Generalized Hough Transform (GHT)

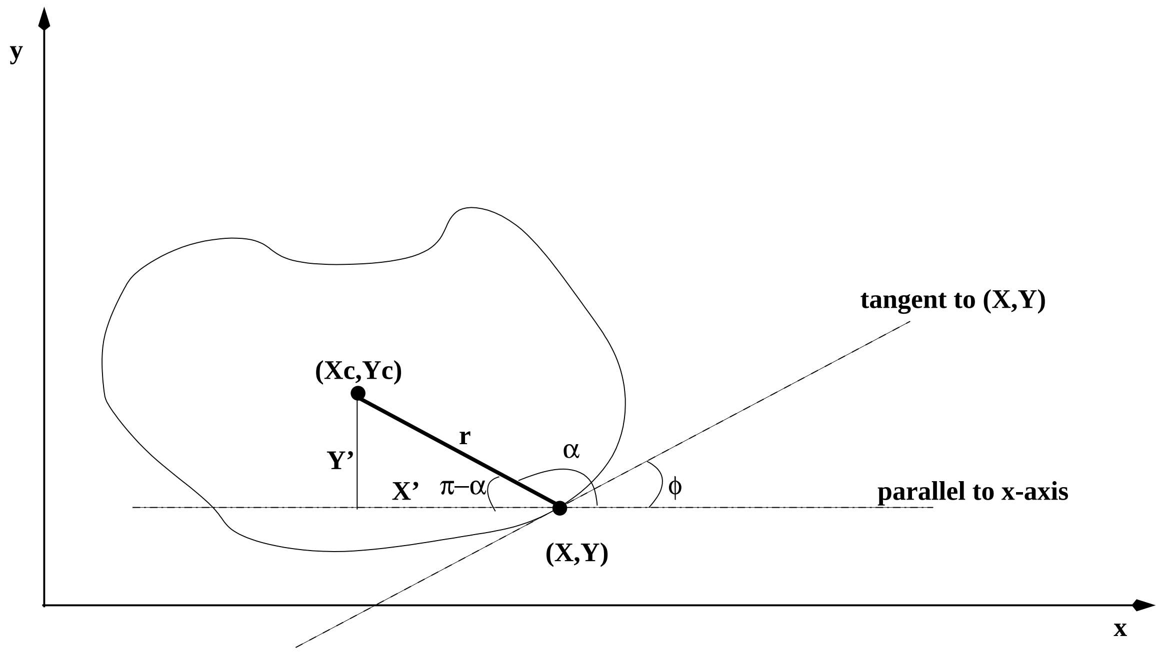
(Ballard and Brown, section 4.3.4, Sonka et al., section 5.2.6)

- The Hough transform was initially developed to detect analytically defined shapes (e.g., lines, circles, ellipses etc.).

- The generalized Hough transform can be used to detect arbitrary shapes (i.e., shapes having no simple analytical form).

- It requires the complete specification of the exact shape of the target object.

• Special case: fixed orientation and size



$$x = x_c + x' \quad \text{or} \quad x_c = x - x'$$
$$y = y_c + y' \quad \text{or} \quad y_c = y - y'$$

$$cos(\pi - \alpha) = \frac{y'}{r} \quad \text{or} \quad y' = rcos(\pi - \alpha) = -rsin(\alpha)$$
$$sin(\pi - \alpha) = \frac{x'}{r} \quad \text{or} \quad x' = rsin(\pi - \alpha) = -rcos(\alpha)$$

- Combining the above equations we have:

$$x_c = x + r\cos(\alpha)$$
$$y_c = y + r\sin(\alpha)$$

<u>Preprocessing step</u>

(1) Pick a reference point (e.g., (x_c, y_c))

(2) Draw a line from the reference point to the boundary.

(3) Compute ϕ (i.e., perpendicular to gradient's direction).

(4) Store the reference point (x_c, y_c) as a function of ϕ (i.e., build the *R-table*)

ϕ_1:	$(r_1^1, \alpha_1^1),\ (r_2^1, \alpha_2^1), \ldots$
ϕ_2:	$(r_1^2, \alpha_1^2),\ (r_2^2, \alpha_2^2), \ldots$
ϕ_n:	$(r_1^n, \alpha_1^n),\ (r_2^n, \alpha_2^n), \ldots$

- The *R-table* allows us to use the contour edge points and gradient angle to recompute the location of the reference point.

Note: we need to build a separate *R-table* for each different object.

Detection

(1) Quantize the parameter space:

$$P[x_{c_{\min}} \cdots x_{c_{\max}}][y_{c_{\min}} \cdots y_{c_{\max}}]$$

(2) for each edge point (x, y)

(2.1) Using the gradient angle ϕ, retrieve from the *R-table* all the (α, r) values indexed under ϕ.

(2.2) For each (α, r), compute the candidate reference points:

$$x_c = x + rcos(\alpha)$$
$$y_c = y + rsin(\alpha)$$

(2.3) Increase counters (voting):

$$++(P[x_c][y_c])$$

(3) Possible locations of the object contour are given by local maxima in $P[x_c][y_c]$

- If $P[x_c][y_c] > T$, then the object contour is located at $x_c, y_c)$

- **General case**

- Suppose the object has undergone some rotation θ and uniform scaling s:

$$(x', y') --> (x'', y'')$$

$$x'' = (x'cos(\theta) - y'sin(\theta))s$$

$$y'' = (x'sin(\theta) + y'cos(\theta))s$$

- Replacing x' by x'' and y' by y'' we have:

$$x_c = x - x'' \text{ or } x_c = x - (x'cos(\theta) - y'sin(\theta))s$$

$$y_c = y - y'' \text{ or } y_c = y - (x'sin(\theta) + y'cos(\theta))s$$

(1) Quantize the parameter space:

$$P[x_{c_{\min}} \cdots x_{c_{\max}}][y_{c_{\min}} \cdots y_{c_{\max}}][\theta_{\min} \cdots \theta_{\max}][s_{\min} \cdots s_{\max}]$$

(2) for each edge point (x, y)

(2.1) Using its gradient angle ϕ, retrieve all the (α, r) values from the *R-table*

(2.2) For each (α, r), compute the candidate reference points:

$$x' = rcos(\alpha)$$
$$y' = rsin(\alpha)$$

for($\theta = \theta_{\min}$; $\theta \leq \theta_{\max}$; θ++)

for($s = s_{\min}$; $s \leq s_{\max}$; s++)

$$x_c = x - (x'cos(\theta) - y'sin(\theta))s$$

$$y_c = y - (x'sin(\theta) + y'cos(\theta))s$$

++($P[x_c][y_c][\theta][s]$);

(3) Possible locations of the object contour are given by local maxima in $P[x_c][y_c][\theta][s]$

- If $P[x_c][y_c][\theta][s] > T$, then the object contour is located at x_c, y_c), has undergone a rotation θ, and has been scaled by s.

• Advantages

- The generalized Hough transform is essentially a method for object recognition.

- It is robust to partial or slightly deformed shapes (i.e., robust to recognition under occlusion).

- It is robust to the presence of additional structures in the image (i.e., other lines, curves, etc.).

- It is tolerant to noise.

- It can find multiple occurences of a shape during the same processing pass.

• Disadvantages

- It requires a lot of storage and extensive computation (but it is inherently parallelizable!).

- Faster variations have been proposed in the literature:

Hierarchical representations

First match using a coarse resolution Hough array

Then selectively expand parts of the array having high matches

Projections

- Instead of having one high-dimensional array, store a few two dimensional projections with common coordinates (e.g., (x_c, y_c), (y_c, θ), (θ, s), (s, x_c)).

- Find consistent peaks in these lower dimensional arrays.