EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel and Z. Pan (Editors)

Mixing Images and Sketches for Retrieving Vector Drawings

Alfredo Ferreira[1], Manuel J.Fonseca[1], Joaquim A.Jorge[1] and Mário Ramalho[2]

[1] INESC-ID/IST/Technical University of Lisbon, R. Alves Redol, 9, 1000-029 Lisboa, Portugal
[2] DEM/IST/Technical University of Lisbon, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Abstract

Current approaches to content-based retrieval of multimedia data usually rely either on query by example or on sketches of the desired image, but not on both. In this paper, we propose a new query specification scheme, where digital images are combined with sketches, after vectorization, taking advantage of both methods. We selected a set of algorithms to perform image vectorization, taking into account the trade-off between vector image quality and processing time. This method of specifying queries is part of a system to retrieve vector drawings, which we briefly describe in this paper.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Storage and Retrieval]: Retrieval Models I.4.6 [Image Processing and Computer Vision]: Edge and Feature Detection

1. Introduction

During the past two decades the widespread use of CAD applications produced a large number of engineering drawings pertaining to domains such as architectural or industrial design. However, current CAD systems provides only conventional database queries or direct-manipulation mechanisms to retrieve them.

To retrieve past drawings designers and engineers usually depend on queries based on textual annotations or drawing metadata, forcing them to recall part numbers, dates, locations or client and designer names, as presented in [Bak90, CW91]. In some cases the only way to find a drawing is browsing through large and deep file directories or navigate a complex maze of menus and dialogs for component libraries.

Despite the considerable work on content-based retrieval of multimedia information, most focused on image databases, as surveyed by Shi-Kuo Chang [CPR99]. These systems tend to use color, contours and texture as main features for content description, which is unsuitable for describing vector drawings.

Recently, several approaches to retrieve vector drawings by content had been presented. Gross' Electronic Cocktail Napkin [GD96] addressed a visual retrieval scheme based on diagrams, to indexing databases of architectural drawings. Berchtold's S3 system [BK97] supports managing and retrieving industrial CAD parts, through contour matching. Park's approach [PU99] retrieves mechanical parts based on the dominant shape and spatial relationships. Leung and Chen proposed a sketch retrieval method [LC03] for general unstructured free-form hand-drawings. Funkhouser et. al. [FMK*03] describe a method for retrieving 3D shapes using sketched countours. However, these methods use exclusively sketches to specify queries or pictures to query by example.

Mixing images with sketches provides us with a powerful new way to define queries for content-based retrieval of vector drawings. We take advantage of existing hard-copies of drawings and simultaneously explore users' visual memory and ability at sketching. With our approach users can add new elements to vectorized drawings by sketching new shapes or delete existing entities using a simple gesture command. This way, they can start with a digitized drawing and then apply editing commands to refine it.

Images and sketches are integrated by first applying a vectorization process, converting raster drawings into vector figures. To that end we selected a set of existing algorithms from computer vision. To achieve a vectorization solution which is usable interactively, we privileged performance in detriment of accuracy. Indeed, we tuned the selected algorithms to minimize user intervention during vectorization, assuring an almost automatic process. Then users can edit the vectorized image to refine queries, to select rel-

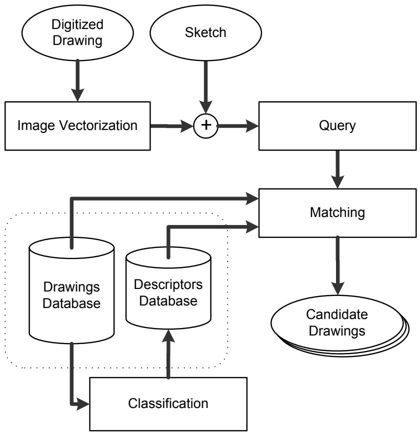

Figure 1: *System Architecture.*

evant parts, to delete unnecessary shapes or to add new visual elements. In this paper we present the integration of this new method of constructing queries into our content-based retrieval system for vector drawings [FJJ04].

In the remainder of this paper, we first provide an overview of the overall retrieval system architecture where this method of mixing images and drawings will be integrated. Section 3 describes the vectorization process. The next section explains how we combine sketches and digital images to specify queries and how we identify features extracted from queries. We conclude the paper by presenting some experimental results and future work.

2. Architecture Overview

Our scheme for content-based retrieval of vector drawings through images and hand-sketched queries (see Figure 1) supplies a mechanism to retrieve vector drawings, in electronic format, taking advantage of users' natural ability at sketching and drawing.

This approach is composed by four different components. The classification component creates the descriptors from existing drawings. The image vectorization component is responsible for converting the digitized drawing to vector format. The query component extracts features from sketches and vectorized drawings, creating descriptors. Finally, the matching component compares descriptors stored in the database with the ones produced by the query component, yielding a set of candidate drawings.

2.1. Classification

Typically, two kinds of features are extracted from drawings. Visual features encode information, such as color, texture and shape. Relationship features describe topological and

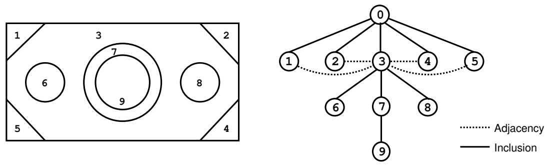

Figure 2: *Detected set of polygons (left) and correspondent topology graph (right).*

spatial relationships among objects in a picture. However, for vector drawings, color and texture are irrelevant. Thus, we focus on topology and geometry.

Technical drawings usually contains many details that are not relevant for a visual query and increases the cost of searching. Therefore, our classification process starts by applying a simplification step to eliminate most useless elements, speeding up queries. During this simplification step we remove visual details (i.e. small-scale features) while retaining the perceptually dominant elements and shpes.

After simplification, we identify visual elements, namely polygons and lines, and extract shape and topological information from the resulting drawings. Relationships thus extracted are compiled in a topology graph, where "parent" edges mean `Inclusion` and "sibling" connections mean `Adjacency`, as illustrated in Figure 2.

However, topology graphs are not directly used for searching similar drawings, since graph matching is a NP-complete problem. We use the correspondent graph spectrum instead. For each topology graph to be indexed in a database we compute descriptors based on its spectrum [CRS97]. In this way, we reduce the problem of isomorphism between topology graphs to computing distances between their descriptors.

To support partial drawing matches, we also compute descrpitors 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 [FJJ04] and then computing descriptors at each level. This combination of sub-graphs descriptors and levels of detail provides a powerfull 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 [FJ01] to compute a set of geometric features, which enables us to use either drawing data or sketches as input. By using geometric features instead of polygon classification, we can index and store potentially unlimited families of shapes. Experimental evaluation [FBRJ04] revealed that this techique outperforms other methods to describe shapes, such as Fourier descriptors or Delaunay triangulation, yielding better precision figures for all recall values.

The geometry and topology descriptors thus computed are

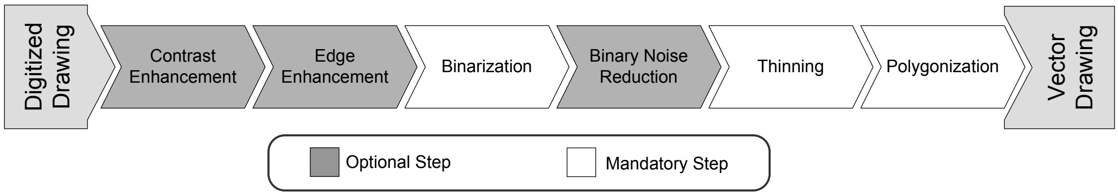

Figure 3: *Image Processing Steps.*

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. 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 and sketches) are processed by the same pipeline.

We developed a new multidimensional indexing structure, the NB-Tree [FJ03] , which provides an efficient indexing mechanism for high-dimensional data points. The NB-Tree is a simple, yet efficient indexing structure, which uses dimension reduction to speed-up search process. 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 make searching faster, we first select a set of drawings topologically similar to the query, by performing a KNN query to the topology indexing structure and then use geometric information to further refine the set of candidates.

A more detailed description of classification, query and matching components can be found in [Fon04].

3. Vectorization

When searching for a drawing or similar drawings based on a printed version, we first convert the hardcopy to digital format. The quality of such digital images may vary drastically depending of the characteristics of all parts of the sensor chain and illumination technology [DSW99].

The loss of quality incurred during image acquisition makes the vectorization (*i.e.* raster-to-vector conversion) a complex process. Until now no methods devised can be considered as sufficiently stable and robust to work as stand-alone "black boxes" [TASD*99]. According to Karl Tombre et al. [TASD*98], one important factor for robustness is to minimize the number of parameters and thresholds needed in the vectorization process. Thus, in our approach we intend to achieve an acceptable vectorization result (digitized query) with minimal configuration requirements. Therefore, our system requires reduced user intervation during vectorization process.

Our approach starts by performing contrast and edge enhancement, followed by a thresholding process that yields a binary version of the original image. Then, we apply a noise reduction algorithm followed by a thinning step before converting image to vector format.

In Figure 3 we depict the image processing pipeline, which is composed by a sequential application of the steps referred in the previous paragraph. Depending on image quality, contrast and edge enhancement might be skipped, as well as binary noise reduction.

In our work we assume that printed drawings have no text or symbols. However, this restriction can be eliminated by applying a text/symbols/graphics segmentation method before polygonization. Ramel and Vincent [RV03] presented several strategies for localization and recognition of graphical entities in line drawings. Additionally, we also assume that drawings are digitized to a raster gray-scale format and saved in non-compressed image files. If drawings are digitized in binary format, first steps can be discarded, at the cost of external pre or post processing in order to refine or connect lines after image acquisition.

3.1. Contrast and Edge Enhancement

Accurate visual interpretation of images depends to a great extent on their visual contrast. To improve the appearance of images and effectiveness of analysis we use histogram transformations. These well known methods enhance image contrast by changing the shape of the histogram and are extensively documented [Hum75, PAA*87, Pra91, SOS00].

Depending on the original image quality we might apply histogram expansion or, in lower quality images, histogram transformation by specifying the desired shape of the output histogram. In our approach we considered three possible slope values (m) for output histogram: a positive slope with value $m = 1$, that enhances contrast in low intensities; a negative slope with value $m = 1$, that enhances contrast in high intensities; or a zero slope ($m = 0$), corresponding to histogram equalization.

The main goal of the algorithms presented in this paper is to detect lines from original images in digital format. Therefore, it is important to enhance images in order to improve the effectiveness of line detection when their quality is not good enough. To accomplish this, we apply the edge enhancement algorithm suggested in [SOS00], which uses a high-pass filtering technique after performing a noise reduction step.

3.2. Binarization and Noise Reduction

Shading and textures are not relevant to the presented work, since we are only interested in performing line detection. Therefore gray-scales image must be converted to a binary image.

Binarization methods can be divided in two main categories: global techniques and locally adaptive techniques. Global techniques determine a single threshold level for the image and perform conversion using that value for all pixels, ignoring their context. On the other hand, locally adaptive techniques use context information to adapt the threshold for each pixel according to its neighbors.

Since in this paper we assume that images are acquired using a flatbed scanner that provides controlled lightning conditions, the best results are achieved, when the original paper drawing has good quality, by using global techniques. However poor quality originals may require locally adaptive techniques. Therefore we suggest that the choice of binarization method must be done during image processing, by selecting between a global technique based on histogram shape or a locally adaptive region averaging technique, both described in [SOS00].

In the current version of our system, the choice between global and locally adaptive techniques is made by the user during the vectorization process. This implies that users are able to identify the best technique by looking at the digitized image.

Unappropriate storage, mishandling and aging of paper drawings can seriously degrade the quality of the original document. Additionally, the acquisition process can introduce noise on the captured images. Noise removal on binary images improves greatly the reliability and robustness of thinning and line detection algorithms. However, complete noise removal is hard to achieve in real time applications. In our approach we aim to perform just an efficient noise reduction that executes in a short period of time. Therefore,

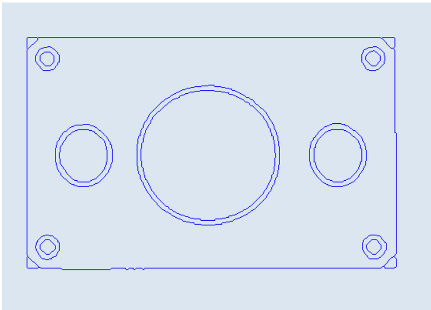

Figure 4: *Vector version of drawing.*

by considering characteristics of drawings, we use the noise reduction algorithm based on kFill filter [O'G92a], which enables control of corner rounding and length shortening effects.

3.3. Thinning

Since our purpose is to find a sketchable version of original drawings and we are concerned mainly with topology and geometry, the thickness of lines that composes the original drawing is not important. We are only interested in its one pixel wide skeleton. Lam et al made comprehensive reviews [LLS92, LS95] of several skeletonization methods and conclude that there are no general thinning algorithm.

We decided to use the thinning algorithm described in [O'G90], due to its simplicity and effectiveness in computing the skeleton of elongated segments, despite the lack of precision when determining line crossings and corners of thick lines, since we do not aim to achieve a precise version of the original image.

In our approach we consider that technical drawings do not contain filled shapes, only lines and arcs. However, the presented work can be extended to consider such elements by using a segmentation method to separate filled shapes from lines and then perform contour detection on the filled elements instead of thinning.

3.4. Polygonization

Since our classification mechanism is based on shape geometry and spacial relationships, some loss of detail during this process is acceptable and does not greatly affect the query results. This is crucial when deciding which vectorization method to use. Therefore, we use an adaptation of the primitives chain-coding method proposed by O'Gorman [O'G92b] followed by a polygonization procedure.

We adapt this well-known primitive chain-coding method in order to construct sets of linked points instead of codes, thus identifying branches and intersections. Each of these sets yields an approximation to a smooth curve, which can

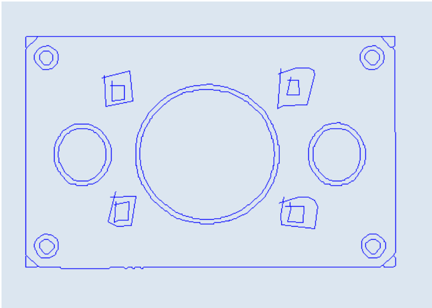

Figure 5: *Sketches combined with vectorized drawing.*

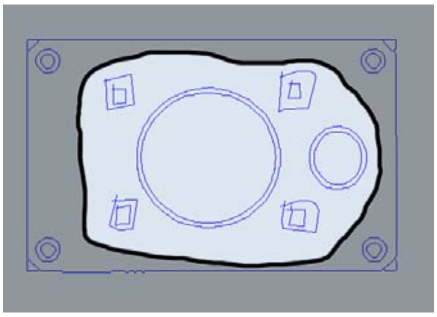

Figure 6: *Editing and selecting part of a drawing.*

be represented by a set of connected line segments. To compute this set we use a polygonization procedure based on the Douglas-Peucker algorithm [DP73], that reduces the number of points within a given threshold. Consequently, the presented method produces an approximate vector version of the original drawing composed by a set of line segments, as depicted in Figure 4.

4. Combining Images and Sketches

Our vectorization process produces an approximate vector version of raster images of drawings, where the main characteristics and features of drawings are preserved. Based on the yielded set of line segments we construct a query from which topological and geometrical features will be extracted, as described in Section 2.2.

After finishing vectorization, users can sketch on the resulting drawing to add, change or delete visual elements. Through the use of sketches it is possible to enrich the query in order to obtain better results or select just part of the sketch to use as query. Figure 5 illustrates the result of vectorizing an image of a drawing with some sketches added to convey more information.

Figure 6 depicts an example where the user edited the vectorized drawing, deleting some shapes and sketching others. In this case, a region was selected to be used as query, while the rest is ignored

Combining sketches with digitized drawings opens up the possibility to search for drawings similar to the original or to those that can be specified by adding or deleting entities.

5. Experimental Results

We developed a prototype which integrates all components described above. This prototype was used to conduct performance tests and a user evaluation. Figure 7 depicts the user interface of the prototype, which is divided in three main areas: a blue sketching area where users can sketch the queries, the results area where five results are displayed simultaneously and the buttons area which comprises control buttons.

5.1. Performance

Performance tests were executed in a Intel Pentium III 1GHz computer with 512MB RAM memory, running Windows XP. During these tests we concluded that processing an image with 823×1167 pixels scanned from an A4 paper using a resolution of 100dpi takes about sixty seconds to produce a vectorized version of the drawing.

Using now the query created by combining sketches with images (see Figure 5) to retrieve drawings from a database of 78 elements, we obtain the desired drawing in the top five reults, as illustrated in Figure 8. To produce these results it takes less than twenty seconds. These results fulfill our expectations for this version of the prototype since it is able to find two drawings identical to the query and another two that only differs in the location of some shapes. All other drawings in the database are less similar to the query than these four that are returned among the top five results.

5.2. Experimental Evaluation

The user evaluation tests involved three draftspeople and was divided in three parts. First, we explained the experiment and introduced the prototype to users. During the second

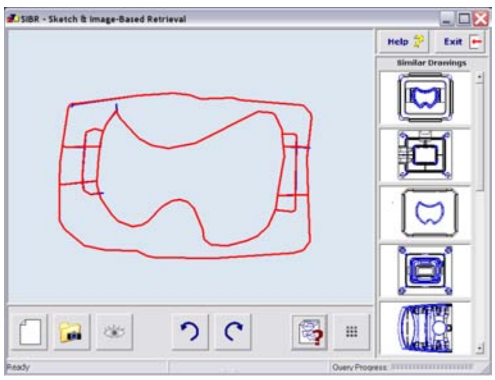

Figure 7: *Sketch-based retrieval prototype.*

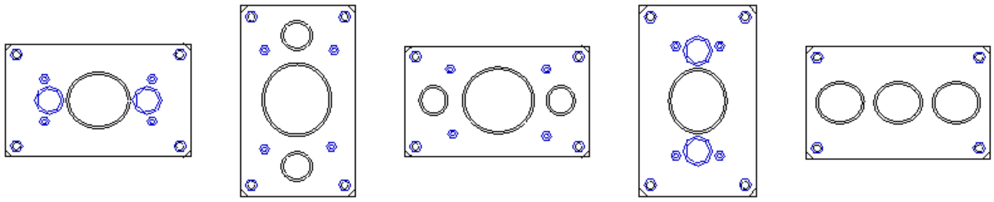

Figure 8: *Set of five candidate drawings.*

part users executed two sets of queries using the prototype, first sketching basic drawings and after querying for simple technical drawings. Finally, users answered a questionnaire, where we try to figure out their profile, their opinions about the prototype and their evaluation of the user interface. We also asked them, in an informal manner, about suggestions and ideas to improve the current version of the SBR prototype. This experiment involved a database with 78 drawings and 12 queries.

Notwithstanding the low number of users involved, preliminary results are very encouraging. In the majority of the queries, sought drawings were among the first five results and almost always within the first ten results, which gave some trust to users. One of the things that we observed during the execution of tasks was that users did not care about where in the order of retrieval the intended drawing appears, the important fact being that it was there. One of the users produced this comment "It [the system] found it [the drawing]! That is what counts!".

In summary, 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 in contrast to more traditional approaches. Moreover, users liked the new scheme of mixing sketches and digitized drawings, because they feel they can save time specifying queries.

6. Conclusions and Future Work

We have presented an approach to content based retrieval of vector drawings that allows combining digital images with sketches to formulate queries. Our approach allows users in a CAD environment to search for objects similar to existing paper drawings. Using the vectorization method presented in this paper and the classification scheme described in [FJJ04] we were able to execute successful queries based on sketch-edited scanned drawings within a reasonable time which allows for interactive usage.

Although results from our preliminary tests were very promising, we think that the number of users involved and the size of the drawing sets used did not come up to our expectations. We would like to had have performed tests with thousands of drawings and dozens of users, to state with more confidence that our approach has good performance and accuracy for very large sets of drawings. Even though we used a small database in our tests, our approach has the potential to deal with large sets of drawings. The indexing structure, that could prove the main bottle neck during retrieval, has shown good performance for data sets around one million elements, as described in [Fon04]. Therefore, we believe that our approach will present good performance for large sets of drawings.

From users' comments and suggestions, and from our observations, we are improving our prototype and algorithms in cooperation with actual users. We plan to test the new version of the prototype with a sizeable population and with a larger database of drawings, to obtain more grounded results and insights. Moreover, we intend to use other elements in classification and retrieval. Using textual information from drawings we would enrich our database with different data such as sizes or part references. Then users will be able to specify their queries not only trough digitized drawings and sketches but also employing textual information.

References

[Bak90] BAKERGEM D. V.: Image collections in the design studio. In *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Age* (1990), MIT Press, pp. 261–272.

[BK97] BERCHTOLD S., KRIEGEL H.-P.: S3: Similarity search in cad database systems. In *Proceedings of the International Conference on Management of Data (SIGMOD 1997)* (Tucson, Arizona, USA, 1997), ACM Press, pp. 564–567.

[CPR99] CHANG S.-K., PERRY B., ROSENFELD A.: *Content-Based Multimedia Information Access*. Kluwer Press, 1999.

[CRS97] CVETKOVIC D., ROWLINSON P., SIMIC S.: *Eigenspaces of Graphs*. Cambridge University Press, United Kingdom, 1997.

[CW91] CLAYTON M., WIESENTHAL H.: Enhancing the sketchbook. In *Proceedings of the Associa-*

tion for Computer Aided Design in Architecture (ACADIA 1991) (Los Angeles, CA, 1991).

[DP73] DOUGLAS D., PEUCKER T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer 10* (1973), 112–122.

[DSW99] DEMANT C., STREICHER B., WASZKEWITZ P.: *Industrial Image Processing : Visual Quality Control in Manufacturing*. Springer Verlag, 1999, ch. Chapter 8. "Overview:Image Acquisition and Ilumination".

[FBRJ04] 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.

[FJ01] FONSECA M. J., JORGE J. A.: Experimental Evaluation of an on-line Scribble Recognizer. *Pattern Recognition Letters 22*, 12 (2001), 1311–1319.

[FJ03] FONSECA M. J., JORGE J. A.: Indexing high-dimensional data for content-based retrieval in large databases. In *Proceedings of the 8th Int. Conference on Database Systems for Advanced Applications (DASFAA'03)* (Kyoto, Japan, Mar. 2003), IEEE Computer Society Press, pp. 267–274.

[FJJ04] FONSECA M. J., JR. A. F., JORGE J. A.: Content-Based Retrieval of Technical Drawings. *Special Issue of International Journal of Computer Applications in Technology (IJCAT) "Models and methods for representing and processing shape semantics" (to appear)* (2004).

[FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A search engine for 3d models. *ACM Trans. on Graphics 22*, 1 (January 2003).

[Fon04] FONSECA M. J.: *Sketch-Based Retrieval in Large Sets of Drawings*. PhD thesis, Instituto Superior Técnico / Universidade Técnica de Lisboa, Jul 2004.

[GD96] GROSS M., DO E.: Demonstrating the Electronic Cocktail Napkin: a paper-like interface for early design. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 1996)* (1996), pp. 5–6.

[Hum75] HUMMEL R. A.: Histogram modification techniques. *Computer Graphics Image Processing 4*, 3 (September 1975), 209–224.

[LC03] LEUNG W. H., CHEN T.: Hierarchical Matching for Retrieval of Hand-Drawn Sketches. In *Proc. of the IEEE ICME'03* (Baltimore, USA, July 2003), vol. 2, IEEE Press, pp. 29–32.

[LLS92] LAM L., LEE S., SUEN C.: Thinning methodologies-a comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence 14* (1992), 869–885.

[LS95] LAM L., SUEN C.: An evaluation of parallel thinning algorithms for character recognition. *IEEE TPAMI 17* (1995), 914–919.

[O'G90] O'GORMAN L.: K x k thinning. *Computer Vision Graphics and Image Processing (CVGIP) 51*, 2 (August" 1990), 195–215.

[O'G92a] O'GORMAN L.: Image and document processing techniques for the right pages electronic library system. In *International Conf. on Pattern Recognition (ICPR)* (1992), pp. 260–263.

[O'G92b] O'GORMAN L.: Primitives chain code. In *Progress in Computer Vision and Image Processing (CVIP92)* (1992), pp. 167–183.

[PAA*87] PIZER S. M., AMBURN E. P., AUSTIN J. D., CROMARTIE R., GESELOWITZ A., GREER T., ROMENY B. T. H., ZIMMERMAN J. B.: Adaptive histogram equalization and its variations. *Computer Vision, Graphics and Image Procesing 39*, 3 (1987), 355–368.

[Pra91] PRATT W. K.: *Digital Image Processing*. John Wiley & Sons Inc., 1991.

[PU99] 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.

[RV03] RAMEL J.-Y., VINCENT N.: Strategies for line drawing understanding. In *Proc. of the 5th IAPR Int. Workshop on Graphics Recognition* (Barcelona, Catalonia, Spain, 2003), IAPR.

[SOS00] SEUL M., O'GORMAN L., SAMMON M. J.: *Practical Algorithms for Image Analysis*. Cambridge University Press, 2000.

[TASD*98] TOMBRE K., AH-SOON C., DOSCH P., HABED A., MASINI G.: Stable, robust and off-the-shelf methods for graphics recognition. In *Proc. of the 14th Int. Conf. on Pattern Recognition* (Australia, 1998), IAPR, pp. 406–408.

[TASD*99] TOMBRE K., AH-SOON C., DOSCH P., MASINI G., TABBONE S.: Stable and robust vectorization: How to make the right choices. In *Proc. of the 3rd IAPR Intl. Workshop on Graphics Recognition* (Jaipur, India, 1999), IAPR, pp. 3–16.