

Features

Feature types:

- 1) global : e.g. color distribution (histograms)
- 2) local : e.g. edges or corners

Goal:

characterize images
and/or
local neighborhoods

Edge detection

Requirements:

- correspond to scene elements
- invariant (illumination, pose, viewpoint, scale)
- reliable detection

Edge detection

edge = location with change in image



Edge detection steps

- 1) Smooth to reduce noise
(without affecting edges)
- 2) Enhance edges
- 3) Detect edges
- 4) Localize edges

Image gradient

Image: $I(x, y)$

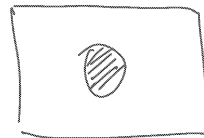
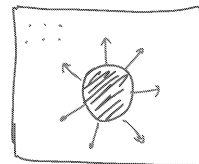


Image gradient:

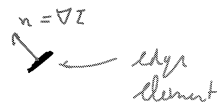
$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

$$= \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$



Edge detection using gradients

"edge" = edge element



magnitude: $|\nabla I| = \sqrt{I_x^2 + I_y^2}$

angle: $\theta = \tan^{-1} \left(\frac{I_y}{I_x} \right)$

*Edge detection:

edge map $\rightarrow E(i, j) = \begin{cases} 1 & \text{if } |\nabla I(i, j)| > \tau \\ 0 & \text{otherwise} \end{cases}$

[illegible]

This image shows a blank sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

Central differences

$$\frac{\partial}{\partial x} I(x, y) \approx \frac{I(x+h, y) - I(x-h, y)}{2h}$$
$$= \frac{I(x+1, y) - I(x-1, y)}{2}$$

Sobel filter

Smooth and then take derivative

$$\Delta_y = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The diagram shows a 3x3 matrix A on the left, with elements a_{ij} and b_{ij} in the first and second rows respectively. An arrow labeled Δx points from A to a 3x3 matrix on the right, which represents $\text{Im}(A)$. This matrix has elements b_{ij} in the first row and a_{ij} in the second row. A second arrow labeled Δy points from the $\text{Im}(A)$ matrix to a 3x3 matrix on the bottom right, which represents A again, with elements a_{ij} in the first row and b_{ij} in the second row. The word "imag" is written below the first matrix.

Diagram illustrating the integration by parts formula using the product rule. The diagram shows four graphs arranged in a 2x2 grid, connected by arrows indicating the relationships between the functions and their derivatives.

- Top Left:** Graph of $f(x)$ vs x .
- Top Right:** Graph of $f(x) \cdot h(x)$ vs x . The area under the curve is shaded.
- Bottom Left:** Graph of $f'(x)$ vs x .
- Bottom Right:** Graph of $f'(x) \cdot h'(x)$ vs x . The area under the curve is shaded.

Arrows and labels indicating the relationships:

- Horizontal arrow from $f(x)$ to $f(x) \cdot h(x)$: $f(x) \cdot h'(x)$
- Vertical arrow from $f(x)$ to $f'(x)$: $f'(x)$
- Horizontal arrow from $f'(x)$ to $f'(x) \cdot h'(x)$: $f'(x) \cdot h(x)$
- Vertical arrow from $f(x) \cdot h(x)$ to $f'(x) \cdot h'(x)$: $-f(x) \cdot h'(x)$

The final result is:

$$f(x) \cdot h(x) - \int f'(x) \cdot h'(x) dx$$

Perfect interpolator: under certain conditions
(Nyquist) perfect reconstruction. $f_s \geq 2f_m$

$$h(x) = \text{sinc}(x) = \frac{\sin(\pi x/T)}{(\pi x/T)}$$



approximate $h(x)$ with $g(x)$

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on its right side, suggesting it's resting on a surface.This image shows a blank sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

Accurate derivatives

$$I_x = I * G_x \quad f[x] * g[x]$$

$$I_y = I * G_y$$

Using separable property of Gaussians:

1D convolution with horizontal Gaussian derivative

$$I_x = I * G'_x * G_y \leftarrow \text{1D convolution with vertical Gaussian}$$

$$I_y = I * G'_y * G_x \leftarrow \text{1D convolution with horizontal Gaussian}$$

1D convolution with vertical Gaussian derivative

Accurate derivatives

$$I_x = I * G'_x * G_y$$



$$I_y = I * G'_y * G_x$$

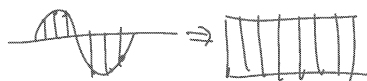


Gaussian derivatives

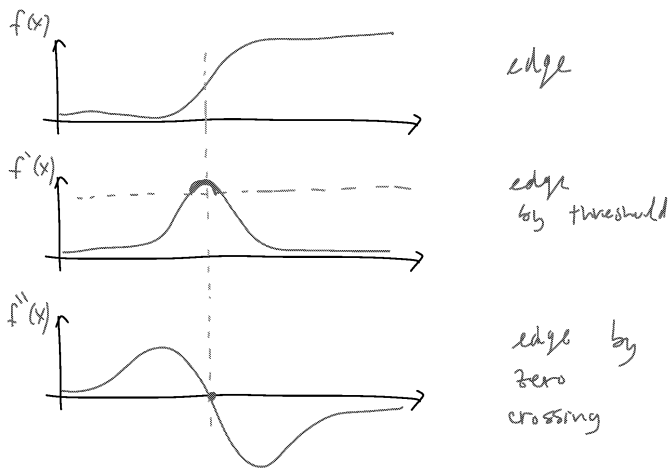
$$G(x) = e^{-\frac{x^2}{2\sigma^2}}$$



$$G'(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$



Laplacian of Gaussian



2D second derivative as scalar quantity

$$\text{Laplacian: } \Delta I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} = I_{xx} + I_{yy}$$

$$I_x = I(x+1) - I(x)$$

$$I_{xx} = (I(x+1) - I(x)) - (I(x) - I(x-1))$$

$$= I(x+1) - 2I(x) + I(x-1) \Rightarrow \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$


$$I_{yy} = \dots \Rightarrow \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$


$$I_{xx} + I_{yy} \Rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Laplacian of Gaussian (LOG)

Smooth with a Gaussian before applying Laplacian

$$H = \nabla^2 (I * G) = \underbrace{\nabla^2 G}_{\text{LOG}} * I$$

$$G = e^{-\frac{r^2}{2\sigma^2}} \quad (r^2 = x^2 + y^2)$$


$$\nabla^2 G = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}}$$


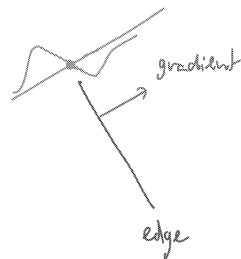
Edge detection using LOG

- 1) Compute LOG: $H = (D^2 G) * I$
- 2) Threshold : $E(i,j) = \begin{cases} 0 & \text{if } H(i,j) < 0 \\ 1 & \text{if } H(i,j) \geq 0 \end{cases}$
- 3) Mark edges at transitions $0 \rightarrow 1$
 $1 \rightarrow 0$
(scan left-to-right and top-to-bottom)

Canny edge detection

Detect edges at zero crossings of second order directional derivative taken along the gradient

$M = \text{gradient}$
if $|M| > \tau$ detect
edges at zero crossing
of $\frac{\partial^2}{\partial n^2} (I * G)$



Canny edge detection

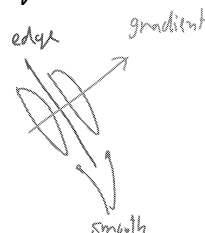
Comments

1) attempt detection only if gradient magnitude is large enough ($|M| > \tau$)

2) smooth along edges to preserve edge

3) Alternative to zero crossing
of $\frac{\partial^2}{\partial n^2} (I * G)$ is

maximum of $\frac{\partial}{\partial n} (I * G)$



Directional derivative expression

gradient: $\nabla = \nabla (I * G)$

directional derivative: $\frac{\partial}{\partial n} (I * G) \equiv \nabla \cdot \nabla (I * G)$

$$\nabla \cdot \nabla (I * G) = \nabla (I * G) \cdot \nabla (I * G)$$

$$= |\nabla (I * G)|^2$$

Canny summary

If $|\nabla (I * G)| > \tau$ set edge at maximum of $|\nabla (I * G)|$ along the direction of $\nabla (I * G)$

Additional components:

- 1) non-maximum suppression
- 2) hysteresis thresholding



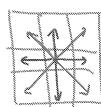
Non-maximum suppression

Need: local maximum of gradient magnitude in direction of gradient

$$\nabla (I * G) \equiv (I_x, I_y)$$

$$\theta = \tan^{-1} \left(\frac{I_y}{I_x} \right)$$

$$\theta^* = \text{round} \left(\frac{\theta}{45} \right) * 45$$



compare neighbors after discretization

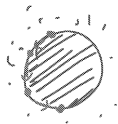
$$E(i,j) = \begin{cases} 1 & \text{if } \nabla (I * G) \text{ is a local maximum} \\ 0 & \text{otherwise} \end{cases}$$

Hysteresis thresholding

Use τ_H to start tracking and τ_L to continue ($\tau_H > \tau_L$)

1) Initialize array of visited pixels

$$v(i,j) = 0$$



2) Scan image T-B, L-R:

if $v(i,j)$ & $|\nabla I| > \tau_H$ start tracking an edge

3) Search for additional neighbors in direction orthogonal to ∇I such that $|\nabla I| > \tau_L$

Summary edge detection

* Edge detection requires finding discrete image derivatives:

- forward difference
 - central difference
 - Sobel
 - Gaussian derivatives
 - Zero crossing of LOG (2nd order)
 - Directional derivatives (Canny)
- } first order derivatives