Multimedia Systems 9: 15–30 (2003)
Digital Object Identifier (DOI) 10.1007/s00530-002-0075-y

Multimedia Systems

Evaluation of MPEG-7 shape descriptors against other shape descriptors

Dengsheng Zhang, Guojun Lu

Gippsland School of Computing and Information Technology, Monash University, Churchill, Victoria 3842, Australia

Abstract. Shape is an important image feature – it is one of the primary low level image features exploited in content-based image retrieval (CBIR). There are generally two types of shape descriptors in the literature: contour-based and region-based. In MPEG-7, the curvature scale space descriptor (CSSD) and Zernike moment descriptor (ZMD) have been adopted as the contour-based shape descriptor and region-based shape descriptor, respectively. In this paper, the two shape descriptors are evaluated against other shape descriptors, and the two shape descriptors are also evaluated against each other. Standard methodology is used in the evaluation. Specifically, we use standard databases, large data sets and query sets, commonly used performance measurement and guided principles. A Java-based client-server retrieval framework has been implemented to facilitate the evaluation. Results show that Fourier descriptor (FD) outperforms CSSD, and that CSSD can be replaced by either FD or ZMD.

Key words: Fourier descriptor – Curvature scale space – Moments – Grid descriptor – CBIR – Shape

1 Introduction

Shape is one of the primary low level image features exploited in content-based image retrieval (CBIR). There are generally two types of shape representation methods in the literature: the region-based and contour-based methods. The classification of the varieties of shape methods is given in Fig. 1. In the next section, common shape methods used for image retrieval are briefly discussed. For a comprehensive shape review, the reader is referred to Loncarie [1].

1.1 Contour-based shape representations

Contour shape representations exploit only shape boundary information. Contour based methods gain popularity because it is usually simple to acquire and is descriptive sufficiently in many applications. There are generally two types of very different approach for contour shape modeling: conventional and structural. Conventional approaches treat the boundary as a whole, and a feature vector derived from the whole boundary is used to describe the shape. The measure of shape similarity is usually the Euclidean distance between the feature vectors. Structural approaches break the shape boundary into segments, known as *primitives*, using certain criterion. The final representation is usually a string or a tree, and the measure of shape similarity is string matching or graph matching. In the following subsections, these two methods are briefly discussed.

Correspondence to: D. Zhang
(e-mail: dengsheng.zhang@infotech.monash.edu.au)

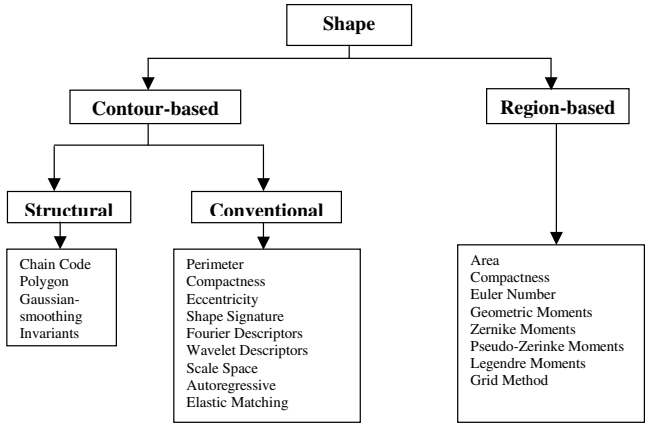


Fig. 1. Taxonomy of shape description techniques

1.1.1 Conventional shape representations

Conventional shape representations include global shape descriptors [2], shape signatures [3], spectral descriptors [4–12], curvature scale space (CSS) [13], elastic matching [14] and autoregressive method [4]. Global descriptors such as area, circularity (perimeter^2/4$\pi\times$area), eccentricity (length of major axis/length of minor axis), and axis orientation used in QBIC can only discriminate shapes with large dissimilarities,

so they are usually used as filters to eliminate false hits, or combined with other shape descriptors to discriminate shapes. The elastic matching [14] is similar to global descriptors, because only three global features are extracted from the boundary.

Shape signatures such as complex co-ordinates, curvature and angular representations are local representations of shape features in nature, and they are sensitive to noise and are not robust. In addition, shape representation using shape signatures requires intensive computation during similarity calculation, due to the complex normalization of rotation invariance. Consequently, these representations need further processing using a spectral transform such as the Fourier or wavelet transform.

The Autoregressive (AR) method [4] is based on the stochastic modeling of a 1D function f obtained from the shape. A linear AR model expresses a value of a function as a linear combination of a certain number of preceding values. Specifically, each function value in the sequence has some correlation with previous function values, and can therefore be predicted through a number of, say, m observations of previous function values. The drawback of the AR method is that, in the case of complex boundaries, a small number of AR parameters is not sufficient for description. The choice of m is a complicated problem, and is usually decided empirically. Besides, the physical meaning associated with each descriptor is not clear.

Spectral descriptors include the Fourier descriptor (FD) and wavelet descriptor (WD). They are derived from spectral transform on shape signatures. With FD, global shape features are captured by the first few low frequency terms, while higher frequency terms capture the finer features of the shape. The FD overcomes not only the weak discrimination capability of the global descriptors, but also the noise sensitivity in the shape signature representations. Other advantages of FD method include easy normalization and compactness. Eichmann et al. [15] used the short-time Fourier descriptor (SFD) in an attempt to locate local boundary features more accurately. However, Zhang and Lu [9] have found that FD outperforms SFD in image retrieval, because SFD does not make use of global boundary features which are very robust to shape variations. Furthermore, SFD is not rotation invariant – the best shift matching is needed to match two sets of SFDs. Recently, several researchers have proposed the use of WD for shape representation [8, 16]. Similar to SFD, though, WD is not rotation invariant, which means that a complex matching scheme is required. In Yang et al. [8], the similarity measurement algorithm needs $2^L \times N$ all-level shift matching, where L is the number of levels of resolution of the wavelet transform and N is the number of normalized boundary points. In Tieng and Boles [16], the number of matchings for similarity measurement is not only large but is also dependent on the complexity of the shape, since the similarity measurement is all-level shift matching of all the zero-crossing points of the wavelet approximation of the shape. Furthermore, the WD of a rotated shape will be very different from the WD of the original shape, even after shifting reorder. This is because uniform windows are used in calculating the dyadic wavelets.

Asada and Brady [17] used curvature scale space (CSS) to derive "primitive events" from the shape boundary. Mokhtarian and Abbasi [13, 18, 19] have used CSS for image retrieval. The CSS method is a method between the conventional and structural approaches. The feature extraction process is done globally, however, the extracted features are essentially a structural representation of shape in nature. Therefore, a non-conventional matching scheme has to be found.

1.1.2 Structural shape representations

Another member of the contour shape analysis family is *structural shape representation*. With the structural approach, shapes are broken down into boundary segments called *primitives*. An invariant is derived from each segment to represent the curve segment. A common method of deriving primitives is to first apply an approximation process, such as polygon or polynomial approximation [20, 21], curve fitting and Gaussian smoothing [22, 23]. Then the primitives are found by determining the breakpoints of the contour. After selection of the primitives, a method of organizing the primitives is determined, so that classification and searching can be conducted efficiently. The various structural methods differ in the selection of primitives and the organization of the primitives for shape representation. Shape *invariants* [12, 24] can also be viewed as a structural approach, because they also represent shape based on boundary primitives.

The main merit of the structural approach is its capability to do partial matching. However, there are several drawbacks with structural methods:

1. The main drawback of the structural approach is the generation of primitives and features. Because there is no formal definition for an object or shape, the number of primitives required for each shape is not known. In addition, the process to generate primitives is not algorithmic, and is usually empirical. Therefore, it difficult to apply it to general cases.
2. Ambiguous matching. Since the shape and its representation is a many-to-one mapping, the matching of one or more features does not guarantee full shape matching.
3. Failure to capture global shape features, which are equally important for the shape representation, because structural representation does not preserve the topological structure of the object. Variations of the object boundary can cause significant changes to local structures. In these cases, global features are more reliable.
4. Structural methods have a higher computational and implementation complexity than conventional techniques.

1.2 Region-based shape representations

In region-based techniques, all pixels within a shape region are taken into account to obtain the shape representation. Common region-based methods use moment descriptors to describe shape [2, 25–29]. These include geometric moments, Legendre moments, Zernike moments and pseudo Zernike moments.

Geometric moments representations interpret a normalized gray level image function as a probability density of a 2D random variable. The first seven invariant moments, derived from the second and third order normalized central moments, are given by Hu [25]. There is no general rule in acquiring higher order invariants. Orthogonal moments using the Legendre polynomial, Zernike polynomials and pseudo-Zernike

polynomials have been proposed [27] to obtain more moment invariants for accurate shape description. Arbitrary order of moment invariants can be constructed through these orthogonal moments. It has been shown [28] that Zernike moments and pseudo-Zernike moments outperform other moments in terms of noise sensitivity, redundancy and reconstruction error. Recently, Zernike moments have been used for image retrieval and have shown good results [30].

The grid method has also been used in several applications [31–33]. The grid-based method attracts interest for its simplicity in representation, is intuitive, and also agrees with the shape coding method in MPEG-4.

Since region-based shape representations combine information across an entire object, rather than exploiting information just at the boundary points, they can capture the interior content of a shape. Other advantages of region-based methods are that they can be employed to describe non-connected and disjoint shapes. However, region-based representations do not emphasize contour features, which are equally crucial for human perception of shapes.

1.3 Techniques to be evaluated

In the development of MPEG-7, six principles have been set to evaluate the overall performance of a shape descriptor: good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance, and hierarchical coarse to fine representation [34]. According to these principles, and based on the above discussion, conventional shape methods such as the FD, CSSD, moments and grid methods are suitable for image retrieval.

CSSD and ZMD have been adopted by MPEG-7 as the contour-based shape descriptor and region-based shape descriptor respectively [35–38]. However, these two descriptors are not comprehensively evaluated against other important descriptors. In Abbasi and Mokhtarian [1], CSSD is compared with FD, but the comparison is not conclusive in three respects. First, the FD is derived from boundary coordinates – recent findings show that a FD derived from a centroid distance function significantly outperforms a FD derived from boundary coordinates [10]. Secondly, the evaluation uses a fish database, which does not reflect a general application. Thirdly, only a small set of "carefully" selected shapes are used to test the query.

ZMD has only been evaluated with geometric moments on region-based shapes [30]. It has not been evaluated with the recently proposed grid method, which has been used in several applications. Since region-based methods can be applied to contour shapes, it is also appropriate to evaluate them in a contour shape database and against contour shape descriptors.

The evaluation of different shape descriptors is usually challenging due to the lack of a standard methodology. One of the most contentious issues in the evaluation is the test database. The evaluation of shape descriptors in the literature generally uses their own data and query sets, which are either too application-dependent or unacceptably small. Recognizing this important issue, the MPEG-7 developers have established a shape database combining data sets from several active research groups involved in the development of MPEG-7. The database is of a reasonable size and generality. It has been subjectively tested and organized into a number of individual data sets to test the shape descriptors' behavior under various distortions. At the moment, the authors have not found any other shape database which is more generic and acceptable than this shape database.

It can be claimed that the evaluation in this paper uses standard principles, a standard database, a large data and query set, and common performance measurement.

The rest of the paper is organized as follows. In Sect. 2, two contour-based shape descriptors, FD and CSSD, are described and evaluated. In Sect. 3, three region-based shape descriptors, ZMD, the geometric moment descriptor (GMD) and the grid descriptor (GD), are described and evaluated. The region-based descriptor is evaluated against the contour-based shape descriptors in Sect. 4, and the paper concludes in Sect. 5.

2 Evaluation of contour-based shape descriptors

In this section, two contour-based shape descriptors, FD and CSSD, are described and evaluated. FD is described in Sect. 2.1, CSSD is described in Sect. 2.2, a comparison of CSSD and FD will be given in Sect. 2.3, and the evaluation results are discussed in Sect. 2.4.

2.1 Fourier Descriptor (FD)

In general, the FD is obtained by applying a Fourier transform on a *shape signature*. The set of normalized Fourier transformed coefficients is called the Fourier descriptor of the shape. The shape signature is a one-dimensional function, which is derived from shape boundary coordinates. Different shape signatures have been exploited to obtain FD. Complex coordinates, the curvature function, cumulative angular function, and centroid distance are the commonly used shape signatures. It has been shown [10] that a FD derived from the *centroid distance function* is more effective than a FD derived from other shape signatures.

The first step of computing a FD is to obtain the boundary coordinates $(x(t), y(t))$, t = 0, 1, …, N-1, where N is the number of boundary points. The *centroid distance function* is expressed by the distance of the boundary points from the centroid (x_c, y_c) of the shape

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{1/2}, \quad t = 0, 1, \ldots N-1$$

where

$$x_c = \frac{1}{N}\sum_{t=0}^{N-1} x(t) \quad y_c = \frac{1}{N}\sum_{t=0}^{N-1} y(t).$$

An example of a centroid distance function is shown in Fig. 2.

The discrete Fourier transform of $r(t)$ is then given by

$$a_n = \frac{1}{N}\sum_{t=0}^{N-1} r(t)\exp\left(\frac{-j2\pi\, n\, t}{N}\right),\ n = 0, 1, \ldots, N-1$$

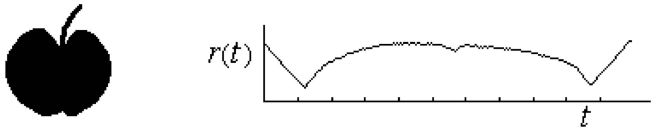


Fig. 2. An apple shape and its centroid distance function

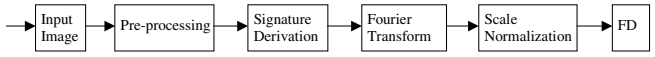


Fig. 3. Block diagram of computing FD

a_n are the Fourier transformed coefficients of $r(t)$. The acquired Fourier coefficients are translation invariant due to the translation invariance of the shape signature. To describe the shape, the acquired Fourier coefficients have to be further normalized so that they are rotation-, scaling- and start point-independent shape descriptors. From Fourier transform theory, the general form of the Fourier coefficients of a contour $r(t)$ generated through translation, rotation, scaling, and change of start point from the original contour $r(t)^{(o)}$ is given by:

$$a_n = \exp(jn\tau) \cdot \exp(j\phi) \cdot s \cdot a_n^{(o)}$$

where a_n and $a_n^{(o)}$ are the Fourier coefficients of the generated shape and the original shape, respectively, τ and ϕ are the angles incurred by the change of start point and the rotation, respectively; s is the scale factor. Now consider the following expression:

$$b_n = \frac{a_n}{a_0} = \frac{\exp(jn\tau) \cdot \exp(j\phi) \cdot s \cdot a_n^{(o)}}{\exp(j\tau) \cdot \exp(j\phi) \cdot s \cdot a_0^{(o)}} = \frac{a_n^{(o)}}{a_0^{(o)}} \exp[j(n-1)\tau] = b_n^{(o)} \exp[j(n-1)\tau]$$

where b_n and $b_n^{(o)}$ are the normalized Fourier coefficients of the generated shape and the original shape, respectively. The normalized coefficient of the derived shape b_n and that of the original shape $b_n^{(o)}$ have a difference of $\exp[j(n-1)\tau]$. If we ignore the phase information and use only the magnitude of the coefficients, then $|b_n|$ and $|b_n^{(o)}|$ are the same. In other words, $|b_n|$ is invariant to translation, rotation, scaling, and change of start point. The set of magnitudes of the normalized Fourier coefficients of the shape $\{|b_n|, 0 < n \leq N\}$ is used as the Fourier descriptor, denoted as $\{f_n, 0 < n \leq N\}$. Since the centroid distance is a real value function, only half of the coefficients are distinct, therefore only half of the FD features are needed to index the shape. Finally, a feature vector consisting of half of the normalized FD features is created to index each shape: $\mathbf{f} = \{f_1, f_2, \ldots, f_{N/2}\}$. The similarity between a query shape Q and a target shape T is determined by the *city block distance* d between their FDs: $d = \sum_{i=1}^{N/2} |f_i^Q - f_i^T|$ The whole process of computing the FD from a shape is given in Fig. 3.

In the implementation, 10 very complex shapes are selected from the database to simulate the worst convergence of the Fourier series of their boundary representations, the average spectrum of the 10 shapes show that 60 FD features are sufficient to describe the shape if FD features with normalized magnitude greater than 0.01 are taken as significant features. Based on this initial estimation, we test the retrieval performance using a different number of FD features (i.e., 5, 10, 15, 30, 60, 90) to find the appropriate number of FD features needed for shape description. It is found that the performance of retrieval using 15, 30, 60 and 90 features is almost the same. The retrieval performance only degrades slightly when using 10 FD features. The test reveals that when the number of FD features is above 15, the retrieval performance does not improve significantly with an increased number of FD features, and the retrieval performance does not degrade significantly when the number of FD features is reduced down to 10 FD features. The results suggest that for efficient retrieval, 10 FD features is sufficient for shape description [10]. This finding also reduces the computation of FD from $O(N^2)$ to $O(N)$ (N is the number of boundary points), because only 10 Fourier coefficients are needed.

2.2 Curvature Scale Space Descriptor (CSSD)

Mokhtarian et al. [13] propose the use of curvature scale space for shape retrieval. In this section, CSSD is described in detail. The computation of CSSD is given in algorithm forms to make it more convenient for implementation.

2.2.1 Computing Curvature Scale Space Descriptor

Basically, the CSS method treats shape boundary as a 1D signal, and analyzes this 1D signal in scale space. By examining zero crossings of curvature at different scales, the concavities/convexities of shape contour are found. These concavities/convexities are useful for shape description because they represent the perceptual features of shape contour.

The first step of the process is the same as that in computing FD – the output is the boundary coordinates $(x(t)y(t))$, $t = 0, 1, 2, \ldots, N-1$. The second step is scale normalization, which samples the entire shape boundary into a fixed number of points so that shapes with a different number of boundary points can be matched. The other two main steps in the process are the *CSS contour map* computation and *CSS peaks* extraction. The CSS contour map is a multi-scale organization of the inflection points (or curvature zero-crossing points). To calculate the CSS contour map, curvature is first derived from shape boundary points $(x(t)y(t))$, $t = 0, 1, 2, \ldots, N-1$:

$$k(t) = (\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t))/(\dot{x}^2(t) + \dot{y}^2(t))^{3/2} \qquad (1)$$

where $\dot{x}(t), \dot{y}(t)$ and $\ddot{x}(t), \ddot{y}(t)$ are the first and the second derivatives at location t, respectively. Curvature *zero-cross points* are then located in the shape boundary. The shape is then evolved into the next scale by applying Gaussian smoothing:

$$x'(t) = x(t) * g(t, \sigma), \quad y'(t) = y(t) * g(t, \sigma) \qquad (2)$$

where $*$ means convolution, and $g(t, \sigma)$ is *Gaussian function*. As σ increases, the evolving shape becomes smoother. New curvature zero-crossing points are located at each scale. This process continues until no curvature zero-crossing points are found. The evolution process is demonstrated in Fig. 4.

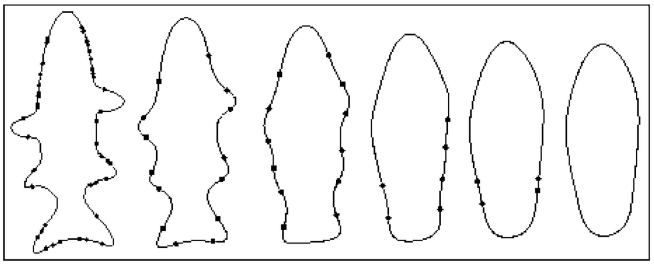


Fig. 4. The evolution of shape boundary as scale σ increases [13]. From left to right: σ = 1, 4, 7, 10, 12, 14. The points marked on the boundary are the inflection points

The CSS contour map is composed of all curvature zero-crossing points $zc(t, \sigma)$, where t is the location and σ is the scale at which the zc point is obtained. In practice, σ does not increase by integer value. Instead, it increases by fractional value, 0.01 for example. The acquired zero-crossing points are then plotted onto the (t, σ) plane to create the CSS contour map (Fig. 5 (middle)). The algorithm for computing the CSS contour map is given below.

Algorithm of computing CSS contour map

1. Normalize shape to a fixed number of boundary points;
2. Create an array $ZC[\ ][\ ]$ to record curvature zero crossing points at each scale;
3. Compute curvatures of each position t at current scale σ according to Eq. (1);
4. Record each curvature zero crossing point at current scale σ to $ZC[\sigma][t]$;
5. Smooth the boundary according to Eq. (2);
6. Repeat step 3–5 until no curvature zero crossing points are found;
7. Plot all curvature zero crossing points found in $ZC[\ ][\ ]$ onto Cartesian space to create CSS contour map.

The CSS contour branches are usually close on the top, with some exceptions of small gap (1 to 2 points) at the top of the branches. The peaks, or the local maxima of the CSS contour map (only those peaks higher than the threshold are considered) are then extracted out and sorted in descending order of σ. For example, the peaks of CSS contour map in Fig. 5 are (84, 249), (33, 215), (112, 64), (7, 52), (84, 39), (60, 37), (32, 22) (see Fig. 5 (right)). The peak locations are not readily available, and must be extracted from the CSS contour image through a separate process. The extraction algorithm is given in the following:

Algorithm of extracting CSS contour peaks

1. Scanning from the top row of CSS contour map;
2. If a zero-crossing point is found at a location (i, j), check the above neighbor points $(i-1, j-1)$, $(i-1, j)$ and $(i-1, j+1)$. If the three above neighbor points are non-zero-crossing points, then the location (i, j) is a peak candidate; find all the peak candidates in row i;
3. For each peak candidate (i, j) at row i, check its neighbor peak candidates, if a neighbor candidate (i, k) is found over five points away, then (i, j) is a peak. If a neighbor candidate is found within five points, there is a peak in the middle $(i, (j + k)/2)$;

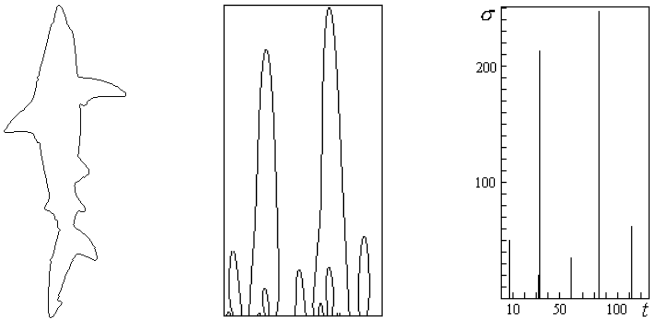


Fig. 5. A fish shape (left) and its CSS contour map (middle), CSS peak map (right)

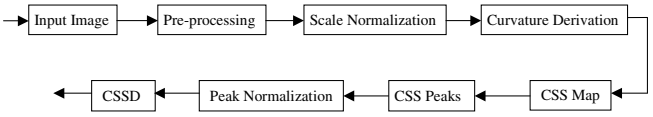


Fig. 6. Block diagram of computing CSSD

4. Repeat steps 2 and 3 for each row, until all the CSS peaks are found.

The next step is to normalize all the obtained CSS peaks. The average height of all the peaks extracted from the database is used for peak normalization. Finally, the normalized CSS peaks are used as CSS descriptors to index the shape. For convenience, here the CSS peak map will be used to illustrate CSSD.

The whole process of computing CSSD is shown in Fig. 6.

2.2.2 Matching CSS descriptor

The CSS descriptor is translation invariant. Scale invariance is achieved by normalizing all the shapes into a fixed number of boundary points. In our implementation, this number is 128 points. Since rotation of shape causes circular shifting of CSS peaks on the t axis, the rotation invariance is achieved by circular shifting the highest peak (*primary peak*) to the origin of the CSS map. The similarity between two shapes A and B is then measured by the summation of the peak differences between all the matched peaks and the peak values of all the unmatched peaks [13]. To increase robustness, four schemes of circular shifting matching are applied to tolerate variations of peak heights of potential matching peaks (more schemes of circular shift matching can be applied to obtain more accurate matching). The four schemes of shift matching are: shifting the primary peak of A (other peaks of A are shifted accordingly) to match the primary peak of B; shifting primary peak of A to match the *secondary peak* (second highest CSS peak) of B; shifting the secondary peak of A to match the primary peak of B; shifting the secondary peak of A to match the secondary peak of B. Since a mirror shape has different CSS descriptors from the original shape, the matching has to include the mirrored shape matching. Altogether, eight schemes of circular shift matching are needed to fulfil the matching between two sets of CSS descriptors. The fact that the corresponding peaks of two similar shapes are usually not matched exactly

indicates that matching between two sets of CSS descriptors also needs to accept a certain tolerance of position variation between two potentially corresponding peaks. In the implementation, this tolerance value is 5% of all of the boundary points, which means that if two peaks are within 7-point distance they are matched, otherwise they are not matched. The matching algorithm is given as follows:

Algorithm of matching two sets of CSSD

1. Shift the primary peak of both sets to the left most. Call MATCH procedure to calculate matching cost for this match $Dist_{pp}$.
   MATCH:
   Start from the left most; if a peak in set 1 and a peak in set 2 are within a horizontal distance $d_i < 7$, they are a matched pair (p_i^1, p_i^2). Peaks in the two sets, which are not matched pairs are 'singles' denoted as s_j. The matching cost for this match is then:

$$Dist(CSSD1, CSSD2) = \sum_i (|p_i^1 - p_i^2| + d_i) + \sum_j s_j$$

2. Shift the primary peak of set 1 to the left most and shift the secondary peak of set 2 to the left most. Call MATCH procedure to calculate matching cost for this match $Dist_{ps}$.
3. Shift the secondary peak of set 1 to the left most and shift the primary peak of set 2 to the left most. Call MATCH procedure to calculate matching cost for this match $Dist_{sp}$.
4. Shift the secondary peak of set 1 to the left most and shift the secondary peak of set 2 to the left most. Call MATCH procedure to calculate matching cost for this match $Dist_{ss}$.
5. Match set 1 and mirror set of set 2 using steps 1–4. The matching cost for this match is $Dist_{mirror}$.
6. The distance between the two sets of CSSD is $Dist(CSSD1, CSSD2) = \min\{Dist_{pp}, Dist_{ps}, Dist_{sp}, Dist_{ss}, Dist_{mirror}\}$

2.3 Comparison of FD and CSS Descriptor

In this section, a comparison of retrieval performance and computational efficiency of FD and CSSD is given in detail.

2.3.1 Comparison of retrieval effectiveness

To test the retrieval performance of the FD and CSSD, a Java-based indexing and retrieval framework which runs on a Windows platform is implemented. The retrieval test is conducted on an MPEG-7 contour shape database [30]. The MPEG-7 contour shape database consists of shapes acquired from real world objects. It takes into consideration common shape distortions in nature and the inaccurate nature of shape boundaries from segmented shapes. It is designed to test a contour shape descriptor's behavior under different shape distortions. The database consists of three parts: Sets A, B and C. Set A has two parts, Set A1 and Set A2, each consisting of 420 shapes of 70 classes. Set A1 is for testing scale invariance, and Set A2 is for testing rotation invariance. Set B has 1400 shapes, which have been classified into 70 classes. Set B is for testing similarity-based retrieval and the shape descriptors' robustness to various arbitrary shape distortions. Set C consists of 200 affine transformed bream fish and 1100 marine fish, which are unclassified. The 200 bream fish are designated as queries. Set C is for testing the shape descriptors' robustness to non-rigid object distortions. Since all of the member IDs in each class of the sets are known, the retrieval is conducted automatically.

Commonly used performance measurement, i.e., a precision and recall pair, is used to evaluate the query result [39]. Precision P is defined as the ratio of the number of retrieved relevant shapes r to the total number of retrieved shapes n, i.e. $P = r/n$. Precision P measures the accuracy of the retrieval. Recall R is defined as the ratio of the number of retrieved relevant images r to the total number m of relevant shapes in the whole database, i.e., R= r/m. Recall that R measures the robustness of the retrieval performance. For Sets A and B, all shapes in the sets are used as queries. For Set C, the 200 bream fish are used as queries. For each query, the precision of the retrieval at each level of the recall is obtained. The final precision of retrieval using a shape descriptor is the average precision of all the queries. The average retrieval precision and recall using FD and CSSD are shown in Fig. 7a–d. Online retrieval using the two contour shape descriptors can be accessed at http://www.gscit.monash.edu.au/∼dengs.

It can be seen from the precision recall charts that FD outperforms CSSD significantly on the performance of scaling, rotation, affine, and similarity retrieval, indicating that FD is more robust to general boundary variations than CSSD. In the experiments, it has been found that CSSD robustness to boundary variations is very limited. It is not robust to common boundary variations such as defections and distortions. For example, in the database, there are occluded apple shapes for testing occlusion retrieval. The two occluded apple shapes are both retrieved in the first screen (Fig. 8a) using FD; the ranks of the two occluded apple are 5 and 13, respectively. The CSSD fails to retrieval any of the occluded apples in the first 36 retrieved shapes (Fig. 8b), four example apple shapes and their CSSD are shown in Fig. 9a–d. The CSSD also has very poor performance on the fork shape (Fig. 8e), while FD has very high performance on this shape (Fig. 8d). CSSD is easily trapped by shapes with five prominent protrusions. Four example fork shapes and their CSSD are shown in Fig. 10a–d.

From Figs. 9 and 10, it can be seen that CSSD is able to preserve the number of convexity features on the boundary in the presence of distortions (Fig. 9a,d and Fig. 10a,b,d). However, defections add new peaks to the map (Figs. 9b,c and 10c), which consequently add net cost to the matching result. The peak heights change drastically in the presence of distortions (Figs. 9d and 10c,d); in particular, the peak positions have changed so significantly that they cannot be matched properly by circular shift in many cases. For example, the two highest peaks of Fig. 9c will not be matched to the two highest peaks in Fig. 9d, because the difference between the gaps of the two highest peaks in the two peak maps is greater than 7. Similarly, the two highest peaks in Fig. 10a will be out of match with the peaks in Fig. 10c,d.

Even though the number of peaks of two CSSDs (of two similar shapes) is the same and there is a match between the two highest peaks in horizontal positions – for example, in the case of Fig. 9a,d – they are very different descriptors after

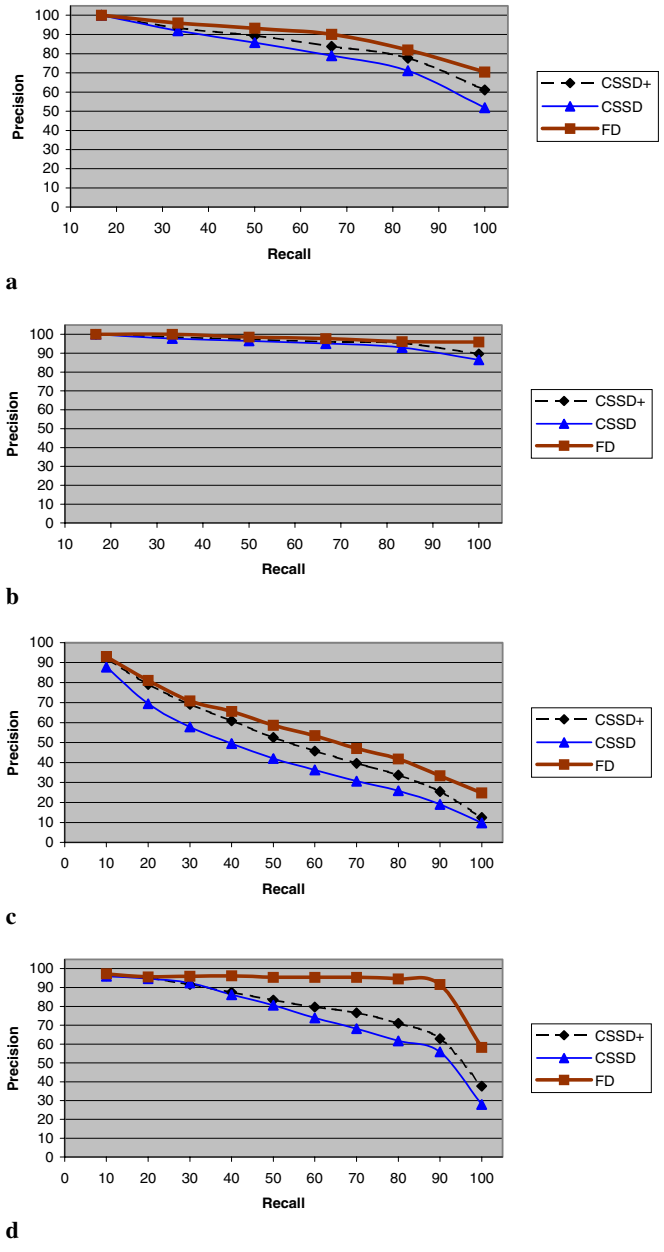


Fig. 7a–d. Average precision and recall of retrieval using FD, CSSD and CSSD+ on MPEG-7 contour shape database CE-1. **a** Average precision-recall of 420 retrievals using FD, CSSD and CSSD+ on Set A1. **b** Average precision-recall of 420 retrievals using FD, CSSD and CSSD+ on Set A2. **c** Average precision-recall of 1400 retrievals using FD, CSSD and CSSD+ on Set B. **d** Average precision-recall of 200 retrievals using FD, CSSD and CSSD+ on Set C

normalization, due to the different order of the height of the two peaks (especially when the height of the two highest peaks has a large difference). The increase of peaks and mismatch of peaks adds a heavy cost to the matching result, effectively resulting in false retrievals.

In recognizing the problem of sensitivity of CSSD to local variations, MPEG-7 also recommends combing CSSD with global shape descriptors such as eccentricity and circularity to form a more robust shape descriptor (the weights of the global

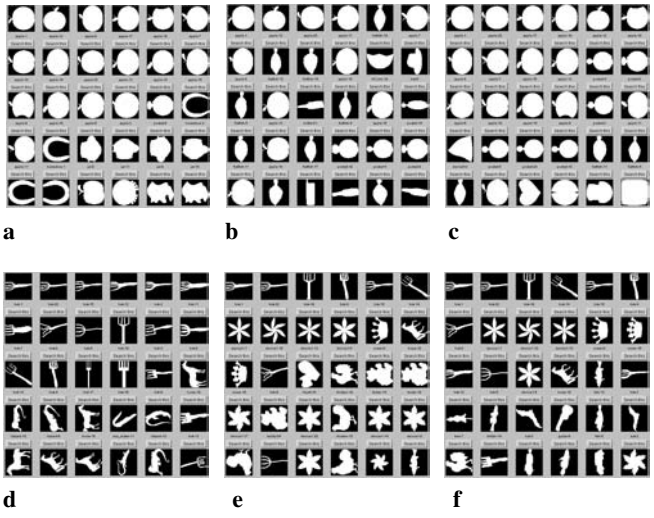


Fig. 8a–f. Retrieval of apple shapes using **a** FD; **b** CSSD; **c** CSSD+. Retrieval of fork shapes using. **d** FD; **e** CSSD; **f** CSSD+. In all the screen shots, the top left shape is the query shape and the retrieved shapes are arranged in descending order of similarity to the query. The screen shots are retrieval examples from Set B

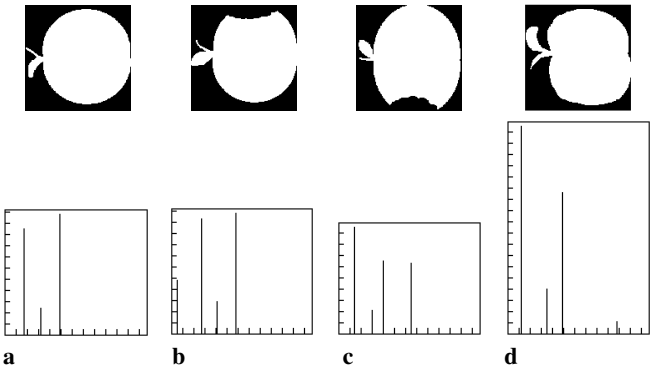


Fig. 9a–d. Four apple shapes on the top and their corresponding CSSD at the bottom

descriptors are provided in the document) [35]. The combined shape descriptor is denoted as CSSD+. The retrieval performance of CSSD+ is also shown in Fig. 7 and the corresponding retrievals for the above three queries are shown in Fig. 8c,f. It can be observed that CSSD+improves CSSD, however its retrieval performance on all the sets is still lower than FD.

2.3.2 Comparison of computational efficiency

To compare the computational efficiency of the two shape descriptors, the feature extraction and retrieval are tested on the Windows platform of a Pentium III-866 PC with 256M memory. The average time taken for the feature extraction and retrieval on Set B of the MPEG-7 contour shape database is given in Table 1. It can be seen from Table 1 that FD is much more efficient, especially in terms of average retrieval time. On average, the retrieval time of FD is less than one-third the retrieval time of CSSD.

Table 1. The elapsed time of feature extraction and retrieval for 1400 shapes

Time / Shape descriptors	Total time of feature extraction of 1400 shapes	Average time of feature extraction of each shape	Total time of retrieval of 1400 queries	Average time of retrieval of each query
FD	80960 ms	57.8 ms	49894 ms	35.6 ms
CSSD	120629 ms	86.1 ms	163570ms	116.8ms

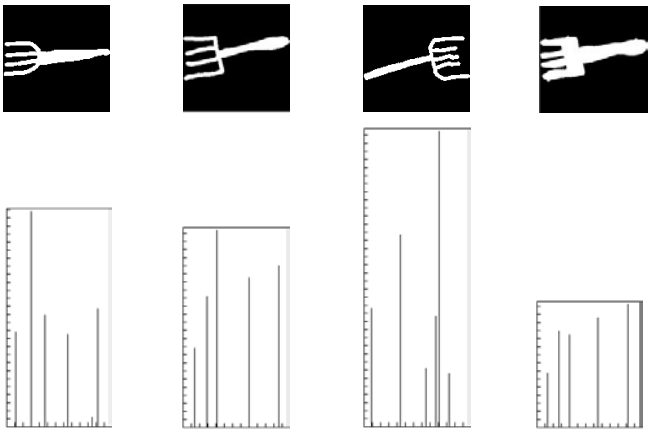


Fig. 10a–d. Four fork shapes on the top and their corresponding CSSD at the bottom

2.4 Discussion

In the previous section, the two contour shape descriptors FD and CSSD are evaluated in detail. The contrast of the two descriptors is given here. The similarities between FD and CSSD are as follows:

- Both FD and CSSD are robust to boundary noise. With FD, the more significant lower frequencies preserve shape global structures which are robust to noise on the boundary. Noise influence is eliminated through truncation of high frequencies. With CSSD, higher peaks capture merged convexities (concavities) which are robust to noise on the boundary. Noise influence is eliminated through thresholding out short peaks.
- Both representations are compact. The number of FD features needed to describe shape is 10, while the average number of CSSD features needed to describe shape is 8 including global descriptors.

The differences between FD and CSSD are as follows:

- Feature domains. A FD is obtained from a spectral domain while CSSD is obtained from a spatial domain.
- Dimensions. The dimension of FD is constant (once the number of coefficients is chosen), while that of CSSD varies for each shape.
- Feature extraction complexity. The computation process of CSSD is more complex than that of FD. The computation of CSSD requires scaling normalization before CSSD extraction, and the extraction of the CSSD feature takes three processes, i.e., CSS map computation, height adjusted CSS map computation, and CSS peaks extraction.
- Online matching computation. The online matching of two sets of FDs is simply the city block distance between two feature vectors of 10 dimensions. The online matching of two sets of CSSD involves at least eight schemes of circular shift matching, and for each scheme of circular shift matching, it involves eight shifts and the city block distance calculation between two feature vectors of eight dimensions.
- Type of features captured. FD captures both global and local features, while CSSD captures only local features.
- Parameters or thresholds influence. For FD, the only parameter is the number of FD features, which is predictable (Sect. 2.1). For CSSD, the parameters are the number of sampling points, the threshold to eliminate short peaks, the tolerance value for peak position matching and the database dependent value used for peak height normalization. The parameters are determined empirically. The parameter difference indicates that FD is more stable than CSSD when they are applied to different applications.
- Hierarchical representation. FD supports hierarchical coarse to fine representation while CSSD does not. To support hierarchical representation, CSSD has to incorporate global shape features such as eccentricity and circularity, which are unreliable.
- Suitability for efficient indexing. FD is suitable to be organized into an efficient data structure, while CSSD is not, due to its variable dimensions and complex distance calculation.

3 Evaluation of region-based shape descriptors

In this section, three region-based shape descriptors, GMD, ZMD, and GD, are described and evaluated. GMD, ZMD, and GD are described in Sects. 3.1– 3.3, respectively. A comparison of the three shape descriptors is given in Sect. 3.4, and evaluation results are discussed in Sect. 3.5.

3.1 Geometric Moment Descriptor (GMD)

The technique based on moment invariants for shape representation and similarity measure is extensively used in shape recognition. Moment invariants are derived from moments of shapes, and are invariant to 2D geometric transformations of shapes. The central moments of order $p + q$ of a two-dimensional shape represented by function $f(x, y)$ are given by

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) p, q = 0, 1, 2... \tag{3}$$

where $\bar{x} = \mu_{10}/m$, $\bar{y} = \mu_{01}/m$ and m is the mass of the shape region. μ_{pq} are invariant to translation. The seven moment invariants were derived by Hu [25]: $\Phi_1, \Phi_2, \ldots \Phi_7$.

A feature vector consisting of the seven moment invariants, $\mathbf{f} = (\Phi_1, \Phi_2, \ldots \Phi_7)$, is used to index each shape in the database. The values of the computed moment invariants are usually small – values of higher order moment invariants are close to zero in some cases. Besides, there are always outliers in the feature values, i.e., abnormal values generated due to noise or other uncertain factors. Therefore, the acquired invariants need to be further normalized. Several normalization methods can be used, including min-max normalization, zscore normalization, and sigmoidal normalization. Our results show that zscore produces the best result. Basically, zscore normalization translates the input variable data so that the mean is zero and the variable is one [40]. It computes the mean and standard deviation of the input data, and then transforms each input value by subtracting the mean and dividing by the standard deviation. In mathematical form, it is given by

$$y' = \frac{y - mean}{std} \tag{4}$$

where y is the original value, y' is the new value, and the *mean* and *std* are the mean and standard deviation of the original range, respectively. Since zscore normalization is based on the standard deviation of the example population, it is especially suitable for the situation where there are outliers in feature values. The commonly used min-max normalization is easily affected by outliers.

The advantage of using GMD is that it is a very compact shape representation and the computation is low.

3.2 Zernike Moment Descriptor (ZMD)

Teague [27] proposed the use of orthogonal moments to recover the image from moments based on the theory of orthogonal polynomials, and introduced Zernike moments, which allow independent moment invariants to be constructed to an arbitrarily high order. The complex Zernike moments are derived from Zernike polynomials:

$$V_{nm}(x, y) = V_{nm}(\rho\cos\theta, \rho\sin\theta) = R_{nm}(\rho)\exp(jm\theta) \tag{5}$$

where

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!(\frac{n+|m|}{2}-s)!(\frac{n-|m|}{2}-s)!}\rho^{n-2s} \tag{6}$$

where ρ is the radius from (x, y) to the shape centroid, θ is the angle between ρ and the x-axis, and n and m are integers and subject to $n - |m|$ = *even*, $|m| \leq n$. Zernike polynomials are a complete set of complex-valued function orthogonals over the unit disk, i.e., $x^2 + y^2 = 1$. The complex Zernike moments of order n with repetition m are then defined as:

$$A_{nm} = \frac{n+1}{\pi}\sum_x\sum_y f(x, y)V^*_{nm}(x, y), \quad x^2 + y^2 \leq 1 \tag{7}$$

where * means complex conjugate. Due to the constraint of $n - |m|$ = *even* and $m < n$, there are $n/2$ repetitions of moments in each order n.

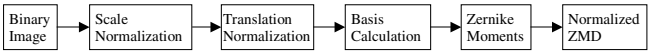


Fig. 11. Block diagram of computing ZMD

Since Zernike basis functions take the unit disk as their domain, this disk must be specified before moments can be calculated. In our implementation, all shapes are normalized into a unit circle of fixed radius of 64 pixels. The unit disk is then centered on the *center of mass* of the shape. This makes the moments obtained both scale and translation invariant. Rotation invariance is achieved using only the magnitudes of the moments. The magnitudes are then normalized into [0, 1] by dividing them by the mass of the shape. The similarity between two shapes indexed with Zernike moments descriptors is determined by the city block distance between the two Zernike moments vectors. A block diagram of the whole process of computing ZMD is shown in Fig. 11.

The theory of Zernike moments is similar to that of Fourier transform, to expand a signal into a series of orthogonal basis. However, the computation of a Zernike moments descriptor does not need to know boundary information, making it suitable for more complex shape representation. Like the Fourier descriptor, Zernike moment invariants can be constructed to arbitrary order, thus overcoming the drawback of geometric moment in which higher order moment invariants are difficult to construct. The precision of shape representation depends upon the number of moments truncated from the expansion. For efficient retrieval, the first 36 moments of up to order 10 are used in our implementation.

3.3 Grid Descriptor (GD)

The grid descriptor was proposed by Lu and Sajjanhar [32]. It has been used in MARS [31] and other applications [33]. When Lu and Sajjanhar proposed the grid method, it was only applied to contour-based shape, and this convention is also followed by Chakrabarti et al. [31] and Safar et al. [33]. In this section, it is improved to describe both contour and region shape.

3.3.1 Grid Method

In grid shape representation, a shape is projected onto a grid of fixed size, 16×16 grid cells, for example. The grid cells are assigned the value of 1 if they are covered by the shape (or covered beyond a threshold), and 0 if they are outside the shape. A shape number consisting of a binary sequence is created by scanning the grid in a left–right and top–bottom order, and this binary sequence is used as the shape descriptor to index the shape.

For two shapes to be comparable using grid descriptors, several normalization processes have to be done to achieve scale, rotation, and translation invariance. A block diagram of computing grid descriptor for a contour-based shape is given in Fig. 12.

It begins with finding out the major axis (MA), i.e., the line joining the two furthest points on the boundary. Rotation

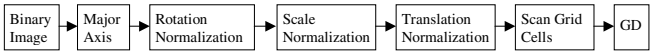


Fig. 12. Block diagram of computing GD

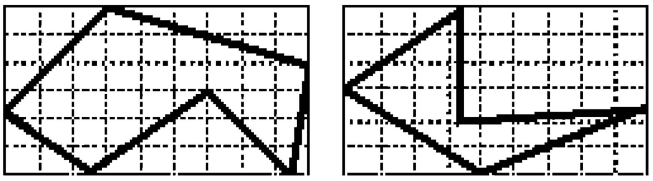


Fig. 13. Grid representation of two shapes

normalization is achieved by turning the shape so that the major axis is parallel with the x-axis. Scale normalization can be done by resizing the shape so that the length of the major axis is equal to the preset grid width, and by shifting the shape to the upper-left of the grid, the representation is translation invariant. The next step is scanning the grid cells so that a binary value is calculated for each cell based on the coverage of the cell by the shape boundary. Finally, a binary sequence is generated as the shape descriptor. The distance between two sets of grid descriptors is simply the number of elements having different values. For example, the grid descriptors for the two shapes in Fig. 13 are 001111000 011111111 111111111 111111111 111110011 001100011 and 001100000 011100000 111100000 111100000 011111100 000111000, respectively, and the distance between the two shapes will be 27 by XOR operation on the two sets.

Since horizontally flipped and vertically flipped shapes will have representations different to the original shape even after normalization, the matching has to take into consideration the two types of flipped shapes. To avoid multi-normalization results for mirrored and flipped shapes, the centroid of the rotated shape may be restricted to the lower-left part, or a mirror and a flip operation on the shape number are applied in the matching stage.

3.3.2 Improving the grid method for region shape

The above GD computing algorithm is for contour-based shape, and it assumes That shape boundary coordinates have been known. In this section, it is extended into describing region shape. The main improvement to the grid method is the major axis finding and region interpolation after scale and rotation.

Normally, the major axis is found by traversing all the points on the shape Boundary, and the line joining the two boundary points with the furthest distance is the major axis. However, for region shape, boundary information is not known *a priori*. It is impossible to find the major axis of a region shape by traversing all the points in the shape region – the computation would be $O(N^2)$ (N is the number of image pixels). Therefore, an optimized algorithm for finding an approximated major axis is proposed. The optimized *major axis algorithm* (MAA) involves three steps [41]: (i) finding the bounding box of the shape; (ii) finding the pair of boundary points in a number of directions (360 in our case); and (iii) finding the two points of the furthest apart in the found boundary points. This reduces the MA computation to less than $O(N)$.

An interpolation process is needed for rotation normalization, because after arbitrary angle rotation, the region points are scattered. A similar interpolation is also needed for the scale normalization (interpolation is not needed for contour shape, because contour shape does not consider interior content – all the interior point values are the same as the boundary point value). An 8×8 nearest neighbor interpolation technique is used to fix up the rotation impairment. Specifically, if the number of shape pixels within the 8×8 neighborhood of a pixel under consideration is greater than 15, then the pixel under consideration is reinstated as a shape pixel. Scale interpolation is achieved by spreading or shrinking the current pixel along the x and y directions to the new scaled coordinates.

3.4 Comparison of Geometric Moment, Zernike Moment and Grid descriptors

In this section, a comparison of retrieval performance and computational efficiency of GMD, ZMD, and GD is given in detail.

3.4.1 Comparison of retrieval effectiveness

In this section, we compare the retrieval performance of the three shape descriptors. Since region-based shape descriptors can be applied to both contour and region shape, two sets of experiments are carried out. One test is carried out on the MPEG-7 contour shape database CE-1, and the other test is carried out on the MPEG-7 region shape database CE-2. CE-1 has been described in Sect. 2.3. CE-2 has been organized by MPEG-7 into six datasets (Sets A1, A2, A3, A4, B) and the whole database CE-2. CE-2 is designed to test a region shape descriptor's behavior under different shape variations. The use of each data set in the region shape database is given in detail in the following:

- Set A1 consists of 2881 shapes from the whole database, and it is used for testing scale invariance. In Set A1, 100 shapes have been organized into 20 groups (five similar shapes in each group), which can be used as queries for test the retrieval. In our experiment, all 100 shapes from the 20 groups are used as queries to test the retrieval.
- Set A2 consists of 2921 shapes from the whole database, and it is used for testing rotation invariance. In Set A2, 140 shapes have been organized into 20 groups (seven similar shapes in each group), which can be used as queries to test the retrieval. In our experiment, all 140 shapes from the 20 groups are used as queries to test the retrieval.
- Set A3 consists of 3101 shapes from the whole database, and it is used for testing rotation/scale invariance. In Set A3, 330 shapes have been organized into 30 groups (11 similar shapes in each group), which can be used as queries to test the retrieval. In our experiment, all 330 shapes from the 30 groups are used as queries to test the retrieval.
- Set A4 consists of 3101 from the whole database, and it is used for testing robustness to perspective transform. In Set A4, 330 shapes have been organized into 30 groups (11 similar shapes in each group), which can be used as queries

to test the retrieval. In our experiment, all 330 shapes from the 30 groups are used as queries to test the retrieval.

- Set B consists of 2811 shapes from the whole database, and it is used for subjective test. In Set B, 682 shapes have been manually sorted out into 10 classes by MPEG-7. The number of similar shapes in each class is, respectively, 68, 248, 22, 28, 17, 22, 45, 145, 45, and 42. In our experiment, all 682 shapes from 10 classes are used as queries to test the retrieval.
- The whole database consists of 3621 shapes, 651 of which have been organized into 31 groups (21 similar shapes in each groups). For the 21 similar shapes in each group, there are 10 perspective-transformed shapes, 5 rotated shapes and 5 scaled shapes. The 31 groups of shapes reflect overall shape operations, and they test the overall robustness of a shape descriptor. The whole database is 17–29% larger in size than the individual sets.

Each shape in the individual data set of the two databases is indexed using the three described region shape descriptors. The test methods are the same as those used in Sect. 2.3, e.g., the retrieval is carried out both automatically and online. Online retrieval using these three region shape descriptors can be accessed at http://www.gscit.monash.edu.au/∼dengs/.

The precision recall is used for evaluation of retrieval effectiveness. For each query, the precision of the retrieval at each level of the recall is obtained. The final precision of retrieval using a shape descriptor is the average precision of all the query retrievals using that shape descriptor. The average precision and recall of the retrieval on each data set are shown in Figs. 14a–d and 15a–f. Some screen shots are shown in Fig. 18.

From the precision-recall charts in Figs. 14 and 15, it can be seen that ZMD outperforms GMD and GD in all the data sets tested. For simple shape transformation such as scaling, rotation, and affine transform, ZMD has a much higher performance (Figs. 14a,b,d and 15a–c). It also has robust performance for generic shape variations (Figs. 14c and 15f). On average, its retrievals are more perceptually acceptable than GMD and GD (Fig. 15e). However, ZMD has an intrinsic problem for a perspective transformed shape or a shape with relatively large stretching. This is because of the concentric circular scanning method it uses during the sampling process. For a perspective-transformed shape, when the scanning moves from the center of the shape to the periphery, it encounters more positions without shape information. This is contrasted with a non-transformed shape, where all the scanned positions contain shape information. As can be expected, the derived ZMD for a perspective transformed shape and a non-transformed shape will be quite different because a different amount of information has been used. The concentric circular sampling problem causes a significantly lower retrieval effectiveness for perspective-transformed shapes and generally distorted shapes compared with retrieval in other sets (Fig. 15d,f). This problem will be examined in future research.

GMD generally has robust performance for a simple contour shape or a shape with simple transformations such as scaling, rotation, and affine transformation (Fig. 14a,b,d), shapes in Sets A1, A2, and C are generally simple compared with shapes in Set B). GMD even hits a 100% retrieval precision for rotated contour shapes (Fig. 14b). For complex shapes and

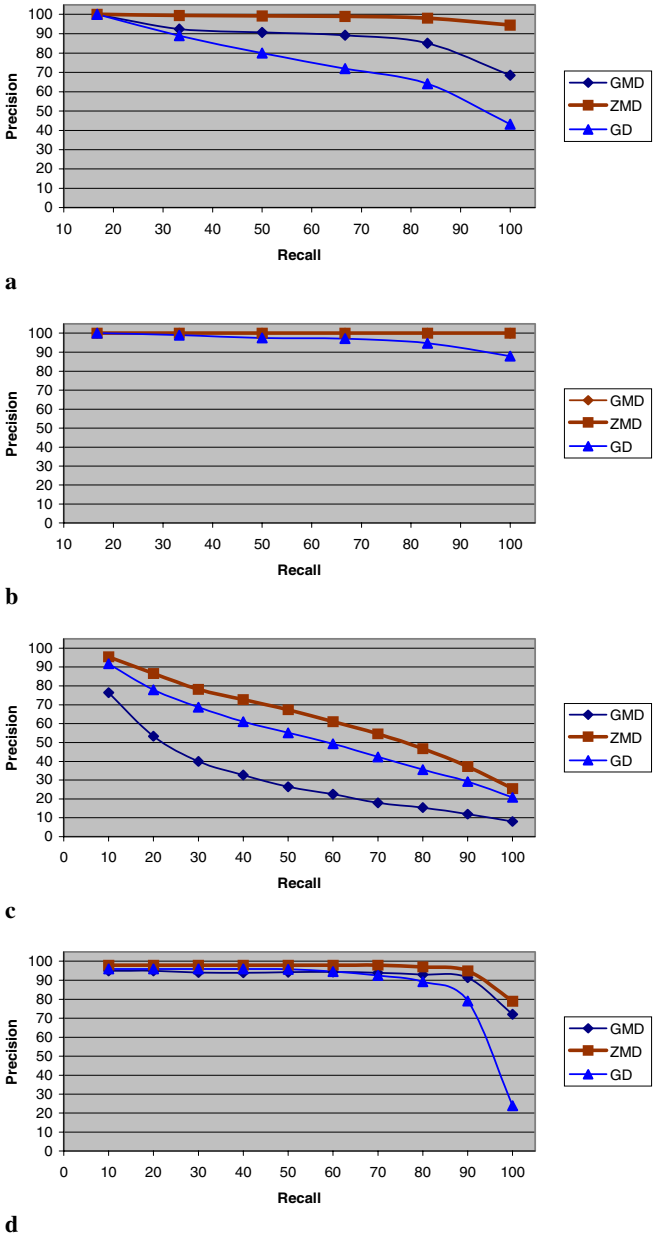


Fig. 14a–d. Average precision-recall of the three region-based shape descriptors on MPEG-7 contour shape database. **a** Average precision-recall of 420 retrievals using three region shape descriptors on Set A1 of CE-1. **b** Average precision-recall of 420 retrievals using three region shape descriptors on Set A2 of CE-1. Both GMD and ZMD have 100% retrieval precision. **c** Average precision-recall of 1400 retrievals using three region shape descriptors on Set B of CE-1. **d** Average precision-recall of 200 retrievals using three region shape descriptors on Set C of CE-1

shapes with generic variations, it cannot describe shape as accurately (Fig. 14c). This is also supported by its retrieval performance on region shapes. For example, in the region shape database test, it only produces satisfactory performance on rotation (Fig. 15b), while on all the other data sets, it performs poorly. It is notable that GMD is very sensitive to scaling for region shape (Fig. 15a). This is because region shape usually has rich interior content, so when it is under distortion its con-

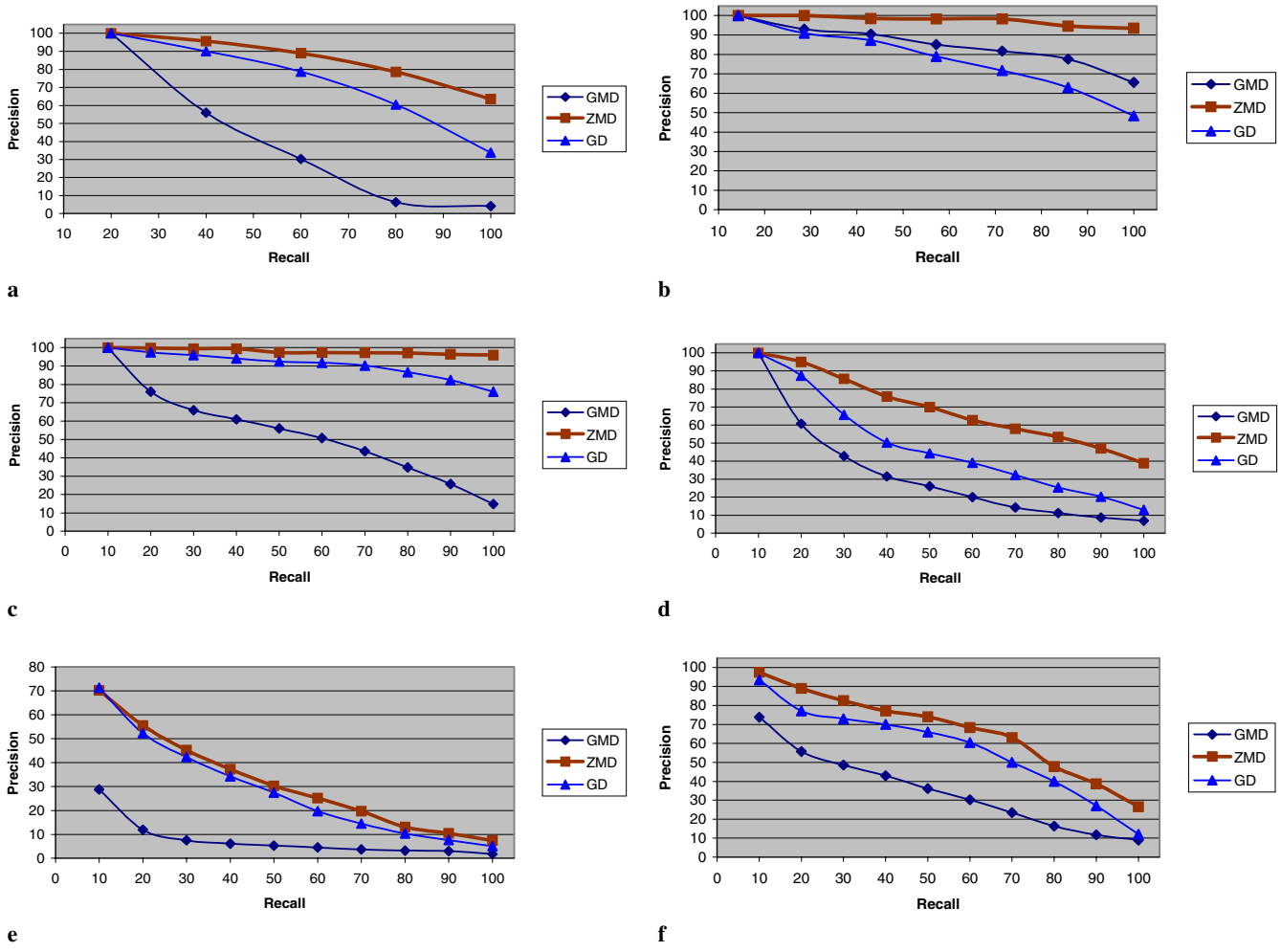


Fig. 15a–f. Average precision-recall of the three region-based shape descriptors on MPEG-7 region shape database. **a** Average precision-recall of 100 retrievals using three region shape descriptors on Set A1 of CE-2. **b** Average precision-recall of 140 retrievals using three region shape descriptors on Set A2 of CE-2. **c** Average precision-recall of 330 retrievals using three region shape descriptors on Set A3 of CE-2. **d** Average precision-recall of 330 retrievals using three region shape descriptors on Set A4 of CE-2. **e** Average precision-recall of 682 retrievals using three region shape descriptors on Set B of CE-2. **f** Average precision-recall of 651 retrievals using three region shape descriptors on CE-2

Table 2. Time of feature extraction and retrieval of CE-2 shapes using region shape descriptors

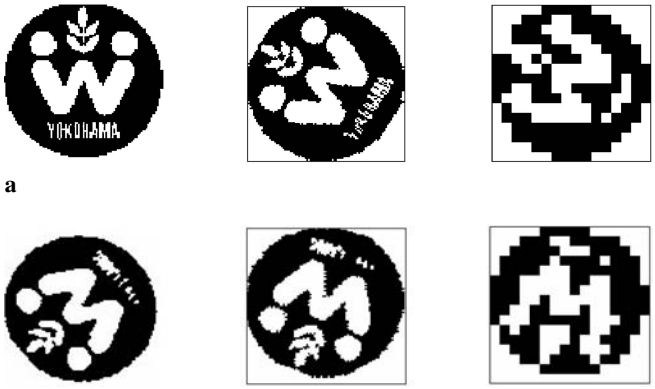
Time Descriptor	Total time of feature extraction of 3621 shapes (*ms*)	Average time of feature extraction of each shape (*ms*)	Total time of retrieval of 651 queries (*ms*)	Average time of retrieval of each query (*ms*)
ZMD	4325010	1194.4	63854	98
GD	2628034	725.7	729909	1121.2
GMD	748176	206.6	33380	51.2

tent changes substantially. This is difficult for GMD to deal with because it is only derived from the three lower order moments. For example, on the right of Fig. 18c, GMD can only retrieve shapes of a similar size. If the first shape in this group is used as a query, it cannot retrieve any of the other shapes in the group, because the first shape is seven times smaller than the other shapes in this group. Scaling is not a problem for GD and ZMD (see the right of Fig. 18a,b). Perceptually, GMD is poor in describing general shapes. This can be observed in the subjective retrieval (Fig. 15e), where GMD has an unacceptably lower performance. It can be said that GMD is suitable for describing simple shape – during retrieval, GMD can usually retrieve shapes that are similar to the query shape.

For simple shapes and shapes under simple transformations, GD is outperformed by GMD (Figs. 14a,b,d and 15b), because GMD is good at describing simple shape and shape with simple transformation. However, GD is more accurate in describing complex shape, and is much more robust to generic shape variations compared with GMD (Figs. 14c, 15a,c–f). From Fig. 15e, GD is also far more perceptually accurate than GMD. Perceptually, GD's description capability is comparable with ZMD. The main problem associated with GD is its

a **b**

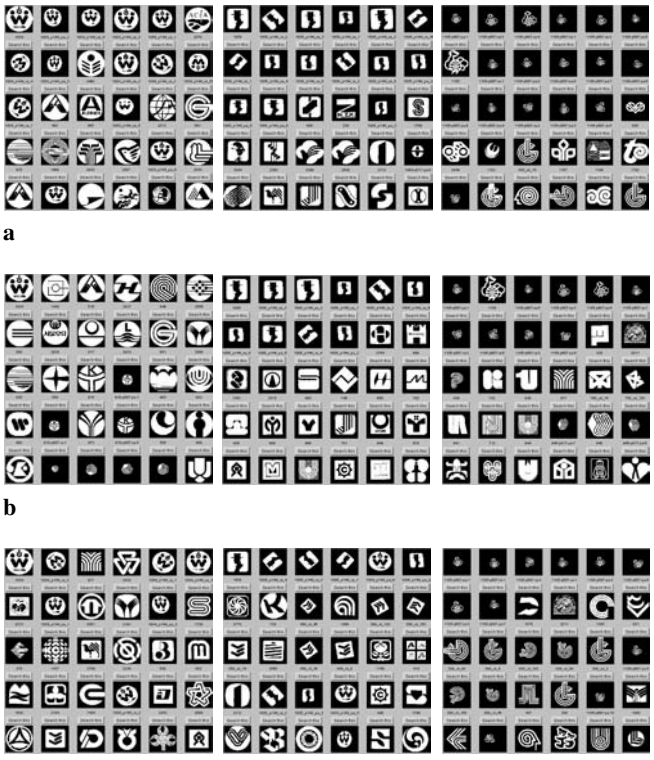
Fig. 16a,b. Two similar shapes with very different grid representation. **a** a nail shape (*left*) and its grid representation (*right*); **b** another nail shape (*left*) and its grid representation (*right*)



a

b

Fig. 17a,b. From left to right: original shape, normalized shape and grid representation



a

b

c

Fig. 18a–c. Screen shots of shape retrieval using **a** ZMD; **b** GD; **c** GMD. In all the screen shots, the top left shape is the query shape. The retrieved shapes are arranged in descending order of similarity to the query

problematic major axis normalization. The major axis is sensitive to noise and can be unreliable even if there is no noise. For example, the two shapes in Fig. 16a,b are perceptually similar, but are very different under grid representation, as the major axis of shape (a) is horizontal, while the major axis of shape (b) is vertical. On the other hand, if the major axis is reliable for a type of shapes, the retrieval can be quite accurate. For example, in Fig. 18b, the second and third retrievals are quite accurate, because the major axis for these two shapes is reliable. The rotation normalization does not guarantee interior rotation invariance. For example, the rotation normalization does not work for the two region shapes in Fig. 17. This is also reflected in the retrieval (see the left of Fig. 18b). The problem caused by major axis normalization also explains why GD is easily outperformed by GMD in the retrieval of rotated shapes (Figs. 14b and 15b). The accuracy of shape representation using GD also depends upon the cell size and the threshold to determine the binary value of a cell based on its coverage by the shape. The online retrieval usually involves high computation due to the high dimensionality of the feature vectors (for a shape of 192×192 pixels using cell size of 12×12 pixels, the dimension is 196).

The main reason why ZMD is more robust than GMD and GD is that ZMD captures spectral features in circular directions. The spectral feature is robust to noise and shape variations, while GMD and GD only capture features in the spatial domain that is sensitive to noise and other variations. Besides, ZMD does not have the rotation problem in GD because it uses polar space, and ZMD does not have the scaling problem in GMD because it uses more moment features.

3.4.2 Comparison of computational efficiency

To study the efficiency of the three region-based descriptors during the feature extraction and online matching, we test feature extraction and shape matching using MPEG-7 region shape database CE-2. All shapes in CE-2 are used to calculate the average feature extraction time, and the 651 classified shapes are used as queries to calculate the average online retrieval time. The time taken for feature extraction and retrieval on CE-2 using the three region shape descriptors are given in Table 2. It can be seen from Table 2 that both ZMD and GD involve more expensive offline computation than GMD, while the online matching of GD is the most expensive among the three region shape descriptors.

3.5 Discussion

In the previous subsections, ZMD, GD, and GMD are described and studied in detail. The comparison of the three region-based shape descriptors is given in the following:

- Feature domains. ZMD captures circular features from the spectral domain, while GD and GMD are only extracted from the spatial domain.
- Compactness. The dimensions of GMD and ZMD are low, while the dimensions of GD are high.
- Robustness. Based on the precision recall, ZMD is most robust to shape variations among the three region-based

shape descriptors. GD is more robust than GMD for complex shapes and shapes with arbitrary variations. GMD is only more robust than GD for simple shapes under simple transformations.

- Computation complexity. The extraction of GD and ZMD involves expensive computation, while it is simple to extract GMD. The online matching of GD is the most expensive.
- Accuracy. ZMD is more suitable for generic shape description. GD is suitable for situations where exact matching is needed. GMD is suitable for situations where very rough matching is needed, for example, it can serve as an initial matching before a refining matching is taken.
- Hierarchical representation. Both ZMD and GD support hierarchical representation. The number of ZMDs can be adjusted to meet hierarchical requirements. For GD, hierarchical representation can be achieved by adjusting the cell size, or by combining it with eccentricity and circularity. GMD does not support hierarchical representation because higher geometric moment invariants are difficult to obtain.

4 Evaluation of Zernike Moment Descriptor against Fourier Descriptor and Curvature Scale Space Descriptor

In the above two sections, two contour-based shape descriptors and three region-based shape descriptors have been evaluated, respectively. It has been found that for contour-based shape descriptors, FD outperforms CSSD; and for region-based shape descriptors, ZMD outperforms GD and GMD. Since the region-based shape descriptor can be applied to general shapes, in this section, ZMD is compared with the contour-based shape descriptors FD and CSSD. The purpose is to test whether the region-based shape descriptor outperforms the contour-based shape descriptors.

4.1 Comparison of retrieval effectiveness

Since FD and CSSD can only be applied to contour shapes, the comparison is conducted on the MPEG-7 contour shape database CE-1. The test method is the same as that described in Sect. 2.3. The average retrieval precision and recall using the three shape descriptors on different data sets of the MPEG-7 contour shape database CE-1 are shown in Fig. 19a–d.

It can be seen from Fig. 19a that ZMD outperforms FD and CSSD on all the Data sets of the MPEG-7 contour shape database.

4.2 Comparison of computational efficiency

To compare the computational efficiency of the three shape descriptors, the feature extraction and retrieval are tested on Set B of the MPEG-7 contour shape database. The time taken for feature extraction and retrieval is given in Table 3. It can be seen from Table 3 that both contour-based shape descriptors are more efficient than ZMD in the feature extraction. FD is also more efficient than ZMD in online retrieval. However, ZMD is more efficient than CSSD in online retrieval.

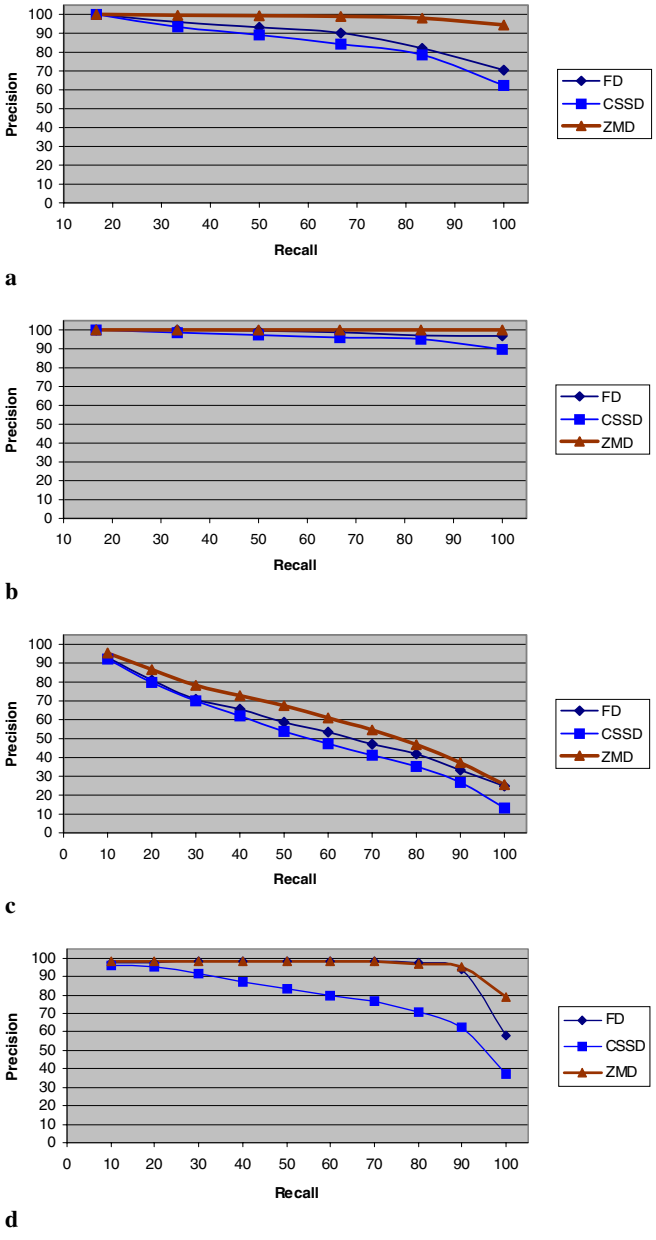


Fig. 19a–d. Average precision recall of FD, CSSD and ZMD on MPEG-7 contour shape database CE-1. **a** Average precision recall of 420 retrievals using FD, CSSD, and ZMD on Set A1. **b** Average precision-recall of 420 retrievals using FD, CSSD, and ZMD on Set A2. **c** Average precision recall of 1400 retrievals using FD, CSSD, and ZMD on Set B. **d** Average precision recall of 200 retrievals using FD, CSSD, and ZMD on Set C

4.3 Discussion

In the above, ZMD has been compared with FD and CSSD. Results show that in terms of retrieval effectiveness, ZMD outperforms both of the contour-based shape descriptors. In terms of computational efficiency, FD shows advantage over ZMD. CSSD is only more efficient than ZMD in feature extraction, and it is less efficient than ZMD in online retrieval. For image retrieval, feature extraction is usually carried out offline; online retrieval efficiency is more essential. Therefore, overall,

Table 3. Time of feature extraction and retrieval using FD, CSSD, and ZMD

Time / Shape descriptors	Total time of feature extraction of 1400 shapes	Average time of feature extraction of each shape	Total time of retrieval of 1400 queries	Average time of retrieval of each query
FD	80960 ms	57.8 ms	49894 ms	35.6 ms
CSSD	120629 ms	86.1 ms	163570 ms	116.8 ms
ZMD	1576681 ms	1126.2 ms	136642 ms	97.6 ms

ZMD is more desirable than CSSD for contour shape retrieval. Since ZMD can be applied to generic shapes, ZMD is better than CSSD in general applications.

5 Conclusions

In this paper, two MPEG-7 shape descriptors have been evaluated against three other shape descriptors according to the six principles set by MPEG-7. The implementation of CSSD has been given in algorithm form. The grid descriptor has been improved to describe region-based shape, and an optimal major axis algorithm suitable for general shape normalization has been proposed. The experimental results in the paper were obtained using standard shape databases and common performance measurements.

In terms of computation complexity, robustness, hierarchical coarse to fine Representation, and retrieval accuracy, the Fourier descriptor (FD) outperforms The curvature scale space descriptor (CSSD). The main problems with CSSD are that it does not capture global features, and the matching is too complex. FD is also more stable than CSSD when applied to different applications, because the computation of FD is simpler and involves fewer parameters than that of CSSD.

Overall, the Zernike moments descriptors (ZMD) is most suitable for region-based shape retrieval among the three region shape descriptors studied. ZMD captures spectral shape features, which are more robust than spatial shape features. However, the use of concentric circular sampling causes a problem in describing perspective transformed or stretched shapes. GMD is a very inaccurate shape descriptor because it is only derived from the three lower order moments. GD is less robust than ZMD due to the use of the major axis as the scaling and rotation normalization. However, GD outperforms GMD significantly in a generic retrieval test, and GD agrees more with human perception than GMD.

Overall, ZMD outperforms CSSD, therefore we conclude that CSSD can be replaced by ZMD. If computational efficiency and storage are essential requirements, FD can be used as a replacement to CSSD for contour shape description.

References

1. Loncaric S (1998) A survey of shape analysis techniques. Patt Recogn 31(8):983-1001
2. Niblack W et al (1993) The QBIC Project: Querying images by content using color, texture and shape. SPIE Conf on storage and retrieval for image and video databases, Vol 1908, San Jose, CA, pp 173–187
3. Davies ER (1997) Machine vision: theory, algorithms, practicalities. Academic Press
4. Kauppinen H, Seppanen T, Pietikainen M (1995) An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Trans PAMI 17(2):201–207
5. Marin FJS (2000) Automatic recognition of biological shapes with and without representations of shape. Artif Intell in Med 18(2):173–186
6. Marin FJS (2001) Automatic recognition of biological shapes using the Hotelling transform. Comput Biol Med 31(2):85–99
7. Persoon E, Fu K (1977) Shape discrimination using Fourier descriptors. IEEE Trans Syst, Man Cybern SMC-7(3):170–179
8. Yang HS, Lee SU, Lee KM (1998) Recognition of 2D object contours using starting-point-independent wavelet coefficient matching. J Visual Comm Image Represent 9(2):171–181
9. Zhang DS, Lu G (2001) A comparison of shape retrieval using Fourier descriptors and short-time Fourier descriptors. Proc Second IEEE Pacific-Rim Conf on Multimedia (PCM01), Beijing, China, pp.855–860
10. Zhang DS, Lu G (2002) A comparative study of Fourier descriptors for shape representation and retrieval. Proc Fifth Asian Conf on Computer Vision (ACCV02), Melbourne, Australia, pp 646–651
11. Zahn CT, Roskies RZ (1972) Fourier descriptors for plane closed curves. IEEE Trans Comput c-21(3):269–281
12. Huang C-L, Huang D-H (1998) A content-based image retrieval system. Image Vision Comput 16:149–163
13. Mokhtarian F, Abbasi S, Kittler J (1996) Efficient and robust retrieval by shape content through curvature scale space. Int Workshop on Image DataBases and Multimedia Search, Amsterdam, The Netherlands, pp 35–42
14. Del Bimbo A, Pala P (1997) Visual image retrieval by elastic matching of user sketches. IEEE Trans PAMI 19(2):121–132
15. Eichmann G et al (1990) Shape representation by Gabor expansion. SPIE Vol 1297, Hybrid Image and Signal Processing II, pp 86–94
16. Tieng QM, Boles WW (1997) Recognition of 2D object contours using the wavelet transform zero-crossing representation. IEEE Trans PAMI 19(8):910–916
17. Asada H, Brandy M (1986) The curvature primal sketch. IEEE Trans PAMI 8(1):2–14
18. Abbasi S, Mokhtarian F, Kittler J (1999) Curvature scale space image in shape similarity retrieval. Multimedia Syst 7(6):467–476
19. Abbasi S, Mokhtarian F, Kittler J (2000) Enhancing CSS-based shape retrieval for objects with shallow concavities. Image Vision Comput 18(3):199–211
20. Groskey WI, Neo P, Mehrotra R (1992) A pictorial index mechanism for model-based matching. Data Knowl Eng 8:309–327
21. Mehrotra R, Gary JE (1995) Similar-shape retrieval in shape data management. IEEE Comput pp 57–62
22. Berretti S, Del Bimbo A, Pala P (2000) Retrieval by shape similarity with perceptual distance and effective indexing. IEEE Trans Multimedia 2(4):225–239

23. Dudek G, Tsotsos JK (1997) Shape representation and recognition from multiscale curvature. Comput Vision Image Understanding 68(2):170–189
24. Li SZ (1999) Shape matching based on invariants. In: Omidvar O (ed) Shape Analysis: Progress in Neural Networks 6. Ablex, NJ, pp 203–228
25. Hu M (1962) Visual pattern recognition by moment invariants. IRE Trans Infor Theory IT-8:179–187
26. Liao SX, Pawlak M (1996) On image analysis by moments. IEEE Trans PAMI 18(3):254–266
27. Teague MR (1980) Image analysis via the general theory of moments. J Opt Soc Am 70(8):920–930
28. Teh C-H, Chin RT (1988) On image analysis by the methods of moments. IEEE Trans PAMI 10(4):496–513
29. Taubin G, Cooper DB (1991) Recognition and positioning of rigid objects using algebraic moment invariants. SPIE Conf on Geometric Methods in Computer Vision, Vol 1570, pp. 175–186
30. Kim W-Y, Kim Y-S (2000) A region-based shape descriptor using Zernike moments. Signal Process: Image Comm 16:95–102
31. Chakrabarti K, Binderberger MO, Porkaew K, Mehrotra S (2000) Similar shape retrieval in MARS. Proc IEEE Int Conf on Multimedia and Expo (CD-ROM), New York, NY
32. Lu G, Sajjanhar A (1999) Region-based shape representation and similarity measure suitable for content-based image retrieval. Multimedia Syst 7(2):165–174
33. Safar M, Shahabi C, Sun X (2000) Image retrieval by shape: a comparative study. Proc IEEE Int Conf on Multimedia and Expo (CD-ROM), New York, NY
34. Kim H, Kim J (2000) Region-based shape descriptor invariant to rotation, scale and translation. Signal Process: Image Comm 16:87–93
35. Jeannin S (ed) (2000) MPEG-7 Visual part of experimentation Model Version 5.0. ISO/IEC JTC1/SC29/ WG11/N3321, Nordwijkerhout
36. Martínez JM (2002) MPEG-7 Overview (version 7). ISO/IEC JTC1/SC29/WG11 N4674, Jeju
37. Martinez JM, Koenen R, Pereira F (2002) MPEG-7: The generic multimedia content description standard, part 1. IEEE Multimedia 9(2):78–87
38. Martinez JM (2002) Standards – MPEG-7: Overview of MPEG-7 description tools, part 2. IEEE Multimedia 9(3):83–93
39. Del Bimbo A (1999) Visual information retrieval. Morgan Kaufmann, pp 56–57
40. Kennedy RL, Lee Y, Roy BV, Reed CD, Lippmann RP (1998) Solving data mining problems through pattern recognition. Prentice Hall
41. Zhang DS, Lu G (2002) Enhanced generic Fourier descriptor for object-based image retrieval. Proc IEEE Int Conf on Acoustics, Speech, and Signal Processing (ICASSP2002), Orlando, FL, Vol 4, pp 3668–3671
42. Mehtre BM, Kankanhalli MS, Lee WF (1997) Shape measures for content based image retrieval: a comparison. Infor Process Manage 33(3):319–337