Computer Vision

Unit-2

Shapes and Regions

Binary shape analysis: skeletons and thinning - other methods for shape recognition; Boundary pattern analysis: Boundary tracking procedures - centroidal profiles; Line detection - Hough transform - Foot of normal method; circle and ellipse detection: Hough-based scheme - Ellipse detection methods

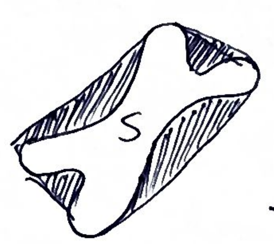
* Binary Shape Analysis

Boundary Segments

→ The boundary of an image can be decomposed into segments - it is useful to extract information about the concave parts of objects

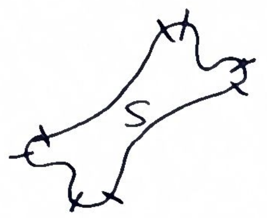
→ This is done by calculating the convex hull of a region.

→ The convex hull H of an arbitrary set S, is the smallest set containing S → H - S = convex deficiency



→ The convex deficiency is the shaded region

→ The convex deficiency can be followed to mark features. This is done by following the contour of S and marking points that transition in or out of the convex deficiency



Skeletons

→ An important approach to representing the structural shape of a plane region is to reduce it to a graph.

→ This reduction may be accomplished by obtaining the skeleton of the region via a skeletonizing / thinning algorithm.

→ The skeleton of a region may be defined as the medial axis transformation (MAT)

→ The MAT of a region R with border B is as follows:

- For every point P in R, we find the closest neighbor in B. If p has more than one neighbor, it is said to belong to the medial axis (skeleton)

→ The MAT is based upon the prairie fire concept

→ Consider an image region as a prairie of uniform, dry grass, and suppose that a fire is lit along its border.

→ All fire fronts will advance into the region at the same speed.

→ The MAT of the region is the set of points reached by more than one fire front at the same time.

Skeletonizing Algorithms

Thinning algorithms iteratively delete boundary points of a region subject to the constraints that the deletion of those points:

(i) does not remove end points

(ii) does not break connectivity

(iii) does not cause excessive thinning of the region

1. Consider an 8-neighborhood as follows:

p9	p2	p3
p8	p1	p4
p7	p6	p5

2. Flag a contour point p1 for deletion if the following conditions are satisfied:

(a) $2 \leq N(p_1) \leq 6$

(b) $T(p_1) = 1$

(c) $p_2 \cdot p_4 \cdot p_6 = 0$

(d) $p_4 \cdot p_6 \cdot p_8 = 0$

where $N(p_1)$ is the number of non-zero neighbors of p_1.

$T(p_1)$ is the number of 0-1 transitions in the ordered sequence $p_2, p_3 \ldots p_8, p_9, p_2$

For eg

0	0	1
1	p1	0
1	0	1

$T(p_1) = 3$

$N(p_1) = 4$

(3) In the next step, conditions (a) and (b) are the same, but (c) and (d) are changed as follows:

$$(c')\ p_2 \cdot p_4 \cdot p_8 = 0$$

$$(d')\ p_2 \cdot p_6 \cdot p_8 = 0$$

(4) Step 2 is applied to every border pixel in the binary region under consideration

(5) If all the conditions are satisfied, the point is flagged for deletion, but not deleted until all border points have processed.

After processing all border points, those which were flagged are deleted (changed to 0)

(6) Then, step 3 is applied to the resulting data in exactly the same manner as step 2.

(7) This procedure is repeated iteratively until no further points are deleted.

* Boundary Tracking?

→ Before objects can be matched with their boundary patterns, means must be found for systematically tracking around the boundary of all the objects in an image.

→ It is necessary to ensure that:

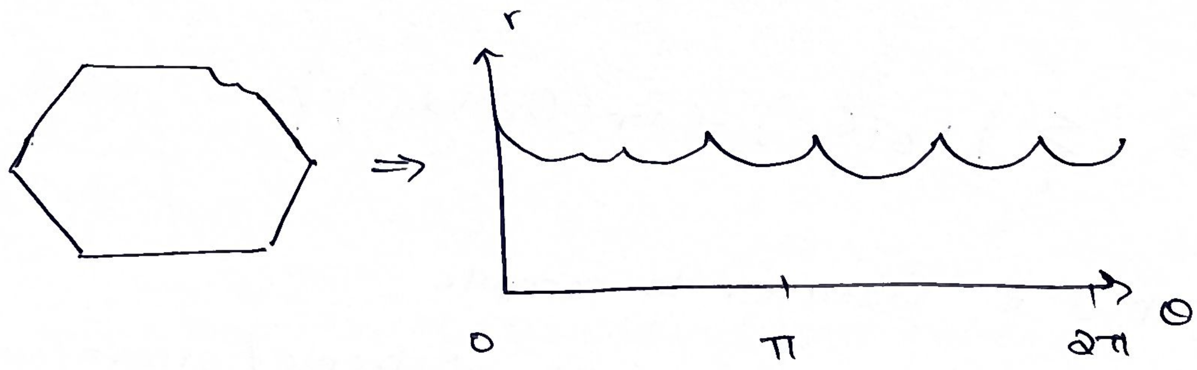
(i) direction is not reversed

(ii) know once the whole boundary has been traced once

(iii) record which object boundaries have been encountered

Centroidal Profiles

→ A centroidal profile is a polar coordinate system that is set up relative to the centroid of the object and the object is plotted as an (r, θ) graph.



→ The 1D graph obtained is matched against the corresponding graph for an idealized object of the same type.

→ Since objects generally have an arbitrary orientation, it is necessary to slide the idealized graph along that obtained from the image data obtained, until the best match is obtained

→ possible orientation : α_j

boundary graph = B

template graph T

→ The match for each possible orientation α_j of the object is tested by measuring the differences in radial distance between the boundary graph B & template graph T, for various values of θ and summing their squares to give a difference measure D_j for the quality of the fit:

$$D_j = \sum_i \left[r_B(\theta_i) - r_T(\theta_i + \alpha_j) \right]^2$$

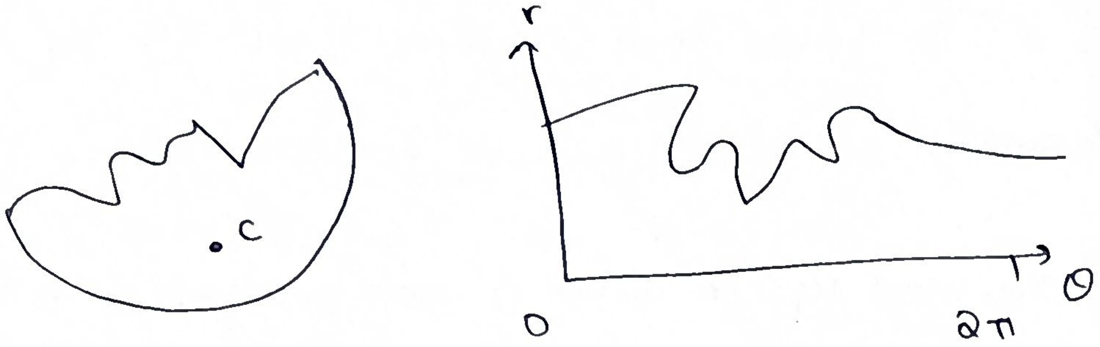
→ Alternatively, the absolute magnitude of the differences can be used:

$$D_j = \sum_i \left| r_B(\theta_i) - r_T(\theta_i + \alpha_j) \right|$$

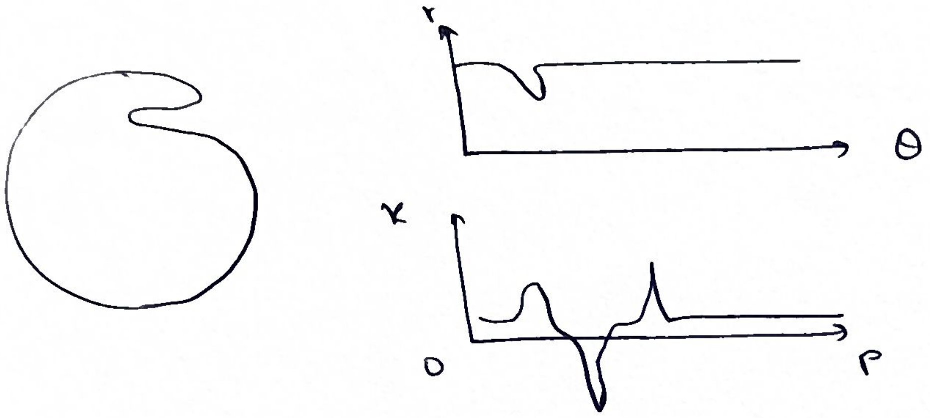
This method is (i) easier to compute
(ii) less biased by extreme/erroneous diff. values,

* Problems with the Centroidal Profile Approach

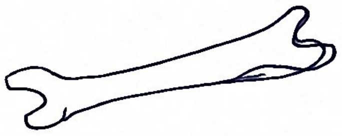
① Any major defect or occlusion of the object boundary can cause the centroid to be moved away from its true position, whole profile is grossly distorted.



② The (r,θ) can be multivalued for a certain class of object. This has the effect of making the matching process partly 2D and leads to complication and excessive computation



③ It is difficult to obtain an accurate centroidal profile for the region near the centroid of elongated objects such as spanners or screwdrivers



(s, ψ) Plot

→ Better suited than the (r,θ) graph when there are defects and occlusions

→ The (s,ψ) graph does not require prior estimation of the centroid or some other reference point, since it is computed directly from the boundary.

→ It is a plot of the tangential orientation, ψ, as a fn. of boundary distance s.

→ The graph has a ψ value that increases by 2π for each circuit of the boundary, i.e $\psi(s)$ is not periodic in s.

→ This problem is tackled by making a comparison with the shape of a circle of the same boundary length P.

Then $\Delta\psi = \psi - \left(\frac{2\pi s}{P}\right)$

a $(s, \Delta\psi)$ graph is plotted.

* <u>Line Detection</u>

$\boxed{\text{Hough Transform}}$

→ The basic concept involved in locating lines by the Hough Transform is point-line duality.

→ A point P can be defined either as a pair of coordinates or in terms of the set of lines passing through it.

→ The HT is applied by parametrizing the slope-intercept equation:

$$y = mx + c$$

→ Every point on a straight edge is then plotted as a line in (m,c) space.

→ There may be near unlimited ranges of the (m,c) values. This is overcome by using two sets of plots, the first corresponding to slopes of less than 1.0 and the second to slopes of 1.0 or more.

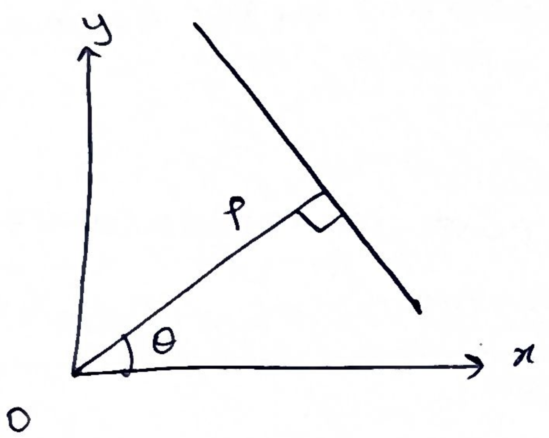
When the slope is > 1.0, the equation is:

$$y = \tilde{m}x + \tilde{c}$$

where $\tilde{m} = \frac{1}{m}$

→ The usage of 2 equations can be eliminated by using a normal (θ, ρ) form : $\rho = x\cos\theta + y\sin\theta$

eg. the point $P_1(x_1, y_1)$ ⇒ $\rho = x_1\cos\theta + y_1\sin\theta$

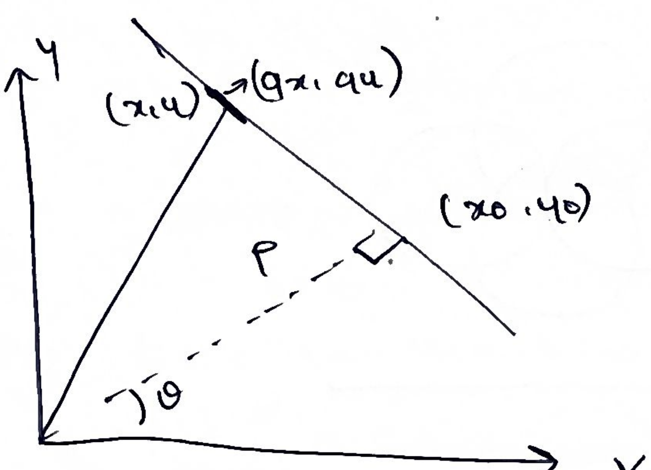


Foot of Normal Method

→ Consider an edge with pixel coordinates (x, y)

→ Compute the intensity gradient (g_x, g_y) for each edge is pixel.

→ Consider (x_0, y_0) as the foot of the normal from the origin



Then : $\frac{g_y}{g_x} = \frac{y_0}{x_0}$ &

$$(x - x_0)x_0 + (y - y_0)y_0 = 0$$

Solve to find the coordinates (x_0, y_0)

$$x_0 = v q_x$$

$$y_0 = v q_y$$

$$v = \frac{x q_x + y q_y}{q_x^2 + q_y^2}$$

* Circle and Ellipse Detection

Hough Transform for Circle Detection

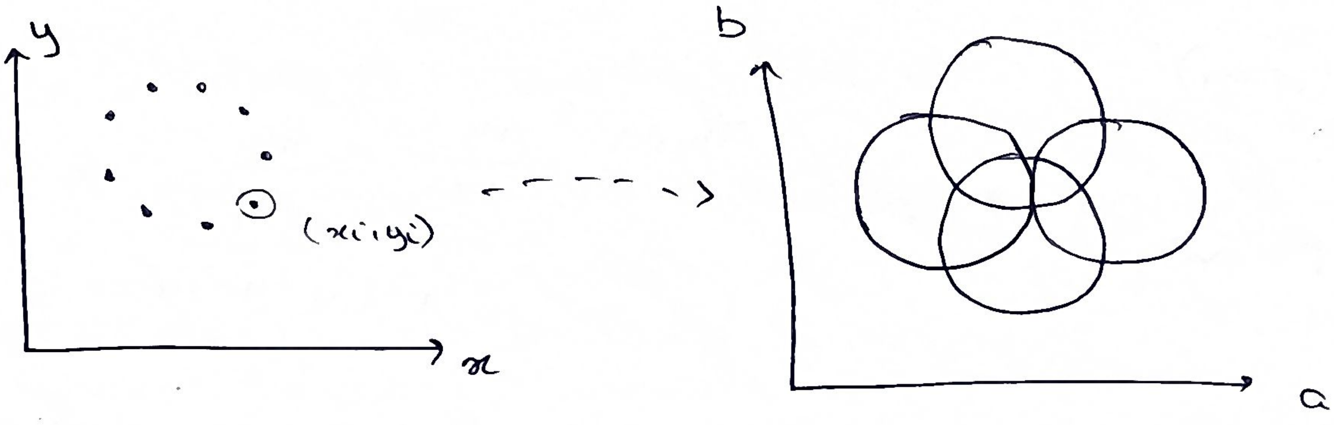
→ The equation of a circle is: $(x_i - a)^2 + (y_i - b)^2 = r^2$

→ A circle of specific radius r is uniquely identified by the location of its center, i.e the parameter pair (a, b).

→ For circle detection of known radius r, a point in image space becomes a circle in Hough space.

→ If the radius is known, it is a 2D Hough space. The accumulator array is $A(a, b)$

→ If the array is unknown, it is a 3D Hough space. The accumulator array is $A(a, b, r)$



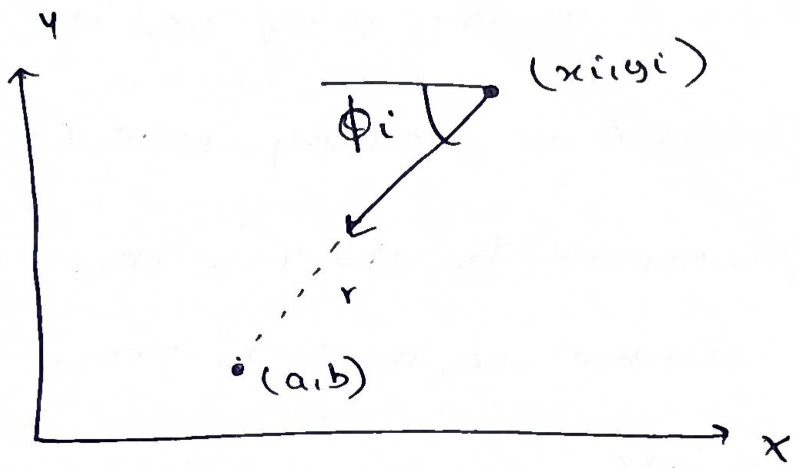
Usage of Gradient Information

- If a gradient-based edge detector, for each point (x_i, y_i), the edge direction ϕ_i can be found.

- For a specific radius r, one only needs to move a distance r in the direction of ϕ_i to obtain an estimate of the center of the circle.

$$a = x_i - r\cos\phi_i$$

$$b = y_i - r\sin\phi_i$$



Hough Transform for Ellipse Detection

- The equation of an ellipse is:

$$\frac{(x_i\cos\vartheta + y_i\sin\vartheta - a)^2}{d_1^2} + \frac{(-x_i\sin\vartheta + y_i\cos\vartheta - b)^2}{d_2^2} = 1$$

(a, b) = center of ellipse

d_1 = length of major axis

d_2 = length of minor axis

ϑ = angle between major axis and horizontal axis.

- An ellipse is then uniquely identified by the location of its center, size & orientation, i.e by the parameter tuple $(a, b, d_1, d_2, \vartheta)$

→ For arbitrary ellipse detection, there is a 5D Hough space. The accumulator array is $A(a, b, d_1, d_2, \theta)$.

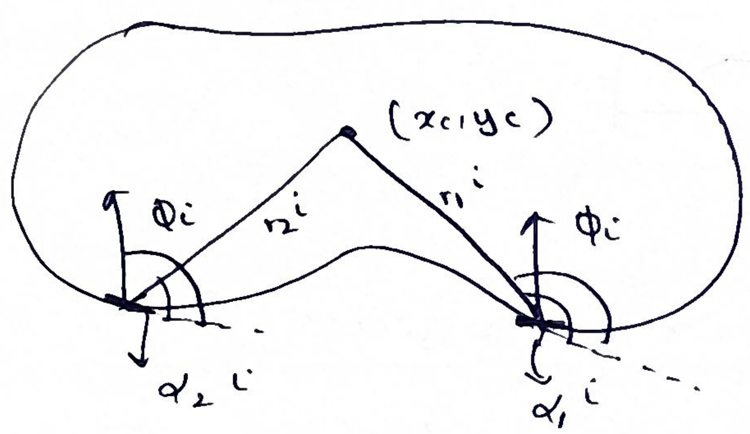
* Generalized Hough Transform

→ Preprocess a sample of the shape

(i) Run a gradient-based edge detector, so that for each border pixel there is an edge pixel and its associated orientation

(ii) Choose an arbitrary point C as the center of the shape.

(iii) Compute for the given shape what distance and in which direction one needs to move the edge pixel to reach the center point C.



Algorithm

1. Find the object center (x_c, y_c) given the edges (x_i, y_i, ϕ_i)

2. Create an accumulator array $A(x_c, y_c)$

3. Initialize : $A(x_c, y_c) = 0 \quad \forall (x_c, y_c)$

4. For each edge point (x_i, y_i, ϕ_i):

For each entry r_k^i in the shape table compute:

$$x_c = x_i + r_k^i \cos \alpha_k^i$$

$$y_c = y_i + r_k^i \sin \alpha_k^i$$

Increment the accumulator : $A(x_c, y_c) = A(x_c, y_c)+1$

5. Find the local maxima in $A(x_c, y_c)$.

* Features of Hough Transform

1. In order to handle different scales s, a new dimension must be added in the Accumulator.

2. In order to handle different rotations θ, a new dimension must be added in the Accumulator.

3. HT is insensitive to occlusion

4. HT can handle discontinuities

5. It is insensitive to noise.

6. HT is very effective on simple shapes (lines, circles, ellipses etc)