Sketch-Based Retrieval of Drawings using Spatial 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

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. However, the increase in precision was not as high as we expected. To understand why, we analyzed sketched queries performed by users in previous experimental sessions and we present here the achieved conclusions.

Key words: Sketch-Based Retrieval, Vector Drawing Retrieval, Spatial Proximity, Topology

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 [1], 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 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, but not as much as we were expecting. To identify the reasons for this, we analyzed data collected during previous experimental studies with users.

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 spatial proximity into our topology graph. In Section 5, we describe the prototype and the experimental evaluation, comparing the solutions with and without proximity. Section 6, presents the results from analyzing sketches from experimental tests with users and finally, in section 7 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 [2]. 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.

Berchtold and Kriegel developed the S3 system [3] to support managing and retrieving industrial CAD parts. Drawings are described by their geometry (2D contour) and thematic attributes. The S3 system relies exclusively on
matching contours, ignoring spatial relations and shape information, making this method unsuitable for retrieving complex multi-shape drawings.

Another approach for retrieving engineering CAD drawings was developed by Müller and Rigoll, based on stochastic models [4]. Drawing databases are searched using sketches or shapes which represent details in mechanical parts. Their approach aims to retrieve images containing certain specified details and locating these details in the retrieved images. Authors represent drawings and queries using a pseudo 2-D Hidden Markov Model augmented with filler states. Their method only supports simple queries, representing a single element. More complex queries including several elements with spatial relationships between them are not contemplated.

A system that supports the retrieval of complex 2D drawings was presented in 1999 by Park and Um [5]. Their approach is based on dominant shapes, where objects are described by recursively decomposing its shape into a dominant shape, auxiliary components and their spatial relationships (inclusion and adjacency). These and the decomposition of blocks are stored in a complex graph structure. Visual elements in drawings are classified using a set of geometric primitives (triangles, circles, etc.). However, this small set of base geometric primitives and the not-so-efficient matching algorithm, based on the breadth-first tree matching, make it hard to handle large databases of drawings.

In the work of Beretti and Del Bimbo [6], authors describe shapes from a drawing by decomposing them 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 [7] for general unstructured free-form hand-drawings stored in the form of 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 also considering spatial relationships between strokes [8]. 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.

Paulson and Hammond [9] proposed a system that uses sketch symbols
as a mean of querying a database of existing media albums. Users associate multi-stroke sketches to each album, during creation, which are then compared to sketched queries, using a dual-classifier recognition algorithm that uses global features to describe an entire sketch. Although, this approach works well while comparing sketches with sketches, it is not suitable to compare simple sketches with complex drawings, like clip art drawings.

Another approach for matching hand-drawn sketches is the line-based representation of sketches proposed by Namboodiri and Jain [10]. To skirt around the problem of identifying basic shapes from a sketch, drawings are represented as a set of straight lines. While the algorithm is simple to implement it presents scalability problems to more complicated drawings and larger datasets, since it entails sequential search and one-to-one comparisons. Moreover, the conversion into straight lines is very dependent of the way users draw sketches.

Mascio et al. [11] presented an approach for sketch-based retrieval of drawings using a CBIR engine. Although they retrieve vector drawings, authors first convert them into raster images and then apply a set of CBIR techniques. Although authors do not present any evaluation of their solution, we think that converting vector drawings to raster images is not the best approach, since it consumes time and causes the loss of information during the process.

Liang et al. [12] developed a solution for drawing retrieval based on our prior solution [1]. Authors included some differences, such as the use of eight topological relationships 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, and reducing recall.

Pu and Ramani, developed an approach that analyzes drawings as a whole [13]. Authors proposed two methods to describe drawings. 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, insensible 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 [14] 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), as illustrated in Figure 2, 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 ele-
ments, to get more precise results.

In this paper, we present and evaluate a new approach to describe the
topological arrangement of visual elements in a drawing, which takes into
account the spatial proximity. This approach tries to enrich the topological
description of drawings, making the retrieval process more precise.

3. Overview of the System

The new algorithm developed to code spatial proximity between items in
a drawing was integrated in our general framework for sketch-based retrieval
of drawings, developed previously [1]. 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 3.
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.

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 [15]. We simplified the eight topological relationships defined by Egenhofer (**Disjoint**, **Meet**, **Overlap**, **Contain**, **Inside**, **Cover**, **Covered-By** and **Equal**), starting from his neighborhood graph for topological relationships. By reducing the number of topological relationships, we make our approach less restrictive and the topology graph simpler. Relationships thus extracted are compiled in a **Topology Graph**, where “parent” edges mean Inclusion and “sibling” connections mean Adjacency, as illustrated in Figure 4. While these relationships are weakly discriminating, they do not change with rotation and translation.

![Figure 4: Drawing and Topology graph.](image)

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 [16]. In this way, we reduce the problem of isomorphism between topology graphs to the computation of 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 and then computing descriptors at each level [1]. 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 [17]. 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 use of sketches to specify queries is supported by studies performed by Gross and Do [18, 19, 20], which show that designers use a small set of common graphical elements to describe the same drawings.

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.

Since we need to index a large quantity of descriptors from topology and geometry, we included at the core of our approach an efficient multidimensional indexing structure for storing these descriptors. This indexing structure, the NB-Tree [21], is a simple, yet efficient indexing structure, which uses dimension reduction. It works by mapping multidimensional points (topology and geometry descriptors) 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. Spatial 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 4. 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 5). 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 [16], 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. In [22] authors argue that these collisions occur rather infrequently, a claim seemingly verified by our experiments. Indeed, from experiences performed with 100,000 randomly generated graphs versus a set of 10 candidate
similar graphs, we measured precision and recall values, which show that it 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. This observation is supported by Duncan and Humphreys [23], which state that our human visual field is divided in several structural units that share some common property, such as, spatial proximity, hue, shape and motion. Although, color and movement are not relevant for the retrieval of drawings, spatial proximity and shape are taken into account when users draw sketches to specify queries.

![Figure 6: Using the adjacency weight to differentiate between far and near objects.](image)

To that end we devised a new mechanism to include spatial proximity into our topology graph (see Figure 6). 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.

To code proximity in the topology graph, we associate weights to the adjacency links of the graph (see Figure 7). 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 [24].

5. Experimental Evaluation

We developed a search engine prototype for vector drawings, using our sketch-based retrieval framework and the new mechanism to describe spatial 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 spatial proximity.

Figure 8 shows the sketch of a query, while Figure 9 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
5.2. Experimental Results

To evaluate our new approach of coding spatial 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 OpenClipart library\(^1\), 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

\(^1\)www.openclipart.org
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 (query). 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 spatial proximity, we also evaluated the order in which we perform the matching steps. Typically, our framework performs first a comparison 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: topology plus geometry (TG); topology with proximity plus geometry (TpG); geometry plus topology (GT); and geometry plus topology with proximity (GTP). 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. We determined the precision value at each point by varying the recall value. To calculate the effectiveness for each configuration, we computed the average precision from the three queries (see Table 1).

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 spatial proximity notion in the topology graph we can improve precision in both configurations (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. Figure 10 illustrates the precision & recall values for the latter case.

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,
Table 1: Precision of the four configurations.

<table>
<thead>
<tr>
<th>Recall</th>
<th>TG</th>
<th>TpG</th>
<th>GT</th>
<th>GTp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.123</td>
<td>0.130</td>
<td>0.096</td>
<td>0.099</td>
</tr>
<tr>
<td>0.1</td>
<td>0.123</td>
<td>0.130</td>
<td>0.096</td>
<td>0.099</td>
</tr>
<tr>
<td>0.2</td>
<td>0.123</td>
<td>0.130</td>
<td>0.096</td>
<td>0.099</td>
</tr>
<tr>
<td>0.3</td>
<td>0.109</td>
<td>0.116</td>
<td>0.082</td>
<td>0.084</td>
</tr>
<tr>
<td>0.4</td>
<td>0.109</td>
<td>0.116</td>
<td>0.082</td>
<td>0.084</td>
</tr>
<tr>
<td>0.5</td>
<td>0.039</td>
<td>0.057</td>
<td>0.043</td>
<td>0.045</td>
</tr>
<tr>
<td>0.6</td>
<td>0.039</td>
<td>0.057</td>
<td>0.043</td>
<td>0.045</td>
</tr>
<tr>
<td>0.7</td>
<td>0.000</td>
<td>0.004</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>0.8</td>
<td>0.000</td>
<td>0.004</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>0.9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

there is not much that the adjacency weights can do.

On the other hand, the increase of 1% in precision, due to the use of spatial proximity to describe drawings content, although relevant, was below our expectations. To understand this, we analyzed data collected in past tests with users and present the results in the next section.

Figure 10: Precision vs Recall for the topology filtering configuration, without and with spatial proximity.
6. Sketch Analysis

As we have seen in the previous section, not only the improvement of precision due to the inclusion of the spatial proximity to describe drawings content, was very small, but also the overall precision was smaller than in a previous system to retrieve technical drawings [1]. To perceive this we decided to study and analyze data collected from previous sessions of tests with users, dedicating special attention to the sketches they performed as queries. We divided the analysis in two parts, one for tests with the system for technical drawing retrieval and another for tests with clip art drawing retrieval systems. Our goal was to verify if there was any sketching difference in these two types of vector drawings.

6.1. Technical Drawings

In the context of technical drawings, we analyzed results from two test sessions. One where we asked users to draw sketches of a drawing, and a second where users performed sketches to be used as queries to the retrieval system.

The first data analyzed were collected during a drawing experiment with three draftspersons from the mould industry. This experiment occurred with the intent of understanding how users sketch 2D views of mechanical parts.

Figure 11: Technical drawing used as example for users to sketch.
We divided the experiment in two parts. First we asked them to draw (using a Tablet PC) a technical drawing shown in a A0 sheet of paper (see Figure 11), without any constraints or suggestions on how they should perform their sketches. Second, we asked users to draw a quicker and simplified version of the same technical drawing, representing only the main features they perceive from each drawing.

In the first part, users were very concerned about details and the accuracy of the resulting sketch (see Figure 12). We noticed that users always started their sketches by drawing the contour of the part, and then they added some details carefully, paying special attention to accuracy.

At the beginning of the second sketching session we told users that the main point was to make a quick sketch, without many details, representing only the main features from each drawing. As a result, sketches produced were much simpler than those produced before (see Figure 13), and the time spent on each drawing decreased significantly.
From the two sketching sessions of the experiment we noticed that users tend to sketch a lot of details and to use too much accuracy, mainly because they were trying to reproduce the overall drawing. After we tell them that the goal was not to reproduce the complete drawing, but to sketch what they think are the most relevant elements that could be used to identify or distinguish the drawing, users started drawing less complex sketches. We also observed that all users tend to identify the same relevant shapes from a drawing and that there is some coherence among sketches performed by all the users, validating our approach of using sketches as the main query mechanism.

The second data analyzed were collected during an experimental evaluation of our system to retrieve technical drawings, using the approach described in Section 3. Tests were done with six draftspeople from the mould industry. The goal of the experiment was to evaluate our system for sketch-based retrieval of technical drawings. During the experimental session we presented a printed drawing (see Figure 14 left) to users and asked them to sketch a query for that drawing, submit it to the system and analyze the returned results. In the present case, only the sketched query is relevant for our analysis. So, we examine the different sketches performed for each query, during the experimental session. As we can see from Figure 14 users drew simple sketches with not much detail, but with the relevant features of the drawing.

6.2. Clip art Drawings

Results related to clip art drawings were collected from two test sessions with users, one to evaluate a previous version of a clip art retrieval system, BajaVista [25] and another from the recent system that includes spatial prox-
imity, Indagare [26, 27]. The first session involved twelve users and the other was carried out with ten users. None of the 22 users had previous experience with tablet devices or any other pen-based system.

In both sessions, we asked users to search elements, by giving them an oral description, e.g. “search for a car”, “search for an house”, etc. Figure 15 shows some of the sketches performed by users in both sessions, revealing their simplicity and the small amount of visual elements included.

6.3. Discussion

As we can see from the analysis of drawings from both application domains, technical drawings and clip art drawings, sketches produced by users are different. While in the case of technical drawings users draw more complete sketches with several visual elements, and consequently defining a richer topological configuration; for clip art drawings, users produce simpler sketches, with fewer elements and with a poorer topological description. Figures 16 and 17 illustrates the difference of the resulting topology graph for the two domains. As we can see, the graph for the technical drawing is richer than the graph for clip art sketches. Indeed, during tests with users, we observed that they typically draw a very small number of shapes, and consequently do not specify topology, but only geometry. Moreover, users put a lot of effort in trying to sketch correctly what they are searching for.

Another reason for the difference between the two types of sketches could be in the experiment itself. While in the case of retrieving technical drawings, users were presented with a drawing on paper to search for a similar one, for clip art drawings we did not present any sample, giving just an oral description of the element to find.

To confirm these observations, we plan to perform two new sets of tests with users. First, we will use our new approach with spatial proximity in our system for retrieving technical drawings, to measure precision improvement. Second, we will perform tests using the Indagare system, but this time giving
Figure 16: Topology graph for a sketch of a technical drawing.

Figure 17: Topology graph for a sketch of a clip art drawing.

clip art drawings as examples, to see if users draw more complete sketches of the element to find.

7. 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 spatial 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 spatial proximity, the improvement achieved was small (only 1%). After analyzing data from previous sessions of user tests, we concluded that clip art drawings, contrarily to technical drawings, are more geometric than topological.

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, which takes into account the drawing as a whole and each of the shapes in it. 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


