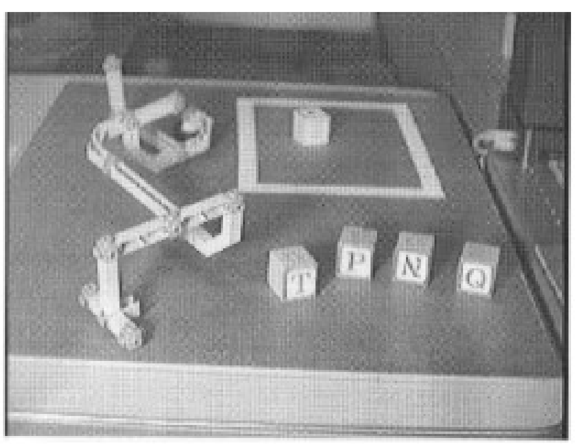
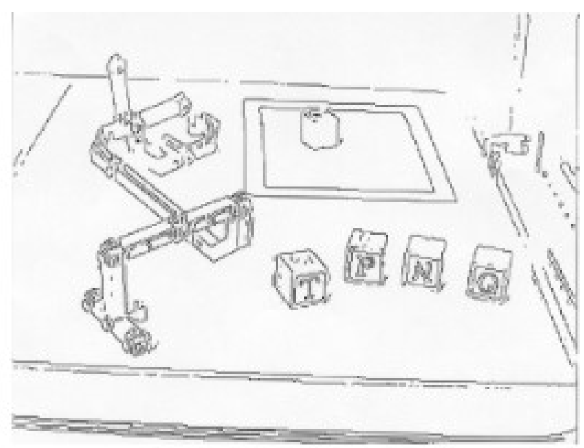
# CANNY EDGE DETECTION AND DETECTION OF CORNERS

# EDGES

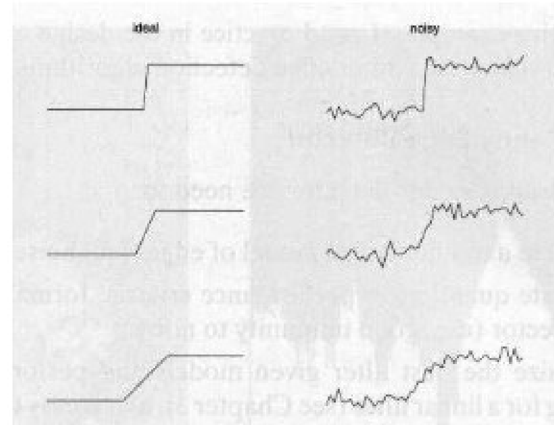
- Edges are **significant local changes of intensity** in an image.
- Edges typically occur on the boundary between two different regions in an image.
- Goals:
  - Produce a line drawing of a scene from an image of that scene.
  - Important features can be extracted from the edges of an image (e.g., corners, lines, curves).
  - These features are used by higher-level computer vision algorithms (e.g., recognition).

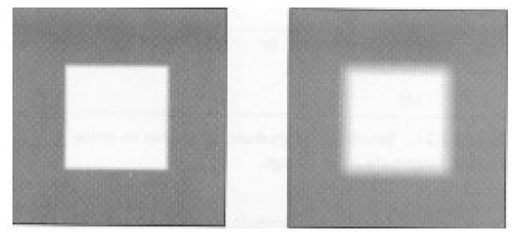


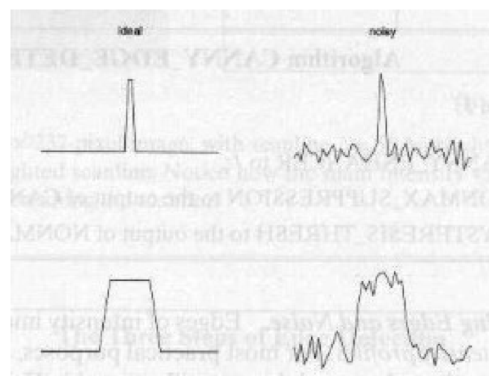


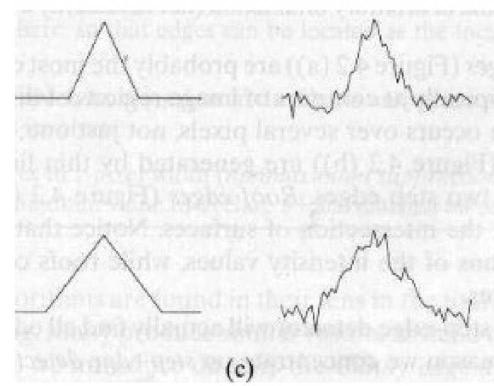
# TYPES OF EDGES

- Step Edge: the image intensity abruptly changes from one value to one side of the discontinuity to a different value on the opposite side
- Ramp Edge: a step edge where the intensity change is not instantaneous but occur over a finite distance.
- Ridge (Line)  Edge: the image intensity abruptly changes value but then returns to the starting value within some short distance (generated usually by lines).
- Roof Edge: a ridge edge where the intensity change is not instantaneous but occur over a finite distance (generated usually by the intersection of surfaces)





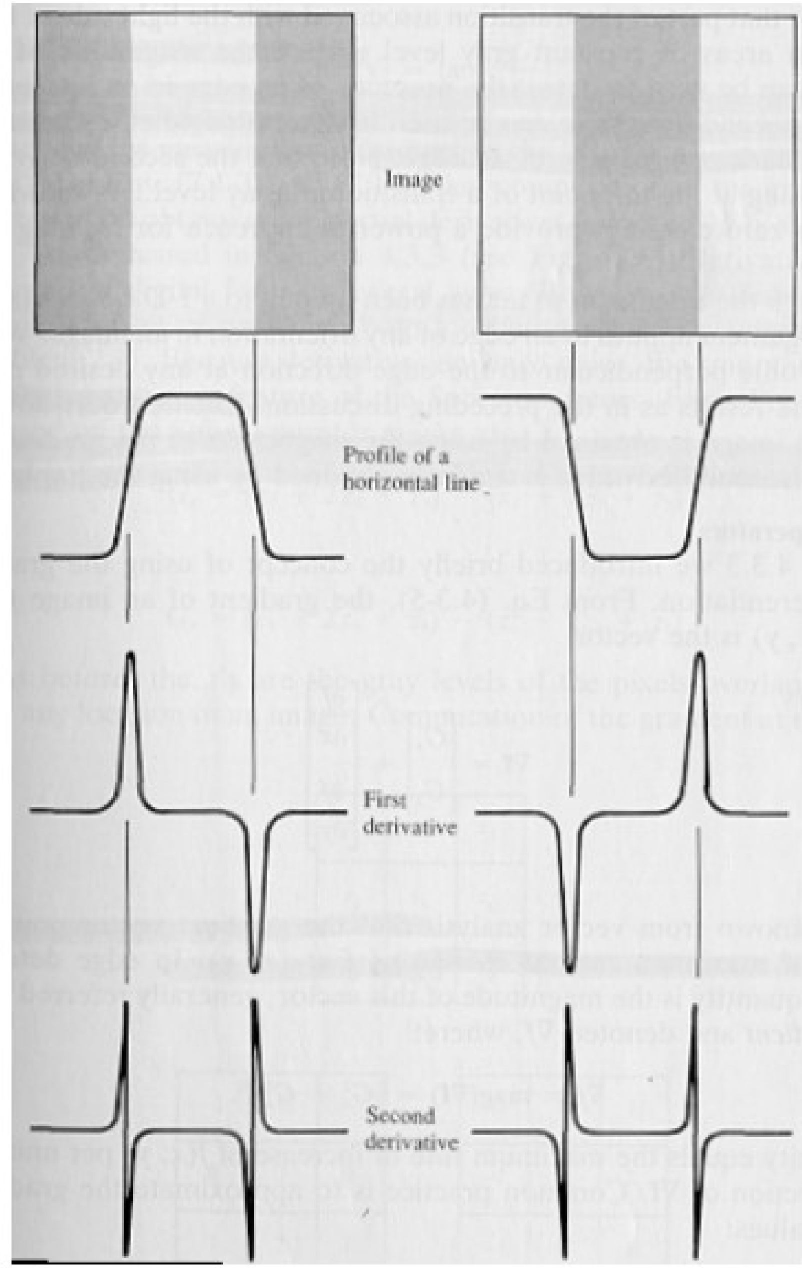



# STEPS OF EDGE DETECTION

1. **Smoothing**: suppress as much noise as possible, without destroying the true edges.
2. **Enhancement**: apply a filter to enhance the quality of the edges in the image (sharpening).
3. **Detection**: determine which edge pixels should be discarded as noise and which should be retained (usually, thresholding provides the criterion used for detection).
4. **Localization**: determine the exact location of an edge (sub-pixel resolution might be required for some applications, that is, estimate the location of an edge to better than the spacing between pixels). Edge thinning and linking are usually required in this step

# EDGE DETECTION USING DERIVATIVES

- An image is a 2D function, so operators describing edges are expressed using partial derivatives.
- Points which lie on an edge can be detected by:
  1. detecting local maxima or minima of the first derivative
  2. detecting the zero-crossing of the second derivative



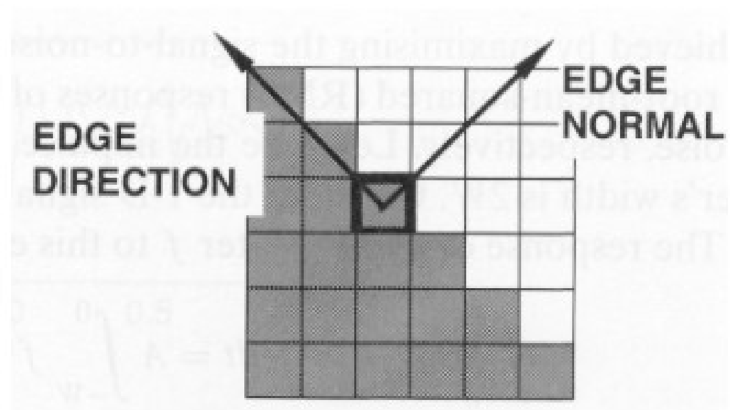
# EDGE AND GRADIENT

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

- The gradient is a vector which has certain magnitude and direction:

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

- The magnitude of gradient provides information about the strength of the edge.
- The direction of gradient is always perpendicular to the direction of the edge (the edge direction is rotated with respect to the gradient direction by -90 degrees).




$G_x$ $\begin{bmatrix} -1 & 1 \end{bmatrix}$

$G_y$ $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

# ROBERT'S METHOD

- The **Roberts filter** is one of the simplest edge detection operators, using **2×2 kernels** to approximate gradients.

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

$G_x = (z_9 - z_5)$ and $G_y = (z_8 - z_6)$

$$\nabla f = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

| | |
|---|---|
| -1 | 0 |
| 0 | 1 |

| | |
|---|---|
| 0 | -1 |
| 1 | 0 |

# ROBERT'S METHOD

- It is computationally efficient and ideal for real-time applications.
- It is sensitive to **diagonal** edge detection.
- Less accurate for detecting edges in smooth regions due to its **small kernel size**.
- **Sensitive to noise** because it uses a small neighborhood.

# SOBEL'S METHOD

- Mask of even size are awkward to apply.
- The smallest filter mask should be 3x3.
- The difference between the third and first rows of the 3x3 mage region approximate derivative in x-direction, and the difference between the third and first column approximate derivative in y-direction.

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

# SOBEL'S METHOD

- it smooths out noise better than Roberts.
- It enhances edges along both **X and Y directions**.
- Computationally more expensive than Roberts.
- Less effective in detecting fine, high-frequency edges compared to smaller operators.

# CANNY EDGE DETECTION

1. Smooth image with a Gaussian filter
2. Approximate gradient magnitude and angle (use Sobel, Prewitt . . .)

$$M[x,y] \approx \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\alpha[x,y] \approx \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

3. Apply nonmaxima suppression to gradient magnitude
4. Double thresholding to detect strong and weak edge pixels
5. Reject weak edge pixels not connected with strong edge pixels

*[Canny, IEEE Trans. PAMI, 1986]*

# SMOOTHING USING GAUSSIAN KERNEL

5 x 5 Gaussian kernel

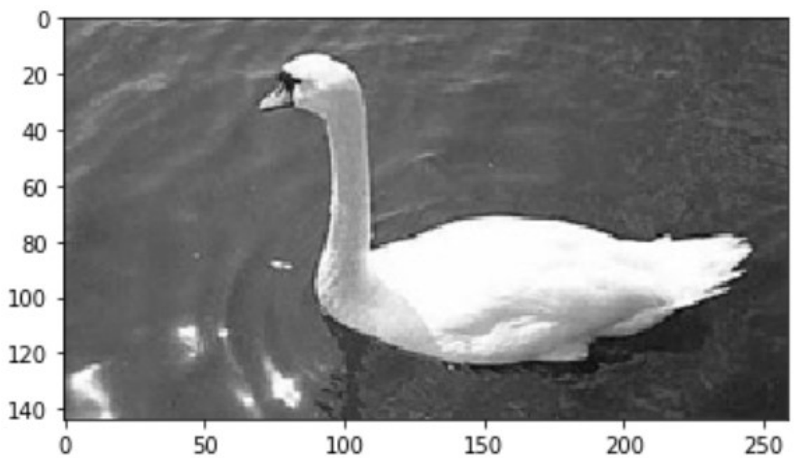
$$\frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

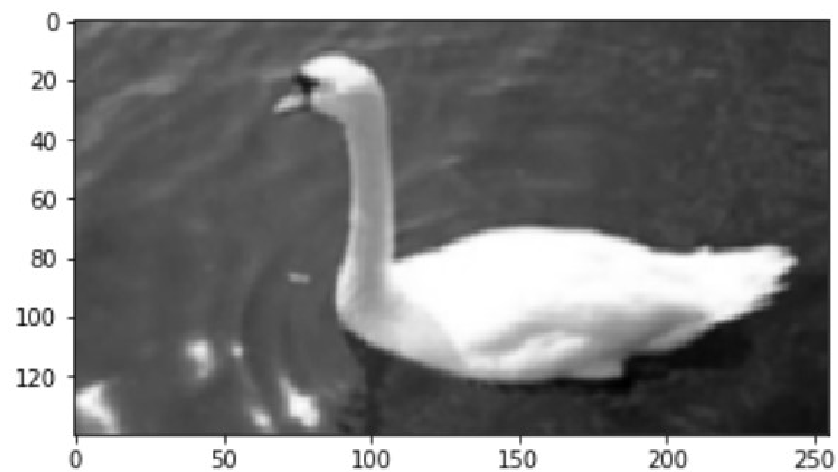
$$\frac{1}{273}$$

| | | | | |
|---|---|---|---|---|
| 1 | 4 | 7 | 4 | 1 |
| 4 | 16 | 26 | 16 | 4 |
| 7 | 26 | 41 | 26 | 7 |
| 4 | 16 | 26 | 16 | 4 |
| 1 | 4 | 7 | 4 | 1 |

$$Filter : (2k+1) \times (2k+1) \qquad -2 \le k \le 2$$

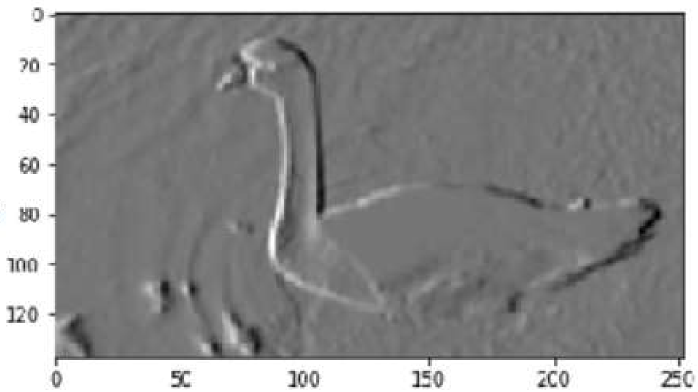
$$x = i - (k+1); y = j - (k+1) \qquad 1 \le i, j \le 2k+1$$



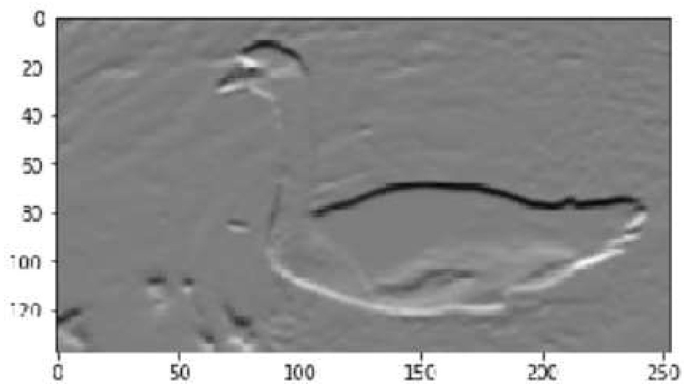



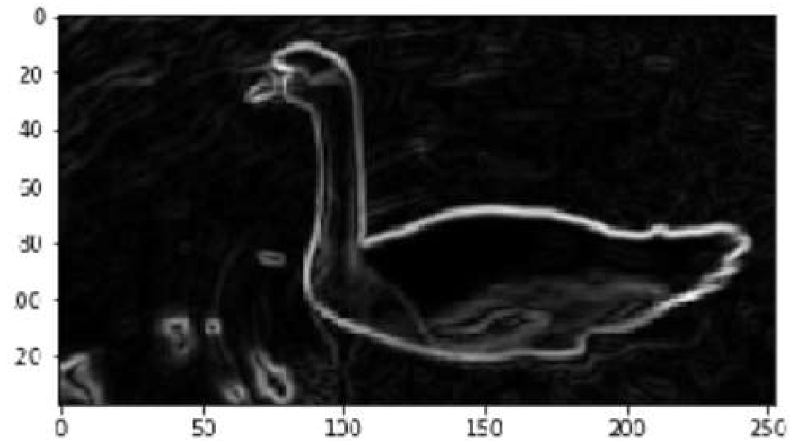
# MAGNITUDE

$F_x$



$F_y$

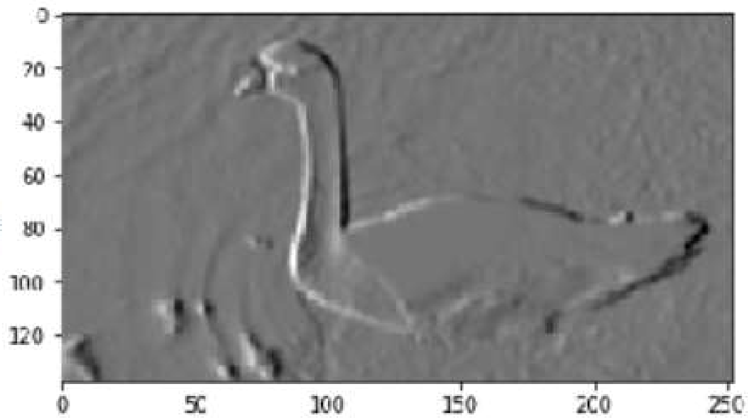


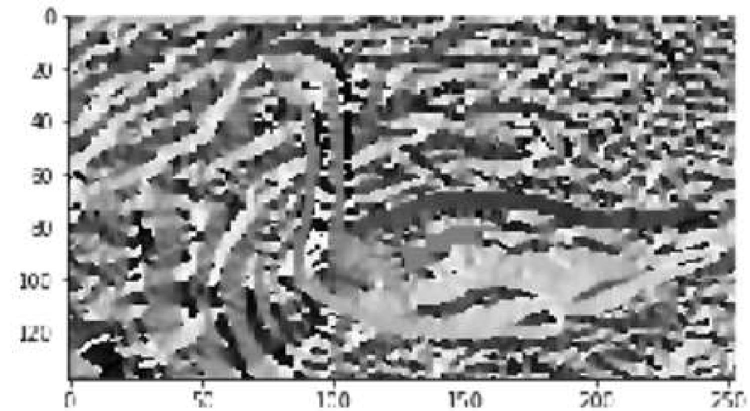


$$G = \sqrt{(F_x^2 + F_y^2)}$$

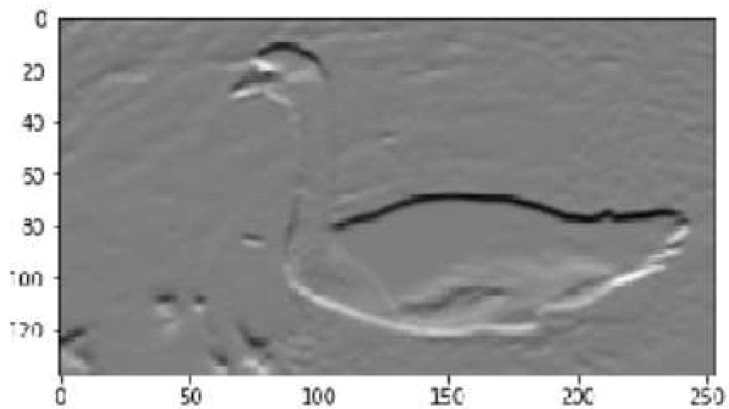
# ORIENTATION

$F_x$





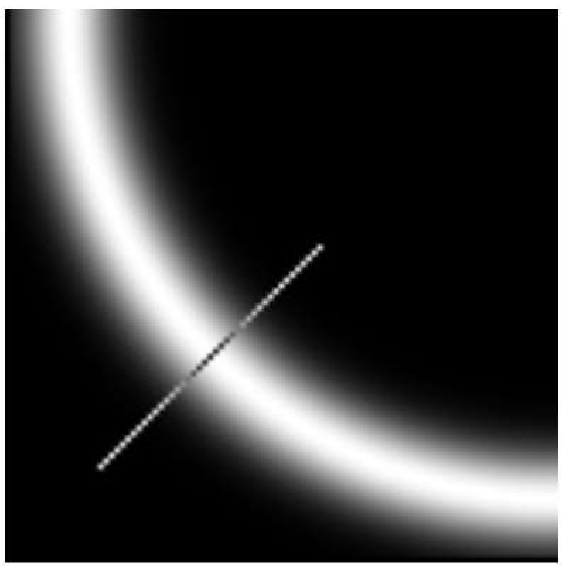
$F_y$

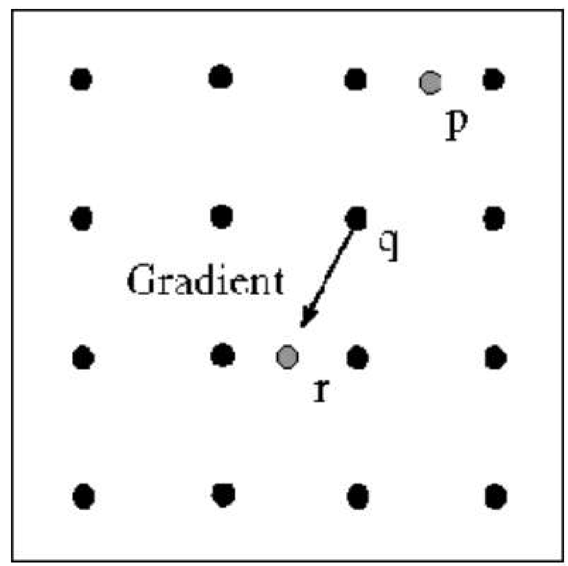


$$\theta = tan^{-1}\left(\frac{F_y}{F_x}\right)$$

# NON-MAXIMUM SUPPRESSION

- Thin the edges by keeping only pixels with the highest gradient magnitude along the edge direction.
- Thin multi-pixel wide "ridges" down to single pixel width






Check if pixel is local maximum along gradient direction

- requires checking interpolated pixels p and r

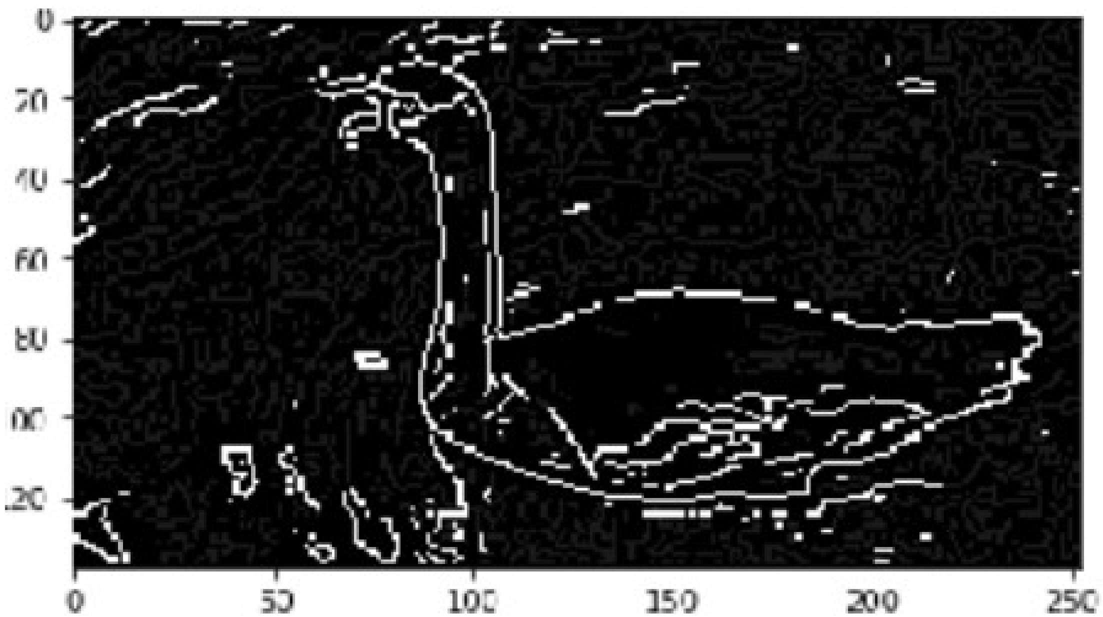
# THRESHOLDING AND LINKING

- Define two thresholds: low and high
- Upper threshold based on the max intensity.
- lower threshold based on some percentage of the upper threshold.
- Example:
  - Upper threshold - 90% of max
  - lower threshold - 35%

| | | |
|---|---|---|
| <= lower threshold | lower threshold < intensity < upper threshold | >= upper threshold |
| irrelevant = 0 | weak = low threshold | strong= 255 |

# THRESHOLDING AND LINKING

- replace with the strong edge if any of the neighboring pixels is strong, else make it irrelevant.



# EXAMPLE: CANNY EDGE DETECTION

Original image

gap is gone

Strong + connected weak edges

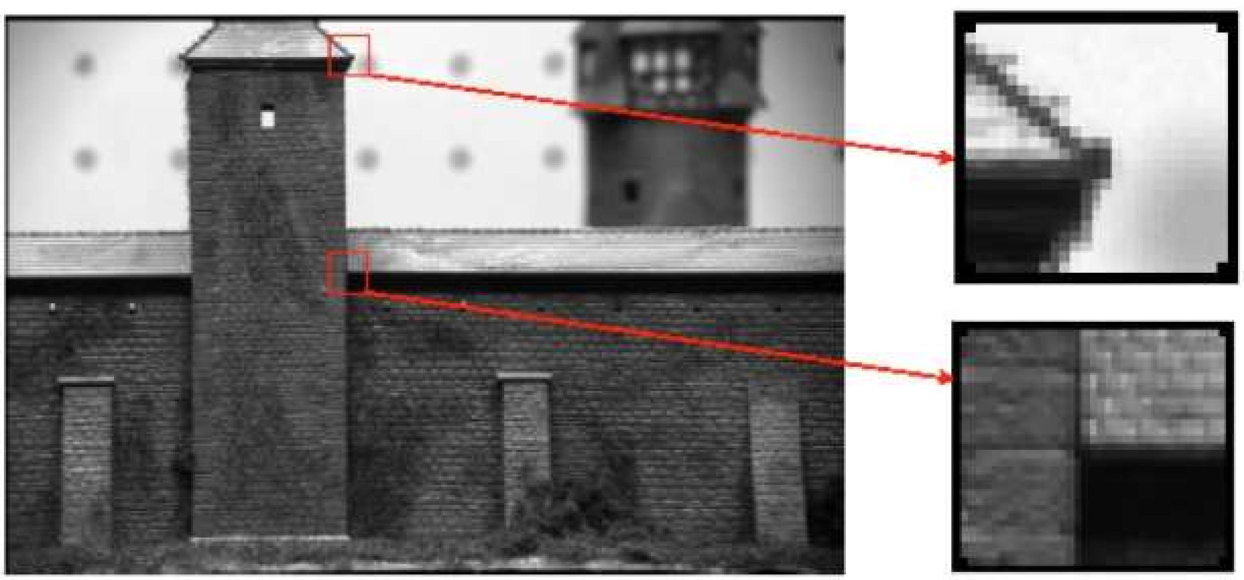
Strong edges only

Weak edges

courtesy of G. Loy

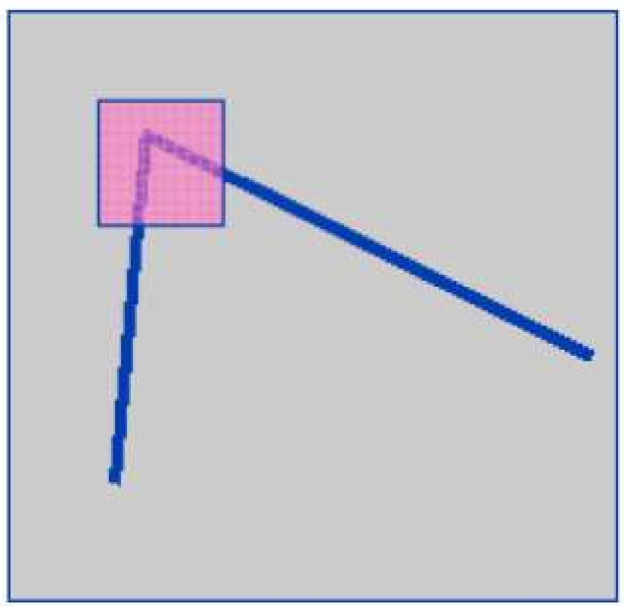
# DETECTION OF CORNERS (INTEREST POINTS)

- A corner as a rapid change of direction in a curve.
- Corners are highly effective features because they are distinctive and reasonably invariant to viewpoint.
- Corners are unique, match patches with corners
- What are corners? Junctions of contours (edges)
- Corners appear as large changes in intensity in different viewpoints (stable/unique)

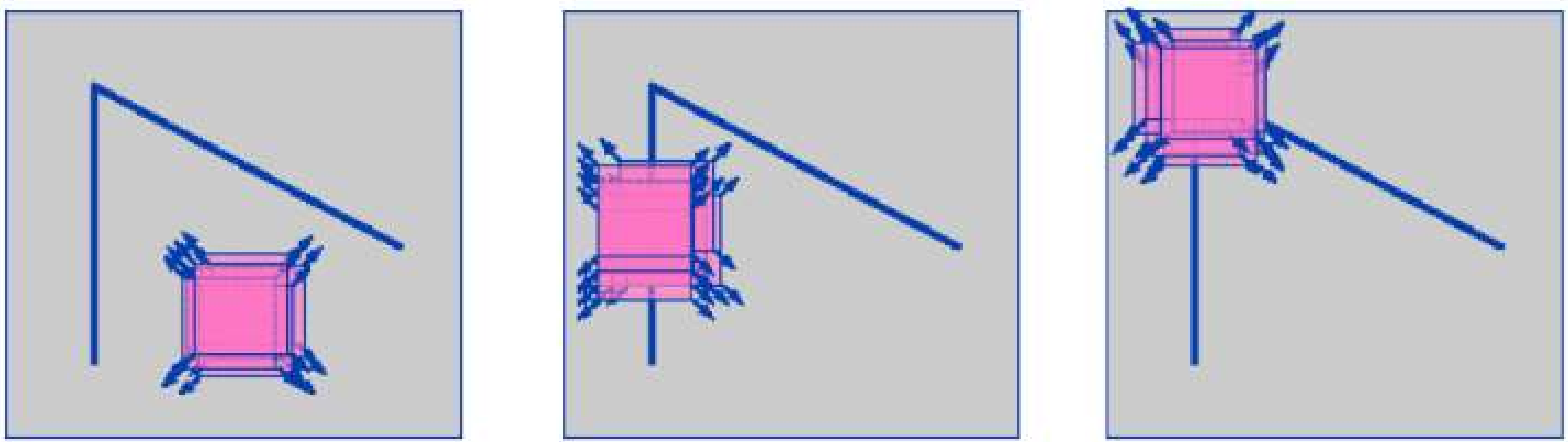


# BASIC IDEA: CORNER POINT DETECTION

- **Patch at corner:** shifting window in any direction yields large change in intensity
- **Is patch at corner?** shift window in multiple directions,
- if large intensity changes, patch is a corner



# HARRIS CORNER DETECTION: BASIC IDEA



**Flat region:** no intensity change in all directions

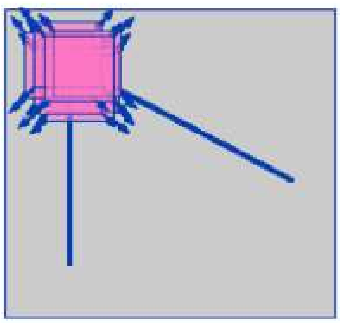
**Edge:** no change along edge directions

**Corner:** significant intensity change in many directions

- Harris corner detector gives mathematical approach for determining which case holds

# HARRIS CORNER DETECTOR (MATH)

- Shift patch by [*u,v*] and compute change in intensity
- Change of intensity for shift [*u,v*]



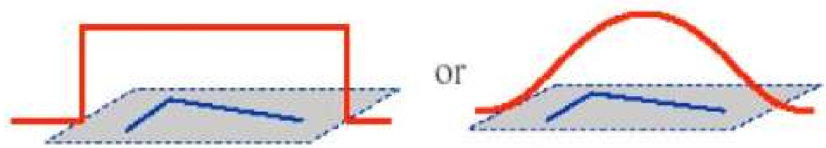
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

Window function $w(x,y)$ =




1 in window, 0 outside

Gaussian

# HARRIS DETECTOR: INTUITION

- Change of intensity for shift [*u*,*v*]

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

**For nearly flat regions:** this quantity almost 0
**For distinctive (corner) :** this quantity almost large
Hence... Corners? Patches where E(u,v) is **LARGE**

# HARRIS DETECTOR: APPROXIMATION USING TAYLOR SERIES

$$f(x+u,y+v) = f(x,y) + uf_x(x,y) + vf_y(x,y) +$$

**First partial derivatives**

$$\frac{1}{2!}\left[u^2 f_{xx}(x,y) + uvf_{xy}x,y + v^2 f_{yy}(x,y)\right] +$$

**Second partial derivatives**

$$\frac{1}{3!}\left[u^3 f_{xxx}(x,y) + u^2 v f_{xxy}(x,y) + uv^2 f_{xyy}(x,y) + v^3 f_{yyy}(x,y)\right]$$

**Third partial derivatives**

$$+ \ \ldots \text{ (Higher order terms)}$$

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

$$\approx \sum \left[I(x,y) + uI_x + vI_y - I(x,y)\right]^2$$

**First order approximation**

# HARRIS DETECTOR: APPROXIMATION USING TAYLOR SERIES

$$\sum [I(x+u, y+v) - I(x,y)]^2$$

$$\approx \sum [I(x,y) + uI_x + vI_y - I(x,y)]^2 \quad \text{First order approx}$$

$$= \sum u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2$$

$$= \sum \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{Rewrite as matrix equation}$$

$$= \begin{bmatrix} u & v \end{bmatrix} \left( \sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

# HARRIS CORNER DETECTION

- For small shifts [*u, v*], we have following approximation

$$E(u,v) \cong [u,v] \;\; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where *M* is the 2x2 matrix

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

# HARRIS CORNER DETECTION

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- If we compute derivatives *A(u,v)*, *B(u,v)* and *C(u,v)*

$$A(u,v) = I_x^2(u,v)$$

$$B(u,v) = I_y^2(u,v)$$

$$C(u,v) = I_x(u,v) \cdot I_y(u,v)$$

- Can express matrix *M* as

$$\boldsymbol{M} = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} = \begin{pmatrix} A & C \\ C & B \end{pmatrix}$$

# HARRIS CORNER DETECTION

- Smooth *A(u,v), B(u,v)* and *C(u,v)* individually with linear gaussian filter

$$\bar{M} = \begin{pmatrix} A * H^{G,\sigma} & C * H^{G,\sigma} \\ C * H^{G,\sigma} & B * H^{G,\sigma} \end{pmatrix} = \begin{pmatrix} \bar{A} & \bar{C} \\ \bar{C} & \bar{B} \end{pmatrix}$$

- Since the matrix $\overline{M}$ is symmetric, it can be diagonalized

$$\bar{M}' = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

where $\lambda_1, \lambda_2$ are eigenvalues of matrix $\overline{M}$ defined as:

$$\begin{aligned} \lambda_{1,2} &= \frac{\operatorname{trace}(\bar{M})}{2} \pm \sqrt{\left(\frac{\operatorname{trace}(\bar{M})}{2}\right)^2 - \det(\bar{M})} \\ &= \frac{1}{2}\left(\bar{A} + \bar{B} \pm \sqrt{\bar{A}^2 - 2\bar{A}\bar{B} + \bar{B}^2 + 4\bar{C}^2}\right) \end{aligned}$$

# HARRIS CORNER DETECTION EXAMPLE