Vahe Karamian - Computer Science 519 - Computer Vision - Homerowk II

Fourier Transformation:

A very large class of important computational problems falls under the general rubic of Fourier transform methods or spectral methods. For some of these problems, the Fourier transform is simply an efficient computational tool for accomplishing certain common manipulations of data. In other cases, we have problems for which Fourier transform is itself of intrinsic interest. Fourier methods have revolutionized fields of science and engineering, from radio astronomy to medical imaging, from seismology to spectroscopy. Indeed. the wide application of Fourier methods must be credited principally to the existance of the fast Fourier transform. In other words, if you speed up any nontrivial algorithm by a factor of a million or so, the world will beat a path towards finding useful applications for it. The most direct application of the Fast Fourier Transform (FFT) are to the convolution or deconvolution of data, correlation and autocorrelation, optimal filtering, power spectrum estimation, and the computation of Fourier integrals.

Formula / Logic:

A physical process can be described either in the time domain, by values of some quantity h as function of time t, e.g., h(t), or else in the frequency domain, where the process is specified by giving its amplitude H (generally a complex number indicating phase also) as a function of frequency f, that is H(f), with -infinity < f < infinity. For many purposes it is useful to think of h(t) and H(f) as being two different representations of the same funtion. One goes back and forth between these two representations by means of the Fourier transformation equations,

$$H(f) = \int_{-\infty}^{\infty} h(t) \cdot e^{2 \cdot \pi \cdot i \cdot f \cdot t} dt$$

$$h(t) = \int_{-\infty}^{\infty} H(f) \cdot e^{-2 \cdot \pi \cdot i \cdot f \cdot t} dt$$

If t is measured in secondes, then f in the equation is in cycler per second, or Herts (the unit of frequency). However, the equations work with other units as well. If h is a funtion of position x (in meters), H will be a function of inverse wavelength (cycles per meter), and so on. Physicists and mathematicians are more used to using angular frequency ω, which is given in radians per sec.

The relation between ω and f, H(ω) and H(f) is $\omega \equiv 2 \cdot \pi \cdot f$ $\quad H(\omega) \equiv H(f)_{f = \frac{\omega}{2 \cdot \pi}}$

and therefore our previous equation would look like this

$$H(\omega) = \int_{-\infty}^{\infty} h(t) \cdot e^{2 \cdot i \cdot \omega \cdot t} dt$$

$$h(t) = \int_{-\infty}^{\infty} H(\omega) \cdot e^{-i \cdot \omega \cdot t} d\omega$$

In the time domain function, function h(t) may happen to have one or more special symmetries. It might be purly real or purly imaginary or it might be even, h(t) = h(-t), or odd, h(t) = -h(-t). In the frequency domain, these symmetries lead to relationships between H(f) and H(-f). The followinf table gives the correspondence between symmetries in the two domains:

If ...	then ...
h(t) is real	H(-f) = [H(f)]*
h(t) is imaginary	H(-f) = -[H(f)]*
h(t) is even	H(-f) = H(f) [i.e., H(f) is even]
h(t) is odd	H(-f) = -H(f) [i.e., H(f) is odd]
h(t) is real and even	H(f) is real and even
h(t) is real and odd	H(f) is imaginary and odd
h(t) is imaginary and even	H(f) is imaginary and even
h(t) is imaginary and odd	H(f) is real and odd

In image processing, and image is a function of two parameters in a plane. One possible way to investigate its properties is to decompose the image function using a linear combination of orthonormal functions. The Fourier transformation uses harmonic functions for the decomposition. In image processing the two-dimensional Fourier transform is defined by the integral

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\cdot e^{-2\cdot\pi\cdot i\cdot(x\cdot u+y\cdot v)}\,dx\,dx$$

For image processing purposes it is resonable to assume that the Fourier transform of periodic functions always exists. An inverse Fourier transform is defined by

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\cdot e^{2\cdot\pi\cdot i\cdot(x\cdot u+y\cdot v)}\,du\,dv$$

Parameters (x,y) denote image co-ordinates, and co-ordinates (u,v) are called spatial frequencies. The function f(x,y) on the left hand side of the equation can be interpreted as a linear combination of simple periodic patterns e^(2πi(xu+yv)). The real and imaginary components of the pattern are sine and cosine functions, and the function F(u,v) is a weight function which represents the influence of the elementary patterns. The following properties of Fourier transformation are interesting from the image processing point of view: (**F** denote the Fourier transform)

<u>Linearity</u>
F{af1(x,y) + bf2(x,y)} = aF1(u,v) + bF2(u,v)
<u>Shift of the origin in the image domain</u>
F{f(x-a,y-b)} = F(u,v)e^(-2πi(au + bv))
<u>Shift of the origin in the frequency domain</u>
F{f(x,y)e^(2πi(uox+voy))} = F(u-uo,v-vo)
<u>Symmetry: If f(x,y) is real valued</u>
F(-u,-v) = F*(u,v) where * denotes complex conjugate.
<u>Duality of convolution: Convolution and its Fourier transform are related by</u>
F{(f*h)(x,y)} = F(u,v)H(u,v)
F{f(x,y)h(x,y)} = (F*H)(u,v)

Usage:

Reconstructive image methods, such as magnetic resonance imaging (MRI) or X-ray computed tomography (CT), generate their images in two steps. First, the object to be imaged is probed using som form of physical radiation, and then the image is reconstructed from the outcome of the probing. Although the probing arrangements differ widely, a remarkablle large number of reconstructive imaging methods gather data which may be interpreted, either directly or after some preprocessing, as the example of the Fourier transform of the wanted image. Examples of this kind include MRI, spotlight-mode synthetic aperture radar, radio interferometry, CT, and various methods of differaction tomography. The reconstruction of all these cases mounts to computing a 2- or 3-D signal f from a sampled version of its Fourier transform f^, or it can be cast in this form.

Wavelet Transformation:

Like tha fast Fourier Transform (FFT), the discrete wavelet transform (DWT) is a fast, linear operation that operates on a data vector whose length is an integer power of two, transforming it into a numerical different vector of the same length. Also like the FFT, the wavelet transform is invertible and in fact orthogonal - the inverse transform, when viewed as a big matrix, is simply the transpose of the transform. Both FFT and DWT, therefore, can be viewed as a rotation in function space, from the input space (or time) domain, where the basis functions are the unit vectors ei, or Dirac delta functions in the continuum limit, to a different domain. For the FFT, this new domain has basis functions that are familiar sines and cosines. In the wavelet domain, the basis functions are somewhat more complicated and have the fanciful names "mother functions" and "wavelets." Wavelets represent another approach to decomposing complex signals into sums of basis functions, in this respect they are similar to Fourier decomposition approaches, but they have an important difference.

Fourier functions are localized in frequency but not in space, in the sense that they isolate frequencies, but not isolated occurences of those frequencies (that is, not throughout the domain of interest of the signal). This means that small frequency changes in a Fourier transform will produce changes everywhere in the time domain. Wavelets are local in both frequency (via dialations) and time (via translation), because of this they are able to analyze data at different scales or resolutions much better than simple since and cosine can. Modeling a spike in a function (a noise dot, for example) with a sum of infinite functions wil be hard because of its strict locality, while functions that are already local will be naturally suited to the task. This means that such functions lend themselves to more compact representation via wavelets, sharp spikes and discontinuities normally take fewer wavelet bases to represent than if sine-cosine basis functions are used. The wavelet basis is then provided by the functions

$$\Phi_{(s,l)}(x) = 2^{-\left(\frac{s}{2}\right)} \cdot \Phi\left(2^{-s} \cdot x - l\right)$$

Here the scale factor s indicates the wavelet's width (a power of2) and the location index l its position (and integer).

Note that Φ(s,l) are self-similar and are selected to be orthonormal, so

$$\int \Phi_{\left(s_1,l_1\right)} \cdot \Phi_{\left(s_2,l_2\right)} \, d\blacksquare = 0 \qquad \text{if} \quad s_1 \neq s_2 \quad \text{or} \quad l_1 \neq l_2$$

and it is thus possible to represent other functions as a linear combination of the $\Phi(s,l)$.
Wavelets have been used with enormous success in data compression, for example, in fingerprint data reduction, and in image noise suppression. It is possible to erase to zero the contribution of wavelet components that are "small" and correspond to noise without erasing the important small detail in the underlying image. Thus, there is noise suppression without the bluring characteristics of Fourier filters.

Hough Transform:

The Hough Transform has been developed by Paul Hough in 1962 and patented by IBM. It became in the last decade a standard tool in the domain of artificial vision for the recognition of straight lines, circles and ellipses. The Hough Transform is particularly robust to missing and contaminated data. It can also be extended to non-linear characteristic relations and made resistant to noise by use of anti-aliasing techniques.

The Hough Transform has been originally developed to detect analytically representable features in binarized images, such as straight lines, circles or ellipses. The characteristic relation of the sought-for feature is back-projected in the parameter space. Each set pixel (xi,yi) defines a relation between the parameters of the characteristic relation which can be represented as a curve in the parameter space. Each pixel (xi,yi) along the lines will generate by back-projection a straight line of equation . These lines intersect at the locus characterizing the line. All points belonging to the curve have been mapped into a single location in the transformed space, allowing an easier detection. The Hough transform involves a peak finding algorithm to detect the features. A noteworthy characteristic of the HT is that it does not require the connectedness of the co-linear points. Segmented lines will generate a peak in the parameter space and the lacking segments simply do not contribute to the transform. On the other side, artifact peaks might be generated in presence of noise and high density of features by coincidental intersections in the parameter space. To a certain extent, artifacts can be avoided by using anti-aliasing techniques and adapted peak finding algorithms. Also the HT treats each image point independently, allowing a parallel implementation of the method.

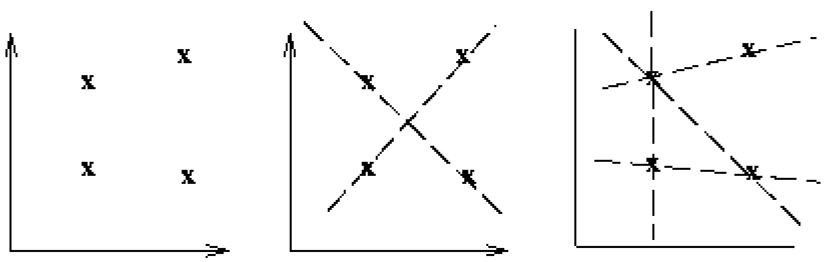
If an image consists of objects with known shape and size, segmentation can be viewed as a problem of finding this object within an image. Typical tasks are to locate circular pads in printed circuit boards, or to find objects to specific shapes in aerial or satellite data, etc. One of many possible ways to solve these problems is to move a mask with an appropriate shape and size along the image and look for correlation between the image and the mask. Unfortunately, the specified mask often differs too much from the object's representation in the processed data, because of shape distortion, rotation, soom, etc. One very effective method that can solve this problem is the Hough transform, which can be used successfully in segmentation of overlapping or semi-occluded objects.

The original Hough transform was designed to detect straight lines and curves, and this original method can be used if analytic equations of object borderlines are known, no prior knowledge of region position is necessary. A big advantage of this approach is robustness of segmentation results; that is, segmentation is not too sensitive to imperfect data or noise. Neverthless, it is often impossible to get analytic expressions describing boarders.

The basic idea of the method can be seen from the simple problem of detecting a straight line in an image. A straight line is defined by two points A = (x1,y1) and B = (x2,y2). All straight lines going through the point A are given by the expression y1 = kx1 + q for some values k and q. this means that he same equation can be interpreted as an equation in the parameter space k, q; all the straight lines going though the point A are then represented by the equation q = -x1k + y1. Straight lines going through the point B can likewise be represented as q = -x2k + y2. The only common point of both straight lines in the k, q parameter space is the point which in the original image space represents the only existing straight line connecting point A and B.

This means that any straight line in the image is represented by a single point in the k, q parameter space and any part of this straight line is transformed into the same point. The main idea of line detection is to determine all the possible line pixels in the image, to transform all lines that can go through these pixels into corresponding points in the parameter space, and to detect the points (a,b) in the parameter space that frequently resulted from the Hough transform of lines y = ax + b in the image.

The Hough technique is useful for computing a global description of a feature(s) (where the number of solution classes need not be known a priori), given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (e.g. coordinate point) indicates its contribution to a globally consistent solution (e.g. the physical line which gave rise to that image point). As a simple example, consider the common problem of fitting a set of line segments to a set of discrete image points (e.g. pixel locations output from an edge detector). The diagram below shows some possible solutions to this problem.

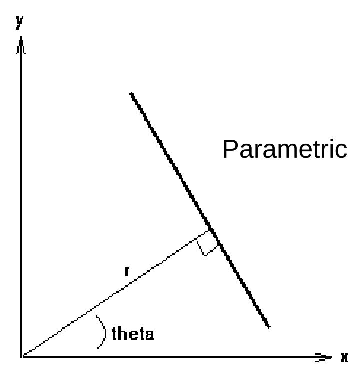


(a) Coordinate points. (b) and (c) Possible straight line fittings.

We can analytically describe a line segment in a number of forms. However, a convenient equation for describing a set of lines uses parametric or normal notion:

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) := r$$

where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-Axis. For any point (x,y) on this line, r and θ are constant.



Parametric description of a straight line.

In an image analysis context, the coordinates of hte points of edge segments (xi,yi) in the image are known and therefore serve as constants in the parametric line equation, while r and θ are the unknown variables we seek. If we plot the possible (r,θ) values defined by each (xi,yi), points in cartesian image space map to curves (i.e. sinusoids) in the polar Hough parameter space. This point to curve transformation is the Hough tranformation for straight lines. When viewed in Hough parameter space, points which are collinear in the cartesian image space become readily apparent as they yield curves which intersect at a common (r,θ) point.

Algorithm: Curve detection using the Hough transform

Consider an arbitrary curve represented by an equation f(x,a)=0, where a is the vector of curve parameters.

1. Quantize parameter space within the limits of parameters a. The dimensionality n of the parameter space is given by the number of parameters of the vector a.

2. Form an n-dimensional accumulator array A(a) with structure matching the quantization of parameter space; set all elements to zero.

3. For each image point (x1,x2) in the appropriately thresholded gradient image, increase all accumulator cells A(a) if f(x,a) = 0

$$A(a) = A(a) + \Delta A \quad \text{for all a inside the limits used in step 1.}$$

4. Local maxima in the accumulator array A(a) corresponding to realizations of curves f(x,a) that are present in the original image.

The transform is implemented by quantizing the Hough parameter space into finite intervals or accumulator cells, i.e. multidimensional array. As the algorithm runs, each (xi,yi) is transformed into a discretized (r,θ) curve and the accumulator cells which lie along this curve are incremented. Peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

We can use this same procedure to detect other features with analytical descriptions. For instance, int the case of circles, the parametric equation is

$$(x-a)^2 + (y-b)^2 = r^2$$

where a and b are the coordinates of the center of the circle, and r is the radius. In this case, the computational complexity of the algorithm begins to increase as we now have three coordinates in the parameter space and a 3D accumulator. In general, the computation and the size of the accumulator array increase polynomially with the number of parameters. Thus, the basic Hough technique described hee is only practical for simple curves.

The randomized Hough transform offers a different approach to achieve increased efficiency; it randomly selects n pixels from the edge image and determines n parameters of the detected curve followed by incrementing a single accumulator cell only. Recent extensions to the randomized Hough tranform use local information about the edge image and apply Hough transform process to a neighborhood of the edge pixel. If the parametric representations of the desired curves or region boarders are known, this method works very well, but unfortunately this is not often the case. The desired region boarders can rarely be described using a parametric boundary curve with a small number of parameters; in this case, a generalized Hough transform can offer the solution.

This method constructs a parametric curve, region boarder description based on sample situations detected in the learning stage. Assume that shape, size, and rotation of the desired region are known. A reference point x^R is chosen at any location inside the sample region, then an arbitrary line can be constructed starting at this reference point aiming in the direction of the region border. The border direction, edge direction is found at the intersection of the line and the region border. A reference table (R-Table) is constructed, and intersection parameters are sorted as a function of the border direction at the intersection point; using different lines aimed from the reference point, all the distances of the reference point to region borders and the border directions at the intersections can be found. The resulting table can be ordered according to the border directions at the intersection points.

This implies that there may be more than one (r,α) pair for each ϕ that can determine the co-ordinates of a potential reference point.

$$
\begin{array}{ll}
\phi_1 & \left[(r_1)^1, (\alpha_1)^1\right], \left[(r_1)^2, (\alpha_1)^2\right], \ldots, \left[(r_1)^{n1}, (\alpha_1)^{n1}\right] \\
\phi_2 & \left[(r_2)^1, (\alpha_2)^1\right], \left[(r_2)^2, (\alpha_2)^2\right], \ldots, \left[(r_2)^{n1}, (\alpha_2)^{n1}\right] \\
\phi_3 & \left[(r_3)^1, (\alpha_3)^1\right], \left[(r_3)^2, (\alpha_3)^2\right], \ldots, \left[(r_3)^{n1}, (\alpha_3)^{n1}\right] \\
\phi_k & \left[(r_k)^1, (\alpha_k)^1\right], \left[(r_k)^2, (\alpha_k)^2\right], \ldots, \left[(r_k)^{n1}, (\alpha_k)^{n1}\right]
\end{array}
$$

Assuming no rotation and known size, remaining description parameters required are the co-ordinates of the reference point

$$\left[(x_1)^R, (x_2)^R\right]$$

If size and rotation of the region may vary, the number of parameters increase to four.

Each pixel x with a significant edge in the direction $\phi(x)$ has co-ordinates of potential reference points

$$\{x1 + r(\phi)\cos[\alpha(\phi)],\ x2 + r(\phi)\sin[\alpha(\phi)]\}$$

This must be computed for all possible values of r and α according to the border direction $\phi(x)$ given in the R-Table. The following algorithm presents the generalized Hough transform in the most genral of cases in which rotation (τ) and size (S) may both change. If either there is no change in rotation (τ=0) or there is no size change (S=1), the resulting accumulator data structure A is simpler.

Generalized Hough transform

1. Construct and R-Table description of the desired object.

2. Form a data structure A that represents the potential reference points $A(x1,x2,S,\tau)$ set all accumulator cell values $A(x1,x2,S,\tau)$ to zero.

3. For each pixel (x1,x2) in a threshold gradient image, determine the edge direction $\Phi(x)$;

find all potential reference points x^R and increase all $A(x^R,S,\tau)$, $A(x^R,S,\tau) = A(x^R,S,\tau) + \Delta A$ for all possible values of rotation and size change,

$$\left(x_1\right)^R = x_1 + r(\phi)\cdot S\cdot\cos(\alpha(\phi)+\tau) \qquad \left(x_2\right)^R = x_2 + r(\phi)\cdot S\cdot\sin(\alpha(\phi)+\tau)$$

4. The location of suitable regions is given by local maxima in the A data structure.

As mentioned before, the Hough transform was initially developed to detect analytically defined shapes, such as lines, circles, or ellipses in general images, and the generalized Hough transform can be used to detect arbitrary shapes. However, even the generalized Hough transform requires the complete specification of the exact shape of the target object to achieve precise segmentation. Therefore, it allows detection of objects with complex but pre-determined shapes.

Chapter 4 Summary:

Image Pre-processing
-Operations with images at the lowest level of abstraction, both input and output are intensity images, are called pre-processing.
-The aim of pre-processing is an improvement of the image data that suppresses unwilling distortion or enhance some image features important for further processing.
-Four basic type of pre-processing methods:

(1) Brightness transformation
-There are two classes of pixel brightness transformation
(A) Brightness corrections, modify pixel brightness taking into account its original brightnes and its position in the image.
(B) Gray-scale transformations, change brightness without regard to position in the image.
-Frequently used brightness transformations include:
*Brightness thresholding
*Histogram equalization
*Logarithmic gray-scale transforms
*Look-up table transforms
*Pseudo-color transforms
-The goal of histogram equalization is to create an image with equally distributed brightness levels over the whole brightness scale.

(2) Geometric transformation
-Geometric transformation permit the elimination of the geometric distortions the occure when and image is captured. A geometric transform typically consists of two basic steps:
(A) Pixel co-ordinate transformations, map the co-ordinates of the input image pixel to a point in the output image, affine and bilinear tranforms are frequently used
(B) Brightness interpolation, the output point co-ordinates do not usually match the digital grid after the transform and interpolation is employed to determine brightness of output pixels; nearest-neighbor, linear, and bi-cubic interpolations are frequently used.

(3) Local neightborhood pre-processing
-Local pre-processing methods use a small neighborhood of a pixel in an input image to produce a new brightness value in the output image. For the pre-processing goal, two groups are common: smoothing and edge detection.
(A) Smoothing aims to suppress noise or other small flunctuations in the image; it is equivalent to suppressing high frequencies in the GFourier transform domain. Smoothing approaches are based on direct averaging blur image edges. More sophisticated approaches reduce bluring by averaging in homogenous local neighborhoods.
(B) Edge is a property attached to an individual pixel and has two components, magnitude and direction. Gradient operators determine edges, locations in which the image function undergoas rapid changes; they have a similar effect to suppressing low frequencies in the Fourier transform domain. Most gradient operator can be expressed using convolution masks, examples include Roberts, Laplace, Prewitt, Sobel, Robinson, and Kirsch operators. The main disadvantage of convolution edge detectors is their scale dependence and noise sensitivity. There is seldom a sound reson for choosing a particular size of a local neightborhood operator
-Other local pre-processing operations include line finding, line thinning, line filling, and interest point detection

(4) Image restoration
-Image restoration metods aim to suppress degradation using knowledge about its nature. Most image restoration methods are based on deconvolution applied globally to the entire image

Chapter 5 Summary:

Image Segmentation
-The main goal of image segmentation is to divide an image into parts that have a strong correlation with objects or areas of the real world depicted in the image. Segmentation methods can be divided into three groups: thresholding, edge-based segmentation and region-based segmentation. Each region can be represented by its closed boundary, and each closed boundary describes a region. Image data ambiguity is one of the main segmentation problems, often accompanied by information noise. The more a priori information is available to the segmentation process, the better the segmentation results that can be obtained.
(A) Thresholding
-Thresholding represents the simplest image segmentation process, and it is computationally inexpensive and fast. A brightness constant called a threshold is used to segment objects and background. Single thresholds can either be applied to the entire image, global thresholding, or can vary in image parts, local thresholding. Only under very unusual circumstances can thresholding be successful using a single threshold for the whole image.
-Optimal thresholding determines the threshol as the closest gray-level corresponding to the minimum probability between the maxima of two or more normal distributions. Such thresholding results in minimum error segmentation.

(B) Edge-based image segmentation
-Edge based segmenation relies on edges found in an image by edge detecting operators, these edges mark image locations of discontinuities in gray-level, color, texture, etc. The most common problem of edge-based segmentation, cause by image noise or unsuitable information in an image, are an edge presence in locations where there is no border, and no edge presence where a real border exists.
-Edge relaxation, edge properties are considered in the context of neighboring edges. If sufficient evidence of the border exists, local edge strength increase and vice versa. Using a global relaxation (optimization) process, continuous borders are constructed.
-Three types of region boarders may be formed: inner, outer, and extended. The inner boarder is always part of a region, but the outer border never is. Therefore, using inner or outer border definition, two adjacent regions never have a common border. Extended borders are defined as single common borders between adjacent regions still being specified by standard pixel co-ordinates.
-Hough transform segmentation is applicable if objects of known shape are to be detected within an image. The Hough transform can detect straight lines and curves, object borders, if their analytic equations are known. It is robust in recognition of occluded and noisy objects.

(C) Region-based image segmentation
-Region growing segmentation should satisfy the following condition of complete segmentation, euqtion (5.1), and maximum region homogeneity, equations (5.31), and (5.32).

-Three basic approaches to region growing exist: region merging, region spliting, and split-and-merge region growing

(1) Region merging starts with an oversegmented image which regions satisfy the equations above.
(2) Region splitting is the opposite of region merging. Region splitting begins with an undersegmented image which does not satisfy conditions in the equation (5.31). Therefore, the existing image regions are sequentially plit to satisfy conditions (5.1), (5.31), and (5.32)
(3) A combination or splitting and merging may result in a method with the advantages of both other approaches. Split-and-merge approaches typically use pyramid image representations. Because both split-and-merge processing options are available, the starting segmentation does not have to satisfy either condition (5.31) or (5.32)

Chapter 6 Summary:

Shape representation and description
-Region description generates a numeric feature vector or a non-numeric syntactic description world, which characterizes properties, for exmaple shape, of described region. While many practical shape description methods exist, there is no generally accepted methodology of shape description. Further, it is not known what is important in shape.
-The shape classes represent the generic shapes of the objects belonging to the same classes. Shape classes should emphasize shape differences among classes, while the shape variations within classes should not be reflected in the shape class description
-Region identification assigns unique labels to image regions. If nonrepeating ordered numerical labels are used, the largest integer label gives the number of regions in the image.
-Counter-based shape descriptors
-Chain codes describe an object by a sequence of unit-size line segment with a given orientation, called Freeman's code.
-Simple geometric border representations are based on geometric properties of described regions
*Bondary length
*Curvature
*Bending Energy
*Signature
*Chord distribution
-Fourier shape descriptors can be applied to closed curves, co-ordinates of which can be treated as periodic signals.
-Region-based shape descriptors
-Simple geometric descriptors use geomertic properties of described regions:
*Area
*Euler's number
*Projections
*Height, width
*Eccentricity
*Elongatedness
*Rectangularity
*Direction
*Compactness

-More complex shapes can be described using region decomposition into smaller and simpler sub-regions. Objects can be represented by a planar graph with nodes representing sub-regions resulting from region decomposition. Region shape can then be described by the graph properties. There are two general types:

(1) Region thinning, leads to the region skeleton that can be described by a graph. Thinning procedures often use a medial axis transform to construct a region skeleton. Under the medial axis definition, the skeleton is the set of all region points which have the same minimum distance from the region boundary for at least two separate boundary points.
(2)Region decomposition, considers shape recognition to be a hierarchical process. Shape primitives are defined at the lower level, primitives being the simplest elements which form the region. A graph is constructed at the higher-level, nodes result from primitives, arcs describe the mutual primitive relations.
-Shape classes represent the generic shapes of the objects belonging to the class and emphasize shape difference among classes. A widely used representation of in-class shape variations is determined of class-specific regions in the feature space.