Retrieving ClipArt Images by Content*

Manuel J. Fonseca, B. Barroso, P. Ribeiro, Joaquim A. Jorge

Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
Fax: +351.21.3145843
mjf@inesc-id.pt, {bamb,pdsr}@mega.ist.utl.pt, jaj@inesc-id.pt

Abstract. Nowadays there are a lot of vector drawings available for inclusion into documents, which tend to be achieved and accessed by categories. However, to find a drawing among hundreds of thousands is not easy. While text-driven attempts at classifying image data have been recently supplemented with query-by-image content, these have been developed for bitmap-type data and cannot handle vectorial information. In this paper we present an approach to index and retrieve ClipArt images by content, using topological and geometric information automatically extracted from drawings. Additionally, we introduce a set of simplification heuristics to eliminate redundant information and useless elements. Preliminary usability tests to our prototype show promising results and suggest good acceptance of sketching as a query mechanism by users.

1 Introduction

Currently there are a huge number of drawings that users can integrate into their documents. However, to use one of those images, they have to browse through large and deep file directories or navigate a complex maze of categories previously defined to organize drawings. Furthermore, such search becomes humanly impossible when the number of drawings increases. One possible solution is to manually catalog all drawings by adding textual descriptions. However, this approach is not satisfactory, because it forces users to know in detail the meta-data used to characterize drawings. Yong Rui [1] analyzed several content-based image retrieval systems that use color and texture as main features to describe image content. On the other hand, vector drawings are represented in structured form requiring different approaches from image-based methods.

* This work was funded in part by the Portuguese Foundation for Science and Technology, project 34672/99 and the European Commission, project SmartSketches IST-2000-28169.

In the past years there have been some research works in retrieving drawings. Gross' Electronic Cocktail Napkin [2] addressed a visual retrieval scheme based on diagrams, to indexing databases of architectural drawings. Berchtold's S3 system [3] supports managing and retrieving industrial CAD parts, through contour matching. Park's approach [4] retrieves mechanical parts based on dominant shapes and spatial relationships. Leung proposed a sketch retrieval method [5] for general unstructured free-form hand-drawings.

We can observe two things from existing content-based retrieval systems for drawings. The first is scalability: most published works use databases with few elements (less than 100). The second is complexity: drawings stored in the database are simple elements not representing sets of real drawings, such as ClipArt images.

We will now describe our approach to retrieve drawings by content privileging the use of spatial relationships and geometric information. Moreover, we perform automatic simplification, classification and indexation of existing drawings, to make the retrieval process both more effective and accurate. Additionally, fast and efficient algorithms to perform similarity matching between sketched queries and a large database of ClipArt drawings are required. Finally, we implemented a prototype to retrieve WMF ClipArts and performed usability tests, which show very encouraging results and suggest good acceptance of sketching by users.

2 Our Approach to Retrieve Vector Drawings

Our approach solves both scalability and complexity problems by developing mechanisms for retrieving drawings, in electronic format through hand-sketched queries, taking advantage of user's natural ability at sketching and drawing. Moreover, unlike the majority of existing systems, our method was developed to support large sets of drawings. To that end, we devised a multidimensional indexing structure that scales well with growing data set size. Figure 1.a shows our system architecture, identifying its main components.

2.1 Classification

Content-based retrieval of pictorial data, such as digital images, drawings or graphics, uses features extracted from the correspond-

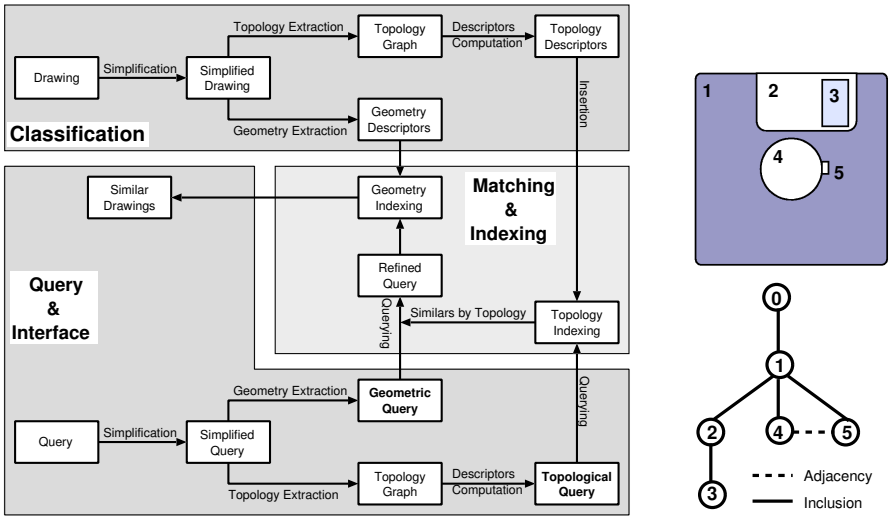

Fig. 1. a) System architecture for our approach. b) ClipArt drawing (top) and correspondent topology graph (bottom).

ing picture. Typically, two kinds of features are used. Visual features encode information, such as color, texture and shape. Relationship features describe topological and spatial relationships among objects in a picture. However, for vectorial drawings, color and texture are irrelevant features. We focus on topology and geometry.

Our classification process starts by applying a simplification step, to eliminate most useless elements. The majority of ClipArt drawings contains many details, which are not necessary for a visual query and increase the cost of searching. We try to remove visual details (i.e. small-scale features) while retaining the perceptually dominant elements and shapes in a drawing. The main goal of this step is to reduce the number of entities to analyze in subsequent steps of the classification process, in order to speed up queries.

After simplification we identify visual elements, namely polygons and lines, and extract shape and topological information from drawings. We use two relationships, **Inclusion** and **Adjacency**, which are a simplified subset of the topological relationships defined by Egenhofer [6]. Relationships thus extracted are compiled in a Topology Graph, where "parent" edges mean `Inclusion` and "sibling" connections mean `Adjacency`, as illustrated in Figure 1.b. While

these relationships are weakly discriminating, they do not change with rotation and translation.

However, topology graphs are not directly used for searching similar drawings, since graph matching is a NP-complete problem. We use the corresponding graph spectra instead. For each topology graph to be indexed in a database we compute descriptors based on its spectrum [7]. In this way, we reduce the problem of isomorphism between topology graphs to computing distances between descriptors. To support partial drawing matches, we also compute descriptors for sub-graphs of the main graph. Moreover, we use a new way to describe drawings hierarchically, by dividing them in different levels of detail [8] and then computing descriptors at each level. This combination of sub-graphs descriptors and levels of detail, provides a powerful way to describe and search both for drawings or sub-parts of drawings, which is a novel feature of our work.

To acquire geometric information about drawings we use a general shape recognition library called CALI [9]. This enables us to use either drawing data or sketches as input, which is a desirable feature of our system. We use CALI to compute a set of geometric features such as area and perimeter ratios from special polygons such as the convex hull, the largest area triangle inscribed in the convex hull or the smallest area enclosing rectangle, among others. Using geometric features instead of polygon classification, allows us to index and store potentially unlimited families of shapes. Experimental evaluation [10] revealed that this technique outperforms other methods to describe shapes, such as Fourier descriptors, grid-based descriptors or Delaunay triangulation, yielding better precision figures for all recall values. We obtain a complete description of geometry in a drawing, by applying this method to each geometric entity of the figure. The geometry and topology descriptors thus computed are inserted in two different indexing structures, one for topological information and another for geometric information, respectively.

2.2 Query and Matching

Our system includes a Calligraphic Interface to support the specification of hand-sketched queries, to supplement and overcoming limitations of conventional textual methods. The query component

performs the same steps as the classification process, namely simplification, topological and geometric feature extraction, topology graph creation and descriptor computation. This symmetrical approach is unique to our method. In an elegant fashion two types of information (vector drawings + sketches) are processed by the same pipeline.

We developed a new multidimensional indexing structure, the NB-Tree [11], which provides an efficient indexing mechanism for high-dimensional data points. The NB-Tree is a simple, yet efficient indexing structure, using dimension reduction. It maps multidimensional points to a 1D line by computing their Euclidean Norm. In a second step we sort these points using a B^+-Tree on which we perform all subsequent operations.

Computing the similarity between a hand-sketched query and all drawings in a database can entail prohibitive costs especially when we consider large sets of drawings. To speed up searching, we divide our matching scheme in a two-step procedure. First, we select a set of drawings topologically similar to the query, by performing a KNN query to the topology indexing structure. This step works as a filter, reducing the number of potential candidates to compare in the next step. Second, we use geometric information to further refine the set of candidates.

3 Simplification Heuristics

To simplify drawings we used a set of heuristics that explore their specific features and human perception. This reduces both the information present in drawings, storage space and processing time. We focus our heuristics in three particularities of ClipArt drawings: color gradients, contours and small area polygons.

Color Gradient Many ClipArt drawings use continuous and overlapped polygons with small changes in color, to achieve a gradient effect. Since our approach describes drawings using only topology and geometry, color is not relevant for retrieval. However, we use color information to simplify drawings, by grouping polygons with similar colors into a single polygon.

Contour Lines We found out that many shapes were defined using two polygons, one to specify the filled region and another just

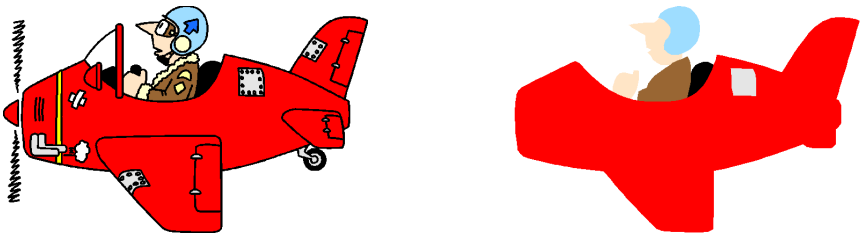

Fig. 2. Application of Heuristic 3. Original (left) with 591 polygons and simplified (right) with 13 polygons.

to define the contour. Since the second polygon do not convey any additional information, this heuristic goal is to eliminate them.

Small Area Polygons ClipArt drawings have a lot of small area polygons to describe details that users ignore while specifying queries. This heuristic discards small polygons, when comparing to the largest one. The biggest challenge here was the definition of "small". To that end we used several percentages of the biggest polygon during simplification and asked users to analyze the results. We also considered trade-offs between simplification and precision values.

4 Evaluation and Experimental Results

Whereas the critical step in classification (using our approach) is drawing simplification, in nearest neighbor matching search dominates the resource usage.

Simplification and Classification To determine the degree of simplification we applied the three heuristics to a set of 30 drawings randomly selected and we counted the number of polygons and lines before and after simplification. We found out that for this set we achieved a simplification degree of around 80%, on average, for lines and polygons. It is important to notice that after simplification, users still recognize drawings.

We also measured classification times on a AMD Duron @ 1.3GHz with 448MB of RAM, running Windows XP. We classified 968 drawings in 7 minutes and 55 seconds, yielding an average of 0.49 seconds

per each drawing. This is the overall classification time, which includes simplification, geometric and topological feature extraction, descriptors computation and insertion in the indexing structures. The resulting indexing structures required a storage space of 16.8 MB (excluding drawings). We can consider that the classification process is fast and that the storage space required is relatively small, making this approach suitable for large data sets of drawings.

Indexing Structure We shortly describe experimental comparison of our indexing structure (NB-Tree) to the most popular approaches available, such as the SR-Tree, the A-Tree and the Pyramid Technique. From Figure 3 we can see that the NB-Tree outperforms all the structures evaluated when data dimension and data set size increases. A more detailed evaluation can be found in [11].

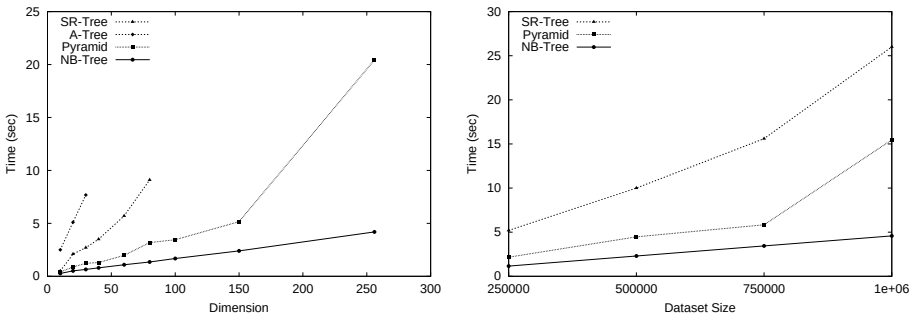

Fig. 3. Search times for K-NN as a function of dimension (left) and data set size (right).

ClipArt Retrieval We developed a prototype to retrieve ClipArt drawings through hand-sketched queries (see Figure 4). On the top-left we can see the sketch of a cloud and on the bottom results returned by the implied query, ordered from left to right. We also provide a way to perform Query-by-Example allowing the user to select a result and use it as query.

In order to assess acceptance and recognition-level performance, we conducted preliminary usability tests involving twelve users and a database of 968 drawings from several categories. These drawings were classified using our hierarchical scheme to produce descriptors

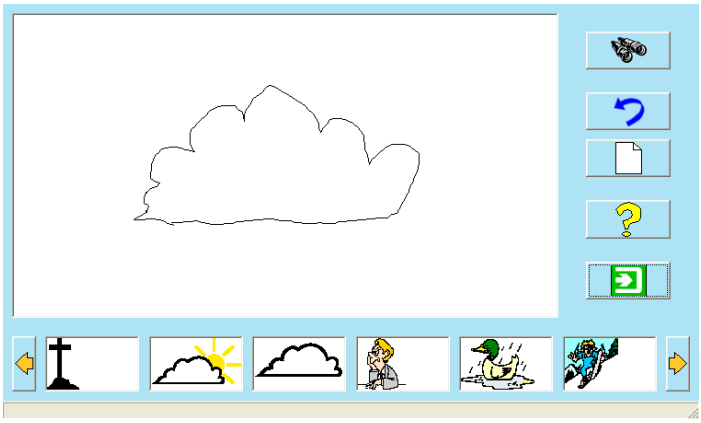

Fig. 4. ClipArt finder prototype.

for each level of detail and for each subpart. Resulting descriptors were then inserted into two databases, one for topology and another for geometry, using our NB-Tree.

Task 1 comprises the searching of a drawing by providing a verbal description of objects. The main goal of this task was to measure user satisfaction about returned results when using sketches and Query-by-Example. Results revealed that searching using sketches was in general less successful than using Query-by-Examples. This is due mainly to people drawing skills. However, when users resorted to Query-by-Example functionality, they found results more satisfactory. Query times, using an AMD Duron @ 1.3GHz were between 1 and 2 seconds, which most users found satisfactory.

In Task 2 we asked users to search for drawings depicted on a paper and we checked in what position the corresponding drawing appeared in results. We observed that the best results were achieved for drawings containing collections of easy-to-draw shapes and with a strong topological component. In these cases, the topological filtering is effective and reduces the number of drawings to compare in the geometric matching. Furthermore, easy-to-draw shapes assure that users will sketch something very similar to the desired drawing.

Finally, we collected users opinions through a questionnaire. Users liked the interaction paradigm very much (sketches as queries), were satisfied with returned results and pleased with the short time they had to spend to get what they wanted.

5 Conclusions and Future Work

We have presented a generic approach suitable for content-based retrieval of drawings. Our method hinges on recasting the general picture matching problem as an instance of graph matching using vector descriptors. To this end we index drawings using a *topology graph* which describes adjacency and containment relations for parts and subparts. We then transform these graphs into descriptor vectors in a way similar to hashing to obviate the need to perform costly graph-isomorphism computations over large databases, using spectral information from graphs. Finally, a novel approach to multidimensional indexing provides the means to efficiently retrieve sub-drawings that match a given query in terms of its topology. We described the overall process to simplify drawings using a set of heuristics and usability tests performed using our prototype to retrieve ClipArts. Users were generally pleased with both the returned drawings and using sketches as the main query mechanism. We are currently working towards converting this application into a Sketch-Based Web search engine for ClipArt drawings.

References

1. Rui, Y., Huang, T.S., Chang, S.F.: Image Retrieval: Current Techniques, Promising Directions, and Open Issues. Journal of VCIR **10** (1999) 39–62
2. Gross, M., Do, E.: Demonstrating the Electronic Cocktail Napkin: a paper-like interface for early design. In: Proceedings of CHI'96. (1996) 5–6
3. Berchtold, S., Kriegel, H.P.: S3: Similarity in CAD Database Systems. In: Proc. of the Int. Conference on Management of Data (SIGMOD'97). (1997)
4. Park, J., Um, B.: A New Approach to Similarity Retrieval of 2D Graphic Objects Based on Dominant Shapes. Pattern Recognition Letters **20** (1999) 591–616
5. Leung, W.H., Chen, T.: Hierarchical Matching for Retrieval of Hand-Drawn Sketches. In: Proceedings of IEEE ICME'03. (2003) 29–32
6. Egenhofer, M.J., Al-Taha, K.K.: Reasoning about Gradual Changes of Topological Relationships. Volume 639 of LNCS. Springer-Verlag (1992) 196–219
7. Cvetkovic, D., Rowlinson, P., Simic, S.: Eigenspaces of Graphs. Cambridge University Press, United Kingdom (1997)
8. Fonseca, M.J., Jr., A.F., Jorge, J.A.: Content-Based Retrieval of Technical Drawings. Special Issue of IJCAT (to appear) (2004)
9. Fonseca, M.J., Jorge, J.A.: Experimental Evaluation of an on-line Scribble Recognizer. Pattern Recognition Letters **22** (2001) 1311–1319
10. Fonseca, M.J., Barroso, B., Ribeiro, P., Jorge, J.A.: Retrieving Vector Graphics Using Sketches. In: Proc. of the Smartgraphics Symposium'04 (to appear). (2004)
11. Fonseca, M.J., Jorge, J.A.: Indexing High-Dimensional Data for Content-Based Retrieval in Large Databases. In: Proceedings of DASFAA'03. (2003) 267–274