International Journal of Image and Graphics, Vol. 2, No. 3 (2002) 1–19

SHAPE-BASED IMAGE RETRIEVAL APPLIED TO TRADEMARK IMAGES

OSSAMA El BADAWY* and MOHAMED KAMEL†

Pattern Analysis and Machine Intelligence Laboratory
Department of Systems Design Engineering, University of Waterloo
200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada
obadawy@pami.uwaterloo.ca
mkamel@pami.uwaterloo.ca

Received 1 July 2001
Revised 28 November 2001
Accepted 13 March 2002

We propose a new shape-based, query-by-example, image database retrieval method that is able to match a query image to one of the images in the database, based on a whole or partial match. The proposed method has two key components: the architecture of the retrieval and the features used. Both play a role in the overall retrieval efficacy. The proposed architecture is based on the analysis of connected components and holes in the query and database images. The features we propose to use are geometric in nature, and are invariant to translation, rotation and scale. Each of the suggested three features is not new per se, but combining them to produce a compact and efficient feature vector is. We use hand-sketched, rotated and scaled query images to test the proposed method using a database of 500 logo images. We compare the performance of the suggested features with the performance of the *moments invariants* (a set of commonly-used shape features). The suggested features match the moments invariants in rotated and scaled queries and consistently surpass them in hand-sketched queries. Moreover, results clearly show that the proposed architecture significantly increases the performance of the two feature sets.

Keywords: Shape Analysis; Shape Representation; Shape-Based Image Retrieval; Image Databases; Trademark Image Retrieval.

1. Introduction

Today, a lot of images are being generated at an ever increasing rate by diverse sources such as earth orbiting satellites, telescopes, reconnaissance and surveillance planes, fingerprinting and mug-shot-capturing devices, biomedical imaging, payment processing systems, and scientific experiments.

There is a pressing need to manage all this information that keeps pouring into our life each day. By *managing* images we more specifically want to be able to efficiently store, display, group, classify, query and retrieve those images. An image database management system is generally required if we are to get the most out of a large image collection. One of the major key services that should be offered by an image database system is its ability to provide *content-based image retrieval*, or

CBIR. The goal of content-based image retrieval is to retrieve database images that contain certain visual properties, as opposed to retrieving images based on other properties such as creation date, file size or author/photographer name. For example, we might be interested in retrieving images where blue is the most abundant color (color-based image retrieval). Another example is when we ask the system to, say, get us all images that contain squares or rectangles (shape-based image retrieval).

Application areas in which CBIR plays a principal role are numerous and diverse. Among them are art galleries and museum management, trademark and copyright database management, geographic information systems, law enforcement and criminal investigation, weather forecasting, retailing and picture archiving systems.

The focus of this paper is on shape-based image retrieval using a feature extraction approach. The main goal is to be able to retrieve database images that are most similar to a hand-sketched query image. Figure 1 shows a block diagram of the image retrieval model used in the feature extraction approach. The input images are pre-processed to extract the features which are then stored along with the images in the database. When a query image is presented, it is pre-processed to extract its features which are then matched with the feature vectors present in the database. A ranked set of images with high matching scores are presented at the output.

We want to be able to retrieve database images similar, in whole or in part, to a user-supplied query image. In particular, we want to be able to retrieve database

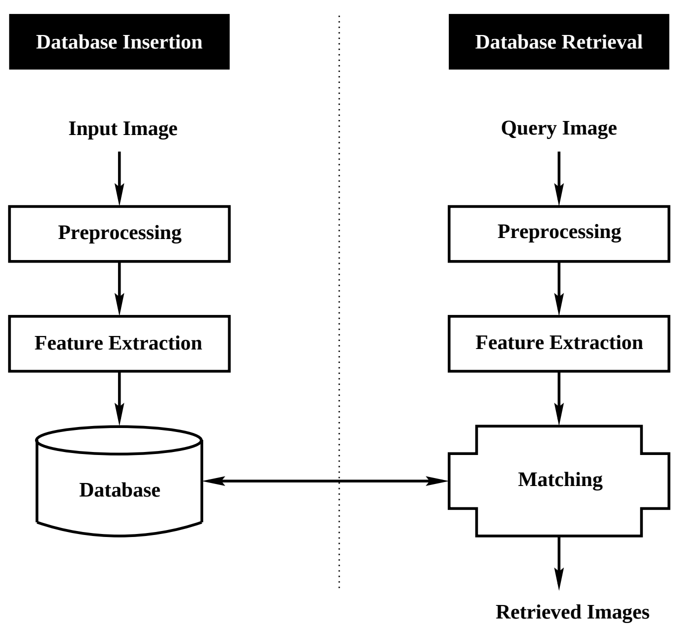


Fig. 1. Feature extraction approach to image retrieval.

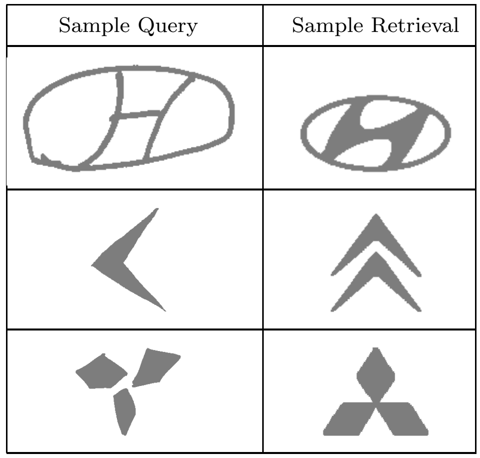


Fig. 2. Examples of queries that we want to be able to correctly answer. Left column: Query image. Right column: Retrieved image.

images based on (see Figure 2):

- a *whole* match with scaled, hand-sketched queries (first row),
- a *partial* match with scaled, rotated, hand-sketched queries (second row),
- a match where the number of components in the query and the retrieved database image is different but the overall shape is similar (third row).

We propose a retrieval architecture based on the analysis of connected components and holes in the image. Any feature set can be used with this architecture, but we suggest the use of a set of three known object properties as image features suitable for shape-based retrieval. The suggested feature set is invariant to translation, rotation and scale. The performance of the proposed features is compared to that of the moments invariants feature set in two ways: once in conjunction with the proposed architecture and once on their own. All images used are binary images.

The rest of the paper is organized as follows. Section 2 provides a literature review of recent shape-based image retrieval research. Then, in Sec. 3, the necessary theoretical background is presented and the shape feature set is explained in detail. Section 4 details the proposed architecture. Performance analysis results are presented in Sec. 5. Finally, the conclusion and a discussion of future work are provided in Sec. 6.

2. Overview of Current Methods

Nowadays, numerous shape analysis and representation techniques exist. These include chain codes, polygonal approximations, signatures, edge histograms, Fourier descriptors, moments and image morphology.[1] As shape is one of the main characteristics of images, it has been used a lot in image database systems as a way of automatically retrieving images by content. In almost all implemented systems, other image cues (i.e. texture and color), in addition to shape, are also used for image retrieval. In this paper, we focus more on shape-based retrieval.

Safar presents a shape representation method[2] based on minimum bounding circles and touch-point vertex-angle sequences and compares it with other common representation like Fourier descriptors and Delaunay triangulation. In another paper,[3] the authors conduct a comparative study on various shape-based representation techniques under various uncertainty scenarios, like the presence of noise and when the exact corner points are unknown. Lu presents a region-based approach to shape representation and similarity measure.[4] Kwok uses edge histograms as shape features (and other color features) for retrieval from a database of 110 flower images[5] while Xu presents a shape representation algorithm based on the morphological skeleton transform. The algorithm represents shapes as the union of special maximal rectangle contained in the shape.

An algorithm that uses snakes and invariant moments is presented by Adoram,[7] and Kim a modified Zernike moment descriptor that takes into account the importance of the outer form of the shape to human perception proposes.[8] Muller formulates a deformation-tolerant stochastic model suitable for sketch-based retrieval.

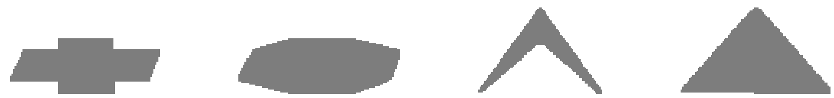
IBM's QBIC system primarily uses statistical features to represent the shape of an object.[10] These include a combination of area, circularity, eccentricity, major axis orientation and algebraic invariant moments. The ART-MUSEUM system was developed for the purpose of archiving a collection of artistic paintings.[11] In this system, the database consists of the contour images, constructed by extracting all the prominent edge points in a painting. Hand-sketched queries are matched with database images by dividing the images into 8×8 blocks and computing a global correlation indicator. In STAR,[12] both contour Fourier descriptors and moments invariants are used for shape representation and similarity measurement.

Several research papers tackled the trademark matching and retrieval issue using different approaches: invariant moments,[13] chain coding,[14] edge histograms and deformable template matching.[15] In the Artisan system,[16] all images are segmented into components and geometric similarity features are extracted from the boundaries of these components.

3. Shape Analysis

The word "shape" is very commonly used in everyday language, usually referring to the appearance of an object. More formally, a shape is: *all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.*[17]

If one is interested in shape-based image retrieval, then obviously, color and texture will not play a useful role in distinguishing between various shapes. Consequently, binary images suffice for extracting good shape features. This section presents a discussion of shape analysis in binary images. It particularly discusses



solidity = 0.86 solidity = 0.45

(a) (b)

Fig. 3. Examples of object solidities. The solidity can be computed as the area of the object divided by the area of its convex hull; (a) solid object, (b) less-solid object.

each of the three shape properties suggested for use as a compact shape-description feature vector.

There are many properties that can be used in describing shapes. These include measurements of area and perimeter, length of maximum dimension, moments relative to the centroid, number and area of holes, area and dimensions of the convex hull and enclosing rectangle, number of sharp corners, number of intersections with a check circle and angles between intersections. All these measures have the property of characterizing a shape but not of describing it uniquely. Here, we propose the usage of a set of object properties as shape features suitable for shape-based image retrieval. The ideas behind the use of those properties as shape features are fairly simple to grasp. We start by discussing each of the features and then we provide a discussion of the heuristics behind them.

3.1. *The solidity*

The *solidity* of an object can be defined as the proportion of the pixels within the convex hull of the object that are also in the object.[18] It can be computed as the ratio between the object area and the area of the corresponding convex hull. Figure 3 shows two examples of object solidities. In the figure, for each of (a) and (b), the original image is shown on the left and the corresponding filled convex hull is shown on the right. The solidity value, computed as the area of the original image divided by the area of the filled convex hull image, is shown under the convex hull image. Figure 3(a) shows a quite solid object; the solidity is 0.86. On the other hand, a "not-quite-so-solid" object is shown in Fig. 3(b). This is reflected by a solidity of only 0.45.

3.2. *The eccentricity*

The *eccentricity* of an object can be defined as the eccentricity of the ellipse representing the unit-standard-deviation contour of its points. If we view an object image as a set of points in two-dimensional Cartesian space, then the parameters of the unit-standard-deviation ellipse are easily computed from the covariance matrix of the points.[19] The eccentricity of an ellipse is the ratio of the distance between the foci of the ellipse and its major axis length. The eccentricity is always between zero and one. (Zero and one are degenerate cases; an ellipse whose eccentricity is zero is actually a circle, while an ellipse whose eccentricity is one is a line segment).

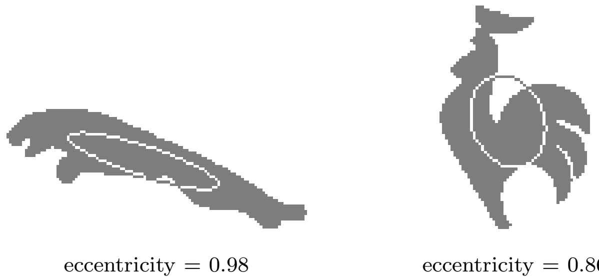


Fig. 4. Examples of object eccentricities. The object on the left is more elongated, and hence has a larger eccentricity, than the object on the right.

Figure 4 shows two examples of object eccentricities. We note that the object on the left is more elongated than the object on the right. This is reflected by a larger eccentricity for the object on the left. The unit-standard-deviation contours are shown superimposed on the images.

3.3. *The extent*

The *minimum bounding rectangle* (MBR) of an object is the smallest rectangle that totally encloses the object. The *extent* of an object can be defined as the proportion of the pixels within the minimum bounding rectangle of the object that are also in the object.[18] It can be computed as the object area divided by the area of the MBR. Figure 5 shows two examples of object extents (the objects here are again logo images). For both part (a) and part (b), the original objects are shown on the left, and the objects enclosed in their MBRs, along with the extent values, are shown on the right.

Some basic shapes (a circle, a square, a rectangle, an ellipse, a pentagon, a hexagon, a star and a crescent) are shown in Fig. 6 along with their corresponding solidity, eccentricity and extent values. We note that, as expected, the solidity of the first six shapes is either one or very close to one. Indeed, any *filled convex* polygon will have a solidity value of one (the fact that the solidity value for the

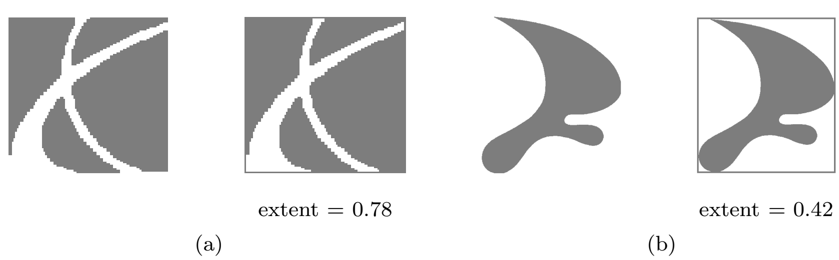


Fig. 5. Examples of object extents. The extent can be computed as the area of the object divided by the area of its minimum bounding rectangle (MBR); (a) Object with high extent value, (b) object with smaller extent value.

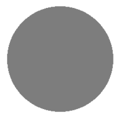
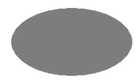
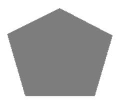
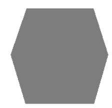
Image	S	C	X	Image	S	C	X
	0.99	0	0.79		1	0.29	1
	1	0.96	1		0.99	0.83	0.78
	0.99	0.45	0.73		0.99	0.40	0.85
	0.51	0.91	0.38		0.47	0.44	0.35

Fig. 6. Some basic shapes and their corresponding solidity (S), eccentricity (C), and extent (X) values.

circle, ellipse, pentagon and hexagon is 0.99 is because those objects are digitally represented by a finite set of pixels that approximates the geometry of the real object). The eccentricity values are also much in accordance with the shapes of the basic objects; the most elongated objects (the rectangle, the crescent and the ellipse) have the highest eccentricity values, whereas the least elongated (the circle) has zero eccentricity. As for the extent, its values are also in accordance with the general shapes of the basic objects (a value of one for the square and the rectangle, a higher extent value for the hexagon than the pentagon, ellipse or circle).

We note that these three features are able, on their own, to easily discriminate between the distinct shapes in Fig. 6. It is worth noting though that the pentagon and the hexagon in the figure exhibit feature values that are pretty close. This is due to the fact that a hexagon and a pentagon can be considered quite similar shapes (although not identical of course).

4. Shape Retrieval Architecture

In this section, we present an architecture for shape-based image retrieval that can be used in conjunction with any feature set. However, we propose the use of a feature vector comprised of the solidity, eccentricity and extent object properties. We will denote this feature set as the SCX feature set.

The main goal is to retrieve images from the database that closely match a hand-sketched shape image. The matching does not have to be based on the whole image. That is, the user-drawn sketch can match a part of a database image. We use the term "partial matching" to denote this process.

A strongly desired property of any shape-based image retrieval system is for the retrieval to be invariant to scale and rotation. Generally, the features used are responsible for achieving this invariance. The SCX feature set, like many others, is invariant to scale and rotation.

The proposed image retrieval architecture encompasses many stages starting as early as the image preprocessing stage, going into the feature extraction stage, and ending with the image query stage. We will discuss each stage in detail in the following sections.

4.1. *Preprocessing*

Each image in the database is first pre-processed prior to the feature extraction process. The goal of the preprocessing phase is to appropriately prepare the image for the feature extraction. The three main tasks performed in the preprocessing are small-object removal, small-hole filling, and boundary smoothing. The first two operations are vital since we do not generally want any small sporadic group of pixels to be treated as an image component, nor do we want any too-small hole to be treated as an real hole. The last operation, object smoothing, is useful only in cases where the object boundaries are horizontal or vertical. It basically removes any parasitic spurs from the object binary image.

4.2. *Feature extraction*

In describing the feature extraction stage of the architecture, we will use the SCX feature set as our backbone feature vector. However, it should be kept in mind that any other feature set can be used.

In the proposed architecture, an image is transformed not to a single feature vector as is usually the case, but to several feature vectors all referring to the image of interest.

Figure 7 shows the steps required to transform a binary image depicting a shape to the feature space. An image is transformed to several feature vectors (or database records). The first record V_i pertains to image i as a whole. Then, L records, v_{ij} $j = 1, \ldots, L$, follow the first one (L is the number of components in the image). Each of those L records represents a component of the image.

An example would help clarify this stage. Consider the three images shown in Fig. 8. The images on the left and in the middle contain two components and one hole each, whereas the one on the right contains three components and two holes. Table 1 shows how they would be represented in the database.

The left-most column in the table contains the image number. In database terminology, this number acts as a "foreign key" referring to the "primary key" in another table containing unique image numbers and file names. The second column is a whole-image flag; it has a value of one for database records that describe whole images, and a value of zero for database records describing image components. There are two uses for this column (also a database field). The first is to distinguish partial

Feature extraction from a binary image database

Procedure PopulateDatabase

foreach image Y_i in the database, $i = 1, 2, \ldots, N$

{*Segment* Y_i *into its* L_i *components using connected components labeling*}

foreach component Z_j in Y_i, $j = 1, 2, \ldots, L_i$

$M_j \leftarrow$ number of holes in component Z_j

$S_j \leftarrow$ solidity of *filled* component

$C_j \leftarrow$ eccentricity of *filled* component

$X_j \leftarrow$ extent of *filled* component

$H_j \leftarrow$ image whose pixels are the set of all pixels in the holes of Z_j

{*Fill all holes in* H_j}

$S'_j \leftarrow$ solidity of the set of all pixels in H_j

$C'_j \leftarrow$ eccentricity of the set of all pixels in H_j

$X'_j \leftarrow$ extent of the set of all pixels in H_j

{*form vector* $v_{ij} = (1, M_j, S_j, C_j, X_j, S'_j, C'_j, X'_j)$}

endfor

$F_i \leftarrow Y_i$ with all holes filled

$S_i \leftarrow$ solidity of the set of all pixels in F_i

$C_i \leftarrow$ eccentricity of the set of all pixels in F_i

$X_i \leftarrow$ extent of the set of all pixels in F_i

$H_i \leftarrow$ image whose pixels are the set of all pixels in the holes of Y_i

{*Fill all holes in* H_i}

$S'_i \leftarrow$ solidity of the set of all pixels in H_i

$C'_i \leftarrow$ eccentricity of the set of all pixels in H_i

$X'_i \leftarrow$ extent of the set of all pixels in H_i

$M_i \leftarrow \sum_{j=1}^{L_i} M_j$

{*form vector* $V_i = (L_i, M_i, S_i, C_i, X_i, S'_i, C'_i, X'_i)$}

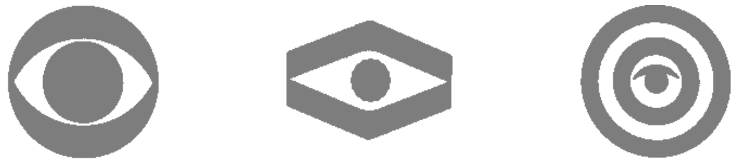
{*vector* V_i *and vectors* v_{ij}, $j = 1, 2, \ldots, L_i$, *are added to the database*}

endfor

endProcedure

Fig. 7. Steps for populating the feature database.

matches from whole-image matches, and the second is to turn on or off the partial-match system feature. The third column is the number-of-components (N) feature. It always has a value of one for components. The fourth column is the number-of-holes (H) feature. The last six columns are the solidity (S), eccentricity (C) and extent (X) of components and holes.



(a) (b) (c)

Fig. 8. Feature extraction example.

Table 1. Feature values corresponding to images in Fig. 8.

Image number	Whole-image flag	N	H	S	C	X	S'	C'	X'
1	1	2	1	0.99	0.18	0.78	0.40	0.86	0.29
1	0	1	1	0.99	0.18	0.78	0.99	0.75	0.71
1	0	1	0	0.99	0.21	0.78	0	0	0
2	1	2	1	0.99	0.78	0.73	0.50	0.96	0.27
2	0	1	1	0.99	0.78	0.73	0.98	0.92	0.54
2	0	1	0	0.98	0.21	0.78	0	0	0
3	1	3	2	0.99	0.22	0.78	0.58	0.22	0.45
3	0	1	1	0.99	0.22	0.78	0.99	0.22	0.78
3	0	1	1	0.99	0.23	0.78	0.98	0.23	0.78
3	0	1	0	0.79	0.59	0.53	0	0	0

For whole images, S, C and X values are the solidity, eccentricity and extent of the image as a whole, but with any holes filled, S', C' and X' are the solidity, eccentricity and extent of the filled holes in the image treated as one object. For image components, on the other hand, S, C and X values are the solidity, eccentricity and extent of the filled component, and, S', C' and X' are the solidity, eccentricity and extent of the filled holes in that component treated as one object (refer to the algorithm in Fig. 7).

4.3. *The matching process*

Querying-by-example is the most common way to search image databases by content.[20] In the case of shape-based image retrieval, a hand-sketch of the sought shape is an adequate means for querying, and we indeed use it. In the proposed system, the matching procedure is as follows:

(i) The query image is first pre-processed (see Sec. 4.1). It is worth noting here that, for the sake of simplifying the system, the hand-sketched query image is assumed to be filled and well sketched (in the sense that it has to be closed if it depicts a closed shape).

(ii) The whole-image feature vector V is extracted from the image. The choice of only extracting V as opposed to also extracting the components feature vectors v_j's depends on what we are interested in. If we want to retrieve database images that match a *part* of the query image, then there are two approaches; either to extract the v_j's and use them in a multi-query-image fashion (which will complicate the query), or to simply sketch only the part that we are interested in from the beginning and use it as the query image.

(iii) (Optional step) Restrict the database to records containing values close to the N and H values of the query image. For example, if for the query image, $N = 3$ and $H = 2$, then we might choose to restrict the database to records where $N \in \{2, 3, 4\}$ and $H \in \{1, 2, 3\}$. This is equivalent to indexing and will help speed up the retrieval when the database gets larger.

(iv) The Euclidean distance was chosen as a distance measure for retrieving the closest images from the database because of its simplicity (other measures can be investigated/used as well). The matching is based on the feature vector $(N, H, S, C, X, S', C', X')$. Depending on whether partial matching is turned on or off (using the whole-image-flag field), some of the retrievals might be based on the query image matching part of the database image.

4.4. *Enhancing the matching process*

To add more power to partial matching, the following modifications were introduced to the architecture. Each image in the database is represented by three aliases: the original image, a dilated version and an eroded version. A new field is added to the database to denote whether the image is the original one or the dilated/eroded version. This flag will be mainly used in order to enable or disable object matching based on dilated/eroded versions of database images. It should be mentioned that a dilated/eroded image is *only* added to the database if the operation changes the number of components or holes in the image. A square 5×5 structuring element is repeatedly used in the morphing until the number of components or holes change.

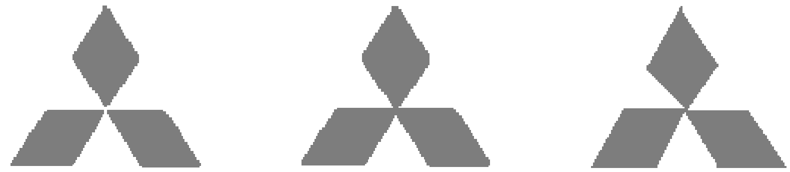
The above-mentioned modifications have two positive effects (discussed in the following two subsections):

(i) enhancing whole-image matching; and
(ii) enhancing partial matching.

On the other hand, the disadvantage is a much larger number of records (an upper limit of about three times the original number). This should not degrade the speed of the retrieval since the indexing mechanism alleviates that. A larger number of records might, however, affect the accuracy of the retrieval. Whether the advantages outweigh the disadvantages is the decisive factor in whether to opt for this modification or not (we can always, for each query image, perform two queries, one with the dilation/erosion enabled and one with it disabled).

4.5. *Enhancing whole-image matching*

Consider the image shown in Fig. 9(a), its dilated version shown in Fig. 9(b), and the query image shown in Fig. 9(c). Obviously, according to the discussion in



(a) (b) (c)

Fig. 9. Enhancing whole-image matching; (a) original image, (b) dilated image and (c) query image.

Table 2. Feature values for the images in Fig. 9.

Image number	Morph. flag	Whole-image flag	N	H	S	C	X	S'	C'	X'
123	0	1	3	0	0.62	0.23	0.32	0	0	0
123	0	0	1	0	0.95	0.83	0.53	0	0	0
123	0	0	1	0	0.91	0.74	0.52	0	0	0
123	0	0	1	0	0.95	0.84	0.55	0	0	0
123	1	1	1	0	0.77	0.23	0.44	0	0	0

Sec. 4.2, the query image (c) will not match image (a) but will match image (b) (if it existed in the database). This is precisely why including image (b) as an alias for the original image (a) in the database will help retrieve the correct image. We should note that image (b) will not be physically added to the database; only a feature vector (or a number of feature vectors) will represent it in the database. This (these) vector(s) will "point" (using the image number field) to the original image. Table 2 shows the feature vectors as they would appear in the database. The second column in Table 2 is a flag that has the value of one for morphologically modified (dilated or eroded) images and a value of zero for original images. We note that since there are no holes in the image in Fig. 9(a), S', C' and X' all have a value of zero. In addition, since erosion will *not* change the number of components in the image, there are no new records showing a value of N greater than 3.

4.6. *Enhancing partial matching*

Similar to the previous discussion, consider the image shown in Fig. 10(a), its eroded version shown in Fig. 10(b), and the query image shown in Fig. 10(c). Again, according to the discussion in Sec. 4.2, the query image (c) will not partially match image (a) but will partially match image (b) (if it existed in the database). This is precisely why including image (b) as an alias for the original image (a) in the database will help retrieve the correct image.

4.7. *Discussion*

The choice of the solidity (S), eccentricity (C), and extent (X) shape measures (or features) is not a necessary condition. One can freely substitute them with other

(a) (b) (c)

Fig. 10. Enhancing partial matching; (a) original image, (b) eroded image, and (c) query image.

measures. Alternatively, other measures can be added to them to form an extended feature vector. However, the use of the SCX feature vector has some interesting properties. As discussed in Sec. 3, they are invariant to rotation, position and scale change. They all have a value range from zero to one; a decent range for features that does not need further normalization in general.

As a matter of fact, the solidity, eccentricity and extent do not uniquely describe a given object; one can easily come up with two different objects with the same values for the three features. However, the goal is to identify "similar" objects and not to uniquely describe them. That is, two objects with close solidity, eccentricity, and extent values are in general similar.

5. Experimental Results

We used a database of 500 logo images to test the performance of the system. The bigger part of the image database used in this paper was obtained from the department of Computer Science at Michigan State University. The original database consisting of 1100 images was used by A. K. Jain and A. Vailaya in an image retrieval case study on trademark images.[15] According to Jain and Vailaya, the database used was created by scanning a large number of trademarks from several books at 75 dpi. In our research, we used a subset of this 1100-image database, as well as some logos collected over the Internet, for a total of 500 images. The reason for using a subset of those images is that we are primarily concerned with shape-based image retrieval, and therefore, we confined the experiments to images whose main characteristic is shape. Image with texture nature were automatically removed from the analysis based on their number of components and/or holes (texture images have a large number of components and/or holes).

5.1. *Invariance to rotation and scale change*

In order to test the capability of the solidity, eccentricity and extent (SCX) shape measures to describe objects regardless of their spatial orientation or scale, two tests were performed. In the first test, each of the 500 images in the database was randomly rotated and used as a query image. In the second test, the images were randomly scaled. The angles of rotation uniformly varied between 30° and 330°, whereas the scaling factor uniformly varied between ×0.25 and ×1.75. The SCX

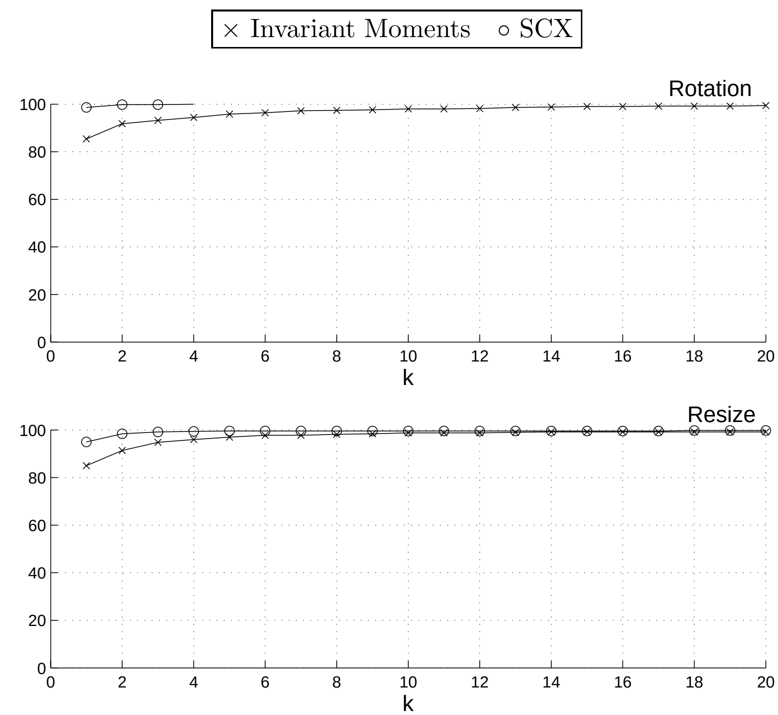


Fig. 11. Rotation and scale invariance of the SCX feature set compared with the moments invariants. The vertical axis is the percentage of the images in the datanase that were in the first k (horizontal axis) retrievals.

feature set was compared against the widely used moments invariants feature set. The seven invariant moments as well as the SCX features are extracted from the object image as a whole. That is, the proposed architecture was not used in this test because the purpose was to test the rotation and scale invariance of the feature sets used. The Euclidean distance measure is used in both cases.

Figure 11 shows two plots, the lower pertains to the scale invariance test and the upper to the rotation invariance test. In each plot, the vertical axis is the percentage of the images in the database that were in the first k (horizontal axis) retrievals. In the lower plot for example, about 97% of the 500 query images returned the correct database image as the first hit (for the SCX feature set). The plots in Figure 11 are based on the average of 20 runs.

From the plots, it is clear that for both rescaled and rotated images, the performance of the SCX feature set and the performance of the moments invariants closely match each other. In addition, both feature sets perform well in terms of being invariant to rotation and scale.

5.2. *Sketch-based retrieval*

To test the performance of the system on hand-sketched images, we used the same test procedure as in the previous section, but now using a test set of 50

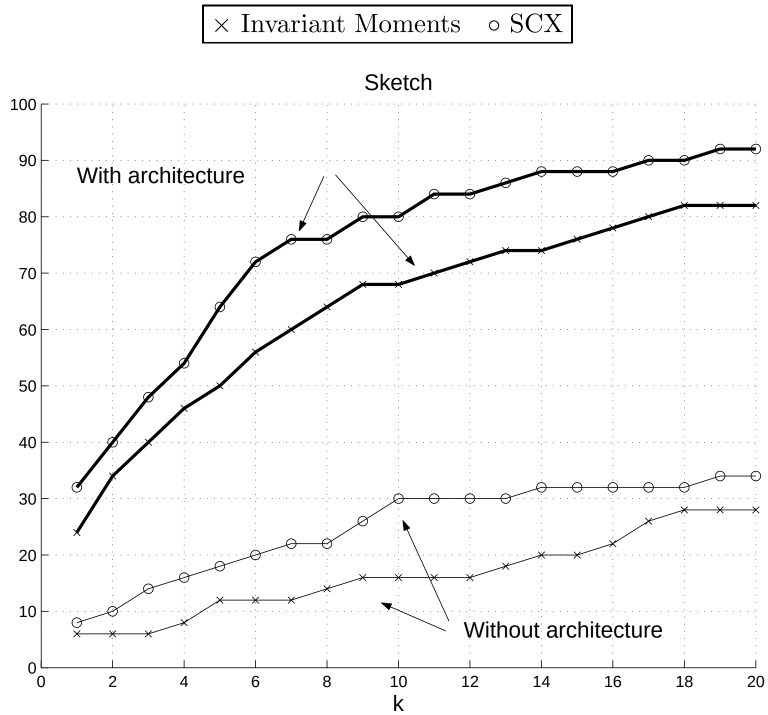


Fig. 12. Performance for sketch-based retrieval of the SCX feature set compared with the moments invariants. The vertical axis is the percentage of the images in the database that were in the first k (horizontal axis) retrievals.

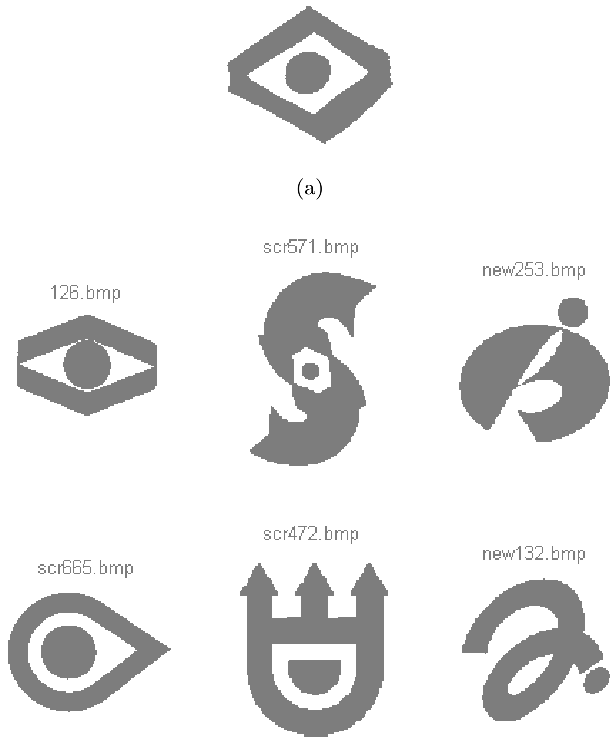
hand-sketched images. Five volunteers participated in the test, each sketching his/her own version of the selected 50 images. Moreover, now we compared the performance of the two feature sets both inside and outside the proposed architecture.

Figure 12 clearly indicates that the performance of the SCX feature set coupled with the proposed retrieval architecture yields the best results. In conclusion, the SCX feature set outperforms the moments invariants for hand-sketched retrieval and the proposed architecture significantly increases the performance of both feature sets.

The reason for the performance gap between the SCX and moments is that moments tend to be low level features that are sensitive to changes in pixel layouts. A good example of this layout difference is the one that exists between database images and their hand-sketched versions.

Figures 13 and 14 show a couple of queries and the best six retrieved images in each case. In each figure, part (a) shows the query image and part (b) shows the retrieved images (top left is the best match, top midde is the second-best match, and so forth). In Fig. 13, the first candidate returned by the system was the one sought. In Fig. 14, on the other hand, the sought image was the third candidate returned by the system. However, the first and second candidates returned by the system are a good example of partial matching (where the query image matches a component of

August 1, 2002 13:54 WSPC/164-IJIG 00069



(a)

(b)

Fig. 13. Sample query (1); (a) query image and (b) best matches starting from top left and proceeding to the right.

the database image). A good example of how an erosion operation helps retrieve the correct candidate is shown in Fig. 13 where the first candidate, containing one component, was retrieved as an answer to a query image containing two components.

6. Conclusion and Future Work

In this paper, a shape-based image retrieval system was presented. Both the architecture of the system and the features used contribute to the efficacy of the system. The feature vector used is based on the solidity, eccentricity and extent values for components and holes in the image under investigation. The features used are not the only choice. One can add to, or modify, the feature vector.

We showed that the performance of the proposed architecture combined with the features used matches that of the widely-used moments invariants feature set in retrieving rotated and scaled query images, and surpasses it for hand-sketched query images.

We have assumed throughout this work that the hand-sketched query images are drawn with closed contours and that they are appropriately filled. A prospective

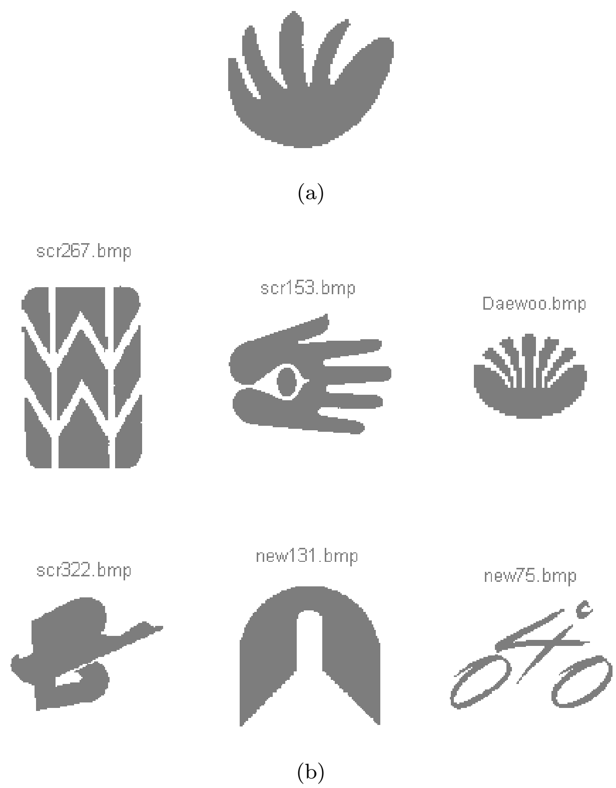


Fig. 14. Sample query (2); (a) query image and (b) best matches starting from top left and proceeding to the right.

research direction might be to look at ways of relaxing these assumptions by automatically processing the hand-sketched image to put it in a suitable form.

A natural extension of the system would be to store information not only about single image parts, but also about n parts at a time. That is, each part is analyzed on its own and features are extracted from it, then each possible two-part combination is analyzed and features are extracted from it, and so on.

In this work, we have not attempted to use suitable spatial indexing structures. However, such structures become essential when it comes to large databases.

References

1. R. C. Gonzales and R. E. Woods, *Digital image processing* (Addison-Wesley, 1992).
2. M. Safar, C. Shahabi, and X. Sun, in *Proc. Int. Conf. Multimedia and Expo*, Vol. 1 (2000), pp. 141–144.
3. M. Safar, C. Shahabi, and C. Tan, in *Proc. Int. Database Engineering and Applications Sym.* (2000), pp. 337–345.
4. G. Lu and A. Sajjanhar, *Multimedia Systems* **7**, 165–174 (1999).
5. K.-W. Hung and M. Aw-Yong, in *Proc. Int. Conf. Systems, Man and Cybernetics*, Vol. 2 (2000), pp. 1484–1488.

6. J. Xu, in *Proc. Int. Conf. Image Processing*, Vol. 2 (1999), pp. 862–866.
7. M. Adoram and M. S. Lew, in *Proc. Int. Conf. Multimedia Computing and Systems*, Vol. 2 (1999), pp. 597–602.
8. H. Kim, J. Kim, D. Sim, and D. Oh, in *Proc. Int. Conf. Multimedia and Expo*, Vol. 1 (2000), pp. 307–310.
9. S. Muller and G. Rigoll, in *Proc. Workshop Content-Based Access of Image and Video Libraries* (1999), pp. 23–37.
10. M. Flickner *et al.*, *IEEE Computer*, pp. 23–32 (September, 1995).
11. T. Kato, T. Kurita, and H. Shimogaki, *J. information processing of Japan* **12**(2), 134–143 (1991).
12. C. P. Lam, J. K. Wu, and B. Mehtre, in *Proc. 2nd Asian Conf. Computer Vision*, Vol. III (1995), pp. 214–217.
13. D. Mohamad, G. Sulong, and S. Ipson, in *Proc. 2nd Asian Conf. Computer Vision*, Vol. I (1995), pp. 439–444.
14. G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni, *Pattern Recognition* **27**(8), 1005–1018 (1994).
15. A. K. Jain and A. Vailaya, *Pattern Recognition* **31**(9), 1369–1390 (1998).
16. J. P. Eakins, J. M. Boardman, and M. E. Graham, *IEEE Multimedia*, 53–63 (April/June, 1998).
17. D. G. Kendall, *Advances in Applied Probability* **9**, 428–430 (1977).
18. Image Processing Toolbox for use with MATLAB, The Mathworks (1997).
19. R. J. Schalkoff, *Pattern Recognition*: *Statistical*, *Structural and Neural Approaches* (J. Wiley, 1992).
20. V. N. Gudivada and V. V. Raghavan, *IEEE Computer*, 18–22 (September, 1995).

Ossama El Badawy received his B.Sc. in Computer Science from Alexandria University, Egypt, in 1994. He received his Master's degree in Systems Design Engineering from the University of Waterloo, Canada, in 2000. He is currently working towards a Doctoral degree at the University of Waterloo.

Mohamed Kamel is currently professor and director of the Pattern Analysis and Machine Intelligence Laboratory at the Department of Systems Design Engineering, University of Waterloo, Canada. Prof. Kamel holds a Canada Research Chair in Cooperative Intelligent Systems. He has authored and co-authored over 170 papers in journals and conference proceedings, holds two patents and numerous technical and industrial project reports. Under his supervision, 40 Ph.D. and M.A.Sc. students have completed their degrees.

Prof. Kamel is a member of ACM, AAAI, CIPS, APEO and a senior member of IEEE, editor-in-chief of the *International Journal of Robotics and Automation*, associate editor of four international journals and has been guest editor for special issues in four Journals.

He is a member of the board of directors and co-founder of Virtek Vision International in Waterloo. He is a consultant to many companies including NCR, IBM, Nortel, VRP and CSA.