documents on paper into an appropriate electronic form [1]. Such an encoding can be one of several forms: an editable description, a concise representation from which the document can be (exactly) reconstructed: a high-level semantic description which can be used to answer queries, *etc.*

Document understanding as a research endeavor consists of studying all processes involved in taking a document through various representations: from a scanned or facsimile multi-page document to high-level semantic descriptions of the document. Some of the types of representation that are useful are: editable descriptions, descriptions that enable exact reproductions and high-level semantic descriptions about document content.

This report is a definition of research subdomains within DU. The field is subdivided into five subdomains, as follows:

1. *Modular architectures* are necessary to partition document understanding research and development into manageable units. Such compartmentalization, however, brings in issues of how to maintain communication and integrate results from each of the subdomains.

2. Documents consist of text (machine-printed and handwritten), line drawings, tables, maps, half-tone pictures, icons, etc. It is necessary to *decompose* a document into its component parts in order to process these individual components. Their *structural analysis*, in terms of spatial relationships and logical ordering, is necessary to invoke modules in appropriate order and to integrate the results of the appropriate modules.

3. *Model-based OCR* refers to recognizing words of text using lexicons and higher level linguistic and statistical context. Its importance arises from the fact that is often impossible to recognize characters and words in isolation without knowing its context.

4. *Understanding tables, diagrams and images* and integrating them with accompanying text is a problem involving spatial reasoning. This is the least well-understood area of DU.

5. *Performance evaluation under distortion and noise* refers to methods for determining data sets on which evaluation is based and methods for reporting performance.

# 2 Modular Architectures for Document Understanding

## 2.1 Introduction

Deriving a useful representation from a scanned document requires the development and integration of many subsystems. The subsystems have to incorporate in themselves the necessary image processing, pattern recognition and natural language processing techniques so as to adequately bridge the gap from paper to electronic media [2].

A DU system should be capable of handling documents with varied layouts, containing text, graphics, line drawings and half-tones. Several special-purpose modules are required to process different types of document components. It is essential to have a global data representation structure to facilitate communication between processes. This also allows independent subsystem development without being concerned with communication protocol.

In discussing DU it is useful to note that significant research is still required for extracting descriptions at the desired level of detail so that exact paper documents can be exactly replicated, *e.g.,* fonts are not typically recognized in today's OCR systems.

## 2.2 Functional Architecture

A functional architecture specifies the major functional components without concerning itself with practical considerations such as shared resources [3]. The functional modules and interactions of a DU system are shown in Figure 1. The DU task is divided into three conceptual levels: *document image analysis*, *document image recognition* and *document understanding*. Within these levels there are several processing modules (or tools): binarization, area-segmentation, area-labeling, OCR, photograph analysis, graphics analysis, picture understanding, natural language processing, and graphics understanding. The interaction between modules allows interpretation of individual sub-areas be combined to form higher level of representation, *e.g.*, the interpretation of photograph caption by natural language processing module and objects on photograph located by photograph analysis module can be used by photograph understanding module to label the objects identity.

This architecture is capable of processing a large variety of documents and allows documents to be processed at different levels of detail. This system is being developed in CEDAR to read different kinds of documents such as newspapers, mail pieces, forms, technical journals and utility bills.

Input to the system is a high-resolution gray-scale image (*e.g.,* 300ppi). Using gray-scale imagery increases image analysis capability, *e.g.*, gray-scale character recognition [4], analysis of half-tone photographs. The output of the system is a description of the contents of document components. An editable description should contain the following entries: (i) component identities and locations on the document, *e.g.,* text, graphics, half-tones, *etc.*, (ii) spatial relationships between components, (iii) layout attributes, *e.g.,* component size, number of lines in text block, *etc.*, and (iv) logical grouping of components.

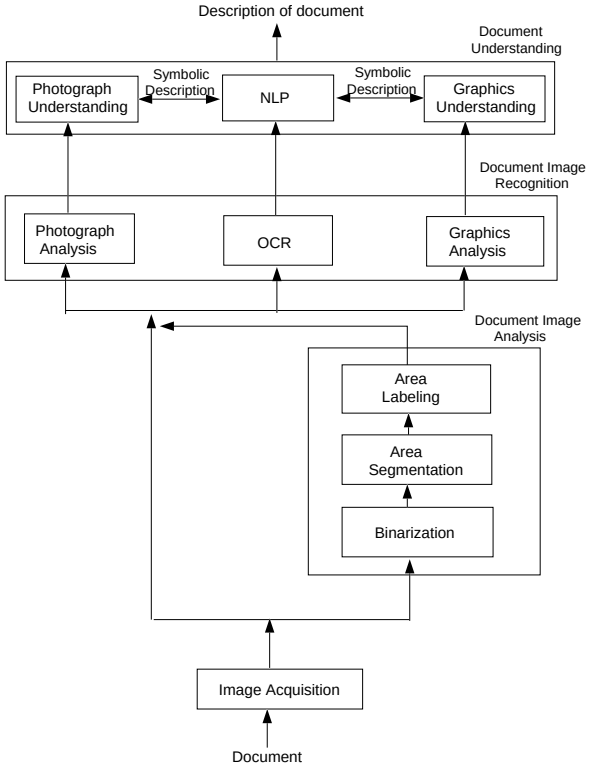


Figure 1: A functional architecture for document understanding.

System capability is determined by the level of representation that the system can derive from document image. Three types of information can be derived from a document: (i) *layout (geometric) structure* is a physical description of the building components of a document, *i.e.,* size, location, spatial relationships between components, (ii) *logical structure* is a grouping of layout components based on *human interpretation* of the content of the components and the spatial constraints between components, and (iii) *content interpretation* contains coded data of a component which can be used to derive logical structure or can be stored for later access. A system whose output contains only the layout structure is a *document layout analysis system* and a system whose output contains all three types of information is a *DU system.* From this perspective, layout analysis is an intermediate step of DU [5].

## 2.3 Representation Levels

In order to develop a system for DU, it is necessary to establish the levels of data representation as that the data passed between tools are well-defined. Five levels of representation, each at an increasing level of abstraction are distinguished:

1. Pixel - the most primitive representation of a document. Pixel information is mainly used in area-segmentation and area-labeling.

2. Connected component - formed by a group of connected black pixels. It is more time-efficient to access a connected component than a group of pixels. It is an appropriate representation for character recognition, line-drawing analysis.

3. Symbol - output of the recognition subsystems, *e.g.*, text, graphic commands, line-drawing descriptions.

4. Frame - representation of a group of document components as formed by layout structure analysis and logical structure analysis.

5. Tree (or graph) - arrangement of logical units on a document.

This hierarchical representation of data has several advantages:

1. It is space-efficient which allows different tools to use the same set of data.

2. Data at a particular level are linked bidirectionally to its immediate neighboring levels (except pixel and tree levels which have only one neighboring level). The links facilitate robust control and problem solving strategies which require selective data access to different levels.

3. This representation supports different control strategies: (i) *top-down* which starts at the tree level, (ii) *bottom-up* which starts at the pixel level, (iii) *opportunistic* which accesses data at different level according to problem state.

4. A tool can use data from more than one level by following data linkage.

## 2.4 System Architecture

Several document processing systems have been proposed during the last decade to handle special classes of documents [6]. In many cases, however, the system design was largely guided by the application domain and the required output.

We describe here a DU system architecture that is *not* confined or guided by the types of documents being considered [7]. The system should only contain a robust control (*i.e., domain independent*), a general knowledge representation scheme to describe documents of interest and well-defined levels of data representation to store intermediate document data. Tools should be developed independently in accordance with specifications and their integration to overall system requires no significant modifications.

Figure 2 shows the organization of the DU system being developed in CEDAR. The architecture allows for parallel development of different subsystems. System integration, testing and technology transfer become feasible since (i) the functionality of each of the system components is clearly specified, and (ii) the interactions between components are facilitated by global data representation and central working memory. It consists of three major components:

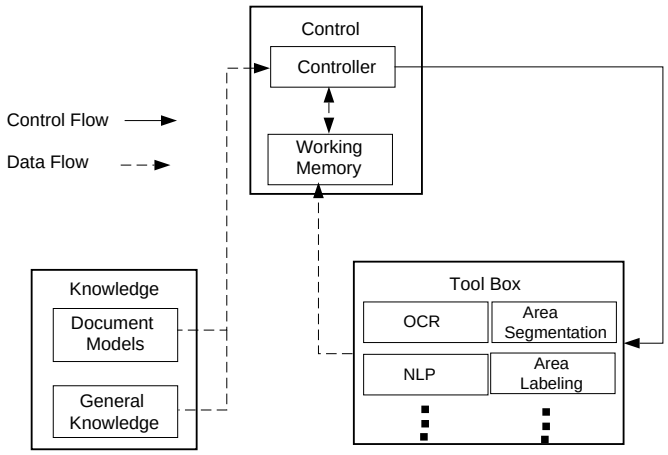


Figure 2: A system architecture for document understanding.

1. *Tool box* contains all the modules needed for document processing. The tools correspond, for example, to each of the functional components shown in Figure 1 . Tools developed for different conceptual levels are coordinated by the control.

2. The *knowledge base* consists of two sub-components: *document models* and *general knowledge*. A document model describes the aspects of a document domain or a group of documents that share similar layout structure. The expressive power of the model representation dictates the capability of a DU system to handle different kinds of documents. General knowledge on the other hand is shared by different document domains. It describes the tasks that are needed to locate and identify document components, such as text blocks and line segments. A task is carried out by one of the modules in the tool box. The general knowledge can apply to objects of different domains since they share similar structural information. Lexicons used by different tools such as for OCR and NLP are stored in document models.

3. *Control* is the most critical issue in DU system design. Its functions includes: (i) selective use of tools, (ii) intelligent combination of data extracted from document sub-areas to generate a representation of the scanned document. The *controller* examines the problem state in the *working memory* and uses the knowledge in the knowledge base to determine which modules in the tool box should be used. Working memory is a temporary storage where different levels of data will be stored during document processing and will be updated after each module activation. The search process stops when all the objects specified in the document model have been located.

Tool interaction is determined by the knowledge. The general knowledge defines the dependency

or the activation order of tools, *e.g.,* area-labeling can only be activated after area-segmentation. A document model defines the tool interactions needed in different document sub-areas since each sub-area may require different level of interpretation, *e.g.,* recognizing recipient (name and address) on a business letter requires both OCR and NLP while reading the title of a technical document only needs OCR.

## 2.5 Document Descriptions

For a given document the output of a DU system is an representation of its contents. In particular, editable descriptions are of interest. Recent work has emphasized the importance of maintaining the structure of a document at all stages of editing, storage and transmission. The Office Document Architecture (ODA) and the Standard Generalized Markup Language (SGML) are two international standards for the representation and interchange of structured documents [8]. ODA has close links with the office automation and communications world. This reflected particularly strongly in its content architectures. Therefore, ODA is commonly used as the standard output format of a document understanding system. SGML is more closely linked publishing and printing communities where great flexibility of document design and layout are of prime importance. ODA provides most of the basic framework needed as the basis of an interactive editing system as well as for document storage and transfer.

Other work at CEDAR has focused on deriving a semantic representation (typically a semantic network) of the contents of photo-caption pairs [9]. This involves an integration of natural language understanding and image understanding.

## 2.6 Future Directions

Each of the document processing modules described above still needs further development. The control needs to be more sophisticated as the number of modules and the interactions between modules increased.

A user-friendly system interface is necessary to allow non-technical uses to define new document models. A new document description format is needed to store details such that exact paper document can be reproduced from electronic storage.

# 3 Decomposition and Structural Analysis

## 3.1 Introduction

A *document* image is a visual representation of a printed page such as a journal article page, a facsimile cover page, a technical document, an office letter, *etc.* Typically, it consists of blocks of text, *i.e.*, letters, words, and sentences that are interspersed with tables, and figures. The figures can be symbolic icons, gray-level images, line drawings, or maps. A digital document image is a two-dimensional array representation of a document image obtained by optically scanning and raster digitizing a hardcopy document. It may also be an electronic version that was created for publishing or drawing applications available for computers.

Methods of deriving the blocks can take advantage of the fact that the structural elements of a document are generally laid down in rectangular blocks aligned parallel to the horizontal and vertical axes of the page. The methods can also use several types of knowledge including *visual, spatial,* and *linguistic.* Visual knowledge is needed to determining objects from the background. Labeling blocks involves the use of spatial knowledge, *e.g. layout of a typical document.* Determining the font and identity of characters also involves spatial knowledge. Reading words in degraded text is a process that involves spatial as well as linguistic knowledge, *e.g.*, lexicon of acceptable words. Determining the role of a block of text, *e.g.*, "is this a title?", is a process requiring spatial, syntactic as well as semantic knowledge. Considerable interaction among different types of knowledge is necessary. For instance, assigning a role to a textual region may require not only knowledge of the spatial layout, but also an analysis of its textual syntax and semantics, and an interpretation of neighboring pictorial regions.

The document decomposition and structural analysis task can be divided into three phases [1]. Phase 1 consists of *block segmentation* where the document is decomposed into several rectangular blocks. Each block is a homogeneous entity containing one of the following: text of a uniform font, a picture, a diagram, or a table. The result of phase 1 is a set of blocks with the relevant properties. A textual block is associated with its font type, style and size; a table might be associated with the number of columns and rows, *etc.* Phase 2 consists of *block classification.* The result of phase 2 is an assignment of labels (title, regular text, picture, table, *etc.*) to all the blocks using properties of individual blocks from phase 1, as well as spatial layout rules. Phase 3 consists of *logical grouping* and ordering of blocks. For OCR it is is necessary to order text blocks. Also the document blocks are grouped into items that "mean" something to the human reader (author, abstract, date, *etc.*), and is more than just the physical decomposition of the document. The output of phase 3 is a hierarchical tree-of-frames, where the structure is defined by the shape of the tree and the content is stored entirely in the leaves. The tree-of-frames can be converted to the standard SGML (Standard Generalized Markup Language) representation to ensure portability, easy information retrieval, editability and efficient semantic indexing of the document.

## 3.2 Block segmentation

Approaches for segmenting document image components can be either top-down or bottom-up. Top-down techniques divide the document into major regions which are further divided into sub-regions based upon knowledge of the layout structure of the document. Bottom-up methods progressively refine the data by layered grouping operations.

### 3.2.1 Top-down methods

Four different approaches to document segmentation have been experimented with at CEDAR.

*Smearing* is based on the Run-Length Smoothing Algorithm (RLSA) [10]. It merges any two black pixels which are less than a threshold apart, into a continuous stream of black pixels. The method is first applied row-by-row and then column-by-column, yielding two distinct bit maps. The two results are then combined by applying a logical AND to each pixel location. The output has a smear wherever printed material (text, pictures) appears on the original image.

The *X-Y cut* method assumes that a document can be represented in the form of nested rectangular blocks [11]. A "local" peak detector is applied to the horizontal and vertical "projection profiles" to detect local peaks (corresponding to thick black or white gaps) at which the cuts are placed; it is local in that the width is determined by the nesting level of recursion, *e.g.*, gaps between paragraphs are thicker than those between lines.

The *Hough transform* approach exploits the fact that documents have significant linearity. There exist straight lines in tables and diagrams. Centroids of connected components corresponding to text also line up. Columns of text are separated by straight rivers of white space. Text itself can be viewed as thick textured lines. The Hough transform is a technique for detecting parametrically representable forms like straight lines in noisy binary images. In fact, we have shown that the Hough transform is truly a representation of the projection profiles of the document in every possible orientation [12]. The accumulator array can be checked for the particular orientation which has the maximum number of transitions to and from a minimum value. Transitions corresponding to text are usually regular and uniform in width and thus are easy to identify. Maximum values are registered at the center line of the characters and slightly lower values corresponding to the ascender and descender lines. The analysis of the accumulator array has an added advantage. It can provide the angle of skew in the document. Since the segmentation phase assumes that blocks align parallel to the page margins skew detection and correction is an essential preprocessing step.

*Activity identification* is a method developed at CEDAR for the task of locating destination address blocks on mailpieces. The goal is to use a simple technique for documents with simple layout structure (few blocks). It quickly closes into the area of maximum activity by identifying regions with high density of connected components. The method is useful for goal-oriented tasks like finding the destination address block on a facsimile cover page which is predominantly empty.

### 3.2.2 Bottom-up methods

The image is first processed to determine the individual connected components. At the lowest level of analysis, there would be individual characters and large figures. In the case of text, the characters are merged into words, words are merged into lines, lines into paragraphs *etc.*

The application of different operators for bottom-up grouping can be coordinated by a rule based system. Most characters can be quickly identified by their component size. However, if characters touch a line, as is often the case in tables and annotated line-drawings, the characters have to be segmented from the lines. One technique for segmenting characters from line structures is to determine the high neighborhood line density (NLD) areas in the line structures. Following is an example of the type of rules that can be used.

| | |
|---|---|
| *if* | connected components size is larger than a threshold |
| *then* | the connected component is a figure (with likelihood $L_i$) |
| *if* | neighborhood line density (NLD) of a figure is high |
| *then* | the high NLD area is a character area (with likelihood $L_j$) |

## 3.3 Block Classification

Blocks determined by the segmentation process need to be classified into one of a small set of predetermined document categories. Knowledge of the layout structure of a document can aid the classification process. For instance, if it is known *a priori* that the document at hand is a facsimile cover page, then inferences like the central block must be labeled as the destination address and the top of the document must be labeled as the name of the organization *etc.* are plausible. However, to ensure portability, document specific formatting rules should be avoided.

A statistical classification approach using textural features and feature space decision techniques has been developed at CEDAR [13] to classify blocks in newspaper images. Two matrices, whose elements are frequency counts of black-white pair run lengths and black-white-black combination run lengths, are used to derive the texture information. Three features are extracted from the matrices to determine a feature space in which labeling of blocks as picture, regular text, headlines (major and minor), and annotations (*e.g.*, captions to photographs) is accomplished using linear discriminant functions.

A method of associating labels like line-of-characters, paragraph, title, author, text columns, picture, table, line-drawing, *etc.*, to the blocks can be done on the basis of their spatial structure and relative and absolute positions in the document. Following are rules that can be implemented in a knowledge-based system:

| | |
|---|---|
| *line-of-characters:* | sequence of adjacent character blocks of pre-determined height |
| *paragraph:* | sequence of blocks of line-of-characters of same length |
| *column:* | large blocks with roughly equal frequency of black and white pixels |

**Discriminating between handwriting and machine print:** Separating machine printed text from handwritten annotations is necessary for invoking appropriate recognition algorithms. One method that performs this discrimination well with postal addresses is based on computing the histogram of heights of the connected components. Handwritten components tend to have a wider distribution in heights than print.

**Figure classification:** A figure block can belong to one of the following categories: half-tone picture, line-drawing (diagrams), and table. Pictures are gray-scale images and can be separated from tables and diagrams (binary images) by a histogram analysis of the gray-level distributions. Although, both diagrams and tables predominantly comprise of straight lines and interspersed text, the fact that the straight lines in tables run only vertically and horizontally can be used to advantage. Analysis of the Hough transform accumulator array is used to separate tables from diagrams.

**Goal-oriented document segmentation and classification:** In certain document segmentation tasks it is only necessary to extract a given block(s) of interest. An example of this is locating an address block on a mail piece [14]. Several top-down and bottom-up tools are used to segment candidate blocks. They are resolved by a control structure to determine the destination address block by considering spatial relationships between different types of block segments.

## 3.4 Logical Grouping

It is necessary to provide a logical ordering/grouping of blocks to process them for recognition and understanding. Textual blocks corresponding to different columns have to be ordered for performing OCR.

Journal pages can have complicated layout structures. They are usually multi-columned and can have several *sidebars*. Sidebars are explicit boxes enclosing text and figures used for topics that are not part of the mainstream of text. They can either span all or some of the columns of text. Readers wanting a quick overview usually read the main text and avoid the sidebars. The block classification phase labels both the mainstream of text and the sidebars as textual blocks. It is up to the logical grouping phase to order the blocks of the main text body into a continuous stream by ignoring the sidebars.

Another grouping task pertinent to this phase is matching titles and "highlight" boxes to the corresponding text blocks in the mainstream. When a title pertains to a single-columned block, associating the title with the corresponding text is straightforward. However, titles can span several columns, and sometimes be located at the center of the page without aligning with any of the columns, making the task of logical grouping challenging.

One method of logical grouping using rules of the layout structure of the document is being developed at CEDAR for newspaper images [3]. Following are some examples of rules derived from the literature on "Newspaper design" used in the spatial knowledge module of this approach.

*R21:* Headlines occupying more than one printed line are left-justified

| | |
|---|---|
| *R32:* | Captions are always below photographs, unless two or more photographs have a common caption |
| *R43:* | Explicit box(es) around block(s) signifies an independent unit |

Blocks with different labels (photograph and text) which are not necessarily adjacent might have to be grouped together. For instance, a photograph and its accompanying caption together form a logical unit and must be linked together in the output representation.

### 3.5 Output representation

The layout structure of a document divides and subdivides the document into physical rectangular units, whereas the logical structure divides and subdivides the document into units that "mean" something to the reader. The output representation of the document decomposition module is a tree-of-frames where each node in the tree is a logical block represented by a *frame* and the root of the tree corresponds to the entire document. The shape of the tree reflects the structure of the document and the contents of the document is stored entirely at the leaves of the tree. A frame is an $< attribute, value >$ representation that consists of slots where each attribute can store a value. For instance, a block of text can have the attributes of font type, size, style and a flag to indicate if it is handwritten or machine printed. The OCR will benefit by knowing the values of these attributes ahead of recognition.

SGML provides the syntax (in terms of regular expressions) for describing the logical structure of documents. The tree-of-frames representation can be transformed into the SGML syntax without any loss of information. SGML parsers are available that check the validity of the syntax of the logical structure and evaluate the correctness of the decomposition of the document. SGML also provides with *link* mechanisms that can reproduce the entire document from the syntax rules. Such a bidirectional mapping between a document and its representation provides means for semantic indexing and efficient information retrieval.

### 3.6 Research Priorities

The logical grouping phase has been relatively unexplored by the research community. Its investigation should be a high priority.

There are few document layout rules that are universal to all types of documents. Development of a completely general purpose system compromises accuracy for generality. The trade-offs in terms of portability, robustness, and accuracy need to be further investigated.

A question is whether a two-step approach of block segmentation followed by classification is needed. A purely top-down method (full-analysis scheme) of classification applies various classifying filters in series to the entire document. Each time, regions of the document are marked with labels appropriate to the corresponding filter. For instance, if the text classifying filter is used, all the text in the document is labeled as text and the non-textual regions are left untouched. The full-analysis scheme is inherently sequential and seems inappropriate for handling documents with complex

layout structure. However, the simplicity of the control structure indicates that the method must be explored for applicability on "simple" documents.

The various algorithms relating to textual properties will change complexion for documents in languages other than English. For instance, the logical ordering of blocks will be different in scripts like Arabic (written right to left) and Chinese (written top to bottom). It is to be tested if the techniques of discrimination between machine print and handwriting will succeed with scripts of other languages like Devanagari.

# 4 Model-Based OCR

## 4.1 Current Limits of OCR Technology

Character Recognition, also known as Optical Character Recognition or OCR, is concerned with the automatic conversion of scanned and digitized images of characters in running text into their corresponding symbolic forms. The ability of humans to read poor quality machine print, text with unusual fonts and handwriting is far from matched by today's machines.

As an illustration, the performance of three commercially available OCR software packages for the Macintosh computer (Calera Wordscan Plus 1.0, Caere Omnipage Professional 3.0, and Xerox Imaging Systems AccuText 3.0) were compared on an ordinary office document image. The original document consisted of a typed photocopied facsimile page containing 1,928 characters, 314 words and 15 sentences. Most of the characters were printed in a Helvetica font. The document was scanned at 300 ppi binary on a HP Scanjet Plus scanner. Individual OCR performances on character, word, and sentence levels (top choice) are shown below:

| | % Correct | | |
|---|---|---|---|
| Commercial OCR software | Character | Word | Sentence |
| Calera WordScan Plus 1.0 | 98 | 79 | 13 |
| Caere Omnipage Professional 3.0 | 92 | 56 | 0 |
| Xerox Imaging Systems AccuText 3.0 | 98 | 88 | 33 |

Most errors were caused by one of three image quality deficiencies: underlines in text, "I" and "i" confusions, and distortion of "e"'s. Underlines occurred in 27 words and descenders in these words were often completely obscured. This led to recognition failures in all three packages. The dots on "i"s were often smeared together with the body of the "i" thus causing the case confusions. For "e", the small opening of the "e" was often closed due to the low quality of the images. In sum, recognition deficiencies seem primarily due to imprecise character definitions. The document is quite easily read by a human unfamiliar with the domain.

The task of pushing machine reading technology to reach human capability calls for developing and integrating techniques from several subareas of artificial intelligence research. Since machine reading is a continuum from the visual domain to the natural language understanding domain, the subareas of research include: computer vision, pattern recognition, natural language processing and knowledge-based methods.

## 4.2 Promising Technologies for Improving Robustness

Technologies for improving performance of OCR include improvements both at the front end of processing (image scanning, skew detection and correction, underline removal, segmentation into lines, words and characters and image feature extraction) as well as at the back end (recognition

methodology and use of linguistic and other knowledge). Methods for character recognition can be divided into: recognition without context and recognition with context.

### 4.2.1 Recognition without context

The task is to associate with the image of a segmented, or isolated, character its symbolic identity. It should be noted, however, that segmentation of a field of characters into individual characters may well depend on preliminary recognition.

Although there exist a large number of recognition techniques [15], the creation of new fonts, the occasional presence of decorative or unusual fonts, and degradations caused by faxed and multiple generation of copies, continues to make isolated character recognition a topic of importance.

Recognition techniques involve feature extraction and classification. The extraction of appropriate features is the most important subarea. Character features that have shown great promise are strategically selected pixel pairs, features from histograms, features from gradient and structural maps and morphological features. Features derived from gray scale imagery is a relatively new area of OCR research. Gray scale imagery is not traditionally used in OCR due to the large increase in the amount of data to be used; a restriction that can be removed with current computer technology.

Classification techniques that have found promise at CEDAR are a polynomial classifier, neural networks based on backpropagation, and Bayes classifiers that assume feature independence and binary features. Orthogonal recognition techniques can be combined in order to achieve robustness over a wide range of fonts and degradations.

### 4.2.2 Recognition with context

The problem of character recognition is a special case of the general problem of reading. While characters occasionally appear in isolation, they predominantly occur as parts of words, phrases and sentences. Even though a a poorly formed or degraded character in isolation may be unrecognizable, the context in which the character appears can make the recognition problem simple. The utilization of model knowledge about the domain of discourse as well as constraints imposed by the surrounding orthography is the main challenge in developing robust methods.

Several approaches to utilize models at the word level and at a higher linguistic level are known and utilized in our laboratory.

## 4.3 Word Models

Word models can be represented in different ways. An obvious method is to store the list of legal words, or lexicon. Other methods include n-grams (legal letter combinations), Markov models — which capture first and higher order letter transitional probabilities — and n-gram probabilities. Three distinct approaches to utilizing word models in character recognition can be identified: character-based word recognition, segmentation-based word recognition and word shape recognition.

### 4.3.1 Character-based word recognition

This is a three-step approach. First the word image is segmented into character images. Second, the segmented characters are recognized by using an isolated character recognition technique. Third, the resulting word is corrected by a model of the expected words. The third step is referred to as contextual postprocessing (CPP). For example, a character recognition technique may not be able to reliably distinguish between a u and a v in the second position of q*ote. A CPP technique determines that u is correct since it is very unlikely that quote would be in an English language dictionary. CPP techniques differ in the method of representing word models. Those that utilize lexicons use string matching techniques to measure a distance between the recognized word and each word in the lexicon. One string matching method uses the weighted edit distance, which measures the minimum number of editing operations such as substitution, insertion and deletion, that are necessary to transform one word into the other [16].

### 4.3.2 Segmentation-based word recognition

This is a two-step approach. Contextual information is used in the process of recognizing individual character images. As with the first approach, in the first step, the word image is segmented into individual characters. In the second, recognition, step, features are extracted for each character image. Classification into a word is done by examining the entire (compound) feature set.

Word level knowledge is brought into play during the recognition step. Letter transitional probabilities and class-conditional probabilities of character feature vectors can be simultaneously utilized during recognition by employing the so-called hidden Markov model of classification. Lexical information can be brought to bear on the recognition phase by using an algorithm known as the Dictionary-Viterbi algorithm [17].

### 4.3.3 Word-shape recognition

This is a one-step approach that bypasses segmenting a word image into characters [18]. Features are extracted from the unsegmented word image. The recognition problem is to associate the given word image with one or more words in a lexicon. In terms of implementing the recognizer, words in the lexicon are represented by the expected word features. Such word features may be based on an expected set of fonts.

Word shape analysis can be implemented as a two-step process involving hypothesis generation and testing. In the hypothesis generation phase, a simple set of features are used to generate a small "neighborhood" of words that are similar in shape to the input word. In the second, hypothesis testing phase, those features that discriminate only between such words are computed for fine discrimination.

Word shape recognition can be regarded to be a holistic approach in that it treats the word as a whole. By contrast, the other two word recognition approaches are analytic.

#### 4.3.4 Classifier combination

Each of the three approaches to word recognition have strengths and weaknesses.

The character recognition based method is appropriate when a reliable segmentation can be obtained and the extracted characters are not deformed by size normalization and other procedures that prepare a character for recognition. Certain, but not all, ambiguities in the decisions on character classes can be resolved by CPP. After the recognition stage, the shape information has all been converted into class labels. The class decision suggests only that the character is in a neighborhood of a standard shape of the decided class. Subtle information about how much the character deviates from the standard shape is forever lost. In some cases, this loss could be so severe that the true word cannot be recovered.

The segmentation based word recognition method is suitable for word images when the characters can be reliably segmented but are difficult to recognize in isolation. Examples are images that are so broken that most of the shape features are lost.

The advantage of word shape recognition is that some errors in character segmentation and premature decisions on character identities are avoided. It is especially suitable for images that are difficult to segment into characters, or where the characters are distorted when they are extracted and normalized.

Each of the three approaches result in word recognition performance that is uncorrelated. Thus improved overall performance can be obtained by combining their results [19]. The method of classifier combination is important. One of these is based on the Borda count. The Borda count for a class is the sum of the number of classes ranked below it by each classifier. In the logistic regression method of combination, weights are associated with classifiers to take into account the performance statistics of each individual classifier and correlations of classifiers. In one experiment at our laboratory, 1,671 binary images were used to test recognition algorithms based on each of the three word recognition methodologies. The same lexicon of 33,850 words was used.

| | % Correct top n choices | | | | |
|---|---|---|---|---|---|
| Classifier/Combination | 1 | 2 | 10 | 100 | 500 |
| 1. Char recog with heuristic CPP | 79 | 86 | 91 | 94 | 95 |
| 2. Segmentation-based method | 75 | 84 | 90 | 95 | 96 |
| 3. Word shape method | 59 | 70 | 75 | 90 | 93 |
| 4. Combination by Borda count | 83 | 88 | 95 | 97 | 99 |
| 5. Combination by Logistic Regrsn | 88 | 91 | 95 | 98 | 99 |

### 4.4 Use of linguistic constraints

The next higher level of model knowledge useful in OCR in linguistic syntax. An example of its use follows. A digitized image of a handwritten sentence "He will call you when he is back" was to be recognized. The top two choices of the word recognizer for each of the words in the input was as follows:

he with call pen when he us back

she will will you were be is bank

Although the correct words can be found in the top two choices, it requires the use of contextual constraints in order to override the (sometimes erroneous) top choice. Word recognizers often tend to misread short (less than or equal to 3 characters) words more frequently than longer words. Furthermore, short words tend to be pronouns, prepositions and determiners causing frequent word confusion between very different syntactic categories (*e.g.*, as, an).

In such cases linguistic constraints may be used to select the best sentence candidate or at least to reduce the number of possibilities. Methods can be purely syntactic, purely statistical or hybrid.

### 4.4.1 Syntactic methods

Syntactic methods employ a grammar capturing the syntax of all possible input sentences and reject those sentence possibilities which are not accepted by the grammar. The problems with such a method are (i) the inability of the method to cover all possibilities (especially since informal language on handwritten letters is occasionally ungrammatical), and (ii) the computational complexity of parsing.

### 4.4.2 Statistical methods

Statistical methods employ n-gram statistics reflecting the transition frequencies between the syntactic categories represented by n-tuples of words. A statistical method can be implemented using the same hidden Markov methodology as at the word level. Here the transitional probabilities between syntactic categories and the word-class conditional probabilities associated with the word images would be used [20].

The problem with this approach is that local maxima often cause the correct sentence choice to be overlooked. As an example, subject-verb agreement becomes a problem due to intervening prepositional phrases as in "The folder with all my ideas is missing." Due to the improbable transition frequency between a plural noun (ideas) and the singular form of the verb "to be" (is), this sentence would receive a lower score than the sentence where the word "is" (the second choice of the word recognizer)" is replaced by the preposition "as" (the top choice of the word recognizer). Thus the incorrect sentence would be ranked higher.

### 4.4.3 Hybrid method

An attempt to overcome the problems associated with each of these approaches is being explored in our laboratory. The method tags the words in each candidate sentence based on its part of speech. The candidates are grouped into classes based on identical tag sequences. A two-stage approach is subsequently employed in order to rank the candidate sentences in each class [21].

### 4.5 Research Priorities

Each of the different subareas outlined above have significant open problems. The methods should scale to other alphabetic languages without significant conceptual changes. Such changes will be required when we attempt to move towards syllabic and other writing systems.

# 5 Table, Diagram and Image Understanding

## 5.1 Introduction

Tables, diagrams and images are often integral components of documents. Each of these document component types share the characteristic of being diagrammatic representations. Their interpretation is related to the human brain's capacity for analogical (or spatial) reasoning. They are used to explicitly represent information (and thus permit direct retrieval) of information which can only be expressed implicitly using other representations. Furthermore, there may be a considerable cost in converting from the implicit to the explicit representation.

The interpretation of tables, diagrams and images also share the characteristic that it is usually necessary to integrate visual information (*e.g.*, lines separating columns, arrows, photographs) with information obtained from text (*e.g.*, labels, captions). The central issue here is defining a static representation of meaning associated with each of these document component types. If such a meaning can be captured, it can be incorporated into the data structure representing the overall understanding of the document. This would allow for interactive user-queries at a later time. Hence, document understanding deals with the definition of meaning and how to go about extracting such a meaning.

For each of the three component types, tables, diagrams, and images, we address the following issues: (i) meaning, (ii) complexity of visual processing, (iii) knowledge required, and (iv) system architecture (data and control structures). Tables are usually stand-alone units that have little interaction with accompanying text (except for table captions), diagrams need a higher level of interaction, and photographs usually need a textual explanation. We consider the integration with accompanying text only in our discussion on image understanding.

## 5.2 Table Understanding

Tables are used in documents as an effective means of communicating property-value information pertaining to several data items (keys). The spatial layout of these items communicates the desired associations.

**Meaning:** In the case of table understanding, it is fairly easy to define meaning. The physical layout structure indicates the logical relationship between textual items. Once the logical relationships have been determined, the information may be effectively represented in a relational database. Table-understanding can be considered as a special case of form-understanding since the techniques used for layout analysis are similar.

**Complexity of Visual Processing:** Visual processing consists of two distinct tasks: (i) the extraction of horizontal and vertical lines, and (ii) reading text (OCR). The first task is accomplished

through existing techniques (*e.g.* Hough transform) for detecting straight lines. The second task however, is considerably more difficult due to factors such as text degradation and lines touching the text. To remedy such problems, contextual knowledge regarding the nature of entries in certain rows or columns may be used.

**Knowledge required:** Knowledge regarding the spatial layout of tables may be employed. For example, the entries in a row represent values of various properties for a single key. On the other hand, entries in columns represent different values of the same property for several keys. More complex knowledge is required to process multi-column headings, *etc.*

Knowledge regarding the properties of certain words and characters is also required. For example, if the character % is seen in a column heading, the entries in that column consist of digits and decimal points. Furthermore, the values must be between 0 and 100. If the column label represents a name, then it would not be reasonable to find digits in that column.

**System Architecture:** From the discussion above, it is apparent that a modular, hierarchical control structure is required. The various modules include (i) general context recognition (*e.g.*, scientific tables, inventory tables, *etc.*), (ii) the layout recognition module, (iii) recognition of item type (*e.g.*, the item consists of a telephone number, separated into three parts) and (iv) character recognition module which reads the individual characters in an item. Finally, bidirectional control between the various modules can be successfully employed by utilizing contextual information in a top-down manner and back propagating obvious errors or contradictions detected by the character recognition module.

## 5.3 Diagram Understanding

Diagrams, like tables, are used to convey information which is understood by humans more readily when presented visually. This category is very diverse and includes domains such as maps, engineering drawings, flowcharts, *etc.* The purpose of diagram understanding can be either one or both of the following: (i) to transform the paper representation into a more useful target representation (such as commands to a graphics program, new entries in an integrated text-graphic database, *etc.*), (ii) to derive a more compact representation of the data for the purposes of archival.

**Meaning:** A simplistic view of diagram understanding is the conversion of a raster representation to a vector representation: *i.e.*, to convert a binary pixel representation of line-work into a connected set of segments and nodes. Segments are typically primitives such as straight lines, parametric curves, domain-specific graphical icons and text. In addition, (i) portions of the drawing containing text must be converted to ASCII and (ii) graphical icons must be recognized and converted to their symbolic representation. Line segments have parameters such as start positions, extent, orientation, line width, pattern *etc.* associated with them. Similar features are associated with parametric curves. The connections between segments represent logical (typically, spatial) relationships.

A deeper level of understanding can be attained if groups of primitives (lines, curves, text, icons) are combined to produce an integrated meaning. For example, in a map, if a dotted line

appears between two words (representing city names), and the legend block associates a dotted line with a two-lane highway, it means that a two-lane highway exists between the two cities. It is possible to define meanings for documents such as maps, weather maps, engineering drawings, flowcharts, *etc.* The definition of meaning is somewhat ambiguous in diagrams such as those found in a physics textbook.

**Knowledge Required:** It is necessary to have apriori knowledge of the types and meanings of primitives for a given context. In the case of line drawings, it is necessary to represent the various types of lines and curves that may appear. This could include higher-level knowledge such as angles (formed by the intersection of two lines), directed arrows *etc.* In the domain of geographic maps for example, the symbol resembling a ladder represents a railroad. It is also necessary to have a lexicon of typical textual primitives along with their meaning. This aids both the text recognition process as well as the later stages of understanding.

In addition to the knowledge used for diagram analysis, domain-specific knowledge regarding the interpretation of higher-level units must also be represented. Finally, information contained in an accompanying caption block may also be used to determine the type of diagram, or additionally, to be used in the process of understanding. This area is addressed extensively in the discussion of image understanding.

**Complexity of Visual Processing:** Visual processing includes the following:

1. Separating text from the image: this is a non-trivial process since text may be touching lines. In such cases, a technique is to determine the high neighbourhood line density (NLD) areas in the line structures. Based on the size of connected components and values of NLD, it is possible to separate the characters from the line.

2. Vectorization: There have been several approaches to vectorization including: pixel-based thinning, run-length based vectorization, and contour-based axis computation. Parametric curves can be approximated either through curve-fitting algorithms or piecewise linear approximation.

3. Graphical Icons: are best detected through template-matching procedures.

4. Some other problems that need to be specifically examined are handling of dotted lines, determining end points of lines, effectiveness of thinning, and determination of vector intersections.

**System Architecture:** Here, as in the case of table understanding, a hierarchical, modular system architecture is employed. The lower level modules perform the tasks of contour and text extraction. At the next level, intermediate-level primitives such as polygons, circles and angles are determined. Finally, high-level, domain-specific knowledge is employed to derive a conceptual understanding of the diagram. This information is then converted to the target representation, *e.g.*, commands in a graphics language, icon database, a CAD/CAM file. Once again, bi-directional flow of control is employed, whereby domain-specific knowledge is used in a top-down manner, and inconsistencies propagated in a bottom-up manner.

## 5.4 Image Understanding

The problem of performing general-purpose vision without apriori knowledge is nearly impossible. However, photographs appearing in documents are almost always accompanied by descriptive text, either in the form of a caption (a text block located immediately below the photograph) or as part of the main body of text. In such situations, it is possible to extract visual information from the text, resulting in a *conceptualized graph* describing the structure of the accompanying picture. This graph can then be used by a computer vision system in the top-down interpretation of the picture.

The vision literature refers to *schemas*, a symbolic representation describing the appearance of an object (or a collection of objects, *i.e.*, scene), in terms of visual primitives such as lines, regions *etc.* Several vision systems employ pre-defined schemas in the interpretation of images of outdoor scenes, man-made objects *etc.* Such schemas represent typical scenes in a given domain, but are not specific to any given picture. In the case of understanding images with accompanying descriptive text, it is possible to dynamically generate picture-specific schemas, enabling the employment of efficient picture interpretation strategies.

**Meaning:** "Understanding" the picture refers to the labelling of all visually salient objects in the picture along with relevant spatial relationships between objects. Once these objects have been labelled, they can easily be placed into an integrated text-picture database (representing the meaning of the entire document).

**Extracting Visual Information from Text Describing Picture:** Salient information (in terms of understanding the picture) includes (i) information specifying what objects are present in the picture (*e.g.*, "this photo depicts the main engine of the space shuttle along with the rocket boosters"), (ii) information useful in locating these objects (*e.g.*, "it is in between the fueling truck and the hangar"), and (iii) information used to identify (*i.e.* distinguish between) objects of the same class. A simple form of identification is through spatial constraints (*e.g.*, "to the left of"). Processing intuitively non-spatial methods of identification (*e.g.*, "the main engine is larger and has 3 chambers") is considerably more difficult. The main problem to be addressed here is what we refer to as "the correspondence problem". Unlike the previous two segment types (tables and diagrams), there is not a one-to-one correspondence between words and picture elements.

**Picture Processing:** The subsequent interpretation of the picture is guided by information present in the conceptualised graph. A search planner is employed to (i) determine the most appropriate image processing routines to be invoked for a given task, (ii) oversee the order in which various image processing routines are called , and (iii) restrict the search for objects to areas suggested by the caption. The important point is that complex image processing operations are employed only when earlier, simpler operations are not successful. As an example, consider the problem of identifying human faces in a photograph. Assume that an object-location hypothesization process has succeeded in generating possible face candidates (*i.e.*, areas of the image loosely corresponding to the contours of a human face) [22]. It is possible to identify the faces by using spatial constraints given by the caption (*e.g.*, "Tom Smith, left, Mark Brown, centre and ...") [9].

In the example given, we avoid the process of matching face candidates to a database of pre-stored face models. This methodology can be generalised to other object (natural or man-made) classes.

**System Architecture:** A system to perform the above task is organized as a set of processing modules operating on a common knowledge representation module. The processing modules are (i) the Natural Language Processing (NLP) module which generates the conceptualised graph, (ii) the vision module which carries out image processing tasks and (iii) the interpretation module which directs the operation of the vision module and incorporates the results of visual processing into the intermediate representation.

The intermediate representation must be able to efficiently represent both linguistic as well as pictorial information. Since semantic networks have been used in both NLP and vision, they are a natural choice for representing the required knowledge.

## 5.5 Relationship to other Topics

This topic is most closely related to Topics 1 (Modular Architectures for Document Understanding) and 3 (Model-Based OCR). Since we are computing a new representation of the same image data, it is necessary to interact with the module overseeing the data structure representing the meaning of the entire document. For example, the image area corresponding to one row of a table is now converted into a set of entries in a relational database. In the case of image understanding, the picture area corresponding to an object, along with its label can be entered into a pictorial database.

There is also a strong relationship between table, diagram and image understanding and the OCR module. In the cases of tables and diagrams, the OCR must be employed interactively in order to read the entries in a table, or the text labels in a diagram. In the case of images with descriptive text, the OCR must be used in order to read the descriptive text. In the case of pictures with accompanying caption blocks, this is a straight-forward procedure. If the descriptive text is part of the running text, the OCR is required in locating the relevant descriptive sentences (*e.g.*, "In Figure 1 we show ...").

## 5.6 Guidelines for Focusing Research

Table understanding would be a good place to initially focus research for two reasons. First, the meaning of a table is quite well-defined in terms of a relational database. Secondly, the visual processing required, *i.e.* the extraction of horizontal and vertical lines, is computationally speaking, quite simple. A simple extension of the techniques used in understanding tables could be applied to charts (diagrams such as pie-charts, histograms *etc.*).

Specialized diagrams such as maps, engineering drawings, flowcharts, *etc.* would be a good choice to focus research on since the mappings between the inputs (paper-based) and the target representations are well-defined. If it is desired to understand diagrams other than the types mentioned above or images (half-tones), the problem could be made tractable by restricting the input to those situations where accompanying, descriptive text is available.

# 6 Performance Evaluation under Distortion and Noise

## 6.1 Introduction and Background

Performance evaluation of document understanding systems could be an important guide to research. The results of performance evaluation could be used to allocate resources to difficult, unsolved problems that are significant barriers to achieving high performance.

A precise definition of the goal of the system being measured and any intermediate steps that lead to a solution of that goal are essential to achieve a useful performance evaluation. For example, recognition of the ZIP Code in an address image is the goal of a system for postal address processing. Intermediate steps include segmentation of an address into lines, segmentation of the lines into words, determination of the identity of the ZIP Code, segmentation of the ZIP Code into digits, recognition of the isolated digits, and assignment of a confidence level to the ZIP Code decision.

A model for system performance should be developed and the importance of each intermediate step in achieving the overall goal of the system should be defined. This is to insure that research efforts are properly directed. For example, to achieve the best performance possible in ZIP Code recognition, it may not be necessary to have 100 percent correct line segmentation. The effort needed to improve a system from 95 percent to 100 percent correct may be quite substantial. However, the net gain in recognition rate may be minimal. This could be caused by a secondary reason, such as that the ZIP Codes in the other five percent of the images may have already been located correctly. Thus, improving line segmentation would have no effect on ZIP Code recognition performance. Such effects should be predicted by the performance model.

A representative image database and indicative testing procedure should also be defined. The database should reflect the environment in which the system will be applied and the testing procedure should fairly determine the performance of the intermediate steps as well as the final goal of the system. The image database should contain a mixture of stress cases designed to determine the response of the system to various potential problems as well as a random sample of the images the system is expected to encounter. A selective set of stress cases are useful for initial system development. A large random sample, perhaps on the order of tens of thousands of images, is useful for testing a mature system. Such a two-tiered strategy has proven to be quite useful in the evaluation of postal address recognition systems. Initial development of a system for handwritten ZIP Code recognition used approximately 5000 images of addresses with varying levels of difficulty to demonstrate competence. Techniques currently under development will be demonstrated on 30,000 addresses initially and over 100,000 randomly selected images in later trials.

## 6.2 Performance Evaluation for Document Analysis

*System Goal Definition*

The goal of a system for document analysis could be two-fold. A basic approach to document decomposition would locate and label areas of text, graphics, line-drawn figures, photographs, and so on. An enhanced representation would include the logical structure of the document as well. Each of the areas located by the decomposition process could be used in the generation of a data structure such as an SGML hierarchy [23]. Such a reverse mapping from image to generation language would provide an abundance of information that could be used for indexing or information retrieval.

Figure 3 shows an example document and its SGML segmentation. In addition to the decomposition, the value added by the SGML information is shown. The author, title, section headings, citations, text contents, and so on, are indicated.

Given that the goal of a document analysis system was to produce a similar logical structure, useful intermediate steps might include the measurement of decomposition performance. The ability to locate headings, italicized text, footnote indicators, and so on, are necessary precursors to a logical segmentation. The specific graphical characteristics that are needed, and should therefore be detected, are directly related to the end-goal of the system.

*Performance Model Design*

A systematic definition of the desired performance at each of the intermediate levels in a document analysis system is necessary to predict overall system performance. An analytic model based on the sub-components of a system should be derived. The model should be validated by run-time observations.

System performance modeling has been very useful in our development of the next generation address recognition unit (ARU) for the United States Postal Service. The ARU contains many components that have functional similarities in document analysis systems. For example, line and word segmentation as well as character recognition are algorithms common to both domains. ARU performance modeling has helped establish necessary minimum performance levels in all areas and thus has focused research on the topics where effort is needed. A similar effort in general purpose document analysis should also provide useful results.

*Database and Testing Procedure*

The database used for performance evaluation has two essential components: images and ASCII truth. Traditionally, images for a testing database are generated by scanning selected documents. Truth values are applied by a manual process that includes "boxing" regions in the documents and typing in the complete text within the document. This can be labor intensive and error prone. Furthermore, multiple iterations of manual truthing may be needed as system requirements change.

An alternative is to generate test data directly from the ASCII truth [24, 25]. An application

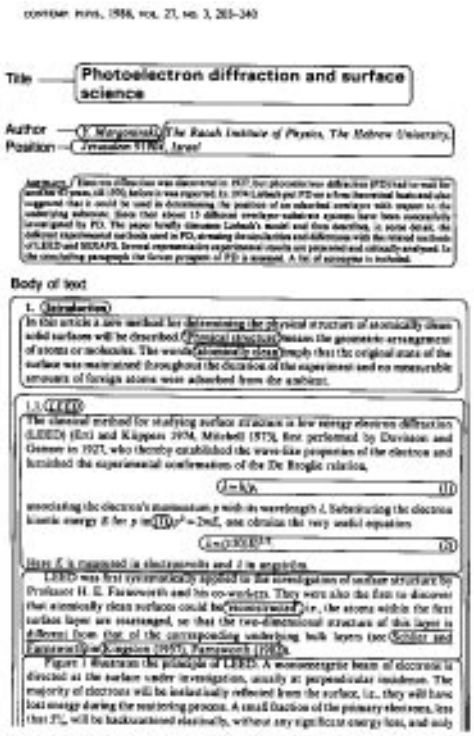


Figure 3: Example SGML structure from "SGML: An author's guide to the Standard Generalized Markup Language," by M. Bryan, Addison Wesley, 1988.

of a similar methodology to document analysis is shown in Figure 4. An SGML representation for a set of documents would be input. The necessary macros would be defined to provide the physical realization of the document. After running a formatting package, the resultant bitmap image of the document would be saved. Such images could then be corrupted with noise to generate test data. Models for different noise sources, such as facsimile or photocopy processes, could be used. It is interesting to note that a similar strategy for developing OCR algorithms based on synthetic noise models has been successful [26].

The advantages of the proposed approach for database generation include the flexibility it provides in the use of different formatting macros. Versions of the same logical document generated in a range of fonts, sizes, styles, and so on, could be utilized. This would allow for the testing of any format-dependent characteristics of the system. Examples of different document formats would not need to be found by an exhaustive search. If it is desired to provide a system with the capability to recognize a certain format of document, that format could be generated synthetically from the database and it would not be necessary to encounter a large number of examples of that format a-priori. An example of this would be if at some date in the future a large number of open-source documents printed in ten point type with three columns per page where each column had a ragged

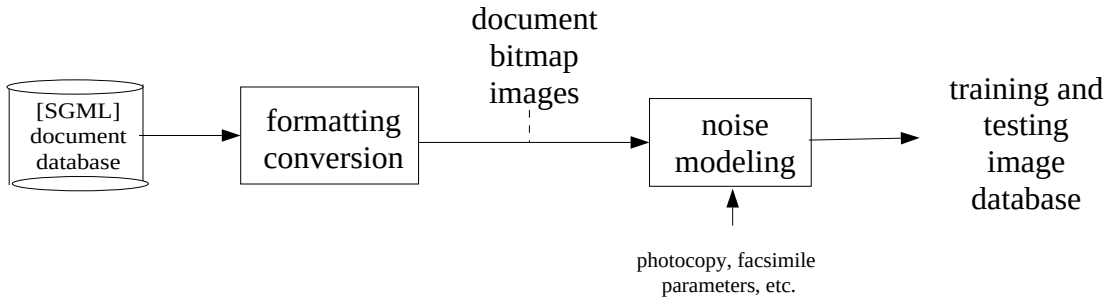


Figure 4: Database generation for document analysis evaluation.

left edge were going to be seen. If a sufficient number of documents in that format were not actually available, they could be generated synthetically. After appropriate training, the system would be ready to process the actual documents when they arrived.

An additional advantage of synthetic data generation is the ability to model noise sources and simulate various reasons for errors. Models for noise caused by repeated photocopying or facsimile transmission would be quite valuable. The performance of document analysis systems operating under various levels of noise could then be characterized. Together with the ability to change formats at will, noise modeling would provide an ideal method for testing a document analysis system under a variety of constraints. Everything from document decomposition to any associated OCR processes could be stress tested.

The procedure used for any comparative testing should be carefully considered. A substantial set of training data, representative of any test data that will be processed, should be provided to all concerned parties. Each group should develop their system on this data and demonstrate performance under a variety of conditions.

One scenario for testing would include the distribution of a quantity of document images without truth. A limited time would be provided for the testing and return of results. A neutral third party would evaluate the performance. Only enough time would be provided for one round of testing.

Another scenario for testing would require participants to install copies of the code for their systems at a neutral location. This party would perform tests on a common database and evaluate results in a standard format. This methodology would eliminate any of the natural bias that occurs when the developers of a system test it themselves.

### 6.3 Conclusions and Future Directions

Evaluation of the performance of document analysis systems was discussed. Meaningful performance evaluation should be related directly to the goals of the system. Intermediate performance measurements should be defined that relate to the end goal of the system and provide useful information about its operation.

A database generation methodology was proposed that produces images from an SGML-type format. Those document images are corrupted by models for different noise sources such as multiple generation photocopies or facsimiles. System performance can then be measured with large varieties of document formats and noise characteristics.

Future work should be directed toward precise definition of pertinent system performance measurements. Exactly what should be measured, and why, as well as the impact of each characteristic should be determined. Also, a methodology for image database generation from ASCII document descriptions should be developed. Tests performed on this database under varieties of conditions could be used to direct document analysis research efforts.

## References

[1] S.N. Srihari. Document Image Understanding. *IEEE Fall Joint Computer Conference*, Dallas, Texas, 1986, pp 87-95.

[2] W. Doster Different States of a Document's Content on its Ways from the Gutenbergian World to the Electronic World. *Proceedings of the Seventh International Conference on Pattern Recognition*, 1984, pp 872-874.

[3] V. Govindaraju, S.W. Lam, D. Niyogi, D. Sher, R.K. Srihari, S.N. Srihari and D. Wang. Newspaper Image Understanding. *Lecture Notes in Artificial Intelligence*, Vol. 444, J. Siekmann (editor), Springer Verlag, NY, NY. 1989, pp 375-386,.

[4] S.W. Lam, A.C. Girardin and S.N. Srihari. Gray-Scale Character Recognition Using Boundary Features. *SPIE/IS&T Symposium on Electronic Imaging Science & Technology*, San Jose, California, 1992.

[5] Y.Y. Tang, C.Y. Suen, C.D. Yan and M. Cheriet. Document Analysis and Understanding: A Brief Survey. *First International Conference on Document Analysis and Recognition*, Saint Malo, France, 1991, pp 17-31.

[6] C. Wang and S.N. Srihari. A Framework for Object Recognition and its Application to Recognizing Address Blocks on Mail Pieces. *International Journal of Computer Vision*, 1987.

[7] S.W. Lam and S.N. Srihari. Multi-Domain Document Layout Understanding. *First International Conference on Document Analysis and Recognition*, Saint Malo, France, 1991, pp 112-120.

[8] H. Brown. Standards for Structured Documents. *The Computer Journal*, Vol. 32, No. 6, 1989, pp 505-514.

[9] R.K. Srihari. *Extracting Visual Information from Text: Using Captions to Label Human Faces in Newspaper Photographs.* Ph.D. Thesis, Department of Computer Science, SUNY at Buffalo, 1991.

[10] K.Y. Wong, R.G. Casey and F.M. Wahl. Document Analysis System. *IBM J. Res. Develop.* **26**, No. 6, 1982, pp 647-656.

[11] G. Nagy, S.C. Seth and S.D. Stoddard. Document Analysis with Expert System. In *Proceedings of Pattern Recognition in Practice II*, Amsterdam, June, 1985.

[12] S.N. Srihari and V. Govindaraju. Textual Image Analysis Using the Hough Transform. *International Journal of Machine Vision and Applications*, **2(3)**, 1989, pp 141-153.

[13] D. Wang and S.N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, **47**, 1989, pp 327-352.

[14] S.N. Srihari. Feature extraction for locating address blocks on mail pieces. *From Pixels to Features*, J.C. Simon (ed.), Elsevier Science Publisher B.V. North-Holland, 1989, pp 261-273.

[15] S.N. Srihari and J.J. Hull. Character Recognition. *Encyclopedia of Artificial Intelligence*, Second Edition, S.C. Shapiro (editor), Wiley Interscience, New York, 1992, pp 138-150.

[16] S.N. Srihari. *Computer Text Recognition and Error Correction.* IEEE Computer Society Press, 1984.

[17] S.N. Srihari, J.J. Hull and R. Choudhari. Integrating Diverse Knowledge Sources in Text Recognition. *ACM Trans. on Office Information Systems*, *1* (1), 1983, pp 68-87.

[18] J.J. Hull. *A Computational Theory of Visual Word Recognition.* Ph.D. Thesis, Department of Computer Science, SUNY at Buffalo, 1988.

[19] T.K. Ho. *A Theory of Multiclassifier Systems and its Application to Visual Word Recognition.* Ph.D. Thesis, Department of Computer Science, SUNY at Buffalo, 1992.

[20] J.J. Hull. Incorporation of a Markov Model of Language Syntax in a Text Recognition Algorithm. In *Proceedings of Symposium on Document Analysis and Information Retrieval*, Las Vega, NV, March 16-18, 1992, pp 174-184.

[21] R.K. Srihari. Combining Statistical and Syntactic Methods in Recognizing Handwritten Sentences. In *preparation.*

[22] V. Govindaraju. *A Computational Model of Face Location.* Department of Computer Science, SUNY at Buffalo, 1992.

[23] A.L. Spitz. Style Directed Document Recognition. *First International Conference on Document Analysis and Recognition,* Saint Malo, France, 1991, pp 611-619.

[24] S. Kahan, T. Pavlidis, H.S. Baird, On the recognition of printed characters of any font and size, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-9, 2, March, 1987, pp 274-288.

[25] J.J. Hull, S. Khoubyari, T.K. Ho, Visual Global Context: Word Image Matching in a Methodology for Degraded Text Recognition, *Symposium on Document Analysis and Information Retrieval* Las Vegas, Nevada March, 1992.

[26] H.S. Baird, Document defect models, *IAPR Workshop on Syntactic and Structural Pattern Recognition,* Murray Hill, New Jersey, June 13-15, 1990, pp 38-46.