Prewitt edge Detector

$$\boxed{\text{image}} \xrightarrow[\text{smoothing in } x]{\text{average}} \boxed{\text{blurred}} \xrightarrow[\text{filtering in } x]{\text{derivative}} \boxed{\text{edges in } x}$$

$$\begin{bmatrix}1\\1\\1\end{bmatrix} \longrightarrow \begin{bmatrix}1 & 0 & -1\end{bmatrix} \longrightarrow \begin{bmatrix}1 & 0 & -1\\1 & 0 & -1\\1 & 0 & -1\end{bmatrix}$$

$$\boxed{\text{image}} \xrightarrow[\text{filtering in } y]{\text{Derivative}} \boxed{\text{blurred}} \xrightarrow[\text{smoothing in } y]{\text{average}} \boxed{\text{edges in } y}$$

$$\begin{pmatrix}1\\0\\-1\end{pmatrix} \longrightarrow \begin{pmatrix}1 & 1 & 1\end{pmatrix} \longrightarrow \begin{pmatrix}1 & 1 & 1\\0 & 0 & 0\\-1 & -1 & -1\end{pmatrix}$$

(convolution operation)

Sobel Edge Detector

$$\boxed{\text{image}} \xrightarrow[\text{smoothing in } x]{\text{average}} \boxed{\text{blurred}} \xrightarrow[\text{filtering in } x]{\text{derivative}} \boxed{\text{edges in } x}$$

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

$$\boxed{\text{image}} \xrightarrow[\text{filtering in } y]{\text{Derivative}} \boxed{\text{blurred}} \xrightarrow[\text{smoothing in } y]{\text{average}} \boxed{\text{edges in } y}$$

$$\begin{pmatrix}1\\0\\-1\end{pmatrix} \longrightarrow \begin{pmatrix}1 & 2 & 1\end{pmatrix} \longrightarrow \begin{pmatrix}1 & 2 & 1\\0 & 0 & 0\\-1 & -2 & -1\end{pmatrix}$$

Sobel

Image I

$$I * \begin{pmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{pmatrix} \to \frac{d}{dx}I$$

$$I * \begin{pmatrix}1 & 2 & 1\\0 & 0 & 0\\-1 & -2 & -1\end{pmatrix} \to \frac{d}{dy}I$$

$$\to \sqrt{\left(\frac{d}{dx}I\right)^2 + \left(\frac{d}{dy}I\right)^2}$$

Threshold

Edges

Marr Hildreth Edge Detector

① Smooth filter by Gaussian filter s

② Apply Laplcian to s

③ Used zero-crossings

↳ Scan along each row, record an edge pt. at the location of zero-crossing.

→ Repeat each step along the column

- Gaussian Smoothing

$$S = g * I$$

S: Smoothed image

- Find laplacian

$$\Delta^2 S = \frac{\partial^2}{\partial x^2} S + \frac{\partial^2}{\partial y^2} S$$

$$\Delta^2 S = \Delta^2 (g * I) = (\Delta^2 g) * I$$

$$g = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\log g = -\sigma\sqrt{2\pi} - \frac{x^2+y^2}{2\sigma^2}$$

$$\frac{1}{g}\cdot\frac{\partial g}{\partial x} = \frac{-2x}{2\sigma^2} = -\frac{x}{\sigma^2}$$

$$\frac{\partial g}{\partial x} = -\frac{gx}{\sigma^2} \Rightarrow \frac{\partial^2 g}{\partial x^2} = \frac{x^2}{\sigma^4} - \frac{g}{\sigma^2}$$

$$\Delta^2 g = -\frac{1}{\sqrt{2\pi}\,\sigma^3}\left(2 - \frac{x^2+y^2}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

(LoG)

Finding Zero Crossing

There're four different cases

1. $\{+,-\}$
2. $\{+,0,-\}$
3. $\{-,+\}$
4. $\{-,0,+\}$

Slope of zero-crossing $\{a,-b\} \rightarrow |a+b|$

- To mark an edge
  - compute slope of zero-crossing
  - Apply a threshold to slope.

The gaussian can be separated into 2 dimensional gaussians

$h(x,y) = I(x,y) * g(x,y)$ $\quad n^2$ multiplication

$h(x,y) = (I(x,y) * g_1(x)) * g_2(y)$ $\quad 2n$ multiplication

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

LoG

2D LoG can be separated into 4 1D convolutions

$$\Delta^2 S = \Delta^2(g * I) = (\Delta^2 g) * I = I * \Delta^2 g$$

(Requires n^2 mul)

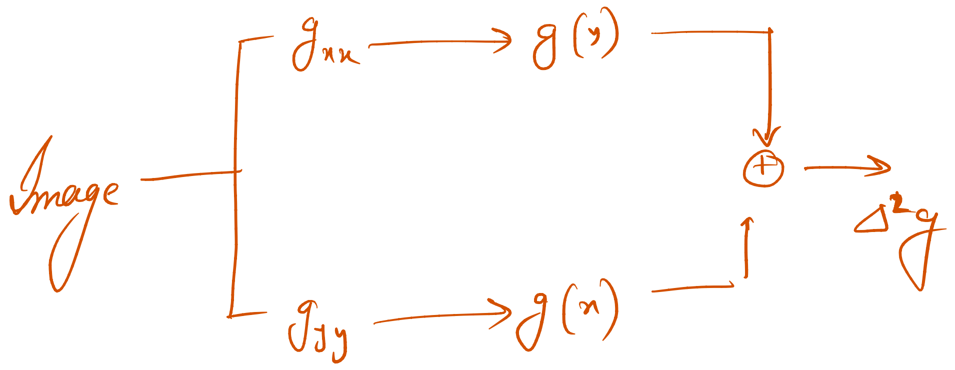
$$\Delta^2 S = (I * g_{xx}(x)) * g(y) + (I * g_{yy}(y)) * g(x)$$

Separability

Gaussian Filtering

Image $\rightarrow g(x) \rightarrow g(y) \rightarrow I*g$

Laplace of Gaussian filtering



LoG Algorithm

- Apply LoG to the Image: Either
  - use 2D filter $\Delta^2 g(x,y)$
  - or use 4 1D filters $g(x)$, $g_{xx}(x)$, $g(y)$, $g_{yy}(y)$
- Find zero-crossings from each row
- Find slope of zero-crossings
- Apply threshold to slope & mark edges.

Quality of an edge

- Robust to noise
- Localization
- Too many or too less application.

What is an interest point?

* Expressive texture
  - The pt. at which the direction of the boundary of object changes abruptly.
  - Intersection pt. between two or more edge segments.

Properties of Interest Point Detectors.

* Detect all (or most) true interest pts.
* No false interest pts.
* Well localized
* Robust w.r.t noise
* Efficient detection.

Where can I use it?

→ object tracking
→ 3D object reconstruction
→ Recognition
→ Stereo calibration.

Image Matching

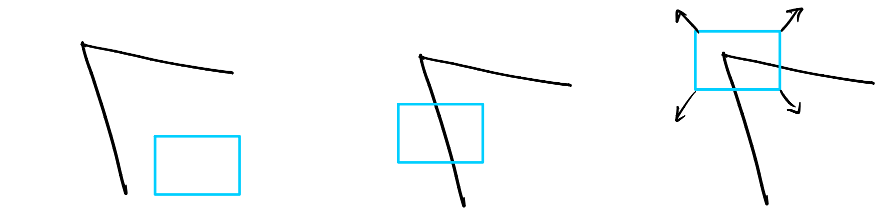
① Detection : Identify the interest pts.

② Description : Extract feature vector descriptor surrounding each interest pt.

③ Matching : Determine correspondence between descriptors in two views.

Harris Corner Detection

- corner pt. can be recognized in a window.
- Shifting a window in any direction should give a large change in intensity.

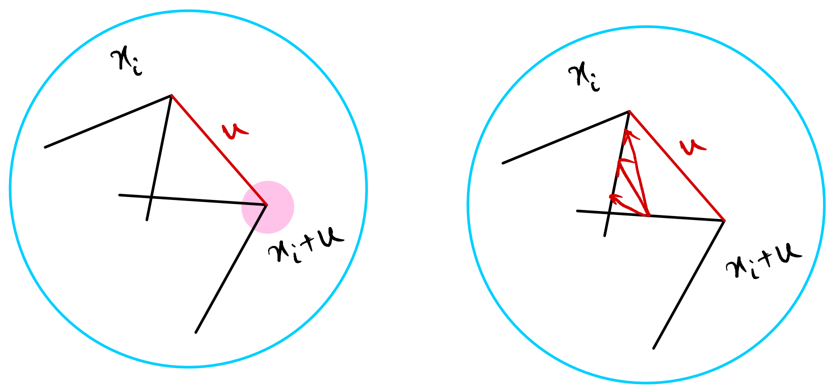


"flat" region: no change in all directions

"edge": no change along the edge direction

"corner" significant

Aperture Problem



minimize
$$SSD = \sum_k \sum_l \left(f(k,l) - h(k,l)\right)^2$$
Sum of squares Difference

$$= \sum_k \sum_l f(k,l)^2 - 2 \sum_k \sum_l f(k,l)\, h(k,l) + \sum_k \sum_l h(k,l)^2$$

$$\text{minimize } SSD = \sum_k \sum_l \left(-2 f(k,l)\, h(k,l)\right)$$

$$\text{maximize } SSD = \sum_k \sum_l 2 f(k,l)\, h(k,l)$$

$$\text{maximize Correlation} = \sum_k \sum_l h(k,l)\, f(k,l)$$

Mathematics of Harris Detector

- change of intensity for the shift (u,v)

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

Taylor Series

$$f(x) = f(a) + (x-a) f_x(a) + \frac{(x-a)^2}{2!} f_{xx}(a) + \frac{(x-a)^3}{3!} f_{xxx}(a) + \cdots$$

$$I(x+tu, y+tv) = \varphi(t)$$

$$\varphi'(t) = \frac{d}{dt}\left(I(x+tu, y+tv)\right) = f_1(x+tu, y+tv)\, u + f_2(x+tu, y+tv)\, v$$

$$\varphi(1) = \varphi(0) + \varphi'(0) + \cdots$$

$$I(x+u, y+v) = I(x,y) + f_x(x+u, y+v)\, u + f_y(x+u, y+v)\, v$$

Intensity

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

$$= \sum_{x,y} \left(I_x u + I_y v\right)^2$$

$$= \sum_{x,y} \left[(u \ v) \begin{pmatrix} I_x \\ I_y \end{pmatrix}\right]^2$$

$$= \sum_{x,y} (u, v) \begin{pmatrix} I_x \\ I_y \end{pmatrix} (I_x \ I_y) \begin{pmatrix} u \\ v \end{pmatrix}$$

$$= \sum_{x,y} (u \ v) \begin{pmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$= U M U^T$$

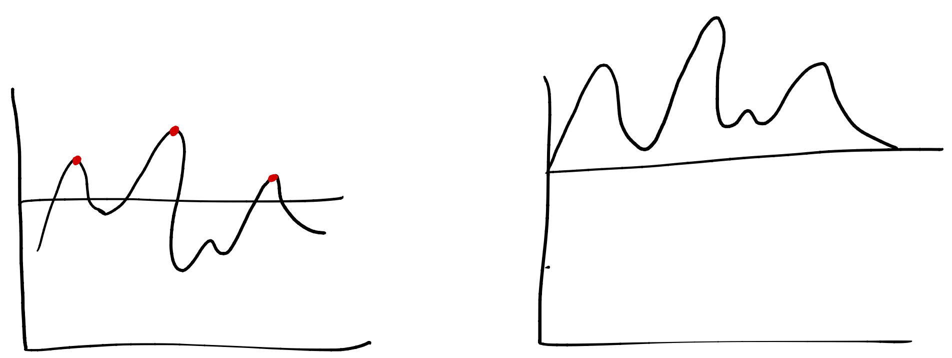
So, $M = S D S^{-1} = S \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} S^{-1}$

$$R = (\lambda_1 \lambda_2) - k(\lambda_1 + \lambda_2)^2$$

Affine Intensity change

$$I \rightarrow aI + b$$

Th.

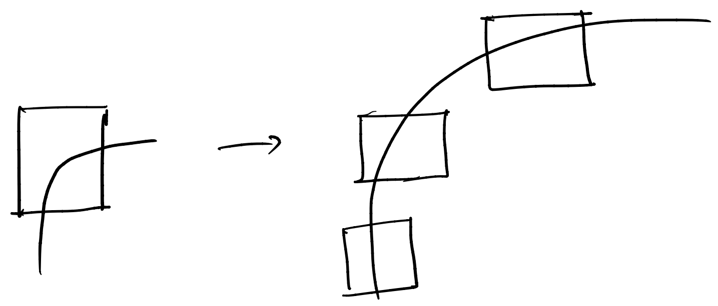


② Image Translation

$$I \rightarrow I + b$$

③ Image rotation → Ellipse remains the same

④ Scaling



Pyramids

- Gaussian and Laplacian pyramids
  - Reduce
  - Expand
- Application of Laplacian pyramids
  - Image Compression
  - Image Compositing

Gaussian Pyramids (Reduce)

$$g_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n)\, g_{l-1}(2i+m, 2i+n)$$

SIFT - Key Point Extraction

1. Extract distinctive invariant features
   → Correctly matched against a large database of features from many images
2. Invariance to image scale & rotation.
3. Robustness to
   → Affine (rotation, scale, shear) distortion.
   → change in 3D
   → Addition of noise

Advantages

→ Locality

→ Distinctiveness

→ Quantity

→ Efficiency

Steps

1. Scale Space Peak Selection
   → Potential locations for finding features
2. Key Point localization
   → Accurately locating the feature key pts.
3. Orientation assignment
   → Assigning orientation to the key pts.
4. Key Point Descriptor
   → Describing the key pt. as a high dimensional vector.

Scales

1. What should be the value of sigma for Canny & LoG edge detection.

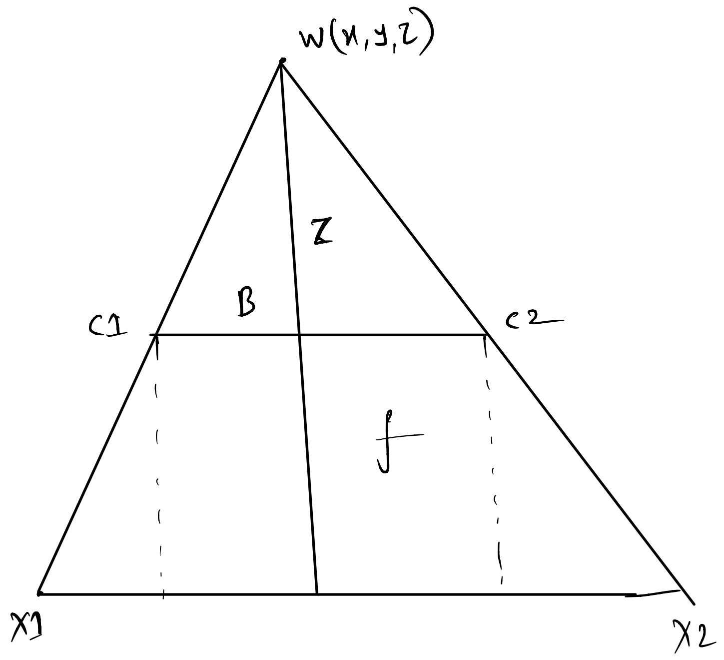
2. If we use multiple sigma values (scales) how do you combine multiple edge maps?

3. Marr-Hildreth:
   * Spatial Coincidence assumption:

Scale Space

* Apply whole spectrum of scales
* Plot zero-crossings vs scales in a scale-space
* Interpret scale space contours
  → contours are arches, open at the bottom, closed at the top.

Internal Tree



$$\frac{X1+\cancel{X}2+B}{B} = \frac{Z+f}{Z}$$

$$\Rightarrow \frac{X1+X2}{B}+1 = \frac{Z+f}{Z}$$

$$\Rightarrow \frac{X1+X2}{B} = \frac{f}{Z}$$

B = 10

Z = 10 cm

$$\Rightarrow B = \frac{Z(X1+X2)}{f}$$

$$\Rightarrow Z = \frac{fB}{(X1+X2)}$$

$$Z = \frac{fB}{d}$$

f = focal length

$$d = \frac{fB}{Z} =$$

$$\frac{bd}{2Z^2} =$$

$$\tan(\theta_1-\theta_2) = \frac{\tan\theta_1 - \tan\theta_2}{1+t\theta_1 \cdot t\theta_2}$$

$$= \frac{\frac{1}{2} - \frac{\cancel{10}5}{\cancel{2}(10+d)}}{1+\frac{1}{2}\cdot\frac{10}{2(10+d)}}$$

$$= \frac{\frac{(10+d)-5}{2(10+d)}}{\frac{4(10+d)+10}{2\cdot 2(10+d)}}$$

$$= \frac{5+d}{\cancel{2(10+d)}} \cdot \frac{2\cdot \cancel{2(10+d)}}{40+4d+10}$$

$$= \frac{5+d}{20+2d+5} = \frac{5+d}{2d+25}$$

$$X \rightarrow D = d(i,j)$$

$$d = D$$

$$\begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \\ a_{n1} & a_{n2} \end{matrix} \longrightarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{13} & A_{13} \\ \vdots & \\ A_{n1} & A_{n2} \end{pmatrix}$$

$$A_{12} = \begin{pmatrix} a_{12} - a_{21} \end{pmatrix} \qquad A_{11} = \begin{pmatrix} a_{11} - a_{11} \\ a_{11} - a_{21} \\ a_{11} - a_{31} \\ \vdots \\ a_{n1} - a_{n1} \end{pmatrix}$$

$$X\_train = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ \vdots & \\ a_{n1} & a_{n2} \end{pmatrix} \longrightarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ \vdots & \\ A_{n1} & A_{n2} \end{pmatrix}$$

$$A_{11} = \begin{pmatrix} a_{11} - a_{11} \\ a_{11} - a_{21} \\ \vdots \\ a_{11} - a_{n1} \end{pmatrix},$$