Chapter 3

Review of Shape Representation and Description Techniques

3.1 Introduction

Shape is an important visual feature and it is one of the primitive features for image content description. However, shape content description is a difficult task. Because it is difficult to define perceptual shape features and measure the similarity between shapes. To make the problem more complex, shape is often corrupted with noise, defection, arbitrary distortion and occlusion.

Shape research has been an active research area for over 30 years. In the past, shape research has been driven mainly by object recognition. As a result, techniques of shape representation and description mostly target at particular applications. Effectiveness or accuracy is the main concern of these techniques. In contrast, in the newly emerged multimedia application CBIR, efficiency is envisaged equally important as effectiveness due to online retrieval demand. In the development of MPEG-7, six requirements has been set to measure a shape descriptor, they are: good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation. In this chapter, shape representation and description techniques in literature are reviewed. Promising shape descriptors for image
retrieval are identified according to the above requirements. Since shape based image retrieval is a sub-topic of CBIR, an overview of CBIR research is given in the first.

### 3.2 Overview of CBIR

Kato [KAT92] first used the term *content-based image retrieval* (CBIR) to describe automatic retrieval of images from a database by color and shape features. The term has since been widely used to describe retrieving desired images from a large image collection on the basis of features that can be automatically extracted from the images themselves. Since then, several commercial CBIR systems and CBIR research prototypes have been available. The best-known and representative commercial CBIR system is the *QBIC* system developed by IBM [NIB93, FLI95]. It offers retrieval by any combination of color, texture or shape as well as by text keyword. Image queries can be formulated by selection from a palette, specifying an example image, or sketching a desired shape on the screen. The features are organized using R*-tree to improve search efficiency. The other well-known commercial CBIR systems are *Virage* [BAC96] which is used by Alsta Visa to facilitate image searching, and *Excalibur* [DOW93, FED96] which is adopted by Yahoo! for image searching.

*Photobook* [PEN96] is the representative research CBIR system developed at MIT. Like QBIC, images in the database are represented by color, shape, texture and other appropriate features. However, it calculates information-preserving features, from which all essential aspects of the original image can be reconstructed. The system provides retrieval of textures, shapes and human faces. Another research CBIR system is *VisualSEEk* developed at Columbia University by Smith and Chang [SMI96]. It offers searching by image region, color, shape and spatial locations, as well as keyword. Users can build up image queries by specifying areas of defined shape and color at absolute or relative locations within the image. Other research prototypes are:

- **Chabot** [OGL95]. It provides a combination of text-based and color-based access to a collection of digitized photographs.

- **MARS** [HUA96]. It was developed at University of Illinois by Huang et al. The system characterize each object within an image by a variety of features, and uses a range of different similarity measures to compare query and stored objects. User feedback is used to adjust feature weights, and if necessary to invoke different similarity measures.

- **Surfimage** [NAS98]. It was developed at INRIA, France. This has a similar philosophy to the MARS system, using multiple types of image features which can be combined in different ways, and offering sophisticated relevance feedback facilities.
• *Netra* [MA97]. It was developed by Ma and Manjunath, it uses color texture, shape and spatial location information to provide region-based searching based on local image properties derived from sophisticated image segmentation.

Currently, CBIR researches mainly focus on three topics: *feature extraction, efficient indexing* and *user interface*.

• **Feature extraction.** Features of image include *primitive features* and *semantic/logical features*. Primitive features such as *color, texture, shape* and *spatial relationships* are quantitative in nature, they can be extracted automatically or semi-automatically. Logical features are qualitative in nature and provide abstract representations of visual data at various levels of detail. Typically, logical features are extracted manually or semi-automatically [YAN01]. One or more features can be used in a specific application. For example, in a satellite information system, the texture features are important, while shape and color features are more important in trademark registration systems. Once the features have been extracted, the retrieval becomes a task of measuring similarity between image features.

• **Efficient indexing.** To facilitate efficient query and searching processing, the image indices needed to be organized into efficient data structure. Image features are approximately represented, they may not have embedded order, they may have interrelated multiple attributes. Therefore, flexible data structures should be used in order to facilitate storage/retrieval in a image retrieval system. Structures such as *k-d-tree, R-tree family, R*-tree, quad-tree, and grid file* are commonly used. Each structures has its advantages and disadvantages; some have limited domains and some can be used concurrently with others.

• **User interface.** In visual information systems, user interaction plays an important role in almost all of its functions. The user interface consists of a query processor and a browser to provide the interactive graphical tools and mechanisms for querying and browsing the database, respectively. Common query mechanisms provided by user interface are: *query by keyword, query by sketching, query by example, browsing by categories, feature selection and weighting, retrieval refining* and *relevance feedback*.

These three tasks comprise three key components of a CBIR system. Among these three main tasks, feature extraction (including similarity measurement) is the most paramount and difficult task. The majority of CBIR research goes into this challenging task. This research focuses on extraction of shape features. In the following, shape representation/description techniques are reviewed in details.
3.3 Classifications of Shape Representation Techniques

Several classification methods are available. The most common and general classification is based on the use of shape boundary points as opposed to shape interior points. The resulting classes are known as boundary and global [PAV80]. Shape representation techniques can also be distinguished between space domain and feature domain. Methods in space domain match shapes on point (or point feature) basis, while feature domain techniques match shapes on feature (vector) basis. Another classification of shape representation techniques is based on the basis of information preservation. Methods which allow for the accurate reconstruction of a shape from its descriptor are called information preserving (IP), while methods only capable of partial reconstruction or ambiguous description are called non-information preserving (NIP). For CBIR purpose, IP is not a requirement.

Figure 3.1. Classification of shape representation and description techniques.

In this work, a hierarchical classification approach is adopted. The variety of shape representation techniques in literature are first classified into contour-based method and region-based method based on whether shape features are extracted from contour only or are extracted from the whole shape region. Under each class, the different methods are further distinguished between structural and global based on whether the shape is represented as a whole or
represented by sub-parts (primitives). It is possible to further separate the different methods under the second hierarchy into feature domain and space domain based on whether shape is represented by points (or point features) or feature vector. The whole hierarchy of the classification is shown in Figure 3.1. In the next sections, these techniques are discussed in details.

3.4 Contour-based Shape Representation 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, a feature vector derived from the integral boundary is used to describe the shape. The measure of shape similarity is either point-based matching or feature-based matching. Discrete approaches break the shape boundary into segments, called primitives using a particular criterion. The 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, these two types of approaches are discussed.

3.4.1 Global Methods

Global contour shape representation techniques usually compute a multi-dimensional numeric feature vector from 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.

3.4.1.1 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 [YON74]. These simple global descriptors usually can only discriminate shapes with large dissimilarities, 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 Figure 3.2(a) is close to 1, it does not correctly describe the shape. In this case, circularity is a better descriptor. The two shapes in Figure 3.2 (b) and (c) have the same circularity, however, they are very different shapes. In this case, eccentricity is a better descriptor.
Other simple global contour shape descriptors are proposed by Peura and Iivarinen [PEU97]. These descriptors include convexity, ratio of principle axis, circular variance, elliptic variance.

![Diagrams](a) (b) (c)

Figure 3.2. Shape eccentricity and circularity.

### 3.4.1.2 Shape Signature

Shape signatures represent shape by a one-dimensional function derived from shape boundary points. Many shape signatures exist, including *complex coordinates, polar coordinates, central distance, tangent angle, cumulative angle, curvature, area and chord-length*. Shape signatures have been extensively studied in [DAV97, OTT91]. Shape signatures are usually normalized into being translation and scale invariant. Shape signature reduces shape matching in 2-D space into 1-D space. However, in order to compensate for orientation changes, the shift matching is needed to find the best matching between two shapes. Alternatively, signature can be quantized into a signature histogram, which is rotation invariant and can be used for matching [SQU00].

Shape signatures are normally sensitive to noise, slight changes in the boundary can cause large error in the matching. Therefore, it is impractical or undesirable to directly use shape signature for shape retrieval. Further process is necessary to reduce the matching load.

### 3.4.1.3 Boundary Moments

Boundary moments can be used to reduce the dimension of boundary representation. Assume shape boundary has been represented as a shape signature \( z(i) \), the \( r \)th moment \( m_r \) and central moment \( \mu_r \) can be estimated as [SON93]

\[
 m_r = \frac{1}{N} \sum_{i=1}^{N} [z(i)]^r \quad \text{and} \quad \mu_r = \frac{1}{N} \sum_{i=1}^{N} [z(i) - m_1]^r \quad (3.1)
\]

where \( N \) is the number of boundary points. The normalized moments \( \overline{m}_r = m_r / (\mu_2)^{r/2} \) and \( \overline{\mu}_r = \mu_r / (\mu_2)^{r/2} \) 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 [GON92] treats the amplitude of shape signature function \( z(i) \) as a random variable and creates a histogram \( p(v_i) \) from \( z(i) \). Then, the \( r \)th moment is obtained by

\[
\mu_r = \sum_{i=1}^{K} (v_i - m)^r p(v_i) \quad \text{and} \quad m = \sum_{i=1}^{K} v_i p(v_i) \tag{3.2}
\]

If the area under the signature function \( z(i) \) is normalized to unit area, then moment invariants in (3.2) is simplified as

\[
\mu_r = \sum_{i=1}^{N} (i - m)^r z(i) \quad \text{and} \quad m = \sum_{i=1}^{N} i z(i) \tag{3.3}
\]

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.

### 3.4.1.4 Elastic Matching

Bimbo and Pala has proposed the use of elastic matching for shape based image retrieval [BIM97]. In this approach, a deformed template is generated as the sum of the original template \( \tau(s) \) and a warping deformation \( \theta(s) \)

\[
\varphi(s) = \tau(s) + \theta(s) \tag{3.4}
\]

where \( \tau = (\tau_x, \tau_y) \) is a second order spline and \( \theta = (\theta_x, \theta_y) \) 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:

\[
F = S + B + M = \alpha \int_0^1 \left[ \left( \frac{d\theta_x}{ds} \right)^2 + \left( \frac{d\theta_y}{ds} \right)^2 \right] ds + \beta \int_0^1 \left[ \left( \frac{d^2\theta_x}{ds^2} \right)^2 + \left( \frac{d^2\theta_y}{ds^2} \right)^2 \right] ds + \int_0^1 I_E(\varphi(s)) ds \tag{3.5}
\]

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 \textit{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 have compared the computation complexity of feature extraction with QBIC [NIB93] and QVE [HIR92], 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 has been indicated in Section 3.4.1.1. The shape description is not rotation invariant. Besides, what warping criterion is used for the template deformation is not given. The examples of warping shown in the paper indicate the warping is arbitrary or application dependent.

### 3.4.1.5 Stochastic Method

Time-series models and especially autoregressive (AR) modeling have been used for calculating shape descriptors [KAS81, CHE84, DUB86, EOM90, DAS90, HE91, SEK92]. Methods in this class are based on the stochastic modeling of a 1-D function $f$ obtained from the shape as described in Section 3.4.1.2. A linear autoregressive model expresses a value of a function as a linear combination of a certain number of preceding values. Specifically, each function value in the sequence has some correlation with previous function values and can therefore be predicted through a number of, say, $M$ observations of previous function values. The autoregressive model is a simple prediction of the current radius by a linear combination of $M$ previous radii plus a constant term and an error term:

$$f_i = \alpha + \sum_{j=1}^{m} \theta_j f_{i-j} + \sqrt{\beta} \omega_i$$

where $\theta_j$ are the AR-model coefficients, $m$ is the model order, it tells how many preceding function values the model uses. $\sqrt{\beta} \omega_i$ is the current error term or residual, reflecting the accuracy of the prediction. $\alpha$ 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 [KAS81, CHE84, DUB86]. The
estimated $\theta_j$ are translation, rotation and scale invariant. Parameters $\alpha$ and $\beta$ are not scale invariant. But the quotient $\alpha/\sqrt{\beta}$, which reflects the signal-to-noise ratio of the boundary, is regarded as an invariant. Therefore, the feature vector $[\theta_1, \ldots, \theta_m, \alpha/\sqrt{\beta}]^T$ is used as the shape descriptor.

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

### 3.4.1.6 Scale Space Method

The problem of noise sensitivity and boundary variations in most spatial domain shape methods inspire the use of scale space analysis. The scale space representation of a shape is created by tracking the position of inflection points in shape boundary filtered by low-pass Gaussian filters of variable widths. The inflection points that remain present in the representation are expected to be ‘significant’ object characteristics. The result is usually an interval tree, called ‘fingerprint’, consisting of inflection points (Figure 2.3). The difficulty with this type of approach is the interpretation of the final result.

Asada and Brady first attempt to interpret the interval tree acquired from scale space [ASA84, ASA86]. Their interval tree 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. Since the shapes under analysis are from hardware application, it is possible to interpret high level primitive events from the detected peaks. In this application, the primitive events are defined as corner, smooth join, end, crank, bump/dent.

Mokhtarian and Mackworth [MOK86] adopt Asada and Brady’s interpretation method and extend it to shoreline registration. They call the acquired scale space signature as curvature scale space (CSS) contour image. The peaks of individual branches in the CSS are detected. However, instead of interpreting these peaks as high 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 affine transformed to match with the model curve. The method is later extended for shape retrieval [MOK96a, MOK96b, ABB99, ABB00]. Since the matching algorithm is too complex for retrieval, the new method develops a matching algorithm based on the two highest peaks in each of the two
matching images. All the images under analysis are also scaled into the same size (same number of boundary points) prior to applying scale space.

Daoudi and Matusiak interpret the interval tree obtained from scale space as a geodesic topology [DAO00]. The matching between two shapes now turns into matching between two scale space images using the geodesic distance measure suggested by Eberly [EBE94]. The matching is actually a point to point matching between the two scale space images, this is impractical if shapes in the database are complex, because complex shape can result in very high interval tree.

### 3.4.1.7 Spectral Transform

Spectral descriptors overcome the problem of noise sensitivity and boundary variations by analyzing shape in spectral domain. Spectral descriptors include Fourier descriptors (FD) and wavelet descriptors (WD), they are usually derived from spectral transform on 1-D shape signatures described in Section 3.4.1.2.

One of the most widely used shape description methods in literature is FD [BRI68, GRA72, ZAH72, RIC74, PER77, CHE84, MIT84, LINC87, LINS87, KRZ89, ARB89, ARB90, EIC90, OTT91, RAU94, KAU95, MEH97, RUI97, MAR00]. Different shape signatures are used in FD method. Complex coordinates and cumulative angle function are dominantly used in early works, however, centroid distance function is often used in recent works. Most early works are dedicated to character recognition and object classification. Conventional FD method only deal with closed curve and non-occluded curve, however, Lin et al. and Mitchell et al. proposed the description of partial shape using FD [LINC87, MIT84]. Arbter et al. introduced the affine-invariant FD to take into consideration the affined shape description [ARB89, ARB90]. Granlund introduced the Fourier invariants which describe rotation symmetry of the described shape [GRA72]. Rauber proposed a UNL FD which can describe disjointed or articulated contour shape [RAU94]. 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 is very high. Richard and Hemami introduced a complex distance measure, called true distance measure, for measuring the similarity between two set of FDs [RIC74]. Since the true distance measure requires two Fourier transform for each matching, it involves 15 times more computation than normal distance measure. Rui et al [RUI97] proposed a distance measure to classify similitude transformed characters using Fourier transformed coefficients. The distance measure 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 no different from FD derived from smoothed boundary. It should be noted that most early FD methods only deal with polygonal shape or polygonal approximated shape.

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 physic meaning; (iii) simple to normalize making shape matching a simple task; (iv) capturing 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.

Recently, several researchers have proposed the use of WD [TIE97, YAN98, OHM00] and short-time Fourier descriptors (SFD) [EIC90] for shape description. Although WD has the advantage over FD in that it is of multi-resolution in spatial space, the increase of spatial resolution sacrifices frequency resolution. For example, in [TIE97], only wavelet coefficients of 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 [YAN98], 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 [TIE97], the number of matching 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 with the feature segments on the shape boundary. Therefore, WD suffers the same drawback of primitive determination as that in the structural approach which will be discussed in Section 3.4.2.

3.4.1.8 Discussions

Global contour shape techniques take the whole shape contour as the shape representation and a description scheme is designed for this 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 accurate as possible; on the other hand, 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 can only be combined with other shape descriptors to create practical shape descriptors. Signature based matching are not suitable for online shape matching. Elastic matching and wavelet methods are
complex to implement and match. AR methods involve matrix operations which are expensive and it is difficult to associate AR descriptors with physical meaning. The implementation and matching of CSS are complex. However, the perceptual meaning and compact features of CSS are appealing for shape description and online retrieval. Fourier descriptor is simple to implement, it involves low 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. Boundary moment descriptor is similar to Fourier descriptor, it is easy to acquire. However, unlike Fourier descriptor, only a few lower order moment descriptors have physical interpretation.

3.4.2 Structural Methods

Another member in shape analysis family is structural shape representation. With 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 [PAV82]. The result is encoded into a string of the general form:

\[ S = s_1, s_2, \ldots, s_n \]  

(3.7)

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 attributed 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 \( S \).

3.4.2.1 Chain Code Representation

Chain codes describe an object by a sequence of unit-size line segments with a given orientation. The basis were introduced in 1961 by Freeman [FRE61] 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 a 4-directional or an 8-directional chain code. \( N \)-directional \((N>8\) and \(N=2^k\)) chain code is also possible, it is called general chain code [FRE78].

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.

It is not desirable to use chain code directly for shape description and matching, because it is very sensitive to boundary noise and variations. In addition, the dimension of chain code is high. Chain code is often used as an input to higher level analysis. For example, it can be used for polygon approximation and for finding boundary curvature.

Iivarinen and Visa derive a chain code histogram (CCH) for object recognition [IIV96]. 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 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.

3.4.2.2 Polygon Decomposition

In [GRO90] and [GRO92], 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 features is the editing distance of the two feature strings. For efficiency and robustness, only 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: feature-by-feature matching in the first step and model-by-model matching (shape-by-shape matching) in the second step. In the first step, given a data feature(s) of a query shape, the feature(s) 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 [MEH95], Mehrotra and Gary represent 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 \( S \) 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 \( S \) 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 \( i \)th interest point (Figure 3.4). Four transform parameters are also added to the coordinate set to create the final feature vector, the four parameters are \( S, T_x, T_y, \theta \), 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 organize into kdB-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 doesn’t guarantee 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.

![Figure 3.4. Chain vectors: (a) original boundary description; (b) normalized boundary description.](image)

All the above methods approximate shape as polygon, and the shape is represented as string of line segments which are then organized into tree data structure. While it is expected to work for man made objects, the application of it for general shapes is impractical.
3.4.2.3 Smooth Curve Decomposition

Berretti et al [BER00] extend the model of [GRO90] for general shape retrieval. In [BER00], the curvature zero-crossing points from Gaussian smoothed boundary are used to obtain primitives called tokens. 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 indicate the problem without solving 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, their similar tokens are found by traversing the index tree \( N \) times. The set of retrieved tokens having the same shape identifier forms 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 involves \( O(MN) \) operations (\( M \) and \( N \) are the numbers of tokens of two matching shapes respectively) operations. 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 painting 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, thereby, depends on the scale used in the smoothing stage.

3.4.2.4 Scale Space Method

Dudek and Tsotsos [DUD97] analyze 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 is extracted from each primitive. Then a string of segment descriptors is created to describe the shape. For two shapes \( A \) and \( B \) represented with their string descriptors \( A = (s_1^A, s_2^A, ..., s_N^A) \) and \( B = (s_1^B, s_2^B, ..., s_M^B) \), 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 scale. However, due to the inclusion of length in the segment descriptors, the descriptors are not scale invariant. Only small number (50) of shapes from plant
leaves are tested for the algorithm, no recognition or retrieval rate is reported. The three ad hoc or empirical parameters which are essential to the algorithm make the algorithm very application limited.

3.4.2.5 Syntactic Analysis

Syntactic analysis is inspired by the phenomenon that composition of a natural scene is 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 [FU74]. In syntactic methods, shape is represent with 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 Figure 3.5, the chromosome shape in the left of Figure 3.5 can be represented as a grammatical string of $S$:

$$S = d b a b c b a b d b a b c b a b$$  \hspace{1cm} (3.8)

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 in 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 established by Chomsky. It attempts to simulate the structural and hierarchical nature of human vision system. However, it is not practical in general application because it is impossible to infer a pattern 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.
3.4.2.6 Shape Invariants

Shape invariants can also be viewed as structural approach, because they also represent shape based on boundary primitives. Designers of shape invariants argue that although most of other shape representation techniques are invariant under similitude (rotation, translation and scaling) transform, they depend on viewpoint [SON93]. Therefore, techniques using invariants attempt to represent properties of the boundary configurations which remain unchanged under an appropriate class of transforms.

Generally, invariant theory is based on a collection of transforms that can be composed and inverted. In vision, the projective group of transforms is considered, the group 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 transform is generalized as a group action. Lie group theory is especially useful in designing new invariants [SQU00].

Invariant is usually named according to the number of features used to define it. An order one invariant is defined on a single feature, it is called an unary invariant; an order two invariant is defined between two features, it is called a binary invariant. Similarly, ternary invariant, quaternary invariant and so on. The total number of higher order invariants that exist under a certain group of transform is larger than that of lower order ones [LI99].

Common invariants include (i) geometric invariants such as cross-ratio, length ratio, distance ratio, angle, area [LI99], triangle [HUA98], invariants from coplanar points [SON93]; (ii) algebraic invariants such as determinant, eigenvalues [SQU00], trace [SON93]; (iii) differential invariants such as curvature, torsion and Gaussian curature. 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 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 numbers.

Shape representation using invariants has several problems. First, invariants are usually derived from pure geometric transform of shape. In reality, shape rarely changes according to strict geometric transform, especially shape 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 [LI92, LI99] which is known to be an NP-complete problem [SQU00]. The challenge is thus to find an algorithm that can obtain an acceptable solution in a reasonable time.

Recently, Kliot and Rivlin proposed the use of invariant signature for shape description [KLI98]. 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 curve. 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 the matrix matching. The retrieval is tested on a dataset composed of the SQUID [SQUID] fish database and three small data sets. Efficiency performance is given, however, no average effectiveness performance is shown.

Squire and Caelli also use invariant signature for shape description [SQU00]. The invariant signature is a probability density function derived from the boundary represented by piecewise algebraic curves. The signature is invariant under similitude transform, that is, rotation, translation and scaling transform. A histogram is then created from the quantized invariant 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 powerful than Fourier-Mellin descriptor, indicating that signature needs further processing for robust shape description.

Although the invariant 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 [DAV97], it is not clear whether shape matching using invariant signatures is better than shape matching using common shape signatures.
3.4.2.7 Discussions

The merit of structural approach is 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 structural approach is in 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 unknown. Therefore, the success of applying this method depends on the \textit{a priori} knowledge of the shape boundary features in the database. While structural methods are expected to work under particular assumption, it is impractical to use it 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, the retrieval fails.

- The second drawback of 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 measurement. Since these methods allow partial matching between shapes, sub-graph matching using optimal solution is inevitable. Although most of these methods avoid optimal solution issue, they do not guarantee best match.

- Structural approach fails to capture global shape features which are equally important for the shape representation. The equal importance of local and global shape features can be shown by the example of learning a city, only a few zoom-in views of the city does not mean you have learned the city. Zoom-out views is also essential in learning the city. Unless the city has a unique landmark which distinguishes it from other city, we cannot recognize the city from other cities from the map by just a few zoom-in views. As an example, Figure 3.6 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.

- Non-robustness. Because the structural representation does not preserve the topological structure of the object. Variations of object boundary can cause significant changes to local structures, in this case, global features are more reliable.
3.5 Region-based Shape Representation Techniques

In region based methods, all the pixels within a shape region are taken into account to obtain the shape representation. Common region based methods use moment descriptors to describe shape. Other region based methods include grid method, shape matrix, convex hull and media axis.

3.5.1 Global Methods

Global methods treat shape as a whole, the result representation is a numeric feature vector which can be used for shape description.

3.5.1.1 Geometric Moment Invariants

Historically, Hu published the first significant paper on the use of image moment invariants for two-dimensional pattern recognition applications [HU62]. His approach is based on the work of the 19th century mathematicians Boole, Cayley and Sylvester, and on the theory of algebraic forms:

\[
m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x,y)dx\,dy , \quad p, q = 0, 1, 2, \ldots
\]  

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. Geometric moment invariants has attracted wide attention [TEA80, TEH88, GON92, LIA96, SON93, JAH97] and has been used in may applications [DUD77, BEL91, PRO92, MOH95, MEH97, SAJ97].
The main problem with geometric moments is that only a few invariants derived from lower order of moments is not sufficient to accurately describe shape. Higher order invariants are difficult to derive.

3.5.1.2 Algebraic Moment Invariants

Algebraic moment invariants have been introduced by Taubin and Cooper [TAU91, TAU92], and has been used in QBIC [NIB93] and [SCA94]. The algebraic moment invariants are computed from the first \( m \) central moments and are given as the eigenvalues of predefined matrices, \( M_{ji}, k \), 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 [SCA94] show that the 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.

3.5.1.3 Orthogonal Moments and Other Moments

The algebraic moment transform of (3.9) can be extended to generalized form by replacing the conventional transform kernel \( x^py^q \) with a more general kernel of \( P_p(x)P_q(y) \). Teague [TEA80] uses this idea to introduce orthogonal moments—Legendre moments and Zernike moments—by replacing \( x^py^q \) in (3.9) with Legendre polynomial and Zernike polynomial respectively. Since both Legendre and Zernike polynomials are complete sets of orthogonal basis, Legendre and Zernike moments are called orthogonal moments. Other orthogonal moments are pseudo-Zernike moments which are obtained by using real-valued radial polynomials in Zernike polynomials as the moment transform kernel. Orthogonal moments allow for accurate reconstruction of the described shape. They make optimal utilization of shape information.

Teh and Chin [TEH88] have made a detailed study of orthogonal moments—Legendre moments, Zernike moments, psudo-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 images. 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 noise environment. The number of moments where the error reaches 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 SNR=30, the optimal number
of Zernike moments for shape description is 10. If SNR=200, there is little difference between
using 30 moments and using over 30 moments. They conclude that Zernike moments and pseudo-
Zernike moments are the preferred shape descriptors to others.

Liao and Pawlak [LIA96] extend Teh and Chin’s work by introducing techniques to increase
the accuracy and efficiency of moments. Specifically, they study moments accuracy under
different image resolution. 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 with many moment methods is that it is difficult to correlate high order moments
with shape physical features. Among the many moment shape descriptors, Zernike moments are
the most desirable for shape description. Because they are extracted from spectral domain, they
have similar properties of spectral features which are well understood. Shape description using
Zernike moments proves to be very promising [KIM98, KIM00]. Zernike moment descriptor has
been adopted by MPEG-7 as region-based shape descriptor [ISO00].

3.5.1.4 Grid Based Method

Grid shape descriptor is proposed by Lu and Sajjanhar [LU99] and has been used in
[CHA00] and [SAF00]. Basically, a grid of certain number 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
shape are assigned 1 and those not covered by shape are assigned 0. The shape can then be
represented as a binary feature vector. The binary Hamming distance or city block distance is
used to measure the similarity between two shapes. In order to accommodate translation, rotation
and scaling of the shape, the shape is first normalized before the scanning. The shape is scaled
into a fix sized rectangle, shifted to the upper left of the rectangle and rotated so that the major
axis of the shape are horizontal. Mirrored shape and flipped shape should be considered
separately. Chakrabarti et al [CHA00] improve grid descriptor by using adaptive resolution (AR)
representation and has used it in MARS [HUA96]. The AR grid descriptor is acquired by
applying quadtree decomposition on the bitmap representation of the shape.
The advantages of grid descriptor are its simplicity in representation, conforming to intuition, and also agreeing with shape coding method in MPEG-4. However, the computation is expensive.

### 3.5.1.5 Shape Matrix

Normal shape methods use 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 shape matrix which is derived from a circular raster sampling technique [GOS85]. The idea is similar to normal raster sampling. However, rather than overlaying 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 (Figure 3.7(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. The result matrix representation is invariant to translation, rotation, and scaling. The maximum radius of the shape is used for scale normalization. Since the sampling density is not constant with the polar sampling raster, Taza and Suen represent shape using weighted shape matrix which gives more weight to peripheral samples [TAZ89].

Since shape matrix is a sparse sampling of shape, it is easily affected by noise. Besides, shape matching using shape matrix is expensive. Perui et al. propose shape description based on the relative areas of the shape contained in concentric rings located in the shape center of the mass [PAR86, LON98]. Let $L$ be the maximum radius of the shape $S$ to be described, $C_k$ be the $k^{th}$ ring of $n$ concentric rings obtained by sectioning the maximum radius $L$ into $n$ equal segments. An area-ratio invariant is defined as

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

where $A(\cdot)$ is the area function. The shape descriptor is the feature vector of $x = [x_1, \ldots, x_n]^T$. Although the area ratio descriptor is more compact and robust than the shape matrix, it ignores the shape distribution within the measured ring. Consequently, the two shapes in Figure 3.7 (b) and (c) will be the same under this descriptor.
3.5.1.6 Discussions

Global region based methods treat 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. 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 and shape matrix methods are based on human intuition of perceiving shape, they do not reflect statistical distribution of shape region. They are usually more sensitive to noise and are not as concise as moment invariants.

3.5.2 Structural Methods

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

3.5.2.1 Convex Hull

A region $R$ is convex if and only if any two points $x_1, x_2 \in R$, the whole line segment $x_1x_2$ is inside the region. The convex hull of a region is the smallest convex region $H$ which satisfies the condition $R \subseteq H$. The difference $H-R$ is called the convex deficiency $D$ of the region $R$. The extracting of convex hull can use both boundary tracing method [SON93] and morphological methods [GON92, DAV97]. 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 reduces the computation of extracting convex hull from $O(n^2)$ to $O(n)$. The extracting 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. Figure 3.8(a) illustrates this process. The shape is then represented as a concavity tree (Figure 3.8(b)). Each concavity can be described by its area, bridge length (the line which connects the cut of the concavity), maximum curvature, distance from maximum curvature point to the bridge. The matching between shapes becomes a string matching or a graph matching.

![Figure 3.8](image)

Figure 3.8. (a) Convex hull and its concavities; (b) Concavity tree representation of convex hull [SON93].

### 3.5.2.2 Media Axis

Like the convex hull, 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 [DAV97]. For example, in the case of thick hand-drawn characters, the skeleton may be supposed to be the path traveled by the pen. The basic idea of the skeleton is that 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) [BLU67]. The medial axis is the locus of centers of maximal disks that fit within the shape as illustrated in Figure 3.9. The bold line in the figure is the skeleton of
the shaded rectangle 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 problem. The computation of 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. However, as has been pointed out by Pavlidis [PAV80], obtaining such polygonal approximation can be sufficient for shape description.

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 [MOR94].

![Figure 3.9. Construction of medial axis of a rectangle shape [MOR94].](image)

### 3.5.2.3 Discussions

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

### 3.6 Summary

In this chapter, existing shape techniques have been reviewed. Generally, there are two groups of approaches: contour-based versus region-based. Contour-based approaches are more popular than region-based approaches. Because it is believed that human being discriminates shapes mainly by their contour features. The majority of real world objects have clear contours which are readily available. Therefore, contour methods can easily find applications. Contour-based methods usually involve less computation than their region-based counterparts. However, contour shape descriptors are more easily affected by noise and variations than region-based shape descriptors because they use less shape information than region-based methods. Region-based methods are usually more robust and application independent. However, they usually
involves more computation and region shape descriptors usually needs more storage than contour-based descriptors.

Generally, every shape technique has its advantages and disadvantages. For image description and retrieval, MPEG-7 has set a set of principles to evaluate the suitability of shape technique: good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation. Based on these principles, structural approaches are not recommended because of their computation/implementation complexity and strong application dependence. Signature based matching and wavelet methods are not suitable for online retrieval because of their heavy online matching overhead. Comparatively, global feature-based methods are promising for shape based image retrieval. Particularly, in contour based methods, Fourier descriptor and curvature scale space descriptor are more promising for image retrieval; in region based methods, geometric moments, grid method and Zernike moments are more promising for image retrieval. Therefore, they are selected for our study in the next two chapters.