Multimedia Tools and Applications, 11, 235–255, 2000

A Two-Stage Framework for Polygon Retrieval

LUN HSING TUNG lhtung@cse.cuhk.edu.hk
IRWIN KING king@cse.cuhk.edu.hk
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, People's Republic of China

Abstract. We propose a two-stage framework for polygon retrieval which incorporates both qualitative and quantitative measures of polygons in the first and second stage respectively. The first stage uses Binary Shape Descriptor as a mean to prune the search space. The second stage uses any available polygon matching and similarity measuring technique to compare model polygons with the target polygon. This two-stage framework uses a combination of model-driven approach and data-driven approach. It is more efficient than model-driven approach since it reduces the number of polygons needed to be compared. By using binary string as index, it also avoids the difficulty and inefficiency of manipulating complex multi-dimensional index structure. This two-stage framework can be incorporated into image database systems for providing query-by-shape facility. We also propose two similarity measures for polygons, namely Multi-Resolution Area Matching and Minimum Circular Error Bound, which can be used in the second stage of the two-stage framework. We compare these two techniques with the Hausdorff Distance method and the Normalized Coordinate System method. Our experiments show that Multi-Resolution Area Matching technique is more efficient than the two methods and Minimum Circular Error Bound technique produces better polygon similarity measure than the two methods.

Keywords: polygon matching, image database, content-based retrieval

1. Introduction

Query-by-shape is a fundamental operation in an image database system. It provides an intuitive way to access objects by their outlines. In this paper, we concentrate on polygonal shapes instead of arbitrary shapes. At present stage, our work only handles simple non-degenerate polygons.

The task of a shape query is to find out the set of shapes, out of a set of model shapes, that are similar to or match a given target shape (figure 1). Two kinds of shape queries, namely *matching queries* and *similarity queries*, are considered in this paper. The definitions of these two kind of queries can be found in Section 2.

Background

Considerable work has been carried out on shape matching problem. Leu [9] computes shape moments from boundary pixels of shapes and measures the similarity of shapes by comparing their shape moments. Cox et al. [6] measure the similarity of two shapes by computing the correspondence and relative position of the two shapes' convex hulls such the value of an objective function is minimized. Jagadish [8] represents shapes by rectangle covers and similarity between covers are used as similarity between their associated shapes.

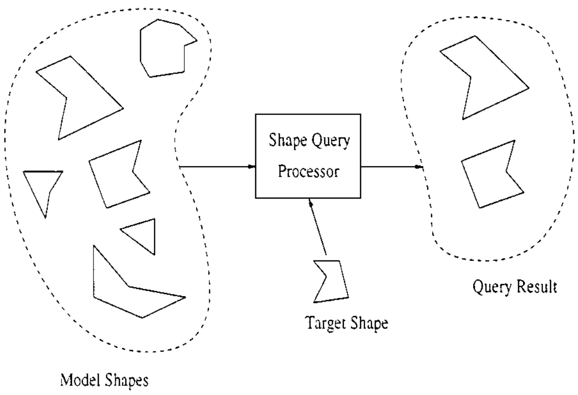


Figure 1. Query-by-shape.

By growing grows template shapes inside a target shapes, Chuang [5] uses the template shape with the largest growth for shape matching. Mehrotra and Gray [11, 12] normalize the coordinates of vertices of polygonal shapes and measure similarity of polygonal shapes by the Euclidean Distance of two coordinate lists. Moreover, the QBIC group from IBM [3, 13] incorporate many techniques for measuring shape similarity which include area, circularity, eccentricity, major axis orientation, algebraic moment invariants, and sketch matching.

Traditional techniques use model-driven approach in which the target shape is compared individually against each model shape. This approach is inefficient because of its linear searching complexity. Other techniques use data-driven approach in which features of shapes are extracted and mapped into a multi-dimensional index structure. Matching is conducted by performing searching in the index tree. The efficiency of data-driven approach highly depends on the efficiency of the Point Access Method (PAM, a data structure that support storage and retrieval of points in a multi-dimensional space) used. Roughly speaking, a large number of dimension is required when mapping shape features into multi-dimensional index structures and PAMs are inefficient under this situation. Moreover, when using PAM, additional computation is required to maintain the complex multi-dimensional index structure whenever the database is modified.

A two-stage framework

We propose a two-stage framework for the polygon matching and retrieval task using a combination of model-driven approach and data-driven approach:

1. The first stage of this two-stage framework maps polygons into binary string using the Binary Shape Descriptor (BSD) [4] technique and uses these binary string as an index. Using this index as a filtering function, a subset of polygons are selected for second stage matching.
2. The second stage of the framework then incorporates any available polygon matching and similarity measuring technique to perform matching on the subset of polygon in a traditional model-driven manner.

This two-stage framework is more efficient than model-driven approach since it reduces the number of polygons needed to be compared. It also avoids the difficulty and inefficiency of manipulating complex multi-dimensional index structures. Instead, it uses string as index which is well studied and efficient indexing techniques are available.

Two Polygon similarity measures

We propose two new polygon similarity measuring techniques. The Multi-Resolution Area Matching (MRAM) is an area based matching technique incorporating Quadtree [14] area coding. Its multi-resolution nature makes it possible to further speed up the query processing task under the two-stage framework. We propose another polygon matching technique using Circular Error Bound (CEB) which is based on an intuitive human concept of polygon resemblance. This technique can only be used to determine whether two polygons resemble each other under certain transformations and user specified tolerance. The idea of CEB method is extended and a polygon similarity measure named Minimum Circular Error Bound (MCEB) is developed which gives translation invariant measure on the similarity between two polygons.

Application

The result of our work can be incorporated into image database systems for providing query-by-shape facility. Figures 2 and 3 show an example of shape query in an sample image database. Figure 2 shows 8 fashion images with their associated model polygons.

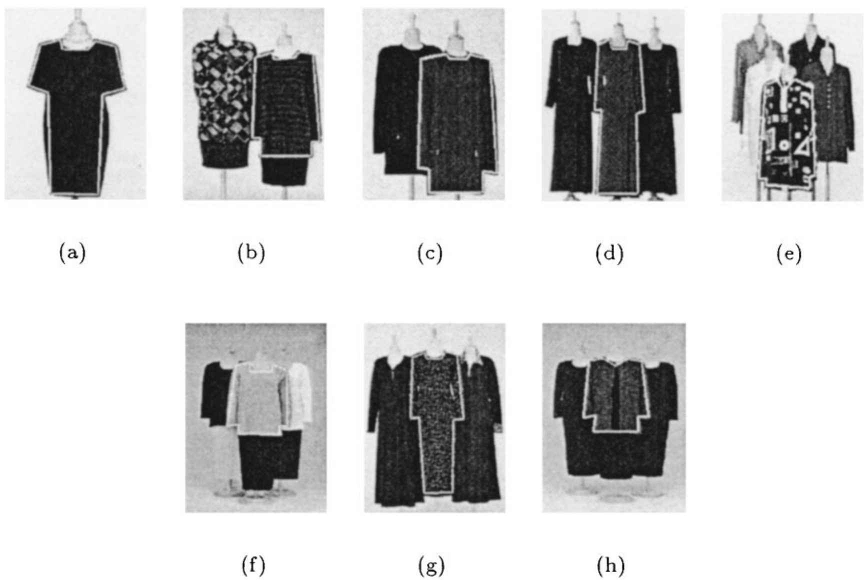


Figure 2. Sample image database.

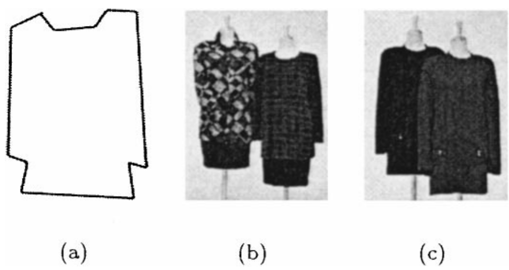


Figure 3. Sample shape query and its results.

Figure 3(a) shows the target polygon of the shape query while figure 3(b) and (c) are the two images that contain model polygons most similar to the target polygon.

While robust automatic segmentation technique and shape similarity measure are still difficult to achieve, it is possible to write applications using practical approaches. As illustrated by previous examples, we show an application that: (1) segments the desired object semi-automatically which requires user intervention in defining shapes contained in images, and (2) represents and matches shape by polygonal approximation. Our two-stage framework fits well for such applications.

The two-stage framework we proposed is implemented in the *Montage* image database system [10] which is currently under development at the Chinese University of Hong Kong. The *Montage* system is an image database system designed and implemented for the fashion, textile, and clothing industry in Hong Kong. It supports feature based retrieval by color histogram, color sketch, shape, and texture.

Paper organization

This paper is organized as follows. Several basic definitions for our discussion are presented in Section 2. We describe the idea of BSD and the computation of Standardized Binary Shape Descriptor (SBSD) in Section 3, which provides a basis for the discussion of the two-stage framework presented in Section 4.2. MRAM and MCEB are described in Section 5 and Section 6 respectively. We present and discuss the experimental result in Section 7. Discussion and possible extension to the two-stage framework are presented in Section 8. Conclusion is made in Section 9.

2. Definitions

Definition 1. A polygon is represented by an ordered list of vertices $P = \{V_1, V_2, \ldots, V_n\}$, where n is the number of vertices of the polygon and $V_i \in \mathbb{R}^2$.

Definition 2. A polygon is simple if no two edges of the polygon cross each other.

Definition 3. A polygon is non-degenerate if $\nexists 1 \leq i \leq n$ such that V_i, V_{i+1}, V_{i+2} are collinear, where $V_{n+1} = V_1$ and $V_{n+2} = V_2$.

Definition 4. A *matching query* is

$$\boldsymbol{R} = \{P_i \mid P_i \in \boldsymbol{P} \wedge \ MATCH(P_i, T)\}$$

where $\boldsymbol{R}$ is the result of the query, $\boldsymbol{P}$ is the set of model polygons, T is the target polygon, and $MATCH(\cdot)$ denotes a polygon matching technique.

Definition 5. A *similarity query* is

$$\boldsymbol{R} = \{P_i \mid P_i \in \boldsymbol{P} \ \wedge \ 1 \leq i \leq n \ \wedge \ P_1 \leq P_2 \leq \cdots \leq P_m\}$$

where $\boldsymbol{R}$ is the result of the query, $\boldsymbol{P}$ is the set of model polygons, n is the number of polygons to be included in $\boldsymbol{R}$, m is the number of polygons in $\boldsymbol{P}$, $n \leq m$, and $P_1 \leq P_2 \leq \cdots \leq P_m$ is the ranking produced by a polygon similarity measuring technique based on the degree of similarity between the model polygons and the target polygon.

3. Binary shape descriptor

We start by introducing the idea of BSD technique, which serves as a polygon classification method in our two-stage framework.

3.1. Basic idea

BSD is a binary string recording the convexities and concavities of the vertices of a polygon. Let '0' denote a convex vertex (interior angle less than π) and '1' denote a concave vertex (interior angle larger than π).

Definition 6. A Binary String Descriptor (BSD) is a string $\{0, 1\}^n$, where n is the number of vertices of the polygon the descriptor is associated with.

BSD is scale and orientation invariant since the measurement of convexity and concavity of a vertex is independent of these properties. However, the specific instance of the BSD of a polygon depends on the selection of the *anchor vertex* (the vertex of the polygon at which we start recording the BSD).

3.2. Standardized binary string descriptor

A polygon can be represented by more than one BSD depending on the sequence of vertices being recorded. For example, a polygon represented by BSD '0010' can also be represented by '0100', '1000' or '0001', depending on the anchor vertex. The idea of standardized BSD is introduced in [4] in order to obtain a unique BSD for a given polygon.

Given a BSD $B = \{0, 1\}^n$, a rotated BSD B_i, for $1 \leq i \leq n$, is another BSD generated by rotating the bits of B such that the ith Most Significant Bit (MSB) of B becomes the MSB of B_i. Let $M(B_i)$ denotes the magnitude of B_i regarding it as a binary integer.

Table 1. n-gons and number of their equivalent classes.

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E	1	2	4	9	15	30	54	101	181	343	624	1173	2183	4106

Definition 7. The Standardized Binary String Descriptor (SBSD) of B is B_j such that $M(B_j) = \min_i M(B_i)$, $1 \leq i \leq n$.

SBSD inherits the scale and orientation invariant properties from BSD and it is independent of the selection of anchor vertex.

3.3. *Number of equivalent classes for n-gons*

SBSD function is a many-to-one mapping, i.e., more than one polygon may have the same SBSD. Polygons having the same SBSD are said to be in the same equivalent class. For polygons with n sides, there are 2^n possible BSDs. However, some of them are invalid. For example, ‘00111’ is invalid since a simple polygon should have at least three convex vertices, thus its BSD should have at least three ‘0’s. Some BSDs are the same after standardization, for example, ‘00011’, ‘01100’ and ‘10001’. For n-gons, the number of equivalent classes (E) is given in [4] as

$$E = \frac{1}{n} \sum_{m \in D_n} m X_n(m) - \left(\left\lfloor \frac{n}{2} \right\rfloor + 2 \right)$$

where D_n is the set of divisors of n,

$$X_n(m) = 2^{n/m} - (X_n(m_1) + \cdots + X_n(m_k))$$

and $m_1, \ldots, m_k$ are the multiples of m belonging to $D_n \backslash \{m\}$.

Table 1 shows the number of equivalent classes for polygons with sides from 3 to 16, where n is the number of polygon vertices and E is the number of equivalent classes.

4. The two-stage framework

4.1. *Database population*

Before we describe the two-stage matching framework, we will first describe how the model polygons are normalized when their associated images are inserted into image databases.

1. *Model polygon acquisition*: Model polygons inside an image can be extracted by automatic segmentation technique. However, as robust segmentation technique is still not available, it is more practical to require user intervention in this process.
2. *Orientation normalization*: The model polygon is rotated such that its longest edge is aligned with the y-axis. In case there are more than one edges having the largest length, their successive edges are recursively examined until the tie situation is broken.

3. *Scale normalization*: After orientation normalization, the model polygon is scale such that it has a unit bounding box.

After the preprocessing, the model polygons and links to their associated images are inserted into image databases. By retrieving these model polygons through shape queries, we can then follow the links and retrieve the images.

4.2. The two-stage framework

Our polygon retrieval framework consists of two stages. The first stage incorporates the BSD technique as a mean to prune the search space. The main idea is to partition polygons into groups according to their SBSDs. Instead of comparing every model polygon with the target polygon, only those within the same equivalent class as the target polygon are selected and compared. Since SBSDs are just binary strings, they can be easily indexed so the filtering process in the first stage can be implemented efficiently.

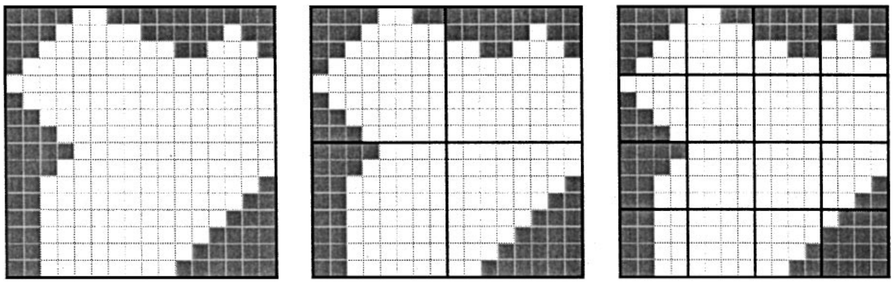
The second stage of the framework incorporates any available polygon matching and similarity measuring techniques. For handling matching queries, the technique incorporated only needs to determine whether two given polygons match each other under user specified tolerance. However, in order to handle similarity queries, the technique incorporated must be able to produce a ranking on the set of model polygons according to the degree of similarity between these polygons and the target polygons. To accomplish this task, the polygon similarity measuring technique incorporated should be able to produce a numerical value as the similarity measure between two polygon. The ranking can then be produced by sorting the model polygons based on their similarity measure to the target polygon.

Putting the two stages together, the framework should look like:

1. $\boldsymbol{Q} = \{P_i \mid P_i \in \boldsymbol{P} \wedge SBSD(P_i) = SBSD(T)\}$ where $\boldsymbol{P}$ is the set of model polygons in a database, $SBSD(\cdot)$ denotes the SBSD function, and T is the target polygon.
2. If the query is an matching query, execute (A). If it is a similarity query, execute (B).
   (A) $\boldsymbol{R} = \{Q_i \mid Q_i \in \boldsymbol{Q} \wedge MATCH(Q_i, T)\}$ where $MATCH(\cdot)$ denotes the polygon matching method selected.
   (B) $\boldsymbol{R} = \{Q_i \mid Q_i \in \boldsymbol{Q} \wedge 1 \leq i \leq n\}$ where n is the number of model polygons to be included in the answer to a query, m is the number of model polygons in a database, $n \leq m$, $\boldsymbol{Q} = \{Q_1, Q_2, \ldots, Q_m\}$, and $Q_1 \leq Q_2 \leq \cdots \leq Q_m$, which is the ranking produced by the polygon similarity measuring technique selected.
3. $\boldsymbol{R}$ is the set of model polygons which is the answer to the query.

5. Multi-resolution area matching

In this section, we propose a multi-resolution area based polygon similarity measuring technique. We describe how the Multi-Resolution Area Information (MRAI) is computed and how similarity between polygons is measured using MRAI.



(a) (b) (c)

Figure 4. Computing MRAI. The maximum resolution computed in this example is 2. (a), (b), and (c) show the frame buffer at resolution level 0, 1, and 2 respectively. The MRAI at resolution level 0 is $\langle \frac{180}{256} \rangle$. The MRAI at resolution level 1 is $\langle \frac{46}{64}, \frac{46}{64}, \frac{45}{64}, \frac{43}{64} \rangle$. The MRAI at resolution level 2 is $\langle \frac{6}{16}, \frac{14}{16}, \frac{6}{16}, \frac{8}{16}, \frac{10}{16}, \frac{16}{16}, \frac{16}{16}, \frac{16}{16}, \frac{5}{16}, \frac{16}{16}, \frac{16}{16}, \frac{13}{16}, \frac{8}{16}, \frac{16}{16}, \frac{13}{16}, \frac{1}{16} \rangle$.

5.1. Computing MRAI

After a model polygon is preprocessed as described in Section 4.1, it is scan-converted onto a frame buffer with $W \times W$ pixels. MRAI is then computed using a Quadtree like approach:

1. MRAI is recorded starting at level 0.
2. At level 0, the whole frame buffer is regarded as a cell. The portion of area covered by the polygon is recorded.
3. At level k, cells are obtained by quartering every cell of level $k-1$. The portion of area covered by the polygon in each level-k cell is recorded. There are 4^k cells at level k.

The MRAI at each level is concatenated to form a complete MRAI vector. The size of this vector depends on K, the maximum resolution level to be recorded, and is given as $L = \sum_{i=0}^{K} 4^i = \frac{4^{K+1}-1}{3}$. In our implementation, $W = 64$ and $K = 3$. Figure 4 shows an example of computing MRAI in which the maximum resolution level is 2.

5.2. Measuring similarity using MRAI

We use the L_p distance to measure the similarity of two polygons at a specific level of resolution. Given polygon A and B, with their MRAI, the similarity of these two polygons at resolution level k is:

$$S_k(A, B) = \left(\sum_{i=1}^{4^k} |A_{ki} - B_{ki}|^p \right)^{1/p}$$

where $S_k(A, B)$ is the similarity measure of A and B at resolution level k, A_{ki} and B_{ki} are the portion of covered area in level k cells of polygon A and B respectively, and $p = 2$ in our implementation.

Matching of two polygons can be done in levels, that is, perform similarity measuring from coarse resolution (level 0) to fine resolution (the maximum resolution level K).

Definition 8. Two polygons A and B are said to be matched at level k if $S_k(A, B) \leq \delta_k$ where δ_k is a predefined threshold value for level k similarity measure.

Definition 9. Two polygons are said to be matched if they are matched at all levels, i.e. the two polygons are similar at level $0, \ldots, K$.

5.3. *Query processing using MRAM*

Shape query processing can be further speeded up by taking advantage on MRAM's multi-resolution characteristic. We describe the procedures for handling matching queries and similarity queries using MRAM. Note that the model polygons mentioned in the following two sessions are those selected by the first stage algorithm of the two-stage framework according to the target polygon.

Matching query. Since level 0 MRAI is just a real number, it can serve as a database index for the model polygons. Matching queries are processed as follows.

1. Only model polygons having level 0 MRAI in the range $[t_0 - \delta_0, t_0 + \delta_0]$ are fetched for further matching where t_0 is the level 0 MRAI of the target polygon and δ_0 is the predefined threshold for level 0 matching as stated in Section 5.2. Since model polygons are indexed on level 0 MRAI, this subset of polygons can be retrieved efficiently.
2. Model polygons selected in the previous step are then compared with the target polygon individually using the approach described in Section 5.2.

Similarity query. Similarity queries are processed using following algorithm

$\boldsymbol{Q} \leftarrow$ model polygons
for $i = 0$ to K do
    sort $\boldsymbol{Q}$ in descending order of $S_i(Q_i, T)$ where $Q_i \in \boldsymbol{Q}$
    $\boldsymbol{Q} \leftarrow \{Q_i \mid Q_i \in \boldsymbol{Q} \wedge 1 \leq i \leq n_i\}$
end for

where T is the target polygon and $S_i(\cdot)$ is the level i similarity measure of two polygons as stated in Section 5.2. In our implementation, n_0, n_1, n_2 and n_3 are 100, 50, 25 and 10 respectively.

6. Minimum circular error bound

In this session, we describe the polygon matching technique using CEB and the new polygon similarity measuring technique named MCEB.

6.1. Polygon matching using circular error bound

The polygon matching technique using CEB is based on an intuitive human definition of polygon resemblance. The intuitive definition of similar polygons is as follows. If two polygons are matched, then each vertex of one polygon is close to its corresponding vertex of another polygon when the two polygons are overlapped. The correspondence between vertices is an one-to-one mapping. Therefore, the definition and the technique we proposed only work on polygons that have the same number of vertices. Before the two polygons are overlapped, translation, scaling and rotation are allowed to be performed on the polygons.

Definition 10. A transformation T is a vector, i.e. $T = \langle t_x, t_y, s_x, s_y, \theta\rangle$ where t_x is translation in x-axis direction, t_y is translation in y-axis direction, s_x is the scaling in x-axis direction, s_y is the scaling in y-axis direction and θ is the rotation about the origin. $T(Q)$ denotes the object obtained by applying T to Q where Q may be a polygon or a vertex.

Definition 11. Given a tolerance vector $E = \langle\epsilon_1, \epsilon_2, \ldots, \epsilon_n\rangle$, $Q = \{U_1, U_2, \ldots, U_n\}$ is said to be matched with $P = \{V_1, V_2, \ldots, V_n\}$ if there exists a transformation T such that $Q' = T(Q) = \{U_1', U_2', \ldots, U_n'\}$ and $\forall_{1\le i\le n} \|V_i - U_i'\| \le \epsilon_i$, where $\|\cdot\|$ denotes the Euclidean norm (figure 5).

Definition 12. Given V_i, ϵ_i and U_i, the ith *Circular Error Bound* (*CEB*), C_i, is a circle with ϵ_i as its radius and $(V_i - U_i)$ as its center.

Note that Definition 11 assumes we already know the pairing of vertices between the two polygons, i.e., V_i should match U_i.

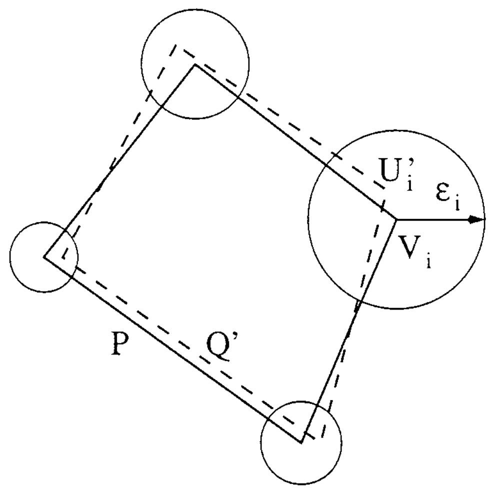


Figure 5. Polygon matching under Definition 9.

The polygon matching task using CEB is formulated as follows:

> "Given two polygons P and Q with a tolerance vector E, the task is to determine whether a transformation T exists such that Q is said to be matched with P under Definition 11."

By Definition 11, the transformation T is an arbitrary vector $\langle t_x, t_y, s_x, s_y, \theta\rangle$. However, in nowadays applications, the transformations in polygon matching task are often restricted to some special cases, for example, translation and (or) scaling only. With restricted transformations, we have efficient solutions for the polygon matching task. In the following sections, we will present the solution for the polygon matching task when (1) only translations are allowed, (2) only translations and uniform scaling in x-axis and y-axis directions are allowed, and (3) only translations and independent scaling in x-axis and y-axis directions are allowed.

6.1.1. Translation. Assume that the transformation T in Definition 11 is restricted to $T = \langle t_x, t_y, 1, 1, 0\rangle$ only.

Proposition 1. *Given $P = \{V_1, V_2, \ldots, V_n\}$, $Q = \{U_1, U_2, \ldots, U_n\}$ and $E = \langle \epsilon_1, \epsilon_2, \ldots, \epsilon_n\rangle$, if the n Circular Error Bounds $C_1, C_2, \ldots, C_n$ of P and Q have common intersection, then Q is matched with P.*

Proof: Assuming $V_i = (a_i, b_i)$ and $U_i = (c_i, d_i)$, by Definition 12, Circular Error Bound C_i is a circle with ϵ_i as its radius and $(a_i - c_i, b_i - d_i)$ as its center. If $C_1, C_2, \ldots, C_n$ have common intersection, then for any point (t_x, t_y) in the common intersection, the distance between this point and the center of any C_i is less than or equal to the radius of C_i. Figure 6

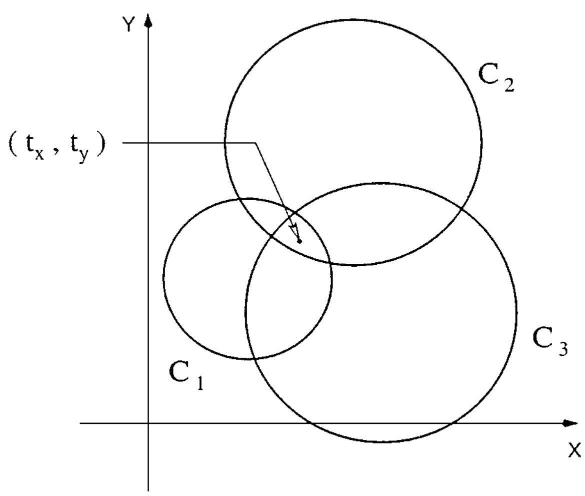


Figure 6. Intersection of circular error bounds.

illustrates this idea when both P and Q are triangles. Thus, $\forall i$, $1 \leq i \leq n$,

$$\sqrt{[(a_i - c_i) - t_x]^2 + [(b_i - d_i) - t_y]^2} \leq \epsilon_i \tag{1}$$

Re-arranging Eq. (1), we have

$$\sqrt{[a_i - (c_i + t_x)]^2 + [b_i - (d_i + t_y)]^2} \leq \epsilon_i \tag{2}$$

which is equivalent to $\|V_i - U_i'\| \leq \epsilon_i$ where $U_i' = T(U_i)$ and $T = \langle t_x, t_y, 1, 1, 0\rangle$. By Definition 11, Q is matched with P. □

6.1.2. Translation and uniform scaling in x-axis and y-axis directions. Assume that the transformation T in Definition 11 is restricted to $T = \langle t_x, t_y, s, s, 0\rangle$, i.e. only translation and uniform scaling in x-axis and y-axis directions are allowed.

Let $U_i = (c_i, d_i)$ and apply the scaling transformation $S = \langle 0, 0, s, s, 0\rangle$ to Q, we have $U_i' = S(U_i) = (sc_i, sd_i)$. Thus, Circular Error Bound C_i of P and Q', where $V_i = (a_i, b_i)$, is a circle with ϵ_i as its radius and $(a_i - sc_i, b_i - sd_i)$ as its center.

Two Circular Error Bounds C_i and C_j intersect each other if and only if

$$\sqrt{[(a_i - sc_i) - (a_j - sc_j)]^2 + [(b_i - sd_i) - (b_j - sd_j)]^2} \leq \epsilon_i + \epsilon_j \tag{3}$$

Re-arranging Eq. (3), we have

$$\begin{aligned}&[(c_i - c_j)^2 + (d_i - d_j)^2]s^2 - 2[(a_i - a_j)(c_i - c_j) + (b_i - b_j)(d_i - d_j)]s\\ &\quad + [(c_i - c_j)^2 + (d_i - d_j)^2 - (\epsilon_i + \epsilon_j)^2] \leq 0\end{aligned} \tag{4}$$

Solving Eq. (4), we get a range, $\mathbb{S}_{ij}$, for s that the inequality holds (figure 7).

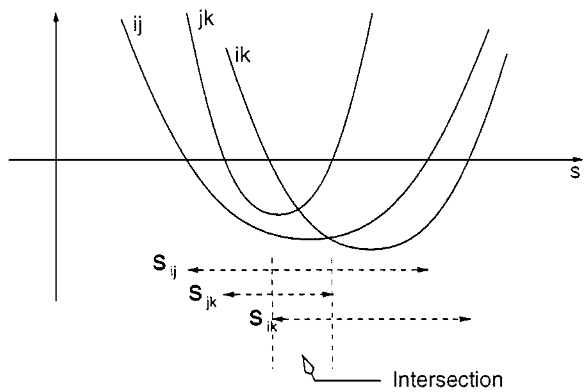


Figure 7. $\mathbb{S}_{ij}$ and its intersection.

Proposition 2. *If* $\cap_{1\le i,j\le n}\mathbb{S}_{ij} \neq \emptyset$, *then* Q *is matched with* P.

Proof: If $\cap_{1\le i,j\le n}\mathbb{S}_{ij} \neq \emptyset$, then $\exists S = \langle 0, 0, s, s, 0\rangle \in \cap_{1\le i,j,\le n}\mathbb{S}_{ij}$ such that Circular Error Bounds $C_1, C_2, \ldots, C_n$ of P and Q' have common intersection, where $Q' = S(Q)$. By Proposition 1, Q' is matched with P. Thus, $\exists T = \langle t_x, t_y, 1, 1, 0\rangle$ such that $\forall_{1\le i\le n}\|V_i - U_i''\| \le \epsilon_i$ where $U_i'' = T(U_i')$. Therefore, $\exists T' = T \circ S = \langle t_x, t_y, s, s, 0\rangle$ such that $\forall_{1\le i\le n}\|V_i - U_i''\| \le \epsilon_i$ where $U_i'' = T'(U_i)$. By Definition 11, Q is matched with P. □

6.1.3. Translation and independent scaling in x-axis and y-axis directions. Assume that the transformation T in Definition 11 is restricted to $T = \langle t_x, t_y, s_x, s_y, 0\rangle$, i.e., independent scaling in x-axis and y-axis directions as well as translation are allowed.

Let $U_i = (c_i, d_i)$ and apply the scaling transformation $S = \langle 0, 0, s_x, s_y, 0\rangle$ to Q, we have $U_i' = S(U_i) = (s_x c_i, s_y d_i)$. Thus, Circular Error Bound C_i of P and Q', where $V_i = (a_i, b_i)$, is a circle with ϵ_i as its radius and $(a_i - s_x c_i, b_i - s_y d_i)$ as its center.

Two Circular Error Bounds C_i and C_j intersect each other if and only if

$$\sqrt{[(a_i - s_x c_i) - (a_j - s_x c_j)]^2 + [(b_i - s_y d_i) - (b_j - s_y d_j)]^2} \le \epsilon_i + \epsilon_j \tag{5}$$

Re-arranging Eq. (5), we have

$$[(a_i - a_j) - (c_i - c_j)s_x]^2 + [(b_i - b_j) - (d_i - d_j)s_y]^2 \le (\epsilon_i + \epsilon_j)^2 \tag{6}$$

Eq. (6) defines an ellipse, $\mathbb{E}_{ij}$, on the s_x-s_y plane. A point (s_x, s_y) in $\mathbb{E}_{ij}$ defines a transformation $S = \langle 0, 0, s_x, s_y, 0\rangle$ such that when S is applied to Q, the Circular Error Bounds C_i and C_j, of $S(Q)$ and P, intersect each other (figure 8).

Proposition 3. *If* $\forall_{1\le i,j\le n}\mathbb{E}_{ij}$ *have common intersection*, *then* Q *is matched with* P.

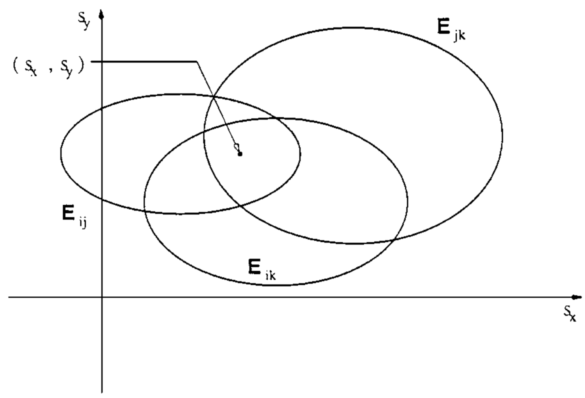


Figure 8. $\mathbb{E}_{ij}$ and its intersection.

Proof: If $\forall_{1\le i,j\le n}\mathbb{E}_{ij}$ have common intersection, then for any point (s_x, s_y) in the common intersection, Circular Error Bounds $C_1, C_2, \ldots, C_n$ of of P and Q', intersect each other, where $Q' = S(Q)$ and $S = \langle 0, 0, s_x, s_y, 0\rangle$. By Proposition 1, Q' is matched with P. Thus, $\exists T = \langle t_x, t_y, 1, 1, 0\rangle$ such that $\forall_{1\le i\le n}, |V_i - U_i''| \le \epsilon_i$ where $U_i'' = T(U_i')$. Therefore, $\exists T' = T \circ S = \langle t_x, t_y, s_x, s_y, 0\rangle$ such that $\forall_{1\le i\le n}, \|V_i - U_i''| \le \epsilon_i$ where $U_i'' = T'(U_i)$. By Definition 11, Q is matched with P. □

6.2. Minimum circular error bound

The polygon matching techniques presented in Section 6.1 only deal with queries of whether a polygon Q is matched with another polygon P subject to some tolerances (the tolerance vector E) and restrictions on transformation. Thus, these techniques can only be used to handle matching queries. By extending the idea of these techniques, we propose a similarity measure of polygons named Minimum Circular Error Bound (MCEB). Since MCEB is defined as the optimal value over all possible translations, it is a translation invariant similarity measure of polygons.

Definition 13. The Minimum Circular Error Bound, $\xi \in \mathbb{R}$, of a polygon $Q = \{U_1, U_2, \ldots, U_n\}$ comparing to another polygon $P = \{V_1, V_2, \ldots, V_n\}$ is defined as

$$\xi = \min_{\forall_{t_x,t_y} T=(t_x,t_y)} \max_{1\le i\le n} \|V_i - T(U_i)\|$$

ξ can be calculated as follows. Let $V_i = (a_i, b_i)$ and $U_i = (c_i, d_i)$. Further assume that the tolerance vector $E = \langle \epsilon_1, \epsilon_2, \ldots, \epsilon_n\rangle$ where $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_n$. The Circular Error Bound C_i is a circle with ϵ_i as its radius and $(a_i - c_i, b_i - d_i)$ as its center. If two Circular Error Bounds C_i and C_j intersect each other, we have

$$\sqrt{[(a_i - c_i) - (a_j - c_j)]^2 + [(b_i - d_i) - (b_j - d_j)]^2} \le \epsilon_i + \epsilon_j \tag{7}$$

Since $\epsilon_i = \epsilon_j$, we denote the value of ϵ_i and ϵ_j as ϵ_{ij}. The minimal value of ϵ_{ij} that Eq. (7) holds is

$$\epsilon_{ij} = \frac{1}{2}\sqrt{[(a_i - c_i) - (a_j - c_j)]^2 + [(b_i - d_i) - (b_j - d_j)]^2}$$

The MCEB of the two polygons Q and P is

$$\xi = \max_{1\le i,j\le n} \epsilon_{ij}$$

such that for $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_n \ge \xi$, $\forall_{1\le i,j\le n}$ C_i and C_j intersect each other. That is, for $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_n \ge \xi$, Circular Error Bounds $C_1, C_2, \ldots, C_n$ of Q and P have common intersection and Q is matched with P under Proposition 1.

7. Experimental result

We compare the MRAM and MCEB techniques with the Hausdorff Distance method [2, 7] and the Normalize Coordinate System (NCS) method [11, 12]. We select these two methods for comparison since both of them measure polygon similarity based on the distance between polygon vertices, as the MCEB technique does.

Hausdorff Distance is defined as follows.

Definition 14. Given two finite point sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, the Hausdorff Distance is

$$H(A, B) = \max(h(A, B), h(B, A))$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

and $\|\cdot\|$ is some underlying norm on the points of A and B.

In our experiment, we use Euclidean norm for computing Hausdorff Distance.

The NCS method represents a polygon with an ordered list of normalized coordinates. The normalization is carried out as follows. Assume that a n-sided polygon is represented by an ordered list of n 2D points. A pair of points in the original list of coordinates is chosen to form a basis vector. A new coordinate system (the normalized one) is defined using the basis vector as a unit vector along the x-axis and each point in the original list is transformed to the new system. This process produces a list of normalized coordinates. Since the first two points of these normalized coordinates must be $(0, 0)$ and $(1, 0)$, they can be removed from the list. Thus, a n-sided polygon is represented by an ordered list of normalized coordinates with size of $n - 2$.

Table 2 summaries the main characteristics of the four methods. More detailed discussion is presented below.

Table 2. Comparison of MRAM, MCEB, Hausdorff Distance and NCS. n is number of vertices of polygons to be handled. m is the number of polygons in the database. L is size of the MRAI vectors.

	MRAM	MCEB	Hausdorff Distance	NCS
Handle polygons with different number of sides	Yes	No	Yes	No
Require preprocessing	Yes	No	No	Yes
Storage requirement	$O(L)$	$O(n)$	$O(n)$	$O(n)$
Time complexity	$O(mL)$	$O(mn^2)$	$O(mn^2)$	$O(mn)$
Require vertex correspondence	No	Yes	No	Yes
Translation invariant	No	Yes	No	No

1. MRAM and Hausdorff Distance method can handle polygons with different number of sides but MCEB and NCS method can only handle polygons with same number of sides.
2. Among the four method, MRAM and NCS require preprocessing of polygons. MRAM requires the computation of MRAI and NCS requires the normalization of coordinates, while MCEB and Hausdorff Distance method operate directly on the original polygon data.
3. MRAM requires $O(L)$ storage where L is the size of the MRAI depending on the maximum resolution level chosen in an implementation. All other three methods requires $O(n)$ storage where n is the number of sides of polygons to be processed.
4. The computational complexity of MRAM, MCEB, Hausdorff Distance and NCS are $O(L)$, $O(n^2)$, $O(n^2)$ and $O(n)$ respectively. Thus, for handling m model polygons, the running time complexity of the four methods are $O(mL)$, $O(mn^2)$, $O(mn^2)$ and $O(mn)$ respectively. Though in general L is larger than n or even n^2, the running time of MRAM method in average is less than other three methods because of its multi-resolution strategy. Table 3 shows the relative similarity query processing time of the four methods using the simple system we described in [17]. The experiments are conducted on a SunSparc workstation 10/30. Several polygon databases are used in the experiments. Each database consisted of 9000 polygons with particular number of sides. We assume an uniform distribution on the number of polygons in equivalent classes. The statistic obtained in each experiment is normalized using the time spent by the fastest method in that experiment. For example, in the experiment of 3-sided polygons, MRAM method is the fastest one so its speed index is 1. Hausdorff Distance method is the slowest one in this experiment and has a speed index of 1.67, meaning that its query processing time is 1.67 times MRAM's.
5. Both MCEB and NCS method require the knowledge of the vertex correspondence between the two polygons being compared. However, this problem can be handled by a simple heuristic. Given a polygon, we choose the first end point of the longest edge as the starting vertex. When there are more than one edge having the largest length, we simply pick one randomly. In our implementation, we pick the first one the implementation found when such situations occur. This simple strategy gives satisfactory result in empirical experiment.
6. An ideal similarity measure of polygons should be translation invariant. MCEB is the only one among the four method that has this property. Other three methods simply align

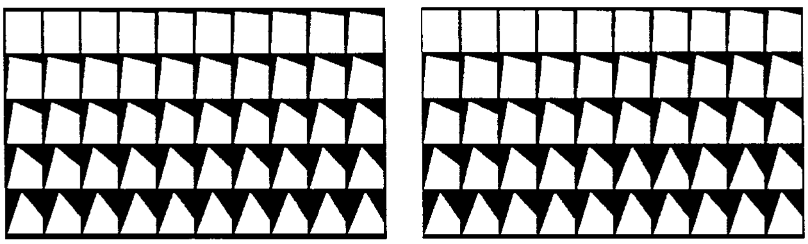
Table 3. Relative processing time for a single shape query.

n-gon	Number of equivalent classes	Total number of polygons	MRAM	MCEB	Hausdorff distance	NCS
3	1	9000	$\boxed{1.00}$	1.59	1.67	1.49
4	2	9000	$\boxed{1.00}$	1.22	1.33	1.18
5	4	9000	$\boxed{1.00}$	1.15	1.27	1.16
6	9	9000	$\boxed{1.00}$	1.09	1.14	1.06
7	15	9000	$\boxed{1.00}$	1.04	1.06	1.01
8	30	9000	$\boxed{1.00}$	1.02	1.03	$\boxed{1.00}$

the two polygons at an anchor point as the solution to this problem. In [1], the computational complexity of translation invariant Hausdorff Distance is given as $O(n^4 \log^3(n^2))$, which is much larger than MCEB's.

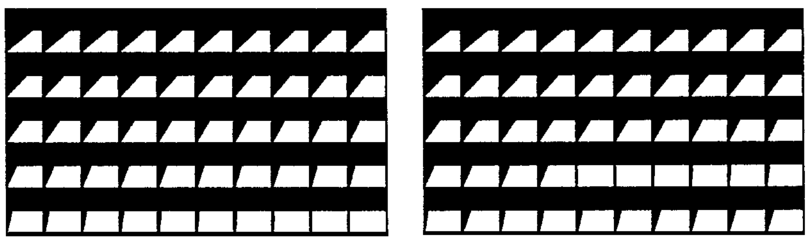
Experiments have also been carried out to examine the quality of visual ranking produces by the four methods. By 'quality of visual ranking', we mean the degree of resemblance between human polygon similarity rankings and those produced by the four methods.

1. Figures 9(a) and 10(a) show two set of polygons created by generating in-between polygons from the first and last polygon in each set. The polygons are in white color and are placed from left to right, top to bottom. Similarity rankings are performed using the four methods where the first polygon in each set serves as the target polygon. The rankings produced by the four methods are compared with the original ones and the number of mis-ranked polygons is used as a measure of the quality of the rankings, where small numbers indicate high quality rankings. MRAM, MCEB and NCS methods succeed in producing exactly the same rankings as the original ones but the Hausdorff Distance method produces rankings with quality measures of 9 and 13 for the two polygon sets respectively. Figures 9(b) and 10(b) show the rankings produced by the Hausdorff Distance method.



(a) (b)

Figure 9. Result of similar ranking experiment 1.



(a) (b)

Figure 10. Result of similar ranking experiment 2.

Figure 11. Result of similar ranking experiment 3.

2. Figure 11(a) shows the polygons of another experiment. The first polygon is used as the target polygon. In this experiment, only MCEB and Hausdorff Distance method produce polygon ranking as the original one. Both MRAM and NCS method produce a ranking different from the original one, as shown in figure 11(b).

Our experiments show that MCEB method is the only one which passes the two tests.

8. Discussion

From Table 1, we observe that the number of equivalent classes are relatively small when the polygons being handled are with small number of sides. For example, all triangles will be in one equivalent class. Thus, SBSD may not be a good method for polygon classification in these situations. A possible solution to this problem is to record the angle of a vertex in more discrete levels (rather than convex and concave only). For example, if 4 discrete levels are used ($0 < \theta \leq \frac{\pi}{2}$, $\frac{\pi}{2} < \theta \leq \pi$, $\pi < \theta \leq \frac{3\pi}{2}$ and $\frac{3\pi}{2} < \theta < 2\pi$), there will be 2 equivalent classes for triangles instead of 1. If 8 discrete levels are used, then there will be 6 equivalent classes for triangles. Thus, the number of discrete levels used to record the vertex angle can be chosen according to the number of sides of polygons a system is expected to process.

Pruning the search space using only SBSDs may have some shortcomings as illustrated in the following example. In figure 12, there are three polygons named P, Q, and R. Polygon

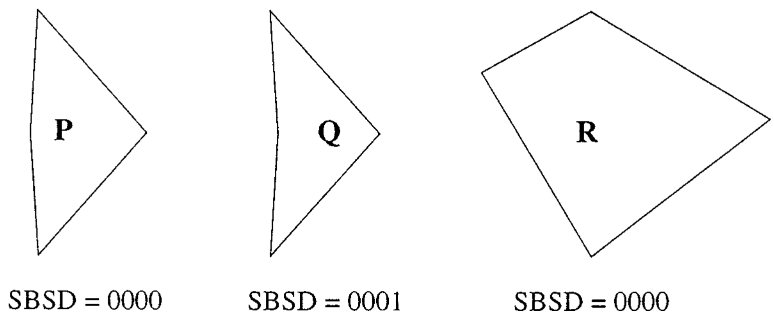


Figure 12. The problem of using only SBSD as filtering function.

P and R are in the same equivalent class according to their SBSDs while polygon Q is in another equivalent class. Thus, polygon Q will never be selected and compared with polygon P no matter how similar they are visually. In order to overcome this shortcoming, we propose that instead of selecting model polygons having exactly the same SBSD as the target polygon we now select model polygons having SBSDs within a predefined tolerance of Hamming Distance from the SBSD of the target polygon. With this modification, the first stage of the two-stage framework should look like:

1. $\boldsymbol{Q} = \{P_i \mid P_i \in \boldsymbol{P} \wedge \mathit{HAMDIST}(\mathit{SBSD}(P_i), \mathit{SBSD}(T)) \leq \delta\}$ where $\boldsymbol{Q}$ is the set of polygons selected for second stage matching, $\boldsymbol{P}$ is the set of model polygons, T is the target polygon, $\mathit{SBSD}(\cdot)$ denoted the SBSD function, $\mathit{HAMDIST}(\cdot)$ denotes the function that computes the Hamming Distance between two binary strings, and δ is the user specified Hamming Distance tolerance.

Note that the original two-stage framework is a particular instance of the modified one, with the Hamming Distance tolerance δ always set to 0. This modification provides a systematic way for controlling the degree of search space pruning by allowing user to make tradeoff between quality of query result and speed of query processing. A small tolerance produces larger pruning effect but with higher risk of producing worse query results. On the other hand, large tolerance may produce better query results but has a smaller pruning effect.

9. Conclusion

We propose a two-stage framework for the polygon retrieval task which incorporates qualitative and quantitative measures of polygon in the first stage and second stage respectively. The first stage uses SBSDs as a mean to prune the search space and reduce the number of polygons needed to be compared with the target polygon. The second stage incorporates any available polygon matching and similarity measuring technique to compare model polygons with the target polygon. This two-stage framework is more efficient than model-driven approach since it reduces the number of polygons needed to be compared. It also avoids the difficulty and inefficiency of manipulating complex multi-dimensional index structures. Instead, it uses string as index which is well studied and efficient indexing techniques are available. We also propose two polygon similarity measuring techniques named MRAM and MCEB. We compare these two techniques with the Hausdorff Distance method and NCS method. Our experiments show that MRAM technique is more efficient than the two methods and MCEB technique produces better polygon similarity ranking than the two methods.

References

1. P.K. Agarwa, M. Sharir, and S. Toledo, “Applications of parametric searching in geometric optimization,” in Proc. of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms, 1992, pp. 72–82.

2. H. Alt, B. Behrebds, and J. Blömer, "Approximate matching of polygonal shapes," Annals of Mathematics and Artificial Intelligence, Vol. 13, pp. 251–265, 1995.
3. J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic, "Automatic and semi-automatic methods for image annotation and retrieval in QBIC," SPIE, Vol. 2420, pp. 24–35, 1995.
4. B. Bhavnagri, "A method for representing shape based on an equivalent relation on polygons," Pattern Recognition, Vol. 27, No. 2, pp. 247–260, 1994.
5. J.H. Chuang, "A potential-based approach for shape matching and recognition," Pattern Recogntion, Vol. 29, No. 3, pp. 463–470, 1996.
6. P. Cox, H. Maitre, M. Minoux, and C. Ribeiro, "Optimal matching of convex polygons," Pattern Recognition Letters, Vol. 9, pp. 327–334, 1989.
7. D.P. Huttenlocher and W.J. Rucklidge, "A multi-resolution technique for comparing images using the Hausdorff distance," Department of Computer Science, Cornell University, Technical Report TR92–1328, 1992.
8. H.V. Jagadish, "A retrieval technique for similar shapes," in Proc. of the ACM SIGMOD Int. Conf. on the Management of Data, May 1991, pp. 208–217.
9. Jia Guu Leu, "Computing a shape's moments from its boundary," Pattern Recognition, Vol. 24, No. 10, pp. 949–957, 1991.
10. I. King and T.K. Lau, "A feature-bassed image retrieval database for the fashion, textile, and clothing industry in Hong Kong," in Proc. of International Symposium Multi-Technology Information Processing '96, 1996, pp. 233–240.
11. R. Mehrotra and J.E. Gray, "Feature-based retrieval of similar shapes," in Proc. 9th International Conference on Data Engineering, 1993, pp. 108–115.
12. R. Mehrotra and J.E. Gray, "Similar-shape retrieval in shape data management," IEEE Computer Magazine, pp. 57–62, Sept. 1995.
13. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker, "The QBIC project: Querying images by content using color, texture, and shape," SPIE, Vol. 1908, pp. 173–187, 1993.
14. H. Samet, Design and Analysis of Spatial Data Structure, Addison-Wesley, 1989.
15. L.H. Tung and I. King, "A two-stage framework for polygon retrieval using minimum circular error bound," in Proceedings to the 9th International Conference on Image Analysis and Processing (ICIAP'97), Vol. I, Sept. 1997, pp. 567–574. Lecture Notes in Computer Science, Vol. 1310.
16. L.H. Tung, I. King, P.F. Fung, and W.S. Lee, "A two-stage framework for efficient simple polygon retrieval in image databases," in Proc. of International Symposium Multi-Technology Information Processing '96, 1996, pp. 146–153.
17. L.H. Tung, I. King, P.F. Fung, and W.S. Lee, "Two-stage polygon representation for efficient shape retrieval in image databases," in Proc. of the 1st International Workshop on Image Databases and Multimedia Search, 1996.

Alan Tung received the B.S. and M.Phil. degree in computer science from the Chinese University of Hong Kong in 1995 and 1997 respectively. His publications include "A Two-Stage Polygon Representation for Efficient Shape Retrieval in Image Databases" in "Proc. of the 1st International Workshop on Image Databases and Multimedia

Search", "A Two-Stage Framework for Efficient Simply Polygon Retrieval in Image Databases" in "Proc. of International Symposium Multi-Technology Information Processing '96", and "A Two-Stage Framework for Polygon Retrieval Using Minimum Circular Error Bound" in "Proc. of the 9th International Conference on Image Analysis and Processing (ICIAP'97)".

Irwin King received the B.S. degree in Engineering and Applied Science from California Institute of Technology, Pasadena, in 1984. He received his M.Sc. and Ph.D. degree in Computer Science from the University of Southern California, Los Angeles, in 1988 and 1993 respectively. While working on his graduate degree, he worked at the Xerox Special Information System (XSIS) unit on the design and implementation of the "Analyst" System based on the SmallTalk environment. He joined the Department of Computer Science and Engineering at the Chinese University of Hong Kong in 1993. His research interests include multimedia systems—content-based retrieval methods for image databases; image processing—face analysis and computing; and neural networks—unsupervised learning theory for visual processing. He is a member of ACM, IEEE Computer Society, International Neural Network Society (INNS), and Asian Pacific Neural Network Assembly (APNNA).