NEGATIVE SHAPE FEATURES FOR IMAGE DATABASES CONSISTING OF GEOGRAPHIC SYMBOLS

AYA SOFFER
Computer Science and EE Department
University of Maryland Baltimore County
Baltimore, MD 21250 and
CESDIS, Goddard Space Flight Center
E-mail: soffer@cs.umbc.edu

HANAN SAMET
Computer Science Department and
Center for Automation Research and
Institute for Advanced Computer Science
University of Maryland at College Park
College Park, Maryland 20742
E-mail: hjs@umiacs.umd.edu

Abstract

A method for representing geographic symbols for storage and retrieval in an image database is presented. Symbols are characterized by a collection of features that describe their shape. Many of these geographic symbols are composed of a circle (or rectangle) enclosing one or more small shapes. A new representation of such symbols based on their interior with the shapes considered as holes, termed a *negative symbol*, is described. A set of shape features used to characterize geographic symbols is presented. These features are appropriate for symbols that are represented by their negation as well as for symbols that are composed of only one part and thus are not represented by their negation. Negative symbols along with these features have been used successfully to index maps in a map image database system. Results of experiments testing the accuracy of the database using these features are given.

1 Introduction

Consider a database containing a large number of images each composed of several symbols that have some meaning (e.g., the symbol on a paper map that represents a museum). Such images are called *symbolic images* and their domain includes maps, engineering drawings, and floor plans. For example, suppose that we wish to find all of the images in the database that contain a particular symbol. There are generally two methods of achieving this capability. In the first method, termed *classification*, each symbol in each image is classified (i.e., it is replaced by its symbolic meaning). In the second method, termed *abstraction*, each symbol is represented by some properties of its visual representation (e.g., shape, length, connectivity, genus, etc.) termed a *feature vector*. Classification can potentially be performed using template matching. However, template matching is limited to cases where all instances of the same symbol have the same scale and orientation. Therefore, classification should be performed using a more robust method such as statistical pattern recognition based on some descriptive features of the symbols. In both methods for storing images it is desirable to represent a symbol by a small number of numeric descriptors.

In our work, we concentrate on databases consisting of images that contain geographic symbols. These images could be, for example, the symbol layer of a tourist map. An interesting characteristic of such symbols is that many of them tend to be composed of a circle (or rectangle) enclosing one or more small shapes (e.g., the beach and hotel symbols in Figure 1 which shows the geographic symbols that we have studied along with their semantic meaning). Ideally, we would like to represent each symbol in the database with only one feature vector in order to simplify the search process at query time. Therefore, for symbols that are composed of several shapes, we must select one of these shapes to represent it. This may lead to ambiguity and as a result we may not be able to distinguish between some symbols. In order to overcome this problem, we propose to represent such symbols by the interior of the circle with the small shapes considered as holes in this object (termed a *negative symbol*). For example, the "beach" symbol in Figure 1 would be represented by the interior of a circle with the two wiggly lines as holes.

Most image databases treat images as a whole and index them on the basis of color and texture [7, 8, 14]. However, shape features have also been used [1, 11]. In all of these cases the objects that are being indexed based on shape are assumed to be simple (i.e., composed of only one part). Most research in map recognition has concentrated on skeletonization and vectorization methods [15]. Some research has been done on separating the layers of scanned maps [13]. Recognizing geographic symbols in the context of map recognition has also been considered [5, 16]. The methods employed in these studies are either very computationally expensive or require the user to explicitly build a semantic model that is used to perform the classification of map objects. Thus, these methods are not applicable in the context of an image database.

In this paper, we study the appropriateness of using the negative symbol representation. In addition, we describe a set of shape features that can be used to characterize symbols represented by their negation as well as symbols that are only composed of one part and thus are not represented by their

Figure 1: Geographic symbols and their semantic meaning.

negation (termed *positive symbols*). These features should be easy to compute since we need to calculate them for a large number of images. Furthermore, the number of features should be relatively small in order to facilitate efficient comparison of features. Finally, these features should be effective in discriminating between different geographic symbols. We have used negative symbols along with these features to successfully index maps in a map image database system that we have developed [10]. In this paper, we assume that we are using the classification approach to storing images. That is, we are classifying the geographic symbols as the map images are input to the system. However, these same features can also be stored directly in the database if using the abstraction method [12].

The rest of this paper is organized as follows. Section 2 presents an overview of the symbol recognition method that we used. Section 3 motivates our use of negative symbols. Section 4 describes the shape features that we use to represent geographic symbols. Section 5 discusses the implementation and experimental results. Section 6 contains concluding remarks.

2 Overview of Symbol Recognition Method

Figure 2 is an overview of the method. The symbol layer of the map is scanned. Next, segmentation is performed which identifies the individual symbols to be classified. Features are extracted for each individual symbol. Finally, each symbol is classified. In our system, segmentation is performed via a connected

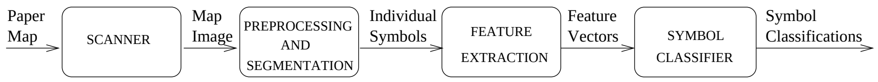

Figure 2: Block diagram of symbol classification system

component labeling algorithm. This results in a labeled image in which each pixel has a region number as its value.

As mentioned above, complex map symbols are represented by the interior of the symbol with the enclosed shapes considered as holes. This enables us to represent symbols such as the beach symbol that are composed of more than one piece by only one connected component. Our system assumes that the symbols may be distinguished from each other by just one connected component using this method.

Symbols are classified based on the shape features that are computed for the connected component using a *weighted bounded several-nearest neighbor classifier* [2]. This classifier makes use of a training set with several representative feature vectors for each symbol. We construct an initial training set library by analyzing the legend and inserting the feature vector of the connected component that was chosen to represent each symbol into the training set. The training set is later expanded by adding feature vectors of symbols that are identified by the user as having been erroneously classified. For more details about this process, see [10].

3 Why Use Negative Symbols?

We encountered several problems when using only positive symbols. We use Figure 3, a typical image used in our application, to illustrate the difficulties with positive symbols. Recall, that our goal is to represent each symbol by only one feature vector in order to simplify the search process at query time.

The main problem with positive symbols is ambiguity. For example, using only positive symbols we cannot distinguish between the hotel ◙ symbol and the youth hostel ■ symbol. If we choose to represent the hotel ◙ by the square inside the circle, then there is no difference between it and the youth hostel ■ symbol. The circle cannot represent the hotel ◙ either, as it is common to many symbols. However, using negative symbols the hotel ◙ is represented by a circle with a square cut out of it and the youth hostel ■ is represented by a square, and thus we can distinguish between the two. Another source of ambiguity is that using positive symbols we cannot distinguish between a letter that is enclosed in a circle, and is thus a legend symbol, and a letter that

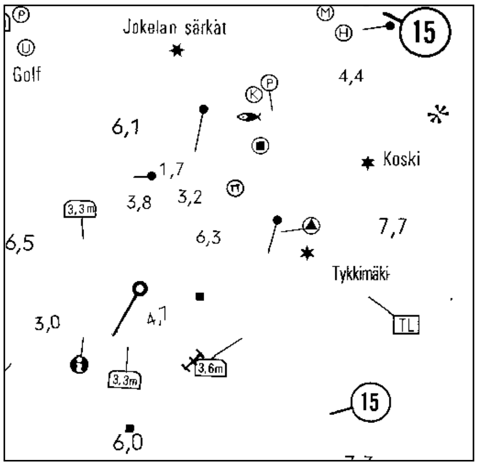

Figure 3: Example map tile

is part of a word. For example, we cannot distinguish between the symbol for cafe Ⓚ and the "K" in the word "Kuski" (see Figure 3). However, using negative symbols the cafe Ⓚ is represented by a circle with a "K" cut out of it, and thus the "K" in the word "Kuski" will not be erroneously considered a cafe Ⓚ .

It is worth noting that if we were to represent symbols by more than one region, we would encounter additional problems using positive symbols. For example, we could represent the hotel as a combination of a square and a circle. However, we would then need to also model the topological layout (that is, that the square is inside the circle). The database search process would need to verify this at query time, and as a result searching would be much more complex. In addition, if two symbols touch each other (e.g., ⓀⓅ), then the two circles would be considered one region and we could not recognize either symbol. However, this is not a concern when using negative symbols. Although the circles touch, the interiors do not, and thus we have two separate regions and this problem does not arise.

As part of the segmentation process, we set a threshold for the minimal acceptable size of a segment. Any segment (i.e., connected component) that is smaller than this threshold is not considered a potential symbol. When using positive symbols, this threshold has to be smaller than when using negative symbols since a symbol may be represented by a relatively small component. For example, we could represent the beach symbol by one of the waves .

The size of the wave is relatively small. In order to ensure that we do not discard it during segmentation, the minimal size for acceptable segments needs to be relatively small as well. As a result, each image contains more segments and thus the database is larger. In addition, a larger number of segments that do not represent valid symbols will potentially be erroneously classified as valid symbols (false hits). Using negative symbols, the beach symbol is represented by a circle with the two waves cut out. The size of this region is much larger than each individual wave. We can thus set a higher threshold and each individual wave is discarded during segmentation.

To summarize, the reason that map symbols are enclosed in a circle is to group several pieces into one and to delineate the difference between symbols and text. The negative symbol representation naturally captures this grouping using only one connected region. Furthermore, the size of this region is large relative to the size of the individual components that make up the positive symbol, and thus many non-relevant symbols can be filtered out during segmentation by use of a simple threshold.

4 Shape Feature Selection

The selection of the particular shape features that are used to characterize the symbols is one of the most important factors in achieving high recognition rates. Numerous shape features have been described in the literature (e.g., [6]). These features are generally either based on the boundary or interior representation of the object. The boundary-based shape features capture the "jaggedness" and complexity of the object. In order to use these features, the object is usually approximized by a polygon. Measurements such as the number of sides, the relative length of these sides, and the angles between the sides of the polygonal approximation are then used to describe the shape of the object. Since we are performing symbol recognition as part of the input process of an image database, we wanted to keep the computation as simple and quick as possible. Thus, we did not use such features.

The features that we use are all based on the interior representation of the symbols as output by the connected component labeling process. Recall that each symbol is represented by one of its connected components. In particular, in the case of complex symbols, this component is the negation of the symbol (i.e., the interior of the circle). Figure 4 demonstrates this process for an example symbol. The symbol has three regions after connected component labeling. Region 3, which is the interior of the circle with the letter "H" cut out is selected to represent this symbol. The shape features that we compute approximate the boundary complexity and the shape regularity of the symbol's

(a) (b) (c)

Figure 4: Processing one example symbol: (a) Binary representation of the symbol. (b) Symbol after connected component labeling. Each region has a different color. (c) Bit map of the symbol after connected component labeling. Region 3 is selected to represent this symbol.

representative component based on the area of this component, the perimeter of the component, and the major and minor axes of the component. In addition to these global features, we also compute some local features pertaining to holes in the component. We define a hole as a region that is enclosed in another region either vertically or horizontally. These features are required in order to distinguish between symbols that are represented by their negation. The global shape is very similar for all of these components as they are all basically circles. The discriminating factor between them is the shape of the holes. Note that all of the features that we compute require only one pass over the image following connected component labeling. We use the following shape features.
F_1: First invariant moment [4], which is defined as

$$M_1 = \mu_{20} + \mu_{02}.$$

Here μ_{pq} is the central moment given by

$$\mu_{pq} = \sum_i \sum_j (i - \overline{x})^p (j - \overline{y})^q I(i,j),$$

where $I(i,j)$ is the intensity (grey level) at point (i,j) in image I, and $\overline{x}$ and $\overline{y}$ are defined as follows:

$$\overline{x} = \frac{m_{10}}{m_{00}} \qquad \overline{y} = \frac{m_{01}}{m_{00}}.$$

m_{pq} is the two-dimensional $(p+q)$th order moment and is defined as

$$m_{pq} = \sum_i \sum_j i^p j^q I(i,j).$$

This feature is invariant to scale and orientation. The invariant moment is computed for the connected component that was selected to represent the symbol. For the example symbol in Figure 4, we compute the invariant moment for region 3 and its value is 0.2078. Note that since this symbol is represented by its interior, we are actually computing the moment of the "white" part of the symbol in its binary representation.

F_2: Eccentricity [9], which is defined as

$$E = \frac{|D - W|}{D},$$

where W and D are the length of the minor and major axis of the component. Eccentricity is a measure of the elongation of the symbol. It is 0 for a square and approaches 1 for elongated shapes. Eccentricity is useful for characterizing positive symbols (i.e, symbols not represented by their negation) such as the scenic view symbol in Figure 1. However, it does not discriminate well between negative symbols since the length of the axes is not affected by the holes in their representative component, and eccentricity will thus have similar values for all such symbols. For the example symbol in Figure 4 which is represented by region 3: width = 15, height = 16, and thus eccentricity = 0.9375.

F_3: Circularity [3], which is defined as

$$C = \frac{P^2}{4\pi A},$$

where P is the perimeter of the component, and A is its area. Circularity equals 1 for a circle and takes on larger values for distortions therefrom. It is useful for characterizing both positive and negative symbols. The area and perimeter of negative symbols is computed for the component that represents the interior in effect measuring the "white" region in the binary representation of the symbol. For the example symbol, area = 175, perimeter = 92, and thus circularity = 3.848.

F_4: Rectangularity, which is defined as

$$PB = \frac{A}{A_{bb}},$$

where A is the area of the component, and A_{bb} is the area of the minimal rectangle enclosing the component. Rectangularity equals 1 for a rectangle and approaches 0 for least rectangular shape. If the bounding box is restricted to being parallel to the axes, then this feature is not invariant to rotation. Rectangularity is not very effective for negative symbols since the bounding box is not affected by the hole area. However, since the component area is affected by the hole area, there will be some variation in rectangularity between negative symbols. For the example symbol, area = 175, bounding box area = 240, and thus rectangularity = 0.729.

F_5: Hole area ratio, which is defined as

$$HAR = \frac{A_h}{A},$$

where A is the area of the component and A_h is the total area of all holes in the component. Hole area ratio is most effective in discriminating between symbols that have big holes (e.g., the first aid station symbol) and symbols with small holes (e.g., the beach symbol). For the example symbol, area = 175, hole area = 32, and thus hole area ratio = 0.18.

F_6: Horizontal gaps ratio, which is defined as

$$HGR = \frac{HG^2}{A},$$

where A is the area of the component, and HG is the number of horizontal gaps in the component (i.e., the number of pixels in the object whose right neighbor is a hole). This feature discriminates among symbols based on the shape of the holes themselves. For the example symbol, area = 175, horizontal gaps = 18, and thus horizontal gaps ratio = 1.85.

F_7: Vertical gaps ratio, which is defined as

$$VGR = \frac{VG^2}{A},$$

where A is the area of the component, and VG is the number of vertical gaps in the component (i.e., the number of pixels in the object whose bottom neighbor is a hole). It has the same characteristics as the horizontal gaps ratio. For the example symbol, area = 175, vertical gaps = 8, and thus vertical gaps ratio = 0.366.

5 Implementation and Experiments

The shape features described above were incorporated into the feature extractor of a map image database system developed by us [10]. Using these shape features, the system was tested on the red sign layer of the GT3 map of Finland. The scale of the map is 1:200,000. The layer was scanned at 240dpi. Figure 3 shows the extracted red sign layer. An initial training set containing 22 symbols was constructed from the legend. 60 tiles were used to expand the training set to 100 samples. The remaining tiles were processed automatically.

In the context of an image database, the classifier is evaluated in terms of the accuracy of the results of a query requesting images that contain particular symbols. We evaluated the accuracy using two error types that are commonly used in document analysis studies. A type I error occurs when an image that meets the query specification was not retrieved by the system (a miss), and a type II error occurs when an image that the system retrieved for a given query does not meet the query specification (a false hit). Note that type I and type II errors correspond to the recall and precision metrics, respectively, used in information retrieval experiments.

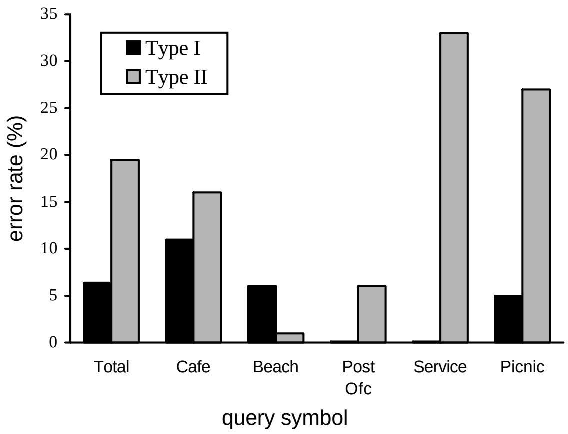

Figure 5: Type I and type II error rates.

We computed the error rates by querying the database for all tiles containing each symbol in our application. Type II error rates were calculated by counting how many results did not meet the query specification for each symbol. The total type II error rate is the total number of incorrect results (over all symbols) divided by the total number of results. Type I errors, were calculated by determining which tiles contain each symbol, and checking whether any result tiles were missed. We did this for 50 tiles (out of the 425 tiles) chosen at random and for each one of the symbols. Type I error rates were computed for each symbol in addition to a total type I error rate which is the total number of missed results divided by the total number of results we should have had.

Figure 5 reports the total type I and type II error rates, as well as these error rates for a few of the symbols. The total type I error rate was 6% (i.e., 94% of the tiles that should have been retrieved were in fact retrieved by the system). The rates varied from 0% for the post office Ⓟ symbol to 11% for the cafe Ⓚ symbol. The total type II error rate was 19% (i.e., 81% of the tiles that were retrieved did in fact contain the desired symbol). It varied from 1% for the beach ⊜ symbol to 33% for the service station Ⓗ symbol. We attribute the variance in the error rates between different symbols to the ability of the system to distinguish between them based on the selection of shape features, the classification method, and the content of the training set used for classification. Although the results that we report here are for one particular training set, we experimented with various configurations, and these results were consistent in all cases. In order to achieve lower error rates, more features would most likely be required.

6 Concluding Remarks

Representing geographic symbols that are composed of more than one component by their negation was effective for distinguishing between such symbols. The experimental results showed that with the particular shape features that were used, we can on average retrieve 94% of the required images. We are currently analyzing the effects of the individual features both analytically via principal component analysis, and empirically by using subsets of the features for classification. In addition, we plan to explore whether using additional features such as higher order moments, minimum bending energy of the curvature, and other curvature-based features will improve the accuracy. Furthermore, for symbols that are represented by their negation we need to investigate other features that can characterize the shape and the distribution of the holes as an aggregate that are invariant to scale and orientation (e.g., average circularity and eccentricity of the holes).

Acknowledgements

We are grateful to Karttakeskus, Map Center, Helsinki, Finland for providing us the map data. The support of USRA/CESDIS and NASA Goddard Space Flight Center is gratefully acknowledged. The support of the National Science Foundation under GrantIRI-92-16970 is gratefully acknowledged.

References

[1] A. Del Bimbo and P. Pala. Image indexing using shape-based visual fetaures. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume III, pages 351–355, Vienna, Austria, August 1996.

[2] J.L. Blue, G.T. Candela, P.J. Grother, R. Chellappa, and C.L. Wilson. Evaluation of pattern classifiers for fingerprints and OCR applications. *Pattern Recognition*, 27(4):485–501, April 1994.

[3] S.B. Gray. Local properties of binary images in two dimensions. *IEEE Transactions on Computers*, C-20(5):551–561, May 1971.

[4] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8(2):179–187, February 1962.

[5] R. Kasturi and J. Alemany. Information extraction from images of paper-based maps. *IEEE Transactions on Software Engineering*, 14(5):671–675, May 1988.

[6] M.D. Levine. *Vision in Man and Machine*. McGraw-Hill, New York, 1982.

[7] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture, and shape. In *Proceeding of the SPIE, Storage and Retrieval of Image and Video Databases*, volume 1908, pages 173–187, San Jose, CA, February 1993.

[8] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *Proceeding of the SPIE, Storage and Retrieval of Image and Video Databases II*, volume 2185, pages 34–47, San Jose, CA, February 1994.

[9] A. Rosenfeld and J.L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1(1):33–61, July 1968.

[10] H. Samet and A. Soffer. MARCO: MAp Retrieval by COntent. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):783–798, August 1996.

[11] S. Sclaroff. Encoding deformable shape categories for efficient content-based search. In A. W. M. Smeulders and R. Jain, editors, *Proceedings of the 13th International Conference on Pattern Recognition*, pages 107–114, Amsterdam, The Netherlands, August 1996.

[12] A. Soffer and H. Samet. Retrieval by content in symbolic-image databases. In *Proceedings of the SPIE, Storage and Retrieval of Still Image and Video Databases IV*, volume 2670, pages 144–155, San Jose, CA, February 1996.

[13] S. Suzuki and T. Yamada. MARIS: Map recognition input system. *Pattern Recognition*, 23(8):919–933, August 1990.

[14] M. Swain. Interactive indexing into image databases. In *Proceeding of the SPIE, Storage and Retrieval for Image and Video Databases*, volume 1908, pages 95–103, San Jose, CA, February 1993.

[15] N. Tanaka, T. Kamimura, and J. Tsukumo. Development of a map vectorization method involving a shape reforming process. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 680–683, Tsukuba Science City, Japan, October 1993.

[16] H. Yamada, K. Yamamoto, and K. Hosokawa. Directional mathematical morphology and reformalized hough transformation for analysis of topographic maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):380–387, April 1993.