Proc of the 16th Int. Conf. on Pattern Recognition, vol. III, pp. 521-524, Quebec City, Canada, August 2002.

Content-based Image Retrieval Using Fourier Descriptors on a Logo Database*

Andre Folkers
Institute for Signal Processing
Medical University of Lübeck
23569 Lübeck, Germany
E-mail: folkers@isip.mu-luebeck.de

Hanan Samet
Computer Science Department
University of Maryland at College Park
College Park, Maryland 20742
E-mail: hjs@umiacs.umd.edu

Abstract

A system that enables the pictorial specification of queries in an image database is described. The queries are comprised of rectangle, polygon, ellipse, and B-spline shapes. The queries specify which shapes should appear in the target image as well as spatial constraints on the distance between them and their relative position. The retrieval process makes use of an abstraction of the contour of the shape which is invariant against translation, scale, rotation, and starting point that is based on the use of Fourier descriptors. These abstractions are used in a system to locate logos in an image database. The utility of this approach is illustrated using some sample queries.

1. Introduction

A basic requirement of an image database is to perform content-based searches for images. In [9] we presented a pictorial query specification tool for objects represented by points. In this paper we extend this tool to permit query primitives that have a spatial extent such as ellipses, rectangles, polygons, and B-splines. Breaking down the image in terms of these primitives can be viewed as a classification approach. At times, it could be the case that the objects in the image cannot be decomposed into components that are comprised of these primitive shapes. In this case, we need an abstraction of the contour of the shape which ideally is invariant against translation, scale, rotation, and starting point, while still representing the essential form of the contour. We propose a method based on Fourier descriptors to achieve this effect. A comparison with other shapes representations, e.g. with invariant moments was made in [4]. In the rest of this paper we describe our query specification and shape description methods, and study their effectiveness by querying a database of logos obtained from the US patent office (e.g., Figure 2(c)). For related work with logos, see [1, 7, 12] where the goal 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 whose components consist of user-defined shapes, while also providing a way of specifying the extent of the similarity.

*This work was supported in part by the National Science Foundation under Grants EIA-99-00268, IIS-00-86162, and EIA-00-91474.

2. Image Processing

The shapes in a logo image are preprocessed by morphological closing and opening operations using structure elements of different sizes. This either disconnects components connected only by a few pixels (opening) or connects components which are only a few pixels apart from each other (closing). Furthermore, the closing or opening smooths the contours of the shapes.

Each image component is approximated with each of the following shape types: rectangle, ellipse, polygon, and B-spline. Next, the quality of these approximation shape types is evaluated, and the image component is classified as the shape type that approximates it best. In case of a tie, the approximation shape type with the smaller number of parameters is given priority, e.g., classification as an ellipse is preferred to classification as closed B-spline. Currently we can only handle the rectangle, ellipse, polygon, and B-spline shapes. Future work includes recognizing a straight line, arc, chord, pie slice, and freehand drawing shapes.

3. Logo database

The components of the logo images are stored using a prototype spatial database system called SAND (spatial and non-spatial data) developed at the University of Maryland [2]. In SAND, data is stored in tuples consisting of attributes for geometric entities such as points, lines, polygons, etc. (some of which are supported for arbitrary number of dimensions), in addition to traditional ones like integers, floating point numbers, and character strings. SAND can index both spatial – including high-dimensional points – and non-spatial data using different methods (e.g., PMR-trees, R-trees, kd-trees [8]). The central relation is the table of extracted image components. For each image component, we store the result of the classification which is one of the approximation shape types described above, a minimum bounding box as a rough approximation of the spatial extent of the component, and a reference to the image from which the component was extracted. The possible classifications are represented by the query primitives (e.g., a polygon, ellipse, etc.). In Section 5 we describe how pictorial queries are specified using these query primitives. All query primitives can have individual spatial representations, e.g., the representation of a polygon consists of a list of points, while an ellipse can be represented using a rectangle. Thus, we

have different relations to store the spatial representation of image components.

In some cases, it may be impossible to classify the image components using one of the approximation shape types. In this case, we represent the component as a feature vector, termed its *abstraction* in contrast to its *classification*. The database contains both the classification and the abstraction for each image component, where in some cases the classification may be unknown. The advantage of storing both the classification and abstraction in the database for each component of a logo image is that this provides more flexibility for the query processing as the user may want to use the abstraction for a particular query image component instead of its classification (see [10] for more details).

4. Fourier Descriptors

Each image of the logo database is decomposed into its image components, i.e., its connected components. As an abstract representation of image components, we use their Fourier descriptors, which are made invariant against translation, scale, rotation, and their starting point. We retain the phase which contains essential information about the contour of the image component.

4.1. Definition and Properties

Consider the N contour points of an image component as a discrete function $\mathbf{x}(n) = (x_1(n), x_2(n))$. Using this function, we can define a discrete complex function $u(n)$ as

$$u(n) = x_1(n) + jx_2(n).$$

u(n) can be transformed into the frequency domain by the *Discrete Fourier Transformation* (DFT). The result can be transformed back into the spatial domain via the *Inverse Discrete Fourier Transformation* (IDFT) without any loss. DFT and IDFT are defined as $a(k)$ and $u(n)$, respectively:

$$a(k) = \frac{1}{N}\sum_{n=0}^{N-1} u(n)\ e^{-j2\pi kn/N}\ \ k = -N/2, \ldots, N/2-1$$

$$u(n) = \sum_{k=0}^{N-1} a(k)\ e^{j2\pi kn/N} \qquad n = -N/2, \ldots, N/2-1$$

The coefficients $a(k)$ are also called Fourier descriptors [6]. They represent the discrete contour of a shape in the Fourier domain.

Certain geometric transformations of the contour function $u(n)$ can be related to simple operations in the Fourier domain. Translation by $u_0 \in \mathbb{C}$ affects only the first Fourier descriptor $a(0)$, while the other Fourier descriptors retain their values. Scaling of the contour with a factor α leads to scaling of the Fourier descriptors by α. Rotating the contour by an angle θ_0 yields a constant phase shift of θ_0 in the Fourier descriptors. Changing the starting point of the contour by n_0 positions results in a linear phase shift of $2\pi n_0 k/N$ in the Fourier descriptors [6].

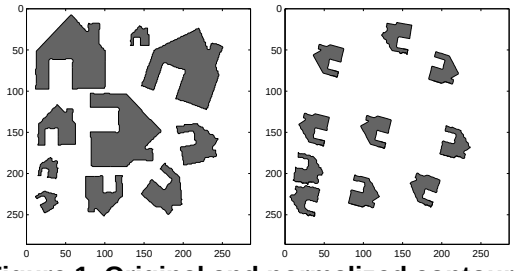


Figure 1. Original and normalized contours.

4.2. Normalization

The contour functions are made invariant against translation by setting the first Fourier descriptor $a(0)$ to zero which moves the centroid of the contour onto 0. Since the contours are traced counterclockwise and describe a nonzero area, we can rely on the fact that the second Fourier descriptor $a(1) = r_1\ e^{j\varphi}$ is nonzero [5] (tracing it clockwise would imply that $a(-1)$ is nonzero for a contour with nonzero area). Therefore, we can divide all Fourier descriptors by the magnitude of the second Fourier descriptor to obtain a scale invariant vector: $a(k) = a(k)/|a(1)|$.

Rotation invariance could be achieved by simply taking the magnitude of each Fourier coefficient, but hereby an essential part of the information about the contour is lost. To achieve rotation invariant Fourier descriptors that still represent the shape of the original contour, we can use the orientation of the basic ellipse, which is defined by the Fourier descriptors $a(1) = r_1\ e^{j\varphi_1}$ and $a(-1) = r_{-1}\ e^{j\varphi_{-1}}$, as

$$u_e(n) = a(-1)\ e^{-j2\pi n/N} + a(1)\ e^{j2\pi n/N}.$$

With a few transformations, and with the abbreviation $\tilde{\varphi} = (\varphi_1 - \varphi_{-1})/2$, we get

$$u_e(n) \quad = \quad e^{j(\varphi_{-1}+\varphi_1)/2} \left[r_{-1}\ e^{-j(\tilde{\varphi}+2\pi n/N)} + r_1\ e^{j(\tilde{\varphi}+2\pi n/N)}\right]$$

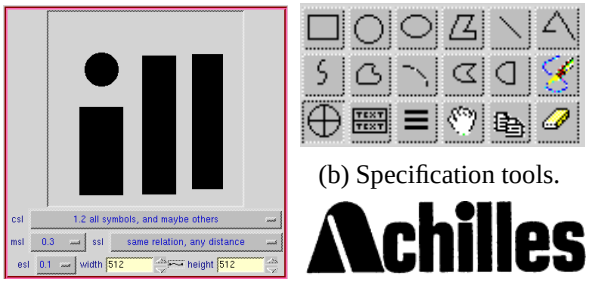
which shows the rotation φ_e of the basic ellipse as:

$$\varphi_e = (\varphi_{-1} + \varphi_1)/2.$$

Using the orientation of the basic ellipse leads to an ambiguity of π radians. Therefore, the 'rotation invariant' Fourier descriptors are only rotation invariant modulo a rotation by π radians. Figure 1 illustrates this for a set of equal contours at different scales and rotations which are normalized by the above operations.

Now, only the position of the starting point remains to be normalized. This can be done by subtracting the phase of the second Fourier descriptor, weighted by k, from the phase of all Fourier descriptors, that is,

$$a(k) = a(k)\ e^{-j\varphi_1 k}.$$



(a) Specification canvas. (b) Specification tools. (c) Result.

Figure 2. Query Specification and result.

After this normalization, the starting point is approximately at angle 0.

5. Pictorial Query Specification

We briefly review how individual pictorial queries are specified using our method. For more details and examples, see [3, 11]. The *matching similarity level* (*msl*) is a value between 0 and 1. Components in the database are considered to match a query component if they are similar with respect to a certain similarity measurement (see Section 6). The *extent similarity level* (*esl*) is an upper bound for the difference in spatial extent. The *contextual similarity level* (*csl*) specifies how well the content of a database image D matches that of a query image Q. The choices are:

1. All components, and no others.
2. All components, and maybe others.
3. Any of the components, but no others.
4. Any of the components, and maybe others.

The *spatial similarity level* (*ssl*) specifies how close the database image D and the query image Q are with respect to distance and directional relation between the symbols in the query. We distinguish between five different levels which are defined as

1. exact same location
2. same relation, bounded distance
3. same relation, any distance
4. any relation, bounded distance
5. any relation, any distance.

A pictorial query consists of query primitives. The user selects the desired query primitive using check buttons (see Figure 2(b)). Next, an instance of the query primitive can be drawn onto the canvas (see Figure 2(a)). Currently, the implementation supports the rectangle, circle, ellipse, polygon, and closed B-Spline query primitives. Note, that circles are included into the interface in order to help the user to draw real circles. During the processing phase they are handled as ellipses. Query primitives can be positioned arbitrarily. The matching, contextual, spatial, similarity, and extent level for the query is defined using combo boxes (see the bottom of Figure 2(a)).

6. Pictorial Query Processing

A query can be processed in either classification or abstraction mode. In the classification mode, the database is searched for the specified query primitives. In order to check the extent similarity of an image component and a query primitive we compute the factors α_1 and α_2 that scale the respective query primitive and image component so that it fits into a rectangle of width 1. Then the difference is determined as $|\alpha_1 - \alpha_2|$ which is required to be less than *esl*. Similarity between two ellipses or two rectangles is measured using the aspect ratio of their minimum bounding boxes. Similarity of two polygons is measured using the ratio between the area of the intersection polygon and the maximum area of both polygons. This ratio becomes 1 if the polygons are equal, and is zero if they do not intersect. Similarity of two closed B-Splines is measured using the same method, whereby the control points are interpreted as defining polygons. This makes sense as a B-spline is uniquely determined by its control polygon.

Due to the ambiguity of π radians in the 'rotation invariant' Fourier descriptors that was described in Section 4, we maintain two Fourier descriptors in the abstraction mode for each component in the query image after which the starting point of both vectors is normalized. Next, the database is searched for both vectors. For this, we use a spatial index provided by SAND to perform a fast search for vectors in the database which are at a small Euclidean distance from the vectors of the query component. The maximum distance for the nearest neighbor search is determined by the value of *msl*. Currently we use 16 Fourier descriptors ($k = -8, \ldots, -1, 2, \ldots, 9$), which result in 32-dimensional vectors (as we have both real and imaginary parts, and we ignore $a(0)$ and $a(1)$ since they are 0 and 1, respectively, due to normalization). The factors α_1 and α_2 are given by the first Fourier descriptor of the respective component as $1/|a(1)|$, and the extent similarity check is done as described above.

After a set of matching images has been found, the contextual similarity and spatial similarity check is performed. This has been described in detail in [3].

7. Sample Queries and Results

Figure 2(a) is an example of a pictorial query consisting of four shapes: a circle and three rectangles. These shapes are required to have the same spatial relation as ssl=3. The contextual similarity level (i.e., 2) stipulates that all components in the query image must occur in the database image, while the database image may contain other components. The extent similarity level is set to 0.1 which allows quite different sizes for the matching components. Figure 2(c) shows the result of this query.

Figure 3 shows six logos that satisfy the contextual constraint of containing at least three rectangles and one circle

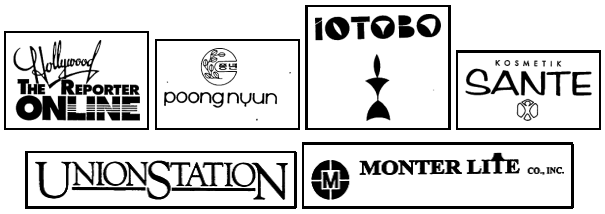
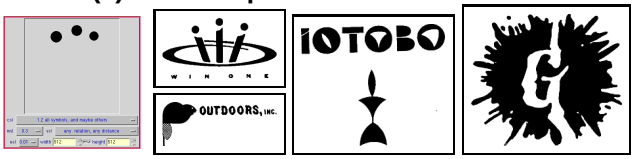


Figure 3. Logos retrieved by the query in Figure 2(a) with no spatial constraints.



(a) (b) and (c) (d) (e)

Figure 4. (a) Sample query with three circles, and (b)–(e) the logos that satisfy the contextual constraint of containing three circles.

for the query in Figure 2(a). The logo in Figure 2(c) also fulfills it. This logo is the only one in the database that also satisfies the query's spatial constraint.

Figure 4(a) is another query. It contains three circles. csl=2 and ssl=3 (i.e. the components may have any distance but the directional relation must be similar). Figure 4(b)–(e) shows some logos that only satisfy the contextual constraints. Closer scrutiny of Figure 4(e) reveals the absence of any circles. The shapes are obviously misclassified. The other two result logos (Figures 4(d) and 4(c)) actually contain three circles. The best result of the query is shown in Figure 4(b). Here we find the circles in the desired spatial configuration.

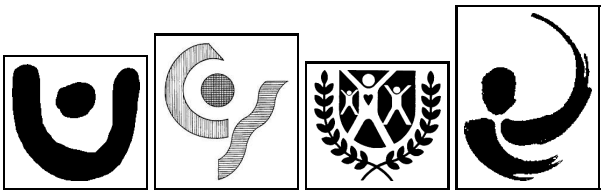
Finally, Figure 5 shows an example query processed in abstraction mode and the retrieved logos. It contains a u-bend and a circle. csl=2 and there are no spatial constraints. The sizes are supposed to be similar. The result images are ordered by their matching rank, i.e., Figure 5(b) has first rank and is the best match. All result logos actually contain a u-bend and a circle, except for the one in Figure 5(h). Note, that the u-bend occurs at different rotations.

8. Concluding Remarks and Future Work

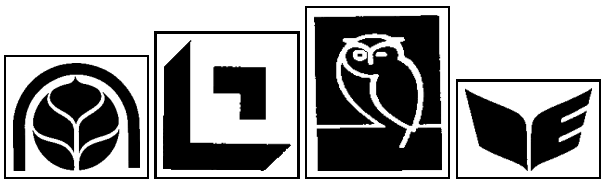
Both classification and abstraction queries produced acceptable results. The main advantages of the abstraction approach compared with the classification approach is that we have a greater flexibility in choosing an arbitrary shape of the components of the query image. Future work involves extending the pictorial query specification tool to incorporate other shapes as well as freehand shapes.

References

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



(a) Query (b) 0.162 (c) 0.213 (d) 0.214



(e) 0.237 (f) 0.303 (g) 0.313 (h) 0.340

Figure 5. (b)–(h)Logos that are retrieved based on Fourier descriptors using query image (a), msl=0.6, csl=2, esl=0.01, and ssl=5.

[2] C. Esperança and H. Samet. An overview of the SAND spatial database system. Submitted for publication.

[3] A. Folkers, A. Soffer, and H. Samet. Processing pictorial queries with multiple instances using isomorphic subgraphs. In *Proc. ICPR'00*, volume 4, pages 51–54, Barcelona, Spain, Sept. 2000.

[4] D. Heesch and S. Rüger. Combining features for content-based sketch retrieval – A comparative evaluation of retrieval performance. In F. Crestani, M. Girolami, and C. J. van Rijsbergen, editors, *Proc. of the 24th BCS-IRSG European Colloquium on IR Research*, Lecture Notes in Computer Science 2291, pages 41–52, Berlin, 2002. Springer.

[5] B. Jähne. *Digital Image Processing*. Springer, Berlin, 4 edition, 1997.

[6] A. K. Jain. *Fundamentals of Digital Image Processing*. Information and Systems Science Series. Prentice Hall, 1989.

[7] M. Jaisimha. Wavelet features for similarity based retrieval of logo images. In *Proc. of SPIE Document Recognition III*, volume 2660, pages 89–100, San Jose, CA, Jan. 1996.

[8] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.

[9] A. Soffer and H. Samet. Pictorial query specification for browsing through spatially referenced images databases. *Jour. of Vis. Lang. and Comp.*, 9(6):567–596, Dec. 1998.

[10] A. Soffer and H. Samet. Two approaches for integrating symbolic images into a multimedia database system: a comparative study. *VLDB Journal*, 7(4):253–274, Dec. 1998.

[11] A. Soffer, H. Samet, and D. Zotkin. Pictorial query trees for query specification in image databases. In *Proc. ICPR'98*, volume 1, pages 919–921, Brisbane, Australia, Aug. 1998.

[12] P. Suda, C. Bridoux, B. Kämmerer, and G. Maderlechner. Logo and word matching using a general approach to signal registration. In *Proc. ICDAR '97*, pages 61–65, Ulm, Germany, Aug. 1997.