Signal Processing: *Image Communication* 18 (2003) 141–156

SIGNAL PROCESSING:
IMAGE
COMMUNICATION

www.elsevier.com/locate/image

A shape-based image retrieval method using salient edges

Jun Wei Han*, Lei Guo

Department of Automatic Control, Northwestern Polytechnical University, Xi'an, 710072, People's Republic of China

Received 13 August 2001; received in revised form 2 May 2002; accepted 18 September 2002

Abstract

Content-based image retrieval is emerging as an important research area with applications in digital libraries and multimedia databases. In this paper, we present a novel five-stage image retrieval method based on salient edges. In the first stage, the Canny operator is performed to detect edge points. Then, the Water-Filling algorithm is employed to extract edge curves. In the third stage, salient edges are selected and the shape features in terms of the salient edges are yielded. In the fourth stage, a similarity measure, namely the integrated salient edge matching, that integrates properties of all the salient edges, is introduced, and used to compare the similarity of the query image with the images in the database. Finally, the best matches are returned in similarity order. The presented approach is easy to implement and can be efficiently applied to retrieve images with clear edges. Preliminary experimental results on a database containing 6500 images are very promising.

Keywords: Content-based image retrieval (CBIR); Salient edge; Shape; Edge curve; Integrated salient edge matching (ISEM); Image database

1. Introduction

As the amount of digital image data available on the Internet and in digital libraries is rapidly growing, there is a great need for efficient image indexing and access tools in order to fully utilize this massive digital resource. *Content-based image retrieval* (CBIR) is a research area dedicated to address this issue and substantial research efforts have been made. Thus far a number of image search systems have been developed. Among them, the IBM QBIC [6] is one of the earliest systems. Recently, other systems are implemented such as IBM T.J. Watson [28], VIRAGE [10], NEC AMORA [21] and Bell Laboratory [22], MIT Photobook [23,24], Berkeley Blobworld [2], Columbia VisualSEEK and WebSEEK [29], and Standford WBIIS [32]. However, a user friendly and yet conceptually simple image search system still demands further research effort. The current study is intended to meet the demand as far as we can.

The architecture of a typical CBIR system is shown in Fig. 1. It contains two main components: a database archive (the process of creating an image database) and a database query. During the archive, images and videos are processed to extract features describing their contents (colors, textures, shapes, and camera and object motions) and the features are then stored in a database. During the

*Corresponding author.
E-mail addresses: junweihan@263.net (J.W. Han), lguo@nwpu.edu.cn (L. Guo).

PII: S0923-5965(02)00116-9

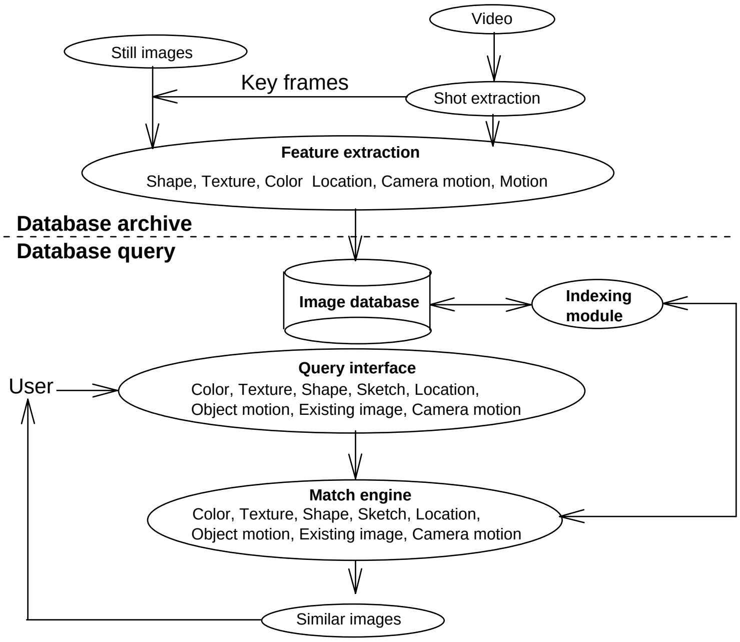
Video
Still images
Key frames
Shot extraction
Feature extraction
Shape, Texture, Color Location, Camera motion, Motion
Database archive
Database query
Image database
Indexing module
User
Query interface
Color, Texture, Shape, Sketch, Location, Object motion, Existing image, Camera motion
Match engine
Color, Texture, Shape, Sketch, Location, Object motion, Existing image, Camera motion
Similar images

Fig. 1. The architecture of a general CBIR system.

query, the user composes a query graphically. Features are generated from the graphical query and then input to a matching engine that finds images or videos from the database with similar features. In the CBIR system, similarity queries are done against the database of pre-extracted features using distance functions between the features, which is called match engine. The match engine interacts with an indexing module to speed up the search. Users interact with the query interface to generate a query specification, resulting in the features that define the query.

The rest of this paper is organized as follows. Section 2 presents a brief survey of the related work. Section 3 describes the details of the proposed method. Section 4 gives the preliminary experimental results. Finally, the conclusions are drawn in Section 5.

2. Related work

Shape features provide a powerful clue for object identification. Humans can recognize characteristic objects solely from their shapes. This distinguishes shape from other elementary visual features, such as color, texture or motion.

Recently, shape features have attracted much attention in the image retrieval research fields. As far as image retrieval is concerned, a good shape descriptor should capture characteristic shape features in a concise manner, which renders itself to fast searching and browsing. Also, it should be invariant to scaling, rotation, and translation [1,27]. Two major shape representations [1,27], namely the region-based [15,16] shape descriptor and the contour-based shape descriptor [1,3,4,8,9,11,13,19,20,23,27,30,33,34], are currently used in shape-based retrieval systems. Whilst the former uses the entire shape region to extract a meaningful description that is most useful when the objects have similar spatial distributions of pixels, the latter uses only boundary information of objects, and is suitable to describe objects that have similar contour characteristics. The proposed shape descriptor in this paper is based on boundaries.

Many contour-based image retrieval approaches are now available. In [8], shapes are represented as an ordered set of boundary features. Each boundary is coded as an ordered sequence of vertices of its polygonal approximation. Features are collections of a fixed number of vertices. This representation allows for a rough evaluation of the shape similarity that is defined as the Euclidean distance between the boundary feature vectors in the query and those associated with the target images. Del Bimbo et al. [4] introduce an image retrieval algorithm by elastic matching of shapes and image patterns. In this elastic matching approach, rectangular areas enclosing objects of interest are selected manually, the intensity gradient maps for these areas are then obtained, and a hill-climbing optimization algorithm is applied to these gradient maps to identify the boundaries of the objects. In the ART MUSEUM project [11], Hirata et al. propose a method to compute similarities of a user sketch with images in the database. In their work, color oil painting images are passed through a series of steps including size regularization, edge detection, thinning and shrinking. The resulting abstract images are then compared with the user sketch by a template matching method. Pentland et al. [23] present a shape model in terms of "interconnectedness" of shape features, e.g., edges, corners or high-curvature points. Mehrotra and Gray [19] use for shape representation a collection of a few adjacent interest points, such as maximum local curvature boundary points or vertices of the shape boundary's polygonal approximation. Gu et al. [9] attempt to apply an affine-invariant feature descriptor in the light of corners to retrieve images. Mokhtarian et al. [20], Bober [1] and Sikora [27] use a curvature scale space image (CSS), which is a multi-scale organization of the inflection points (or curvature zero crossing points) of the contour. This descriptor also includes of eccentricity and circularity values of the original and filtered contours. A CSS retrieval is used for matching and indicates the heights of the most prominent peak, and horizontal and vertical positions on the remaining peaks in the so-called CSS image. Jain and Vailaya [13] employ a histogram of the edge directions as the shape feature. Takahashi et al. [30] also use edge directions to search images. Zhou et al. [33,34] introduce edge-based structural features. They use a Water-Filling algorithm to obtain edge curves. And then, structural features such as "MaxFillingTime", "MaxForkCount" and "ForkHistogram" are extracted in accordance with the edge curves. Finally, the query image is compared with images of database by a structural matching algorithm.

In this paper, we provide an effective image retrieval algorithm using salient edges, which relies on the assumption that there are significant edges that characterize the objects in images. Also, a prototype image retrieval system has been developed in terms of the proposed algorithm. Both theoretical analysis and preliminary experimental results show that the proposed algorithm has the following prominent advantages:

1. The shape descriptors are simple and yet efficient. They are translation, rotation and scaling-invariant to a certain extent.
2. The overall similarity measure approach significantly reduces the adverse effect of inaccurate edge extraction.

3. The salient edge-based image retrieval approach

3.1. Background and motivation

Edge information is usually used for extracting object shape. In general, edge points that are defined as "special points" with sharp intensity changes convey little structural information of the image [19,33]. Therefore, they are inappropriate to represent shape features effectively. It is widely believed [3,19,27,33,1] that edge contours are suitable to describe shape features since they embody rich structural information. In [33], Zhou et al. indicate that shape extraction requires edge linking as a reprocessing step. Without edge linking, "shape" is out of the question; and features such as "edge length" or "edge smoothness" are hardly meaningful too. However, it does not mean that all edge contours are beneficial to describe shape features. In [11], a contour-based image retrieval technique is presented. Its shape descriptor relies on all edge contours of the image. However, Refs. [19,33,34] argue that this shape descriptor does not perform well with images having many extraneous edges after edge detection. In this scenario, extraneous edges only serve as noise to confuse any similarity calculating algorithm. Therefore the retrieval accuracy of this algorithm is limited. In [3], experiments suggest that when humans see the edge maps, they tend to ignore some "details" and remember only a few prominent edges. There are also evidences [19,34] indicating that humans pay more attention to salient edges of an image.

Hence, in order to improve retrieval performance, we reckon that a shape-based method, which computes the similarities between images in the light of salient edges, possibly with long edge curves, is desirable. This new idea has three good qualities. The first is that when edge pixels are grouped into edge curves, the curves that are deemed insignificant, namely the curves relatively shorter than others, can be ignored in similarity computation. These short edge curves either may be noisy patterns or tend to be ignored by humans. Secondly, in contrast to unorganized edge pixels, edge curves explicitly convey structural information of the objects inside images, making it more accurate to compute similarities between images. Finally, a reduced number of features used to characterize the image results in significant speed-up in similarity computation.

The proposed algorithm mainly includes two parts: feature extraction and similarity measure calculation. In the stage of feature extraction, three shape descriptors (fork ratio, rotation frequency and corner frequency) in accordance with salient edges are produced. Then in the latter stage, a many-to-many measure principle is used to calculate the similarities between images. Details of the algorithm will be given in the remaining sections.

Note that the provided algorithm can only be applied to images with clear edges such as building images, face images and trademark images. In this paper, we do not intend to design an algorithm that can search general images with good performance.

3.2. Feature extraction

The steps of extracting shape features are shown in Fig. 2. The edge detection and thinning step first identifies perceptually significant edge pixels. Unrelated pixels in the edge map convey little information. Consequently, the Water-Filling algorithm [33,34] is applied to obtain edge curves. Then, some shorter edge curves are eliminated and significant curves are extracted. Finally, the shape feature vectors are generated in terms of salient edges.

3.2.1. Edge detection and thinning

The *Canny* operator has both differencing and smoothing effects and the latter is useful to reduce noise in resulting edge maps. Thus, in this paper, we use it to extract and thin edge pixels.

3.2.2. Edge curve extraction using water-filling algorithm

Unorganized pixels in the edge map gotten from the previous phase convey little information about the structures of objects in the original image unless they are organized. As mentioned before, edge linking is very important for shape extraction. However, edge linking from an edge map is not an easy thing. A heuristic graph search is not globally optimal and thus runs the risk of losing vital information [33,34], whilst a globally optimal method such as dynamic programming (DP) may be computationally too expensive. Both the heuristic search algorithm and DP link edge are absolutely motivated by the direction of image understanding. The goal is to obtain "optimal" edge curves that are helpful in understanding the image content. Nevertheless, in practice, not only these over-complex "optimal" procedures cannot achieve promising performance, but also they are not easy to calculate. As matter of fact, these approaches are usually difficult to carry out in general purpose CBIR systems.

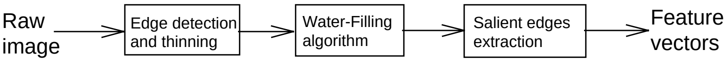

Fig. 2. The flow of the shape feature extraction process.

However, if the goal is to image retrieval and matching only, i.e., finding a representation that gives similar "numbers" whenever two images have similar content, and vice versa, the issue becomes "how to effectively and efficiently represent information embedded in the edge map?" As a consequence, in this paper, the expectation to edge linking is reduced. It only plays the roles of grouping edge pixels into edge curves and keeping the structural information of images as much as possible. Like [33,34], we utilize the Water-Filling algorithm for linking edges. It can be fulfilled much more easily and still achieve performance better than the heuristic search algorithm as demonstrated by our experiments.

The Water-Filling algorithm could be regarded as a simulation of "flooding of connected canal system". Starting from an end point, the algorithm performs a first raster scan on the edge map and fills in water at the first edge pixel encountered that has less than 2 neighbors. The waterfront then flows along the edges in the order indicated by the numbers. When there are more than one possible paths to go at one point, the waterfront will fork. The assumptions implied by the algorithm are: (1) the water supply at the starting pixel is unlimited; (2) the water flows at a constant speed in all directions; and (3) the water front will stop only at a dead-end, or at a point where all possible direction have been filled. For more details about Water-Filling algorithm, the reader is referred to [33] or [34].

3.2.3. Salient edge extraction

The target of this step is two-fold: prominent edge curves that are likely to attract humans' attention, are to be pricked off, whilst short edge curves be eliminated in order to reduce extraneous edges and noise in the edge map.

We rank the edge curves obtained from the previous stages according to the length decreasing order and then prick off the L longest edge curves as the salient edges. The set of salient edges is denoted by: $C = \{c_1, c_2, \ldots, c_L\}$, where $c_1, c_2, \ldots, c_L$ are salient edges, and ranked in the length decreasing order.

In the real-world images, many edge contours have several branches. Fig. 3(a) shows an edge curve with two branches at the fork e. In fact, it is very difficult to deal with edge curves with branches. Refs. [1,27,34] do not address the issue. Here we try to overcome it by developing a scheme of assigning the *representative edge curve*. This scheme obeys two principles: *maximal length highest priority* (MLHP) and *maximal smoothness highest priority* (MSHP).

Let the curves, c_b, c_c, c_d, displayed by Fig. 3(b)–(d), are three potential candidates for the *representative edge curve* of Fig. 3(a). By and large, edge contours of objects are smooth. In [5] Farag and Delp indicate that in local neighborhoods, edge curves do not undergo frequent changes such as from horizontal to vertical or from horizontal to diagonal. Assume that trajectory of an edge curve is $T = \{t_0, t_1, \ldots, t_n\}$, and t_{i-1}, t_i, t_{i+1} are three continuous points on the trajectory T. Fig. 4 shows an example of three continuous points within a 3×3 neighborhood. As we can see from the Fig. 4, when positions of t_{i-1} and t_i are fixed, the point t_{i+1} has seven possible extension directions that are denoted as $t^0_{i+1}, \ldots, t^6_{i+1}$. We use the *point smoothness* (PS) to denote the smoothness of

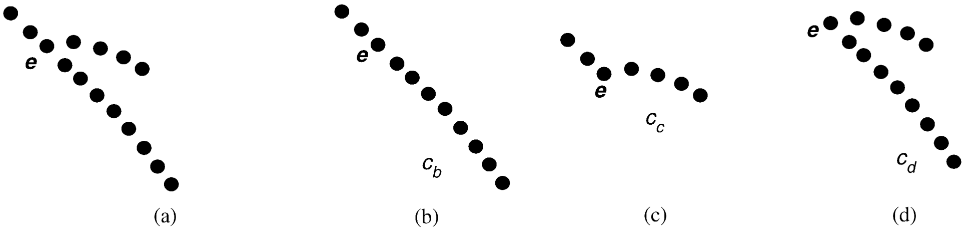

Fig. 3. An example of edge curve with two branches: (a) an edge curve with two branches, (b,c,d) three potential candidates for the representative edge curve.

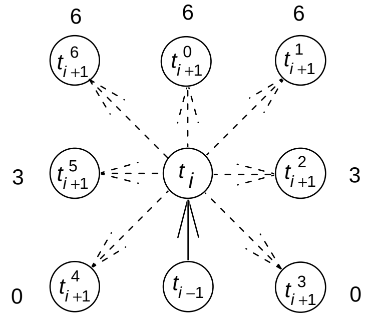

Fig. 4. An example of three continuous points within a 3 × 3 neighborhood.

a point on the trajectory. The PS is determined by extension direction of the point. If the extension direction of t_{i+1} is t^0_{i+1} or t^1_{i+1} or t^6_{i+1}, we let the PS of t_{i+1} be 6, if the extension direction of t_{i+1} is t^5_{i+1} or t^2_{i+1}, PS is associated with the value of 3, otherwise PS is set 0. Assume that the PS of $t_i\ (2<i\leqslant n)$ is denoted by ps_i and the *choice priority* of the edge curve T is denoted by cp_T. The *choice priority* cp_T can be computed by

$$cp_T = \sum_{i=2}^{n} ps_i. \tag{1}$$

Similarly, we can determine the *choice priorities* of c_b, c_c and c_d easily. They are represented by cp_b, cp_c and cp_d, respectively.

Hence, by the MLHP and the MSHP rules, the *representative edge curve* is the one with the largest *choice priority*. Mathematically, let c_r be the *representative edge curve* of Fig. 3(a), it necessarily meets the condition

$$\{c_r |\, r = a \text{ or } b \text{ or } c, \quad \text{and} \quad cp_r = \max(cp_b, cp_c, cp_d)\}. \tag{2}$$

At this point, a problem should be mentioned. In some cases, while using MSHP as a criterion for determination of a *representative edge curve*, one may also "miss" some important corners. Fortunately, our experiments indicate that branches of most edge curves with branches are much shorter than the finally selected *representative edge curves*. Therefore, in these scenarios, MLHP plays a more important role than MSHP in the process of assigning *representative edge curves*. It implies that, in most cases, the important corners cannot be missed.

3.2.4. Feature vectors

In order to retrieve images, it is necessary to efficiently compare two images to determine if they have a similar content. To this end, an efficient matching scheme has to be developed that makes use of the discriminatory information contained in the extracted features.

Let $\{(x,y),\ x,y = 1,2,\ldots,N\}$ be a two-dimensional $(N \times N)$ image pixel array, and denote by $F(x,y)$ the gray-scale intensity value at pixel (x,y). Further, let $f: F \to \vec{x}$ be a mapping from the image space onto the n-dimensional feature space, $\vec{x} = \{x_1, x_2, \ldots, x_n\}$, where n is the number of features used to represent the image. The difference between two images, F_1 and F_2, can be expressed as the distance, d, between the respective feature vectors, $\vec{x}_1$ and $\vec{x}_2$. Consequently, measuring the similarity, S, between two images can be formulated as one of computing the distance between their feature vectors. Two images are more similar means smaller distance between their feature vectors. The problem of retrieval can then be posed as follows: given a query image P, retrieve a subset of the images from the image database, $\mathbf{M}$, such that

$$S(P,M) \geqslant o \Leftrightarrow d(f(P), f(M)) \leqslant t, \quad M \in \mathbf{M}, \tag{3}$$

where o and t are user-specified thresholds. Alternatively, instead of specifying the threshold, a user can require the system to output, say, the top-20 images that are most similar to the query image.

Feature primitives are the quantities associated with or calculated from an image that can serve as bases for constructing feature vectors. For our cases, we propose the following three quantities as primitives for shape features:

1. *Fork ratio*. Fork ratio gives measure of complexity of the structure of an edge curve. Following Refs. [33,34], fork count is defined as the total number of branches that the waterfront has forked during the filling of an edge curve. Assume that c_i is a salient edge of

an image, its fork count is fc_i, and its length is l_i. The *fork ratio*, fr_i, of c_i is computed by

$$fr_i = fc_i/l_i. \tag{4}$$

2. *Rotation frequency*. Rotation frequency is designed to survey the bent degree of a salient edge curve. Let $T = \{t_0, t_1, \ldots, t_n\}$ be a salient edge and t_{i-1}, t_i, t_{i+1} be three continuous points on the T. Again taking Fig. 4 for example, as analyzed in Section 3.2.3, t_{i+1} has seven possible extension directions that are $t^0_{i+1}, \ldots, t^6_{i+1}$. If its extension direction is t^0_{i+1}, t_{i-1}, t_i and t_{i+1} thus form a straight line in the local 3×3 neighborhood. On the contrary, if t_{i+1}'s extension direction is any one of $t^1_{i+1}, t^2_{i+1}, \ldots, t^6_{i+1}$, we say that there is a rotation at the point t_i. Assume that the total rotation number of T is rn_T, *rotation frequency* of T, rf_T, can be computed by

   $$rf_T = rn_T/(n+1), \tag{5}$$

   where $n+1$ is the length of T. If the value of *rotation frequency* of a salient edge is 0, it means that this salient edge is a straight line. Obviously, a larger *rotation frequency* means that the curve is more bent.
3. *Corner frequency*. Corner frequency is a feature most probably associated with an edge curve's smoothness. Corners are important image features since they often correspond to unique features of an object or a scene. A corner point on an edge curve subtends a sharp angle between its neighboring points. Assume that a discrete salient edge curve is described by $c_k = [x_k, y_k]'$, $k = 0, 1, \ldots, n-1$, where x_k and y_k denote the x and y coordinates in the 2D image plane. To detect corners, the curve is first smoothed by

   $$x^{sm}_k = \frac{1}{2\omega_1 + 1} \sum\nolimits_{l=k-\omega_1}^{l=k+\omega_1} x_l, \qquad y^{sm}_k = \frac{1}{2\omega_1 + 1} \sum\nolimits_{l=k-\omega_1}^{l=k+\omega_1} y_l. \tag{6}$$

   To prevent artificial variations in the estimated angular values, ω_1 is a small integer. Here, we let $\omega_1 = 3$. The angle ϕ_k associated with each curve point c_k is then computed as the angle between the two vectors (c^{sm}_{k+1}, c^{sm}_k) and (c^{sm}_k, c^{sm}_{k-1}) using the smoothed curve points,

   $$\phi_k = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right), \tag{7}$$

   where

   $$a = |c^{sm}_{k-1} - c^{sm}_k|, \quad b = |c^{sm}_{k+1} - c^{sm}_k|, \quad c = |c^{sm}_{k+1} - c^{sm}_{k-1}|. \tag{8}$$

Significant corners are extracted from those points which have the local minimum angular values. Further, to prevent them being too closely located with one another, only one significant corner is allowed within each small curve interval ω_2 (in our algorithm, we set $\omega_2 = 2$). That is, an angle ϕ_k is significant corner provided that

$$\{\phi_k : |\,\phi_k = \text{local minimum angle}; \quad \phi_k = \min\nolimits_{l \in (k-\omega_2, k+\omega_2)} \phi_l\}. \tag{9}$$

Let the corner number of a salient edge, c_i, be cn_i and its length be l_i, $i = 1, 2, \ldots, n$, its *corner frequency*, cf_i, is defined as follows:

$$cf_i = cn_i/l_i. \tag{10}$$

Corner frequency conveys a rough measure of the smoothness of an edge curve. The larger the *corner frequency*, the more frequent sharp changes the curve undergoes.

As mentioned in Section 2, good feature descriptors should be invariant to scaling, rotation, and translation. Obviously, if a salient edge is rotated or translated, its fork count, total rotation number and corner number are all invariant. Hence, *fork ratio*, *rotation frequency* and *corner frequency* are all invariant to rotation and translation. Since the feature primitives are normalized with respect to the length of the salient edge curve, they are invariant to scaling theoretically. Nevertheless, the edge detector itself may not be scaling invariant. For example, in small scale and low-resolution cases, the edge detector may miss edges, which it would otherwise extract successfully when the resolution is high enough. Fortunately, our feature primitives are based on salient edges that are hardly influenced by the image scales. Therefore, the proposed feature primitives are scaling-invariant if the change in image size is within a reasonable range.

The shape feature vectors are then constructed in terms of the three feature primitives. Let $C = \{c_1, c_2, \dots, c_i, \dots, c_L\}$ be the set of salient edges, which have been obtained from an image P, and c_i's feature, f_i, be defined as $f_i = (fr_i, rf_i, cf_i)$, for $i = 1, \dots, L$. The *feature vector*, $\vec{f}_P$, of the image P is constructed as follows:

$$\vec{f}_P = [f_1, f_2, \dots, f_i, \dots, f_L]. \tag{11}$$

Note that the number of salient edges, L, is not a constant, which is relevant to the complexity of the image. In general, the more complex the images are, the larger the number L is. As to the image P, assume that N denotes the total number of its edge curves. We determine L by

$$L = \begin{cases} N & N < 10, \\ N/2 & 10 \leqslant N \leqslant 30, \\ N/3 & N > 30. \end{cases} \tag{12}$$

In summary, the proposed shape features have the following desired characteristics:

- They can represent structural information embedded in the edge maps, which is effective in the CBIR.
- They are translation, rotation and scaling-invariant to a certain extent.
- They can be computed easily.
- They are obtained from organized edge curves, and hence not very sensitive to noise.

3.3. Similarity measure

Image retrieval systems usually represent image features by multi-dimensional feature vectors. For example, the feature vector in Eq. (11) represents the whole shape features of an image, and each of its bins describes the features of the corresponding salient edge. The image database is searched by these feature vectors, and only the images that have the closest feature vectors to that specified in the query are retrieved. For such a search, a measure of similarity between feature vectors has to be defined. A simple and common used similarity measure is the bin-to-bin (also called one-to-one) matching, which compares contents of corresponding feature vector bins only. However, it is argued that this similarity measure is inappropriate to our proposed feature vector by two reasons:

1. The bin-to-bin matching implies that all feature vectors have the same bin size. However, in our approach, the bin size of an image equals to its amount of salient edges, and there is no guarantee that feature vectors of images have the same bin size.
2. In the proposed feature vector, each bin represents the features of a salient edge. However, in practice, accurate edge curves are not ensured. Therefore, bin-to-bin matching is inherently risky to unacceptably many "false matchings", and may cause the serious performance degradation in terms of similarity measure.

Thus, in order to reduce the influence of inaccurate edge extraction, we reason that the many-to-many matching strategy might be available. In the field of CBIR, the Earth Mover's Distance (EMD) [25,26] is a distinguished similarity measure based on many-to-many matching. The EMD measures the minimal cost that must be paid to transform one distribution into another. It is same as the transportation problem and can be solved efficiency by linear optimization algorithms that take advantage of its special structure. The drawback of the EMD is that it requires a very complex optimization procedure, which always results in much longer retrieval time for a practical image retrieval system. Alternatively, integrated region matching (IRM) [18,31] can be used as the image similarity measure, which essentially is a simplified version of EMD. Instead of solving a liner programming problem directly, it adopts a greedy algorithm to achieve an approximation. Although the integrated region matching claims to be robust to inaccurate segmentation, it actually still yields the poor retrieval performance. The reason is that, it incorporates the characteristics of all segmented regions, but many extraneous and meaningless regions could lead to many rough-and-tumble matchings.

In this section, we employ the many-to-many matching strategy combing EMD and IRM to the salient edges based image retrieval. Imitating IRM, we name it as *integrated salient edge matching* (ISEM). It "softens" the matching by allowing for matching a salient edge of one image to several salient edges of another image. It is remarked that by incorporating the features of all the salient edges and hence making full use of the shape information within an image, the similarity measure may gain robustness against inaccurate edge curves. The centric idea behind ISEM is that the most similar salient edge pair is matched first. The similarity measure between two images is calculated as the weighted sum of the similarity between salient edges pairs. The weights are determined by a scheme of minimizing the overall cost subjected to certain constraints (Eqs. (17)–(20)).

We adopt a simple example shown by Fig. 5 to illustrate the difference between one-to-one matching and many-to-many matching scheme. Fig. 5(a) displays the salient edges of an image. It only includes a circle and a straight line. Fig. 5(b) is the same image as Fig. 5(a), except that for some reasons (noise, obscurity, and so on), the edges cannot be extracted accurately. It is seen from Fig. 5(b) that the salient edge of circle is divided into two separate salient edges. In Fig. 5(a), we use s_1^1 to represent the circle and s_2^1 to represent the straight line. In Fig. 5(b), those three salient edges are represented by s_1^2, s_2^2 and s_3^2, respectively. Let us consider the matching between the two images. Considering the one-to-one matching scheme, according to the length, s_1^1 could be matched with s_1^2, and s_1^2 could be matched with s_2^2, resulting in a "false matching". However, if the many-to-many matching is used instead, it is guided by the similarity between the matching pair. Thus, s_1^1 could be matched with both s_1^2 and s_2^2, and s_2^1 could be matched with s_3^2. The "false matchings" might be avoided.

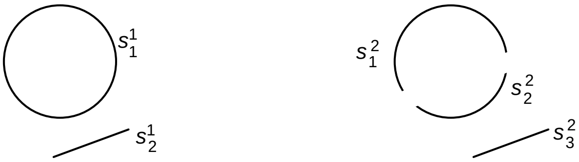

Fig. 5. A simple example to illustrate the difference between one-to-one matching and many-to-many matching scheme (a) An image with two salient edges, (b) the same image with three salient edges because of inaccurate edge extraction.

Details of ISEM are as follows. Assume that image P and P' are represented by salient edge sets $C = \{c_1, c_2, \ldots, c_m\}$ and $C' = \{c'_1, c'_2, \ldots, c'_n\}$, respectively. Let $\vec{f} = [f_1, f_2, \ldots, f_m]$ and $\vec{f}' = [f'_1, f'_2, \cdots, f'_n]$ be the feature vector of images P and P', respectively. Further, let l_i be the length of c_i, $i = 1, \ldots, m$, and l'_j be the length of c'_j, $j = 1, \ldots, n$, respectively. With these notations, several terminologies are in order.

Definition 1. *Salient edge significance* (SES). SES is a measure of the importance of a salient edge within an image. The SES of a salient edge, c_i, is defined as

$$ses_i = l_i / \sum_{k=1}^{m} l_k. \tag{13}$$

Definition 2. *Distance between salient edges* (DBSE). DBSE is defined as the Euclidean distance between the features of two salient edges. The distance between c_i and c'_j is

$$d(c_i, c'_j) \equiv d(f_i, f'_j) = \sqrt{(fr_i - fr'_j)^2 + (rf_i - rf'_j)^2 + (cf_i - cf'_j)^2}. \tag{14}$$

Definition 3. *Similarity between salient edges* (SBSE). The similarity between c_i and c'_j is

$$s(c_i, c'_j) = \exp(-\alpha \cdot d(c_i, c'_j)). \tag{15}$$

In the following parts of this section, $s(c_i, c'_j)$ will be written as $s_{i,j}$ in short. In order to discriminate different distances more distinctly, we multiply the DBSE by a large constant α. In the algorithm, we set $\alpha = 100$.

Definition 4. *Significance honor* (SH). In order to increase robustness against edge extraction errors, a salient edge in one image is allowed to be matched to several salient edges in another image. A matching between c_i and c'_j is assigned with a *significance honor*, $sh_{i,j}$, which fulfills the requirements given in Eqs. (17)–(22).

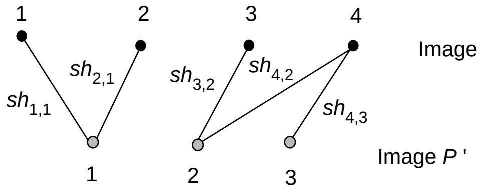

Fig. 6. The graphical explanation of ISEM.

The *significance honor* indicates the importance of the matching for determining similarity between images. ISEM between two images can then be represented by an edge-weighted graph. Fig. 6 gives a graphical explanation. Every vertex corresponds to a salient edge. If two vertices are linked, the two salient edges are matched with an SH represented by the weight on the edge. If two vertices are not linked, the corresponding salient edges are either in the same image or the SH of matching them is zero. The match between images is characterized by similarities between salient edges and their SHs. A matching between salient edges that contributes to computing the similarity between two images is referred to as a *valid matching*.

Definition 5. *Similarity between two images* (SBTI). SBTI is defined as a weighted sum of the similarity between salient edge pairs, with weights being the SHs to be determined in the sequel. That is,

$$S(P,P') = \sum_{i,j} sh_{i,j} s_{i,j}, \tag{16}$$

where $s_{i,j}$ can be computed by Eq. (15). This similarity measure is called the *integrated salient edge matching measure* between the mentioned images.

The value of SBIT is bounded between 0 and 1. When two images are same, the similarity between them equals 1. Now the issue of measuring similarity between images can be regarded as one of determining the *significance honors*, $sh_{i,j}$. In order to yield good similarity measure, we put some constraints on $sh_{i,j}$.

$$sh_{i,j} \geqslant 0, \quad 1 \leqslant i \leqslant m, \quad 1 \leqslant j \leqslant n, \tag{17}$$

$$\sum_{j=1}^{n} sh_{i,j} = ses_i, \quad i = 1, \ldots, m, \tag{18}$$

$$\sum_{i=1}^{m} sh_{i,j} = ses'_j, \quad j = 1, \ldots, n, \tag{19}$$

$$\sum_{i=1}^{m} ses_i = \sum_{j=1}^{n} ses'_j = 1, \tag{20}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} sh_{i,j} = 1. \tag{21}$$

Constraints of (17)–(21) ensure that all the salient edges play a role for measuring similarity. We also expect those *valid matchings* to link the most similar salient edges at the highest priority. Another constraint that we put on $sh_{i,j}$ is called "the Most Similar one with Highest Priority (MSHP)" [18,31]. The MSHP attempts to assign as much SH as possible to the *valid matching* with maximal similarity. The MSHP can be described by

$$\max s_{i,j} \Rightarrow \max sh_{i,j}. \tag{22}$$

The ISEM algorithm is implemented by following seven steps [31] under constraints of (17)–(22).

Step 1. Set two sets: A and B. Initially, let $A = \varnothing$, and $B = \{(i,j) | 1 \leqslant i \leqslant m;\ 1 \leqslant j \leqslant n\}$;
Step 2. Select the maximum $s_{i,j}$ for $(i,j) \in B$. Label the corresponding (i,j) as $(i_{\max}, j_{\max})$;
Step 3. $\min(ses_{i_{\max}}, ses'_{j_{\max}}) \rightarrow sh_{i_{\max}, j_{\max}}$;
Step 4. If $ses_{i_{\max}} \leqslant ses'_{j_{\max}}$, set $sh_{i_{\max},j} = 0, j \neq j_{\max}$; otherwise, set $sh_{i,j_{\max}} = 0, i \neq i_{\max}$;
Step 5. $ses_{i_{\max}} - \min(ses_{i_{\max}}, ses'_{j_{\max}}) \rightarrow ses_{i_{\max}}$;
$ses'_{j_{\max}} - \min(ses_{i_{\max}}, ses'_{j_{\max}}) \rightarrow ses'_{j_{\max}}$;
Step 6. $A + (i_{\max}, j_{\max}) \rightarrow A$; $B - (i_{\max}, j_{\max}) \rightarrow B$;
Step 7. If $B \neq \varnothing$, go to Step 2; otherwise, stop.

The similarity measure, ISEM, has the following characteristics:

- If two images are the same, the similarity between them equals 1, which agrees with humans' intuition.
- Although all the salient edges of two images are considered, only *valid matchings* have contributions to the similarity between these two images,

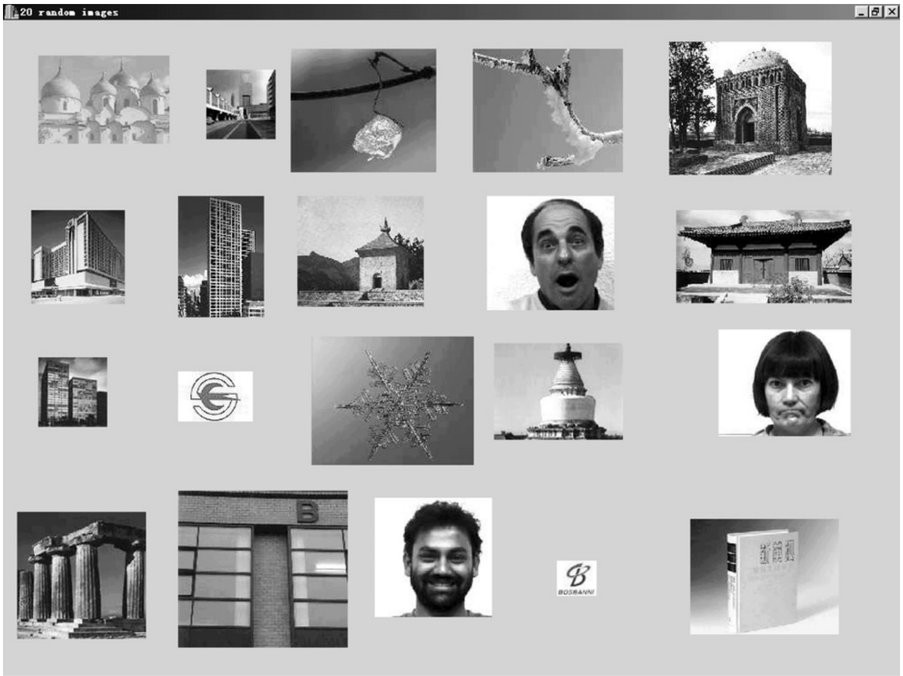

Fig. 7. Random picks from our image database.

since the SHs of *invalid matchings* are zero. Therefore, if two images have no *valid matching*, their similarity is 0.

- We use the minimal SES of two matched salient edges as their matching SH due to the fact that if one of the two salient edges is insignificant, the matching of them is meaningless, even though the other salient edge is very important.
- By allowing many-to-many matching to be valid, the proposed similarity measure is comparatively robust against inaccurate edge extraction.

4. Preliminary experimental results

4.1. System overview

We have implemented a prototype image retrieval system with C++Builder in accordance with the proposed approach. The image database is implemented on SQL Server. It contains about 6500 images that involve building images, landscape images, trademark images, face images and *etc*. These images are collected from various sources such as Corel stock photo library, images downloaded from webs, key frames from digitized video of television serials, images captured using a digital camera and Yale research Lab face database.[1] All images from our database are converted to gray images. Fig. 7 shows 20 images that are randomly picked out from the image database.

The prototype system can provide users with query either by example or by sketch. Query by example means that users present a query image and then the system returns 50 best matching images from the image database. The system offers a tool of drawing sketches for users as well. After users have drawn the query sketch, the system performs similarity matchings in the image database. Finally, the 50 most similar images with the query sketch are outputted. This is called query by sketch.

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

At any time during the retrieval process, users can "double click" on an image returned by the system to view its information that includes its original size, its origin, and other parameters. Users can also choose this image as an example image to perform a new query.

4.2. Retrieval by example

Fig. 8 illustrates a typical query by example. The top-left image is a query image provided by the user. The system outputs the query results according to the similarity order from left to right and top to bottom. In this figure, we only show the top-10 query results.

4.3. Retrieval by sketch

Fig. 9 gives an example of query by sketch. A query sketch drawn by the user is shown at the upper-left corner. Then, sketched edges are considered as salient edges and retrieval is performed. Finally, images in the database that are most similar to the query are displayed.

4.4. Retrieval efficiency

Our aim is to develop an efficient image retrieval scheme. In [13,14], Jain et al. indicate that an efficient CBIR scheme must have the following features:

- *Accuracy*: It must be accurate, i.e. the retrieved images must resemble the query image.
- *Speed*: Since image database typically has thousands of images, it must be "real-time". That is, its query speed must be fast.

4.4.1. Accuracy

Like [7,12,17], we define the retrieval accuracy as follows:

(1) The retrieval accuracy for a query image Q is defined as

$$\eta_q = \begin{cases} (u/V) \times 100\% & \text{if } V \leqslant W, \\ (u/W) \times 100\% & \text{if } V > W, \end{cases} \tag{23}$$

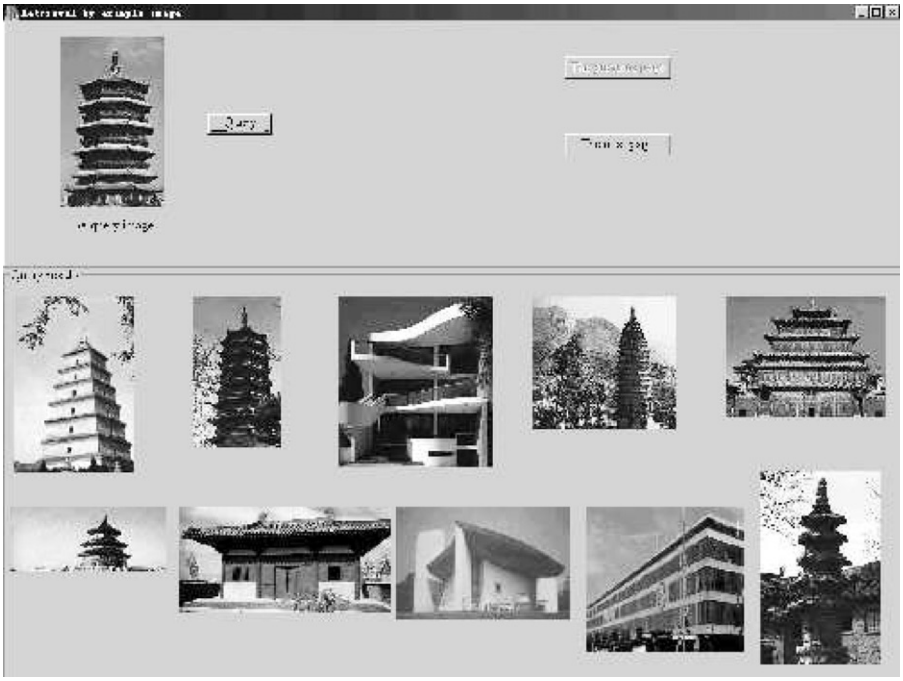

Fig. 8. A typical example of query by example image.

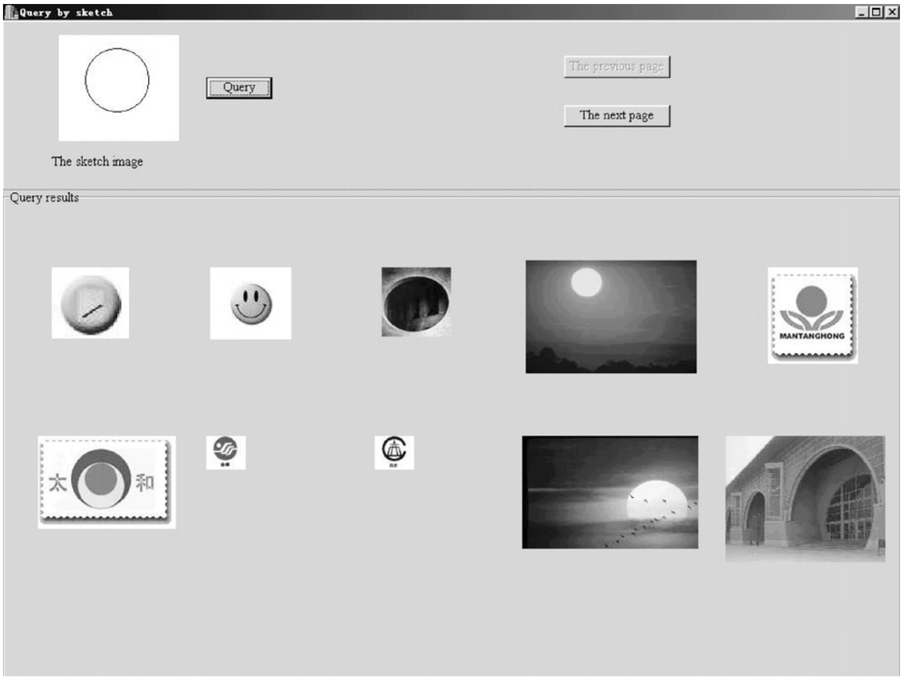

Fig. 9. A typical example of query by sketch.

where W is the number of similar images of Q found in database by subjects, V is the number of similar images of Q retrieved in database by system, and u is the number of similar images found by both humans and system. More specifically, the retrieval system entitles the user to select V. For example, if a user selects 10 as the value of V, he means to require the system output the 10 images that are the most similar to his query image. In our system, the value of V can be 10, 20, 30, 40 or 50. W is used to estimate the retrieval performance in the experiment. For each query, the retrieval system will output the top-V images according to the similarity order. Nevertheless, the user may not agree with the query results by the system. W is then the number of similar images retrieved by the user after he browses the whole database. In our experiments, for each query, we let five subjects provide similar images by their own perception, and the final similar images are those accepted by not less than three subjects.

(2) The retrieval accuracy of a retrieval algorithm is defined as the average retrieval accuracy for G images, that is,

$$\eta = \frac{1}{G}\sum_{q=1}^{G} \eta_q. \tag{24}$$

We randomly select 200 images from our image database and 50 sketch images drawn by users to evaluate retrieval accuracy based on Eqs. (23) and (24). The average W of 250 queries is 26. In order to verify the effectiveness of our method, we have implemented the methods of [1] and [33] for comparison purposes. Fig. 10 shows a graph illustrating retrieval accuracy of these three schemes as a function of the demanded number of similar images returned by system.

In [1], Bober presents an MPEG-7 Contour-Based Descriptor. This descriptor is in the light of curvature scale-space (CSS) representations of contours. A CSS index is used for matching and indicates the heights of the most prominent peaks in the so-called CSS image. In [33], a contour based descriptor based on structural features such as “MaxFillingTime”, “MaxForkCount”, etc., is

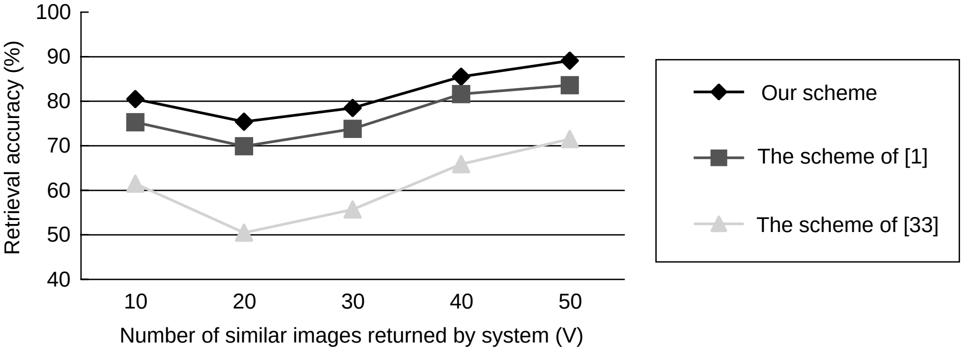

Fig. 10. Retrieval accuracy of three schemes according to the number of similar images returned by system.

generated. Compared with the proposed algorithm, we argue that the schemes of [1] and [33] have several disadvantages that may reduce their retrieval accuracy.

- Although CSS descriptor [1] extracts shape features at multiple scales, it represents shape features only by features of prominent peaks on the contours, which may not describe structural information of edges and global features of the edge curve.
- In [33], its shape descriptor contains some structural information of edges. However, it only relies on several special edges in the image such as the most complicated edge (with maximal forks), and the longest edge. Those special edges may not represent the shape features of images very appropriately.
- Both retrieval scheme of [1] and retrieval scheme of [33] use one-to-one matching strategy during similarity measure process. In fact, none of edge extraction approaches promises to extract accurate edge curves. Hence, this one-to-one matching strategy is harmful to retrieval performance.

As be seen from Fig. 10, the preliminary experimental results show that the proposed retrieval scheme has the highest accuracy, and the retrieval scheme of [33] has the lowest accuracy. This observation does support the above analyses.

4.4.2. Speed

Generally, retrieval speed is measured by average retrieval time [31]. The average retrieval time mainly relies on the size of image database, the software and hardware conditions that the system runs on, the feature size, and the matching number between two images in the similarity measure process. All the tested schemes are implemented under Microsoft Windows 2000. The machine we used is a Pentium-4 1.7G PC with 512 MB DRAM main memory. The development kit is C+ + Builder 6.0. Our image database contains 6500 images. We also use those 250 images used in Section 4.4.1 to evaluate average time of those three retrieval schemes. Since they are implemented under the same software and hardware conditions, and retrievals are performed in the same database, the retrieval time mainly depends on the feature size of each image and the matching number between two images. In the proposed approach, the average number of salient edges in an image is 15. Therefore, its average feature size of each image is 15×3, and its average matching number between two images is 15×15. In the method of [1], the average of CSS edges in an image is 15. As a consequence, its average feature size of each image is 15×16, and its average matching number between two images is 15. In the method of [33], its descriptor contains 18 feature primitives, and its matching number between two images is 1. Fig. 11 displays the retrieval time of the three approaches.

In conclusion, by experimental results shown by Figs. 10 and 11, we find our retrieval scheme has the highest accuracy but the longest time compared with two other retrieval approaches.

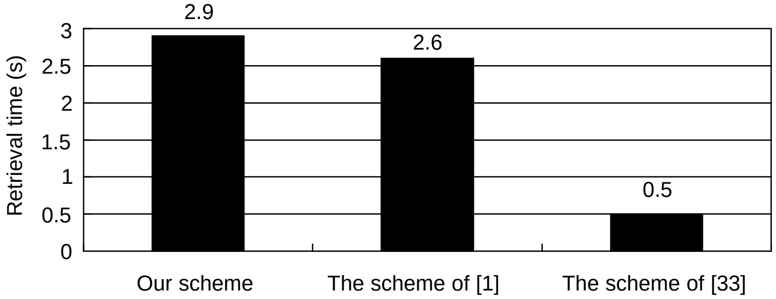

Fig. 11. Retrieval time comparison of three schemes.

5. Conclusions

The main contributions of this paper are summarized below:

- A simple shape feature representation in terms of salient edges has been proposed and demonstrated effective.
- A similarity measure that integrates properties of all the salient edges has been described, which is capable of dealing with inaccurate edge curves.
- An application to a database of about 6500 images is implemented.
- A detailed performance comparison with two other contour-based methods is provided.

Future work includes exploring effective indexing schemes that can improve the retrieval speed, and adding spatial relationships of salient edges to the shape descriptor.

Acknowledgements

This work was supported by the "973" project (China) and NSFC. The authors would like to thank Dr. Gong Cheng who helped to improve the quality of the paper. They would also like to acknowledge the anonymous reviewers and the associate editor for their helpful comments and constructive suggestions.

References

[1] M. Bober, MPEG-7 visual shape descriptors, IEEE Trans. Circuits Systems Video Technol. 11 (6) (2001) 716–719.

[2] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, J. Malik, Blobworld: a system for region-based image indexing and retrieval, in: Proceedings of the Visual Information Systems, Amsterdam, The Netherlands, June 1999, pp. 509–516.

[3] Y. Chans, Z.B. Lei, D. Lopresti, S.Y. Kung, A feature-based approach for image retrieval by sketch, Proc. SPIE 3430 (1998) 72–82.

[4] A. Del Bimbo, P. Pala, S. Santini, Image retrieval by elastic matching of shapes and image patterns, in: Proceedings of the IEEE Multimedia, Hiroshima, Japan, 1996, pp. 215–218.

[5] A.A. Farag, E.J. Delp, Edge linking by sequential search, Pattern Recognition 28 (5) (1995) 611–633.

[6] M. Flickner, H. Sawhney, et al., Query by image and video content: the QBIC system, IEEE Comput. 28 (9) (1995) 23–32.

[7] C.S. Fuh, S.W. Cho, K. Essig, Hierarchical color image region segmentation for content-based image retrieval system, IEEE Trans. Image Process. 9 (1) (2000) 156–162.

[8] W.J. Grosky, R. Mehrotra, Index-based object recognition in pictorial data management, Comput. Vision Graph. Image Process. 52 (3) (1990) 416–436.

[9] Y.H. Gu, T. Tjahjadi, Corner-based feature extraction for object retrieval, in: Proceedings of the IEEE Conference on Image Processing, Kobe, Japan, 1999, pp. 119–123.

[10] A. Gupta, R. Jain, Visual information retrieval, Comm. ACM 40 (5) (1997) 70–79.

[11] K. Hirata, T. Kato, Query by visual example, Advances in Database Technology EDBT'92, Third International Conference on Extending Database Technology, New York, USA, March 1992.

[12] I.S. Hsieh, K.C. Fan, Multiple classifiers for color flag and trademark image retrieval, IEEE Trans. Image Process. 10 (6) (2001) 938–950.

[13] A. Jain, A. Vailaya, Image retrieval using color and shape, Pattern Recognition 29 (1996) 1233–1244.

[14] A.K. Jain, A. Vailaya, Shape-based retrieval: a case study with trademark image database, Pattern Recognition 31 (9) (1998) 1369–1390.

[15] H.K. Kim, J.D. Kim, Region-based shape descriptor invariant to rotation, scale and translation, Signal Processing: Image Communication 16 (2000) 87–93.

[16] W.Y. Kim, Y.S. Kim, A region-based shape descriptor using Zernike moments, Signal Processing: Image Communication 16 (2000) 95–102.

[17] V.D. Lecce, A. Guerriero, An evaluation of the effectiveness of image features for image retrieval, J. Visual Commun. Image Representation 10 (1999) 351–362.

[18] J. Li, J.Z. Wang, G. Wiederhold, IRM: integrated region matching for image retrieval, in: Proceeding of ACM Multimedia, California, USA, 2000.

[19] R. Mehrotra, J. Gray, Similar-shape retrieval in shape data management, IEEE Comput. 28 (1995) 57–62.

[20] S.F. Mokhtarian, S. Abbasi, J. Kittler, Efficient and robust retrieval by shape content through curvature scale space, in: Proceedings of the International Workshop on Image Database and Multimedia Search, New York, USA, 1996, pp. 35–42.

[21] S. Mukherjea, K. Hrrata, AMORE: a world wide web image retrieval system, Proc. World Wide Web 2 (3) (1999) 115–132.

[22] A. Natsev, R. Rasto, WALRUS: a similarity retrieval algorithm for image database, SIGMOD Rec. 28 (2) (1999) 395–406.

[23] A. Pentland, R.W. Picard, S. Sclaroff, Photobook: tools for content-based manipulation of image database, Proc. SPIE 2185 (1994) 34–47.

[24] R.W. Picard,T. Kabir, Finding similar patterns in large image databases, in: Proc. Internat. Conf. Acoust. Speech, Signal Process., Vol. 5, 1993, pp. 162–164.

[25] Y. Rubner, L.J. Guibas, C. Tomasi, The earth mover's distance, multi-dimensional scaling, and color-based image retrieval, in: Proceedings of the DARPA Image Understanding Workshop, May, 1997, pp. 661–668.

[26] Y. Rubner, C. Tomasi, L.J. Guibas, A metric for distributions with applications to image databases, IEEE International Conference on Computer Vision, Bombay, India, January 1998.

[27] T. Sikora, The MPEG-7 visual standard for content description—an overview, IEEE Trans. Circuits Systems Video Technol. 11 (6) (2001) 702–796.

[28] J.R. Smith, S.F. Chang, Image classification and querying using composite region templates, Internat. J. Comput. Vision Image Understanding 75 (1999) 165–174.

[29] J.R. Smith, S.F. Chang, VisualSEEK: a fully automated content-based image query system, in: Proceedings of the ACM Multimedia, Boston, MA, USA, November 1996, pp. 87–98.

[30] N. Takahashi, M. Iwasaki, T. Kunieda, Y. Wakita, N. Day, Image retrieval using spatial intensity features, Signal Processing: Image Communication 16 (2000) 45–57.

[31] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. PAMI 23 (9) (2001) 1–17.

[32] J.Z. Wang, et al., Content-based image indexing and searching using Daubechies' Wavelets, Internat. J. Digital Libraries 1 (4) (1998) 311–328.

[33] X.S. Zhou, T.S. Huang, Edge-based structural features for content-based image retrieval, Pattern Recognition Lett. 22 (2001) 457–468.

[34] X.S. Zhou, Y. Rui, T.S. Huang, Water-Filling: a novel way for image structural feature extraction, in: Proceedings of the IEEE Conference on Image Processing, Kobe, Japan, 1999, pp. 570–574.