RELATIONAL SKELETONS FOR RETRIEVAL IN PATENT DRAWINGS

Benoit Huet, Nicholas J. Kern, Gennarino Guarascio and Bernard Merialdo

Institut Eurecom
2229 Route des cretes,
06904 Sophia-Antipolis, France.
Benoit.Huet@eurecom.fr

ABSTRACT

This paper presents the evaluation of a number of algorithm alternatives for content based retrieval from a database of technical drawings representing patents. The objective is to help patent evaluators in their quest for a possible patent bearing too much similarity with the one under investigation. To achieve this, we have devised a system where images (drawings) are represented using attributed graphs based on the extracted line-patterns or histograms of attributes computed from the graphs. Retrieval is either performed using histogram comparison [6] or thanks to a graph similarity measure [9]. Promising results are presented along with possible work extention.

1. INTRODUCTION

The topic of recognition of objects from large libraries of images has attracted much interest over the past decade [1, 2, 3, 4]. Most of the literature has focussed on using low-level image characteristics such as color [1], texture [5] or local feature orientation [3] for the purposes of recognition. One of the most efficient ways to realize recognition is to encode the distribution of image characteristics in a histogram [1, 6]. Recognition is achieved by comparing the histogram for the query and those for the images residing in the library.

There has been very limited work aimed towards the use of content-based retrieval techniques for solving real life problems. Indeed, most of the papers in the field to date are concerned with indexing home photo albums. The problem of finding relevant technical drawing has caused limited interest to the research community [7, 8]. An area where the visual search of line drawings is critical is the verification of the originality of an invention. Most patents are described in detail to the patent office with the help of technical or engineering drawings. It would be a great asset to such an organization to be able to query the patent database on the basis of visual cues.

This work was funded by the European Patent Office (EPO) under number CAFS4126/00/00025.

The study presented in this paper is aimed at evaluating the feasibility of such a task based on the methods for graph– and histogram–based image retrieval described in the work of Huet and Hancock [9, 6]. The goal is thus to find similar *patents*, on the basis of similar *images*.

We will start with an overall presentation of the system. Section 3 explain our choice of line extraction algorithm. We then describe the data-base's construction along with the retrieval algorithms (Section 4). Section 5 comments the results obtained with our method. We then conclude and propose future work.

2. OVERVIEW OF THE SYSTEM

The system for content based image retrieval is basically divided into four parts, as described below:

1. **Pre-Processing (optional)**: Here, images may be splitted in a number of smaller ones such that each image contains a single object prior to further processing. This stage may be omitted.

2. **Line Extraction**: This stage extracts the lines from the, eventually pre-processed, bitmap image. This is in itself a multi-stage process. We evaluated three line extraction schemes: the Canny edge detector, a thinning algorithm and Voronoi–Skeletons implementation. Once linear features have been extracted a polygonization procedure is performed to obtain the line segments.

3. **Data-base construction**: This stage creates the database on which retrieval is to be performed. The two retrieval algorithm operate on different representation of the images:
   Graph Construction: The set of lines is transformed in a 6-nearest-neighbor-graph and Euclidean invariant attributes are computed on the edges of the graph.

Histogram Construction: The graph attributes are entered into a two dimensional histograms for rapid image comparison.

4. **Data-base Querying**: The comparison between the images takes place. This is performed using one of the following methods:
   Graph based Retrieval: The query and image graphs are compared using a variant of the Hausdorff distance [9]. This allows to retrieve the most similar images from the set of graphs.
   Histogram based Retrieval: The relational histogram of the query image is compared with all those from the data-base using the Bhattacharyya correlation measure [6].

Having briefly presented the different processing steps of the system we will now provided more detailed information.

3. LINE EXTRACTION

The algorithms for indexing the data-base are based on line-segment pair attributes. The bitmap-images have to be converted to a line representation. Among the major problems faced here, the fact that a "line" in the drawings can have arbitrary thickness, is the most important.

Ideally, we would like to select the lines that represent "important" properties of the image only. Unfortunately, the definition of "important" is highly non–trivial. Additionally there is no unique line representation of an image. For these two reasons *all* lines are taken into account. Thus, the retrieval algorithms have to cope with a (potentially large) number of "noisy" lines, that don't represent important image features.

Initially, we experimented with standard edge detection algorithms like the canny edge detector (see [10]). However it is clear that edge detectors are not well suited for this type of image as they detect two edges per lines. The use of a line operator is a possible solution to this problem [11]. However, a different mask is required for various line width which is not very satisfactory. Thus, we have decided to use a skeletonization algorithm to normalize the line representations in our images. The skeleton (or medial axis) of an object is usually defined using grass fire analogy: when a grass fire is lit on all sides of the object simultaneously, the skeleton is represented by the lines where two fires meet and extinguish themselves.

We experimented with both thinning algorithms and voronoi skeletons as described in [12]. The main problem of thinning algorithms is, that they do not guarantee that in all cases the result will be the medial axis of the shape under consideration. For this reason, we opted for the use of voronoi-skeleton and used the program developed by Ogniewicz [12].

The polygonization transforms the lists of connected pixels that constitute the skeletons into line-segments. The method consist in fitting a line between the first and the last point of the pixel-list. Compute the maximal perpendicular distance of between curve and line points. If the maximal distance is sufficiently large and the resulting two sub-lines are sufficiently long, the line is split. The algorithm is called recursively for each of the sub–lines created.

4. DATA-BASE CONSTRUCTION AND RETRIEVAL

We are interested in representing images using line-segments. Raw information about lines (eg. length, orientation) are sensitive to transformations of the images, like rotations, translations, scaling. In order to make the representation invariant, line attributes are computed from pairwise relations between lines. Our vector of attributes is composed of

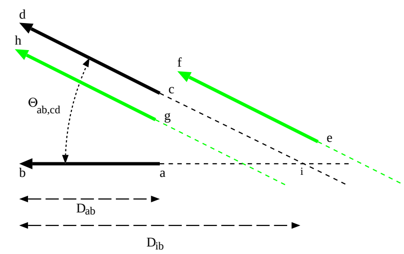

Fig. 1. Definition of the pairwise attributes

two parts: the relative angle and the relative position of the two lines. The relative angle is computed as follows

$$\Theta_{ab,cd} = \arccos\left(\frac{x_{ab} \cdot x_{cd}}{|x_{ab}||x_{cd}|}\right)$$

where x_{ab} indicates the vector line from a to b.

The directed relative position $\vartheta_{x_{ab},x_{cd}}$ is defined as the normalized length ratio between the oriented baseline vector x_{ab} and the vector x_{ib} joining the end (b) of the baseline segment (ab) to the intersection of the segment pair (cd).

$$\vartheta_{x_{ab},x_{cd}} = 1/\left(\frac{1}{2} + \frac{D_{ib}}{D_{ab}}\right)$$

In an attempt to reduce the number of attributes representing an image and to increase the influence of local visual information a n-nearest neighbor graph is constructed from the line-segment set [9]. Nodes in the graph represent lines and edges the n closest lines (The distance between two lines is the Euclidean distance of the line centers). The attributes related to a pair of line-segments are kept on the directed edge which connects them such that every edge is attributed with a vector $v^G_{(i,j)} = (\Theta_{(i,j)}, \vartheta_{(i,j)})$ of pairwise attributes. We can thus represent our image by a 3–tuple

$G = (V, E, A)$, $E \subseteq V \times V$ where V indicates the set of nodes; E represents the edges of the graph. Finally, the set $A = \left\{ v^G_{(i,j)} | v^G_{(i,j)} = (\Theta_{(i,j)}, \vartheta_{(i,j)}), (i,j) \in E \right\}$ denotes the set of pairwise attribute–vectors attached to the edges of the graph. For the discussion below, we introduce also $C_i^Q = \{ j | (i,j) \in E^Q \}$ which denotes all the edges departing from a given node i in a graph Q. In the following discussion we will assume to match a "query–graph" $G^Q = (V^Q, E^Q, A^Q)$ against a graph from the database $G^D = (V^D, E^D, A^D)$.

4.1. Histogram–based Retrieval

The idea underpinning the histogram-based selection is to construct an histogram from the relative attributes. Although it is possible to include the attributes of all possible edges $V \times V$, we only use the ones from the NN–graph described above. These attributes describe the *local* environment of each line. The assumption thereby is, that the local attributes are more robust to occlusion, missing or extra features and noise. The resulting histogram is called *structurally gated* [6]. Since the attribute vectors are of dimension two, 2D–histograms are employed, with one dimension for the relative angle and one for the relative position. The normalized histogram may be written as follows:

$$h^G(i,j) = \frac{H(i,j)}{\sum_{I=1}^{n_\alpha} \sum_{J=1}^{n_\lambda} H(I,J)}$$

We used $n_\alpha = 36$ with a uniform distribution in $[-\pi, \pi]$ for relative angles and for the relative positions, we used $n_\lambda = 12$. The distance between two histograms is given by the Bhattacharyya distance:

$$B(h^Q, h^D) = -\ln \sum_{I=1}^{n_\alpha} \sum_{J=1}^{n_\lambda} \sqrt{h^Q(I,J) \times h^D(I,J)}$$

This measure does not take into account zero histogram-entries. For highly structured histograms (ie. those, that are not uniformly populated), this can lead to the selection of matches in which there is a strong congruence between the structure of query and data–base histogram.

4.2. Graph–based Retrieval

The task here is to measure the "distance" between two graphs. The measure we describe here, is derived from the directed Hausdorff distance. The Hausdorff distance aims at computing the distance between two sets of unordered observations, where the correspondences between the individual items are not known. For a set of unitary attributes it can be formulated as

$$H_U(G^Q, G^D) = \max_{i \in V_Q} \min_{I \in V_D} ||u_I^D - u_i^Q||$$

where we have attributed every node with a set $u^D_\cdot$ of attributes. The complete description of the modification of the hausdorff distance can be found in [9]. The basic idea is to use the Hausdorff distance twice. Once for comparing the attribute vectors and once for comparing complete nodes. Thus, the modified Hausdorff distance becomes

$$H^M(G^Q, G^D) = \max_{i \in V^Q} \min_{I \in V^D} \max_{j \in C_i^Q} \min_{J \in C_I^D} ||v^Q_{(I,J)} - v^D_{(i,j)}||$$

The Hausdorff distance is very sensitive to noise. The idea to overcome this issue is to make the hausdorff distance "softer" by replacing the $\max$ test by weighted sums. The final distance can thus be formulated as

$$\begin{aligned} H(G^Q, G^D) \quad = \quad & \frac{1}{||V^Q||} \sum_{i \in V^Q} \max_{I \in V^D} \\ & \frac{1}{||C_i^Q||} \sum_{j \in C_i^Q} \max_{J \in C_I^D} \Gamma_\sigma(||v^Q_{(I,J)} - v^D_{(i,j)}||) \end{aligned}$$

To select from a complete DB of images, it is sufficient to compare the query against each image. The weighting function Γ_σ used in our experiments is of the following Gaussian form:

$$\Gamma_\sigma(\rho) = \exp\left(-\frac{\rho^2}{\sigma}\right)$$

which explains the change of the $\min$ into $\max$ in the final formula.

5. EXPERIMENTS

The drawing data-base was supplied by the EPO and was accompanied by a set of correspondences between the files (ie. similar *patents* - the drawings aren't necessarily similar). Thanks to this ground truth the performance could be evaluated. However, the patent evaluator judge similarity based on more that just direct visual properties, which make a high level of recall accuracy difficult to obtain for a recognition algorithm.

In figure 5 we present a query example, which demonstrates that the algorithm is able to find complex objects. Two of the corresponding images are ranked very high (2 and 4) and both have some apparent similarity. The last similar image is retrieved in 20th position. Unfortunately, not all queries work as well as the above.

In order to compare the different retrieval approaches we performed a number of experiments and measured accuracy based on the relevance of the images returned. The occurrences of similar images were counted in a window of the 30 best results. The *recall* is then computed as the ratio of the number of similar images found within the window by the total number of similar images. In case of cut images, each sub-images was taken as a query and compared

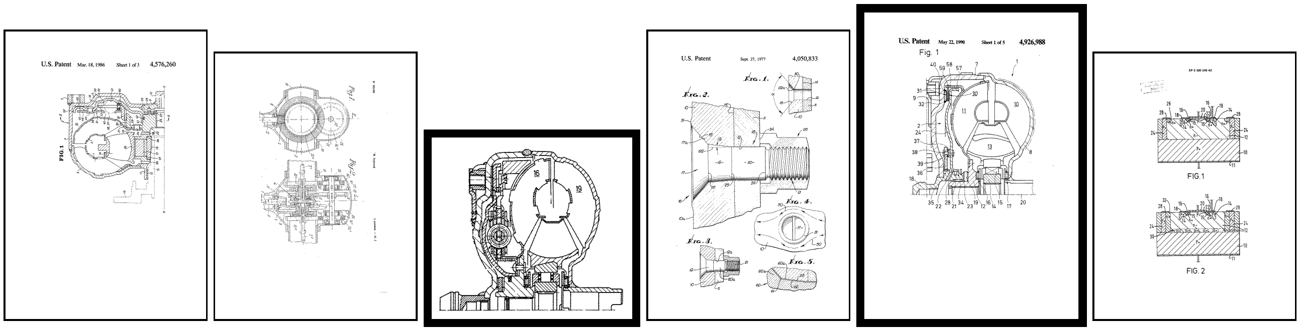

Fig. 2. Retrieval Result of 1 using Graph-based Selection
The images are presented in a left–to–right with left-most one used as query and similar images highlighted.

Retrieval Method	Image Pre-Processing	Result
Histogram	Normal	16.67
Histogram	Cut	36.46
Graph	Normal	**47.25**
Graph	Cut	**52.16**

Table 1. Results of Retrieval

against the complete database. Then the recall for all the sub–images was computed and the maximum taken.

Table 1 shows the average results (over 57 queries) of our experiments on a database of 180 patents. It is clear that the best performance are obtained using the graph similarity measure derived from the hausdorff distance.

Dividing the images in sub-images (cut) greatly increases the recall rate in the case of histogram representation but has minor effect on the graph similarity measure. This is to be expected because of the global nature of histograms. However, recall performances are in all case better using graph rather than histogram comparison. The better performance of the graph based technique comes at a cost. The search takes a considerable amount of time which is directly dependent on the size and complexity of the images to be compared.

5.1. Conclusions

The goal of this work was to investigate methods to efficiently select patents from a data-base on the basis of the visual content of technical drawings.

After carefully selecting, among several candidates, the most appropriate line extraction algorithm, two different retrieval algorithms were evaluated, one based on a relational graph similarity measure [9] and the other on structural histogram comparison [6]. Thus a serie of experiments was conducted. The results show, that the methods work reasonably well for some images and poorly in other situation.

We find the results rather encouraging. In an attempt to improve the performance we will study the possibility of enhancing the representation with other basic features than line-segment as well as research methods for organizing a library of attributed graphs for efficient retrieval.

6. REFERENCES

[1] M. J. Swain and D. H. Ballard, "Indexing via colour histograms," *ICCV'90*, pp. 390–393, 1990.

[2] R. W. Picard, "Light-years from Lena: Video and image libraries of the future," *ICIP'95*, vol. 1, pp. 310–313, 1995.

[3] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognition*, vol. 29, no. 8, pp. 1233–1244, 1996.

[4] A. del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *PAMI*, vol. 19, no. 2, pp. 121–132, February 1997.

[5] G. L. Gimelfarb and A. K. Jain, "On retrieving textured images from an image database," *Pattern Recognition*, vol. 29, no. 9, pp. 1461–1483, 1996.

[6] B. Huet and E.R. Hancock, "Line pattern retrieval using relational histograms," *PAMI*, vol. 21, no. 12, pp. 1363–1370, December 1999.

[7] J. P. Eakins, K. Shields, and J. M. Boardman, "ARTISAN - a shape retrieval system based on boundary family indexing," *Storage and Retrieval for Image and Video Databases IV, Proc SPIE 2670, San Jose*, pp. 17–28, February 1996.

[8] T. Syeda-Mahmood, "Indexing of technical line drawing databases," *PAMI*, vol. 21, no. 8, pp. 737–751, August 1999.

[9] B. Huet and E. R. Hancock, "Fuzzy relational distance for large-scale object recognition," *CVPR*, pp. 138–143, June 1998.

[10] J. Canny, "A computational approach to edge detection," *PAMI*, vol. 8, no. 6, pp. 679–698, 1986.

[11] J. Canny, "Finding edges and lines in images," in *MIT AI TR*, 1983.

[12] R. L. Ogniewicz and O. Kubler, "Hierarchic Voronoi skeletons," *Pattern Recognition*, vol. 28, no. 3, pp. 343–359, 1995.