14/09/2021

Lecture 7
DIP

ARAVIND
2019102014

Obtaining gaussian filter coefficients

$$\sum_{k=0}^{N} {}^{N}C_k = 2^N \rightarrow \text{Uses pascal's triangle}$$

Use row n of pascal's triangle as a 1-D, n-point approximation of Gaussian filter

→ $N = S - 1$ → N^{th} row

→ Require kernel size to extend to most of Gaussian area

→ Heuristic:

For $\sigma = 1$,

use $5 \times (\sigma)$ is used to cover 98.76% of area

So, we relate continous gaussian to pascal's discretisation

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{\frac{-x^2}{2\sigma^2}} \Rightarrow \int_{-\infty}^{\infty} g_\sigma(x)\,dx = 1$$

To get 2-D gaussian

Take row, transpose & find outer product.

$$\frac{1}{64}\begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{i8} \end{bmatrix} \times \frac{1}{64}\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{i8} \end{bmatrix} \quad \text{for a } S = 7 \times 7$$

N^{th} row

Formula:

$$\frac{1}{\sqrt{2\pi}\,\sigma} = \frac{{}^{N}C_{N/2}}{2^N}$$

More sigma → More blurring

Kernal sizes are odd numbers to evenly distribute both sides

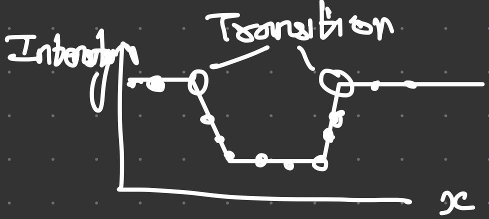
Edge Detection:

→ Sudden discontinuities

→ Important to understand image

→ Need to check pixels next to each other

→ Coefficients sum = 0



→ Second derivative gives zero crossing so its more important

→ Horizontal lines : y-derivative

Vertical lines : x-derivative

→ Prewitt Edge Filter:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \qquad G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Magnitude of gradient

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

Orientation of gradient

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

→ Edge will be $\perp^{lar}$ to gradient

for $\nabla f = \left[\frac{\partial f}{\partial x}, 0\right] \Rightarrow \theta = 0$ ← Horizontal

$\nabla f = \left[0, \frac{\partial f}{\partial x}\right] \Rightarrow \theta = 90^\circ$

Vertical

Laplacian filter:

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \qquad \nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) - 4f(x,y) + f(x,y-1)$$

-ve laplacian based on center

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

when 0, no influence ⇒ look at 4 neighbors (non-diagonal)

Sums up to 0

Sobel filter:

Sobel X $\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$ Sobel Y $\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

↳ Vertical filter ↳ Horizontal filter

Diff from prewitt edge as more priority to center (1^{far})

Image sharpening:

Using laplacian it can be done

↳ Obtain $\nabla^2 I(u,v)$ from $I(u,v)$

↳ for visualization add 128 ⇒ $\nabla^2 I(u,v) + 128$

If we added 255 ⇒

→ Better method:

$$I'(u,v) = I(u,v) + \nabla^2 I(u,v)$$

Unsharp masking

Steps:

1) Blur image
2) Subtract original from blurred image (Output mask)
3) Add mask back to original image

$$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$$

$$g(x,y) = f(x,y) + k \cdot g_{mask}(x,y)$$

Sharpened image

Weight k

when $k \geq 0$ ($k=1$) → unsharped masking

when $k > 1$, → high boost filtering

Unsharp Masking/Highboost Filtering as spatial filters

−1	−1	−1
−1	W	−1
−1	−1	−1

$W = 9A - 1$

If $A = 1$, unsharp masking ($I'(u,v) = I(u,v) + \nabla^2 I(u,v)$)

$A > 1$, Original image is added back (Highboost filtering)

Corner cases consideration:

↳ Padding with 0

↳ Replicate from boundary pixel

Everything discussed till now its linear (including gaussian)

Non-linear spatial filters

1. MAX : Ability to remove pepper noise
2. MIN : Ability to remove salt noise
3. MEDIAN : Ability to remove salt & pepper noise

Also known as rank/order statistics filters

Other examples of spatial filters :

1. Geometric mean
2. Harmonic mean
3. Contra harmonic mean
4. Mid point filter
5. Alpha trimmed mean filter

Bilateral filtering performs edge preserving smoothing