LECTURE OUTLINE

- Image Features
- Edge Detection
  - Canny Edge Detector
  - Roberts Edge Detector
  - Sobel Edge Detector

Image Features

IMAGE FEATURES

In computer vision, the term *image feature* refers to two possible entities:

- Global feature
- Local feature

Local image features: Local, meaningful (associated to interesting scene elements), detectable parts of the image.

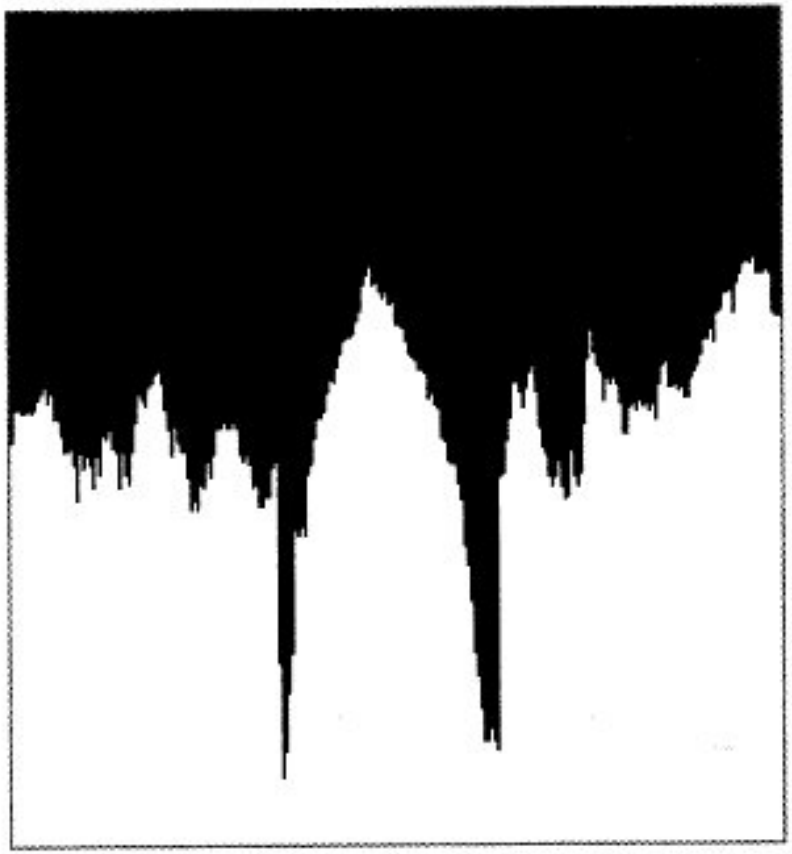
Edge Detection

Edges: Edge points (edgels, edge elements), or simply edges, are pixels at or around which the image values undergo a sharp variation.

Edge Detection: Given an image corrupted by acquisition noise, locate the edges most likely to be generated by scene elements, not by noise.

Edge Detection

(a)



(b)

Notice that image noise too causes intensity variations, which results in spurious edges.

Figure 4.1 (a) A 325 × 237-pixel image, with scanline $i = 56$ highlighted. (b) The intensity profile along the highlighted scanline. Notice how the main intensity variations indicate the borders of the hair region along the scanline.

Edge Detection

Edges: Edge points (edgels, edge elements), or simply edges, are pixels at or around which the image values undergo a sharp variation.

The term “edge” is also used to refer to connected chains of edge points, that is, contour fragments. Edge points are sometimes called edgels (for “edge elements”).

Edge Detection

Edge detection in computer vision is typically a 3-step process:

1) Noise smoothing
2) Edge enhancement
3) Edge localization
   - thinning wide edges to 1-pixel width
   - thresholding

Edge Detection

1) Noise smoothing: Suppress as much of the noise as possible, without destroying the true edges. In the absence of specific information, assume the noise is Gaussian.

2) Edge Enhancement: Design a filter responding to edges. The filter's output should be large at edge pixels and low elsewhere, so that edges can be located as the local maxima in the filter's output.

3) Edge Localization: Decide which local maxima in the filter's output are edges and which are just caused by noise. This involves:

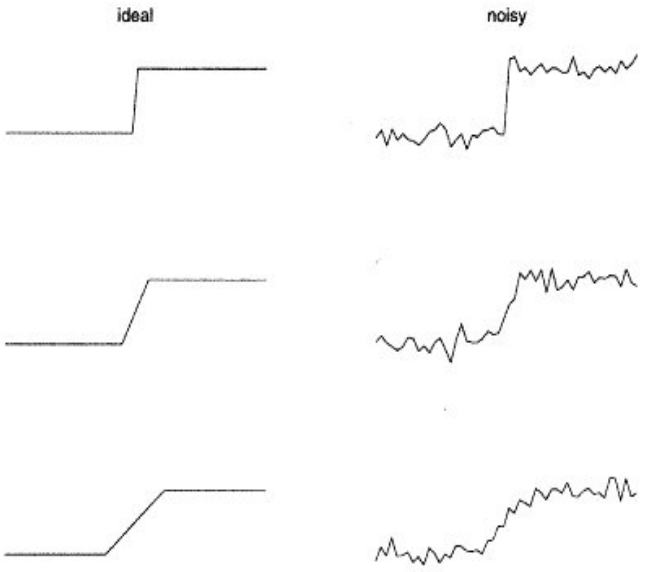
- thinning wide edges to 1-pixel width (*nonmaximum suppression*)
- establishing the minimum value to declare a local maxima an edge (*thresholding*)

Modeling Edges and Noise

Edges of intensity images can be modeled according to their intensity profiles.

- Step edges
- Ridge edges
- Roof edges

Modeling Edges and Noise

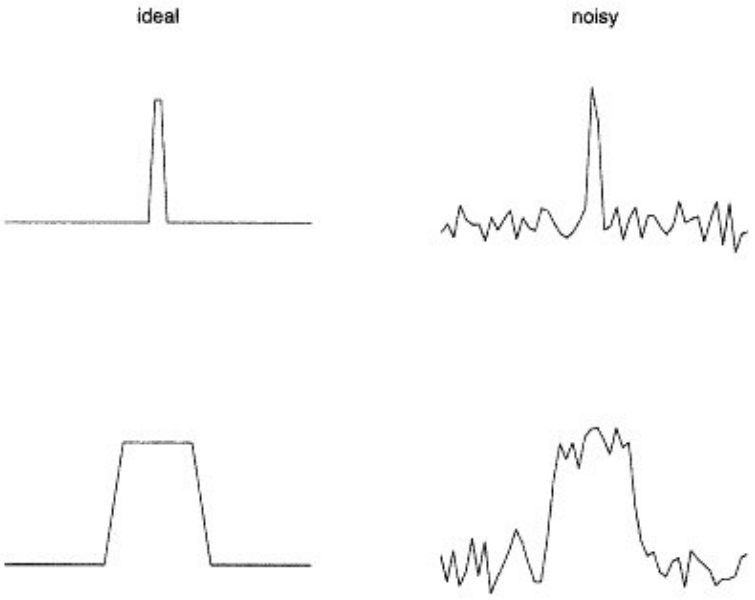


(a)

Left: ideal step (top, transition occurs over one pixel) and ramp edges. Right: corresponding noisy version, obtained by adding Gaussian noise (standard deviation 5% of the step height).

Step edges occur typically at contours of image regions of different intensities. When the transition occurs over several pixels, then they are called ramp edges.

Modeling Edges and Noise

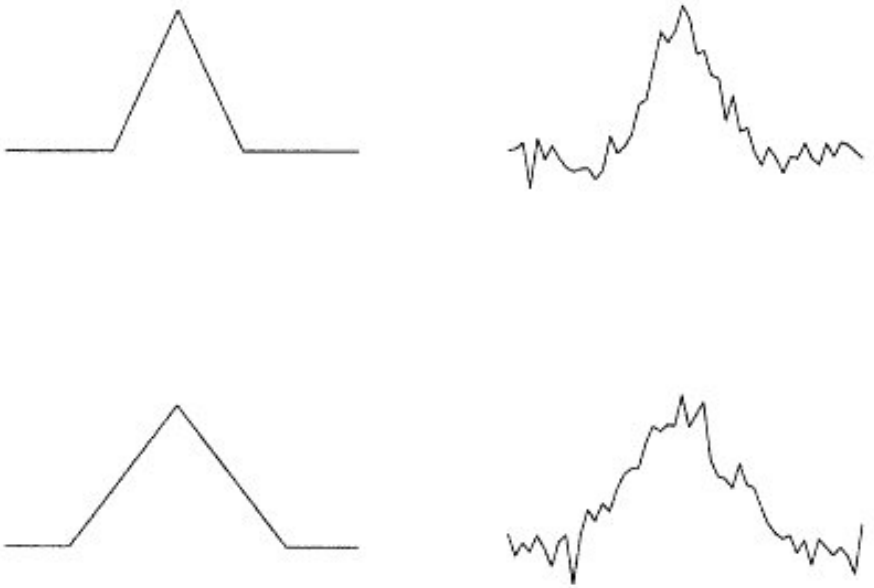


(b)

Left: ideal ridge edges. Right: corresponding noisy version, obtained as for step edges.

Ridge edges are generated by thin lines. Wide ridges can be modeled by two step edges.

Modeling Edges and Noise



(c)

Left: ideal roof edges. Right: corresponding noisy version, obtained as for step edges.

Roof edges may appear along the intersection of surfaces. (relatively rare)

Edge Descriptor

We will focus on step edge detectors.

Edge normal: The direction (unit vector) of the maximum intensity variation at the edge point. The direction that is perpendicular to the edge.

Edge direction: the direction that is perpendicular to the edge normal, and thus tangent to the contour of which the edge is part.

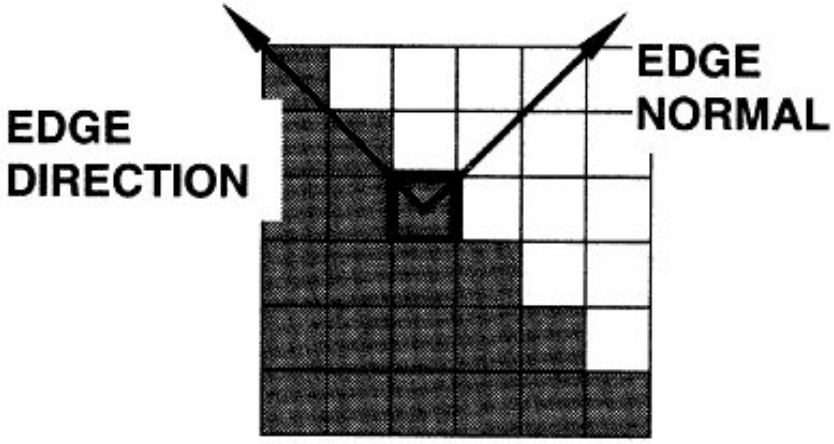


Figure 4.3 Illustration of edge normal and edge direction. The edge position considered is (2,2), with the origin in the upper left corner.

Edge Descriptor

Edge position or center: the image position at which the edge is located (1 for edge, 0 for no edge).

Edge strength: a measure of the local image contrast.

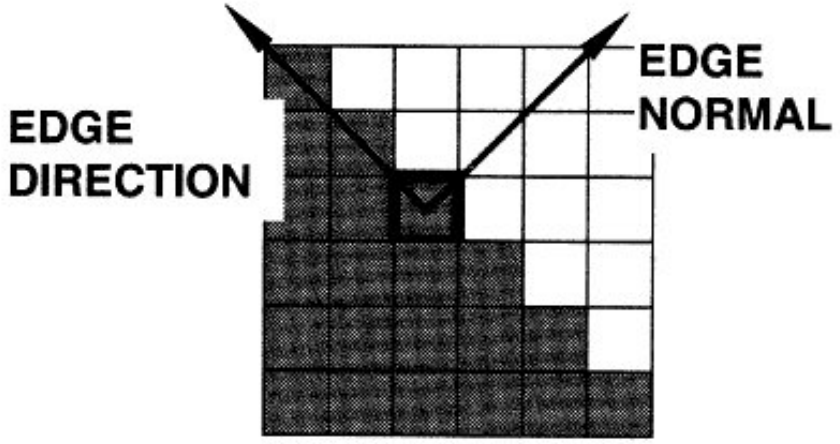


Figure 4.3 Illustration of edge normal and edge direction. The edge position considered is (2,2), with the origin in the upper left corner.

Criteria for Optimal Edge Detection:

1. Good Detection
2. Good Localization
3. Single-response Constraint

The localization-detection Tradeoff:

We can reach an optimal compromise between the location and detection criteria by adjusting the filter's spatial scale, but we cannot improve both criteria simultaneously.

A very good approximation of the ideal step edge detector is the first derivative of a Gaussian.

Algorithm CANNY_ENHANCER

1. Apply Gaussian smoothing to the input image I, obtaining J = I*G (where G is a Gaussian with zero mean and std. deviation σ).
2. For each pixel (i,j):
   a. Compute the gradient components, J_x and J_y.
   b. Estimate the edge strength, $e_s(i,j)$.
   c. Estimate the orientation of the edge normal, $e_o(i,j)$.

The output is a strength image, E_s, and an orientation image E_o.

Algorithm CANNY_ENHANCER

$$e_s(i, j) = \sqrt{J_x^2(i, j) + J_y^2(i, j)}$$

$$e_o(i, j) = \arctan \frac{J_y}{J_x}$$

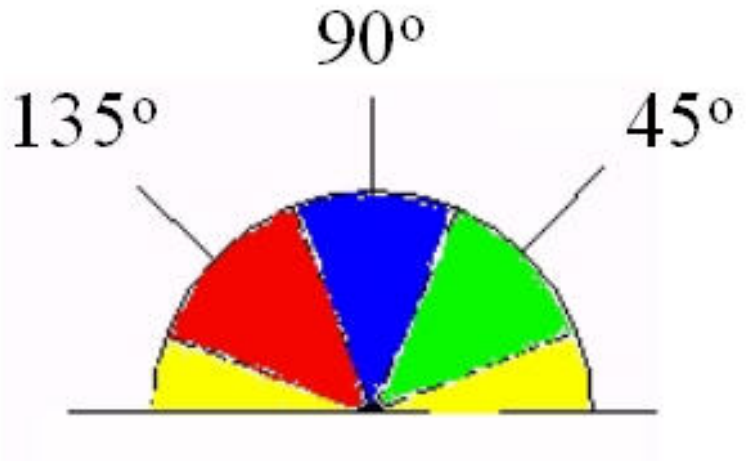
Algorithm NONMAX_SUPPRESSION

The strength image may contain wide ridges around the local maxima. To produce 1-pixel wide edges:

For each pixel (i,j):

1. Find the direction $\hat{d}_k$ (out of 0, 45, 90 and 135 degrees), which best approximates the direction $E_o(i,j)$.
2. If $E_s(i,j)$ is smaller than at least one of its two neighbors along $\hat{d}_k$, assign $I_N(i,j) = 0$ (suppression); otherwise assign $I_N(i,j) = E_s(i,j)$.

Canny Edge Detector



Any edge direction falling within the **yellow range** (0 to 22.5 and 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the **green range** (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the **blue range** (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the **red range** (112.5 to 157.5 degrees) is set to 135 degrees.

Algorithm NONMAX_SUPPRESSION

Figure 4.5 Strength images output by CANNY_ENHANCER run on Figure 4.1, after nonmaximum suppression, showing the effect of varying the filter's size, that is, the standard deviation, σ_f, of the Gaussian. Left to right: $\sigma_f = 1, 2, 3$ pixel.

Algorithm HYSTERESIS_THRESH

The input is I_N ,(the output of NONMAX_SUPPRESSION), the edge orientation image, and two thresholds (τ_l and τ_h) such that $\tau_l < \tau_h$. For all edge points in I_N:

1. Locate the next unvisited edge pixel $I_N(i,j)$ such that $I_N(i,j) > \tau_h$;
2. Starting from $I_N(i,j)$, follow the chains of connected local maxima, in both directions perpendicular to the edge normal, as long as $I_N(i,j) > \tau_l$. Mark all visited points, and save a list of the locations of all points in the connected contour found.

Note that hysteresis thresholding performs edge tracking. It finds chains of connected contours.

Algorithm HYSTERESIS_THRESH

Figure 4.6 Output of HYSTERESIS_THRESH run on Figure 4.5, showing the effect of varying the filter's size. Left to right: $\sigma_f = 1, 2, 3$ pixel. The grey levels has been inverted (black on white) for clarity.