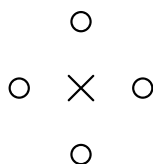


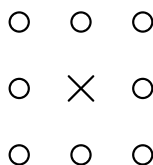
2-D Neighborhoods

- 4-point neighborhood



$$\partial(i, j) = \{(i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1)\}$$

- 8-point neighborhood



$$\partial(i, j) = \left\{ \begin{array}{l} (i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1) \\ (i + 1, j + 1), (i - 1, j + 1) \\ (i + 1, j - 1), (i - 1, j - 1) \end{array} \right\}$$

- More generally, a *Neighborhood System* is any mapping with the two properties that:
 1. For all $s \in S$, $s \notin \partial s$
 2. For all $r \in S$, $r \in \partial s \Rightarrow s \in \partial r$

Boundary Conditions

- How do you process pixels on the boundary of an image??
- Consider the following example using a 4-point neighborhood

A small example image

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>
<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>

4 neighbors of *l*

	<i>l</i> ₁	
<i>l</i> ₄	<i>l</i>	<i>l</i> ₂
	<i>l</i> ₃	

- Free boundary condition

$$\begin{aligned}\partial l &= \{h, p, k\} \\ \partial p &= \{l, o\}\end{aligned}$$

- Toriodal boundary condition (asteroids)

$$\begin{aligned}\partial l &= \{h, i, p, k\} \\ \partial p &= \{l, m, d, o\}\end{aligned}$$

- Reflective boundary condition

$$\begin{aligned}l_1 &= h, \quad l_2 = k, \quad l_3 = p, \quad l_4 = k \\ p_1 &= l, \quad p_2 = o, \quad p_3 = l, \quad p_4 = o\end{aligned}$$

Edge Detection

- Edges
 - Edges naturally occur in images due to the discontinuities form by occlusion.
 - Edges often delineate the boundaries between distinct regions.
 - Edges often contain important visual and semantic information.
- Edge detection:
 - The process of identifying pixels that fall along edges.
 - As with any detect process subject to a trade-off between false alarm and miss detection rates.
- Performance Metrics:
 - Evaluation of edge detection schemes can be difficult.
 - Correct labeling of edge and non-edge pixels often requires subjective interpretation.
 - Best choice of edge detection scheme usually depends on task.
 - Performance metrics exist and usually use synthetic data input for evaluation.

Gradient Based Edge Detection

- Compute local estimate of gradient

$$\nabla f(x, y) = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

- From these, compute gradient magnetude and angle.

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

- Apply threshold to the magnitude of gradient

$$\text{edge} \quad |\nabla f| \geq T$$

$$\text{no edge} \quad |\nabla f| < T$$

- Choosing T
 - Too large \Rightarrow missed detections
 - Too small \Rightarrow false alarms

How to Compute Gradient

- Directional derivatives can be computed by applying a spatial filter.
- Conventional (off center)

$$\begin{bmatrix} \boxed{-1} & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} \boxed{-1} & 0 \\ 1 & \end{bmatrix}$$

- Roberts (off center)

$$\begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix}$$

- Prewitt (on center)

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Sobel (on center)

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Edge Thining

- Thresholding of gradient magnitude generally produces a thick edge.
- Edge should be thinned to produce most accurate result.

1. Set $S = \{s : |\nabla f(s)| \geq T\}$
2. Set $D = \emptyset$ (detected edge points)
3. For each $s \in S$
 - (a) Compute θ = gradient direction at s .
 - (b) Select out P = all pixels in direction θ starting at s within maximum distance d_{max} from s .
 - (c) If $|\nabla f(s)| \geq \max_{p \in P} \{|\nabla f(p)|\}$, then

$$D \leftarrow D + \{s\}$$

