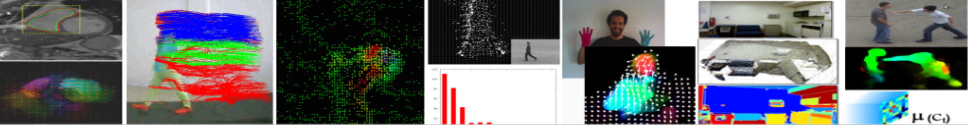


Spatial Processing: A convolution introduction

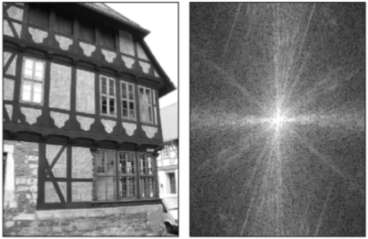
Isail Salazar, Fabio Martínez

Biomedical **I**maging, **V**ision and **L**earning **L**aboratory. **BivL^{2}ab**
Motion **A**nalysis and **C**omputer **V**ision. **MACV**
Universidad Industrial de Santander

September 21, 2018

Spatial and frequency domain

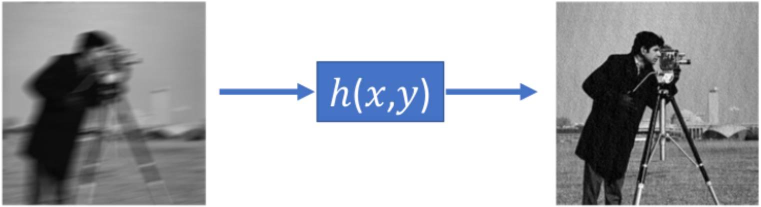
- **Spatial domain techniques:** Direct manipulation of image pixels (intensity values).
- **Frequency domain techniques:** Manipulation of Fourier transform or Wavelet transform of an image



Spatial domain Frecuency domain

This class is about spatial domain!

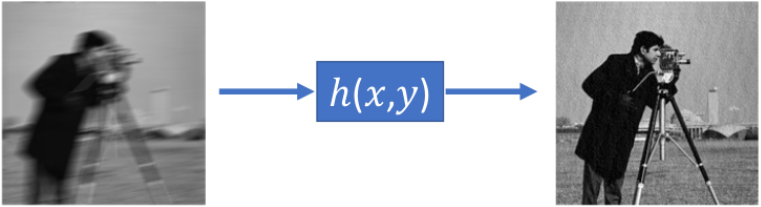
LSI Systems: Linear Shift-Invariant



- Any LSI system can be characterized in the spatial domain by a single function $h(x, y)$, i.e., the system's impulse response.

$$f(x, y) \to h(x, y) \to g(x, y) \qquad g(x, y) = f(x, y) * h(x, y)$$

LSI Systems: Linear Shift-Invariant



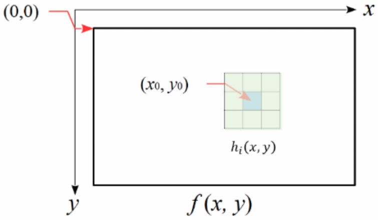
- Any LSI system can be characterized in the spatial domain by a single function $h(x, y)$, i.e., the system's impulse response.
- It can be shown that the input/output relationship of a LSI system is given by the convolution operation:

$$f(x, y) \to h(x, y) \to g(x, y) \qquad g(x, y) = f(x, y) * h(x, y)$$

2D Convolution

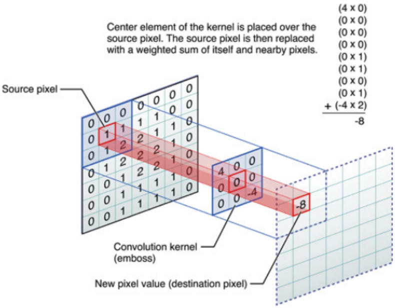
$$\boxed{g(x, y) = f(x, y) * h(x, y)}$$

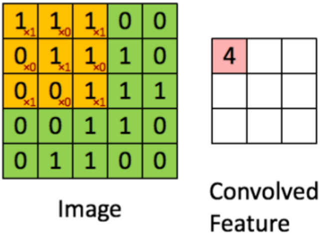
$$g(x, y) = \sum_{u=0}^{m} \sum_{v=0}^{n} f(x, y) h(u - x, v - y),$$ where $h(x, y)$ is a $n \times m$ kernell



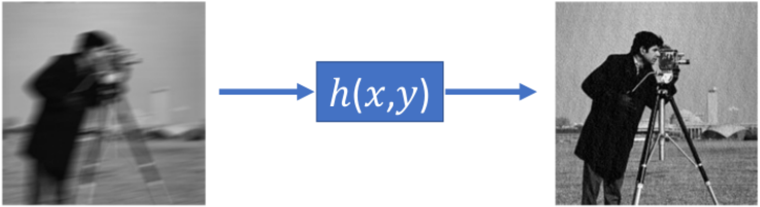
$$h_i(x, y) = h(-x, -y)$$

2D Convolution





2D Convolution: spatial filtering



- The impulse response can also be used to design systems with a given effect on the images.
- Examples: image denoising, image enhancement, feature extraction

What is a filter?

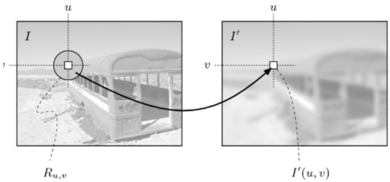
- **Filters:** combine the pixel of actual position with their neighbors
  - **Blurring:** compute the average intensity of block pixels

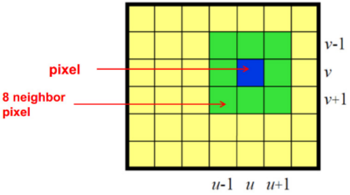
MEAN FILTER

Mean Filter

- Replace each pixel by average of pixel + neighbors
- For 3x3 neighborhood:

$$I'(u,v) \leftarrow \frac{p_0+p_1+p_2+p_3+p_4+p_5+p_6+p_7+p_8}{9}$$





Mean Filter

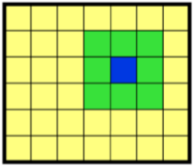
$$I'(u,v) \leftarrow \frac{p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8}{9}$$

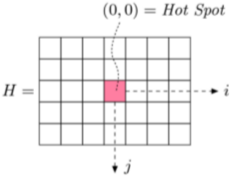
$$\begin{aligned} I'(u,v) \leftarrow \tfrac{1}{9} \cdot [\, & I(u-1,v-1) + I(u,v-1) + I(u+1,v-1) + \\ & I(u-1,v) + I(u,v) + I(u+1,v) + \\ & I(u-1,v+1) + I(u,v+1) + I(u+1,v+1)\,] \end{aligned}$$

$$H(i,j) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filter operation can be expressed as a matrix
Example: averaging filter

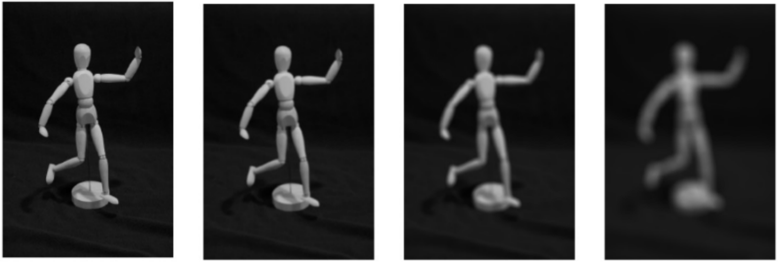
Filter matrix also called filter mask ***H(i,j)***





Mean Filter

Go to Notebooks and enjoy !!



Original $\quad 7 \times 7 \quad 15 \times 15 \quad 41 \times 41$

GAUSSIAN FILTER

Weighted Mean Filters

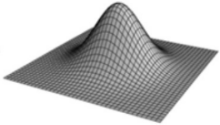
A special configuration is achieved by weighting each of the pixel in the windows according to the spatial location!

- Pixel closer at center are more important

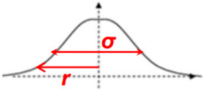
$1/16$	$2/16$	$1/16$
$2/16$	$4/16$	$2/16$
$1/16$	$2/16$	$1/16$

Gaussian Filter

$$G_\sigma(r) = e^{-\frac{r^2}{2\sigma^2}} \quad \text{or} \quad G_\sigma(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- where
  - σ is width (standard deviation)
  - r is distance from center

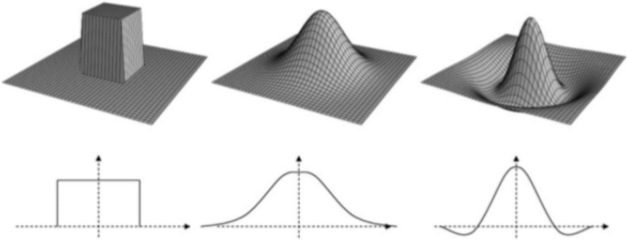


0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Gaussian filter

We can extend the same concept!..

- **Smoothing:** + coeffients (weighted average). E.g box, gaussian
- **Difference** filters: + and – weights. E.g. Laplacian



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

Box

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

Gaussian

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

Laplacian

DERIVATIVE FILTERS

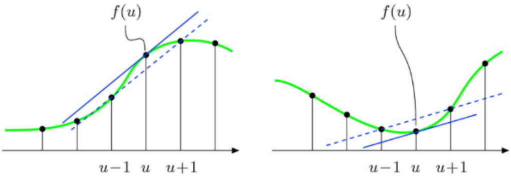
Derivative Filters

- Recall that we can compute derivative of discrete function as

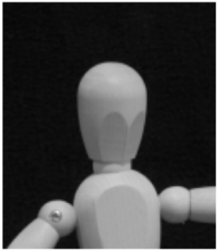
$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{2} = 0.5 \cdot \big(f(u+1) - f(u-1)\big)$$

- Can we make linear filter that computes central differences

$$H_x^D = \begin{bmatrix} -0.5 & \mathbf{0} & 0.5 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} -1 & \mathbf{0} & 1 \end{bmatrix}$$

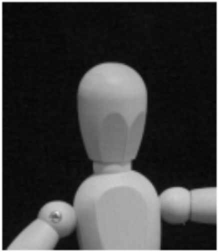


x-Derivative of Image using Central Difference



$* \begin{bmatrix} -0.5 & 0 & 0.5 \end{bmatrix} =$

y-Derivative of Image using Central Difference



$$* \begin{bmatrix} -0.5 \\ \mathbf{0} \\ 0.5 \end{bmatrix} =$$

Prewitt and Sobel Filters

- Prewitt Operator

$$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & \mathbf{0} & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & \mathbf{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

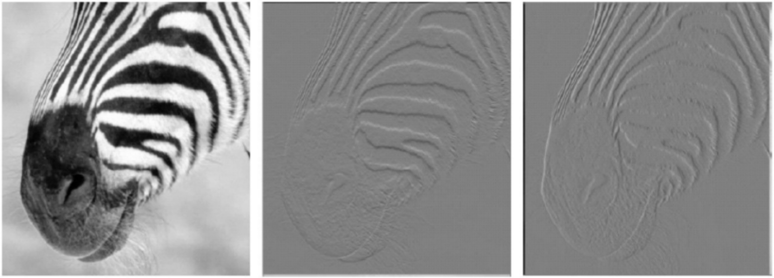
Written in separable form

$$H_x^P = \begin{bmatrix} 1 \\ \mathbf{1} \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & \mathbf{0} & 1 \end{bmatrix} \quad \text{and} \quad H_y^P = \begin{bmatrix} 1 & \mathbf{1} & 1 \end{bmatrix} * \begin{bmatrix} -1 \\ \mathbf{0} \\ 1 \end{bmatrix}$$

- Sobel Operator

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & \mathbf{0} & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & \mathbf{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Prewitt and Sobel Filters

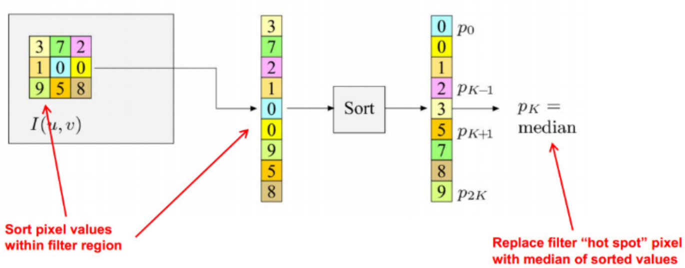


Median Filters

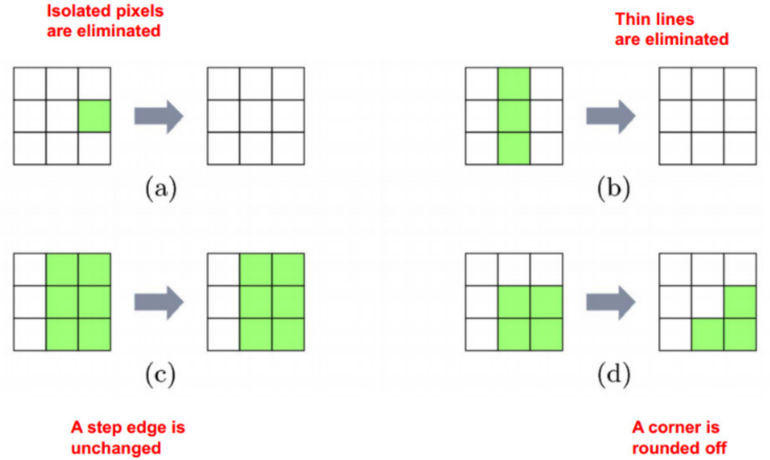
Median Filter

- Much better at removing noise and keeping the structures

$$I'(u,v) \leftarrow \text{median}\left\{I(u+i, v+j) \mid (i,j) \in R\right\}$$



Median Filter

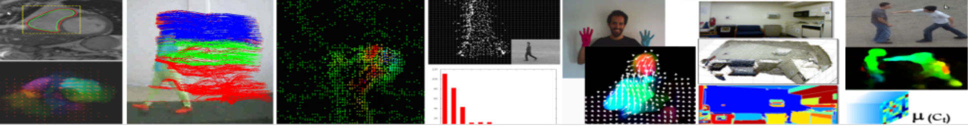


Removing noise with Median Filter

(a) Original Image with Salt-and-pepper noise

(b) Linear filter removes some of the noise, but not completely. Smears noise

(c) Median filter salt-and-pepper noise and keeps image structures largely intact. But also creates small spots of flat intensity, that affect sharpness



Thank you for your attention . . .

. . . It's time to wake up ... famarcar@uis.edu.co