PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 37 (2004) 1–19

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Review of shape representation and description techniques

Dengsheng Zhang*, Guojun Lu

Gippsland School of Computing and Info. Tech., Monash University, Churchill, Vic 3842, Australia

Received 1 August 2002; accepted 16 July 2003

Abstract

More and more images have been generated in digital form around the world. There is a growing interest in finding images in large collections or from remote databases. In order to find an image, the image has to be described or represented by certain features. Shape is an important visual feature of an image. Searching for images using shape features has attracted much attention. There are many shape representation and description techniques in the literature. In this paper, we classify and review these important techniques. We examine implementation procedures for each technique and discuss its advantages and disadvantages. Some recent research results are also included and discussed in this paper. Finally, we identify some promising techniques for image retrieval according to standard principles.

Keywords: Shape; Image retrieval; CBIR; Review; Shape descriptor

1. Introduction

As information objects are digitized, more and more digital images have been generated. There is an urgent demand for effective tools to facilitate the searching of images. The goal to find a similar image (object) from large collections or from remotely distributed databases is shared not only by researchers, educators and professionals, but also by general users. Shape is an important visual feature and it is one of the basic features used to describe image content. However, shape representation and description is a difficult task. This is because when a 3-D real world object is projected onto a 2-D image plane, one dimension of object information is lost. As a result, the shape extracted from the image only partially represents the projected object. To make the problem even more complex, shape is often corrupted with noise, defects, arbitrary distortion and occlusion.

Shape representation generally looks for effective and perceptually important shape features based on either shape boundary information or boundary plus interior content. Various features have been designed, including *shape signature*, *signature histogram*, *shape invariants*, *moments*, *curvature*, *shape context*, *shape matrix*, *spectral features* etc. These various shape features are often evaluated by how accurately they allow one to retrieve similar shapes from a designated database. However, it is not sufficient to evaluate a representation technique only by the effectiveness of the features employed. This is because the evaluation ignores other important characteristics of a shape representation technique. For example, in the new multimedia application *content-based image retrieval* (CBIR), efficiency is envisaged as equally important as effectiveness due to the online retrieval demand. In fact, MPEG-7 has set several principles to measure a shape descriptor, that is, good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation [1].

Good retrieval accuracy requires a shape descriptor be able to effectively find perceptually similar shapes from a database. Perceptually similar shapes usually means rotated, translated, scaled shapes and affinely transformed shapes. The descriptor should also be able to find noise affected shapes, variously distorted shapes and defective

* Corresponding author. Tel.: +61-3-99026772; fax: +61-3-99026842.
E-mail addresses: dengsheng.zhang@infotech.monash.edu.au (D. Zhang), guojun.lu@infotech.monash.edu.au (G. Lu).

0031-3203/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2003.07.008

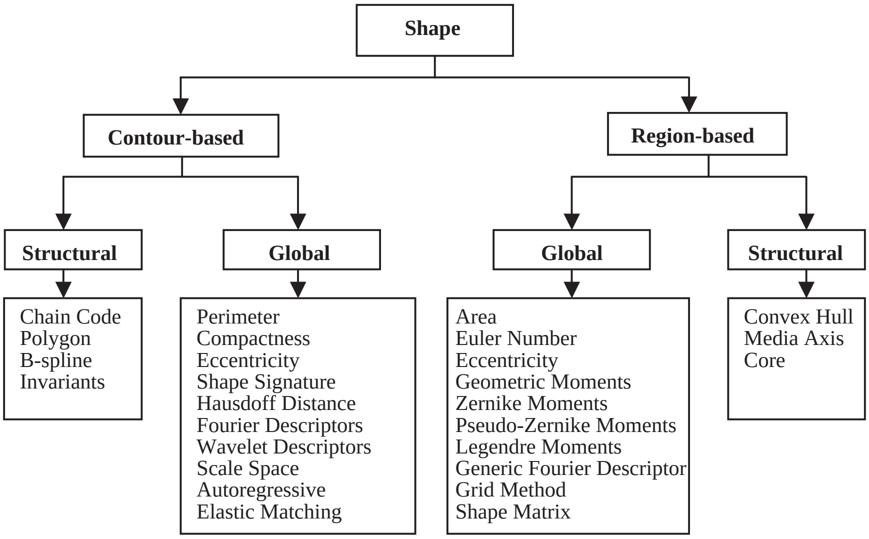

Fig. 1. Classification of shape representation and description techniques.

shapes, which are tolerated by human beings when comparing shapes. This is known as the robustness requirement. Compact features are desirable for indexing and online retrieval. If a shape descriptor has a hierarchical coarse to fine representation characteristic, it can achieve a high level of matching efficiency. This is because shapes can be matched at coarse level to first eliminate large amount dissimilar shapes, and at finer level, shapes can be matched in details. A desirable shape descriptor should be application independent rather than only performing well for certain type of shapes. Low computation complexity is an important characteristic of a desirable shape descriptor. For a shape descriptor, low computation complexity means minimizing any uncertain or ad hoc factors that are involved in the derivation processes. The fewer the uncertain factors involved in the computation processes, the more robust the shape descriptor becomes. In essence, low computation complexity means clarity and stability.

Many shape representation and description techniques have been developed in the past. A number of new techniques have been proposed in recent years. There are also many new shape applications in recent years. In this paper, we review and examine important shape representation and description techniques, and indicate their pros and cons. The retrieval performance and comparison results will be discussed where available. Finally, promising shape descriptors are identified according to the principles mentioned above. The rest of the paper is organized as follows. In Section 2, the classification of shape representation and description techniques are given. Section 3 discusses contour-based shape representation techniques. Region-based shape representation techniques are discussed in Section 4. The paper is concluded in Section 5.

2. Classification of shape representation and description techniques

Shape representation and description techniques can be generally classified into two class of methods: *contour-based* methods and *region-based* methods. The classification is based on whether shape features are extracted from the contour only or are extracted from the whole shape region. Under each class, the different methods are further divided into *structural* approaches and *global* approaches. This sub-class is based on whether the shape is represented as a whole or represented by segments/sections (primitives). These approaches can be further distinguished into *space domain* and *transform domain*, based on whether the shape features are derived from the spatial domain or from the transformed domain. The whole hierarchy of the classification is shown in Fig. 1. In the following sections, these techniques are discussed in details.

3. Contour-based shape representation and description techniques

Contour shape techniques only exploit shape boundary information. There are generally two types of very different approaches for contour shape modeling: *continuous approach* (global) and *discrete approach* (structural). Continuous approaches do not divide shape into sub-parts, usually a feature vector derived from the integral boundary is used to describe the shape. The measure of shape similarity is usually a metric distance between the acquired feature vectors. Discrete approaches break the shape boundary into segments, called *primitives* using a particular criterion. The

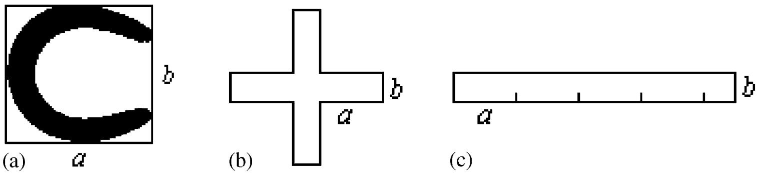

Fig. 2. Shape eccentricity and circularity.

final representation is usually a string or a graph (or tree), the similarity measure is done by string matching or graph matching. In the following we discuss these two types of approaches.

3.1. Global methods

Global contour shape representation techniques usually compute a multi-dimensional numeric feature vector from the shape boundary information. The matching between shapes is a straightforward process, which is usually conducted by using a metric distance, such as Euclidean distance or city block distance. Point (or point feature) based matching is also used in particular applications.

3.1.1. Simple shape descriptors

Common simple global descriptors are *area*, *circularity* (perimeter^2/area), *eccentricity* (length of major axis/length of minor axis), *major axis orientation*, and *bending energy* [2]. These simple global descriptors usually can only discriminate shapes with large differences, therefore, they are usually used as filters to eliminate false hits or combined with other shape descriptors to discriminate shapes. They are not suitable to be standalone shape descriptors. For example, the eccentricity of the shape in Fig. 2(a) is close to 1 ($a = b$), it does not correctly describe the shape, because perceptually it is an elongated shape. In this case, circularity is a better descriptor. The two shapes in Fig. 2(b) and (c) have the same circularity ($a = 2b$), however, they are very different shapes. In this case, eccentricity is a better descriptor.

Other simple global contour shape descriptors have been proposed by Peura and Iivarinen [3]. These descriptors include *convexity*, *ratio of principle axis*, *circular variance* and *elliptic variance*.

3.1.2. Correspondence-based shape matching

Correspondence-based shape matching works in the space domain. In contrast to feature-based shape representation techniques, correspondence-based shape matching measures similarity between shapes using point-to-point matching. In other words, every point on the shape is treated as a feature point. The matching is conducted on 2-D space.

Hausdorff distance is a classical correspondence-based shape matching method, it has often been used to locate objects in an image and measure similarity between shapes [4–9]. Given two shapes represented by two set of points: $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p\}$ and $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_q\}$, the Hausdorff distance between A and B is defined as

$$H(A,B) = \max(h(A,B), h(B,A)\}, \tag{3.1}$$

where

$$h(A,B) = \max_{a\in A}\min_{b\in B}\|\mathbf{a} - \mathbf{b}\| \tag{3.2}$$

and $\|\cdot\|$ is the underlying norm on the points of A and B, usually Euclidean distance. However, this distance measure is too sensitive to noise or outlier. A single point in A that is far from anything in B will cause $h(A,B)$ to be large. Therefore, a modified Hausdorff distance is introduced by Rucklidge [8]:

$$h^f(A,B) = f^{th}_{a\in A}\min_{b\in B}\|a - b\|, \tag{3.3}$$

where $f^{th}_{x\in X}g(x)$ denotes the f^{th} quantile value of $g(x)$ over set X, for some value of f between 0 and 1. For example, the 1th quantile value is the maximum and the 1/2th quantile value is the median. In practice, f is usually set to be 1/2 [4]. The advantage of shape matching using Hausdorff distance is that shape can be matched partially. However, the Hausdorff distance is not translation, scale and rotation invariant. In order to match a model shape with a shape in the image, the model shape has to be overlapped on the image in different positions, different orientations and different scales. As the result, the matching is prohibitively expensive. Chetverikov and Kenokh propose an efficient matching by *chamfer distance transform* [4]. This reduces a translation invariant matching from $O(N^2)$ of computations to $O(N)$, where N is the number of image points. Traditional Hausdorff shape matching only allows similarity matching, that is, shape matching allowing translation, rotation and scaling changes. Rucklidge extends Hausdorff distance matching into affine invariant matching [8]. For this purpose, a set of affine models are generated from the model shape. Since the space of affine transformations from the model shape is large, an efficient matching scheme is introduced by only examining a small part of the space of the affine transformations. Despite the efficiency effort, the matching load is still unacceptably high. The matching of even a small model shape with a normal image can take half an hour on an eight-processor Sun SPARCServer 1000 [8].

Shape matching using Hausdorff distance is sensitive to noise and slight variations. Recently, Belongie et al. propose a correspondence-based shape matching method using

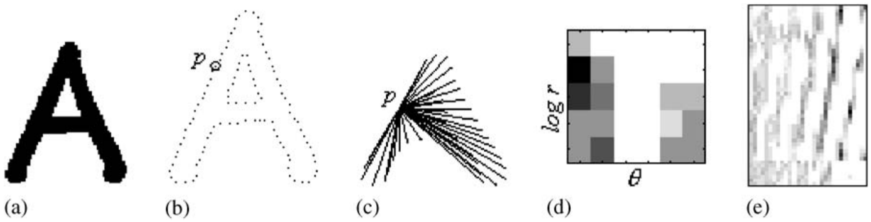

Fig. 3. Shape context. (a) a character shape; (b) edge image of (a); (c) a point p on shape (a) and all the vectors started from p; (d) the log-polar histogram of the vectors in (c), the histogram is the context of point p; (e) the context map of shape (a), each row of the context map is the flattened histogram of each point context, the number of rows is the number of sampled points. (reprinted from [10]).

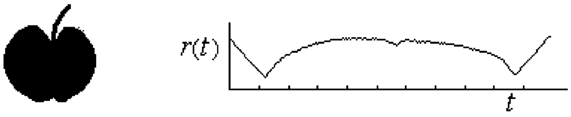

Fig. 4. An apple shape and its centroid distance signature.

shape contexts [10]. Shape matching using shape contexts is an improvement to traditional Hausdorff distance based methods. It extracts a global feature, called shape context, for each corresponding point. The matching between corresponding points is then the matching between the context features. To extract the shape context at a point p, the vectors of p to all the other boundary points are found (Fig. 3(c)). The length r and orientation θ of the vectors are quantized to create a histogram map which is used to represent the point p (Fig. 3(d)). The histogram of each point is flattened and concatenated to form the context of the shape (Fig. 3(e)). To make the histogram more sensitive to positions of nearby points than to those of points farther away, these vectors are put into log-polar space. For example, in Fig. 3 [10], (a) is a character shape, (b) is the edge image of (a), (c) is the set of vectors at point p, (d) is the histogram or context extracted from the vectors, (e) is the context of the shape.

The matching of two shapes is done by matching two context maps of the shapes, which is a matrix-based matching. It minimizes the total cost of matching between one context matrix and all the permutations of another context matrix. In order to reduce the matching overhead, it samples the boundary at a number of points and uses the shortest augmenting path algorithm for the matrix matching.

3.1.3. Shape signature

A *Shape signature* represents a shape by a one dimensional function derived from shape boundary points. Many shape signatures exist, they include *centroidal profile*, *complex coordinates*, *centroid distance* (Fig. 4), *tangent angle*, *cumulative angle*, *curvature*, *area* and *chord-length* [11–13]. Shape signatures are usually normalized into being translation and scale invariant. In order to compensate for orientation changes, shift matching is needed to find the best matching between two shapes. Most of the signature matching is normalized to shift matching in 1-D space, however, some signature matching requires shift matching in 2-D space, such as the matching of *centroidal profiles* [11]. In either case, the matching cost is too high for online retrieval.

In addition to the high matching cost, shape signatures are sensitive to noise, and slight changes in the boundary can cause large errors in matching. Therefore, it is undesirable to directly describe shape using a shape signature. Further processing is necessary to increase its robustness and reduce the matching load. For example, a shape signature can be simplified by quantizing the signature into a signature histogram, which is rotationally invariant.

3.1.4. Boundary moments

Boundary moments can be used to reduce the dimensions of the boundary representation. Assuming the shape boundary has been represented as a shape signature $z(i)$, the rth moment m_r and central moment μ_r can be estimated as [14]

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r \quad \text{and} \quad \mu_r = \frac{1}{N}\sum_{i=1}^{N}[z(i) - m_1]^r, \tag{3.4}$$

where N is the number of boundary points. The normalized moments $\bar{m}_r = m_r/(\mu_2)^{r/2}$ and $\bar{\mu}_r = \mu_r/(\mu_2)^{r/2}$ are invariant to shape translation, rotation and scaling. Less noise-sensitive shape descriptors can be obtained from $F_1 = (\mu_2)^{1/2}/m_1$, $F_2 = \mu_3/(\mu_2)^{3/2}$, and $F_3 = \mu_4/(\mu_2)^2$.

The method in [15] treats the amplitude of the shape signature function $z(i)$ as a random variable v and creates a histogram $p(v_i)$ from $z(i)$. Then, the rth moment is obtained by

$$\mu_r = \sum_{i=1}^{K}(v_i - m)^r p(v_i) \quad \text{and} \quad m = \sum_{i=1}^{K} v_1 p(v_i). \tag{3.5}$$

The advantage of boundary moment descriptors is that it is easy to implement. However, it is difficult to associate higher order moments with physical interpretation.

3.1.5. Elastic matching

Bimbo and Pala have proposed the use of *elastic matching* for shape based image retrieval [16]. According to this approach, a deformed template is generated as the sum of the original template $\boldsymbol{\tau}(s)$ and a warping deformation $\boldsymbol{\theta}(s)$

$$\boldsymbol{\varphi}(s) = \boldsymbol{\tau}(s) + \boldsymbol{\theta}(s), \tag{3.6}$$

where $\boldsymbol{\tau} = (\tau_x, \tau_y)$ is a second order spline and $\boldsymbol{\theta} = (\theta_x, \theta_y)$ is the deformation. The similarity between the original shape of the template and the shape of the object in the image is measured by minimizing a compound function:

$$\begin{aligned} F &= S + B + M \\ &= \alpha \int_0^1 \left[\left(\frac{\mathrm{d}\theta_x}{\mathrm{d}s} \right)^2 + \left(\frac{\mathrm{d}\theta_y}{\mathrm{d}s} \right)^2 \right] \mathrm{d}s \\ &\quad + \beta \int_0^1 \left[\left(\frac{\mathrm{d}^2\theta_x}{\mathrm{d}s} \right)^2 + \left(\frac{\mathrm{d}^2\theta_y}{\mathrm{d}s} \right)^2 \right] \mathrm{d}s \\ &\quad + \int_0^1 I_E(\boldsymbol{\varphi}(s))\,\mathrm{d}s \end{aligned} \tag{3.7}$$

where I_E is the object image, S and B are called strain energy and bend energy respectively, while M measures the degree of overlapping between the deformed template and the object in the image. The three quantitative measures are not sufficient to measure the similarity between shapes, therefore *shape complexity* N (measured as the number of 0's of the curvature function associated with the templates contour) and *correlation* C (between the curvature function associated with the template and that associated with the deformed one) are also taken into account in the similarity measure. Finally, the five parameters (S, B, M, N, C) are classified by a back-propagation neural network.

This approach is not practical for online image retrieval, mainly because of the computation and matching complexity. The authors compared the computation complexity of this feature extraction with QBIC [17] and QVE [18], and demonstrated that the number of CPU operations is less than that of QBIC and QVE. However, a number of steps of the deformation process is needed to complete a matching. This makes the matching extremely expensive, although the aspect ratio checking and composite filtering (based on relationships matching for multiple templates) are used in the initial stage and M are used for filtering in the deformation process. The aspect ratio used for the filtering can cause false rejection as will be indicated in Section 3.1.1. The shape description is not rotationally invariant. Also, the warping criteria used for the template deformation are not given. The examples of warping shown in that paper indicate the warping is arbitrary or application dependent.

3.1.6. Stochastic method

Time-series models and especially *autoregressive* (AR) modeling has been used for calculating shape descriptors [19–25]. Methods in this class are based on the stochastic modeling of a 1-D function f obtained from the shape as described in Section 3.1.3. A linear autoregressive model expresses a value of a function as the 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 autoregressive model is a simple predictor of the current radius by a linear combination of M previous radii plus a constant term and an error term:

$$f_t = \alpha + \sum_{j=1}^{m} \theta_j f_{t-j} + \sqrt{\beta}\omega_t, \tag{3.8}$$

where θ_j are the AR-model coefficients, m is the model order, i.e., tells how many preceding function values the model uses. $\sqrt{\beta}\omega_t$ is the current error term or residual, reflecting the accuracy of the prediction. α is proportional to the mean of function values. The parameters $\{\alpha, \theta_1, \ldots, \theta_m, \beta\}$ are estimated by using the *least square* (LS) criterion [19,21,23]. The estimated θ_j are translation, rotation and scale invariant. Parameters α and β are not scale invariant, but the quotient $\alpha/\sqrt{\beta}$, which reflects signal-to-noise ratio of the boundary, is regarded as a invariant. Therefore, the feature vector $[\theta_1, \ldots, \theta_m, \alpha/\sqrt{B}]^{\mathrm{T}}$ is used as the shape descriptor.

The disadvantage of the AR method is that in the case of complex boundaries, a small number of AR parameters is not sufficient for an adequate description. The choice of m is a complicated problem and is usually decided empirically. Besides, the physical meaning associated with each θ_j is not clear.

3.1.7. Scale space method

The problem of noise sensitivity and boundary variations in most spatial domain shape methods inspires the use of scale space analysis. The *scale space* representation of a shape is created by tracking the position of inflection points in a shape boundary filtered by low-pass Gaussian filters of variable widths. As the width (σ) of Gaussian filter increases, insignificant inflections are eliminated from the boundary and the shape becomes smoother (Fig. 5(a)). The inflection points that remain present in the representation are expected to be 'significant' object characteristics. The result of this smoothing process is an *interval tree*, called 'fingerprint', consisting of inflection points (Fig. 5(b)). The difficulty with this type of approach is the interpretation of the final result.

Asada and Brady first attempted to interpret the interval tree acquired from scale space [26,27]. Their interval trees are acquired from both Gaussian filter and second derivatives of Gaussian filter. The interpretation of the interval tree is based on detecting the peaks of the tree branches from higher scales to lower scales (Fig. 5(c)). Since the shapes under analysis are from hardware application, it is possible to interpret higher level primitive events from the detected peaks. In this application, the primitive events are defined as corner, smooth joint, end, crank, bump/dent.

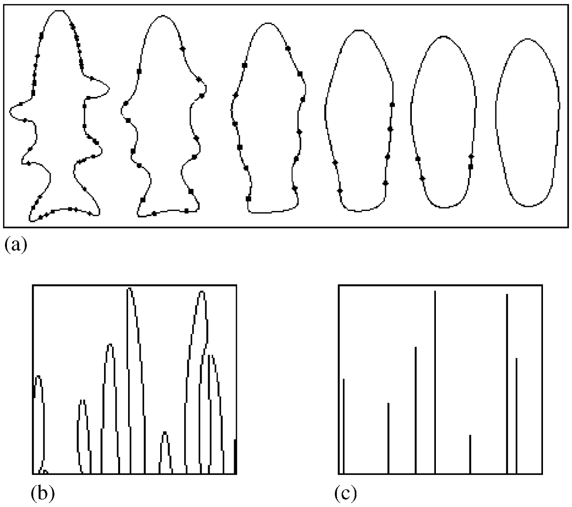

Fig. 5. (a) The evolution of shape boundary as scale σ increases (reprinted from [30]). From left to right: $\sigma = 1, 4, 7, 10, 12, 14$. The points marked on the boundary are the inflection points; (b) the interval tree (or curvature scale space contour map) resulted from the smoothing process; (c) the peaks of the interval tree.

Mokhtarian and Mackworth [28] adopted Asada and Brady's interpretation method and extended it to shoreline registration. They call the acquired scale space signature as a *curvature scale space* (CSS) contour image. The peaks of individual branches in the CSS are detected. However, instead of interpreting these peaks as higher level primitive events, they are used for matching two curves under analysis. The matching proves to be very complex and expensive. It tries to find the best match between the contour branches in the two interval trees. For each individual matching, the template contour branch has to be scaled and shifted to accommodate scale and translation invariance. During each individual matching, the template curve is also affinely transformed to match with the model curve. The method is later extended for shape retrieval [29–32]. Since the old matching algorithm is too complex for retrieval, the new method developed a matching algorithm based on the two highest peaks in each of the two contour images. All the images under analysis are also scaled into the same size (same number of boundary points) prior to applying scale space. However, the four essential empirical parameters involved in the feature extraction and matching processes make this algorithm unstable [33].

Daoudi and Matusiak interpret the interval tree obtained from scale space as a geodesic topology [34]. The matching between two shapes now turns into matching between two scale space images using the geodesic distance measure suggested by Eberly [35]. The matching is actually a point to point 2-D matching between the two scale space images, this can be impractical if shapes in the database are complex, resulting in very high interval tree.

3.1.8. *Spectral transform*

Spectral descriptors overcome the problem of noise sensitivity and boundary variations by analyzing shape in spectral domain. Spectral descriptors include *Fourier descriptor* (FD) and *wavelet descriptor* (WD), they are derived from spectral transforms on 1-D shape signatures described in Section 3.1.3.

One of the most widely used shape description methods is FD [12,19,36–52]. Conventional FD methods only deal with closed curve, however, Lin et al. and Mitchell et al. used FD to describe partial shapes [43,47]. Arbter et al. introduced the affine-invariant FD to take into consideration of affine shape description [36,37]. Granlund introduced the Fourier invariants which describe the rotational symmetry of shapes [40]. Rauber proposed a UNL FD (named after Universidade Nova de Lisboa, Portugal) which is able to describe disjointed or articulated contour shape [49]. The UNL FD is acquired by applying 2-D Fourier transform on the UNL transformed shape image. Even though a feature selection process is followed, the dimension of the feature vector acquired this way is very high. Richard and Hemami introduced a complex distance measurement, called the true distance measurement, for measuring the similarity between two set of FDs [50]. Since the true distance measurement requires two Fourier transforms for each matching, it involves 15 times more computation than a normal distance measurement. Rui et al. [51] proposed a distance measurement to classify similarity transformed characters using Fourier transformed coefficients. This distance measurement is the weighting sum of the variance of magnitude ratios and the variance of phase difference between two sets of Fourier coefficients. The Fourier coefficients are derived from Fourier reconstructed shape boundary rather than from original boundary. This is not different from FD derived from a smoothed boundary. Eichmann et al. proposed the use of a short-time Fourier descriptor (SFD) for shape description [39], however, Zhang and Lu have found that SFD is outperformed by conventional FD methods in shape retrieval [53]. This is because SFD cannot capture global shape features although it can capture local shape features more accurately.

Recently, several researchers have proposed the use of WD for shape description [54–56]. Although WD has the advantage over FD in that it is of multi-resolution in both spatial space and spectral space, the increase of spatial resolution will certainly sacrifice frequency resolution. For example, in [55], only wavelet coefficients of the few low frequencies are used to represent shape. Most importantly, the complicated matching scheme of wavelet representation makes it impractical for online shape retrieval. In [56], 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 [55], the number of matchings for similarity measurement is not only large but also dependent on the complexity of the shape, since the similarity measurement is the all level shift matching of all the zero-crossing points of the wavelet approximation of the shape. Apart from the matching complexity, the dyadic wavelets used can rarely associate the feature segments on the shape boundary. Therefore, WD suffers the same drawback in primitive determination as that in the structural approach, which will be discussed in Section 3.2.

FD is backed by the well-developed and well-understood Fourier theory. The advantages of FD over many other shape descriptors are (i) simple to compute; (ii) each descriptor has specific physical meaning; (iii) simple to do normalization, making shape matching a simple task; (iv) captures both global and local features. With sufficient features for selection, FD overcomes the weak discrimination ability of those simple global descriptors. FD also overcomes the noise sensitivity and difficult normalization in the shape signature representations.

Most FD based works are dedicated to character recognition and object classification. The complex coordinates and the cumulative angle function are dominantly used in these works to derive FD. However, Zhang and Lu [13,57] have found that for general shapes, the centroid distance function is the most desirable shape signature to derive FD. They have also found that 10 FD features are sufficient to represent shape, this is a significant reduction in dimensions of FD compared with 60 FD features usually used in shape representation. Their results show that FD outperforms CSS method in terms of retrieval performance and robustness.

3.1.9. Discussions

Global contour shape techniques take the whole shape contour as the shape representation. The matching between shapes can either be in space domain or in feature domain. For shape description, there is always a trade-off between accuracy and efficiency. On the one hand, shape should be described as accurately as possible; on the other hand, a shape description should be as compact as possible to simplify indexing and retrieval. Efficient offline feature extraction is also desirable. Simple global shape descriptors are compact, however, they are very inaccurate shape descriptors. They need to be combined with other shape descriptors to create practical shape descriptors. Correspondence based shape matching and signature based matching are not suitable for online shape matching, because they all involve the 2-D matching of two shapes. However, if partial matching is a requirement, methods based on Hausdorff distance can be a choice. Elastic matching and wavelet methods are complex to implement and match. Autoregressive (AR) methods involve matrix operations which are expensive and it is difficult to associate AR descriptors with any physical meaning. The implementation and matching of CSS are complex. However, the perceptually meaningful and compact features are appealing for shape description and online retrieval. Fourier descriptor is simple to implement, and involves less computation by either using fast Fourier transform (FFT) or using truncated Fourier transform computation. The resulting descriptor is also compact and the matching is very simple. Compared with CSS, FD is simpler to compute and more robust. Boundary moment descriptor is similar to Fourier descriptor, and is easy to acquire. However, unlike Fourier descriptor, only the few lower order moment descriptors have physical interpretation.

3.2. Structural methods

Another member in the shape analysis family is the *structural shape representation*. With the structural approach, shapes are broken down into boundary segments called *primitives*. Structural methods differ in the selection of primitives and the organization of the primitives for shape representation. Common methods of boundary decomposition are based on polygonal approximation, curvature decomposition and curve fitting [58]. The result is encoded into a string of the general form:

$$S = s_1, s_2, \ldots, s_n, \tag{3.9}$$

where s_i may be an element of a chain code, a side of a polygon, a quadratic arc, a spline, etc. s_i may contain a number of attributes like length, average curvature, maximal curvature, bending energy, orientation etc. The string can be directly used for description or can be used as input to a higher level syntactic analyzer. In the following we describe methods of shape representation and description using S.

3.2.1. Chain code representation

Chain code describes an object by a sequence of unit-size line segments with a given orientation. The method was introduced in 1961 by Freeman [59] who described a method permitting the encoding of arbitrary geometric configurations. In this approach, an arbitrary curve is represented by a sequence of small vectors of unit length and a limited set of possible directions, thus termed the *unit-vector* method. In the implementation, a digital boundary of an image is superimposed with a grid, the boundary points are approximated to the nearest grid point, then a sampled image is obtained. From a selected starting point, a chain code can be generated by using 4-directional or 8-directional chain code. N-directional ($N > 8$ and $N = 2^k$) chain code is also possible, it is called general chain code [60].

If the chain code is used for matching it must be independent of the choice of the first boundary pixel in the sequence. One possibility for normalizing the chain code is to find the pixel in the border sequence which results in the minimum integer number if the description chain is interpreted as a base four number—that pixel is then used as the starting pixel. Alternatively, the boundary can be represented by the *differences* in the successive directions in the chain code instead of representing the boundary by relative directions. This can be computed by subtracting each element of the chain code from the previous one and taking the result modulo *n*, where *n* is the connectivity. After these operations, a rotationally invariant chain code is obtained by a cyclic permutation which produces the smallest number. Such a normalized differential chain code is called the *shape number*. Chain code derived in this way is not scale invariant. Although it is possible to scale two similar shapes into the same size, the resulted shape numbers can have a different number of digits, making it impractical to do matching between two shapes.

The chain code usually has high dimensions and is sensitive to noise. It is often used as an input to a higher level analysis. For example, it can be used for polygon approximation and for finding boundary curvature which is a important perceptual feature.

Iivarinen and Visa derive a *chain code histogram* (CCH) for object recognition [61]. The CCH is computed as $p(k) = n_k/n$, where n_k is the number of chain code values k in a chain code and n is the number of links in a chain code. The CCH reflects the probabilities of different directions present in a contour. The CHH is translation and scale invariant, however, it is only invariant to a rotation of 90°. Therefore, the normalized CHH (NCHH) is proposed. It is defined as $p(k) = l_k n_k / l$, where n_k is the same as in CHH, l_k is the length of the direction k and l is the length of the contour. Although CHH reduces the dimensions of chain code representation, it does not solve the noise sensitivity problem.

3.2.2. Polygon decomposition

In [62,63], shape boundary is broken down into line segments by polygon approximation. The polygon vertices are used as primitives. The feature for each primitive is expressed as a four element string which consists of internal angle, distance from the next vertex, and its x and y coordinates. Obviously the feature is not translation, scale and rotation invariant. The similarity between any two shapes is the *editing distance* of the two feature strings. For efficiency and robustness reason, only a fixed number (5) of sharpest vertices are selected from each shape. Therefore, a collection of features belonging to all models in the database is generated for the feature index. The features are then organized into a binary tree or m-nary tree. The matching between shapes involves two steps, that is, feature-by-feature matching in the first step and model-by-model matching in the second step. In the first step, given a data feature of a query shape, the feature is searched through the index tree, if a particular model feature in the database is found to be similar to the data feature, the list of shapes associated with the model feature are retrieved. In the second step, the matching between the query shape and a retrieved model is matched based on the editing distance between the two string of primitives.

In [64], Mehrotra and Gary represented a shape as a chain of vectors. For a shape, a series of interest points are detected from the polygonal approximation of the shape boundary. Given a shape with n interest points, a pair is chosen to form a basis vector. The basis vector is normalized as a unit vector along the x-axis. All other interest points of the shape are transformed to this coordinate system. The shape is then represented by the coordinate set $(x_1, y_1), \ldots, (x_n, y_n)$, where (x_i, y_i) are the normalized coordinates of the ith interest point (Fig. 6). Four transform parameters are also added to the coordinate set to create the final feature vector, the four parameters are S, T_x, T_y, θ, which represent the scale, translation and angle of the basis vector. To achieve start point independence, all vectors formed by connecting two adjacent points are used as basis vectors in turn. The similarity between two features is measured by the Euclidean distance. Boundary feature vectors are organized into a k dimensional B-tree or k dB-tree. During the query, for a given query feature of the query shape, the index is searched, and a list of shapes with similar feature to the query feature is produced. The matching of one or more features does not guarantee a full shape match. Consequently, once shapes with similar features are retrieved, shape similarity is checked by overlaying each retrieved shape on the query shape and evaluating the amount of overlap between them.

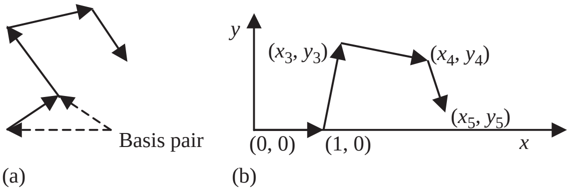

Fig. 6. Chain vectors: (a) original boundary description; (b) normalized boundary description (reproduced from [64]).

All the above three methods approximated a shape as a polygon, and the shape is represented as string of line segments which are then organized into a tree data structure. While it is expected to work well for man made objects, its application for natural objects is impractical.

3.2.3. Smooth curve decomposition

Berretti et al. [65] extended the model used in [62] for general shape retrieval. In [65], the curvature zero-crossing points from a Gaussian smoothed boundary are used to obtain primitives, called tokens (Fig. 7). The feature for each token is its maximum curvature and its orientation, and the similarity between two tokens is measured by the weighted Euclidean distance. Since the feature includes curve orientation, it is not rotation invariant. The authors addressed the problem, but did not solve it. An M-tree is exploited to index the tokens into the feature database. Given a query shape, the retrieval of similar shapes from the database takes two steps. The first step is token retrieval. For all the N tokens on the query shape, the similar tokens are found by traversing the index tree N times. The set of retrieved tokens having the same shape identifier form a potential similar shape. The second step is to match the query shape and the potential similar shape using a model-by-model matching algorithm which is the best match between tokens of the two shapes

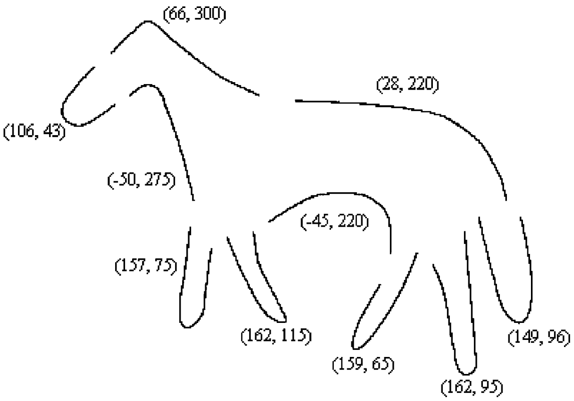

Fig. 7. A horse shape has been divided into different 'tokens'. The numbers corresponding to each token are the curvature and the orientation of the token (reprinted from [65]).

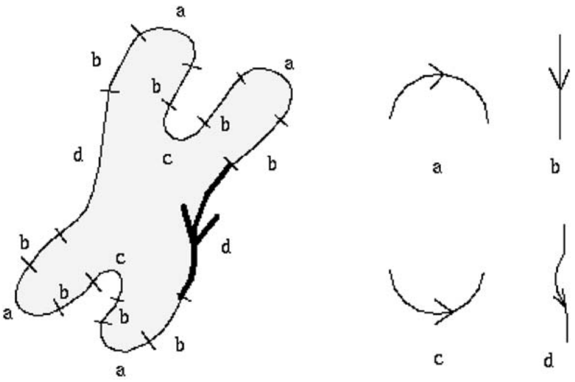

Fig. 8. Structural description of chromosome shape (reprinted from [14]).

and involves $O(MN)$ operations (M and N are the numbers of tokens of two matching shapes, respectively). Matching of tokens in both steps involves thresholding which is ad hoc or empirical. Quantitative retrieval performance (precision and recall) and retrieval efficiency are reported based on a shape database extracted from classical painted images. Since the tree is traversed a number of times in the shape matching, it is not clear whether the indexing is better than model-by-model indexing. Only matching performance using different trees is reported. The matching efficiency also depends on the number of tokens for each shape, and on the scale used in the smoothing stage.

3.2.4. Scale space method

In [62–65], the matching is a *feature-by-feature* matching followed by *model-by-model* matching. Dudek and Tsotsos [66] analyzed shape in scale space and employ a model-by-model matching scheme. In this approach, shape primitives are first obtained from a *curvature-tuned smoothing* technique. A segment descriptor consists of the segment's length, ordinal position, and curvature tuning value extracted from each primitive. A string of segment descriptors is then created to describe the shape. For two shapes A and B represented with their string descriptors $A=(s_1^A, s_2^A \ldots, s_N^A)$ and $B=(s_1^B, s_2^B, \ldots, s_M^B)$, a model-by-model matching using dynamic programming is exploited to obtain the similarity score of the two shapes. To increase robustness and to save matching computation, the shape features are put into a curvature scale space so that shapes can be matched in different scales. However, due to the inclusion of length in the segment descriptors, the descriptors are not scale invariant. Only a small number (50) of shapes from plant leaves were tested for the algorithm, and no recognition or retrieval rate has been reported. The three ad hoc or empirical parameters which are essential in the algorithm make the algorithm application limited.

3.2.5. Syntactic analysis

Syntactic analysis is inspired by the phenomenon that composition of a natural scene is an analog to the composition of a language, that is, sentences are built up from phrases, phrases are built up from words and words are built up from alphabets, etc. [67]. In syntactic methods, shape is represented by a set of *predefined primitives*. The set of predefined primitives is called the *codebook* and the primitives are called *codewords*. For example, given the codewords in the right of Fig. 8, the chromosome shape in the left of Fig. 8 can be represented as a grammatical string of S:

$$S = dbabcbabdbabcbab. \tag{3.10}$$

The matching between shapes can use string matching by finding the minimal number of edit operations to convert one string into another.

A more general method is to formulate the representation as a string grammar. Each primitive is interpreted as a alphabet of some grammar, where a grammar is a set of rules of syntax that govern the generation of sentences formed from symbols of the alphabet. The set of sentences generated by a grammar G is called its language and is denoted as $L(G)$. Here, sentences are strings of symbols (which in turn represent patterns), and languages correspond to pattern class. After grammars have been established, the matching is straightforward. For a sentence representing an unknown shape, the task is to decide in which language the shape represents a valid sentence.

Syntactic shape analysis is based on the theory of formal language [68]. It attempts to simulate the structural and hierarchical nature of the human vision system. However, it is not practical in general applications due to the fact that it is not possible to infer a pattern of grammar which can generate only the valid patterns. In addition, this method needs a priori knowledge for the database in order to define codewords or alphabets. The knowledge is usually unavailable for general applications.

The difference between the syntactic method and the methods discussed in Sections 3.2.1–3.2.4 is that the primitives in the former are predefined and the primitives in the latter are adaptive to the shape database. The predefinition approach is convenient if, in particular application, shape boundary profiles are known in general. However, the alphabet set will be application dependent. The adaptive approach is applicable to general applications, however, it is difficult to determine the types of primitives without the knowledge of the database.

3.2.6. Shape invariants

Shape invariants can also be viewed as a structural approach, because they also represent shapes based on boundary primitives. Designers of shape invariants argue that although most of other shape representation techniques are invariant under similarity transformations (rotation, translation and scaling), they depend on viewpoint [14]. Therefore, techniques using invariants attempt to represent properties of the boundary configurations which remain unchanged under an appropriate class of transformations.

Generally, invariant theory is based on a collection of transformations that can be composed and inverted. In vision, the *projective group* of transformations is considered which contains all the *perspectives* as a subset. The *group* approach provides a mathematical tool for generating invariants. The change of coordinates due to the projective transformation is generalized as a group action. *Lie group* theory is especially useful in designing new invariants.

Invariant is usually named according to the number of features used to define it. An order one invariant is defined on a single feature, and is called an *unary* invariant; an order two invariant is defined between two features, and is called a *binary* invariant; similarly, *ternary* invariant, *quaternary* invariant and so on. The total number of higher order invariants that can exist under a certain group of transformations is larger than that of lower order ones [69].

Common invariants include (i) geometric invariants such as *cross-ratio*, *length ratio*, *distance ratio*, *angle*, *area* [69], *triangle* [70], *invariants from coplanar points* [14]; (ii) algebraic invariants such as *determinant*, *eigenvalues* [71], *trace* [14]; (iii) differential invariants such as *curvature*, *torsion* and *Gaussian curvature*. Geometric invariants and algebraic invariants are suitable in situations where boundaries can be represented by straight lines or algebraic curves. Typical applications of them are for man-made object recognition. If object boundaries cannot be represented by lines or algebraic curves, differential invariants can be formed. Differential invariants are local in nature and are very large in number.

Shape representation using invariants has several problems. First, invariants are usually derived from pure geometric transformation of shape. In reality, shape rarely changes according strict geometric transformations, especially shapes from non-rigid objects. Second, the invariants are very sensitive to boundary noise and errors. Third, design of new invariants is difficult. Fourth, perhaps the most challenging to most of invariant methods is the matching. The matching generally adopts the ‘parts and relationship’ techniques. These techniques require some form of subgraph matching [69,72] which is known to be an NP-complete problem [71]. The challenge is thus to find an algorithm that can find an acceptable solution in reasonable time.

Recently, Kliot and Rivlin proposed the use of *invariant signature* for shape description [73]. Geometric invariants, such as length, angle, areas ratio, cross-ratio and length ratio, are used to derive several invariant signatures (called *multi-valued signature*) for each of the boundary curves. The multi-valued signature is put into a matrix which is used for the matching between two curves on two corresponding shapes. To increase matching efficiency, the signature histogram is constructed and used for the initial matching. Once the curves pass the initial matching, they are subject to matrix matching. The retrieval is tested on a dataset composed of the SQUID [74] fish database and three small data sets. Efficiency performance is given, however, no average effectiveness performance is shown.

Squire and Caelli also use invariance signature for shape description [71]. The invariance signature is a probability density function derived from the boundary represented by piecewise algebraic curves. The signature is invariant under similarity transformations, they are, rotation, translation and scaling transformations. A histogram is then created from the quantized invariance signature and is used for shape description and matching. The histogram is then put into a neural network for character classification. Results show that the invariance signature descriptor is less effective than Fourier-Mellin descriptor, indicating that the signature needs further processing for robust shape description.

Although the invariance signature methods attenuate the load of shape matching using ‘parts and relationship’ method, the matching is still very costly. In addition, shape matching using signature has been well studied by Davies [11], it is not clear whether shape matching using invariance signatures is better than shape matching using common shape signatures.

3.2.7. Discussions

The merits of the structural approach are its capability of handling occlusion problem in the scene and allowing partial matching. However, the merits are at the cost of several drawbacks.

- 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. Therefore, the success of applying this method depends on the a priori knowledge of the shape boundary features held in the database. While structural methods are expected to work under particular assumptions, it is impractical to apply

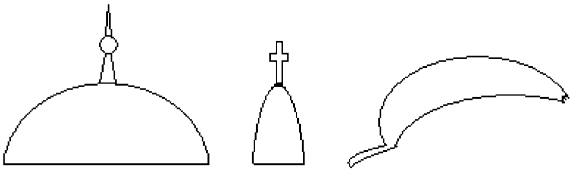

Fig. 9. Ambiguity of structural shape representation.

them for general applications, because it is impossible to know in advance the types of primitives in a generic shape database. For example, if the feature primitives from a query shape are not presented in the database, then the retrieval fails.

- The second drawback of the structural approach is its computation complexity, especially its complex matching. Different from those global methods, the matching schemes used in structural methods adopt *non-metric* similarity measurements. Since these methods allow partial matching between shapes, sub-graph matching using an optimal solution is inevitable. Most of these methods avoid the optimal solution issue, they do not guarantee best match.
- Failure to capture global shape features which are equally important for shape representation. The equal importance of local and global shape features can be shown by the example of observing a city through zooming views. A few zoom-in views of the city does not necessarily mean you have learned much about the city, zoom-out views are also essential to learn more about the city. Unless the city has a unique landmark which distinguishes it from other city, we cannot identify a city from other cities from the map by just a few zoom in views. As an example, Fig. 9 shows two gothic building shapes and a banana shape. Using structural method, the first shape is likely to be more similar to the banana shape although it is more similar to the second shape perceptually. This will not be a problem with global method.
- Sensitivity to noise. Because the structural representation does not preserve the topological structure of the object. Variations of object boundaries can cause significant changes to local structures, therefore, in these cases, global features are more reliable.

4. Region-based shape representation and description techniques

In region based techniques, all the pixels within a shape region are taken into account to obtain the shape representation, rather than only use boundary information as in contour base methods. Common region based methods use moment descriptors to describe shapes. Other region based methods include grid method, shape matrix, convex hull and media axis. Similar to contour based methods, region based shape methods can also be divided into global and structural methods, depending on whether they separate shapes into sub parts or not.

4.1. Global methods

Global methods treat shape as a whole, the result representation is a numeric feature vector which can be used for shape description. Similarity between shapes is simply measured by the metric distance between their feature vectors.

4.1.1. Geometric moment invariants

Historically, Hu published the first significant paper on the use of image moment invariants for two-dimensional pattern recognition applications [75]. His approach is based on the work of the 19th century mathematicians Boole, Cayley and Sylvester, and on the theory of algebraic forms:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y), \qquad p,q = 0,1,2,\ldots \,. \tag{4.1}$$

Using nonlinear combinations of the lower order moments, a set of moment invariants (usually called *geometric moment*), which has the desirable properties of being invariant under translation, scaling and rotation, are derived. The use of higher order moments for pattern analysis has not been addressed. Since the values of the acquired moment invariants are usually very small, a normalization process, such as *zscore normalization* [76], is needed in the implementation.

Geometric moment invariants have attracted wide attention [14,15,77–80] and have been used in many applications [46,81–84]. The main problem with geometric moments is that only a few invariants derived from lower order moments is not sufficient to accurately describe shape. Higher order invariants are difficult to derive.

Zhang and Lu have tested geometric moment invariants on a standard shape database used by MPEG-7 [57]. They have found that geometric moment invariants perform very well on similarity transformed and affinely transformed contour-based shapes. They even outperform grid descriptor for these simple shapes. However, they perform poorly for arbitrarily distorted contour-based shapes. For region-based shapes which have interior content, they only perform satisfactorily on rotated shapes; while for scaled shapes, perspectively transformed shapes and subjective test shapes, they perform poorly. The finding indicates that geometric moment invariants are suitable for describing simple shapes.

4.1.2. Algebraic moment invariants

Algebraic moment invariants have been introduced by Taubin and Cooper [85,86], and has been used in QBIC [17,9]. The algebraic moment invariants are computed from the first m central moments and are given as the eigenvalues of predefined matrices, $M_{[j,k]}$, whose elements are scaled factors of the central moments. Different from Hu's geometric moment invariants, the algebraic moment invariants can be constructed up to arbitrary order and are invariant to affine transformations. However, results from [9] show that algebraic moment invariants performed either very well or very poorly on each of the query objects. They tend to work well on objects where the distribution of the pixels, and not the outline of the shape, is important. On objects where the configuration of the outline is important, as in the difference between an *S* shape and a snake shape, algebraic moment invariants perform poorly.

4.1.3. Orthogonal moments

The algebraic moment transform of (4.1) can be extended to generalized form by replacing the conventional transform kernel $x^p y^q$ with a more general kernel of $P_p(x)P_q(y)$. Teague [79] uses this idea to introduce *orthogonal moments*—Legendre moments and Zernike moments—by replacing $x^p \ y^q$ in (4.1) with Legendre polynomial and Zernike polynomial, respectively.

Legendre moments are given by

$$\lambda_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_x \sum_y P_m(x)P_n(y)f(x,y), \tag{4.2}$$

where

$$P_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2-1)^n$$

Zernike moments are given by

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y V^*_{nm}(x,y)f(x,y), \qquad x^2+y^2 \leqslant 1, \tag{4.3}$$

where

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

and

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

ρ and θ are, respectively, the radius and the angle of pixel (x,y) with respect to the center of gravity of the shape.

Since Legendre and Zernike polynomials are both complete sets of an orthogonal basis, Legendre moments and Zernike moments are called orthogonal moments. Other orthogonal moments are pseudo-Zernike moments which are obtained by using real-value radial polynomials in Zernike polynomials as the moment transform kernel. Orthogonal moments allow for accurate reconstruction of the described shape, and makes optimal utilization of shape information.

Teh and Chin [80] have made a detailed study of orthogonal moments: Legendre moments, Zernike moments, pseudo-Zernike moments; and non-orthogonal moments: geometric moments, complex moments, rotation moments. Their results show that geometric moments, complex moments and pseudo-Zernike moments are less affected by noise, while Legendre moments are more severely affected by noise. Zernike moments and pseudo-Zernike moments have more reconstruction power than Legendre moments for both noisy and normal image. The results also show that the reconstruction error for noisy images reaches a minimum value and then starts to increase as the number of moments increases, indicating that higher order moments are less reliable in a noise environment. The number of moments where the error reaches a minimum depends on the signal-to-noise ratio. This provides a clue for selecting an optimal number of moments to describe shape. For example, under the noise level of $\mathrm{SNR} = 30$, the optimal number of Zernike moments for shape description is 10. If $\mathrm{SNR} = 200$, there is little difference between using 30 moments and using over 30 moments. Their conclusion is that Zernike moments and pseudo-Zernike moments are the preferable shape descriptors.

Liao and Pawlak [78] extend Teh and Chin's work by introducing techniques to increase the accuracy and efficiency of moments. Specifically, they study moment accuracy under different image resolutions. Their results show that coarser quantisation of image produces more accurate moments. They also introduce the *alternative extended Simpson's rule* to speed up calculation of higher order moments.

Moment shape descriptors are usually concise, robust, easy to compute and match. The disadvantage of moment methods is that it is difficult to correlate high order moments with the shape's physical features. Among the many moment shape descriptors, Zernike moments are the most desirable for shape description. Due to the incorporation of a sinusoid function into the kernel, they have similar properties of spectral features which are well understood. Shape description using Zernike moments proves to be very promising [87,88]. Zernike moment descriptor has been adopted by MPEG-7 as a region-based shape descriptor [33].

4.1.4. Generic Fourier descriptor

Although Zernike moment descriptor has a robust performance, it has several shortcomings. First, the kernel of Zernike moments is complex to compute, and the shape has to be normalized into a unit disk before deriving the moment features. Second, the radial features and circular features captured by Zernike moments are not consistent, one is in spatial domain and the other is in spectral domain. It does not allow multi-resolution analysis of a shape in radial direction. Third, the circular spectral features are not captured evenly at each order, this can result in loss of significant features which are useful for shape description. To overcome these shortcomings, a generic Fourier descriptor (GFD) has been proposed by Zhang and Lu [89]. The GFD is acquired by applying a 2-D Fourier transform on a polar-raster

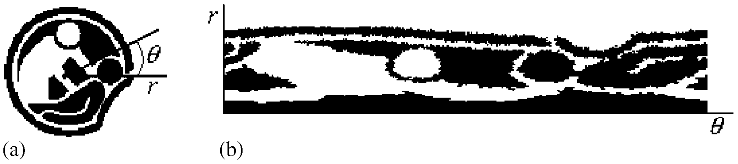

Fig. 10. (a) An original shape in polar space; (b) polar-raster sampled image plotted in Cartesian space.

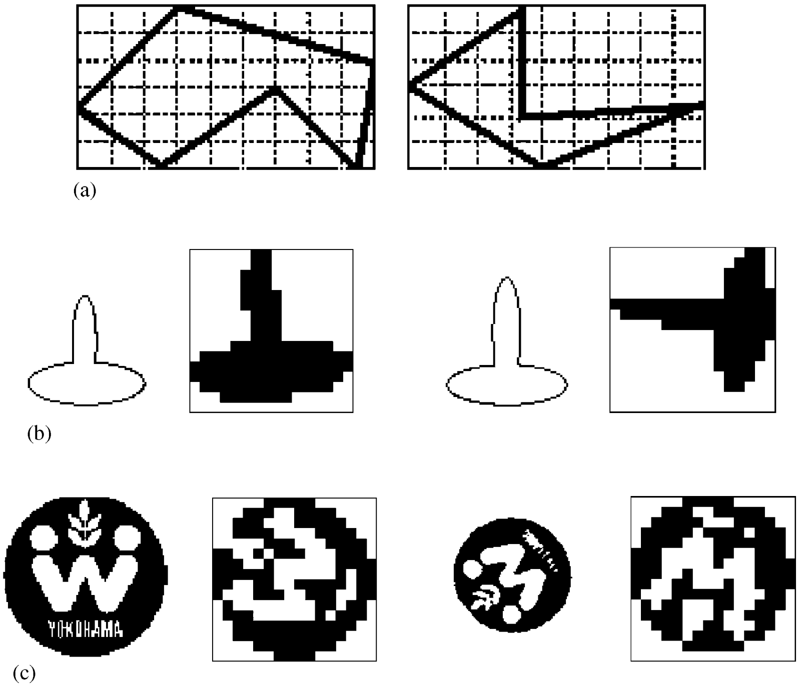

Fig. 11. (a) Grid representation of two contour shapes; (b) two similar contour shapes with different grid representations; (c) two similar region shapes with different grid representations.

sampled shape image (Fig. 10):

$$PF_2(\rho,\phi)=\sum_r\sum_i f(r,\theta_i)\exp\left[j2\pi\left(\frac{r}{R}\rho+\frac{2\pi i}{T}\phi\right)\right], \tag{4.4}$$

where $0 \leqslant r < R$ and $\theta_i = i(2\pi/T)$ $(0 \leqslant i < T)$; $0 \leqslant \rho < R$, $0 \leqslant \phi < T$. R and T are the radial frequency resolution and angular frequency resolution respectively. The normalized coefficients are the GFD. The similarity between two shapes are measured by the *city block* distance between their GFDs.

Compared with Zernike moments, GFD is simpler to compute, the features are pure spectral features and have better retrieval performance due to multi-resolution analysis in both radial and circular directions of the shape. With an enhanced process, GFD can achieve retrieval performance on perspectively transformed shapes as high as it achieves on similarity transformed shapes [90]. Zhang and Lu have also shown that GFD outperforms contour shape descriptors such as CSS, FD and region-based shape descriptors such as Zernike moments, geometric moments and grid method [57].

4.1.5. Grid based method

The grid shape descriptor is proposed by Lu and Sajjanhar [91] and has been used in [92–94]. Basically, a grid of cells is overlaid on a shape, the grid is then scanned from left to right and top to bottom. The result is a bitmap. The cells covered by the shape are assigned 1 and those not covered by the shape are assigned 0. The shape can then be represented as a binary feature vector. The *binary Hamming distance* is used to measure the similarity between two shapes. For example, the grid descriptors for the two shapes in Fig. 11(a) 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 the XOR operation on the two sets. In order to accommodate translation, rotation and scaling of the shape, the shape is first normalized before scanning. The shape is scaled into a fixed size rectangle, shifted to the upper left of the rectangle and rotated so that the major axis of the shape is horizontal. Mirrored and flipped shapes should be considered separately.

Chakrabarti et al. [92] improves grid descriptor by using an adaptive resolution (AR) representation and used it in MARS [94]. The AR grid descriptor is acquired by applying quadtree decomposition on the bitmap representation of the shape.

The advantages of the grid descriptor are its simplicity in representation, conformance to intuition, and also agreement with shape coding method in MPEG-4. The main problem with this method is the major-axis based rotation normalization. The major axis is sensitive to noise and unreliable. For example, the two similar shapes in (b) have very different grid representations. For region-based shapes, the grid representation is not rotation invariant. For example, the two shapes in Fig. 11(c) are similar shapes, however, their grid representations are very different.

4.1.6. Shape matrix

Normal shape methods use rectangular grid sampling to acquire shape information. The shape representation derived this way is usually not translation, rotation and scaling invariant. Extra normalization is therefore required. Goshtasby proposes the use of a *shape matrix* which is derived from a circular raster sampling technique [95]. The idea is similar to normal raster sampling. However, rather than overlay the normal square grid on a shape image, a polar raster of concentric circles and radial lines is overlaid in the center of the mass (Fig. 12(a)). The binary value of the shape is sampled at the intersections of the circles and radial lines. The shape matrix is formed so that the circles correspond to the matrix columns and the radial lines correspond to the matrix rows. Prior to the sampling, the shape is scale normalized using the maximum radius of the shape. The result matrix representation is invariant to translation, rotation, and scaling. Since the sampling density is not constant with the polar sampling raster, Taza and Suen represent shape using a weighed shape matrix which gives more weight to peripheral samples [96].

Since a shape matrix is a sparse sampling of shape, it is easily affected by noise. Besides, shape matching using a shape matrix is too expensive. Perui et al. propose a shape description based on the relative areas of the shape contained in concentric rings located in the shape center of the mass [97,98]. Let L be the maximum radius of the shape S to be described, C_k, be the kth ring of n concentric rings obtained by sectioning the maximum radius L into n equal segments. An area-ratio invariant is defined as

$$x_i = \frac{A(S \cap C_i)}{A(C_i)}, \tag{4.5}$$

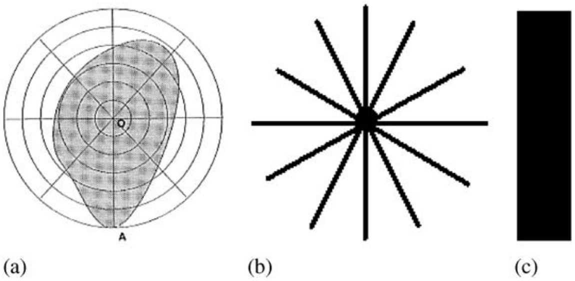

Fig. 12. (a) polar raster sampling of shape (reprinted from [97]); (b) a star shape formed by line strips; (c) a rectangle shape.

where $A(\cdot)$ is the area function. The shape descriptor is the feature vector of $x = [x_1, \ldots, x_n]^{\mathrm{T}}$. Although the area ratio descriptor is more compact and robust than the shape matrix, it ignores the pattern alterations within the measured ring. Consequently, the two shapes in Fig. 12(b) and (c) will be the same under this descriptor. The problems presented here can be solved using the spectral transform discussed in Section 4.1.4.

4.1.7. Discussions

Global region based methods treat the shape region as a whole and make effective use of all the pixel information within the region. These methods measure pixel distribution of the shape region, which are less likely affected by noise and variations. As a result, they usually can cope well with shape of significant defection, which poses a problem for contour-based methods. Particularly popular region methods are moment methods. Moment methods extract statistical distribution of region pixels. The lower order moments or moment invariants carry physical meanings associated with region pixel distribution. However, it is difficult to associate higher order moments with physical interpretation. Grid methods are subject to noise due to the use of the major axis for normalization, and it is not rotation invariant for region-based shapes. Shape matrix methods are not robust due to the sparse sampling approach. Although the area-ratio invariant is more robust than shape matrix representation, it does not capture pattern alterations effectively. The use of spectral transforms on the densely polar raster sampled image takes the advantages of both the moment methods and shape matrix methods, while overcoming the problems presented in all the other region-based methods discussed in this paper.

4.2. Structural methods

Similar to the contour structural methods, region-based structural methods decompose the shape region into parts which are then used for shape representation and description.

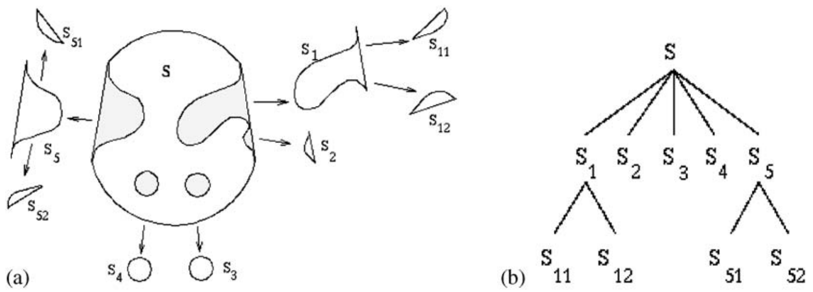

Fig. 13. (a) Convex hull and its concavities; (b) Concavity tree representation of convex hull (reprinted from [14]).

4.2.1. Convex hull

A region R is convex if and only if for any two points $\mathbf{x}_1, \mathbf{x}_2 \in R$, the whole line segment $\mathbf{x}_1\mathbf{x}_2$ is inside the region. The *convex hull* of a region is the smallest convex region H which satisfies the condition $R \subset H$. The difference $H - R$ is called the *convex deficiency D* of the region R. The extracting of the convex hull can use both boundary tracing method [14] and morphological methods [11,15]. Since shape boundaries tend to be irregular because of digitization, noise and variations in segmentation, these effects usually result in a convex deficiency that has small, meaningless components scattered randomly throughout the boundary. Common practice is to first smooth a boundary prior to partitioning. The polygon approximation is particularly attractive, because it can reduce the computation time of extracting the convex hull from $O(n^2)$ to $O(n)$ (n is the number of points in the shape). The extraction of convex hull can be a single process which finds significant convex deficiencies along the boundary. The shape can then be represented by a string of concavities. A fuller representation of the shape may be obtained by a recursive process which results in a concavity tree. Here the convex hull of an object is first obtained with its convex deficiencies, then the convex hulls and deficiencies of the convex deficiencies are found, then the convex hulls and deficiencies of these convex deficiencies—and so on until all the derived convex deficiencies are convex. Fig. 13(a) illustrates this process. The shape is then represented as a concavity tree (Fig. 13(b)). Each concavity can be described by its area, bridge (the line connects the cut of the concavity) length, maximum curvature, distance from maximum curvature point to the bridge. The matching between shapes becomes a string or a graph matching.

4.2.2. Medial axis

Like the convex hull, a region *skeleton* can also be employed for shape representation and description. A skeleton may be defined as a connected set of medial lines along the limbs of a figure [11]. For example, in the case of thick hand-drawn characters, the skeleton may be taken to be the path actually traveled by the pen. In fact, the basic idea of the skeleton is that of eliminating redundant information while retaining only the topological information concerning the structure of the object that can help with recognition. The skeleton methods are represented by Blum's *medial axis transform* (MAT) [99]. The medial axis is the locus of centers of maximal disks that fit within the shape. This is illustrated in Fig. 14. The bold line in the figure is the skeleton of the shaded rectangular shape. The skeleton can then be decomposed into segments and represented as a graph according to certain criteria. The matching between shapes becomes a graph matching. The computation of the medial axis is a rather challenging problem. In addition, medial axis tends to be very sensitive to boundary noise and variations. Preprocessing the contour of the shape and finding its polygonal approximation has been suggested as a way of overcoming these problems. But, as has been pointed out by Pavlidis [100], obtaining such polygonal approximation can be quite sufficient in itself for shape description.

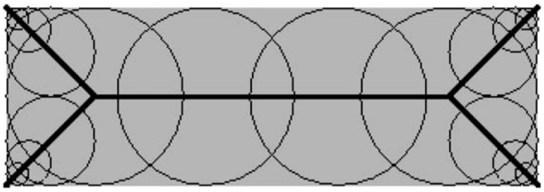

Fig. 14. Construction of the medial axis of a rectangular shape. (reprinted from [101]).

The medial axis obtained by Morse is computed from scale space. The medial axis acquired in this way is called the *core* of the shape [101].

4.2.3. Discussions

The region structural methods have similar problems to the contour structural approach. Apart from their complex computation and implementation, the graph matching is also an issue which needs to be solved itself. It should be noted, both types of region structural methods described here need to know shape contour information.

5. Summary and conclusions

In this paper, existing shape representation and description techniques have been reviewed. Generally, there are two classes of approaches in shape representation and description: contour-based versus region-based. Under each class, the methods can be divided into structural and global methods. The different methods can be further distinguished between methods working in space domain and methods working in transform domain.

Contour-based approaches are more popular than region-based approaches in literature. This is because human beings are thought to discriminate shapes mainly by their contour features. Another reason is because in many of the shape applications, the shape contour is the only interest, whilst the shape interior content is not important. However, there are several limitations with contour-based methods. First, contour shape descriptors are generally sensitive to noise and variations because they only use a small part of shape information, that is, contour information. Second, in many cases, the shape contour is not available. Third, in some applications, shape content is more important than the contour features. These limitations can be overcome by using region-based methods. The findings in the survey are in favor of region-based methods. Region-base methods are more robust as they use all the shape information available; they can be applied to general applications; and they generally provide more accurate retrieval. In addition, region-based methods can cope well with shape defection, which is a common problem for contour-based shape representation techniques. Although region-based methods make use of all the shape information, it is not necessarily more complex than contour-based methods, as some promising methods such as the moment methods and GFD are simple to implement.

Compared with global approaches, structural approaches are too complex to implement. The high indexing and matching complexity makes them a family of unstable shape representations. For example, in a typical structural representation of shape, there are at least four ad hoc parameters involved in the different processes of the feature acquisition and matching. These parameters are: the threshold in the smoothing process, the threshold in the separation of shape into primitives, the threshold in the primitive matching process, and the threshold of determining similarity between primitives of the two shapes in the model matching process. The fact that the majority of the structural approaches use angle as a feature adds another parameter into the system to accommodate the rotation invariant. These parameters need to be finely tuned for different applications. The advantage of structural approaches, however, is that they can do partial matching. This is useful when the contour of the boundary is not closed and large part of the shape is missing or occluded. Partial matching can also be done using methods based on the Hausdorff distance, which is a more expensive matching method. However, the advantage of methods based on the Hausdorff distance is that they can match two general images based on the images' edge maps, which are usually unconnected or even scattered. Hausdorff distance methods can be used to locate a given object in an image, or to match a sub-image in a large image.

In the review, it has been found that methods working in spatial domain suffer from two main drawback: noise sensitivity and high dimension. The problems can be solved in four ways: histogram, moments, scale space and spectral transforms. Among the four solutions, spectral transforms, especially Fourier transform, is the most promising. Although histogram and scale space increase robustness to noise and compactness, matching using histogram and scale space can be very expensive. Moments is robust and compact, however, higher order moments are either difficult to obtain or without physical meaning. Shape representation using Fourier descriptor, either in 1-D space or 2-D space, is simple to compute, robust and compact. Fourier descriptors can be constructed to arbitrary order and all the Fourier descriptors have physical meanings. The retrieval performance of the generic Fourier descriptor (GFD) demonstrates it is a desirable solution to generic shape representations, regardless they are contour-based shapes (without interior content) or region-based shapes (with interior content). The hierachical representation of GFD can achieve the same retrieval efficiency as those 1-D methods like moments, scale space or FD. Although GFD cannot do partial matching, it works well in situations where a significant part of a shape has been missing or occluded.

In summary, structural approaches are useful in applications where partial matching is needed; methods based on the Hausdorff distance are useful for locating objects in an image or sub-image matching. Both types of methods have limited applications. For general shape applications, methods based on complex moments and spectral transforms, such as Zernike moments and GFD, are the best choices. They satisfy the six principles set by MPEG-7: good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation. If storage is a concern, FD can be considered.

Acknowledgements

The authors thank Mr Templar Hankinson of Gippsland School of Computing and Info Tech, Monash University for his valuable help during the preparation of the manuscript. The authors are also grateful for the constructive and valuable comments from the reviewer of this paper.

References

[1] H. Kim, J. Kim, Region-based shape descriptor invariant to rotation, scale and translation, Signal Process. Image Commun. 16 (2000) 87–93.

[2] I. Yong, J. Walker, J. Bowie, An analysis technique for biological shape, Comput. Graphics Image Process. 25 (1974) 357–370.

[3] M. Peura, J. Iivarinen, Efficiency of simple shape descriptors, in: Proceedings of the Third International Workshop on Visual Form, Capri, Italy, May, 1997, pp. 443–451.

[4] D. Chetverikov, Y. Khenokh, Matching for shape defect detection, Lecture Notes in Computer Science, Vol. 1689, Springer, Berlin, 1999, pp. 367–374.

[5] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Comparing images using the Hausdorff distance, Technical Report, CUCS-TR-91-1211, Department of Computer Science, Cornell University, 1991.

[6] D.P. Huttenlocher, W.J. Rucklidge, A multi-resolution technique for comparing images using the Hausdorff distance, Technical Report, TR-92-1321, Department of Computer Science, Cornell University, 1992.

[7] T. Meier, Segmentation for video object plane extraction and reduction of coding artifacts, Ph.D. Thesis, Department of Electrical and Electronic Engineering, University of Western Australia, Australia, 1998.

[8] W.J. Rucklidge, Efficient locating objects using Hausdorff distance, Int. J. Comput. Vision 24 (3) (1997) 251–270.

[9] B. Scassellati, S. Slexopoulos, M. Flickner, Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgments, in: SPIE Proceedings on Storage and Retrieval for Image and Video Databases II, Vol. 2185, San Jose, CA, USA 1994, pp. 2–14.

[10] S. Belongie, J. Malik, J. Puzicha, Matching shapes, in: Proceedings of Eighth IEEE International Conference on Computer Vision (ICCV2001), Vol. I, Vancouver, Canada, July, 2001, pp. 454–461.

[11] E.R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Academic Press, New York, 1997, pp. 171–191.

[12] P.J. van Otterloo, A Contour-Oriented Approach to Shape Analysis, Prentice-Hall International (UK) Ltd, Englewood Cliffs, NJ, 1991, pp. 90–108.

[13] D.S. Zhang, G. Lu, A comparative study of Fourier descriptors for shape representation and retrieval, in: Proceedings of the Fifth Asian Conference on Computer Vision (ACCV02), Melbourne, Australia, January 22–25, 2002, pp. 646–651.

[14] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, Chapman & Hall, London, UK, NJ, 1993, pp. 193–242.

[15] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992, pp. 502–503.

[16] A. Del Bimbo, P. Pala, Visual image retrieval by elastic matching of user sketches, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 121–132.

[17] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, The QBIC project: querying image by content using color, texture and shape, in: Proceedings of SPIE Storage and Retrieval for Image and Video Databases, Vol. 1908, San Jose, CA, USA 1993, pp. 173–187.

[18] K. Hirata, T. Kato, Query by visual example—content based image retrieval, in: Proceedings of the Third International Conference on Extending Database Technology (EDBT'92), Vienna, Austria, 1992, pp. 56–71.

[19] R. Chellappa, R. Bagdazian, Fourier coding of image boundaries, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1) (1984) 102–105.

[20] M. Das, M.J. Paulik, N.K. Loh, A bivariate autoregressive modeling technique for analysis and classification of planar shapes, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1) (1990) 97–103.

[21] S.R. Dubois, F.H. Glanz, An autoregressive model approach to two-dimensional shape classification, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 627–637.

[22] K. EOM, J. Park, Recognition of shape by statistical modeling of centroidal profile, Proceedings of the Tenth International Conference on Pattern Recognition, Vol. 1, Atlantic City, NJ, 1990, pp. 860–864.

[23] K.L. Kashyap, R. Chellappa, Stochastic models for closed boundary analysis: representation and reconstruction, IEEE Trans. Inform. Theory 27 (1981) 627–637.

[24] Y. He, A. Kundu, 2—D shape classification using hidden Markov model, IEEE Trans. Pattern Anal. Mach. Intell. 13 (11) (1991) 1172–1184.

[25] I. Sekita, T. Kurita, N. Otsu, Complex autoregressive model for shape recognition, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 489–496.

[26] H. Asada, M. Brady, The curvature primal sketch, MIT AI Memo 758, 1984.

[27] H. Asada, M. Brandy, The curvature primal sketch, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1) (1986) 2–14.

[28] F. Mokhtarian, A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, IEEE Pattern Anal. Mach. Intell. 8 (1) (1986) 34–43.

[29] F. Mokhtarian, S. Abbasi, J. Kittler, Robust and efficient shape indexing through curvature scale space, Proceedings of the British Machine Vision Conference, Edinburgh, UK, 1996, pp. 53–62.

[30] F. Mokhtarian, S. Abbasi, J. Kittler, Efficient and robust retrieval by shape content through curvature scale space, International Workshop on Image DataBases and Multimedia Search, Amsterdam, The Netherlands, 1996, pp. 35–42.

[31] S. Abbasi, F. Mokhtarian, J. Kittler, Curvature scale space image in shape similarity retrieval, Multimedia Systems 7 (1999) 467–476.

[32] S. Abbassi, F. Mokhtarian, J. Kittler, Enhancing CSS-based shape retrieval for objects with shallow concavities, Image Vision Comput. 18 (2000) 199–211.

[33] S. Jeannin (Ed.), MPEG-7 Visual part of experimentation model version 5.0, ISO/IEC JTC1/SC29/WG11/N3321, Nordwijkerhout, March, 2000.

[34] M. Daoudi, S. Matusiak, Visual image retrieval by multiscale description of user sketches, J. Visual Lang. Comput. 11 (2000) 287–301.

[35] D.H. Eberly, Geometric methods for analysis of ridges in N-dimensional images, Ph.D. Thesis, University of North Carolina at Chapel Hill, 1994.

[36] K. Arbter, Affine-invariant Fourier descriptors, in: J.C. Simon (Ed.), From Pixels to Features, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1989, pp. 153–164.

[37] K. Arbter, W.E. Snyder, H. Burkhardt, G. Hirzinger, Application of affine-invariant Fourier descriptors to recognition of 3-D objects, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 640–647.

[38] E.L. Brill, Character recognition via Fourier descriptors, WESCON, Session 25, Qualitative Pattern Recognition

Through Image Shaping, Los Angeles, CA, 1968, pp. (25/3)1–10.

[39] G. Eichmann, et al., Shape representation by Gabor expansion, Hybrid Image and Signal Processing II, SPIE Vol. 1297, Orlando, Florida, USA 1990, pp. 86–94.

[40] G. Granlund, Fourier preprocessing for hand print character recognition, IEEE Trans. Comput. 21 (1972) 195–201.

[41] H. Kauppinen, T. Seppanen, M. Pietikainen, An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification, IEEE Trans. Pattern Anal. Mach. Intell. 17 (2) (1995) 201–207.

[42] A. Krzyzak, S.Y. Leung, C.Y. Suen, Reconstruction of two dimensional patterns from Fourier Descriptors, Mach. Vision Appl. 2 (1989) 123–140.

[43] C.C. Lin, R. Chellappa, Classification of partial 2D shapes using Fourier descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 9 (5) (1987) 686–690.

[44] C.S. Lin, C.L. Hwang, New forms of shape invariants from elliptic Fourier descriptors, Pattern Recognition 20 (5) (1987) 535–545.

[45] F.J.S. Marine, Automatic recognition of biological shapes with and without representation of shape, Artif. Intell. Med. 18 (2000) 173–186.

[46] B.M. Mehtre, M.S. Kankanhalli, W.F. Lee, Shape measures for content based image retrieval: a comparison, Inf. Process. Manage. 33 (3) (1997) 319–337.

[47] O.R. Mitchell, T.A. Grogan, Global and partial shape discrimination for computer vision, Opt. Eng. 23 (5) (1984) 484–491.

[48] E. Persoon, K.S. Fu, Shape discrimination using Fourier descriptors, IEEE Trans. System Man Cybernet. SMC-7 (3) (1977) 170–179.

[49] T.W. Rauber, Two-dimensional shape description, Technical Report: GR UNINOVA-RT-10-94, University Nova de Lisboa, Portugal, 1994.

[50] C.W. Richard, H. Hemami, Identification of three-dimensional objects using Fourier descriptors of the boundary curve, IEEE Trans. System Man Cybernet. SMC-4 (4) (1974) 371–378.

[51] Y. Rui, A.C. She, T.S. Huang, A modified Fourier descriptor for shape matching in MARS, in: A.W.M. Smeulders, R. Jain (Eds.), Image Databases and Multimedia Search, World Scientific Publishing Co., Singapore, 1997, pp. 165–177.

[52] C.T. Zahn, R.Z. Roskies, Fourier descriptors for plane closed curves, IEEE Trans. Comput. C-21 (3) (1972) 269–281.

[53] D.S. Zhang, G. Lu, A comparison of shape retrieval using Fourier descriptors and short-time Fourier descriptors, in: Proceedings of the Second IEEE Pacific-Rim Conference on Multimedia (PCM01), Beijing, China, October 24–26, 2001, pp. 855–860.

[54] J.R. Ohm, F.B. Bunjamin, W. Liebsch, B. Makai, K. Muller, A. Somlic, D. Zier, A set of visual feature descriptors and their combination in a low-level description scheme, Signal Process. Image Commun. 16 (2000) 157–179.

[55] Q.M. Tieng, W.W. Boles, Recognition of 2D object contours using the wavelet transform zero-crossing representation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (8) (1997) 910–916.

[56] H.S. Yang, S.U. Lee, K.M. Lee, Recognition of 2D object contours using starting-point-independent wavelet coefficient matching, J. Visual Commun. Image Represent. 9 (2) (1998) 171–181.

[57] D.S. Zhang, Image retrieval based on shape, Ph.D. Thesis, Monash University, Australia, March, 2002.

[58] T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, MD, 1982, p. 143.

[59] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Trans. Electron. Comput. EC-10 (1961) 260–268.

[60] H. Freeman, A. Saghri, Generalized chain codes for planar curves, in: Proceedings of the Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan, November 7–10, 1978, pp. 701–703.

[61] J. Iivarinen, A. Visa, Shape recognition of irregular objects, in: D.P. Casasent (Ed.), Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904 (1996) 25–32.

[62] W.I. Groskey, R. Mehrotra, Index-based object recognition in pictorial data management, Comput. Vision Graphics Image Process. 52 (1990) 416–436.

[63] W.I. Groskey, P. Neo, R. Mehrotra, A pictorial index mechanism for model-based matching, Data Knowledge Eng. 8 (1992) 309–327.

[64] R. Mehrotra, J.E. Gary, Similar-shape retrieval in shape data management, IEEE Comput. 28 (9) (1995) 57–62.

[65] S. Berretti, A.D. Bimbo, P. Pala, Retrieval by shape similarity with perceptual distance and effective indexing, IEEE Trans. Multimedia 2 (4) (2000) 225–239.

[66] G. Dudek, J.K. Tsotsos, Shape representation and recognition from multiscale curvature, Comput. Vision Image Understanding 68 (2) (1997) 170–189.

[67] K.S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.

[68] N. Chomsky, Syntactic Structures, Mouton, The Hague, Berlin, 1957.

[69] S.Z. Li, Shape matching based on invariants, in: O. Omidvar (Ed.), Shape Analysis, Progress in Neural Networks, Vol. 6, Ablex, Norwood, NJ, 1999, pp. 203–228.

[70] C.-L. Huang, D.-H. Huang, A content-based image retrieval system, Image Vision Comput. 16 (1998) 149–163.

[71] D.M. Squire, T.M. Caelli, Invariance signature: characterizing contours by their departures from invariance, Comput. Vision Image Understanding 77 (2000) 284–316.

[72] S.Z. Li, Matching: invariant to translations, rotations and scale changes, Pattern Recognition 25 (1992) 583–594.

[73] M. Kliot, E. Rivlin, Invariant-based shape retrieval in pictorial databases, Comput. Vision Image Understanding 71 (2) (1998) 182–197.

[74] http://www.ee.surrey.ac.uk/Research/VSSP/imagedb.

[75] M.K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inf. Theory IT-8 (1962) 179–187.

[76] R.L. Kennedy, Y. Lee, B.V. Roy, C.D. Reed, R.P. Lippmann, Solving Data Mining Problems Through Pattern Recognition, Prentice-Hall, PTR, Upper Saddle River, NJ, 1998, pp. 11–16 (Chapter 9).

[77] B. Jahne, Digital Image Processing—Concepts, Algorithms and Scientific Applications, Springer, Berlin, Heidelberg, 1997, pp. 509–512.

[78] S.X. Liao, M. Pawlak, On image analysis by moments, IEEE Trans. Pattern Anal. Mach. Intell. 18 (3) (1996) 254–266.

[79] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Am. 70 (8) (1980) 920–930.

[80] C.-H. Teh, R.T. Chin, On image analysis by the methods of moments, IEEE Trans. Pattern Anal. Mach. Intell. 10 (4) (1988) 496–513.

[81] A.S. Dudani, K.J. Breeding, R.B. McGhee, Aircraft identification by moment invariants, IEEE Trans. Comput. C-26 (1) (1977) 39–46.

[82] S.O. Belkasim, M. Shridhar, M. Ahmadi, Pattern recognition with moment invariants: a comparative study and new results, Pattern Recognition 24 (12) (1991) 1117–1138.

[83] R.J. Prokop, A.P. Reeves, A survey of moment-based techniques for unoccluded object representation recognition, Graph. Models Image Process. 54 (1992) 438–460.

[84] A. Sajjanhar, A technique for similarity retrieval of shapes, Master Thesis, Monash University, Australia, 1997.

[85] G. Taubin, D.B. Cooper, Recognition and positioning of rigid objects using algebraic moment invariants, SPIE Conference on Geometric Methods in Computer Vision, Vol. 1570, University of Florida, Florida, USA 1991, pp. 175–186.

[86] G. Taubin, D.B. Cooper, Object recognition based on moment (or Algebraic, in: J. Mundy, A. Zisserman (Eds.), Geometric Invariance in Computer Vision, MIT Press, Cambridge, MA, 1992, pp. 375–397.

[87] Y.-S. Kim, W.-Y. Kim, Content-based trademark retrieval system using a visually salient feature, Image Vision Comput. 16 (1998) 931–939.

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

[89] D.S. Zhang, G. Lu, Generic Fourier descriptor for shape-based image retrieval, in: Proceedings of IEEE International Conference on Multimedia and Expo (ICME2002), Vol. 1, Lausanne, Switzerland, August 26–29, 2002, pp. 425–428.

[90] D.S. Zhang, G. Lu, Enhanced generic Fourier descriptor for object-based image retrieval, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002), Vol. 4, Orlando, FL, USA, May 13–17, 2002, pp. 3668–3671.

[91] G.J. Lu, A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval, Multimedia Syst. 7 (2) (1999) 165–174.

[92] K. Chakrabarti, M.O. Binderberger, K. Porkaew, S. Mehrotra, Similar shape retrieval in MARS, in: Proceedings of IEEE International Conference on Multimedia and Expo, Vol. 2, New York, USA, 2000, pp. 709–712.

[93] M. Safar, C. Shahabi, X. Sun, Image retrieval by shape: a comparative study, in: Proceedings of IEEE International Conference on Multimedia and Expo, Vol. 1, New York, USA, 2000, pp. 141–144.

[94] T.S. Huang, S. Mehrotra, K. Ramachandran, Multimedia analysis and retrieval system (MARS) project, in: Proceedings of 33rd Annual Clinic on Library Application of Data Processing—Digital Image Access and Retrieval, University of Illinoisat Urbana-Champaign, Illinois, USA 1996, pp. 101–117.

[95] A. Goshtasby, Description and discrimination of planar shapes using shape matrices, IEEE Trans. Pattern Anal. Mach. Intell. 7 (1985) 738–743.

[96] A. Taza, C. Suen, Discrimination of planar shapes using shape matrices, IEEE Trans. Syst. Man Cybern. 19 (1989) 1281–1289.

[97] S. Loncaric, A survey of shape analysis techniques, Pattern Recognition 31 (8) (1998) 983–1001.

[98] S. Parui, E. Sarma, D. Majumder, How to discriminate shapes using the shape vector, Pattern Recognition Lett. 4 (1986) 201–204.

[99] H. Blum, A transformation for extracting new descriptors of shape, in: W. Whaten-Dunn (Ed.), Models for the Perception of Speech and Visual Forms, MIT Press, Cambridge, MA, 1967, pp. 362–380.

[100] T. Pavlidis, Algorithms for shape analysis of contours and waveforms, IEEE Trans. Pattern Anal. Mach. Intell. 2 (4) (1980) 301–312.

[101] B.S. Morse, Computation of object cores from grey-level images, Ph.D. Thesis, University of North Carolina at Chapel Hill, 1994.

About the Author—DR. DENGSHENG ZHANG received B.Sc. in Mathematics and B.A. in English in 1985 and 1987, respectively, both from China. He spent 12 years on teaching Math and Computing before he was involved in his Ph.D. program in 1999. He received Ph.D. in Computer Technology from Monash University, Australia, in 2002. He is now a lecturer in Gippsland School of Computing and Info Tech of Monash University. Dr. Zhang has over 10 years research experience in the area of multimedia and has published over 20 referred international journal and conference papers. His main research interests include OCR, CSCW, image/video retrieval.

About the Author—DR. GUOJUN LU obtained his Ph.D. in 1990 from Loughborough University of Technology, and BEng in 1984 from Nanjing Institute of Technology (now South East University). He is currently an associate professor at Gippsland School of Computing and Information Technology, Monash University, Australia. He has held positions in Loughborough University of Technology, National University of Singapore, and Deakin University. Dr. Lu's main research interests are in multimedia information indexing and retrieval, multimedia data compression, quality of service management, and multimedia compression. He has published over 50 technical papers in these areas and authored the books Communication and Computing for Distributed Multimedia Systems (Artech House, 1996), and Multimedia Database Management Systems (Artech House, to appear in 1999). He has over ten years research experience in multimedia computing and communications. Projects that he has worked on include UK Alvey project UNISON (on multimedia networking), European RACE project MultiMed (on multimedia applications in telemedicine), fractal image compression, networked multimedia synchronization, and integrated image retrieval systems.