Using Negative Shape Features for Logo Similarity Matching

Aya Soffer [*], 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: {aya,hjs}@umiacs.umd.edu

Abstract

A method for representing and matching logos based on positive and negative shape features is presented. Negative shape features represent an object that consists of several components enclosed in a simple geometric structure (e.g, a square) based on its interior with the components considered as holes. The goal is to find logos in a database that are most similar to a given sample logo. A border is added around logos that are not enclosed in a simple shape. Logos are segmented. Local and global shape features are computed for each component. Two methods for comparing logos represented by positive and negative components are presented and evaluated.

1. Introduction

With the proliferation of systems that store electronic images, there is much interest in similarity based image retrieval. The goal is to retrieve all images that are similar to a given query image. Most work on finding similar images has concentrated on properties of the entire image such as color and texture (e.g., [3, 5]). Shape-based similarity is used in some systems [1]. In most of these cases the objects that are being indexed based on shape are assumed to be simple (i.e., composed of only one part).

In our previous work [7], we devised a method for representing geographic symbols using shape features for storage and retrieval in an image database. Many of these geographic symbols were composed of a circle (or rectangle) enclosing one or more small shapes. We proposed a new representation of such symbols based on their interior with the shapes considered as holes, termed a *negative symbol*. We use the idea of negative features for more complicated symbols that are not necessarily enclosed in a circle or rectangle. As an example application, we selected logo recognition. Negative features are based on having some geometric shape (e.g., square or circle) that encloses several smaller shapes. While, many logos by their design already have this property (e.g. Figure 1a), others do not (e.g., Figure 1b). We propose to artificially add a border around these logos and thus create a negative symbol. The goal is to have a database of logo images, and then given a sample logo, find the logos in the database that are most similar to it. Several studies have reported results on logo recognition using methods such as local invariants, wavelet features, etc.(e.g., [2, 4, 8]). The goal in these studies is to locate a logo in the database based on a sketch of the logo, or with the introduction of noise, or variations in orientation or scale. In contrast, our goal is to find logos that are "similar" to an input logo.

(a) (b) (c)

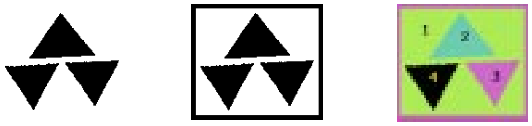
Figure 1. Some sample logos.

We describe two methods for comparing logos that are represented by their positive and negative components. The first method represents each logo by one component (termed *representative-component matching*). The second method represents each logo by several components (termed *multi-component matching*). Logo matching with and without adding the artificial border is compared. We also compare the results using a representative component for each logo to using all components.

2. Overview of Logo Matching Method

The input is a bitmap image of a scanned logo. We add a rectangular border around each logo which is the minimum bounding box of the image with 4 pixels added to

*The support of the National Science Foundation under Grant CDA-950-3994 is gratefully acknowledged.
†The support of the National Science Foundation under Grant IRI-97-12715 is gratefully acknowledged.



(a) (b) (c)

Figure 2. (a) Example logo; (b) with border; (c) after connected component labeling (each component is annotated with its component number).

each side. The added pixels are assigned the color (black or white) of the logo's background. Each logo is segmented into its constituent components using a connected component labeling algorithm. Each region is given a number. One of these regions corresponds to the border that was added around the image. Figure 2a is an example logo from our system, Figure 2b is the logo after adding the border, and Figure 2c has the components resulting from connected component labeling. The component labeled "1" in Figure 2c is the negative component resulting from adding the border. Shape features are extracted for each component that makes up the logo. We use four global shape descriptors (first invariant moment, circularity, eccentricity, and rectangularity) and three local shape descriptors (horizontal gaps per total area, vertical gaps per total area, and ratio of hole area to total area)[7].

Each logo is represented by a set of m feature vectors, where m is the number of regions in the logo following connected component labeling. The similarity between two logos is computed based on this set of feature vectors. We use two methods to compute this similarity. The first method chooses one component to represent each logo. The similarity between two logos using the representative component measure is the weighted Euclidean distance between the feature vectors of these two representative components. The representative component is selected by the user, and will most likely be a negative component whether inherent to the logo or one that was added by the border. In the example logo in Figure 2c, the representative component would be the one labeled "1". That is, a square with three triangles cut out.

The second method compares logos based on all of the components as follows. For each component in logo L_1, find the distance to the component of logo L_2 that is closest to it in feature space. The similarity between two logos is the average of these distances for all components of logo L_1. The similarity score for two identical logos X and Y is 0, since each component in X will have a matching region in Y with distance 0. Notice that this score is not reflexive — that is, $S_{all}(L_1, L_2) \neq S_{all}(L_2, L_1)$ since the number of components in L_1 and L_2 need not be the same.

3. Implementation and Experiments

The database we use consists of 130 logos from the University of Maryland logo database. Evaluating the effectiveness of logo similarity matching is a challenge in itself. The problem is that since our goal is to find logos in our database that are most similar to an input logo, we must first define the notion of similarity in this context. Currently we are using two tests.

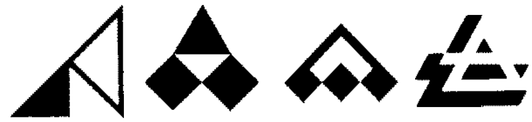


Figure 3. Example logos from a "triangular" class

Figure 4. Example logos from a "long text" class

Figure 5. Example logos from a "stripes" class

For the first test, we identified several classes of logos in our database for which we consider all logos in each class similar to each other. Given any logo from one of these classes as input, we expect to find the other logos from the same class as most similar to it. Figures 3– 5 are examples of three of the classes that we identified. For the second test, we edited a few logos by removing, adding or distorting some of the components that make up these logos. Given this edited logo as input, we expect to find the original logo as most similar to it. See Figures 6 and 7 for examples of edited versions of a logo.

Each test evaluates our method for the following cases. (1) Do not add a border around the logos and compare logos using all of the components for each logo with the multi-component similarity measure described above. (2) Add a border around the logos and use one representative component for each logo with the representative component similarity measure. (3) Add a border around the logos and compare using all of the components for each logo with the multi-component similarity measure.

3.1. Evaluation procedure

The first test, termed *class association*, evaluates retrieval accuracy using the *normalized recall* metric [6] (NR). NR measures how close to the top of the list of retrieved items the relevant (i.e., correct) items appear compared to the ideal retrieval in which all N relevant items appear in the top N positions. We use these measures (assume that there are n_i logos in class C_i): *IAVRR*:

Figure 6. (a) Original logo; (b)–(e) edited logos

Figure 7. (a) Original logo; (b) – (d) edited logo.

ideal average rank of relevant items. Ideally, if the query logo is a member of class C_i, then all n_i logos from the same class C_i should be ranked at the top. Thus, $IAVRR(C_i) = \sum_{j=1}^{n_i} j/n_i$ (since the ideal rankings are $1 - n_i$). *AVRR*: average rank of relevant items for a particular retrieval seeking logos from the same class as query logo Q. $AVRR(Q) = \sum_{L \in class(Q)} R(L)/n_{class(Q)}$, where $R(L)$ is the ranking of logo L in the result, *class(Q)* is the class of Q, and $n_{class(Q)}$ is the number of logos in the class of Q. Normalized recall is the ratio of *AVRR* to *IAVRR* and measures the deviation from the ideal ranking. $NR = AVRR/IAVRR = 1$ for a perfect retrieval, and is > 1 for all other cases. We use the normalized recall metric to compute the average normalized recall for each class, the average normalized recall over all classes, and the average normalized recall over all tested logos.

The second test (edited logos) checks the rank of the original logo when the edited logo is the query logo. Ideally, the original logo should be ranked number one. A lower ranking indicates that the method is less effective.

3.2. Results

Table 1 summarizes the average normalized recall (*ANR*) results for each logo class using the three logo matching methods that we propose. Based on these results, the method that adds a border to the logos and then uses all components for comparison (labeled "Border/All") is best for the class association test. Table 2 gives the rank of the original logo for each of the edited logos from Figure 6 (the "Sun" logo) as input, and the edited logos from Figure 7 (the "Digital" logo) as input. Based on these results, using positive and negative shape features is very effective for logo similarity matching. In particular, the method that adds a border to the logos and uses all components for comparison (labeled "Border/All") is also the best for the edited logo test.

4. Concluding Remarks

Adding borders to logos and characterizing them with both positive and negative shape features for logo similarity matching using our similarity measures is effective both for finding all logos that belong to the same class and for finding a logo given an edited version of it. We are currently analyzing the effects of the individual features. In addition, we plan to explore whether using additional features will improve the accuracy.

	No Border	Border/One	Border/All
triangular	5.72	4.31	2.51
long narrow text	7.50	11.78	1.22
rectangular text	13.57	12.47	1.27
square	8.53	5.73	3.86
stripes	2.67	1.44	1.22
average over all classes	7.60	7.15	2.02
average over all logos	4.56	5.74	1.24

Table 1. Normalized recall results for each logo class using no border, border with one, and with all components (normalized recall $= 1$ for perfect retrieval, > 1 otherwise).

	No Border	Border/One	Border/All
sun(b)	2	43	1
sun(c)	1	2	1
sun(d)	1	1	1
sun(e)	1	1	1
digital(b)	1	1	1
digital(c)	1	2	1
digital(d)	14	120	1

Table 2. Rank of original logo for the edited logos from Figures 6 and 7.

References

[1] A. D. Bimbo and P. Pala. Shape indexing by multi-scale representation. In A. W. M. Smeulders and R. Jain, editors, *Proceedings of the First International Workshop on Image Databases and Multimedia Search*, pages 43–50, Amsterdam, The Netherlands, Aug. 1996.

[2] D. Doermann, E. Rivlin, and I. Weiss. Applying algebraic and differential invariants for logo recognition. *Machine Vision and Applications*, 9(2):73–86, 1996.

[3] C. Faloutsos, R. Barber, W. Equitz, M. Flickner, W. Niblack, and D. Petkovic. Efficient and effective querying by image content. *Journal of Intell. Info. Sys.*, pages 231–62, 1994.

[4] M. Jaisimha. Wavelet features for similarity based retrieval of logo images. In L. M. Vincent and J. J. Hull, editors, *Proc. of SPIE, Document Recognition III*, volume 2660, pages 89–100, San Jose, CA, Jan. 1996. IS&T and SPIE.

[5] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intell.*, 18(8):837–842, Aug. 1996.

[6] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[7] A. Soffer and H. Samet. Negative shape features for image databases consisting of geographic symbols. In *Advances in Visual Form Processing*, pages 569–581, Singapore, 1997. World Scientific.

[8] P. Suda, C. Bridoux, B. Kämmerer, and G. Maderlechner. Logo and word matching using a general approach to signal registration. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pages 61–65, Ulm, Germany, Aug. 1997.