Shape Retrieval from Image Databases through Structural Feature Indexing

Hirobumi Nishida
Software Research Center, Ricoh Co., Ltd.
1-1-17 Koishikawa, Bunkyo-ku, Tokyo 112-0002, Japan
hn@src.ricoh.co.jp

Abstract

Efficient and robust information retrieval from large image databases is an essential functionality for the reuse, manipulation, and editing of multimedia documents. Structural feature indexing is a potential approach to efficient shape retrieval from large databases, but it is sensitive to noise, scales of observation, and local shape deformations. To improve the robustness, shape feature generation techniques are incorporated into structural feature indexing. The feature transformation rules obtained by an analysis of some particular types of shape deformations are exploited to generate features that can be extracted from deformed patterns. Experimental trials with large image databases of boundary contours show that the feature generation significantly improves robustness and efficiency of shape retrieval.

1 Introduction

Efficient and robust information retrieval from large image databases is an essential functionality for the reuse, manipulation, and editing of multimedia documents [6, 15]. Images have several components in terms of information representation, such as color, texture, and shape [3]. Color and texture are mathematically and physically tractable, and their properties and variations can be represented in well-structured forms by some statistical methods [17]. On the other hand, shape is another essential component, but shape analysis and representation are still difficult research subjects in spite of intensive research carried out for decades. Feature indexing techniques [1, 5] are potential approaches to improving efficiency in shape classification and retrieval. However, they are known to be sensitive to noise and shape deformations, and their performance in terms of classification accuracy is degraded drastically even due to small changes of shapes [4].

Efficient and robust retrieval from large image databases by shape [8] is a challenging problem, and shape retrieval has been studied recently for improving efficiency and robustness. For structural organization of large image databases composed of boundary contours of objects, Del Bimbo [2] and Mokhtarian *et al.* [9, 10] apply the curvature scale-space approach to feature indexing, and Sclaroff [18] proposes a method for image indexing with the modal matching [20]. Furthermore, the structural indexing by Stein and Medioni [19] copes with noise and local shape deformations by extracting shape features from several versions of polygonal approximations of boundary contours. However, there are some technical questions against these approaches. The curvature scale-space method requires a large amount of computations for smoothing boundary contours with a number of different support sizes. Sclaroff's method requires the user to specify prototype shape sets spanning the shape space adequately, but this operation is obviously difficult for end-users. Stein-Medioni's method degrades the efficiency by generating a number of polygonal approximation from the boundary contours.

Efficiency and robustness are important, but sometimes incompatible criteria for performance evaluation. The improvement of robustness implies that the retrieval should tolerate certain types of variations and deformations for images. Obviously, it may lead to inefficiency if some brute-force methods are employed such as generating various images with a number of different parameters. A key to achieving both efficiency and robustness is through a compact and well-structured representation of images that tolerate variations and deformations.

In this paper, an efficient, robust method is presented for shape retrieval from image databases composed of boundary contours of objects. The method is mainly based on an indexing technique for structural features, along with a voting technique for ranking model images in terms of extracted features from the query image. In particular, shape feature generation techniques are incorporated into structural indexing to improve the accuracy and robustness of shape classification against noise and local shape transformations [14].

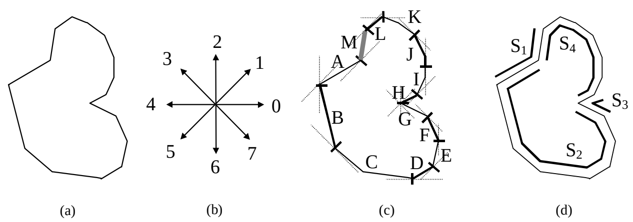

Figure 1. (a) a closed contour with a polygonal approximation, (b) quantized-directional codes when $N = 4$, (c) sub-segments when $N = 4$, (d) segments when $N = 4$.

The design of algorithms and data structures proceeds in the following steps:

(1) Representation: Based on convex/concave structures incorporating quantized-directional features along boundary contours, a compact shape representation with simple, efficient computation is explored so that the contours can be described by a few components with rich features.

(2) Feature transformations: Features based on convex/concave structures are transformed by noise, scales of observations, and local shape deformations. Therefore, to cope with such deformations, an analysis of feature transformations is carried out with respect to some particular types of shape deformations, leading to transformation rules of structural features composed of a small number of distinct cases. Features that can be extracted from deformed patterns caused by noise and local shape deformations are generated by applying the transformation rules to the features extracted from the image patterns.

(3) Model database organization by structural indexing: For efficient manipulation of a large number of models, a large table is constructed for a model set by assigning a table address to a feature and storing there a list of the model identifiers with the corresponding feature. The generated features by the transformation rules are also used to cope with feature transformations due to noise, scales of observation, and local shape deformations.

(4) Retrieval by voting for models: In the retrieval, from the features extracted from the query image, features are also generated by the transformation rules. Model identifier lists are retrieved from the table addresses corresponding to the generated features, and voting is carried out for each model on the lists. The query image is classified efficiently by selecting out some models according to scores based on the number of votes.

This paper is organized as follows: In Section 2, a structural representation of curves by quasi-convex/concave features along with quantized-directional features [11, 12] is outlined. In Section 3, we describe the transformation rules of structural features to generate features that can be extracted from deformed patterns caused by noise and local shape deformations. In Section 4, we describe a shape retrieval system based on the proposed method for structural indexing with feature generation models. Furthermore, the system is demonstrated with large image databases, and the proposed method is validated by systematically designed experiments with a large number of synthetic data. Section 5 is the conclusion.

2 Structural representation of closed contours

The structural representation of closed contours [11, 12] is outlined in this section, based on quasi-convex/concave structures along contours incorporating $2N$ quantized-directional features (N is a natural number). As shown in Fig. 1a, the closed contour is first approximated by a polygon. On a 2-D plane, we introduce N -axes together with $2N$ quantized-direction codes. For instance, when $N = 4$, eight quantized-directions are defined along with the four axes as shown in Fig. 1b. Based on these N -axes together with $2N$ quantized-direction codes, the analysis is carried out hierarchically.

A curve is decomposed into *sub-segments* at extremal points along each of the N -axes. Fig. 1c illustrates the decomposition of a contour shown in Fig. 1a into sub-segments when $N = 4$. For adjacent sub-segments *a* and *b*, suppose that we turn counterclockwise when traversing them from *a* to *b*, and the joint of *a* and *b* is an extremal point along the axes toward the directions $\left(j, j+1 \left(\mathrm{mod}\ 2N\right), \ldots, k\right)$. Then, we write the concatenation of these two sub-segments as $a \xrightarrow{j,k} b$. For instance, the joint of sub-segments *H* and *G* in Fig. 1c is an extremal point along the three axes toward the directions 3, 4, and 5. Therefore, the concatenation of *H* and *G* is written as $H \xrightarrow{3,5} G$. In this way, we obtain the following concatenations for the sub-segments illustrated in Fig 1c.

$$A \xrightarrow{3,4} B,\ B \xrightarrow{5,5} C,\ C \xrightarrow{6,6} D,\ D \xrightarrow{7,7} E,$$
$$E \xrightarrow{0,0} F,\ F \xrightarrow{1,1} G,\ H \xrightarrow{3,5} G,\ H \xrightarrow{7,7} I,$$
$$I \xrightarrow{0,0} J,\ J \xrightarrow{1,1} K,\ K \xrightarrow{2,2} L,\ L \xrightarrow{3,3} M,$$
$$A \xrightarrow{7,7} M$$

By linking local features around joints of adjacent sub-segments, some sequences of the following form can be constructed:

Feature	Model Identifier List
...	...
(3, 4, 2, 3, 1, 3, 2, 1, 1, 1)	3, 78, 346, 897
(3, 4, 2, 3, 1, 3, 2, 1, 1, 2)	89, 298, 485, 837, 917
...	...
(5, 6, 2, 3, 1, 3, 2, 1, 1, 1)	19, 289, 283, 584, 739, 937, 997
...	...

Figure 2. Model database organization by structural indexing. Each table item stores a model identifier list with the segment feature corresponding to the table index.

$$a_0 \xrightarrow{j(1,0),\, j(1,1)} a_1 \xrightarrow{j(2,0),\, j(2,1)} \cdots \xrightarrow{j(n,0),\, j(n,1)} a_n \quad (1)$$

A part of the contour corresponding to a sequence of this form is called a *segment*. Furthermore, the starting point of the segment is defined as the end point of a_0, and the ending point is as the end point of a_n. When a segment is traversed from its starting point to its ending point, one turns counterclockwise around any joints of sub-segments. The following segments, as shown in Fig. 1d, are generated from the 13 sub-segments shown in Fig. 1c:

$$S_1 : A \xrightarrow{7,7} M,$$

$$S_2 : A \xrightarrow{3,4} B \xrightarrow{5,5} C \xrightarrow{6,6} D \xrightarrow{7,7} E \xrightarrow{0,0} F \xrightarrow{1,1} G,$$

$$S_3 : H \xrightarrow{3,5} G,$$

$$S_4 : H \xrightarrow{7,7} I \xrightarrow{0,0} J \xrightarrow{1,1} K \xrightarrow{2,2} L \xrightarrow{3,3} M.$$

A segment is characterized by a pair of integers $\langle r, d \rangle$, *characteristic numbers*, representing the angular span of the segment and the direction of the first sub-segment:

$$r = \sum_{i=1}^{n} \big(j(i,1) - j(i,0) \big)_{\bmod 2N} + \sum_{i=1}^{n-1} \big(j(i+1,0) - j(i,1) \big)_{\bmod 2N} + 2, \qquad (2)$$
$$d = j(1,0)$$

The characteristic numbers are given by $\langle 2,7 \rangle$, $\langle 8,3 \rangle$, $\langle 4,3 \rangle$, and $\langle 6,7 \rangle$, respectively, for the four segments shown in Fig. 1d.

Based on the coordinate system defined by the bounding box of the contour (the upright rectangle just enclosing the shape) such that its center is located at $(0.5, 0.5)$ and the length of its longer side is 1, each segment is associated with eight parameters describing its size and position: location of its starting point (x_S, y_S), location of its ending point (x_E, y_E), location of the center (x_C, y_C) and size (W, H) of its bounding box. Furthermore, in the structural indexing and voting processes, these eight parameters are quantized into L intervals, treated as integers 0 through $L-1$. Therefore, features of a segment are described by ten integers:

$$\begin{array}{l}(r, d, \lfloor L \cdot x_S \rfloor, \lfloor L \cdot y_S \rfloor, \lfloor L \cdot x_E \rfloor, \lfloor L \cdot y_E \rfloor, \\ \quad \lfloor L \cdot x_C \rfloor, \lfloor L \cdot y_C \rfloor, \lfloor L \cdot W \rfloor, \lfloor L \cdot H \rfloor)\end{array}. \qquad (3)$$

Adjacent segments are connected by sharing the first sub-segments or last ones of the corresponding sequences. These two types of connection are denoted by $S \overset{h}{-} T$ and $S \overset{t}{-} T$, respectively, for two adjacent segments S and T. For instance, connections are denoted by $S_1 \overset{h}{-} S_2 \overset{t}{-} S_3 \overset{h}{-} S_4 \overset{t}{-} S_1$ for the four segments shown in Fig. 1d.

In the sequel, we assume that segments are indexed sequentially so that the interior of the image pattern or object lies on the left side.

3 Structural indexing with feature generation models

Features of each segment extracted from the contour curve are described by 10 integers. A large table, as illustrated in Fig. 2, is constructed for a model set by assigning a table address to a feature and storing there a list of the model identifiers with the corresponding feature. Furthermore, classification of the query image is carried out by voting for each model on the lists stored at the table address corresponding to each segment feature.

However, the features are sensitive to noise and local shape deformations, and therefore, the correct model does not necessarily receive many votes as expected for the ideal case. Furthermore, when only one sample pattern is available for each class, statistical learning techniques from training data cannot be employed for obtaining *a priori* knowledge and feature distributions of deformed patterns. To cope with these problems, we analyze the feature transformations caused by some particular types of shape deformations, constructing feature transformation rules. Based on the rules, we generate segment features that can be extracted from deformed patterns caused by noise and local shape deformations. In both processes of model database organization and retrieval, the generated features by the transformation rules are used for structural indexing and voting, as well as the features actually extracted from contours.

The following three types of feature transformations are considered in this work:

(1) Change of convex/concave structures caused by

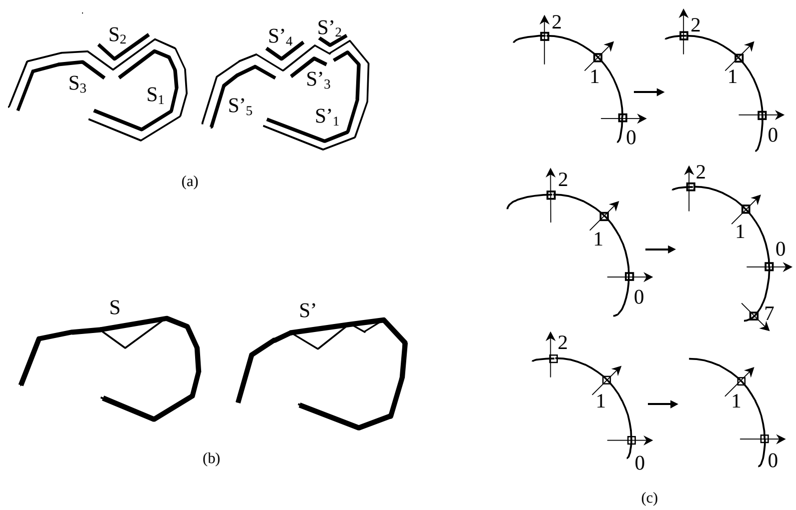

Figure 3. (a) Part of contours similar to one another in terms of global scales, (b) editing structural features by merging segment blocks, (c) transformations of characteristic numbers of segments by small rotations.

perturbations along normal directions on the contour and scales of observation, along with transformations of characteristic numbers (the angular span of the segment and the direction of the first sub-segment).

(2) Transformations of characteristic numbers caused by small rotations.

(3) Transformations of size and location parameters due to noise and local deformations.

We describe these three types of transformation in the rest of this section.

3.1 Transformations of convex/concave structures

The convex/concave structures along the contour are changed by noise and local deformations, and also depend on scales of observations. For instance, two parts of contours shown in Fig. 3a are similar to one another in terms of global scales, but their structural features are different. When $N = 4$, the curve shown on left is composed of three segments connected as $S_1 \overset{t}{-} S_2 \overset{h}{-} S_3$ with characteristic numbers $\langle 6,6 \rangle$, $\langle 2,6 \rangle$, and $\langle 3,2 \rangle$, whereas the one shown on right is composed of five segments connected as $S'_1 \overset{t}{-} S'_2 \overset{h}{-} S'_3 \overset{t}{-} S'_4 \overset{h}{-} S'_5$ with characteristic numbers $\langle 6,6 \rangle$, $\langle 2,6 \rangle$, $\langle 2,2 \rangle$, $\langle 2,6 \rangle$, and $\langle 3,2 \rangle$. To cope with such deformations, structural features on the two contours are edited so that their features can become similar to one another. For instance, the structural features illustrated in Fig. 3a can be edited by merging the two segment blocks $\{S_1, S_2, S_3\}$ and $\{S'_1, S'_2, S'_3, S'_4, S'_5\}$ to segments *S* and *S'* with the characteristic number $\langle 7,6 \rangle$ as shown Fig. 3b.

In general, rules can be introduced for generating characteristic numbers from a segment block (a set of consecutive segments).

RULE 1: From a segment block, a characteristic number is generated according to the following rules:

(1) From a segment block $\left\{S_i \middle| i = 1,2,\ldots,n; S_1 \overset{h}{-} S_2 \overset{t}{-} \cdots \overset{t}{-} S_n\right\}$, where n is odd, with characteristic numbers $\langle r_i, d_i \rangle$, a characteristic number $\left\langle \sum_{i=1}^{n} (-1)^{i+1} r_i, d_n \right\rangle$ is generated if $r_{2i-1} - r_{2i} + r_{2i+1} \geq 2$ for $i = 1,2,\ldots,\lfloor n/2 \rfloor$ and $\sum_{i=1}^{n} (-1)^{i+1} r_i \geq 2$.

(2) From a segment block $\left\{S_i \middle| i = 1,2,\ldots,n; S_1 \overset{t}{-} S_2 \overset{h}{-} \cdots \overset{h}{-} S_n\right\}$, where n is odd, with characteristic numbers $\langle r_i, d_i \rangle$, a characteristic number $\left\langle \sum_{i=1}^{n} (-1)^{i+1} r_i, d_1 \right\rangle$ is generated if $r_{2i-1} - r_{2i} + r_{2i+1} \geq 2$ for $i = 1,2,\ldots,\lfloor n/2 \rfloor$ and $\sum_{i=1}^{n} (-1)^{i+1} r_i \geq 2$.

These rules can be introduced from some mathematical properties mentioned in Nishida [11, 12]. In the structural indexing and voting processes, for an integer M specifying the maximum number of segments to be merged, characteristic numbers are generated by applying RULE 1 to consecutive n segments ($n = 1,3,\ldots, M$).

3.2 Transformations of characteristic numbers by small rotations

The characteristic number $\langle r, d \rangle$ ($r \geq 2$) can be transformed by rotating the shape. Rules can be introduced for generating characteristic numbers by rotating the shape slightly (see Fig. 3c).

RULE 2: By applying a small rotation to the segment, the characteristic number $\langle r, d \rangle$ can be transformed into one of the following: (1) $\langle r,d\rangle$, (2) $\langle r+1,d-1\rangle$, (3) $\langle r+1,d\rangle$, (4) $\langle r-1,d\rangle$ ($r\geq 3$), (5) $\langle r-1,d+1\rangle$ ($r\geq 3$).

For instance, when $N=4$ and $M=3$, six characteristic numbers $\langle 2,7\rangle$, $\langle 8,3\rangle$, $\langle 10,3\rangle$, $\langle 4,3\rangle$, $\langle 6,7\rangle$, and $\langle 12,7\rangle$ are generated from RULE 1 from the four segments illustrated in Fig. 1d with characteristic numbers $\langle 2,7\rangle$, $\langle 8,3\rangle$, $\langle 4,3\rangle$, and $\langle 6,7\rangle$. Then, in total, 28 characteristic numbers $\langle 2,7\rangle$, $\langle 3,6\rangle$, $\langle 3,7\rangle$, $\langle 8,3\rangle$, $\langle 9,2\rangle$, $\langle 9,3\rangle$, $\langle 7,3\rangle$, $\langle 7,4\rangle$, $\langle 10,3\rangle$, $\langle 11,2\rangle$, $\langle 11,3\rangle$, $\langle 9,3\rangle$, $\langle 9,4\rangle$, $\langle 4,3\rangle$, $\langle 5,2\rangle$, $\langle 5,3\rangle$, $\langle 3,3\rangle$, $\langle 3,4\rangle$, $\langle 6,7\rangle$, $\langle 7,6\rangle$, $\langle 7,7\rangle$, $\langle 5,7\rangle$, $\langle 5,8\rangle$, $\langle 12,7\rangle$, $\langle 13,6\rangle$, $\langle 13,7\rangle$, $\langle 11,7\rangle$, $\langle 11,8\rangle$ are further generated by applying RULE 2 to these generated ones.

3.3 Transformation of size and location parameters

The size and location parameters of a segment, namely $(x_{S,}y_S)$, $(x_{E,}y_E)$, (W,H), and $(x_{C,}y_C)$, are also changed by local deformations and noise. In the structural indexing and voting processes, each parameter in $\{x_S, y_S, x_E, y_E, x_C, y_C, W, H\}$ is quantized into L intervals (L is a positive integer), and therefore, we need to take into consideration quantization errors of these parameters along with local shape deformations and noise.

We introduce rules for generating quantized values of the size and location parameters.

RULE 3: Let p be one of the parameters $\{x_S, y_S, x_E, y_E, x_C, y_C, W, H\}$ for a segment, and α be a parameter $0\leq\alpha\leq 1$.

(1) If $i\leq pL\leq i+\alpha/2$ ($0<i<L$), then integers i and $i-1$ are generated as quantized values of p.

(2) If $i+1-\alpha/2\leq pL<i+1$ ($0\leq i<L-1$), then integers i and $i+1$ are generated as quantized values of p.

(3) Otherwise, an integer i is generated as a quantized value of p.

From each segment, at most $5\cdot 2^8\cdot\lceil M/2\rceil$ features (tuples of 10 integers) can be generated by RULE 1, 2, and 3. Furthermore, if we assume that value of parameter p is distributed uniformly over the interval $[0,1]$, $O\left((1+\alpha)^8\cdot Mm\right)$ features, on average, are generated from the contour composed of m segments.

4 Shape retrieval system

A system for shape retrieval from image databases was developed based on the proposed method for structural feature indexing with feature generation models. Some experimental trials were carried out to validate the effectiveness of the proposed method for shape retrieval.

4.1 Outline of the system

In the model database organization by the structural indexing, a feature-model table as shown in Fig. 2 is constructed by computing shape features in a coordinate system specific to each model image. Features are generated from each model pattern by Rules 1—3, and the model identifier is appended to the list stored at the table address corresponding to each generated feature.

In the retrieval by the voting process, from segment features extracted from the query image, features are generated by Rules 1 and 2. Model identifier lists are retrieved from the tables by using the addresses computed from the generated features, and voting is carried out for each model on the lists. Since shape features computed by the proposed method depend on the orientation of images, rotated images of the query image are considered for every $2\pi/32$ degree. Furthermore, mirror images are also created for each rotated image. The 64 images obtained from the query image in this way are treated independently in the voting process by preparing 64 voting boxes per model. The parameters used in the shape feature computation and the feature indexing are set as follows: $N=8$ (16 quantized directions), $M=9$, $L=5$, $\alpha=0.2$.

We now describe the computation of the similarity between the query image and model images. For the i-th model, let c_i be the number of features generated from segment features by Rules 1 and 2. For instance, $c_i=28$ for the contour shown in Fig. 1a when $N=4$ and $M=3$. For describing similarity between the model image i and the rotated/mirrored query image j ($j=0,1,\ldots,63$), the voting score is defined as $s_{ij}=v_{ij}/c_i$, where v_{ij} is the number of votes for the voting box (i,j). In order to take into account the complexity and global properties of images, we also consider the following quantities:

- Similarity of complexity: $\min(c_i/c_0, c_0/c_i)$, where c_0 is the number of generated features from the query image computed in the same way as c_i.
- Similarity of area: $\min(A_i/A_0, A_0/A_i)$, where A_i, A_0 are areas of the model image i and the query

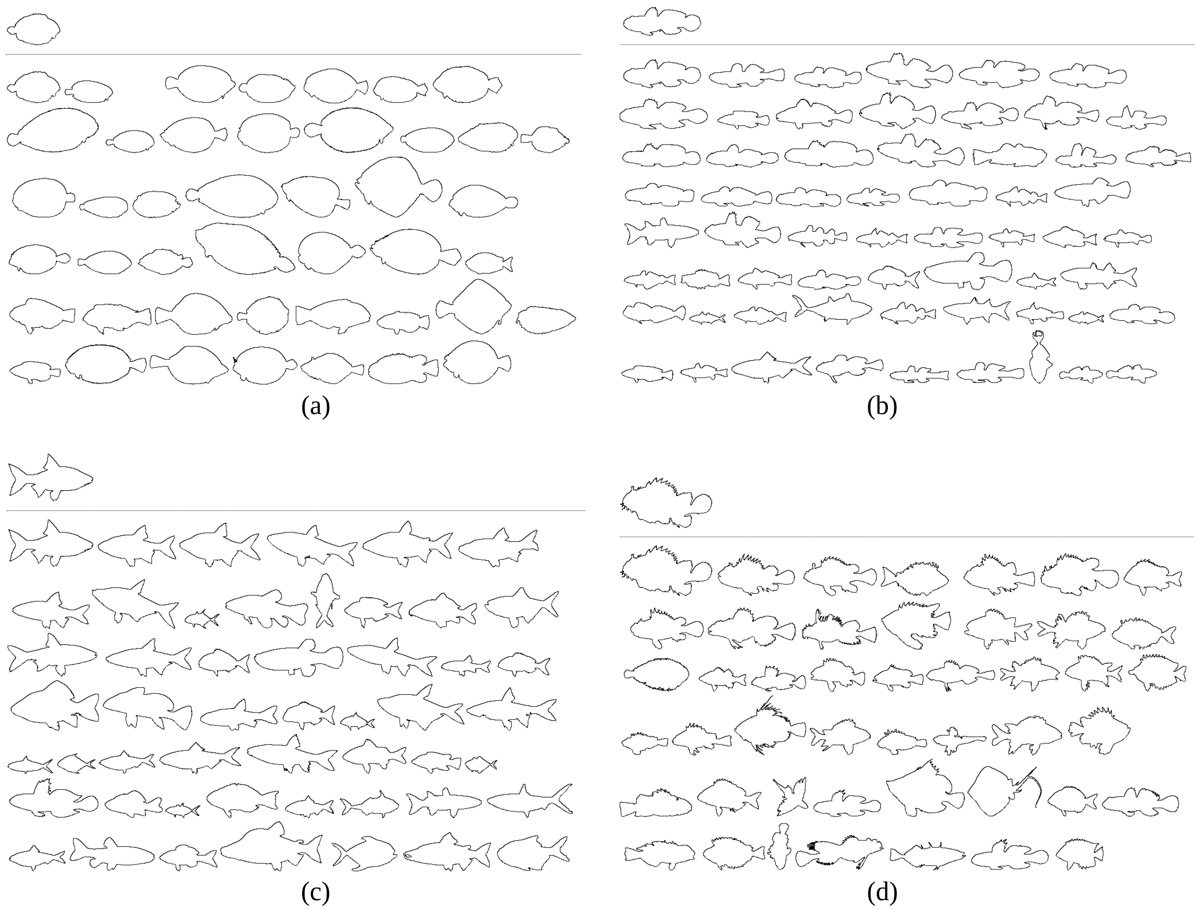

Figure 4. Examples of shape retrieval from the shape database of marine creature images. The query shape is shown at the top, and retrieved shapes are arranged in the descending order of the scores from the top-left.

image, respectively.

- Similarity of length: $\min(L_i/L_0, L_0/L_i)$, where L_i, L_0 are lengths of boundary contours for the model image i and the query image, respectively.
- Similarity of thinness ratio: $\min(T_i/T_0, T_0/T_i)$, where $T_i = 4\pi A_i / L_i^2$, $T_0 = 4\pi A_0 / L_0^2$.

The similarity S_{ij} between the model image i and the rotated/mirrored query image j is defined as the product of these four quantities and s_{ij}. Furthermore, the score Σ_i for ranking the model image i with respect to the query image is defined as $\Sigma_i = \max_j S_{ij}$.

4.2 Examples

The shape feature database was constructed for the image data set publicly available through the www site

`http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html`

from the VSSP Center of the University of Surrey, UK [9, 10]. The data set is composed of 1100 boundary contours of marine creature images, originally scanned from some printed books. Some examples of shape retrieval are presented in Fig. 4. In each figure, the query image is shown at the top, with model images arranged from top-left in the descending order of assigned scores. One query for shape retrieval from the database composed of 1100 model images takes about 1 second with an implementation with C programming language (without optimization) on Sun Sparc Ultra 2. Furthermore, the model construction process, which is fully automated, takes only a few minutes for the image data set composed of 1100 boundary contours. The size of the feature-model table is 6.7Mbytes.

The system was also tested with the shape database composed of 400 boundary contours of plant leaves [7]. Some examples of shape retrieval are presented in Fig. 5.

4.3 Quantitative evaluation

In this section, the proposed algorithm described in Section 3 is evaluated statistically in terms of the

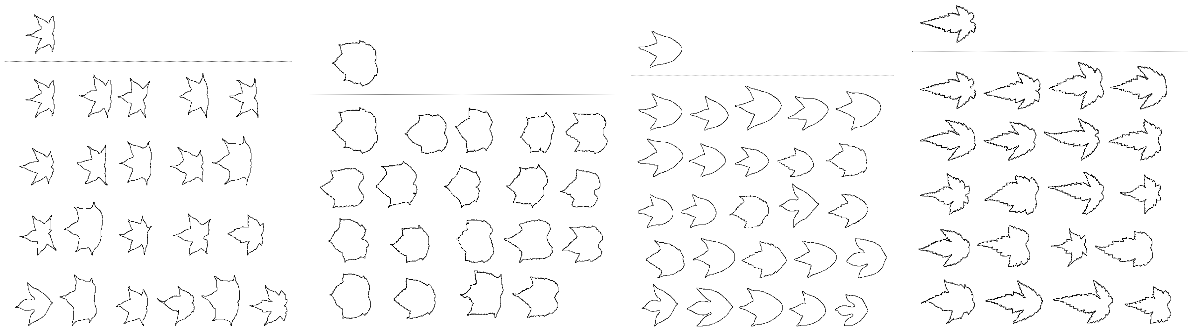

Figure 5. Examples of shape retrieval from the shape database of plant leaf images.

robustness against noise and shape deformations, based on the systematically designed, controlled experiments with a large number of synthetic data. The experimental design is composed of the generation of model patterns and their deformations to be used as testing samples. A modification of the midpoint displacement algorithm based on fractional Brownian motion is employed together with affine transformations for the model pattern generation [13]. From each model pattern expressed as a polygon, a number of testing samples are generated by applying small rotations and random perturbations along the contours. The deformation process for a model pattern is composed of the following two steps:

(1) For a vertex *P* of the polygon, let *A* and *B* be its two adjacent vertices, and *M* be the midpoint of line segment *AB*. For a given parameter β, move point *P* by $r(-\beta,\beta)\cdot\overline{PM}$ along the line passing through *P* and *M*. This operation is applied to all vertices on the polygon. (The function $r(-\beta,\beta)$ returns a real, random number between $-\beta$ and β.)

(2) After the operation (1), the polygon is rotated by angle transformed θ ($-\pi/(2N)\le\theta<\pi/(2N)$).

The main contribution of this work is to incorporate the shape feature generation into the structural indexing for coping with shape deformations and feature transformations. Therefore, for comparison, we adapted Stein-Medioni method [19] to the model database organization and classification, extracting segment features from several versions of polygonal approximations of the shape contour with a variety of error tolerances for approximations. By changing the error tolerance for polygonal approximation of contours with Ramer's method [16] from 1% to 10%, with a step of 1%, of the widest side of the bounding box of the contour, ten versions of polygonal approximations were created for each model image and the query image.

Classification rates are presented for top 1% choices, top 4%choices, and top 10% choices, in Table 1. For instance, for 1000 models, correct models are included in top 40 choices with probability 98.3% for proposed algorithm, when $\beta\in[1.5,2.0]$. Clearly, significant improvements can be observed for the proposed method in terms of classification accuracy and processing time. Therefore, the effectiveness has been verified through the experiments for the shape feature generation models along with the shape representation.

5 Conclusion

We have presented an efficient, robust method for shape retrieval from image databases based on an indexing technique for structural features in terms of convex/concave parts and quantized directional features along contours. In particular, to improve the accuracy and robustness of shape retrieval against noise and local shape transformations, shape feature generation techniques have been incorporated into structural indexing. The feature transformation rules obtained by an analysis of some particular types of shape deformations are exploited to generate features that can be extracted from deformed patterns. The generated features are used in model database organization with feature indexing and retrieval with a voting technique. Experimental trials with large image databases of boundary contours have shown that the shape feature generation significantly improves the robustness and efficiency of shape retrieval.

References

[1] A. Califano and R. Mohan, "Multidimensional indexing for recognizing visual shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 373—392, 1994.

[2] A. Del Bimbo and P. Pala, "Image indexing using shape-

Table 1. Classification rates of deformed patterns by the proposed algorithm, with the level of noise and local shape deformations described by β, in comparison with an adaptation of Stein-Medioni method.

#models	β	Proposed Method				Stein-Medioini			
		Top 1%	Top 4%	Top 10%	Time (ms/sample)	Top 1%	Top 4%	Top 10%	Time (ms/sample)
100	0.0—0.5	96.4	99.7	99.9	8.6	84.1	95.8	97.6	30.9
	0.5—1.0	93.4	99.3	99.8		79.5	93.8	96.4	
	1.0—1.5	84.1	98.1	99.5		69.8	88.7	92.9	
	1.5—2.0	77.3	96.7	98.9		61.3	84.0	90.0	
500	0.0—0.5	98.8	99.9	100.0	11.5	90.2	95.6	97.5	31.7
	0.5—1.0	97.7	99.7	100.0		86.6	93.6	96.4	
	1.0—1.5	94.0	99.0	99.9		78.6	89.3	93.8	
	1.5—2.0	90.6	98.1	99.7		70.3	84.4	90.9	
1000	0.0—0.5	99.0	99.9	100.0	17.2	90.9	95.8	97.6	36.0
	0.5—1.0	98.0	99.7	100.0		87.4	93.9	96.6	
	1.0—1.5	94.7	99.1	99.9		79.6	89.7	94.0	
	1.5—2.0	91.5	98.3	99.7		71.5	84.9	91.1	

based visual features," *Proceedings of 13th International Conference on Pattern Recognition* (Vienna, Austria), August 1996, vol. C, pp. 351—355.

[3] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC System," *Computer*, vol. 28, no. 9, pp. 23—32, 1995.

[4] W.E.L. Grimson and D.P. Huttenlocher, "On the sensitivity of geometric hashing," *Proceedings of 3rd International Conference on Computer Vision*, (Osaka, Japan), December 1990, pp. 334—338.

[5] W.I. Grosky and R. Mehrotra, "Index-based object recognition in pictorial data management," *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 416—436, 1990.

[6] V.N. Gudivana and V.V. Raghavan, "Content-based image retrieval systems," *Computer*, vol. 28, no. 9, pp. 18—21, 1995.

[7] C. Im, H. Nishida, and T.L. Kunii, "Recognizing plant species by leaf shapes - a case study of the Acer family," *Proceedings of 14th International Conference on Pattern Recognition* (Brisbane, Australia), August 16-20, 1998, pp. 1171—1173.

[8] R. Mehrotra and J.E. Gary, "Similar-shape retrieval in shape data management," *Computer*, vol. 28, no. 9, pp. 57—62, 1995.

[9] F. Mokhtarian, S. Abbasi, and J. Kittler, "Robust and Efficient Shape Indexing through Curvature Scale Space," *Proceedings of 6th British Machine Vision Conference, BMVC '96* (Edinburgh, UK), September 1996, pp. 53—62.

[10] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and Robust Retrieval by Shape Content through Curvature Scale Space," *Proceedings of 1st International Workshop on Image Database and Multimedia Search* (Amsterdam, The Netherlands), August 1996, pp. 35—42.

[11] H. Nishida, "Curve description based on directional features and quasi-convexity/concavity," *Pattern Recognition*, vol. 28, no. 7, pp. 1045—1051, 1995.

[12] H. Nishida, "A structural model of curve deformation by discontinuous transformations," *Graphical Models and Image Processing*, vol. 58, no. 2, pp. 164—179, 1996.

[13] H. Nishida, "Matching and recognition of deformed closed contours based on structural transformation models," *Pattern Recognition*, vol. 31, no. 10, pp. 1557—1571, 1998.

[14] H. Nishida, "Structural shape indexing with feature generation models," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 121—136, 1999.

[15] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233—254, 1996.

[16] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol. 1, pp. 244—256, 1972.

[17] E. Saber, A.M. Tekalp, R. Eschbach, and K. Knox, "Automatic image annotation using adaptive color classification," *Graphical Models and Image Processing*, vol. 58, no. 2, pp. 115—126, 1996.

[18] S. Sclaroff, "Deformable prototypes for encoding shape categories in image databases," *Pattern Recognition*, vol. 30, no. 4, pp. 627—641, 1997.

[19] F. Stein and G. Medioni, "Structural indexing: efficient 2-D object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 12, pp. 1198—1204, 1992.

[20] S. Sclaroff and A.P. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545—561, 1995.