Image Understanding & Interpretation

- Preprocessing
- Edge Detection
- Boundary Detection
- Segmentation
- Shape Descriptors
- Understanding & Interpretation

- Segmentation subdivides an image into its constituent regions or objects.
- Segmentation is a prerequisite for all automated image description, understanding, and interpretation.

Edge Detection

- **Edge** – a set of connected pixels, each of which is located at a step transition in gray level.
- An edge is a local phenomenon; may not form a closed path.
- **Crack edge** – the edge is on the dividing line between pixels.
- Edges can have direction.

In practice, most edges are blurred, so an edge is best modeled as a ramp.
- A ramp can be detected with derivatives.
- Derivatives amplify noise, so some form of blurring is often necessary.
- The edge is perpendicular to the gradient vector.
- The gradient vector points “up hill” toward the direction of “steepest climb.”

$$\nabla \mathcal{L} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f \\ \partial x \\ \partial f \\ \partial y \end{bmatrix}$$

Gradient Operators

- **Roberts**
  $$\nabla \mathcal{L} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f \\ \partial x \\ \partial f \\ \partial y \end{bmatrix}$$

- **Prewitt**
  $$\nabla f - \|\nabla \mathcal{L}\| = \sqrt{G_x^2 + G_y^2}$$

- **Sobel**
  $$\alpha( x, y) = \arctan \left( \frac{G_x}{G_y} \right)$$
Laplacian Operators

- The Laplacian produces a negative response on one side of the edge, and a positive response on the other side of the edge.
- The absolute value of the Laplacian will produce double edges.
- The zero-crossing of the Laplacian can be used to locate the edges.
- The sign of the Laplacian can be used to identify the direction of the edge.

\[
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0
\end{bmatrix}
\]

\[\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\]

Laplacian of Gaussian (LoG)

\[h(r) = \frac{1}{2\pi \sigma^2} e^{-\frac{r^2}{2\sigma^2}}\]

\[
\nabla^2 (h \ast f) = (\nabla^2 h) \ast f
\]

\[
\nabla^2 h(r) = \frac{-r^2 \sigma^4}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}}
\]

\[
\nabla^2 h(r) \approx \frac{r}{(2\pi)^3 \sigma^4 (u^2 + v^2)^2} e^{-\frac{r^2}{2\pi \sigma^2 (u^2 + v^2)}}
\]

Edge Linking: Local Processing

- Start with edge detection using Sobel operators.
- Every pixel with \(\nabla f\) greater than some threshold is considered an edge.
- Look in local neighborhood around each edge pixel and compare magnitude and direction of \(\nabla f\) to surrounding pixels.
- Pixels with similar \(\nabla f\) are considered part of the edge.

\[
\begin{bmatrix}
0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2
\end{bmatrix}
\]

Hough Transform

- The Hough transform is used to detect and locate straight lines in an image.
- The Hough transform normally operates on an image after edge detection, so that it detects straight line edges.
- The Hough transform is particularly useful in detecting man-made objects in natural surroundings. (why?)
- The Hough transform can be generalized to detect other types of curves, e.g. squares circles, ellipses, etc.
- The standard version of the Hough transform is roughly equivalent to the Radon transform used to model tomographic imaging systems.
Hough Transform

- Let \( L \) be a line, and \( p_1, p_2, \ldots, p_N \) be points on the line.
- Each point on \( L \), say \( p_n \), can potentially belong to many lines.
- For each point, \( p_n \), make a list, \( A_n \), of all the lines to which it can belong.
- Only one line will be on every list, \( A_1, A_2, \ldots, A_N \), and that line will be \( L \).

Hough Transform Algorithm

- Any line can be represented as a \((\rho, \theta)\) pair, i.e. a point in \((\rho, \theta)\) space.
- Algorithm:
  1. For each edge pixel in the image, draw a curve in \((\rho, \theta)\) space corresponding to every line that includes the edge pixel.
  2. Wherever curves for different edge pixels intersect, the curves add together.
  3. Peaks in \((\rho, \theta)\) space correspond to edges in the image that form straight lines.

Hough Transform for Circles

\[(x - x_c)^2 + (y - y_c)^2 = r^2\]

- Any circle in the image can be represented as a \((x_c, y_c, r)\) triplet, i.e. as a point in \((x_c, y_c, r)\) space.
- For each edge pixel in the image, plot all the points in \((x_c, y_c, r)\) space corresponding to circles that include the edge pixel.
- Search \((x_c, y_c, r)\) space for peaks corresponding to circles.
- Note: For this particular version of the Hough transform, each \((x_c, y_c)\) plane in \((x_c, y_c, r)\) space (constant \(r\)) is really just a correlation between the edge image and a circle of radius \(r\). Correlation can be calculated efficiently using the FFT.

Hough Transform Example

- Original
- After Sobel edge detection
- Hough transform
Generalized Hough Transform

- $R$ – shape reference point
- $(x^R, y^R)$ – coordinates of reference point
- $(r, \alpha)$ – distance and angle from reference point to edge pixel
- $\varphi$ – angle of edge
- The $R$-table contains a description of the shape:

$$
\begin{align*}
R &= (x^R, y^R) \\
\varphi &= \text{angle of edge} \\
(R, \alpha) &= (r, \alpha) - (x^R, y^R) \\
\end{align*}
$$

Construct the $R$-table description of the target shape

- Form data structure, $A(x,y)$, of accumulator cells; initialize all accumulator cells to zero
- Choose an image edge pixel, $(x,y)$, with edge angle $\varphi$, then for all $(r,\alpha)$ in the $R$-table on the line corresponding to $\varphi$:
  $$x^R = x + r \cos(\alpha)$$
  $$y^R = y + r \sin(\alpha)$$
  $$A(x^R, y^R) \leftarrow A(x^R, y^R) + \Delta A$$
- After processing all image edge pixels, a maximum in $A(x,y)$ indicates the reference point of an instance of the target shape.

Generalized Hough Transform

- The Hough transform can be further generalized by including a size parameter, $S$, and a rotation parameter, $\tau$.
  $$\begin{align*}
x^R &= x + r \cos(\alpha) + \tau \\
y^R &= y + r \sin(\alpha) + \tau \\
\end{align*}$$
- Change the data structure $A(x,y)$ to $A(x,y,\tau)$ (or $A(x,y,\tau,S)$).