2014555057 Derya ŞAHİN
2015555038 İsa İMADOĞLU
2014555053 Okan Can SAĞLAM

SHAPE DESCRIPTORS

The different features related to images are texture, shape, color etc. Images when represented by shape compared to the other features, is much more effective in characterizing the content of an image **[1]**. Though, the difficult task of shape descriptors is the precise extraction and representation of shape information. The construction of shape descriptors is even more complex when invariance, with respect to a number of possible transformations, like scaling, shifting and rotation, is required. The overall performance of shape descriptors can be divided into qualitative and quantitative performances. The qualitative characteristics involve their retrieval performance based on the captured shape details for representation. Efficient shape features must have some essential properties such as **[2]**:

- **Identifiability:** Shapes which are found perceptually similar by human have the same features that are different from the others.
- **Translation, Rotation and Scale Invariance:** The location, the rotation and the scaling changing of the shape must not affect the extracted features.
- **Affine Invariance:** The affine transform performs a linear mapping from coordinates system to other coordinates system 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.
- **Reliability:** As long as one deals with the same pattern, the extracted features must remain the same.

Table in Figure 1 shows that classification of shape representation and description techniques**[3].**

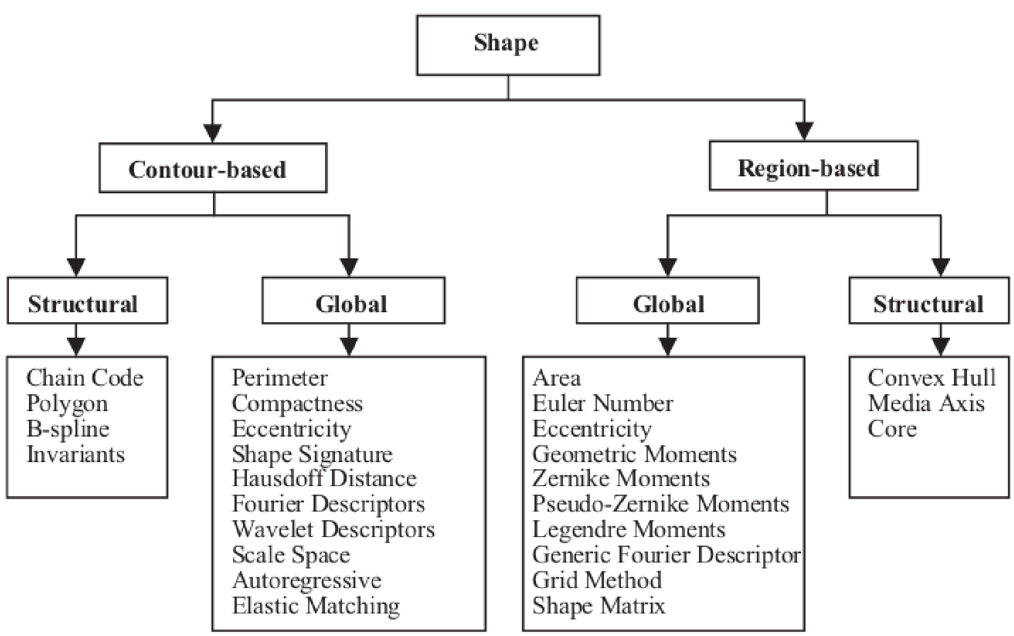
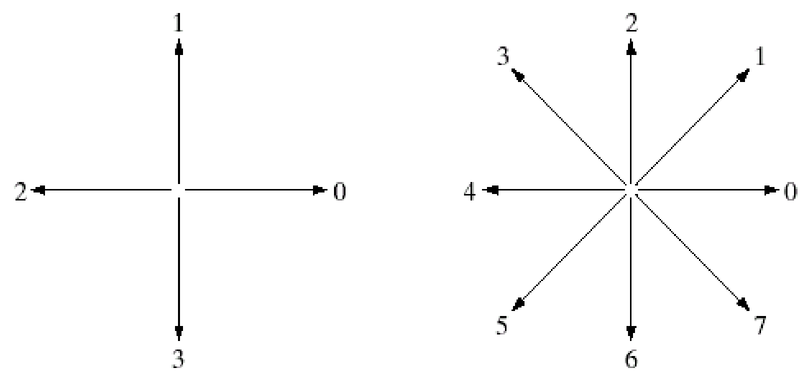


Fig. 1. Classification of shape representation and description techniques.

- **Chain Codes**

Chain codes are used to represent a boundary by a connected sequence of straight line segments of specified length and direction **[4]**. A boundary code formed using a sequence of such directional numbers is referred to as a Freeman chain code. In Figure 11.1 **[5][6] .**



a b

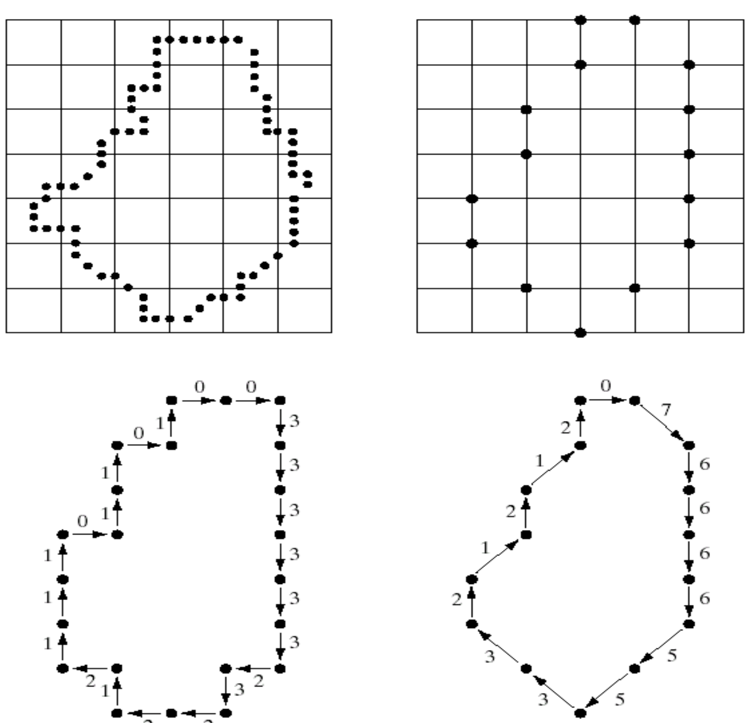
FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

This method generally is unacceptable to be applied for each pair of consecutive pixels:

(a) The resulting chain of codes usually is quite long;

(b) Sensitive to noise: any small disturbances along the boundary due to noise or imperfect segmentation cause changes in the code that may not necessarily be related to the shape of the boundary.

A frequently used method to solve the problem is to resample the boundary by selecting larger grid spacing. A boundary point is assigned to each node of the large grid, depending on the proximity of the original boundary to that node. The accuracy of the resulting code representation depends on the spacing of the sampling grid **[5]**.



a b
c d

FIGURE 11.2 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 4-directional chain code. (d) 8-directional chain code.

Normalization for starting point **[6]***:*
Treat the code as a circular sequence and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude.

Normalization for rotation:
Use the first difference of the chain code instead of the code itself. The difference is simply by counting counterclockwise the number of directions that separate two adjacent elements of the code. There is an example in Figure 11.11 **[5].**

Example: The first difference of the 4-direction chain code 10103322 is 33133030.

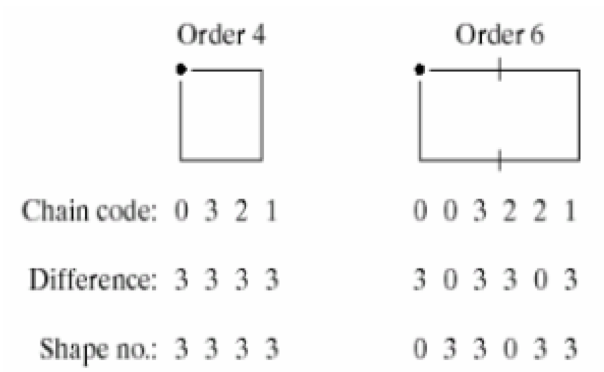
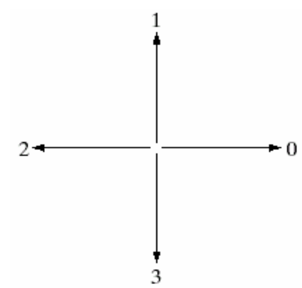
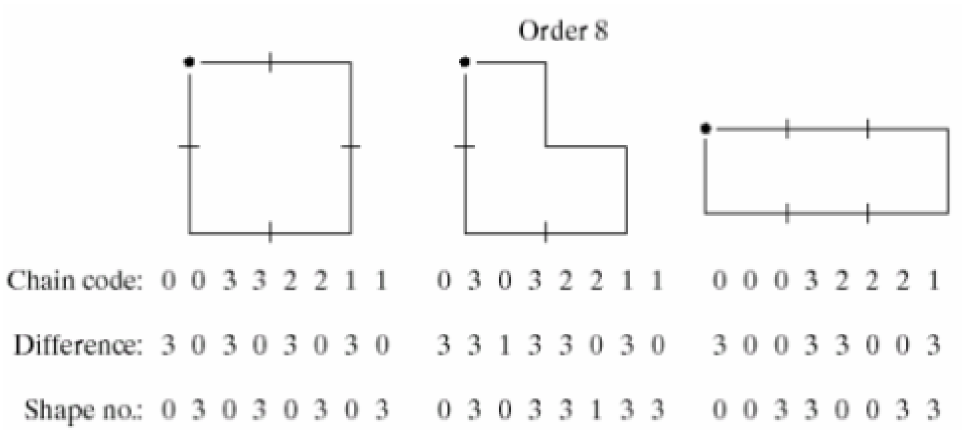


FIGURE 11.11 All shapes of order 4, 6, and 8. The directions are from Fig. 11.1(a), and the dot indicates the starting point.





Chain Code Histogram
The chain code histogram (CCH) is meant to group together objects that look similar to a human observer. It is not meant for exact detection and classification tasks. The CCH is calculated from the chain code presentation of a contour. A simple example is depicted in Figure 1. In Figure 1 (a) are the directions of the eight connected chain code. In Figure 1 (b) is a sample object, a square. The starting

point for the chain coding is marked with a black circle, and the chain coding direction is clockwise. In Figures 1 (c)-(d) are the chain code and the CCH of the contour of the square **[7]**.

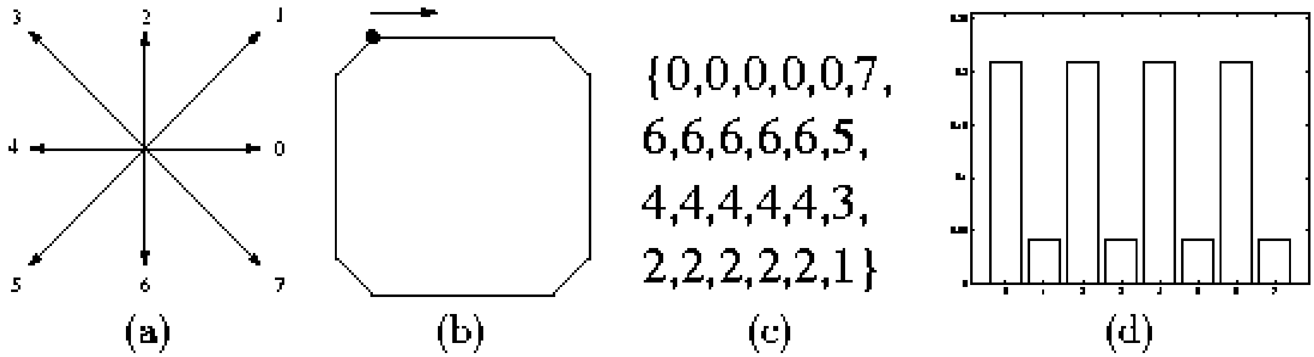


Figure 1: (a) The directions of the eight connected chain code (K=8). (b) A sample object, a square, (c) chain code presentation of the square, and (d) the chain code histogram of the square.

Simple Shape Descriptors [6].

- **Perimeter :** The perimeter of a region is the length of its boundary.
- **Area :** The area of a region is defined as the number of pixels in the region.
- **Compactness :** is often defined as the ratio of squared perimeter and the area of an object. It reaches the minimum in a circular object and approaches infinity in thin, complex objects. The measure applied in this paper is the ratio of the perimeter of a circle with equal area as the original object and the original perimeter. There are examples in Figure 6.25 and Figure 8. There is compactness formula in Figure 7.

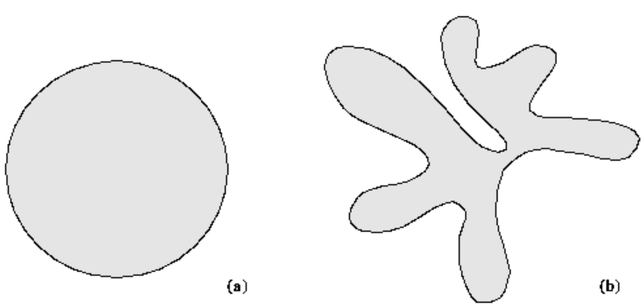


Figure 6.25 *Compactness: (a) Compact, (b) non-compact.*

$$\gamma = \frac{(perimeter)^2}{4\pi(area)}$$

Figure 7

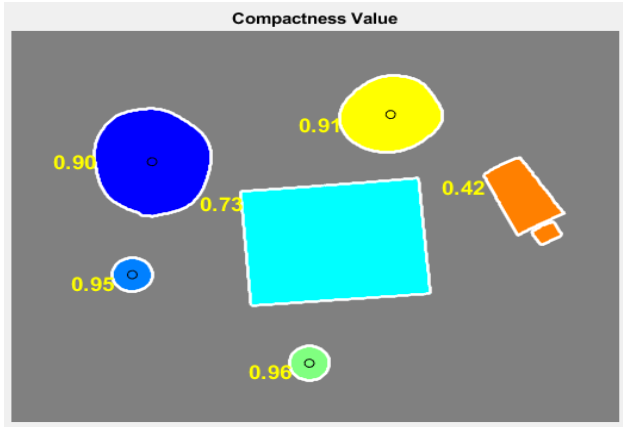


Figure 8

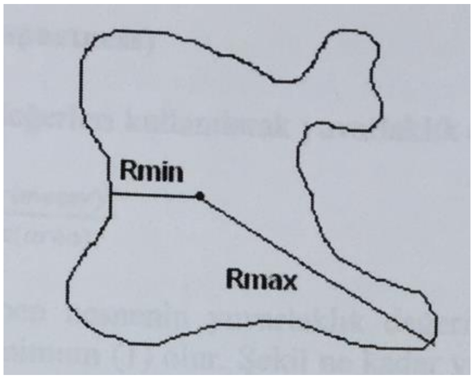
```
r1 = imread('yuvarlak.png');
r2=rgb2gray(r1);
r3 = imbinarize(r2);
r3 = bwareaopen(r3,30);
r3=imfill(r3,'holes');
[B,L] = bwboundaries(r3);
figure,imshow(label2rgb(L,@jet, [.5 .5 .5]));
hold on;
for k=1:length(B)
   sinir = B{k};
   plot(sinir(:,2),sinir(:,1),'w','LineWidth',2);
end
stats = regionprops(L,'Area','Centroid');
alanThreshold = 0.89;
for k=1:length(B)
    sinir = B{k};
    delta_sq = diff(sinir).^2;
    cevre = sum(sqrt(sum(delta_sq,2)));
    alan = stats(k).Area;
    yOrani = 4*pi*alan/cevre^2;
    yOraniString = sprintf('%2.2f',yOrani);
    if yOrani > alanThreshold
    merkez = stats(k).Centroid;
    plot(merkez(1),merkez(2),'ko');
    end
    text(sinir(1,2)-35,sinir(1,1)+13,yOraniString,'Color','y','FontSize',14,'FontWeight','bold');

end
title(['Compactness Value']);
```

Figure 9

In Figure 9, there is a code of Figure 8.

- **Eccentricity :** The simplest is the ratio of major and minor axes of an object in Figure 10.



Eccentricity = Rmax / Rmin

Figure 10

- **Region-Based Eccentricity :** How much a conic section (a circle, ellipse, parabola or hyperbola) varies from being circular. A circle has an **eccentricity of zero**, so the eccentricity shows you how "un-circular" the curve is. Bigger eccentricities are less curved **[8]**.
  Different values of eccentricity make different curves in Figure 11 and Figure 12:

➔ At eccentricity = 0 we get a circle
➔ for 0 < eccentricity < 1 we get an ellipse
➔ for eccentricity = 1 we get a parabola
➔ for eccentricity > 1 we get a hyperbola
➔ for infinite eccentricity we get a line

conic section	equation	eccentricity (e)
Circle	$x^2 + y^2 = r^2$	0
Ellipse	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	$\sqrt{1 - \frac{b^2}{a^2}}$
Parabola	$x^2 = 4ay$	1
Hyperbola	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$\sqrt{1 + \frac{b^2}{a^2}}$

Figure 11

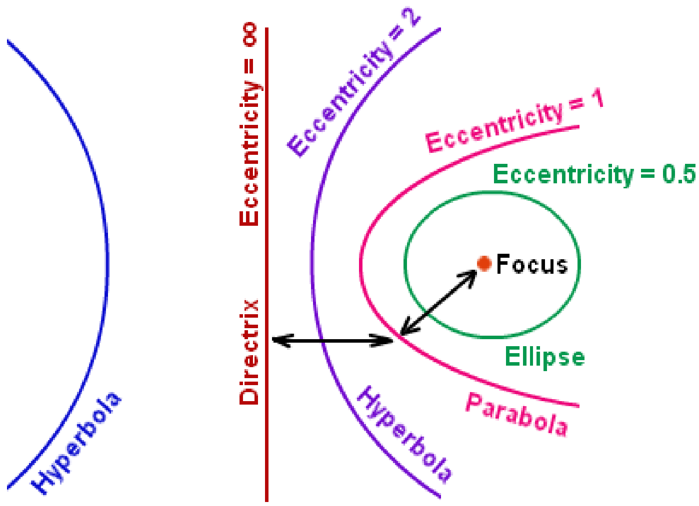


Figure 12

- **Curvature :** is defined as the rate of change of slope in Figure 13.

$$|K(t)|^2 = \Delta(\frac{d^2y}{dt^2})^2 + \Delta(\frac{d^2x}{dt^2})^2$$

Figure 13

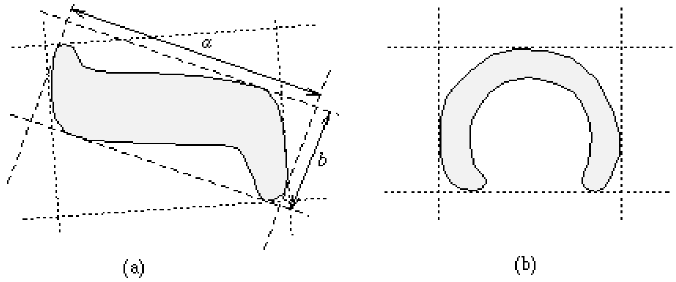
- **Bending Energy :** A shape can be represented by its bending energy represented by

$$E = \frac{1}{T}\int_0^T |K(t)|^2 dt$$

Figure 14

where K(t) is the curvature function, t is the arc length parameter, and T is the total curve length in Figure 14.

- **Elongatedness :** A ratio between the length and width of the region bounding rectangle Figure 15. This criterion cannot succeed in curved regions, for which the evaluation of elongatedness must be based on maximum region thickness **[9]**. Elongatedness can be evaluated as a ratio of the region area and the square of its thickness. The maximum region thickness (holes must be filled if present) can be determined as the number d of erosion steps that may be applied before the region totally disappears in Figure 6.24.



$$elongatedness = \frac{area}{(2d)^2}$$

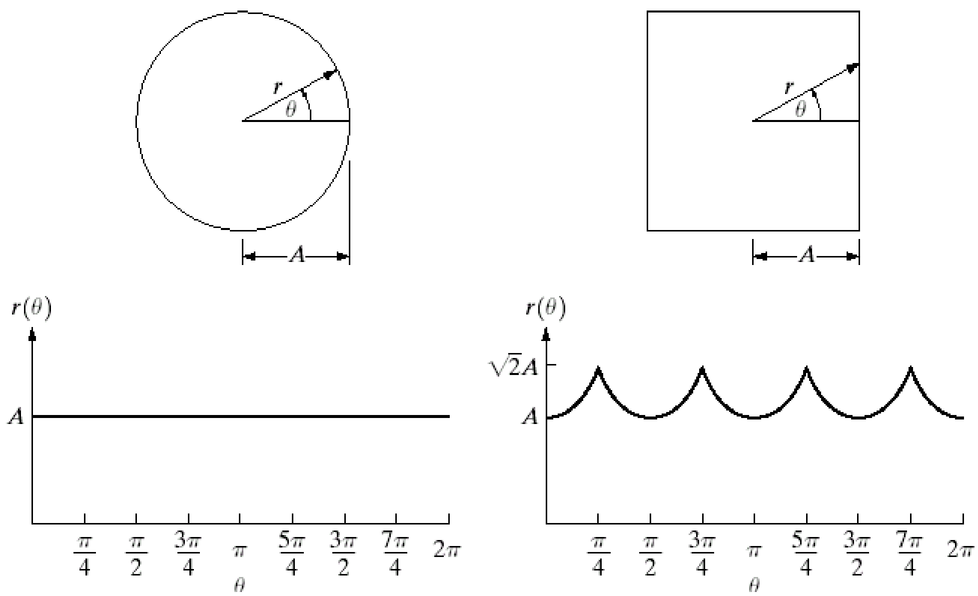
Figure 15

Figure 6.24 *Elongatedness: (a) Bounding rectangle gives acceptable results, (b) bounding rectangle cannot represent elongatedness.*

- **Signature :** Shape signatures are usually normalized into being translation and scale invariant. In order to compensate for orientation changes, shift matching is needed to and the best matching between two shapes **[3]**. There is an example in Figure 11.5 and Fig. 2.

a b

FIGURE 11.5 Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A\sec\theta$ for $0 \leq \theta \leq \pi/4$ and $r(\theta) = A\csc\theta$ for $\pi/4 < \theta \leq \pi/2$.



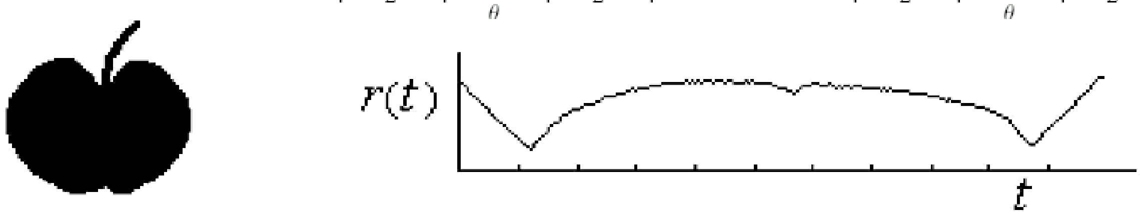


Fig. 2. An apple shape and its centroid distance function

- **Fourier Descriptors :** Fourier descriptors have characteristics, like simple derivation, simple normalization and its robustness to noise, which have made them very popular in a wide range of applications. Fourier descriptors are obtained by applying Fourier transform to a shape signature. The Fourier transform on a complex vector derived from the shape boundary coordinates gives us the Fourier descriptors **[2].** There are formulas in Figure 16 and an example in Figure 17 **[5].**

Let s(k) be a coordinate of a boundary point k :

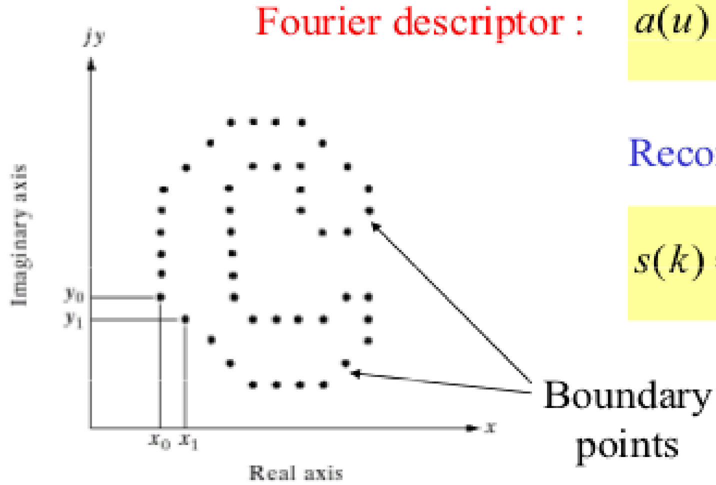
$$s(k) = x(k) + jy(k)$$

Fourier descriptor :

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-2\pi uk/K}$$

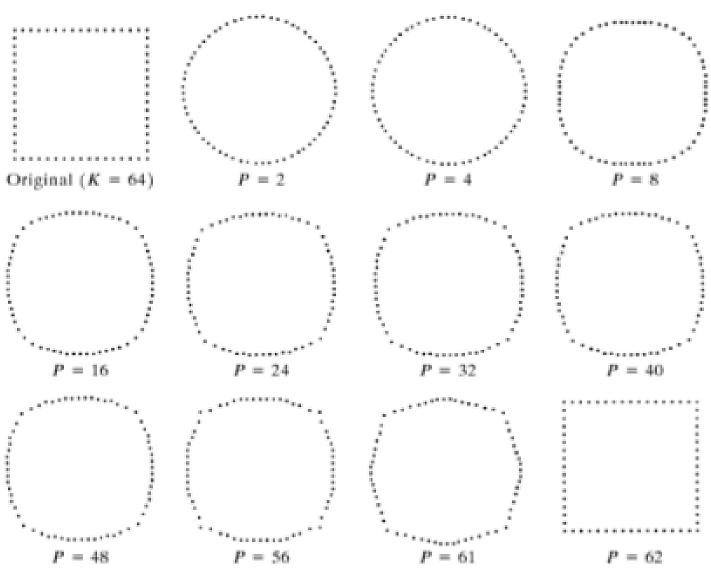
Reconstruction formula

$$s(k) = \frac{1}{K}\sum_{k=0}^{K-1} a(u)e^{2\pi uk/K}$$



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Figure 16



$$\hat{s}(k) = \frac{1}{K}\sum_{k=0}^{P-1} a(u)e^{2\pi uk/K}$$

P is the number of Fourier coefficients used to reconstruct the boundary

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Figure 17

- **Moment Descriptors**

Moments and the related invariants have been extensively analyzed to characterize the patterns in images in a variety of applications. The well-known moments include geometric moments, zernike moments, rotational moments, and complex moments.

Moment invariants are firstly introduced by Hu**[10]**., Hu derived six absolute orthogonal invariants and one skew orthogonal invariant based upon algebraic invariants, which are not only independent of position, size and orientation but also independent of parallel projection. The moment invariants have been proved to be the adequate measures for tracing image patterns regarding the images translation, scaling and rotation under the assumption of images with continuous functions and noise-free.

Two-dimensional (p+q)th order moment are defined as follows:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy \qquad \mathrm{p,q} = 0,1,2,\cdots \tag{1}$$

If the image function f(x,y) is a piecewise continuous bounded function, the moments of all orders exist and the moment sequence $\{m_{pq}\}$ is uniquely determined by f(x,y); and correspondingly, f(x,y) is also uniquely determined by the moment sequence $\{m_{pq}\}$.

One should note that the moments in (1) may be not invariant when f(x,y) changes by translating, rotating or scaling. The invariant features can be achieved using central moments, which are defined as follows:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy \qquad p,q = 0,1,2,\cdots$$

Where

$$\bar{x} = \frac{m_{10}}{m_{00}} \qquad \bar{y} = \frac{m_{01}}{m_{00}}$$

The pixel point (x , y) are the centroid of the image f(x,y).

The centroid moments μ_{pq} computed using the centroid of the image f(x,y) is equivalent to the m_{pq} whose center has been shifted to centroid of the image. Therefore, the central moments are invariant to image translations.

Scale invariance can be obtained by normalization. The normalized central moments are defined as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \ \gamma = (p+q+2)/2, \ \mathrm{p+q} = 2,3,\cdots$$

Based on normalized central moments, Hu introduced seven moment invariants:

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \mu_{03})^2$$
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03)}[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

The seven moment invariants are useful properties of being unchanged under image scaling, translation and rotation.

For example when we tested this five image.

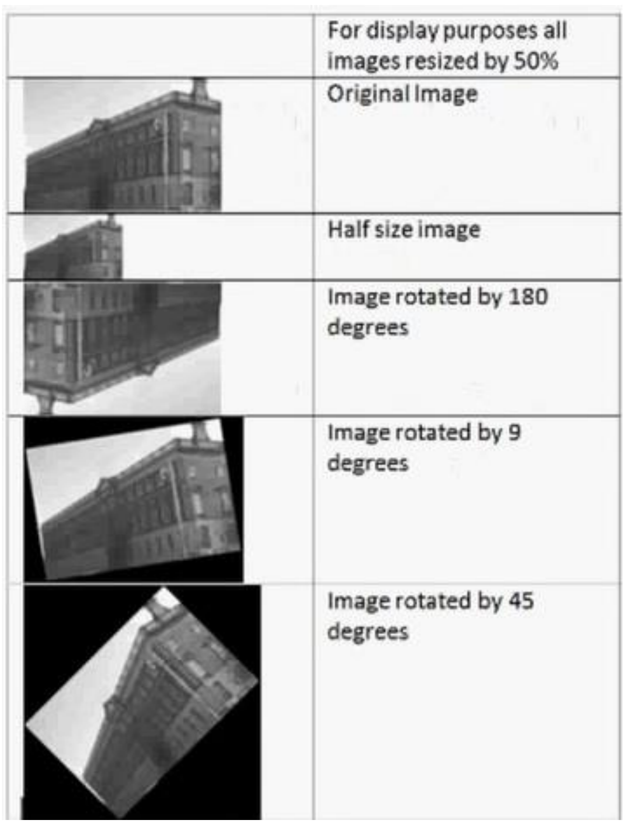


Figure 18 : Testing Images

The seven moments for each of the five image. A log transform was taken to get meaningful values**[11][12]**.

Invariant (|log|ϕ|)	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
Original image	2.862996	6.516763	11.167687	10.416997	21.311533	-15.262956	21.422099
Resized by 50%	2.862996	6.516763	11.167687	10.416997	21.311533	-15.262956	21.422099
Image degrees rotated by 180	2.862996	6.516763	11.167687	10.416997	21.311533	-15.262956	21.422099
Image rotated by 9 degrees	2.862946	6.516841	11.164342	10.417313	21.309686	-15.235646	21.421987
Image rotated by 45 degrees	2.863852	6.519252	11.159553	10.426988	21.316971	-15.174493	21.442456
Max Value - Min Value	0.000906	0.002489	0.008134	0.009991	0.007286	0.098344	0.021735

Figure 19:Result of the Hu's Invariant Moments

As we can see size and rotation don't change the Hu's invariant moments.

- **Polygonal Approximations**

Digital boundaries carry information which may be superfluous for certain applications. Boundary approximations can be sufficent in such cases. Linear piecewise(polygonal) approximations are the most frequently used.

The optimal linear piecewise approximation can ve obtained by choosing the polygon vertices in such a way that the overall approximation error is minimized.

Split techniques divide a curve segment recursively into smaller segments, until each curve can be approximated by a linear segment within an acceptable error range.

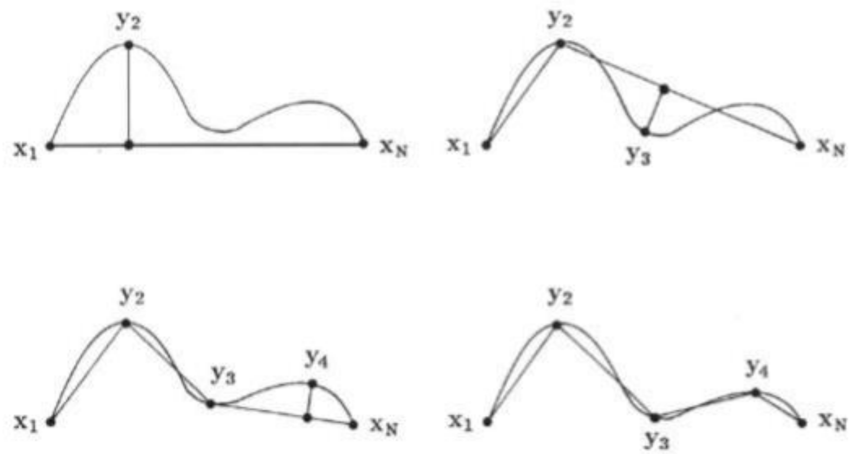


Figure 20 : Splitting method for polygonal approximations.

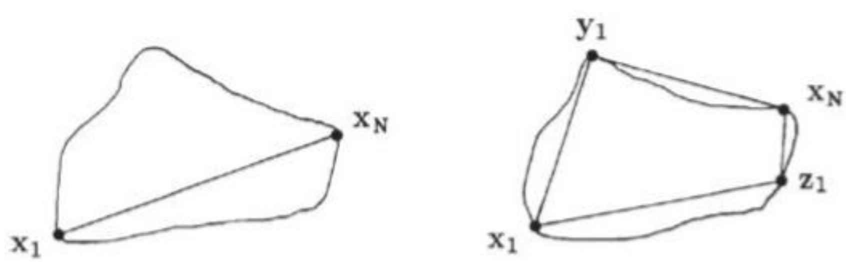


Figure 21 : Splitting method for the linear picewise approximation of a closed curve.

A basic advantage of the splitting approach is that it can be detect the inflection points on a curve and can use them in curve representation**[13]**.

Merge techniques in the polgonal approximation operate in the opposite way.The primary disadvantage of the merge algortihm is the polygon vertices do not coincide with curve inflection points.

This problem can be solved by combining split and merge techniques.

- **Pairwise Geometric Histogram (PGH)**

Although image features can be characterised to some extent by intrinsic attributes such as local image gradients and curvatures, the context of the surrounding shape geometry provides the basis for a much more powerful descriptor.**[14]** By recording the geometrical relationships between a feature and each of the surrounding shape features or primitives the feature is fully defined in terms of its shape context. By carefully selecting appropriate measures and storing these measurements in the form of a frequency histogram, a concise shape descriptor which promotes efficient and robust feature classification can be produced. This frequency histogram is referred to as a pairwise geometric histogram because it records geometric measures made between pairs of image features.

The selection of geometrical measurements with which to form a particular type of pairwise geometric histogram is motivated by two, possibly opposing, requirements. On the one hand it is desirable to make measurements which together, fully define the relationship between a pair of features, producing a unique descriptor. On the other hand, it is important to select measures with good invariance properties to promote efficient classification and which are stable under expected noise conditions to promote robustness.

The geometric relationship between a pair of line segments is well defined by the relative angle between them and the range of perpendicular distance obtained when the endpoints of the second line are projected onto the first. These relationships are depicted in Figure 3.2. Although this does not fully constrain the relationship between the pair of lines (the second line is free to translate parallel to the first) it is invariant to rotations and translations of the line pair. Importantly, these measures also exhibit stability if any of the lines become fractured which frequently occurs in real images. Entries made in the histogram for these measurements are weighted by the product of the lengths of the two line segments. This assigns an equal amount of significance to each edge pixel of the shape and ensures that fragmented entries add up correctly. Figure 3.3 (a) depicts the histogram entry made for the line pair in Figure 3.2 and Figure 3.3 (b) depicts the multiple entries made if the second line becomes broken. Clearly these representations are very similar.

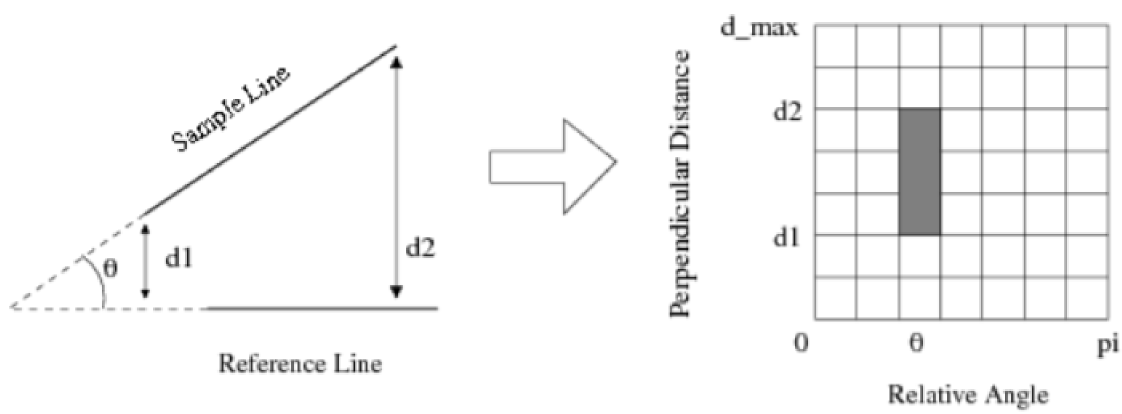


Figure 22: An image that shows Pairwise Geometric Histogram implementation.

- **Morphological Medial Axis Transform (MAT)**

Skeletonization is a process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels.**[15]** To see how this works, imagine that the foreground regions in the input binary image are made of some uniform slow-burning material. Light fires simultaneously at all points along the boundary of this region and watch the fire move into the interior. At points where the fire traveling from two different boundaries meets itself, the fire will extinguish itself and the points at which this happens form the so called `quench line'. This line is the skeleton. Under this definition it is clear that thinning produces a sort of skeleton.

Another way to think about the skeleton is as the loci of centers of bi-tangent circles that fit entirely within the foreground region being considered. Figure 1 illustrates this for a rectangular shape.

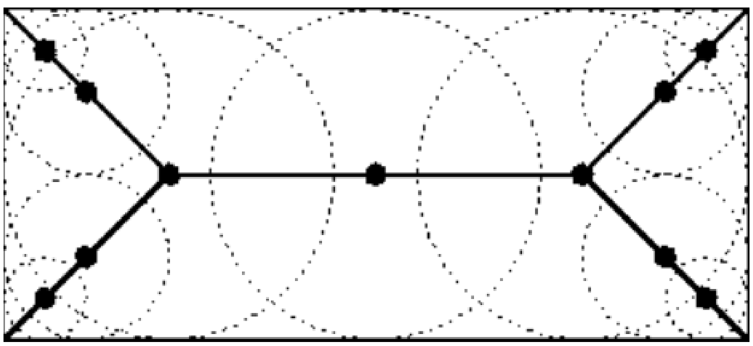


Figure 23: Skeleton of a rectangle defined in terms of bi-tangent circles.

The terms medial axis transform (MAT) and skeletonization are often used interchangeably but we will distinguish between them slightly. The skeleton is simply a binary image showing the simple skeleton. The MAT on the other hand is a grey level image where each point on the skeleton has an intensity which represents its distance to a boundary in the original object.

So how does MAT work? The skeleton/MAT can be produced in two main ways. The first is to use some kind of morphological thinning that successively erodes away pixels from the boundary (while preserving the end points of line segments) until no more thinning is possible, at which point what is left approximates the skeleton. The alternative method is to first calculate the distance transform of the image. The skeleton then lies along the *singularities* (*i.e.* creases or curvature discontinuities) in the distance transform. This latter approach is more suited to calculating the MAT since the MAT is the same as the distance transform but with all points off the skeleton suppressed to zero.

Note: The MAT is often described as being the `locus of local maxima' on the distance transform. This is not really true in any normal sense of the phrase `local maximum'. If the distance transform is displayed as a 3-D surface plot with the third dimension representing the grey value, the MAT can be imagined as the ridges on the 3-D surface.

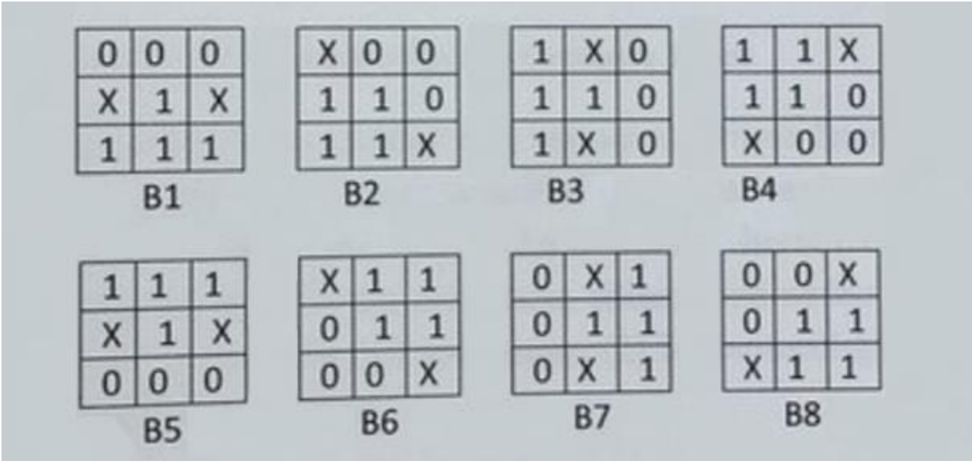


Figure 24: X= Don’t Care, 1= Active Pixel, 2= Passive Pixel

We obtain the last image as A’ by using the formula shown below:

A’ = (((((((AxB1)xB2)xB3)xB4)xB5)xB6)xB7)xB8

- **Topological Medial Axis Transform (MAT)**

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening.**[16]** It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output.

The thinning operation is related to the hit-and-miss transform, and so it is helpful to have an understanding of that operator before reading on.

How does this algorithm work? Like other morphological operators, the behaviour of the thinning operation is determined by a structuring element. The binary structuring elements used for thinning are of the extended type described under the hit-and-miss transform (*i.e.* they can contain both ones and zeros).

The thinning operation is related to the hit-and-miss transform and can be expressed quite simply in terms of it. The thinning of an image *I* by a structuring element *J* is:

thin(I, J) = I – hit_and_miss(I, J)

where the subtraction is a *logical subtraction* defined by $X - Y = X \cap \text{NOT } Y$

In everyday terms, the thinning operation is calculated by translating the origin of the structuring element to each possible pixel position in the image, and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to background (zero). Otherwise it is left unchanged. Note that the structuring element must always have a one or a blank at its origin if it is to have any effect.

The choice of structuring element determines under what situations a foreground pixel will be set to background, and hence it determines the application for the thinning operation.

We have described the effects of a single pass of a thinning operation over the image. In fact, the operator is normally applied repeatedly until it causes no further changes to the image (*i.e.* until*convergence*). Alternatively, in some applications, *e.g. pruning*, the operations may only be applied for a limited number of iterations.

Thinning is the dual of thickening, *i.e.* thickening the foreground is equivalent to thinning the background.

- **Euler Number**

Euler number describes the relation between the number of contiguous parts and the number of holes and 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:

$$Eul = S - N$$

Figure 25: Euler Numbers of per character is 1, -1 and 0, respectively.

References

[1] Various Shape Descriptors in Image Processing – Preetika D'Silva1, P. Bhuvaneswari2
[2] Shape Descriptor/Feature Extraction Techniques - Fred Park
[3] Review of shape representation and description techniques - Dengsheng Zhang∗ , Guojun Lu
[4] Digital Image Processing - Dr. ir. Aleksandra Pizurica & Prof. Dr. Ir. Wilfried Philips
[5] Digital Image Processing – Gonzalez & Woods
[6] https://profs.info.uaic.ro/~ancai/DIP/curs/DIP%20w7%202018.pdf
[7] http://www.cis.hut.fi/research/IA/paper/publications/bmvc97/node2.html
[8] https://www.mathsisfun.com/geometry/eccentricity.html
[9] http://user.engineering.uiowa.edu/~dip/LECTURE/Shape3.html#scalar
[10] H. Ming-Kuei, "Visual pattern recognition by moment invariants," Information Theory, IRE Transactions, vol. 8, pp. 179-187, 1962.
[11] M. R. Teaque, "Image Analysis via the General Theory of Moments," Journal of the Optical Society of America, vol. 70, pp. 920-930, 1980.
[12] J. F. Boyce and W. J. Hossack, " Moment Invariants for Pattern Recognition," Pattern Recognition Letters, vol. 1, pp. 451-456, 1983.
[13]http://wcours.gel.ulaval.ca/2015/a/GIF7002/default/5notes/diapositives/pdf_A15/lectures%20supplementaires/C09b.pdf
[14]https://www.researchgate.net/publication/266658084_Pairwise_Geometric_Histograms_for_Object_Recognition_Developments_and_Analysis
[15] http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm
[16] https://homepages.inf.ed.ac.uk/rbf/HIPR2/thin.htm