ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval
(Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

Capturing Shape and Text for Retrieval

Oliver Lorenz and Gladys Monagan

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

Capturing Shape and Text for Retrieval

Oliver Lorenz and Gladys Monagan
Institut für Informationssysteme, ETH
Zürich, Switzerland

Abstract

We present a system for content-based retrieval of line drawings. Line drawings are schematic diagrams consisting of two kinds of information, text and graphics. The textual part is indexed from the results of word recognition. The indexing for the graphic content uses no *a-priori* knowledge about the content of a drawing and depends only on features derived from the shape of graphic structures. The indexing scheme is used in a vector based information retrieval system. Results and problems in evaluating such a system are discussed.

1 Introduction

Advances in computer hardware and networking capabilities have made images a widespread source of information. Large numbers of digital images are generated and used in applications such as identification and security, medical diagnosis, business data management, and art. An imminent problem to solve is how to store digital images so that they can be retrieved later. Visual browsing for reviewing a selection of images is only feasible for small amounts of data. Efficient and effective tools for retrieval are needed. Current approaches that offer means for retrieving images are called visual information management systems (VIMS) [1], sometimes also referred to as visual database systems [2, 3].

1.1 Content based image retrieval

Many applications require searching based on the content of an image. Traditionally, the description of the content of an image is given as a set of keywords or as a caption. Instead of using such an external description, automatic or semi-automatic image analysis can be applied to create a description based on the image structures. If the images in a collection pertain to a specific application (i.e a homogeneous collection of images), an image model may be created to guide the analysis process. This approach was chosen in semantic image databases [4, 5]. A query is presented to the system in a specialized query language. However, for heterogeneous collections of images, that is, collections of images of various domains where little is known about their content, this approach is not feasible.

1.2 Application independent features to capture shape

Without *a priori* knowledge about the content of an image, the description is limited to application-independent features. Such features describe the characteristics of the structures in the images and may be extracted automatically.

The features selected for the description of an image determine the kind of queries that can be answered. Features based on the histogram, texture and color distribution [6] cannot answer queries for images containing objects of a specific shape. Indexing shape is a difficult problem. The Art Museum system [7] finds paintings from a user's sketch based on edge-correlation between images. This approach does not allow for retrieval of individual objects with unknown position, size, or orientation. The QBIC system [6] uses features such as area, circularity, major axis orientation and moments to describe individual objects in the image.

As the images come from a heterogeneous collection of photographs, the feature extraction requires manual segmentation of the image content.

1.3 Contribution of our work

In our work, we propose the design of an information retrieval (IR) system for retrieving raster images based on the textual content and the shape of graphic structures. Both textual and *graphic features* are extracted automatically and they are independent of the application. This means that the features are chosen without regard as to whether the application is to retrieve maps, diagrams from manuals or drawings. In particular, we attack the problem of retrieving line drawings that we obtain as raster images. Line drawings are schematic diagrams that express an idea or sketch a part (charts, diagrams, electrical circuits). An example is given in Figure 2. One can think of line drawings as consisting of two layers, text and graphics, superimposed on each other. The text layer consists of textual entries from legends and annotations. Textual features are extracted using word recognition methods. Using the textual features, we can retrieve line drawings containing specific words. The graphic layer consists of *graphic primitives*, such as lines, curves, and filled polygons. The graphic features are derived by grouping some graphic primitives. The graphic features used, represent shape-information, and are chosen to be invariant under translation, rotation and scaling. They allow the retrieval of line drawings containing graphic structures similar to the structures contained in the query drawing. The result presented to the user is a list of line drawings from the collection, ranked by the similarity of the graphic and textual features they have in common with the query.

1.4 Structure of the text

In Section 2 we propose descriptions for the textual and graphic content of line drawings and how a vector based IR system can be implemented using graphic features. In particular, we will explain how the matching of similar graphical structures can be implemented without additional computation at query time. Section 3 shows how the approach is used in a practical example. We will show and discuss some results. In Section 4 we will discuss the problems we are facing in evaluating our IR system.

2 Using a vector based IR system for line drawings

An IR system is used to manage a collection, D , of documents, d_j . Queries are answered using an access structure that is based on features extracted from the documents of the collection. This access structure is called the *index* of the collection. The index is based on features, φ_i , from an indexing vocabulary Φ . A query is presented to the retrieval system as an example document, q , and indexed in the same way as the documents in the collection. IR systems do not rely on an exact matching between the query and the documents in the collection, but rather identify relevant documents based on their similarity to the query. The similarity of two documents is a function of the set of features they have in common. An IR system is the implementation of a retrieval method. A retrieval method consists of two parts: an indexing method to extract the indexing features, φ_i , from a document, d_j , and a retrieval function to compare each document to the query, q .

In a *vector based IR system* each document is described by a feature vector, $\vec{d}_j$. An automatic weighting scheme assigns a discrimination value to each feature in every document. The answer returned by the system is a list of documents from the collection ranked by the similarity of the document feature vectors, $\vec{d}_j$, to the feature vector, $\vec{q}$ of the query. A standard similarity measure is the *cosine-measure*.

Most IR systems have been developed for text documents, where the features are *terms*. In this section we will explain how terms can be substituted by different types of features for capturing the textual and graphic content of a line drawing. The difference between the text data extracted from a line drawing and terms in a text document is that the words extracted from the drawing are the result of a word recognition method. Noise and image distortion, as well as the limited accuracy of the recognition process due to unknown position, size and orientation of the textual entries lead to word hypotheses, rather than to unambiguous

readings. In the case of graphic features, we are confronted with an entirely different kind of indexing vocabulary. Here, the features describe the graphical structures in a drawing. As we want to find drawings in a collection that are similar to a query drawing, the matching of similar graphical features is one of our main concerns.

2.1 Text

The first step in recognizing the textual content of a drawing is the segmentation into text entries and graphic structures. The segmentation process is difficult for two reasons. In line drawings, the textual entries are usually very short and graphic elements of the same size as text elements may exist (e.g. small arrows, dashes from dashed lines). The second problem stems from the unknown position and orientation of the textual entries within a drawing. In addition, text entries written with various fonts and sizes may be found in the same drawing. We solve these problems in two steps.

First, we distinguish candidates for character patterns from graphics based on their relative size and structure. Each candidate is passed to an OCR (optical character recognition) method for verification. Our implementation of the OCR method returns, for every character pattern, a list of possible character hypotheses, together with their orientation and a value of confidence. If the pattern could not be identified as a character, a splitting method [8] may be applied for segmenting touching characters.

In the next step we group the individual character candidates into lines of text. We implemented a graph based grouping that is similar to the algorithm proposed in [9]. For every character pattern we create a node in a graph. Nodes are connected with an edge if they are close enough. Individual annotations in the drawing are identified as connected components of the resulting graph. The dominant direction of the edges together with the orientation of the character hypotheses is used to establish the reading direction of the text entries and to verify the character hypotheses. Individual words in the text entries are identified from the relative distance of the characters in the components. The result for an example extracted from Figure 2 is given in Figure 1.

Figure 1: Result of the character grouping. The character patterns identified in the raster image are grouped in a line of text. The beginning of words is indicated with a small circle on the first character of the word.

'L': 0.972 (dir=4)		
unknown		
'a': 0.975 (dir=353)	'o': 0.966 (dir=356)	'O': 0.952 (dir=180)
'T': 0.990 (dir=182)	'l': 0.983 (dir=1)	't': 0.900 (dir=358)
'i': 0.992 (dir=0)	'j': 0.965 (dir=4)	
'T': 0.938 (dir=180)	't': 0.927 (dir=357)	'f': 0.915 (dir=5)
'y': 0.732 (dir=3)		(dir=356)

Table 1: Results of the word recognition for *locality*. The OCR method returns for every character pattern a list of hypotheses, together with a value of confidence and the orientation (0 degrees means upright). We show only the best three hypotheses. Note that we did not apply the character splitting method. Therefore, the touching characters *oc* in the word have not been recognized.

The resulting description of a word consists of the lists of the character hypotheses that were found oriented consistently with the established reading direction. The results for the word `locality` from Figure 2 are given in Table 1. This initial description may be further verified using contextual information. This information can be given as word lists or as probabilities of letter combination or letter substitution. The short text entries and the multitude of fonts used in line drawings do not allow for reliable estimates of probabilities. In our approach we verify a word by comparing it with a dictionary word. The lists of character hypotheses are passed to a dictionary lookup method. Character patterns that were rejected by the OCR method but successfully integrated into a word are handled as an arbitrary number of unrecognized characters. They may be substituted by the lookup method with any matching characters. The result of this verification (Table 2) is a list of words from the directory together with a weight that is computed from the values of confidence of the individual character hypotheses used for the word. These word hypotheses and their weights can now be used in an IR system. The results of the localization and recognition process for an example line drawing are given in Figure 2.

legality	0.97
locality	0.97
lethality	0.97

Table 2: The word hypotheses returned for `locality`. Note how the two unrecognized characters have been substituted. We allowed up to three characters for the substitution. Since the same recognized characters were used for all three word hypotheses the weights are equal.

However, this approach is not without problems. A dictionary is not only language-specific but also domain-dependent: Some drawings originate from technical applications and therefore use terms found only in a dictionary of technical terms. The same holds for drawings containing medical or economic information. Furthermore, the dictionary must contain all forms of a word (e.g. plurals). Stemming methods can only be applied later for a more efficient retrieval process. The weighting of the returned words would also need some refinement. It should not depend only on the value of confidence of the character hypotheses but on the complete contents of the dictionary as the information content (i.e., the *entropy* in information theory) is not the same for all characters of a word. The entropy of a character can be computed with a statistical analysis of all entries in the dictionary.

Depending on the kind of queries we want to answer, it may be sufficient to keep the initial description for words as descriptions for the text. The similarity of two words may be defined as a function of the length of their longest common subsequence (LLCS) [10]. $LLCS(w_i, w_j)$ gives the characters that occur in both words, w_i and w_j , in the same order (e.g. $LLCS(\text{monogram}, \text{mango}) = 3$). This measure can easily be generalized for character hypotheses if the LLCS is not incremented by one but by the value of confidence of a matching character hypothesis. However, this approach cannot be used for an IR system since it involves expensive computation to answer a query.

2.2 Graphic content

In this work we focus on features that are useful in describing the shape of the graphic structures. Methods used in object recognition [11] describe the shape of an object using either *global* or *local* features. Global features, such as area, perimeter or moments, require the segmentation of the content of an image into individual objects. However, segmentation is not possible without *a priori* knowledge about the content of a drawing (i.e., about the domain of the drawing). Local features are computed from parts of an image and can be extracted independent of the semantics of the objects in the image. Normally, local features are extracted along the contours of filled polygons. For line drawings, we extract the local features from the line structures without further segmentation. Retrieval is possible as long as similar graphic structures lead to similar graphic features.

Figure 2: An example of a line drawing with the results of the localization and recognition process. All word hypotheses with their weight are shown next to the text entries.

2.2.1 Token features and indexing features

To use a vector based IR system, every drawing has to be described using a finite indexing vocabulary, Φ , of graphic features. The retrieval method creates the resulting list of similar drawings based on the description vectors, $\vec{d}_j$, and, $\vec{q}$. No further feature computation occurs at query time. In particular, there is no matching at the feature level but only between description vectors. This led us to generalize the idea of the indexing procedure in a vector based retrieval system [12]. We distinguish between *token features*, θ_k , that are extracted from a drawing and *indexing features*, φ_i , that are used for the indexing the drawing. To realize matches of similar features we use a method that is known in signal processing as *anti-aliasing*. Every token feature, θ_k , has a single vote which is distributed among the indexing features, φ_i . The part of the vote that an indexing feature, φ_i , receives from a token feature, θ_k , is a function of the Euclidean distance between the token feature and all neighboring indexing features. In this way we map each token feature to a set of indexing features. The sum of the votes of all token features for a single indexing feature, d_j , is taken as its feature frequency $ff(d_j, \varphi_i)$. We compute the document frequency, $df(\varphi_i)$, similarly. Both, ff and df values can be interpreted as expectation values. This approach to indexing is general enough to allow any

indexing features that can be described as a multi-dimensional vector.

2.2.2 Feature construction

Starting from the raster image of a line drawing, we first identify graphic primitives, such as lines and curves. The primitives are then grouped to derive graphic features that describe the graphic content of a drawing. The grouping is done without incorporating any knowledge about the domain of the drawing. A simple example for a graphic feature is a feature that describes a pair of parallel lines. It can be represented as a vector (*length of first line, length of second line, distance of the two lines, orientation, position*). If we ignore the position and orientation components and normalize the lengths and distance with the length of the longest line, we find a representation (*ratio of lengths, normalized distance*) that is invariant under scaling, rotation and translation.

In [12] we present a feature type to index the graphic part; this feature type is based on local shape. This method computes local features from groups of adjacent line segments in the graphic structures. Every feature is represented as a vector of the first few harmonics of its Fourier approximation. This representation maps the two-dimensional line structure into a single point in the multi-dimensional feature space and proved to be robust for moderate changes in shape.

2.2.3 Relative position

By using local features, we lose the global structure of the drawing. Therefore, we have to address the problem of how to describe the relative position of the local features. Matching the arrangement of features at query time is not possible because we cannot derive the original token features from the indexing features. Even if we could provide a method for this back-transformation, the complexity of the matching is prohibitive at query time.

We conclude that knowledge about the structure of a drawing needs to be incorporated into the indexing features. One solution would be to create a new type of features that includes several indexing features plus their spatial arrangement, A_S . If the description is to be orientation-invariant, we need to combine at least three features into a positional feature ($\varphi_1, \varphi_2, \varphi_3, A_S$). This approach would lead to an explosion in the number of features needed for indexing, because each token feature, θ_k , leads to several indexing features φ_j .

We therefore chose a method that is very similar to the n-gram method used in text based indexing. In the n-grams based method, Instead of using the semantic units of words, overlapping sequences of characters are taken as text features. Similarly, we create overlapping graphical features, that is, the same graphic primitive may be part of more than one group. Similar structures will produce a higher number of similar features.

3 Application and Results

In this section we show how the concepts that were presented earlier may be applied to retrieval of graphic images. In [12] we showed some results using features derived from chains of adjacent lines in the graphic structure of line drawings. Unfortunately the collection used was quite small. To test our system more thoroughly, we have now used a collection of 5400 images of Chinese characters. Our concern was not to create an OCR program for a Chinese alphabet. We regard these images as interesting glyphs and do retrieval exclusively based on local shape. Although we could have used the same features as we did for our experiments on line drawings, this time we extracted the features from the contours. We extracted the sequences of adjacent lines along the contours of the Chinese characters. An example is given in Figure 3. The reason was that our simple image analysis program to extract line primitives was not able to process this input as we had assumed a uniform width of the lines. In Chinese characters, the norm is to use strokes of varying width. However, the features created from the line primitives are again groups of adjacent line segments and the mapping of token features into indexing features was not changed.

Figure 3: Feature extraction for Chinese characters. The image on the left shows a polygonal approximation of the contours. The corners of the polygon are indicated by circles. On the right side we show a sequence of adjacent line segments (in thick lines) used to compute the graphical features.

In Figure 4 we show some results that were produced by our system for the collection of Chinese characters. Although we have not created models for the various subparts of Chinese characters, we are able to retrieve drawings with common subparts using only local similarities. Structural information is incorporated in the overlapping of the graphic features.

4 Evaluation

There are several factors that determine the quality of an IR system. The query formulation process has to be simple enough to allow inexperienced users to use the system. The evaluation time should be short and the answer set should be presented in an intuitive way. Apart from these factors concerning the efficiency of the system, the effectiveness of the system is an even more important aspect of the performance of an IR system.

4.1 Problems in evaluating an image retrieval system

A standard way of evaluating the effectiveness of an IR system is to compare its *recall and precision graph* [13] with graphs of existing systems. The recall and precision graph shows how the precision depends on the recall. The recall and precision graph is determined based on a *test collection* which consists of a collection of documents and a set of queries together with *relevance assessments* that define the two sets D_r^{non} of non-relevant documents and D_r^{rel} of relevant documents for each query. A retrieval method, A , is considered more effective than another method, B , if the recall and precision curve of A is above the curve of B .

Therefore, to evaluate a system the following three elements must be available:

- A *standard collection* of documents,
- a standard set of queries together with relevance assessments and,
- at least one other IR system that can be used for comparison.

This testing environment is available for text-based IR systems, where large standard collections of documents (e.g. the TREC initiative [14]) and a set of queries, with relevance assessments given by experts, are available. Regrettably, there are no such standard evaluation environments available for image data. All

Query	Results (ranked from left to right)						
斤	沂	匠	近	芹	版	所	祈
多	侈	移	移	列	夥	殆	名
王	玉	住	土	匡	往	主	任
區	臨	嶇	嘔	咒	品	嫗	噩
沫	味	沫	採	妹	床	株	未
遁	迫	逢	道	遏	遑	迴	遐
亞	啞	壺	壘	氫	惡	弼	鞋

Figure 4: Some results of our system using the collection Chinese characters. The query drawings are shown in the left column. For each query, the six best ranked drawings are shown in the rows to the right of the query.

image retrieval systems are evaluated on small proprietary collections of images, with a set of *ad hoc* queries. Collections rarely include more than a hundred images. There are a number of reasons why no standard evaluation environment exists for image retrieval systems:

- Most image retrieval systems are experimental systems that investigate specific aspects of a retrieval system such as access structures, similarity functions, query languages and feature selection.
- The features used for indexing have a strong influence on the kind of possible queries. For example, if the features describe the color layout of an image, queries based on shape similarity are not possible.
- Most image retrieval systems are specialized for specific applications.

Therefore, the evaluation experiments of image retrieval systems have always concentrated on individual parts of the system. So far, there are no standard collections available for benchmarking complete image retrieval systems.

4.2 Evaluation of our system

We are currently evaluating our system step-wise: we are evaluating the individual parts of our system based on the goals that we want to fulfill. We evaluate the feature extraction process by testing how much

the features change when the shape of the object varies. We evaluate the mapping from token features to indexing features by analyzing the feature space. Using tools to examine the feature space, we detect the presence of undesirable clusters in the feature space. This indicates that the indexing vocabulary has to be improved. We evaluate the matching at query time by examining the features that caused the match. We identify the graphic features from the query that had the highest influence on the indexing features of the documents of the resulting list.

5 Conclusions

We presented an IR system for the retrieval of line drawings. The restriction to line drawings facilitates the handling of the input (i.e., the feature extraction) without violating the claim for application-independence. We employ features that can be extracted from any line drawing without prior knowledge about their content. The features may be characterized as either textual or graphic. Textual features are extracted from the textual content using word-recognition methods. Graphic features describe the local shape of graphical structures and are derived by grouping graphic primitives. To allow the retrieval of drawings that contain similar shapes without matching at query time, we map the graphical features to a finite vocabulary of indexing graphical features. The indexing features are used in a vector based IR system. The frequency values needed for feature weighting are computed as expectation values. A prototype system has been implemented and some results are presented here. In addition, we described the problems in evaluating an image retrieval system.

References

- [1] Ramesh Jain. Workshop Report: NSF Workshop on Visual Information Management Systems. In *Storage and Retrieval for Image and Video Databases, San Jose, CA, USA*, volume 1908, pages 198 – 218. SPIE, 1993.
- [2] T. L. Kunii, editor. *Visual database systems: Proceedings of the IFIP TC 2/WG 2.6 Working Conference on Visual Database Systems: Tokyo, Japan, April 3 – 7*. Elsevier Science Publishers B.V., 1989.
- [3] E. Knuth and L. M. Wegner, editors. *Visual database systems, II: proceedings of the IFIP TC2/WG2.6 Second Working Conference on Visual Database Systems, Budapest, Hungary, 30 September – 3 October*. Elsevier Science Publishers B.V., 1992.
- [4] F. Rabitti and P. Savino. Querying Semantic Image Databases. In *Image Storage and Retrieval Systems, San Jose, California, USA, February 13 – 14*, volume 1662, pages 69 – 78. SPIE, 1992.
- [5] G. Halin and N. Mouaddib. An Object Oriented Approach to Design a Content-Based Image Retrieval Model. In A.A. Jamberdino and W. Niblack, editors, *Image Storage and Retrieval Systems, San Jose, California, USA, February 13 – 14*, Proc. SPIE 1662, pages 100 – 111. SPIE.
- [6] J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic. Automatic and Semi-Automatic Methods for Image Annotation and Retrieval in QBIC. In *Storage and Retrieval for Image and Video Databases III, San Jose, CA, USA, February 9 – 10*, volume 2420, pages 24 – 35. SPIE, 1995.
- [7] T. Kato. Database architecture for content-based image retrieval. In *Image Storage and Retrieval Systems, San Jose, California, USA, February 13 – 14*, volume 1662, pages 112 – 123. SPIE, 1992.
- [8] G. Monagan. An Approach for Segmenting Touching Numbers in Cadastral Maps. In *Proceedings of the IAPR Workshop on Machine Vision Applications: MVA '94*, pages 427 – 430, Kawasaki, Japan, December 13 – 15 1994.

- [9] L. O’Gorman. Image and Document Processing Techniques for the RightPages Electronic Library System. In *Proceedings of the 11th International Conference on Pattern Recognition, The Hague, The Netherlands*, volume 2, pages 260 – 263. IAPR, 1992.
- [10] S. Kuo and G. R. Cross. A Two-Step String-Matching Procedure. *Pattern Recognition*, 24(7), 1991.
- [11] D. H. Ballard and C. M. Brown. *Computer vision*. Prentice-Hall, 1982.
- [12] O. Lorenz and G. Monagan. A Retrieval System for Graphical Documents. In *4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA, April 24 – 26*, pages 291 – 300, 1995.
- [13] C.J. Van Rijsbergen. *Information Retrieval*. Butterworth & Co Ltd, 1979.
- [14] D. Harman. The Text REtrieval Conference (TREC). In *Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA*, pages 43 – 67, April 1994.