

Sketch-Based Retrieval of Drawings using Topological Proximity

Pedro Sousa Manuel J. Fonseca

Department of Computer Science and Engineering

INESC-ID/IST/Technical University of Lisbon

R. Alves Redol, 9, 1000-029 Lisboa, Portugal

pedrocrs@gmail.com, mjf@inesc-id.pt

Abstract

Currently, there are large collections of drawings from which users can select the desired ones to insert in their documents. However, to locate a particular drawing among thousands is not easy. In our prior work we proposed an approach to index and retrieve vector drawings by content, using topological and geometric information automatically extracted from figures. In this paper, we present a new approach to enrich the topological information by integrating spatial proximity in the topology graph, through the use of weights in adjacency links. Additionally, we developed a web search engine for clip art drawings, where we included the new technique. Experimental evaluation reveal that the use of topological proximity results in better retrieval results than topology alone.

1. Introduction

Nowadays, there are a lot of vector drawings available for integration into documents, either on the Internet or on clip art collections sold in optical media. This large number of drawings makes traditional searching mechanisms, based on browsing and navigation in directories or complex mazes of categories, inadequate. Furthermore, solutions using keywords or tagging are also impracticable since they have to be generated manually. A more adequate solution must take into account information automatically extracted from the content of drawings, instead of information manually generated by people. Although there are several solutions for Content-Based Image Retrieval (CBIR), they cannot be applied to vector drawings, because these are described in a structural manner, requiring different approaches from those used for raster images.

In our prior work [4], we proposed an automatic visual classification scheme based on topology and geometry, to retrieve vector drawings. Our solution takes advantage of users' visual memory and explores their ability to

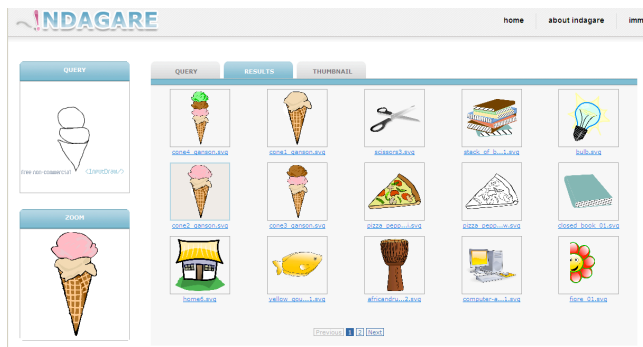


Figure 1. Prototype for drawing retrieval.

sketch as a query mechanism. We used a graph-based technique to describe the spatial arrangement of drawing components, coding topological relationships of inclusion and adjacency through the specification of links between nodes of the graph. Additionally, we used a multidimensional indexing method that efficiently supports large sets of drawings, in combination with new schemes that allow us to hierarchically describe drawings and subparts of drawings by level of detail. This way we are able to perform searches using coarse approximations or parts of the desired drawing.

In this paper, we propose and evaluate a new mechanism to describe the spatial arrangement of elements in a drawing, which takes into account their proximity. To validate this we developed a prototype for the retrieval of clip art drawings, in SVG format (see Figure 1). The prototype allows the search of drawings using sketches, keywords and query by example. Experimental evaluation with users showed that the inclusion of information about proximity in the topology graph increases the precision of our system.

The rest of the paper is organized as follows: Section 2 provides a summary of related work in content-based retrieval of drawings. In section 3 we present an overview of our system architecture. Section 4 describes how we code

proximity into our topology graph. In Section 5, we describe the prototype and the experimental evaluation, comparing the solutions with and without proximity. Finally, in section 6 we conclude and enumerate some future work.

2. Related Work

In the past years there has been a great focus in querying Multimedia databases by content. However, most such work has focused on image databases disregarding the retrieval of vector drawings. Due to their structure these require different approaches from image-based methods, which resort to color and texture as main features to describe content. In the next paragraphs we describe some approaches for content-based retrieval of drawings.

One of the first works dedicated to the retrieval of drawings was Gross's Electronic Cocktail Napkin [7]. This system addressed a visual retrieval scheme based on diagrams, to indexing databases of architectural drawings. Users draw sketches of buildings, which are compared with annotations (diagrams), stored in a database and manually produced by users. Even though this system works well for small sets of drawings, the lack of automatic indexation and classification makes it difficult to scale the approach to real collections of drawings.

In the work of Beretti and Del Bimbo [1] shapes from a drawing are decomposed into tokens that correspond to protrusions of the curve. To compute the similarity between shapes, authors verify if the two shapes share tokens with similar curvature and orientation, within a given threshold. However, the efficiency of the similarity computation depends on the number of tokens in each shape and does not take into account the token order.

Leung and Chen proposed a sketch retrieval method [10] for general free-form hand-drawings stored as multiple strokes. They use shape information from each stroke exploiting the geometric relationship between multiple strokes for matching. Later on, authors improved their system by considering spatial relationships between strokes [11]. Authors use a graph based description, similar to ours, but describing only inclusion, while we also describe adjacency. Their technique has two drawbacks, complexity, since they use a restricted number of basic shapes (circle, line and polygon) and scalability.

Another approach for matching hand-drawn sketches is the line-based representation of sketches proposed by Nambodiri and Jain [13]. In order to skirt around the problem of identifying basic shapes from a sketch, a drawing is represented as a set of straight lines, which is very dependent of the way users draw sketches.

Liang et al. [12] developed a solution for drawing retrieval based on our prior solution [4]. Authors included some differences, such as the use of eight topological rela-

tionships and relevance feedback. Additionally, they segment sketches using vertices, drawing speed and curvature. By using eight topological relationships, the description and comparison will be more restrictive, producing less results.

Pu and Ramani, developed two methods to describe drawings as a whole [9]. One uses the 2.5D spherical harmonics to convert a 2D drawing into a 3D representation, which is independent to rotations. The other method, the 2D shape histogram, creates a signature with the shape distribution, by computing values for points in the surface of the shape. This method is independent of transformations, insensitive to noise, simple and fast. After experimental evaluation, authors decided to combine both methods to get a better descriptor and to increase the system accuracy.

Recently Hou and Ramani [8] presented an approach for contour shape matching of engineering drawings, inspired by the divide and conquer paradigm. They divide the original shape into two levels of representation, a higher level with structure and a lower level with geometry. During matching, they first use the structure level and then the geometry level, to find similar shapes.

From the content-based retrieval systems described above we can observe two things: most published works rely mainly on the geometric description of drawings (mainly contours), discarding the spatial arrangement of drawing items. Second, those who use topology to describe the content of drawings do not explore the proximity between drawing elements, to get more precise results.

3. Overview of the System

The new algorithm developed to code proximity between items in a drawing was integrated in our general framework for sketch-based retrieval of drawings, developed previously [4]. To give context to the reader and to explain some of the topics needed to describe our new proximity mechanism, we shortly present an overview of the overall framework, describing its main components.

Our framework allows the classification, indexing and retrieval of complex vector drawings, such as CAD drawings or clip art drawings. To that end, it uses spatial relationships, geometric information and indexing mechanisms, as illustrated in the architecture on Figure 2.

3.1. Classification

In the context of vector drawings, features such as color and texture, used mainly in the domain of digital images, are not very expressive. Instead, features related to the shape of objects (geometry) and to their spatial arrangement (topology) are more descriptive of drawing contents. So, in our framework we focus on topology and geometry as main features.

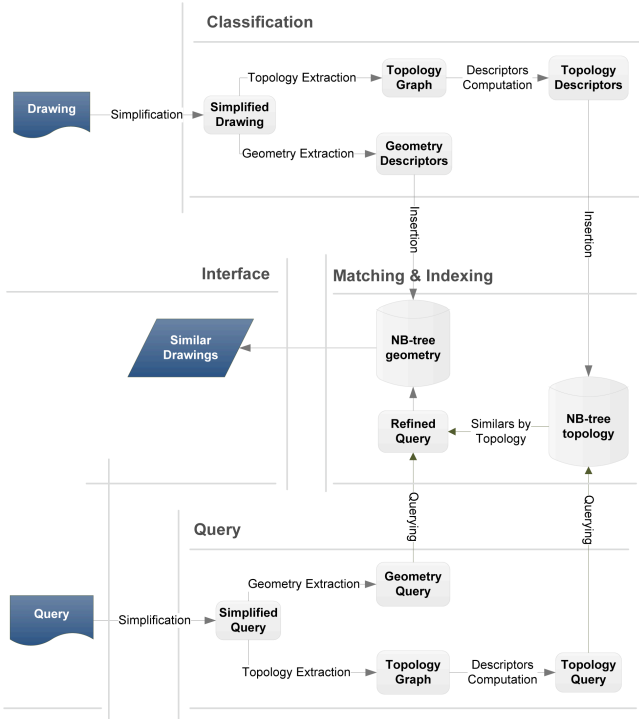


Figure 2. Framework architecture.

Our classification process starts by applying a simplification step, to eliminate most useless elements. The majority of drawings contain 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. This way we reduce the number of entities to analyze in subsequent steps of the classification process, speeding up queries.

After simplification we identify visual elements, namely polygons and lines, and extract geometric and topological information from drawings. We use two relationships, **Inclusion** and **Adjacency**, which are a simplified subset of the topological relationships defined by Egenhofer [3]. Relationships thus extracted are compiled in a **Topology Graph**, where "parent" edges mean Inclusion and "sibling" connections mean Adjacency, as illustrated in Figure 3. While these relationships are weakly discriminating, they do not change with rotation and translation.

Since graph matching is a NP-complete problem, we are not directly using topology graphs for searching similar drawings. We use the corresponding graph spectra instead. For each topology graph to be indexed in a database we compute descriptors based on its spectrum [2]. 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

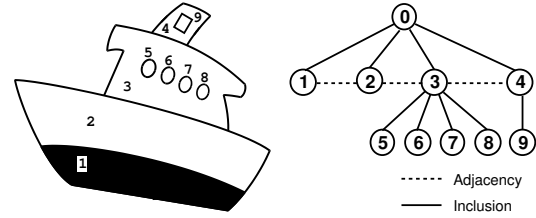


Figure 3. Drawing and Topology graph.

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 and then computing descriptors at each level [4]. This combination of sub-graph descriptors and levels of detail, provides a powerful way to describe and search both for drawings or sub-parts of drawings.

To acquire geometric information about drawings we use a general shape recognition library called CALI [6]. This enables us to use either drawing data or sketches as input. 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 into two different indexing structures, one for topological information and another for geometric information, respectively.

3.2 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.

To improve the searching performance while using large databases of drawings, we included a multidimensional indexing structure in our framework. This indexing structure, the NB-Tree [5], is a simple, yet efficient indexing structure, which uses dimension reduction. It maps multidimensional points to a 1D line by computing their Euclidean Norm. In a second step points are sorted using a B+-Tree on which all subsequent operations are performed.

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, then we use geometric information to further refine the set of candidates.

4. Topological Proximity

In our previous solution we converted spatial relationships (inclusion and adjacency), between visual elements in a drawing, into a topology graph as illustrated in Figure 3. This graph has a well defined structure, being very similar to "a rooted tree with side connections". It has always a root node, representing the whole drawing. Sons from the root represent the dominant blocks (polygons) from the drawing, i.e. blocks that are not contained in any other block. The next level of the graph describes polygons contained by the blocks identified before. This process is applied recursively until we get the complete hierarchy of blocks. As a conclusion, we can say that each graph level adds more drawing details. So, by going down in the depth of the graph, we are "zooming in" in drawing details.

To skirt the problem of graph isomorphism, we use the graph spectra to convert graphs into feature vectors. This way, we reduce the problem of isomorphism between topology graphs to the more simple computation of distances between descriptors.

To generate the graph spectrum we first create the adjacency matrix of the graph, second we calculate its eigenvalues and finally we sort the absolute values to obtain the topology descriptor (see Figure 4). The resulting descriptor is a multidimensional vector, whose size depends on graph (and corresponding drawing) complexity. Very complex drawings will yield descriptors with higher dimensions, while simple drawings will result in descriptors with lower size.

We assume that our topology graphs are undirected graphs, yielding symmetric adjacency matrices and assuring that eigenvalues are always real. Furthermore, by computing the absolute value and sorting it decreasingly, we exploit the fact that the largest eigenvalues are more informative about the graph structure. Additionally, the largest eigenvalues are stable under minor perturbation of the graph structure [2], making the topological descriptors also stable.

Although, isomorphic graphs have the same spectrum, two graphs with the same spectrum need not be isomorphic. More than one graph can have the same spectrum, which gives rise to collisions similar to these in hashing schemes.

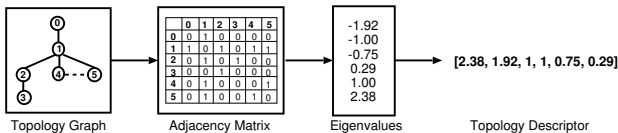


Figure 4. Block diagram for topology descriptor computation.

However, from experiences performed with 100,000 randomly generated graphs versus a set of 10 candidate similar graphs, we have observed that collisions with descriptors of very different graphs still allow us to retrieve the most likely graphs reliably.

While this solution produced good results in the past, we notice that in some cases results could be improved if we take into account the distance between the visual elements in a drawing. To that end we devised a new mechanism to include proximity into our topology graph. Our goal is to be able to differentiate between a drawing with two polygons which are close together and a drawing with two polygons that are far apart, as illustrated in Figure 5.



Figure 5. Using the adjacency weight to differentiate between far and near objects.

To code proximity in the topology graph, we associate weights to the adjacency links of the graph. While in our previous solution we only have an adjacency link when two primitives are connected, now we compute the (normalized) distance between two elements and use this value as the weight of the link. This change in the weights of the topology graph does not affect the stability and robustness of eigenvalues, as ascertained by Sarkar and Boyer [14].

5. Experimental Evaluation

We developed a search engine prototype for vector drawings, using our sketch-based retrieval framework and the new mechanism to describe topological proximity. The database of the system was filled with a set of SVG clip art drawings and experimental evaluation with users was carried out to compare the accuracy of the new algorithm against the previous one.

5.1 Indagare - The Drawing Search Engine

Our drawing search engine prototype, called Indagare (see Figure 1), supports the retrieval of SVG clip art drawings, using sketches, an existing SVG drawing or keywords as queries. This prototype integrates all the functionalities provided by the framework, namely, simplification mechanisms, an indexing structure to optimize the search, geometric description of visual elements and the new developed algorithm to take advantage of proximity.

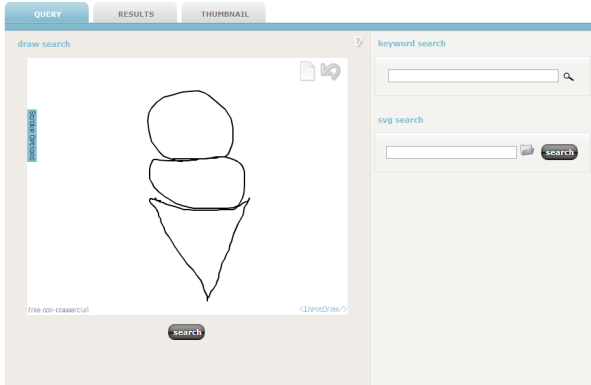


Figure 6. Sketch to search for an ice-cream.

Figure 6 shows the sketch of a query, while Figure 7 presents the results returned by the implied query. If the user wants, he can submit an existing drawing in SVG format or search by keywords (input fields on top right of Figure 6). Moreover, users can also select one of the results and use it to perform Query-By-Example.

5.2 Experimental Results

To evaluate our new approach of coding proximity into the topology graph, we carried out an experiment with ten users. Six of them were male and four were female, with ages between 18 and 58 years old. None of them had previous experience with tablet devices or any other pen-based system.

Our data set of clip art drawings was composed of 20 categories of five drawings each, selected from the Open-Clipart library, yielding a total of 100 vector drawings.

Tests were conducted in two steps. First, we collected the queries by asking each user to draw three sketches, using a digitizing tablet: a balloon, a car and a house. Afterwards, and only at this time, we show all the 100 drawings in the database and requested them to identify the drawings that they considered most similar to each of the sketches they drew, without taking into account their semantic value.

The second step was carried without users' intervention. From the similar drawings selected by the participants, we identified the five more voted, and considered those as the "correct" results for each sketch. Then, we submitted the three sketched queries from each participant to the system and collected the returned results. We configured the system to retrieve 30 results for each query. With these results we computed precision and recall values.

In this experimental test, we evaluated four different system configurations. Besides testing the use of proximity we also evaluated the order in which we perform the matching steps. Typically, our framework performs first a compari-

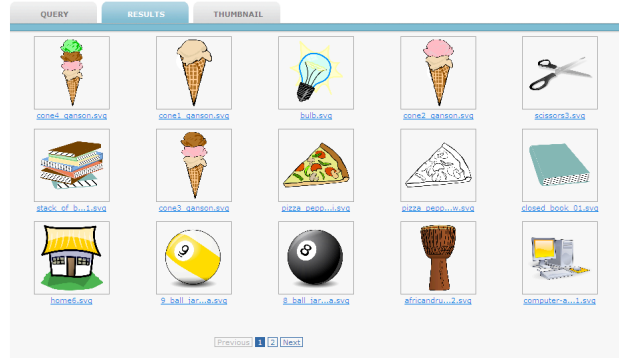


Figure 7. Returned results for the query sketched in Figure 6.

son by topology and then compares the geometry of those topologically similar. Here in these tests, we also tested the other possibility, first a comparison by geometry and then by topology. The goal was to check which feature produces best results as a first filter, geometry or topology.

In summary, we tested the following configurations: i) topology plus geometry; ii) topology with proximity plus geometry; iii) geometry plus topology; and iv) geometry plus topology with proximity. To evaluate the quality of the retrieved results, we calculated precision & recall levels for each configuration, using the 11-Point Interpolated Average Precision method. Precision is the fraction of retrieved drawings that were relevant, while recall is the fraction of relevant drawings that were retrieved. The mean precision for each recall value, of the four configurations is presented in Table 1.

Table 1. Precision of the four configurations.

Recall	Conf. i)	Conf. ii)	Conf. iii)	Conf. iv)
0	0,123	0,130	0,096	0,099
0.1	0,123	0,130	0,096	0,099
0.2	0,123	0,130	0,096	0,099
0.3	0,109	0,116	0,082	0,084
0.4	0,109	0,116	0,082	0,084
0.5	0,039	0,057	0,043	0,045
0.6	0,039	0,057	0,043	0,045
0.7	0,000	0,004	0,005	0,005
0.8	0,000	0,004	0,005	0,005
0.9	0,000	0,000	0,000	0,000
1	0,000	0,000	0,000	0,000

The first thing that we can observe from Table 1 is that filtering firstly by topology yields better results than by geometry. Second, by introducing the proximity notion in the topology graph we can improve precision in both configura-

tions (Topology filtering and Geometry filtering). However, with geometry filtering we only achieve a 0,1% increase, while in the topology filtering the improvement reaches one percent.

The small improvement in the Geometry filtering configuration was foreseeable, because the adjacency weights only play a relevant role in the topology refinement. Therefore, if the geometry filtering retrieves poor results, there is not much that the adjacency weights can do.

6. Conclusions and Future Work

In this paper, we propose a new way to describe the spatial arrangements of visual elements in a drawing. We included the notion of proximity and coded it in the topology graph through the use of adjacency weights. This new algorithm was integrated in our generic framework for sketch-based retrieval of drawings, which recast the general drawing matching problem as an instance of graph matching using vector descriptors. Topology graphs, which describe adjacency and containment relations, are transformed into descriptor vectors, using spectral information from graphs.

The use of proximity to describe the spatial arrangement gets our matching algorithm closer to the human perception, and therefore improving the retrieval effectiveness of our system. This improvement was validate through experimental evaluation with users.

Despite the complete spatial characterization of drawings provided by the use of topological relationships and proximity, the improvement achieved was small (only 1%). These results confirm informal conclusions achieved previously. Clip art drawings, contrarily to technical drawings, are more geometric than topological. Moreover, during tests with users we observed that users typically draw a very small number of shapes, and consequently do not specify topology, but only geometry.

So, improvements in the topological algorithm will produce a small impact in the final results. Additionally, our experimental tests showed that the geometric filtering needs to be improved. To overcome this we are currently developing a new algorithm to compare the geometry between drawings. Informal tests with a preliminary version revealed significant improvements in the precision values, which make us believe that we will be able to achieve better retrieval results in a very near future.

Acknowledgments

This work was funded in part by the Portuguese Foundation for Science and Technology, project A-CSCW, PTDC/EIA/67589/2006.

References

- [1] S. Berretti and A. Del Bimbo. Retrieval by shape similarity with perceptual distance and effective indexing. *Multimedia, IEEE Transactions*, 2(4):225–239, 2000.
- [2] D. Cvetković, P. Rowlinson, and S. Simić. *Eigenspaces of Graphs*. Cambridge University Press, United Kingdom, 1997.
- [3] M. J. Egenhofer and K. K. Al-Taha. Reasoning about gradual changes of topological relationships. *LNCS*, 639:196–219, 1992.
- [4] M. J. Fonseca, A. Ferreira, and J. A. Jorge. Content-Based Retrieval of Technical Drawings. *International Journal of Computer Applications in Technology (IJCAT)*, 23(2–4):86–100, 2005.
- [5] M. J. Fonseca and J. A. Jorge. Indexing High-Dimensional Data for Content-Based Retrieval in Large Databases. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications (DASFAA'03)*, pages 267–274, Kyoto, Japan, Mar. 2003. IEEE Computer Society Press.
- [6] M. J. Fonseca, C. Pimentel, and J. A. Jorge. CALI: An Online Scribble Recognizer for Calligraphic Interfaces. In *Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding*, pages 51–58, Palo Alto, USA, Mar. 2002.
- [7] M. Gross and E. Do. Demonstrating the Electronic Cocktail Napkin: a paper-like interface for early design. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'96)*, pages 5–6, 1996.
- [8] S. Hou and K. Ramani. Structure-oriented contour representation and matching for engineering shapes. *Computer Aided Design*, 40(1):94–108, 2008.
- [9] P. Jiantao and K. Ramani. On visual similarity based 2d drawing retrieval. *Computer Aided Design*, 38(3):249–259, 2006.
- [10] W. H. Leung and T. Chen. User-Independent Retrieval of Free-Form Hand-Drawn Sketches. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'02)*, volume 2, pages 2029–2032, Orlando, Florida, USA, May 2002. IEEE Press.
- [11] W. H. Leung and T. Chen. Hierarchical Matching for Retrieval of Hand-Drawn Sketches. In *Proceedings of the IEEE International Conference on Multimedia and Exposition. (ICME'03)*, volume 2, pages 29–32, Baltimore, USA, July 2003. IEEE Press.
- [12] S. Liang, Z.-X. Sun, B. Li, and G.-H. Feng. Effective sketch retrieval based on its contents. In *Proceedings of Machine Learning and Cybernetics*, volume 9, pages 5266–5272, 2005.
- [13] A. M. Namboodiri and A. K. Jain. Retrieval of on-line hand-drawn sketches. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR'04, Vol. 2*, pages 642–645, Washington, DC, USA, 2004. IEEE Computer Society.
- [14] S. Sarkar and K. Boyer. Quantitative Measures of Change Based on Feature Organization: Eigenvalues and Eigenvectors. Technical report, Image Analysis Research Lab, University of South Florida, 1996.