A Survey of Shape Feature Extraction Techniques

Mingqiang Yang, Kidiyo Kpalma, Joseph Ronsin

A survey of shape feature extraction techniques

Yang Mingqiang(1) (2), Kpalma Kidiyo(1), Ronsin Joseph(1)

July 15, 2008

(1) IETR-INSA, UMR-CNRS 6164, 35043 Rennes, France; (2) Shandong University, 250100, Jinan, China.

1 Introduction

"A picture is worth one thousand words". This proverb comes from Confucius – a Chinese philosopher before about 2500 years ago. Now, the essence of these words is universally understood. A picture can be magical in its ability to quickly communicate a complex story or a set of ideas that can be recalled by the viewer later in time.

Visual information plays an important role in our society, it will play an increasingly pervasive role in our lives, and there will be a growing need to have these sources processed further. The pictures or images are used in many application areas like architectural and engineering design, fashion, journalism, advertising, entertainment, etc. Thus it provides the necessary opportunity for us to use the abundance of images. However, the knowledge will be useless if one can't find it. In the face of the substantive and increasing apace images, how to search and to retrieve the images that we interested with facility is a fatal problem: it brings a necessity for image retrieval systems. As we know, visual features of the images provide a description of their content. Content-based image retrieval (CBIR), emerged as a promising mean for retrieving images and browsing large images databases. CBIR has been a topic of intensive research in recent years. It is the process of retrieving images from a collection based on automatically extracted features.

This paper focuses on presenting the existing approaches of shape-based feature extraction. Efficient shape features must present some essential properties such as:

- identifiability: shapes which are found perceptually similar by human have the same feature different from the others.
- translation, rotation and scale invariance: the location, rotation and scaling changing of the shape must not affect the extracted features.
- affine invariance: the affine transform performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelism" of lines. Affine transform can be constructed using sequences of translations, scales, flips, rotations and shears. The extracted features must be as invariant as possible with affine transforms.
- noise resistance: features must be as robust as possible against noise, i.e., they must be the same whichever be the strength of the noise in a give range that affects the pattern.
- occultation invariance: when some parts of a shape are occulted by other objects, the feature of the remaining part must not change compared to the original shape.
- statistically independent: two features must be statistically independent. This represents compactness of the representation.
- reliable: as long as one deals with the same pattern, the extracted features must remain the same.

In general, shape descriptor is some set of numbers that are produced to describe a given shape feature. A descriptor attempts to quantify shape in ways that agree with human intuition (or task-specific requirements). Good retrieval accuracy requires a shape descriptor to be able to effectively find perceptually similar shapes from a database. Usually, the descriptors are in the form of a vector. Shape descriptors should meet the following requirements:

- the descriptors should be as complete as possible to represent the content of the information items.
- the descriptors should be represented and stored compactly. The size of descriptor vector must not be too large.
- the computation of distance between descriptors should be simple; otherwise the execution time would be too long.

Shape feature extraction and representation plays an important role in the following categories of applications:

- shape retrieval: searching for all shapes in a typically large database of shapes that are similar to a query shape. Usually all shapes within a given distance from the query are determined or the first few shapes that have the smallest distance.

- shape recognition and classification: determining whether a given shape matches a model sufficiently, or which of representative class is the most similar.

- shape alignment and registration: transforming or translating one shape so that it best matches another shape, in whole or in part.

- shape approximation and simplification: constructing a shape of fewer elements (points, segments, triangles, etc.), that is still similar to the original.

Many shape description and similarity measurement techniques have been developed in the past. A number of new techniques have been proposed in recent years. There are 3 main different classification methods as follows:

- Contour-based methods and region-based methods [1]. This is the most common and general classification and it is proposed by MPEG-7. It is based on the use of shape boundary points as opposed to shape interior points. Under each class, 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).

- Space domain and transform domain. Methods in space domain match shapes on point (or point feature) basis, while feature domain techniques match shapes on feature (vector) basis.

- Information preserving (IP) and non-information preserving (NIP). IP methods allow an accurate reconstruction of a shape from its descriptor, while NIP methods are only capable of partial ambiguous reconstruction. For object recognition purpose, IP is not a requirement.

Unlike the traditional classification, the approaches of shape-based feature extraction and representation are classified according to their processing approaches. The figure 1 shows the hierarchy of the classification of shape feature extraction approaches.

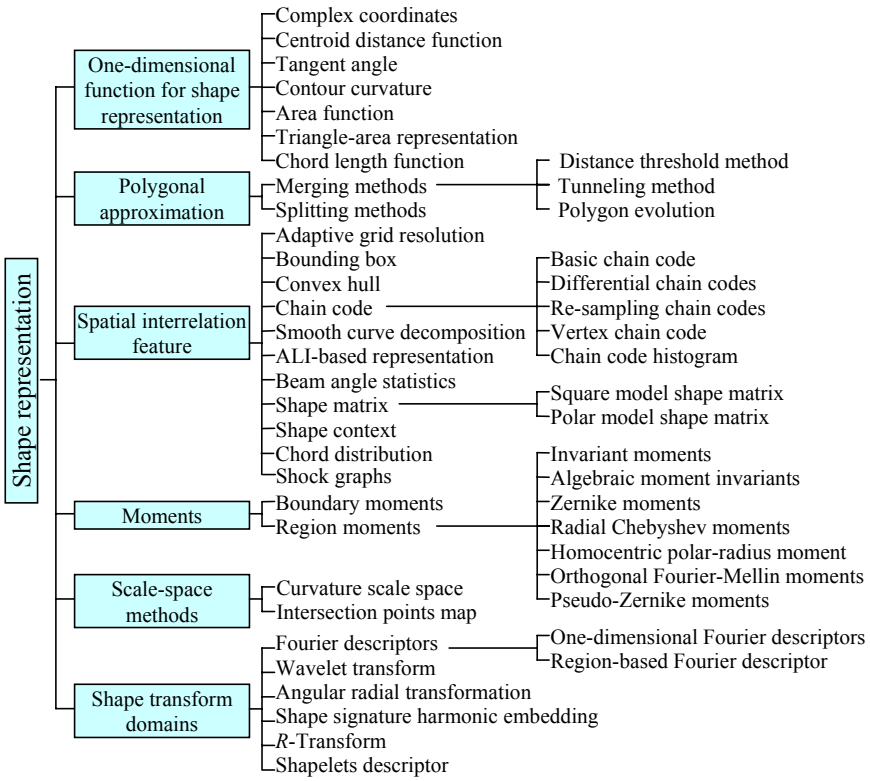


Figure 1: An overview of shape description techniques

Without being complete, in the following sections, we will describe and group a number of these methods together.

2 Shape parameters

Basically, shape-based image retrieval consists of the measuring of similarity between shapes represented by their features. Some simple geometric features can be used to describe shapes. Usually, the simple geometric features 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 stand alone shape descriptors. A shape can be described by different aspects. These shape parameters are Center of gravity, Axis of least inertia, Digital bending energy, Eccentricity, Circularity ratio, Elliptic variance, Rectangularity, Convexity, Solidity, Euler number, Profiles, Hole area ratio. They will be introduced in this section.

2.1 Center of gravity

The center of gravity is also called centroid. Its position should be fixed in relation to the shape.

If a shape is represented by its region function Eq. **??**, its centroid (g_x, g_y) is:

$$\begin{cases} g_x = & \frac{1}{N}\sum_{i=1}^{N} x_i \\ g_y = & \frac{1}{N}\sum_{i=1}^{N} y_i \end{cases} \tag{1}$$

where N is the number of point in the shape, $(x_i, y_i) \in \{(x_i, y_i) \mid f(x_i, y_i) = 1\}$.

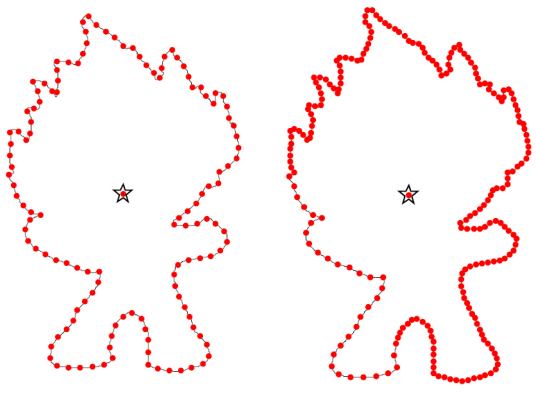
If a shape is represented by its contour Eq. **??**, the position of its centroid is given below:

$$\begin{cases} g_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ g_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \end{cases} \tag{2}$$

where A is the contour's area and

$$A = \frac{1}{2}\left|\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)\right| \tag{3}$$

The position of shape centroid is fixed with different points distribution on a contour. One can notice that the position of the centroid in Figure 2 is fixed no matter how the points distribution is.



(a) (b)

The dots are points distributed on contour by uniformly (a) and non-uniformly (b). The star is the centroid of original contour; the inner dot is the centroid of sampled contour.

Figure 2: Centroid of contour

So using Eq. 2, we can obtain the genuine centroid of a contour under whatever the contour is normalized.

2.2 Axis of least inertia

The axis of least inertia is unique to the shape. It serves as a unique reference line to preserve the orientation of the shape. The axis of least inertia (ALI) of a shape is defined as the line for which the integral of the square of the distances to points on the shape boundary is a minimum.

Since the axis of inertia pass through the centroid of a contour, to find the ALI, transfer the shape and let the centroid of the shape be the origin of Cartesian coordinate system. Let $x\sin\theta - y\cos\theta = 0$ be the parametric equation of ALI. The slope angle θ is estimated as follows:

Let α be the angle between the axis of least inertia and the x-axis. The inertia is given by [2, 3]:

$$I = \frac{1}{2}(a + c) - \frac{1}{2}(a - c)\cos(2\alpha) - \frac{1}{2}bsin(2\alpha)$$

where $a = \sum_{i=0}^{N-1} x_i^2$, $b = 2\sum_{i=0}^{N-1} x_i y_i$, $c = \sum_{i=0}^{N-1} y_i^2$.

Hence,

$$\frac{dI}{d\alpha} = (a-c)\sin(2\alpha) - b\cos(2\alpha)$$

$$\frac{d^2 I}{d\alpha^2} = 2(a-c)\cos(2\alpha) + 2b\sin(2\alpha)$$

Let $dI/d\alpha = 0$, we obtain

$$\alpha = \frac{1}{2}\arctan\left(\frac{b}{a-c}\right),\ -\frac{\pi}{2} < \alpha < \frac{\pi}{2}$$

The slope angle θ is given by

$$\theta = \begin{cases} \alpha + \frac{\pi}{2} & if\ \frac{d^2 I}{d\alpha^2} < 0 \\ \alpha & otherwise \end{cases}$$

2.3 Average bending energy

Average bending energy BE is defined by

$$BE = \frac{1}{N}\sum_{s=0}^{N-1} K(s)^2$$

where $K(s)$ is the curvature function, s is the arc length parameter, and N is the number of points on a contour [4]. In order to compute the average bending energy more efficiently, Young et. al. [5] did the Fourier transform of the boundary and used Fourier coefficients and Parseval's relation.

One can prove that the circle is the shape having the minimum average bending energy.

2.4 Eccentricity

Eccentricity is the measure of aspect ratio. It is the ratio of the length of major axis to the length of minor axis. It can be calculated by principal axes method or minimum bounding rectangle method.

2.4.1 Principal axes method

Principal axes of a given shape can be uniquely defined as the two segments of lines that cross each other orthogonally in the centroid of the shape and represent the directions with zero cross-correlation [6]. This way, a contour is seen as an instance from a statistical distribution. Let us consider the covariance matrix C of a contour:

$$C = \frac{1}{N}\sum_{i=0}^{N-1} \begin{pmatrix} x_i - g_x \\ y_i - g_y \end{pmatrix} \begin{pmatrix} x_i - g_x \\ y_i - g_y \end{pmatrix}^T = \begin{pmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{pmatrix} \tag{4}$$

where

$c_{xx} = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - g_x)^2$

$c_{xy} = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - g_x)(y_i - g_y)$

$c_{yx} = \frac{1}{N}\sum_{i=0}^{N-1}(y_i - g_y)(x_i - g_x)$

$c_{yy} = \frac{1}{N}\sum_{i=0}^{N-1}(y_i - g_y)^2$

$G(g_x, g_y)$ is the centroid of the shape. Clearly, here $c_{xy} = c_{yx}$.

The lengths of the two principal axes equal the eigenvalues λ_1 and λ_2 of the covariance matrix C of a contour, respectively.

So the eigenvalues λ_1 and λ_2 can be calculated by

$$det(C - \lambda_{1,2} I) = det\begin{pmatrix} c_{xx} - \lambda_{1,2} & c_{xy} \\ c_{yx} & c_{yy} - \lambda_{1,2} \end{pmatrix} = (c_{xx} - \lambda_{1,2})(c_{yy} - \lambda_{1,2}) - c_{xy}^2 = 0$$

So

$$\begin{cases} \lambda_1 = \frac{1}{2}\left[c_{xx} + c_{yy} + \sqrt{(c_{xx} + c_{yy})^2 - 4\left(c_{xx}c_{yy} - c_{xy}^2\right)}\right] \\ \lambda_2 = \frac{1}{2}\left[c_{xx} + c_{yy} - \sqrt{(c_{xx} + c_{yy})^2 - 4\left(c_{xx}c_{yy} - c_{xy}^2\right)}\right] \end{cases}$$

Then, eccentricity can be calculated:

$$E = \lambda_2/\lambda_1 \tag{5}$$

2.4.2 Minimum bounding rectangle

Minimum bounding rectangle is also called minimum bounding box. It is the smallest rectangle that contains every point in the shape. For an arbitrary shape, eccentricity is the ratio of the length L and width W of minimal bounding rectangle of the shape at some set of orientations. Elongation, *Elo*, is an other concept based on eccentricity (cf. Figure 3):

$$Elo = 1 - W/L \tag{6}$$

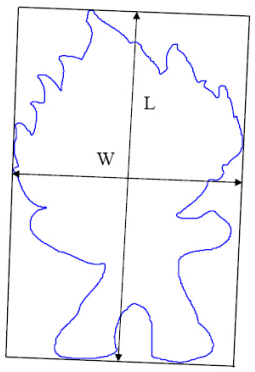


Figure 3: Minimum bounding rectangle and corresponding parameters for elongation

Elongation is a measure that takes values in the range $[0, 1]$. A symmetrical shape in all axes such as a circle or square will have an elongation value of 0 whereas shapes with large aspect ratios will have an elongation closer to 1.

2.5 Circularity ratio

Circularity ratio represents how a shape is similar to a circle [7]. There are 3 definitions:

- Circularity ratio is the ratio of the area of a shape to the area of a circle having the same perimeter:

$$C_1 = \frac{A_s}{A_c} \tag{7}$$

  where A_s is the area of the shape and A_c is the area of the circle having the same perimeter as the shape. Assume the perimeter is $\mathcal{O}$, so $A_c = \mathcal{O}^2/4\pi$. Then $C_1 = 4\pi \cdot A_s/\mathcal{O}^2$. As 4π is a constant, so we have the second circularity ratio definition.

- Circularity ratio is the ratio of the area of a shape to the shape's perimeter square:

$$C_2 = \frac{A_s}{\mathcal{O}^2} \tag{8}$$

- Circularity ratio is also called circle variance, and defined as:

$$C_{va} = \frac{\sigma_R}{\mu_R} \tag{9}$$

where μ_R and σ_R are the mean and standard deviation of the radial distance from the centroid (g_x, g_y) of the shape to the boundary points $(x_i, y_i), i \in [0, N-1]$. They are the following formulae respectively:

$$\mu_R = \frac{1}{N}\sum_{i=1}^{N-1} d_i \quad and \quad \sigma_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1}(d_i - \mu_R)^2}$$

where $d_i = \sqrt{(x_i - g_x)^2 + (y_i - g_y)^2}$.

The most compact shape is a circle. See Figure 4.

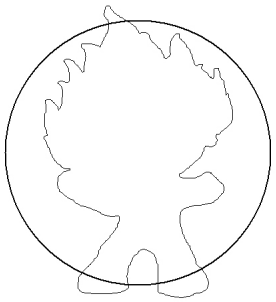


Figure 4: Circle variance

2.6 Ellipse variance

Ellipse variance E_{va} is a mapping error of a shape to fit an ellipse that has an equal covariance matrix as the shape: $C_{ellipse} = C$ (cf. Eq.4). It is practically effective to apply the inverse approach yielding.

We assume

$$V_i = \begin{pmatrix} x_i - g_x \\ y_i - g_y \end{pmatrix}$$

$$d'_i = \sqrt{V_i^T \cdot C_{ellipse}^{-1} \cdot V_i}$$

$$\mu'_R = \frac{1}{N}\sum_{i=1}^{N-1} d'_i \quad and \quad \sigma'_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1}(d'_i - \mu'_R)^2}$$

Then

$$E_{va} = \frac{\sigma'_R}{\mu'_R} \tag{10}$$

Comparing with Eq. 9, intuitively, E_{va} represents a shape more accurately than C_{va}, cf. Figure 5.

Figure 5: Ellipse variance

2.7 Rectangularity

Rectangularity represents how rectangular a shape is, i.e. how much it fills its minimum bounding rectangle:

$$Rectangularity = A_S/A_R$$

where A_S is the area of a shape; A_R is the area of the minimum bounding rectangle.

2.8 Convexity

Convexity is defined as the ratio of perimeters of the convex hull $\mathcal{O}_{Convexhull}$ over that of the original contour $\mathcal{O}$ [6]:

$$Convexity = \frac{\mathcal{O}_{Convexhull}}{\mathcal{O}} \tag{11}$$

Figure 6: Illustration of convex hull

The region R^2 is a convex if and only if for any two points $P_1, P_2 \in R^2$, the whole line segment P_1P_2 is inside the region. The convex hull of a region is the smallest convex region including it. In Figure 6, the outline is the convex hull of the region.

In [6], the authors presented the algorithm for constructing a convex hull by traversing the contour and minimizing turn angle in each step.

2.9 Solidity

Solidity describes the extent to which the shape is convex or concave [8] and it is defined by

$$Solidity = A_s/H$$

where, A_s is the area of the shape region and H is the convex hull area of the shape. The solidity of a convex shape is always 1.

2.10 Euler number

Euler number describes the relation between the number of contiguous parts and the number of holes on a shape. Let S be the number of contiguous parts and N be the number of holes on a shape. Then the Euler number is:

$$Eul = S - N$$

For example

Euler Number equal to *1*, *-1* and *0*, respectively.

2.11 Profiles

The profiles are the projection of the shape to x-axis and y-axis on Cartesian coordinate system. We obtain two one-dimension functions:

$$Pro_x(i) = \sum_{j=j_{min}}^{j_{max}} f(i,j) \quad and \quad Pro_y(j) = \sum_{i=i_{min}}^{i_{max}} f(i,j)$$

where $f(i, j)$ represents the region of shape Eq. **??**. See Figure 7.

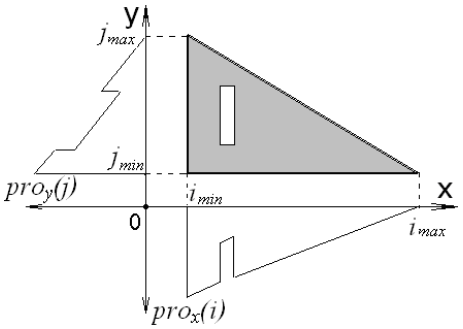


Figure 7: Profiles

2.12 Hole area ratio

Hole area ratio HAR is defined as

$$HAR = \frac{A_h}{A_s}$$

where A_s is the area of a shape and A_h is the total area of all holes in the shape. Hole area ratio is most effective in discriminating between symbols that have big holes and symbols with small holes [9].

3 One-dimensional function for shape representation

The one-dimensional function which is derived from shape boundary coordinates is also often called shape signature [10, 11]. The shape signature usually captures the perceptual feature of the shape [12]. Complex coordinates, Centroid distance function, Tangent angle (Turning angles), Curvature function, Area function, Triangle-area representation and Chord length function are the commonly used shape signatures.

Shape signature can describe a shape all alone; it is also often used as a preprocessing to other feature extraction algorithms, for example, Fourier descriptors, wavelet description. In this section, the shape signatures are introduced.

3.1 Complex coordinates

A complex coordinates function is simply the complex number generated from the coordinates of boundary points, $P_n(x(n), y(n))$, $n \in [1, N]$:

$$z(n) = [x(n) - g_x] + i[y(n) - g_y]$$

where (g_x, g_y) is the centroid of the shape, given by Eq. 2.

3.2 Centroid distance function

The centroid distance function is expressed by the distance of the boundary points from the centroid (g_x, g_y) (Eq. 2) of a shape

$$r(n) = [(x(n) - g_x)^2 + (y(n) - g_y)^2]^{1/2}$$

Due to the subtraction of centroid, which represents the position of the shape, from boundary coordinates, both complex coordinates and centroid distance representation are invariant to translation.

3.3 Tangent angle

The tangent angle function at a point $P_n(x(n), y(n))$ is defined by a tangential direction of a contour [13]:

$$\theta(n) = \theta_n = \arctan \frac{y(n) - y(n-w)}{x(n) - x(n-w)}$$

since every contour is a digital curve; w is a small window to calculate $\theta(n)$ more accurately.

Tangent angle function has two problems. One is noise sensitivity. To decrease the effect of noise, a contour is filtered by a low-pass filter with appropriate bandwidth before calculating the tangent angle function. The other is discontinuity, due to the fact that the tangent angle function assumes values in a range of length 2π, usually in the interval of $[-\pi, \pi]$ or $[0, 2\pi]$. Therefore θ_n in general contains discontinuities of size 2π. To overcome the discontinuity problem, with an

arbitrary starting point, the cumulative angular function φ_n is defined as the angle differences between the tangent at any point P_n along the curve and the tangent at the starting point P_0 [14, 15]:

$$\varphi(n) = [\theta(n) - \theta(0)]$$

In order to be in accordance with human intuition that a circle is "shapeless", assume $t = 2\pi n/N$, then $\varphi(n) = \varphi(tN/2\pi)$. A periodic function is termed as the cumulative angular deviant function $\psi(t)$ and is defined as

$$\psi(t) = \varphi(\frac{N}{2\pi}t) - t \quad t \in [0, 2\pi]$$

N is the total number of contour points.

In [16], the authors proposed a method based on tangent angle. It is called tangent space representation. A digital curve C simplified by polygon evolution is represented in the tangent space by the graph of a step function, where the x-axis represents the arc length coordinates of points in C and the y-axis represents the direction of the line segments in the decomposition of C. For example, figure 8 shows a digital curve and its step function representation in the tangent space.

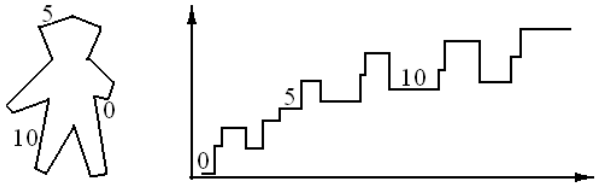


Figure 8: Digital curve and its step function representation in the tangent space

3.4 Contour curvature

Curvature is a very important boundary feature for human to judge similarity between shapes. It also has salient perceptual characteristics and has proven to be very useful for shape recognition [17]. In order to use *K(n)* for shape representation, we quote the function of curvature, *K(n)*, from [18, 19] as:

$$K(n) = \frac{\dot{x}(n)\ddot{y}(n) - \dot{y}(n)\ddot{x}(n)}{(\dot{x}(n)^2 + \dot{y}(n)^2)^{3/2}} \tag{12}$$

Therefore, it is possible to compute the curvature of a planar curve from its parametric representation. If n is the normalized arc-length parameter s, then Eq. 12 can be written as:

$$K(s) = \dot{x}(s)\ddot{y}(s) - \dot{y}(s)\ddot{x}(s) \tag{13}$$

As given in Eq. 13, the curvature function is computed only from parametric derivatives, and, therefore, it is invariant under rotations and translations. However, the curvature measure is scale dependent, i.e., inversely proportional to the scale. A possible way to achieve scale independence is to normalize this measure by the mean absolute curvature, i.e.,

$$K'(s) = \frac{K(s)}{\frac{1}{N}\sum_{s=1}^{N}|K(s)|}$$

where N is the number of points on the normalized contour.

When the size of the curve is an important discriminative feature, the curvature should be used without the normalization; otherwise, for the purpose of scale-invariant shape analysis, the normalization should be performed.

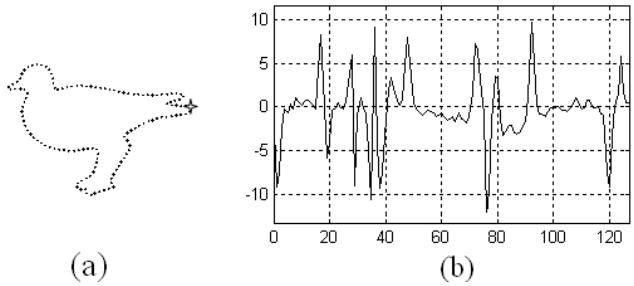
An approximate arc-length parametrization based on the centripetal method is given by the following [19]:

Let $P = \sum_{n=1}^{N} d_n$ be the perimeter of the curve and $L = \sum_{n=1}^{N} \sqrt{d_n}$, where d_n is the length of the chord between points p_n and p_{n+1}, *n=1, 2, ..., N-1.* The approximate arc-length parametrization relations:

$s_1 = 0$;

$s_k = s_{k-1} + \frac{P\sqrt{d_{k-1}}}{L}, k = 2, 3, \ldots, N.$

Starting from an arbitrary point and following the contour clockwise, we compute the curvature at each interpolated point using Eq. 13. Convex and concave vertices will imply negative and positive values, respectively (the opposite is verified for counterclockwise sense). Figure 9 is an example of curvature function. Clearly, as a descriptor, the curvature function can distinguish different shapes.

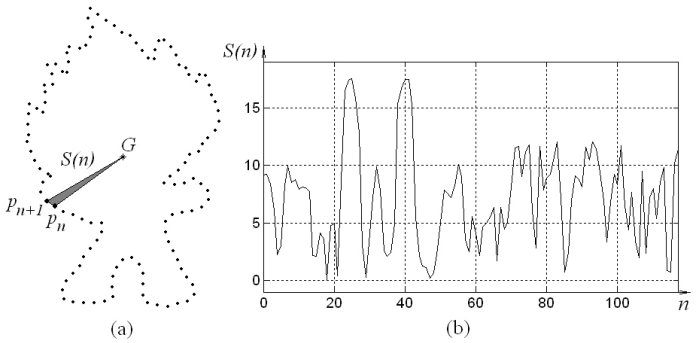


(a) Contours normalized to 128 points; the dots marked star are the starting points on the contours; (b) curvature functions; the curvature is computed clockwise.

Figure 9: Curvature function

3.5 Area function

When the boundary points change along the shape boundary, the area of the triangle formed by two successive boundary points and the center of gravity also changes. This forms an area function which can be exploited as shape representation. Figure 10 shows an example. Let $S(n)$ be the area between the successive boundary points P_n, P_{n+1} and center of gravity G.



(a) Original contour; (b) the area function of (a).

Figure 10: Area function

The area function is linear under affine transform. However, this linearity only works for shape sampled at its same vertices.

3.6 Triangle-area representation

The triangle-area representation (TAR) signature is computed from the area of the triangles formed by the points on the shape boundary [20, 21]. The curvature of the contour point (x_n, y_n) is measured using the TAR as follows.

For each three consecutive points $P_{n-t_s}(x_{n-t_s}, y_{n-t_s})$, $P_n(x_n, y_n)$, and $P_{n+t_s}(x_{n+t_s}, y_{n+t_s})$, where $n \in [1, N]$ and $t_s \in [1, N/2-1]$, N is even. The signed area of the triangle formed by these points is given by:

$$TAR(n, t_s) = \frac{1}{2} \begin{vmatrix} x_{n-t_s} & y_{n-t_s} & 1 \\ x_n & y_n & 1 \\ x_{n+t_s} & y_{n+t_s} & 1 \end{vmatrix} \tag{14}$$

When the contour is traversed in counter clockwise direction, positive, negative and zero values of TAR mean convex, concave and straight-line points, respectively. Figure 11 demonstrates these three types of the triangle areas and the complete TAR signature for the hammer shape.

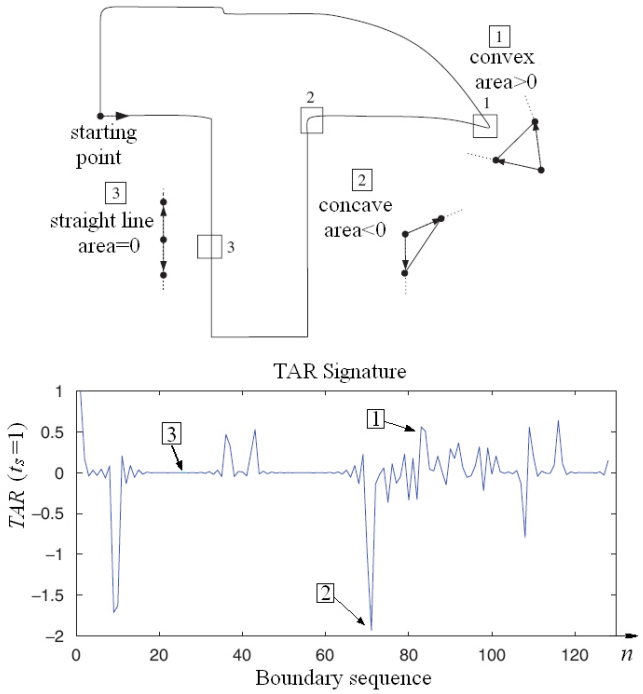
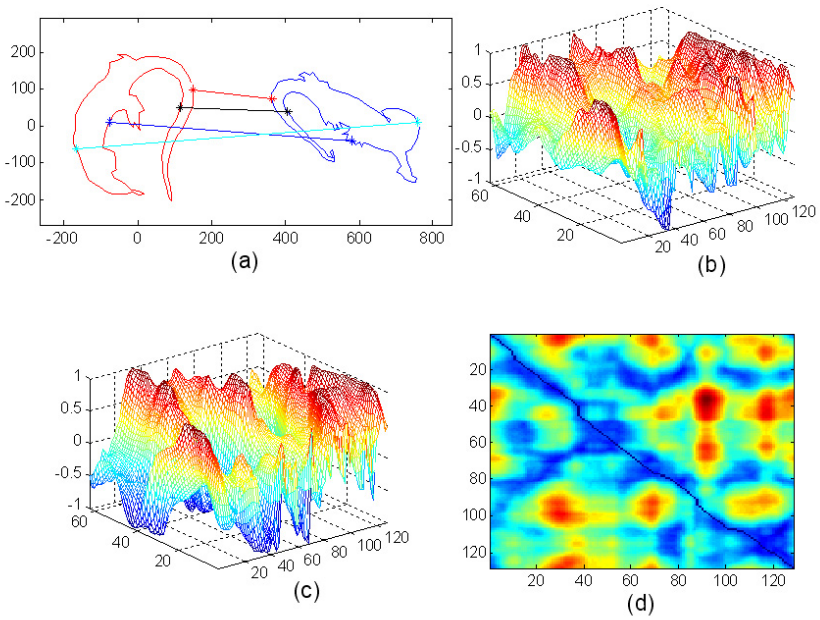


Figure 11: Three different types of the triangle-area values and the TAR signature for the hammer shape

By increasing the length of the triangle sides, i.e., considering farther points, the function of Eq. 14 will represent longer variations along the contour. The TARs with different triangle sides can be regarded as different scale space functions. The total TARs, $t_s \in [1,\ N/2-1]$, compose a multi-scale space TAR.

Figure 12 shows the multi-scale space TAR function and its dynamic space warping (DSW) matching (cf. Subsection 1.3.2). In (a), the correspondent points on the model contour to these on the query contour are consistent with human perception after DSW matching.



(a) The query contour (left) and the model contour (right); (b) the multi-scale space TAR function of the model contour; (c) the multi-scale space TAR function of the query contour; (d) dynamic space warping (DSW) matching table of the multi-scale space TAR functions (b) and (c).

Figure 12: Dynamic space warping (DSW) matching

In [21], authors show that the multi-scale space TAR is relatively invariant to the affine transform and robust to non-rigid transform. The computation complexity of the TAR stage is $O(N^2)$.

3.7 Chord length function

The chord length function is derived from shape boundary without using any reference point. For each boundary point p, its chord length function is the shortest distance between p and another boundary point p' such that line pp' is perpendicular to the tangent vector at p [10].

The chord length function is invariant to translation and it overcomes the biased reference point (which means the centroid is often biased by boundary noise or defections) problems. However, it is very sensitive to noise, there may be drastic burst in the signature of even smoothed shape boundary.

3.8 Discussions

A shape signature represents a shape by a 1-D function derived from shape contour. To obtain the translation invariant property, they are usually defined by relative values. To obtain the scale invariant property, normalization is necessary. In order to compensate for orientation changes, shift matching is needed to find the best matching between two shapes. Having regard to occultation, Tangent angle, Contour curvature and Triangle-area representation have invariance property. In addition, shape signatures are computationally simple.

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.

4 Polygonal approximation

Polygonal approximation can be set to ignore the minor variations along the edge, and instead capture the overall shape. This is useful because it reduces the effects of discrete pixelization of the contour. In general, there are two methods to realize it. One is merging, the other is splitting [22].

4.1 Merging methods

Merging methods add successive pixels to a line segment if each new pixel that is added doesn't cause the segment to deviate too much from a straight line.

4.1.1 Distance threshold method

Choose one point as a starting point, on the contour, for each new point that we add, let a line go from the starting point to this new point. Then, we compute the squared error for every point along the segment/line. If the error exceeds some threshold, we keep the line from the start point to the previous point and start a new line.

In practice, the most of practical error measures in use are based on distance between vertices of the input curve and the approximation linear segments. The distance $d_k(i,j)$ from curve vertex $P_k = (x_k, y_k)$ to the corresponding approximation linear segments (P_i, P_j) is defined as follows (cf. Figure 13):

$$d_k(i,j) = \frac{|(x_j - x_i)(y_i - y_k) - (x_i - x_k)(y_j - y_i)|}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}$$

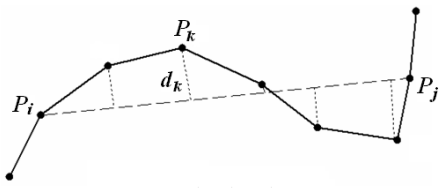


Figure 13: Illustration of the distance from a point on the boundary to a linear segment

4.1.2 Tunneling method

If we have thick boundaries rather than single-pixel thick ones, we can still use a similar approach called tunneling. Imagine that we're trying to lay straight rods along a curved tunnel, and that we want to use as few as possible. We can start at one point and lay as long a straight rod as possible. Eventually, the curvature of the "tunnel" won't let us go any further, so we lay another rod and another until we reach the end.

Both the distance threshold and tunneling methods can do polygonal approximation efficiently. However, the great disadvantage is that the position of starting point will affect greatly the approximate polygon.

4.1.3 Polygon evolution

The basic idea of polygons evolution in [23] is very simple: in every evolution step, a pair of consecutive line segments (the line segment is the line between two consecutive vertices) s_1,s_2 is substituted with a single line segment joining the endpoints of s_1 and s_2.

The key property of this evolution is the order of the substitution. The substitution is done according to a relevance measure K given by

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)},$$

where $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments s_1, s_2 and $l(\alpha)$ is the length of α, $\alpha = s_1 \, or \, s_2$, normalized with respect to the total length of a polygonal curve. The evolution algorithm is assuming that vertices which are surrounded by segments with a high value of $K(s_1, s_2)$ are important while those with a low value are not. Figure 14 is an example.

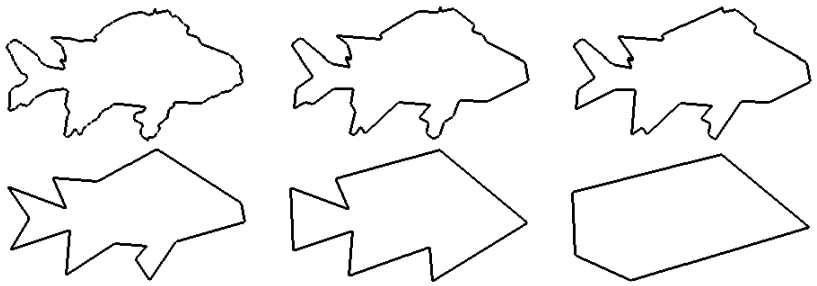


Figure 14: A few stages of polygon evolution according to a relevant measure

The curve evolution method achieves the task of shape simplification, i.e., the process of evolution compares the significance of vertices of the contour based on a relevance measure. Since any digital curve can be regarded as a polygon without loss of information (with possibly a large number of vertices), it is sufficient to study evolutions of polygonal shapes for shape feature extraction.

4.2 Splitting methods

Splitting methods work by first drawing a line from one point on the boundary to another. Then, we compute the perpendicular distance from each point along the boundary segment to the line. If this exceeds some threshold, we break the line at the point of greatest distance. We then repeat the process recursively for each of the two new lines until we don't need to break any more. See Figure 15 for an example.

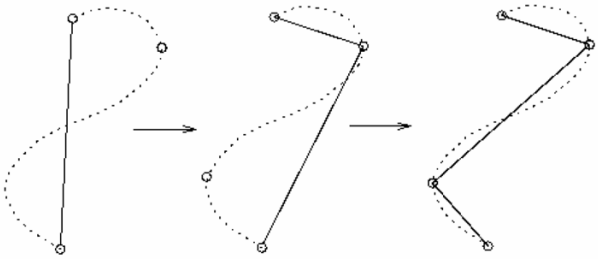


Figure 15: Splitting methods for polygonal approximation

This is sometimes known as the "fit and split" algorithm. For a closed contour, we can find the two points that lie farthest apart and fit two lines between them, one for one side and one for the other. Then, we can apply the recursive splitting procedure to each side.

4.3 Discussions

Polygonal approximation technique can be used as a simple method for contour representation and description. The polygon approximation have some interesting properties:

- it leads to simplification of shape complexity with no blurring effects.
- it leads to noise elimination.
- although irrelevant features vanish after polygonal approximation, there is no dislocation of relevant features.
- the remaining vertices on a contour do not change their positions after polygonal approximation.

Polygonal approximation technique can also be used as preprocessing method for further extracting features from a shape.

5 Spatial interrelation feature

Spatial interrelation feature describes the region or the contour of shape by the relation of their pixels or curves. In general, the representation is done by using their geometric features: length, curvature, relative orientation and location, area, distance and so on.

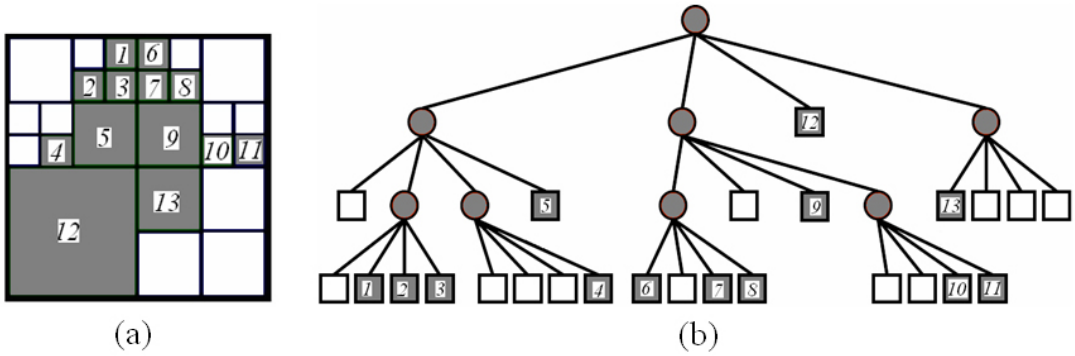
5.1 Adaptive grid resolution

The adaptive grid resolution (AGR) was proposed by [24]. In the AGR, a square grid that is just big enough to cover the entire shape is overlaid on a shape. A resolution of the grid cells varies from one portion to another according to the content of the portion of the shape. On the borders or the detail portion on the shape, the higher resolution, i.e. the smaller grid cells, are applied; on the other hand, in the coarse regions of the shape, lower resolution, i.e. the bigger grid cells, are applied.

To guarantee rotation invariance, it needs to convert an arbitrarily oriented shape into a unique common orientation. First, find the major axis of the shape. The major axis is the straight line segment joining the two points P_1 and P_2 on the boundary farthest away from each other. Then we rotate the shape so that its major axis is parallel to the x-axis. This orientation is still not unique as there are two possibilities: P_1 can be on the left or on the right. This problem is solved by computing the centroid of the polygon and making sure that the centroid is below the major axis, thus guaranteeing a unique orientation.

Let us now consider scale and translation invariance. We define the bounding rectangle (BR) of a shape as the rectangle with sides parallel to the x and y axes just large enough to cover the entire shape (after rotation). Note that the width of the BR is equal to the length of the major axis. To achieve scale invariance, we proportionally scale all shapes so that their BRs have the same fixed width (pixels).

The method of computation of the AGR representation of a shape applies quad-tree decomposition on the bitmap representation of the shape. The decomposition is based on successive subdivision of the bitmap into four equal-size quadrants. If a bitmap-quadrant does not consist entirely of part of shape, it is recursively subdivided into smaller and smaller quadrants until we reach bitmap-quadrants, i.e., termination condition of the recursion is that the resolution reaches that one predefined. Figure 16(a) is an example of AGR.



(a) Adaptive Grid Resolution (AGR) image; (b) quad-tree decomposition of AGR.

Figure 16: Adaptive resolution representations

To represent the AGR image, in [24], quad-tree method is applied. Each node in the quad-tree covers a square region of the bitmap. The level of the node in the quad-tree determines the size of the square. The internal nodes (shown by gray circles) represent "partially covered" regions; the leaf nodes shown by white boxes represent regions with all 0*s* while

the leaf nodes shown by black boxes represent regions with all $1s$. The "all $1s$" regions are used to represent the shape, Figure 16(b). Each rectangle can be described by 3 numbers: its center $C = (C_x, C_y)$ and its size (i.e. side length) S. So each shape can be mapped to a point in $3n$-dimensional space (n is the number of the rectangles occupied by the shape region).

Due to the fact that the normalization before computing AGR, AGR representation is invariant under rotation, scaling and translation. It is also computationally simple.

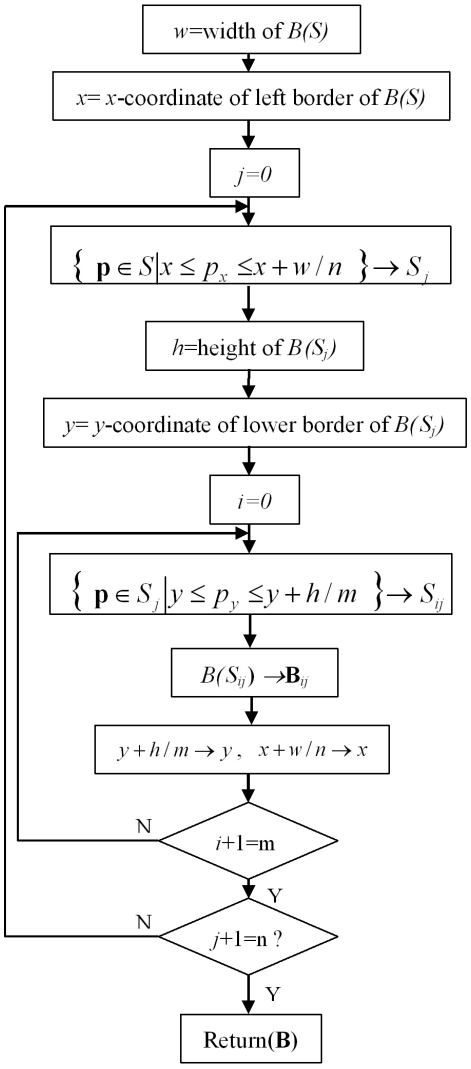


Figure 17: Flowchart of shape divided by bounding box

5.2 Bounding box

Bounding box computes homeomorphisms between 2D lattices and its shapes. Unlike many other methods, this mapping is not restricted to simply connected shapes but applies to arbitrary topologies [25].

To make bounding box representation invariant to rotation, a shape should be normalized by the same method as for AGR (Subsection 5.1) before further computation. After the normalization, a shape S is a set of L pixels, $S = \{p_k \in R^2 | k = 1, 2, \cdots, L\}$ and also write $|S| = L$. The minimum bounding rectangle or bounding box of S is denoted by $B(S)$; its width and height, are called w and h, respectively.

Figure 17 shows the algorithm flowchart based on bounding box that divides a shape S into $m\,(row) \times n\,(column)$ parts. The output $\mathbf{B}$ is a set of bounding boxes.

An illustration of this procedure and its result is shown in Figure 18.

(a) Compute the bounding box $B(S)$ of a pixel set S; (b) subdivide S into n vertical slices; (c) compute the bounding box $B(S_j)$ of each resulting pixel set S_j, where $j = 1, 2, \cdots, n$; (d) subdivide each $B(S_j)$ into m horizontal slices; (e) compute the bounding box $B(S_{ij})$ of each resulting pixel set S_{ij}, where $i = 1, 2, \cdots, m$.

Figure 18: The five steps of bounding box splitting

To represent each bounding box, one method is that partial points of the set of bounding boxes are sampled. Figure 19 shows an example.

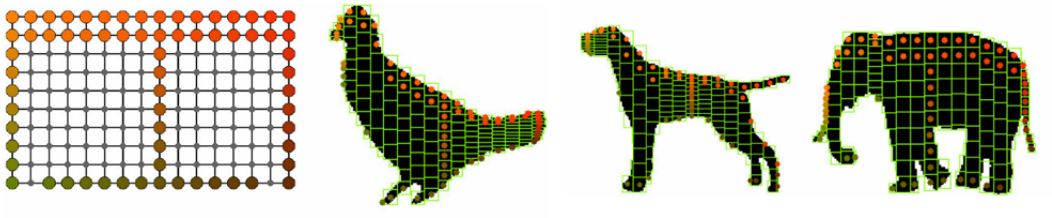


Figure 19: A sample points on lattice and examples of how it is mapped onto different shapes

If $\mathbf{v} = (v_x, v_y)^T$ denotes the location of the bottom left corner of the initial bounding box of S, and $\mathbf{u}_{ij} = (u_x^{ij}, u_y^{ij})$ denotes the center of sample box $\mathbf{B}_{ij}$, then the coordinates

$$\begin{pmatrix} \mu_x^{ij} \\ \mu_y^{ij} \end{pmatrix} = \begin{pmatrix} \left(u_x^{ij} - v_x\right)/w \\ \left(u_y^{ij} - v_y\right)/h \end{pmatrix}$$

provide a scale invariant representation of S. Sampling k points of an $m \times n$ lattice therefore allows to represent S as a vector

$$r = \left[\mu_x^{i(1)j(1)}, \mu_y^{i(1)j(1)}, \cdots, \mu_x^{i(k)j(k)}, \mu_y^{i(k)j(k)}\right]$$

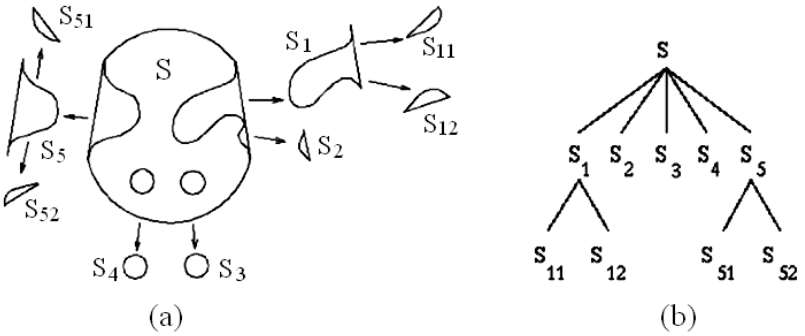
where $i(\alpha) < i(\beta)$ if $\alpha < \beta$ and likewise for the index j.

Bounding box representation is a simple computational geometry approach to compute homeomorphisms between shapes and lattices. It is storage and time efficient. It is invariant to rotation, scaling and translation and also robust against noisy shape boundaries.

5.3 Convex hull

The approach is that the shape is represented by a serie of convex hulls. The convex region has be defined in Sebsection 1.2.8. The convex hull H of a region is its smallest convex region including it. In other words, for a region S, the convex hull $conv(S)$ is defined as the smallest convex set in R^2 containing S. In order to decrease the effect of noise, common practice is to first smooth a boundary prior to partitioning.

The representation of the shape may be obtained by a recursive process which results in a concavity tree. See Figure 20. Each concavity can be described by its area, chord (the line connects the cut of the concavity) length, maximum curvature, distance from maximum curvature point to the chord. The matching between shapes becomes a string or a graph matching.



(a) Convex hull and its concavities; (b) concavity tree representation of convex hull.

Figure 20: Illustrates recursive process of convex hull

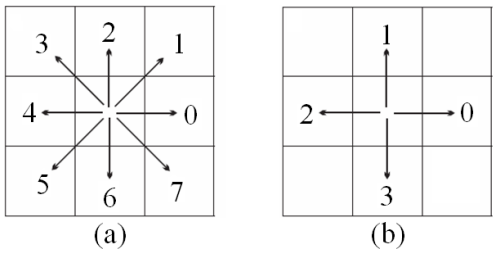
Convex hull representation has a high storage efficiency. It is invariant to rotation, scaling and translation and also robust against noisy shape boundaries (after filtering). However, extracting the robust convex hulls from the shape is where the shoe pinches. [26, 27] and [28] gave the boundary tracing method and morphological methods to achieve convex hulls respectively.

5.4 Chain code

Chain code is a common approach for representing different rasterized shapes as line-drawings, planar curves, or contours. Chain code describes an object by a sequence of unit-size line segments with a given orientation [7]. Chain code can be viewed as a connected sequence of straight-line segments with specified lengths and directions [29].

5.4.1 Basic chain code

Freeman [30] first introduced a chain code that describes the movement along a digital curve or a sequence of border pixels by using so-called 8-connectivity or 4-connectivity. The direction of each movement is encoded by the numbering scheme $\{i|i = 0, 1, 2, \cdots, 7\}$ or $\{i|i = 0, 1, 2, 3\}$ denoting a counter-clockwise angle of $45° \times i$ or $90° \times i$ regarding the positive x-axis, as shown in Figure 21.



(a) Chain code in eight directions (8-connectivity); (b) chain code in four directions (4-connectivity).

Figure 21: Basic chain code direction

By encoding relative, rather than absolute position of the contour, the basic chain code is translation invariant. We can match boundaries by comparing their chain codes, but with the two main problems: 1) it is very sensitive to noise; 2) it is not rotationally invariant. To solve these problems, differential chain codes (DCC) and resampling chain codes (RCC) were proposed.

Differential chain codes (DCC) is encoding differences in the successive 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. This differencing allows us to rotate the object in 90-degree increments and still compare the objects, but it doesn't get around the inherent sensitivity of chain codes to rotation on the discrete pixel grid.

Re-sampling chain codes (RCC) consists in re-sampling the boundary onto a coarser grid and then computing the chain codes of this coarser representation. This smoothes out small variations and noise but can help compensate for differences in chain-code length due to the pixel grid.

5.4.2 Vertex chain code (VCC)

To improve chain code efficiency, in [29] the authors proposed a chain code for shape representation according to vertex chain code (VCC). An element of the VCC indicates the number of cell vertices, which are in touch with the bounding contour of the shape in that element's position. Only three elements "1", "2" and "3" can be used to represent the

bounding contour of a shape composed of pixels in the rectangular grid. Figure 22 shows the elements of the VCC to represent a shape.

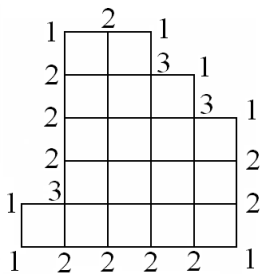


Figure 22: Vertex chain code

5.4.3 Chain code histogram (CCH)

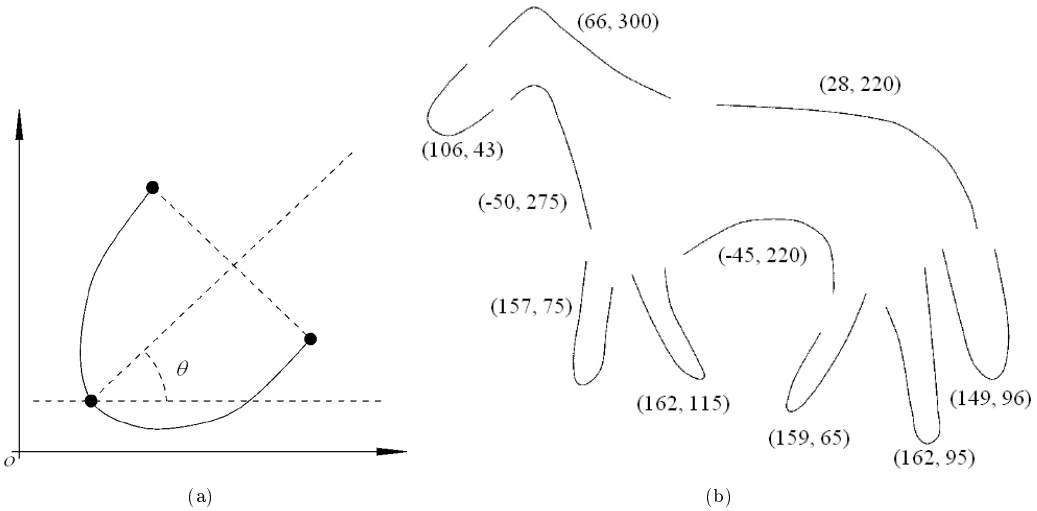
Iivarinen and Visa derive a chain code histogram (CCH) for object recognition [31]. The CCH is computed as $h_i = \sharp\{i \in M$, M is the range of chain code$\}$, $\sharp\{\alpha\}$ denotes getting the number of the value α.

The CCH reflects the probabilities of different directions present in a contour.

If the chain code is used for matching it must be independent of the choice of the starting pixel in the sequence. The chain code usually has high dimensions and is sensitive to noise and any distortion. So, except the CCH, the other chain code approaches are often used as contour representations, but is not as contour attributes.

5.5 Smooth curve decomposition

In [32], the authors proposed smooth curve decomposition as shape descriptor. The segment between the curvature zero-crossing points from a Gaussian smoothed boundary are used to obtain primitives, called tokens. The feature for each token is its maximum curvature and its orientation. In Figure 23, the first number in the parentheses is its maximum curvature and the second is its orientation.



(a) θ is the orientation of this token; (b) an example of smooth curve decomposition.

Figure 23: Smooth curve decomposition

The similarity between two tokens is measured by the weighted Euclidean distance. The shape similarity is measured according to a non-metric distance. Shape retrieval based on token representation has shown to be robust in the presence of partially occulted objects, translation, scaling and rotation.

5.6 Symbolic representation based on the axis of least inertia

In [33], a method of representing a shape in terms of multi-interval valued type data is proposed. The proposed shape representation scheme extracts symbolic features with reference to the axis of least inertia, which is unique to the shape. The axis of least inertia (ALI) of a shape is defined as the line for which the integral of the square of the distances to points on the shape boundary is a minimum (cf. Subsection 1.2.2).

Once the ALI is calculated, each point on the shape curve is projected on to ALI. The two farthest projected points say $E1$ and $E2$ on ALI are chosen as the extreme points as shown in Figure 24. The Euclidean distance between these two extreme points defines the length of ALI. The length of ALI is divided uniformly by a fixed number n; the equidistant points are called feature points. At every feature point chosen, an imaginary line perpendicular to the ALI is drawn. It is interesting to note that these perpendicular lines may intersect the shape curve at several points. The length of each imaginary line in shape region is computed and the collection of these lengths in an ascending order defines the value of the feature at the respective feature point.

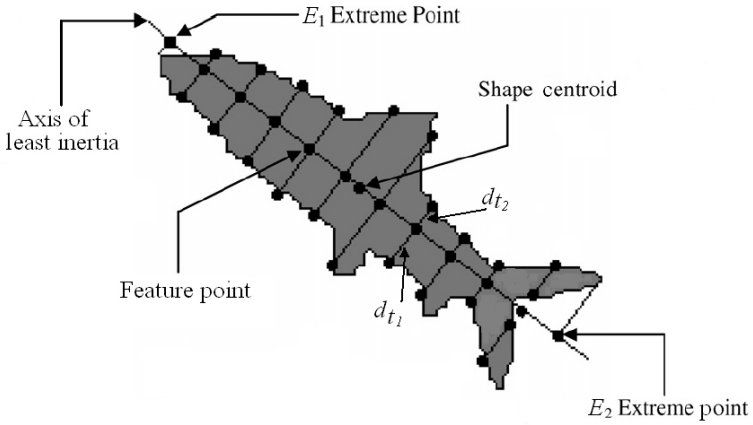


Figure 24: Symbolic features based axis of least inertia

Let S be a shape to be represented and n the number of feature points chosen on its ALI. Then the feature vector F representing the shape S, is in general of the form $F = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_t, ..., \mathbf{f}_n]$, where $\mathbf{f}_t = \{d_{t_1}, d_{t_2}, \cdots d_{t_k}\}$ for some $t_k \geq 1$.

The feature vector F representing the shape S is invariant to image transformations viz., uniform scaling, rotation, translation and flipping (reflection).

5.7 Beam angle statistics

Beam angle statistics (BAS) shape descriptor is based on the beams originated from a boundary point, which are defined as lines connecting that point with the rest of the points on the boundary [34].

Let B be the shape boundary. $B = \{P_1, P_2, \cdots, P_N\}$ is represented by a connected sequence of points, $P_i = (x_i, y_i)$, $i = 1, 2, \cdots, N$, where N is the number of boundary points. For each point P_i, the beam angle between the forward beam vector $V_{i+k} = \overrightarrow{P_i P_{i+k}}$ and backward beam vector $V_{i-k} = \overrightarrow{P_i P_{i-k}}$ in the k^{th} order neighborhood system, is then computed as (see Figure 25, k=5 for example)

$$C_k(i) = (\theta_{V_{i+k}} - \theta_{V_{i-k}})$$

where $\theta_{V_{i+k}} = \arctan \dfrac{y_{i+k} - y_i}{x_{i+k} - x_i}$, $\theta_{V_{i-k}} = \arctan \dfrac{y_{i-k} - y_i}{x_{i-k} - x_i}$

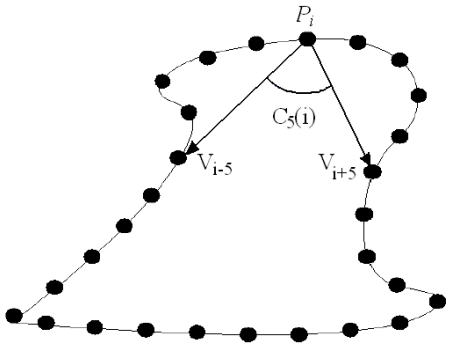
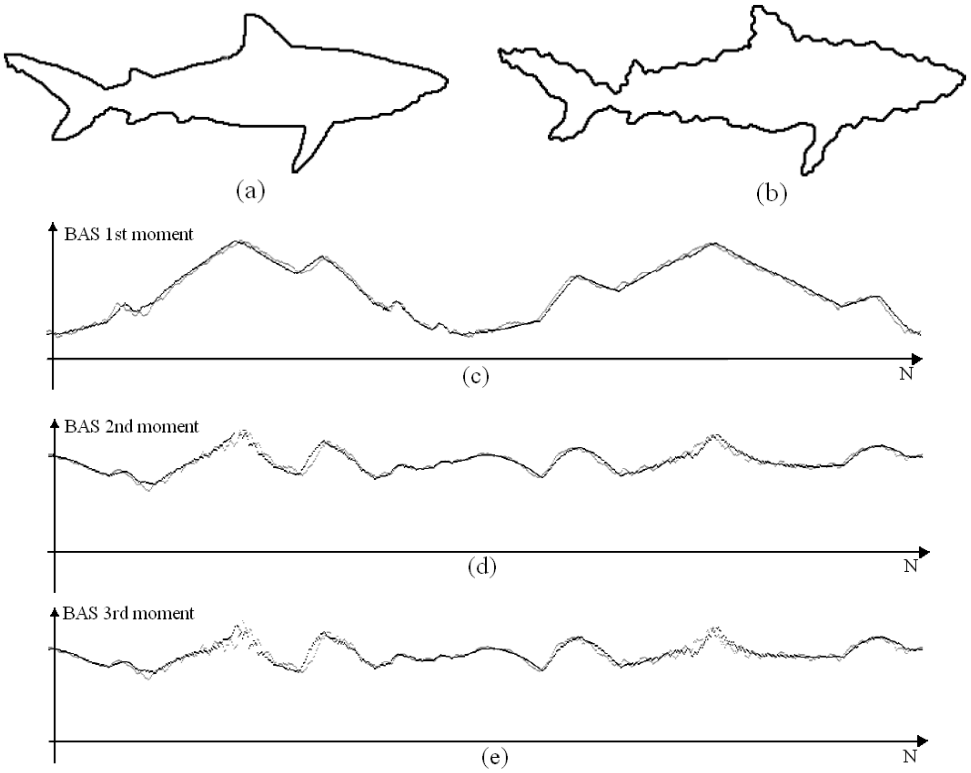


Figure 25: Beam angle at the neighborhood system 5 for a boundary point



(a) Original contour; (b) noisy contour; (c), (d) and (e) are the BAS plot 1^{st}, 2^{nd} and 3^{rd} moment, respectively.

Figure 26: The BAS descriptor for original and noisy contour

For each boundary point P_i of the contour, the beam angle $C_k(i)$ can be taken as a random variable with the probability density function $P(C_k(i))$. Therefore, beam angle statistics (BAS), may provide a compact representation for a shape descriptor. For this purpose, m^{th} moment of the random variable $C_k(i)$ is defined as follows:

$$E[C^m(i)] = \sum_{k=1}^{(N/2)-1} C_k^m(i) \cdot P_k(C_k(i)) \quad m = 1, 2, \cdots$$

In the above formula E indicates the expected value. See Figure 26 as an example.

Beam angle statistics shape descriptor captures the perceptual information using the statistical information based on the beams of individual points. It gives globally discriminative features to each boundary point by using all other boundary points. BAS descriptor is also quite stable under distortions and is invariant to translation, rotation and scaling.

5.8 Shape matrix

Shape matrix descriptor is an $M \times N$ matrix to present a region shape. There are two basic modes of shape matrix: Square model [35] and Polar model [36].

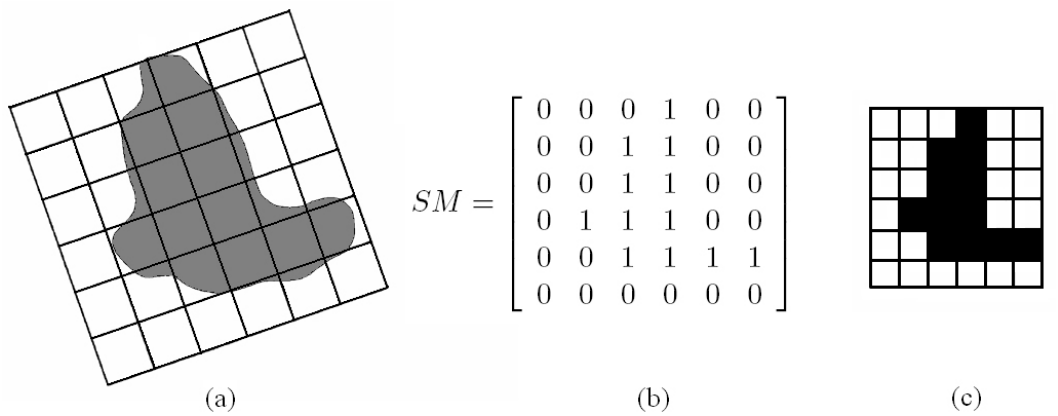
5.8.1 Square model shape matrix

Square model of shape matrix, also called grid descriptor [37, 35], is constructed by the following: for the shape S, construct a square centered on the center of gravity G of S; the size of each side is equal to $2L$, L is the maximum Euclidean distance from G to a point M on the boundary of the shape. Point M lies in the center of one side and GM is perpendicular to this side.

Divide the square into $N \times N$ subsquares and denote S_{kj}, $k, j = 1, \cdots, N$, the subsquares of the constructed grid. Define the shape matrix $SM = [B_{kj}]$,

$$B_{kj} = \begin{cases} 1 \Leftrightarrow & \mu(S_{kj} \cap S) \geq \mu(S_{kj})/2 \\ 0 & otherwise \end{cases}$$

where $\mu(F)$ is the area of the planar region F. Figure 27 shows an example of square model of shape matrix.



(a) Original shape region; (b) square model shape matrix; (c) reconstruction of the shape region.

Figure 27: Square model shape matrix

For a shape with more than one maximum radius, it can be described by several shape matrices and the similarity distance is the minimum distance between these matrices. In [35], authors gave a method to choose the appropriate shape matrix dimension.

5.8.2 Polar model shape matrix

Polar model of shape matrix is constructed by the following steps. Let G be the center of gravity of the shape, and GA is the maximum radius of the shape. Using G as center, draw n circles with radii equally spaced. Starting from GA, and counterclockwise, draw radii that divide each circle into m equal arcs. The values of the matrix are same as in square model shape matrix. Figure 28 shows an example, where n = 5 and m =12. Its polar model of shape matrix is

$$PSM = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

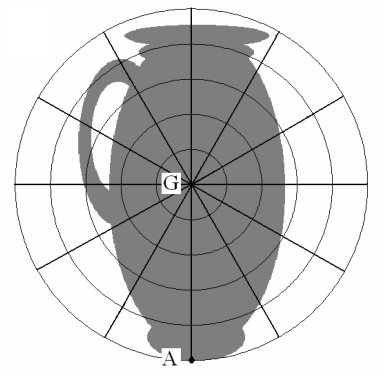


Figure 28: Polar model shape

Polar model of shape matrix is simpler than square model because it only uses one matrix no matter how many maximum radii are on the shape. However, since the sampling density is not constant with the polar sampling raster, a weighed shape matrix is necessary. For the detail, refer to [36].

The shape matrix exists for every compact shape. There is no limit to the scope of the shapes that the shape matrix can represent. It can describe even shapes with holes. Shape matrix is also invariant under translation, rotation and scaling of the object. The shape of the object can be reconstructed from the shape matrix; the accuracy is given by the size of the grid cells.

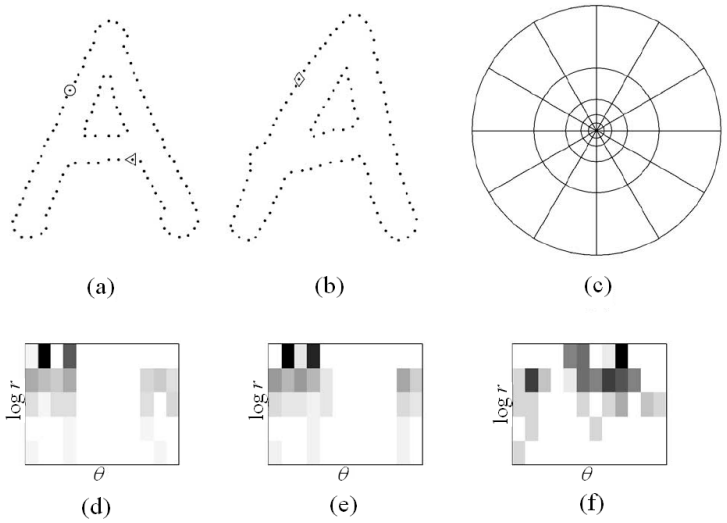
5.9 Shape context

In [38], the shape context has been shown to be a powerful tool for object recognition tasks. It is used to find corresponding features between model and image.

Shape contexts analysis begins by taking N samples from the edge elements on the shape. These points can be on internal or external contours. Consider the vectors originating from a point to all other sample points on the shape. These vectors express the appearance of the entire shape relative to the reference point. This descriptor is the histogram of the relative polar coordinates of all other points:

$$h_i(k) = \sharp\{Q \neq P_i \ : \ (Q - P_i) \in bin(k)\}$$

An example is shown in Figure 29. (c) is the diagram of log-polar histogram that has 5 bins for the polar direction and 12 bins for the angular direction. The histogram of a point P_i is formed by the following: putting the center of the histogram bins diagram on the point P_i, each bin of this histogram contains a count of all other sample points on the shape falling into that bin. Note on this figure, the shape contexts (histograms) for the points marked by $\circ$ (in (a)), $\diamond$ (in (b)) and $\triangleleft$ (in (a)) are shown in (d), (e) and (f), respectively. It is clear that the shape contexts for the points marked by $\circ$ and $\diamond$, which are computed for relatively similar points on the two shapes, have visual similarity. By contrast, the shape context for $\triangleleft$ is quite different from the others. Obviously, this descriptor is a rich description, since as N gets large, the representation of the shape becomes exact.



(a) and (b) Sampled edge points of two shapes; (c) diagram of log-polar histogram bins used in computing the shape contexts; (d), (e) and (f) shape contexts for reference sample points marked by $\circ$, $\diamond$ and $\triangleleft$ in (a) and (b), respectively. (Dark=large value).

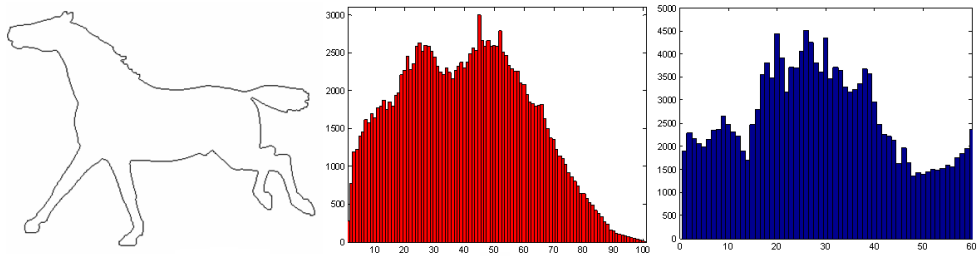
Figure 29: Shape context computation and graph matching

Shape context matching is often used to find the corresponding points on two shapes. It has been applied to a variety of object recognition problems [38, 39, 40, 41]. The shape context descriptor has the following invariance properties:

- translation: the shape context descriptor is inherently translation invariant as it is based on relative point locations.
- scaling: for clutter-free images the descriptor can be made scale invariant by normalizing the radial distances by the mean (or median) distance between all point pairs.
- rotation: it can be made rotation invariant by rotating the coordinate system at each point so that the positive x-axis is aligned with the tangent vector.
- shape variation: the shape context is robust against slight shape variations.
- few outliers: points with a final matching cost larger than a threshold value are classified as outliers. Additional 'dummy' points are introduced to decrease the effects of outliers.

5.10 Chord distribution

The basic idea of chord distribution is to calculate the lengths of all chords in the shape (all pair-wise distances between boundary points) and to build a histogram of their lengths and orientations [42]. The "lengths" histogram is invariant to rotation and scales linearly with the size of the object. The "angles" histogram is invariant to object size and shifts relative to object rotation. Figure 30 gives an example of chord distribution.



(a) Original contour; (b) chord lengths histogram; (c) chord angles histogram (each stem covers 3 degrees).

Figure 30: Chord distribution

5.11 Shock graphs

Shock graphs is a descriptor based on the medial axis. The medial axis is the most popular that has been proposed as a useful shape abstraction tool for the representation and modeling of animate shapes. Skeleton and medial axes have been extensively used for characterizing objects satisfactorily using structures that are composed of line or arc patterns. Medial axis is an image processing operation which reduces input shapes to axial stick-like representations. It is as the loci of centers of bi-tangent circles that fit entirely within the foreground region being considered. Figure 31 illustrates the medial axis for a rectangular shape.

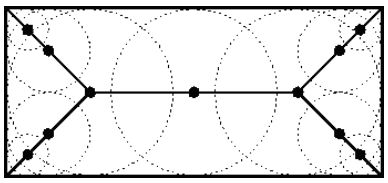


Figure 31: Medial axis of a rectangle defined in terms of bi-tangent circles

We notice that the radius of each circle is variable. This variable is a function of the loci of points on the medial axis. We call this function as the radius function.

A shock graph is a shape abstraction that decomposes a shape into a set of hierarchically organized primitive parts. Siddiqi and Kimia define the concept of a shock graph [43] as an abstraction of the medial axis of a shape onto a directed acyclic graph (DAG). Shock segments are curve segments of the medial axis with monotonic flow, and give a more refined partition of the medial axis segments. Figure 32 is for example.

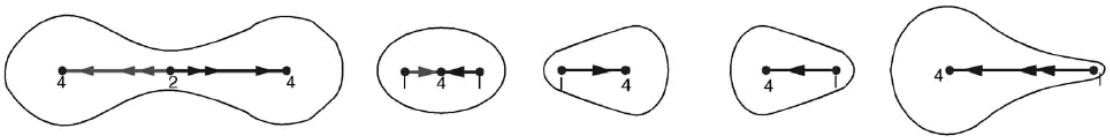


Figure 32: Shock segments

The skeleton points are first labeled according to the local variation of the radius function at each point. Shock graph can distinguish the shapes but the medial axis cannot. Figure 33 shows two examples of shapes and their shock graphs.

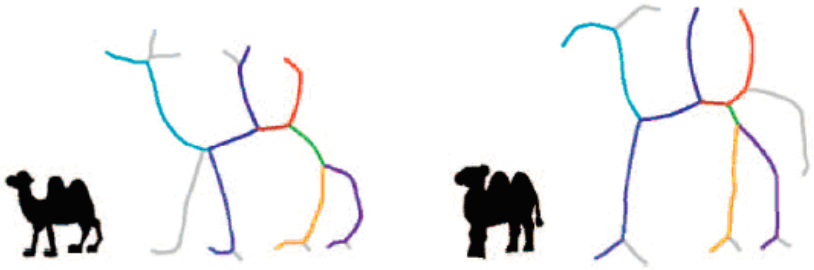


Figure 33: Examples of shapes and their shock graphs

To calculate the distance between two shock graphs, in [44], the authors employ a polynomial-time edit-distance algorithm. It shows this algorithm has the good performances for boundary perturbations, articulation and deformation of parts, segmentation errors, scale variations, viewpoint variations and partial occultation. However the authors also indicate the computation complexity is very high. The matching shape typically takes about 3-5 minutes on an SGI Indigo II (195 MHz), which limits the number of shapes that can be practically matched.

5.12 Discussions

Spacial feature descriptor is a direct method to describe a shape. These descriptors can apply the theory of tree-based (Adaptive grid resolution and Convex hull), statistic (Chain code histogram, Beam angle statistics, Shape context and Chord distribution) or syntactic analysis (Smooth curve decomposition) to extract or represent the feature of a shape. This description scheme not only compresses the data of a shape, but also provides a compact and meaningful form to facilitate further recognition operations.

6 Moments

The concept of moment in mathematics evolved from the concept of moment in physics. It is an integrated theory system. For both contour and region of a shape, one can use moment's theory to analyze the object.

6.1 Boundary moments

Boundary moments, analysis of a contour, can be used to reduce the dimension of boundary representation [28]. Assume shape boundary has been represented as a 1-D shape representation $z(i)$ as introduced in Section 3, the r^{th} moment m_r and central moment μ_r can be estimated as

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

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

$$F_1 = \frac{(\mu_2)^{1/2}}{m_1}, \quad F_2 = \frac{\mu_3}{(\mu_2)^{3/2}} \quad and \quad F_3 = \frac{\mu_4}{(\mu_2)^2},$$

The other boundary moments method treats the 1-D shape feature function $z(i)$ as a random variable $\mathbf{v}$ and creates a K bins histogram $p(v_i)$ from $z(i)$. Then, the r^{th} central moment is obtained by

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

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.

6.2 Region moments

Among the region-based descriptors, moments are very popular. These include invariant moments, Zernike moments, Radial Chebyshev moments, etc.

The general form of a moment function m_{pq} of order $(p+q)$ of a shape region can be given as:

$$m_{pq} = \sum_x \sum_y \Psi_{pq}(x,y) f(x,y) \qquad p,q = 0,1,2\cdots$$

where Ψ_{pq} is known as the *moment weighting kernel* or the *basis set*; $f(x,y)$ is the shape region Eq. **??**.

6.2.1 Invariant moments (IM)

Invariant moments (IM) are also called geometric moment invariants. Geometric moments, are the simplest of the moment functions with basis $\Psi_{pq} = x^p y^q$, while complete, is not orthogonal [30]. Geometric moment function m_{pq} of order $(p+q)$ is given as:

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

The geometric central moments, which are invariant to translation, are defined as:

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

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$

A set of 7 invariant moments (IM) are given by [30]:

$\phi_1 = \eta_{20} + \eta_{02}$

$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$

$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})$
$\cdot \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$

$\phi_6 = (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}^2(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$

$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})$
$\cdot \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$

where $\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma}$ and $\gamma = 1 + (p + q)/2$ for $p + q = 2, 3, \cdots$

IM are computationally simple. Moreover, they are invariant to rotation, scaling and translation. However, they have several drawbacks [45]:

- information redundancy: since the basis is not orthogonal, these moments suffer from a high degree of information redundancy.
- noise sensitivity: higher-order moments are very sensitive to noise.
- large variation in the dynamic range of values: since the basis involves powers of p and q, the moments computed have large variation in the dynamic range of values for different orders. This may cause numerical instability when the image size is large.

6.2.2 Algebraic moment invariants

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 [46]. The algebraic moment invariants can be constructed up to arbitrary order and are invariant to affine transformations. However, algebraic moment invariants performed either very well or very poorly on the objects with different configuration of outlines.

6.2.3 Zernike moments (ZM)

Zernike Moments (ZM) are orthogonal moments [45]. The complex Zernike moments are derived from orthogonal Zernike polynomials:

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

where $R_{nm}(r)$is the orthogonal radial polynomial:

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

$n = 0, 1, 2, \cdots$; $0 \leq |m| \leq n$; and $n - |m|$ is even.

Zernike polynomials are a complete set of complex valued functions orthogonal over the unit disk, i.e., $x^2 + y^2 \leq 1$. The Zernike moment of order n with repetition m of shape region $f(x, y)$ (Eq. **??**) is given by:

$$Z_{nm} = \frac{n+1}{\pi} \sum_r \sum_\theta f(r\cos\theta, r\sin\theta) \cdot R_{nm}(r) \cdot exp(jm\theta) \quad r \leq 1$$

Zernike moments (ZM) have the following advantages [47]:

- rotation invariance: the magnitudes of Zernike moments are invariant to rotation.
- robustness: they are robust to noise and minor variations in shape.
- expressiveness: since the basis is orthogonal, they have minimum information redundancy.

However, the computation of ZM (in general, continuous orthogonal moments) pose several problems:

- coordinate space normalization: the image coordinate space must be transformed to the domain where the orthogonal polynomial is defined (unit circle for the Zernike polynomial).

- numerical approximation of continuous integrals: the continuous integrals must be approximated by discrete summations. This approximation not only leads to numerical errors in the computed moments, but also severely affects the analytical properties such as rotational invariance and orthogonality.
- computational complexity: computational complexity of the radial Zernike polynomial increases as the order becomes large.

6.2.4 Radial Chebyshev moments (RCM)

The radial Chebyshev moment of order p and repetition q is defined as [48]:

$$S_{pq} = \frac{1}{2\pi\rho(p,m)} \sum_{r=0}^{m-1} \sum_{\theta=0}^{2\pi} t_p(r) \cdot exp(-jq\theta) \cdot f(r,\theta)$$

where $t_p(r)$ is the scaled orthogonal Chebyshev polynomials for an image of size $N \times N$:

$$t_0(x) = 1$$
$$t_1(x) = (2x - N + 1)/N$$
$$t_p(x) = \frac{(2p-1)t_1(x)t_{p-1}(x) - (p-1)\left\{1 - \frac{(p-1)^2}{N^2}\right\} t_{p-2}(x)}{p}, \quad p > 1$$

$\rho(p, N)$ is the squared-norm:

$$\rho(p,N) = \frac{N\left(1 - \frac{1}{N^2}\right)\left(1 - \frac{2^2}{N^2}\right)\cdots\left(1 - \frac{p^2}{N^2}\right)}{2p+1} \qquad p = 0, 1, \cdots, N-1$$

and $m = (N/2) + 1$.

The mapping between (r, θ) and image coordinates (x, y) is given by:

$$x = \frac{rN}{2(m-1)}\cos(\theta) + \frac{N}{2}$$

$$y = \frac{rN}{2(m-1)}\sin(\theta) + \frac{N}{2}$$

Compared to Chebyshev moments, radial Chebyshev moments possess rotational invariance property.

6.3 Discussions

Besides the previous moments, there are other moments for shape representation, for example, homocentric polar-radius moment [49], orthogonal Fourier-Mellin moments (OFMMs) [50], pseudo-Zernike Moments [51], etc. The study shows that the moment-based shape descriptors are usually concise, robust and easy to compute. It is also invariant to scaling, rotation and translation of the object. However, because of their global nature, the disadvantage of moment-based methods is that it is difficult to correlate high order moments with a shape's salient features.

7 Scale space approaches

In scale space theory a curve is embedded into a continuous family $\{\Gamma_\sigma : \sigma \geq 0\}$ of gradually simplified versions. The main idea of scale spaces is that the original curve $\Gamma = \Gamma_0$ should get more and more simplified, and so small structures should vanish as parameter σ increases. Thus due to different scales (values of σ), it is possible to separate small details from relevant shape properties. The ordered sequence $\{\Gamma_\sigma : \sigma \geq 0\}$ is referred to as evolution of Γ. Scale-spaces find wide application in computer vision, in particular, due to smoothing and elimination of small details.

A lot of shape features can be analyzed in scale-space theory to get more information about shapes. Here we introduced 2 scale-space approaches: curvature scale-space (CSS) and intersection points map (IPM).

7.1 Curvature scale-space

The curvature scale-space (CSS) method, proposed by F. Mokhtarian in 1988, was selected as a contour shape descriptor for MPEG-7. This approach is based on multi-scale representation and curvature to represent planar curves. For convenience, we copy the nature parametrization equation (Eq. **??**) as following:

$$\Gamma(\mu) = (x(\mu), y(\mu)) \tag{15}$$

An evolved version of that curve is defined by

$$\Gamma_\sigma(\mu) = (X(\mu,\sigma), Y(\mu,\sigma))$$

where $X(\mu,\sigma) = x(\mu) * g(\mu,\sigma)$ and $Y(\mu,\sigma) = y(\mu) * g(\mu,\sigma)$, $*$ is the convolution operator, and $g(\mu,\sigma)$ denotes a Gaussian filter with standard deviation σ defined by

$$g(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-\mu^2}{2\sigma^2})$$

Functions $X(\mu,\sigma)$ and $Y(\mu,\sigma)$ are given explicitly by

$$X(\mu,\sigma) = \int_{-\infty}^{\infty} x(v) \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-(\mu-v)^2}{2\sigma^2}) dv$$

$$Y(\mu,\sigma) = \int_{-\infty}^{\infty} y(v) \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-(\mu-v)^2}{2\sigma^2}) dv$$

The curvature of is given by

$$k(\mu,\sigma) = \frac{X_\mu(\mu,\sigma) Y_{\mu\mu}(\mu,\sigma) - X_{\mu\mu}(\mu,\sigma) Y_\mu(\mu,\sigma)}{(X_\mu(\mu,\sigma)^2 - Y_\mu(\mu,\sigma)^2)^{3/2}}$$

where

$X_\mu(\mu,\sigma) = \frac{\partial}{\partial\mu}(x(\mu) * g(\mu,\sigma)) = x(\mu) * g_\mu(\mu,\sigma)$

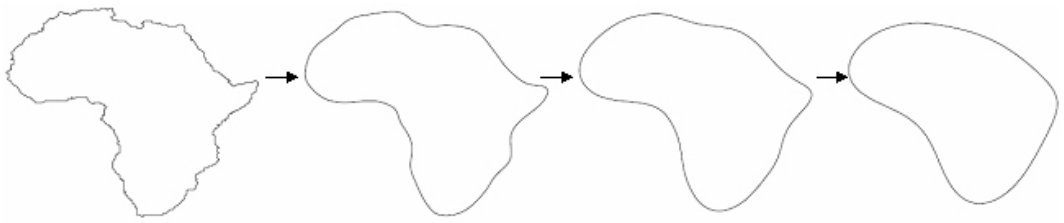
$X_{\mu\mu}(\mu,\sigma) = \frac{\partial^2}{\partial\mu^2}(x(\mu) * g(\mu,\sigma)) = x(\mu) * g_{\mu\mu}(\mu,\sigma)$

$Y_\mu(\mu,\sigma) = \frac{\partial}{\partial\mu}(y(\mu) * g(\mu,\sigma)) = y(\mu) * g_\mu(\mu,\sigma)$

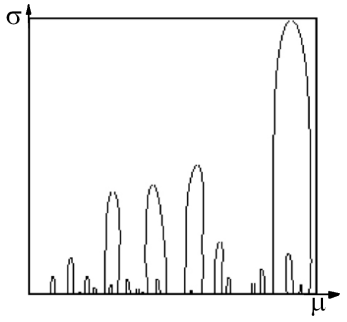
$Y_{\mu\mu}(\mu,\sigma) = \frac{\partial^2}{\partial\mu^2}(y(\mu) * g(\mu,\sigma)) = y(\mu) * g_{\mu\mu}(\mu,\sigma)$

Note that σ is also referred to as a scale parameter. The process of generating evolved versions of Γ_σ as σ increases from 0 to ∞ is referred to as the evolution of Γ_σ. This technique is suitable for removing noise and smoothing a planar curve as well as gradual simplification of a shape.

The function defined by $k(\mu,\sigma) = 0$ is the CSS image of Γ. Figure 34 is a CSS image examples.



(a)



(b)

(a) Evolution of Africa: from left to right $\sigma = 0$(original), $\sigma = 4$, $\sigma = 8$ and $\sigma = 16$, respectively; (b) CSS image of Africa.

Figure 34: Curvature scale-space image

The representation of CSS is the maxima of CSS contour of an image. Many methods for representing the maxima of CSS exist in the literatures [52, 53, 19] and the CSS technique has been shown to be robust contour-based shape representation technique. The basic properties of the CSS representation are as follows:

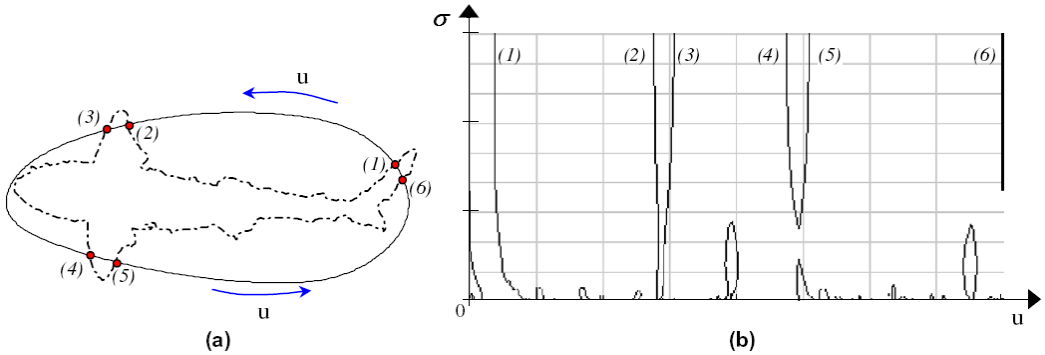
- it captures the main features of a shape, enabling similarity-based retrieval;

- it is robust to noise, changes in scale and orientation of objects;
- it is compact, reliable and fast;
- It retains the local information of a shape. Every concavity or convexity on the shape has its own corresponding contour on the CSS image.

Although CSS has a lot of advantages, it does not always give results in accordance with human vision system. The main drawbacks of this description are due to the problem of shallow concavities/convexities on a shape. It can be shown that the shallow and deep concavities/convexities may create the same large contours on the CSS image. In [54, 55], the authors gave some methods to alleviate these effects.

7.2 Intersection points map

Similarly to the CSS, many methods also use a Gaussian kernel to progressively smooth the curve relatively to the varying bandwidth. In [56], the authors proposed a new algorithm, intersection points map (IPM), based on this principle, instead of characterizing the curve with its curvature involving 2^{nd} order derivatives, it uses the intersection points between the smoothed curve and the original. As the standard deviation of the Gaussian kernel increases, the number of the intersection points decreases. By analyzing these remaining points, features for a pattern can be defined. Since this method deals only with curve smoothing, it needs only the convolution operation in the smoothing process. So this method is faster than the CSS one with equivalent performances. Figure 35 is an example of IPM.



(a) An original contour; (b) an IPM image in the (u, σ) plane. The IPM points indicated by (1)–(6) refer to the corresponding intersection points in (a).

Figure 35: Example of the IPM

The IPM pattern can be identified regardless of its orientation, translation and scale change. It is also resistant to noise for a range of noise energy. The main weakness of this approach is that it fails to handle occulted contours and those having undergone a non-rigid deformation.

7.3 Discussions

As multi-resolution analysis in signal processing, scale-space theory can obtain abundant information about a contour with different scales. In scale-space, global pattern information can be interpreted from higher scales, while detailed pattern information can be interpreted from lower scales. Scale-space algorithm benefit from the boundary information redundancy in the new image, making it less sensitive to errors in the alignment or contour extraction algorithms. The great advantages are the high robustness to noise and the great coherence with human perception.

8 Shape transform domains

The transform domain class includes methods which are formed by the transform of the detected object or the transform of the whole image. Transforms can therefore be used to characterize the appearance of images. The shape feature is represented by the all or partial coefficients of a transform.

8.1 Fourier descriptors

Although, Fourier descriptor (FD) is a 40-year-old technique, it is still considered as a valid description tool. The shape description and classification using FD either in contours or regions are simple to compute, robust to noise and compact. It has many applications in different areas.

8.1.1 One-dimensional Fourier descriptors

In general, Fourier descriptor (FD) is obtained by applying Fourier transform on a shape signature that is a one-dimensional function which is derived from shape boundary coordinates (cf. Section 3). The normalized Fourier transformed coefficients are called the Fourier descriptor of the shape. FD derived from different signatures has significant different performance on shape retrieval. As shown in [10, 53], FD derived from centroid distance function *r(t)* outperforms FD derived from other shape signatures in terms of overall performance. The discrete Fourier transform of *r(t)* is then given by

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

Since the centroid distance function *r(t)* is only invariant to rotation and translation, the acquired Fourier coefficients have to be further normalized so that they are scaling and start point independent shape descriptors. From Fourier transform theory, the general form of the Fourier coefficients of a contour centroid distance function $r(t)$ transformed through scaling and change of start point from the original function $r(t)^{(o)}$ is given by

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

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

$$b_n = \frac{a_n}{a_1} = \frac{exp(jn\tau) \cdot s \cdot a_n^{(o)}}{exp(j\tau) \cdot s \cdot a_1^{(o)}} = \frac{a_n^{(o)}}{a_1^{(o)}} exp[j(n-1)\tau] = b_n^{(o)} exp[j(n-1)\tau]$$

where b_n and $b_n^{(o)}$ are the normalized Fourier coefficients of the transformed shape and the original shape, respectively. If we ignore the phase information and only use magnitude of the coefficients, then $|b_n|$ and $\left|b_n^{(o)}\right|$ are the same. In other words, $|b_n|$ is invariant to translation, rotation, scaling and change of start point.

The set of magnitudes of the normalized Fourier coefficients of the shape $\{|b_n|, 0 < n < N\}$ are used as shape descriptors, denoted as

$$\{FD_n, 0 < n < N\}.$$

One-dimensional FD has several nice characteristics such as simple derivation, simple normalization and simple to do matching. As indicated by [53], for efficient retrieval, 10 FDs are sufficient for shape description.

8.1.2 Region-based Fourier descriptor

The region-based FD is referred to as generic FD (GFD), which can be used for general applications. Basically, GFD is derived by applying a modified polar Fourier transform (MPFT) on shape image [57, 12]. In order to apply MPFT, the polar shape image is treated as a normal rectangular image. The steps are

1. the approximated normalized image is rotated counter clockwise by an angular step sufficiently small.
2. the pixel values along positive x-direction starting from the image center are copied and pasted into a new matrix as row elements.
3. the steps 1 and 2 are repeated until the image is rotated by 360°.

The result of these steps is that an image in polar space plots into Cartesian space.

Figure 36 shows the polar shape image turning into normal rectangular image.

(a) (b)

(a) Original shape image in polar space; (b) polar image of (a) plotted into Cartesian space.

Figure 36: The polar shape image turns into normal rectangular image.

The Fourier transform is acquired by applying a discrete 2D Fourier transform on this shape image.

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

where $0 \leq r = \sqrt{[(x-g_x)^2 + (y-g_y)^2]} < R$, and $\theta_i = i(2\pi/T)$; $0 \leq \rho < R$, $0 \leq \phi < T$. (g_x, g_y) is the center of mass of the shape; R and T are the radial and angular resolutions. The acquired Fourier coefficients are translation invariant. Rotation and scaling invariance are achieved by the following:

$$GFD = \left\{ \frac{|pf(0,0)|}{area}, \frac{|pf(0,1)|}{|pf(0,0)|}, \cdots, \frac{|pf(0,n)|}{|pf(0,0)|}, \cdots, \frac{|pf(m,0)|}{|pf(0,0)|}, \cdots, \frac{|pf(m,n)|}{|pf(0,0)|} \right\}$$

where *area* is the area of the bounding circle in which the polar image resides. m is the maximum number of the radial frequencies selected and n is the maximum number of angular frequencies selected. m and n can be adjusted to achieve hierarchical coarse to fine representation requirement.

For efficient shape description, in the implementation of [57], 36 GFD features reflecting $m = 4$ and $n = 9$ are selected to index the shape. The experimental results have shown GFD as invariant to translation, rotation, and scaling. For obtaining the affine and general minor distortions invariance, in [57], the authors proposed Enhanced Generic Fourier Descriptor (EGFD) to improve the GFD properties.

8.2 Wavelet transform

A hierarchical planar curve descriptor is developed by using the wavelet transform [58]. This descriptor decomposes a curve into components of different scales so that the coarsest scale components carry the global approximation information while the finer scale components contain the local detailed information. The wavelet descriptor has many desirable properties such as multi-resolution representation, invariance, uniqueness, stability, and spatial localization. In [59], the authors use dyadic wavelet transform deriving an affine invariant function. In [60], a descriptor is obtained by applying the Fourier transform along the axis of polar angle and the wavelet transform along the axis of radius. This feature is also invariant to translation, rotation, and scaling. At same time, the matching process of wavelet descriptor can be accomplished cheaply.

8.3 Angular radial transformation

The angular radial transformation (ART) is based in a polar coordinate system where the sinusoidal basis functions are defined on a unit disc. Given an image function in polar coordinates, $f(\rho,\theta)$, an ART coefficient F_{nm} (radial order n, angular order m) can be defined as [61]:

$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}(\rho,\theta) f(\rho,\theta) \rho d\rho d\theta$$

$V_{nm}(\rho,\theta)$ is the ART basis function and is separable in the angular and radial directions so that:

$$V_{nm}(\rho,\theta) = A_m(\theta) R_n(\rho)$$

The angular basis function, A_m, is an exponential function used to obtain orientation invariance. This function is defined as:

$$A_m(\theta) = \frac{1}{2\pi} e^{jm\theta}$$

R_n, the radial basis function, is defined as:

$$R_n(\rho) = \begin{cases} 1 & if\ n = 0 \\ 2\cos(\pi n \rho) & if\ n \neq 0 \end{cases}$$

In MPEG-7, twelve angular and three radial functions are used ($n < 3, m < 12$). Real parts of the 2-D basis functions are shown in Figure 37.

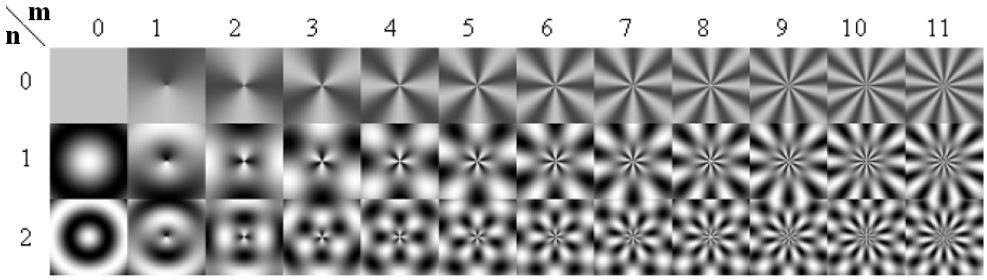


Figure 37: Real parts of the ART basis functions

For scale normalization, the ART coefficients are divided by the magnitude of ART coefficient of order $n = 0, m = 0$.

MPEG-7 standardization process showed the efficiency of 2-D angular radial transformation. This descriptor is robust to translations, scaling, multi-representation (remeshing, weak distortions) and noises.

8.4 Shape signature harmonic embedding

A harmonic function is obtained by a convolution between the Poisson kernel $P_R(r, \theta)$ and a given boundary function $u(Re^{j\phi})$. Poisson kernel is defined by

$$P_R(r, \theta) = \frac{R^2 - r^2}{R^2 - 2Rr\cos(\theta) + r^2}$$

The boundary function could be any real- or complex-valued function, but here we choose shape signature functions for the purpose of shape representation. For any shape signature $s[n], n = 0, 1, \cdots, N-1$, the boundary values for a unit disk can be set as

$$u(Re^{j\phi}) = u(Re^{j\omega_0 n}) = s[n]$$

where $\omega_0 = 2\pi/N$, $\phi = \omega_0 n$.

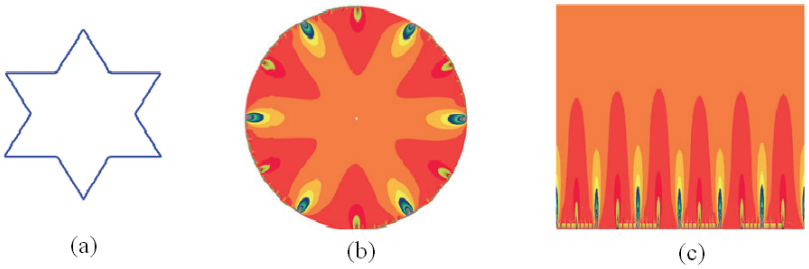
So the harmonic function u can be written as

$$u(re^{j\theta}) = \frac{1}{2\pi}\int_0^{2\pi} u(Re^{j\phi})P_R(r, \phi - \theta)d\phi \tag{16}$$

The Poisson kernel $P_R(r, \theta)$ has a low-pass filter characteristic, where the radius r is inversely related to the bandwidth of the filter. The radius r is considered as scale parameter of a multi-scale representation [62]. Another important property is $P_R(0, \theta) = 1$, indicating $u(0)$ is the mean value of boundary function $u(Re^{j\phi})$.

In [62], the authors proposed a formulation of a discrete closed-form solution for the Poisson's integral formula Eq. 16, so that one can avoid the need for approximation or numerical calculation of the Poisson summation form.

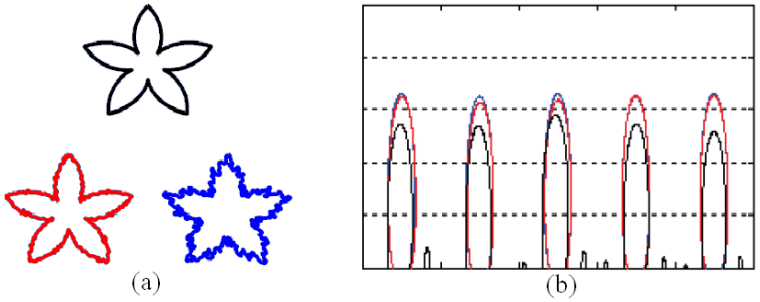
As in Subsection 8.1.2, the harmonic function inside the disk can be mapped to a rectilinear space for a better illustration. Figure 38 shows an example for a star shape. Here, we used curvature as the signature to provide boundary values.



(a) Example shape; (b) harmonic function within the unit disk; (c) rectilinear mapping of the function.

Figure 38: Harmonic embedding of curvature signature

The zero-crossing image of the harmonic functions is extracted as shape feature. This shape descriptor is invariant to translation, rotation and scaling. It is also robust to noise. Figure 39 is for example. The original curve is corrupted with different noise levels, and the harmonic embeddings show robustness to the noise.



(a) Original and noisy shapes; (b) harmonic embedding images for centroid distance signature.

Figure 39: centroid distance signature harmonic embedding that is robust to noisy boundaries

At same time, it is more efficient than CSS descriptor. However, it is not suitable for similarity retrieval, because it is unconsistent to non-rigid transform.

8.5 $\Re$-Transform

The $\Re$-Transform to represent a shape is based on the Radon transform. The approach is presented as follow. We assume that the function f is the domain of a shape, cf. Eq. **??**. Its Radon transform is defined by:

$$T_R(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

where $\delta(.)$ is the Dirac delta-function:

$$\delta(x) = \begin{cases} 1 & if\ x = 0 \\ 0 & otherwise \end{cases}$$

$\theta \in [0,\pi]$ and $\rho \in (-\infty,\infty)$. In other words, Radon transform $T_R(\rho,\theta)$ is the integral of f over the line $L_{(\rho,\theta)}$ defined by $\rho = x\cos\theta + y\sin\theta$.

Figure 40 is an example of a shape and its Radon transform.

(a) Shape (b) Radon transform of (a)

Figure 40: A shape and its Radon transform

The following transform is defined as $\Re$-transform:

$$\Re_f(\theta) = \int_{-\infty}^{\infty} T_R^2(\rho,\theta)d\rho$$

where $T_R(\rho,\theta)$ is the Radon transform of the domain function f. In [63], the authors show the following properties of $\Re_f(\theta)$:

- periodicity: $\Re_f(\theta \pm \pi) = \Re_f(\theta)$
- rotation: a rotation of the image by an angle θ_0 implies a translation of the $\Re$-transform of θ_0: $\Re_f(\theta + \theta_0)$.
- translation: the $\Re$-transform is invariant under a translation of the shape f by a vector $\overrightarrow{u} = (x_0, y_0)$.
- scaling: a change of the scaling of the shape f induces a scaling in the amplitude only of the $\Re$-transform.

Given a large collection of shapes, one $\Re$-transform per shape is not efficient to distinguish from the others because the $\Re$-transform provides a highly compact shape representation. In this perspective, to improve the description, each shape is projected in the Radon space for different segmentation levels of the Chamfer distance transform. Chamfer distance transform is introduced in [64, 65] (See Appendix A for detail).

Given the distance transform of a shape, the distance image is segmented into N equidistant levels to keep the segmentation isotropic. For each distance level, pixels having a distance value superior to that level are selected and at each level of segmentation, an $\Re$-transform is computed. In this manner, both the internal structure and the boundaries of the shape are captured.

Since a rotation of the shape implies a corresponding shift of the $\Re$-transform. Therefore, a one-dimensional Fourier transform is applied on this function to obtain the rotation invariance. After the discrete one-dimensional Fourier transform F, $\Re$-transform descriptor vector is defined as follows:

$$RTD = (\frac{F\Re^1(\frac{\pi}{M})}{F\Re^1(0)}, \cdots, \frac{F\Re^1(\frac{i\pi}{M})}{F\Re^1(0)}, \cdots, \frac{F\Re^1(\pi)}{F\Re^1(0)}, \cdots, \frac{F\Re^N(\frac{\pi}{M})}{F\Re^N(0)}, \cdots, \frac{F\Re^N(\frac{i\pi}{M})}{F\Re^N(0)}, \cdots, \frac{F\Re^N(\pi)}{F\Re^N(0)})$$

where $i \in [1, M]$, M is the angular resolution. $F\Re^\alpha$ is the magnitude of Fourier transform to $\Re$-transform. $\alpha \in [1, N]$, is the segmentation level of Chamfer distance transform.

8.6 Shapelets descriptor

Shapelets descriptor was proposed to present a model for animate shapes and extracting meaningful parts of objects. The model assumes that animate shapes (2D simple closed curves) are formed by a linear superposition of a number of shape bases. A basis function $\psi(s;\mu,\sigma)$ is defined in [66]: $\mu \in [0,1]$ indicates the location of the basis function relative to the domain of the observed curve, and σ is the scale of the function ψ. Figure 41 shows the shape of the basis function ψ at different σ values. It displays variety with different parameter and transforms.

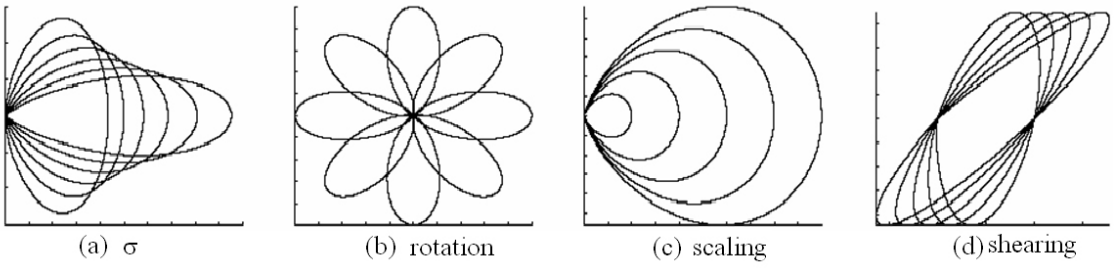


Figure 41: Each shape base is a lobe-shaped curve

The basis functions are subject to affine transformations by a 2 × 2 matrix of basis coefficients:

$$A_k = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}$$

The variables for describing a base are denoted by $\mathbf{b}_k = (A_k, \mu_k, \sigma_k)$ and are termed basis elements. The shapelet is defined by

$$\gamma(s;\mathbf{b}_k) = A_k \psi(s;\mu_k,\sigma_k)$$

Figure 41 (b,c,d) demonstrates shapelets obtained from the basis functions ψ by the affine transformations of rotation, scaling, and shearing respectively, as indicated by the basis coefficient A_k. By collecting all the shapelets at various μ, σ, A and discretizing them at multiple levels, an over-complete dictionary is obtained

$$\Delta = \{\gamma(s;\mathbf{b}_k) : \forall \mathbf{b};\ a\gamma_0,\ a > 0\}.$$

A special shapelet γ_0 is defined as an ellipse.

Shapelets are the building blocks for shape contours, and they form closed curves by linear addition:

$$\Gamma(s) = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \sum_{k=1}^{K} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \psi(s;\mu_k,\sigma_k) + \mathbf{n}(s)$$

Here (x_0, y_0) is the centroid of the contour and $\mathbf{n}$ is residue.

A discrete representation $\mathbf{B} = (K, \mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K)$, shown by the dots in second row of Figure 42, represents a shape. $\mathbf{B}$ is called the "shape script" by analogy to music scripts, where each shapelet is represented by a dot in the (μ,σ) domain. The horizontal axis is $\mu \in [0,1]$ and the vertical axis is the σ. Large dots correspond to big coefficient matrix

$$A_k^2 = a_k^2 + b_k^2 + c_k^2 + d_k^2$$

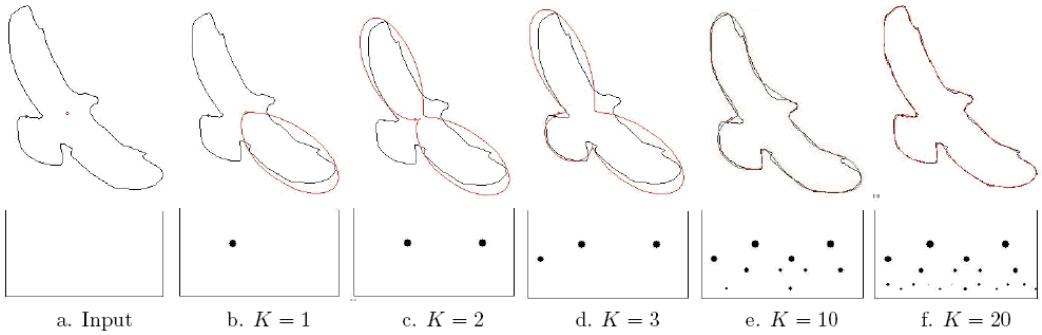


Figure 42: Pursuit of shape bases for an eagle contour

Clearly, computing the shape script $\mathbf{B}$ is a non-trivial task, since Δ is over-complete and there will be multiple sets of bases that reconstruct the curve with equal precision. [66] gave some pursuit algorithms to use shapelets representing a shape.

8.7 Discussions

As a kind of global shape description technique, shape analysis in transform domains takes the whole shape as the shape representation. The description scheme is designed for this representation. Unlike the spacial interrelation feature analysis, shape transform projects a shape contour or region into an other domain to obtain some of its intrinsic features. For shape description, there is always a trade-off between accuracy and efficiency. On 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. For a shape transform analysis algorithm, it is very flexible to accomplish a shape description with different accuracy and efficiency by choosing the number of transform coefficients.

9 Summary table

For convenience to compare these shape feature extraction approaches in this chapter, we summarize their properties in Table 1.

Frankly speaking, it is not equitable to affirm a property of an approach by rudely speaking "good" or "bad". Because certain approaches have great different performances under different conditions. For example, the method area function is invariant with affine transform under the condition of the contours sampled at its same vertices; whereas it is not robust to affine transform if the condition can't be contented. In addition, some approaches have good properties for certain type shapes; however it is not for the others. For example, the method shapelets representation is especially suitable for blobby objects, and it has shortcomings in representing elongated objects. So the simple evaluations in this table are only as a reference. These evaluations are drawn by assuming that all the necessary conditions have been contented for each approach.

10 Conclusion

In this chapter we made a study and a comparison the methods of shape-based feature extraction and representation. About 40 techniques for extraction of shape features have been shortly described and compared. Unlike the traditional classification, the approaches of shape-based feature extraction and representation were classified by their processing approaches. These processing approaches included shape signatures, polygonal approximation methods, spatial interrelation feature, moments approaches, scale-space methods and shape transform domains: in such way, one can easily select the appropriate processing approach. A synthetic table has been established for a fast and global comparison of the performances of these approaches.

Extracting a shape feature in accordance with human perception is not an easy task. Due to the fact that human vision and perception are an extraordinary complicated system, it is a utopia to hope that the machine vision has super excellent performance with small complexity. In addition, choosing appropriate features for a shape recognition system must consider what kinds of features are suitable for the task. There exists no general feature which would work best for every kind of images.

Shape representation			Recon-struc-ture	Invariance				Resistance			Computational complexity
				Translation	Rotation	Scale	Affine transform	Noise	Occultation	Non-rigid deformation	
Shape signatures	Complex coordinates		Yes	Bad	Bad	Bad	Bad	Average	Good	Bad	Low
	Central distance		No	Good	Good	Good	Bad	Average	Good	Bad	Low
	Tangent angle		No	Good	Good	Good	Bad	Bad	Good	Average	Low
	Curvature function		No	Good	Good	Good	Bad	Bad	Good	Average	Low
	Area function		No	Good	Good	Good	Good	Good	Good	Bad	Low
	Triangle-area representation		No	Good	Good	Good	Good	Good	Average	Average	Low
	Chord length function		No	Good	Good	Good	Bad	Bad	Bad	Bad	Low
Polygonal approximation	Merging	Distance threshold	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
		Tunneling	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
		Polygon evolution	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
	Splitting		No	Good	Good	Good	Bad	Good	Bad	Bad	Average
Space interrelation Feature	Adaptive grid resolution		Yes	Good	Good	Good	Bad	Good	Good	Bad	Low
	Bounding box		Yes	Good	Good	Good	Average	Good	Good	Average	Average
	Convex hull		No	Good	Good	Good	Good	Average	Bad	Bad	High
	Chain code	Basic chain code	Yes	Good	Bad	Bad	Bad	Bad	Good	Bad	Low
		Vertex chain code	Yes	Good	Bad	Bad	Bad	Bad	Good	Bad	Low
		Statistic chain code	No	Good	Bad	Bad	Bad	Bad	Bad	Bad	Low
	Smooth curve decomposition		No	Good	Good	Good	Bad	Good	Good	Average	Average
	ALI-based representation		No	Good	Good	Good	Average	Good	Average	Bad	Average
	Beam angle statistics		No	Good	Good	Good	Bad	Good	Bad	Bad	Low
	Shape matrix	Square model	Yes	Good	Good	Good	Bad	Bad	Good	Bad	Average
		Polar model	Yes	Good	Good	Good	Bad	Bad	Good	Bad	Low
	Shape context		No	Good	Good	Good	Bad	Bad	Average	Average	Average
	Chord distribution		No	Good	Good	Good	Bad	Good	Bad	Bad	Low
	Shock graphs		Yes	Good	Good	Good	Good	Good	Good	Good	High
Moments	Boundary moments		No	Good	Good	Good	Bad	Average	Bad	Bad	Low
	Region moments	Invariant moments	No	Good	Good	Good	Bad	Bad	Bad	Bad	Average
		Algebraic Moment	No	Good	Good	Good	Good	Average	Bad	Bad	Average
		Zernike Moments	No	Good	Good	Good	Bad	Good	Average	Average	High
		Radial Chebyshev Moments	No	Good	Good	Good	Bad	Good	Average	Average	High
Scale-space methods	Curvature scale space		No	Good	Good	Good	Average	Good	Good	Average	Average
	Intersection points map		No	Good	Good	Good	Average	Good	Good	Bad	Average
Shape transform domains	Fourier descriptors	1-D Fourier descriptor	No	Good	Good	Good	Bad	Bad	Bad	Bad	Average
		Region-based Fourier descriptor	No	Good	Good	Good	Good	Good	Average	Average	High
	Wavelet transform		No	Good	Good	Good	Good	Average	Average	Bad	Average
	Angular radial transformation		No	Good	Good	Good	Bad	Good	Bad	Bad	High
	Signature harmonic embedding		No	Good	Good	Good	Average	Good	Average	Bad	High
	$\Re$ -Transform		No	Good	Good	Good	Bad	Good	Average	Average	High
	Shapelets descriptor		No	Good	Good	Good	Bad	Good	Bad	Bad	High

Table 1: Properties of shape feature extraction approaches

References

[1] "MPEG-7 overview (version 10)," ISO/IEC JTC1/SC29/WG11, Tech. Rep., 2004.

[2] B. Horn, *Robot Vision*. MIT Press, Cambridge, 1986.

[3] D.-M. Tsai and M. fong Chen, "Object recognition by a linear weight classifier," *Pattern Recognition Letters*, vol. 16, pp. 591–600, 1995.

[4] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31(8), pp. 983–1001, 1998.

[5] I. Young, J. Walker, and J. Bowie, "An analysis technique for biological shape," *Comput. Graphics Image Processing*, vol. 25, pp. 357–370, 1974.

[6] M. Peura and J. Iivarinen, "Efficiency of simple shape descriptors," in *Proc. 3rd International Workshop on Visual Form (IWVF3)*, May 1997.

[7] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, pp. 1–19, 2004.

[8] C. Cheng, W. Liu, and H. Zhang, "Image retrieval based on region shape similarity," in *Proc. 13th SPIE symposium on Electronic Imaging, Storage and Retrieval for Image and Video Databases*, 2001.

[9] A. SOFFER, "Negative shape features for image databases consisting of geographic symbols," in *Proc. 3rd International Workshop on Visual Form*, May 1997.

[10] D. Zhang and G. Lu, "A comparative study of fourier descriptors for shape representation and retrieval," in *Proc. 5th Asian Conference on Computer Vision*, 2002.

[11] H. Kauppinen, T. Seppanen, and M. Pietikainen, "An experimental comparison of auto-regressive and fourier-based descriptors in 2-D shape classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17(2), pp. 201–207, 1995.

[12] R. B. Yadava, N. K. Nishchala, A. K. Gupta, and V. K. Rastogi, "Retrieval and classification of shape-based objects using fourier, generic fourier, and wavelet-fourier descriptors technique: A comparative study," *Optics and Lasers in Engineering*, vol. 45(6), pp. 695–708, 2007.

[13] D. S. Zhang and G. Lu, "A comparative study on shape retrieval using fourier descriptors with different shape signatures," in *Proc. International Conference on Intelligent Multimedia and Distance Education (ICIMADE01)*, 2001.

[14] C. T. Zahn and R. Z. Roskies., "Fourier descriptors for plane closed curves," *IEEE Trans. Computer*, vol. c-21(3), pp. 269–281, 1972.

[15] K.-J. Lu and S. Kota, "Compliant mechanism synthesis for shape-change applications: Preliminary results," in *Proceedings of SPIE Modeling, Signal Processing, and Control Conference*, vol. 4693, March 2002, pp. 161–172.

[16] L. J. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22(10), pp. 1185–1190, 2000.

[17] Y. P. Wang, K. T. Lee, and K.Toraichi, "Multiscale curvature-based shape representation using B-spline wavelets," *IEEE Trans. Image Process*, vol. 8(10), pp. 1586–1592, 1999.

[18] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14(8), pp. 789–805, 1992.

[19] A. C. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink, "Shape representation and recognition through morphological curvature scale spaces," *IEEE Trans. Image Processing*, vol. 15(2), pp. 331–341, 2006.

[20] N. Alajlan, M. S. Kamel, and G. Freeman, "Multi-object image retrieval based on shape and topology," *Signal Processing: Image Communication*, vol. 21, pp. 904–918, 2006.

[21] N. Alajlan, I. E. Rube, M. S. Kamel, and G. Freeman, "Shape retrieval using triangle-area representation and dynamic space warping," *Pattern Recognition*, vol. 40(7), pp. 1911–1920, 2007.

[22] S. Han and S. Yang, "An invariant feature representation for shape retrieval," in *Proc. Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2005.

[23] L. J. Latecki and R. Lakamper, "Convexity rule for shape decomposition based on discrete contour evolution," *Computer Vision and Image Understanding*, vol. 73(3), pp. 441–454, 1999.

[24] K. Chakrabarti, M. Binderberger, K. Porkaew, and S. Mehrotra, "Similar shape retrieval in mars," in *Proc. IEEE International Conference on Multimedia and Expo*, 2000.

[25] C. Bauckhage and J. K. Tsotsos, "Bounding box splitting for robust shape classification," in *Proc. IEEE International Conference on Image Processing*, 2005, pp. 478–481.

[26] E. Davies, *Machine Vision: Theory, Algorithms, Practicalities.* Academic Press, New York, 1997.

[27] R. Gonzalez and R. Woods, *Digital image processing, Second Edition.* PEARSON EDUCATION NORTH ASIA LIMITED and Publishing House of Electronics Industry, 2002.

[28] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision.* Chapman and Hall, London, UK, 1993.

[29] Y. K. Liu, W. Wei, P. J. Wang, and B. Zalik, "Compressed vertex chain codes," *Pattern Recognition*, vol. 40(11), pp. 2908–2913, 2007.

[30] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Information Theory*, vol. IT-8, pp. 179–187, 1962.

[31] J. Iivarinen and A. Visa, "Shape recognition of irregular objects," in *Proc. SPIE, Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, vol. 2904, 1996, pp. 25–32.

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

[33] D. Guru and H. Nagendraswam, "Symbolic representation of two-dimensional shapes," *Pattern Recognition Letters*, vol. 28, pp. 144–155, 2007.

[34] N. Arica and F. Vural, "BAS: a perceptual shape descriptor based on the beam angle statistics," *Pattern Recognition Letters*, vol. 24(9-10), 2003.

[35] J. Flusser, "Invariant shape description and measure of object similarity," in *Proc. 4th International Conference on Image Processing and its Applications*, 1992, pp. 139–142.

[36] A. Taza and C. Suen, "Discrimination of planar shapes using shape matrices," *IEEE Trans. System, Man, and Cybernetics*, vol. 19(5), pp. 1281–1289, 1989.

[37] G. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content based image retrieval," *ACM Multimedia System Journal*, vol. 7(2), pp. 165–174, 1999.

[38] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape context," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24(4), pp. 509–522, 2002.

[39] G. Mori and J. Malik, "Estimating human body configurations using shape context matching," in *Proc. 7th European Conference on Computer Vision*, vol. III, 2002, pp. 666–680.

[40] H. Zhang and J. Malik, "Learning a discriminative classifier using shape context distances," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[41] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[42] S. P. Smith and A. K. Jain, "Chord distribution for shape matching," *Computer Graphics and Image Processing*, vol. 20, pp. 259–271, 1982.

[43] K. Siddiqi and B. Kimia, "A shock grammar for recognition," in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, June 1996, pp. 507–513.

[44] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26(5), pp. 550–571, 2004.

[45] M. E. Celebi and Y. A. Aslandogan, "A comparative study of three moment-based shape descriptors," in *Proc. of the International Conference of Information Technology: Coding and Computing*, 2005, pp. 788–793.

[46] G. Taubin and D. Cooper, "Recognition and positioning of rigid objects using algebraic moment invariants," in *SPIE Conference on Geometric Methods in Computer Vision*, vol. 1570, 1991, pp. 175–186.

[47] R. Mukundan, S. Ong, and P. Lee, "Image analysis by tchebichef moments," *IEEE Trans. Image Processing*, vol. 10(9), pp. 1357–1364, 2001.

[48] R. Mukundan, "A new class of rotational invariants using discrete orthogonal moments," in *Sixth IASTED International Conference on Signal and Image Processing*, 2004, pp. 80–84.

[49] K. Jin, M. Cao, S. Kong, and Y. Lu, "Homocentric polar-radius moment for shape classification," in *Proc. Signal Processing, The 8th International Conference on*, 2006.

[50] C. Kan and M. D. Srinath, "Invariant character recognition with Zernike and orthogonal Fourier-Mellin moments," *Pattern Recognition*, vol. 35, pp. 143–154, 2002.

[51] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee, "Shape measures for content based image retrieval: A comparison," *Pattern Recognition*, vol. 33(3), pp. 319–337, 1997.

[52] J. Peng, W. Yang, and Z. Cao, "A symbolic representation for shape retrieval in curvature scale space," in *Proc. International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, 2006.

[53] D. Zhang and G. Lu, "A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval," *Visual Communication and Image Representation*, vol. 14(1), 2003.

[54] S. Abbasi, F. Mokhtarian, and J. Kittler, "Enhancing css-based shape retrieval for objects with shallow concavities," *Image and Vision Computing*, vol. 18(3), pp. 199–211, 2000.

[55] M. Yang, K. Kpalma, and J. Ronsin, "Scale-controlled area difference shape descriptor," in *Proc. SPIE, Electronic Imaging science and Technology*, vol. 6500, 2007.

[56] K. Kpalma and J. Ronsin, "Multiscale contour description for pattern recognition," *Pattern Recognition Letters*, vol. 27, pp. 1545–1559, 2006.

[57] D. S. Zhang and G. Lu, "Enhanced generic fourier descriptors for object-based image retrieval," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002)*, pp. 3668–3671.

[58] G. C.-H. Chuang and C.-C. J. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Trans. Image Processing*, vol. 5(1), pp. 56–70, 1996.

[59] M. Khalil and M. Bayoumi, "A dyadic wavelet affine invariant function for 2D shape recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25(10), pp. 1152–1164, 2001.

[60] G. Chen and T. D. Bui, "Invariant fourier-wavelet descriptor for pattern recognition," *Pattern Recognition*, vol. 32, pp. 1083–1088, 1999.

[61] J. Ricard, D. Coeurjolly, and A. Baskurt, "Generalizations of angular radial transform for 2D and 3D shape retrieval," *Pattern Recognition Letters*, vol. 26(14), 2005.

[62] S.-M. Lee, A. L. Abbott, N. A. Clark, and P. A. Araman, "A shape representation for planar curves by shape signature harmonic embedding," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[63] S. Tabbone, L. Wendling, and J.-P. Salmon, "A new shape descriptor defined on the radon transform," *Computer Vision and Image Understanding*, vol. 102(1), pp. 42–51, 2006.

[64] G. Borgefors, "Distance transformations in digital images," in *Computer Vision, Graphics, and Image Processing*, June 1986, pp. 344–371.

[65] G. S. di Baja and E. Thiel, "Skeletonization algorithm running on path-based distance maps," *Image Vision Computer*, vol. 14, pp. 47–57, 1996.

[66] A. Dubinskiy and S. C. Zhu, "A multi-scale generative model for animate shapes and parts," in *Proc. Ninth IEEE International Conference on Computer Vision (ICCV)*, 2003.