Image Segmentation II: Gradient Based Segmentation

Edge detection has fundamental importance in image analysis. The purpose of edge detection is to identify areas of an image where a large change in intensity occurs. In typical images, edges characterize object boundaries and are useful for segmentation, registration and identification of objects in the scene.

Edge is a boundary between two homogeneous regions. The gray level properties of the two regions on either side of an edge are distinct and exhibit some local uniformity or homogeneity among themselves.

An edge is typically extracted by computing the derivative of the image intensity function. This consists of two parts:

- Magnitude of the derivative: measure of the strength/contrast of the edge
- Direction of the derivative vector: edge orientation

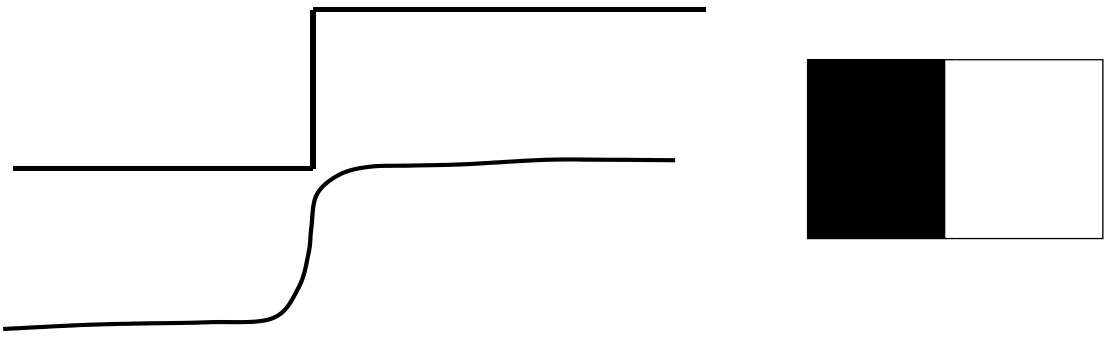


Figure 1: (a) Ideal step edge in 1-D (b)Step edge in 2D

There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

• **Gradient:** The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

•**Laplacian:** The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location. Suppose we have the following signal, with an edge shown by the jump in intensity below:

Edge Detection Techniques:

- **Sobel Operator:**

The operator consists of a pair of 3×3 convolution kernels as shown in Figure 2. One kernel is simply the other rotated by 90°.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 2: Kernels for Sobel operator

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient.

- The gradient magnitude is given by:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

- Typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

- The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(Gy/Gx)$$

Robert's cross operator:

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point.

The operator consists of a pair of 2×2 convolution kernels as shown in Figure 3. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator.

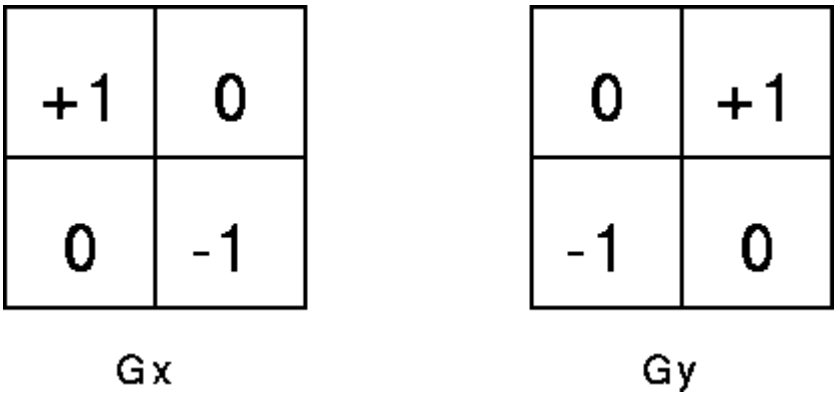


Figure 3: Kernel for Robert's operator

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient.

- The gradient magnitude is given by

$$|G| = \sqrt{Gx^2 + Gy^2}$$

- Approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

- The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan(Gy/Gx) - 3\pi/4$$

Prewitt's operator:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Canny's Edge Detection Algorithm:

The Canny edge detection algorithm is known to many as the optimal edge detector. The canny edge detector first smoothens the image to eliminate the noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed.

Hough transform

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Line detection is used to detect the presence of lines in an image, at a particular orientation. The importance of line detection is used for detecting sharp changes in image brightness.

The *classical* Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc.* A *generalized* Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible.

The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

A convenient equation for describing a set of lines uses *parametric* or *normal* notion:

$$x\cos\theta + y\sin\theta = r$$

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 as shown in the figure 1. For any point *(x, y)* on this line, ***r*** and ***θ*** are constant.

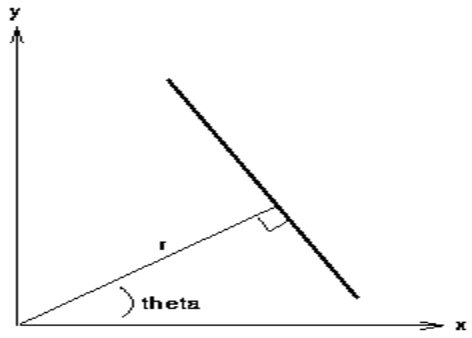


Figure 4: Parametric description of a straight line

A line in the (x, y) space will thus be represented by a point in (r, θ) space. In an image analysis context, the coordinates of the point(s) of edge segments (*i.e.*(x_i,y_i)) 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 (x_i,y_i), 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 transformation for straight lines.

The transform is implemented by quantizing the Hough parameter space into finite intervals or *accumulator cells*. As the algorithm runs, each (x_i,y_i) is transformed into a discretized (***r***,***θ)*** curve and the accumulator cells which lie along this curve are incremented as shown in the figure 2. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

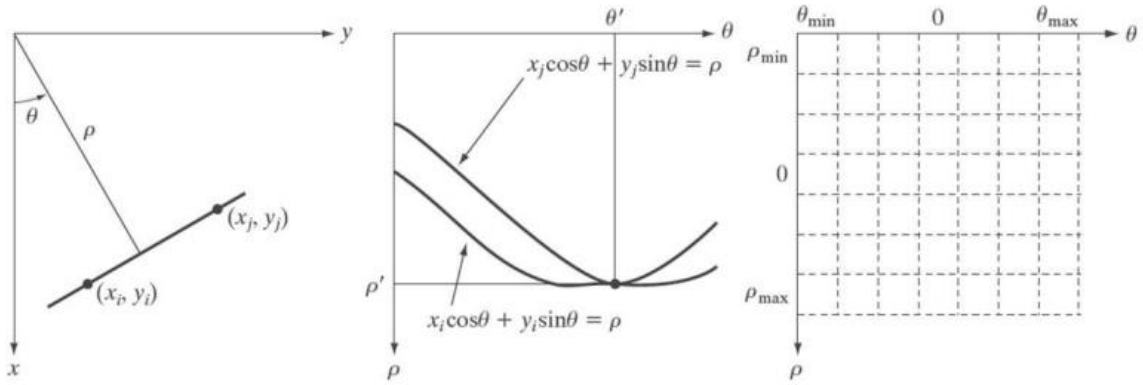


Figure 5:

(a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.

MCQs:

1. _____is set of connected pixel that lie on the boundary between two regions.
   (a) Point (b)edge (c)colour (d) line
2. the Hough transform is used to fit points as _____
   (a) line (b) edge (c)curve (d)ROI
3. Discontinuity approach of segmentation depends upon
   (a) Low frequencies (b) smooth changes (c) abrupt changes (d) contrast

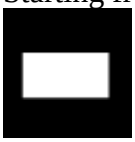
Exercise:

1. Write a MATLAB program to detect the sharp edges in an image in the horizontal direction, vertical direction and both.

2. Read an input image and compute the edges in the image using different edge detectors like Robert, Prewitt, Sobel and Canny. Comment on the result obtained.

3. Test Hough transform for equal size circles on the coins image.
(Hint: Use Matlab inbuilt Functions hough(), houghpeaks(), and houghlines())

4. Starting from the basic image



create a series of images with which you can investigate the ability of the Hough line detector to extract occluded features. For example, begin using translation and image addition to create an image containing the original image overlapped by a translated copy of that image. Next, use edge detection to obtain a boundary description of your subject. Finally, apply the Hough algorithm to recover the geometries of the occluded features.