Image Processing Lecture-5

Pixel neighbor operations - 2

Hazırlayan: M. Kemal GÜLLÜ

Median Filter

- Filtering operation is carried out by ordering pixels within a window including pixel itself.
- It is a non-linear filtering operation.
- This filter is efficient especially for impulse and salt&pepper type noise effects.
- May degreade the edge regions.

25	28	34	x	x
45	41	56	x	x
38	46	29	x	x
x	x	x	x	x
x	x	x	x	x

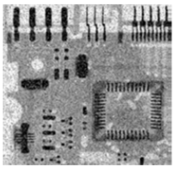
25, 28, 29, 34, 38, 41, 45, 46, 56

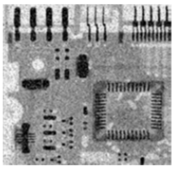
New pixel value

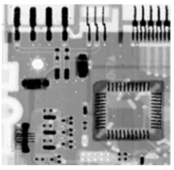
- **imnoise** function of MATLAB can be used to introduce noise.

Median Filter

- Removing salt & pepper noise using median filter


Noisy image


Mean filtering (3x3 kernel)


Median filtering (3x3 block)

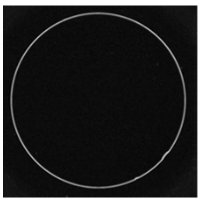
- **medfilt2** function in MATLAB can be used for 2-d median filtering.

Convolution (Sharpening)

- Edge: Sharp brightness changes in images
- The main purpose of sharpening is to make image clearer by improve details in image including the blurry regions.
- Sharpening operation is used to improve printing quality and medical imaging, industrial and robotic applications.

Convolution – Edge Detection

- Edge detection operation aims to find meaningful edge regions in the image.
- It can be used in
  - image segmentation when deciding object borders,
  - object recognition when extracting features,
  - and moving object detection.



Convolution (Sharpening)

- Sharpening can be performed using digital differentiation.
- The response of differentiation is directly related to amplitude of the discontinuity at the position where image operation is applied.
- 1st order derivative of a 1-d function:

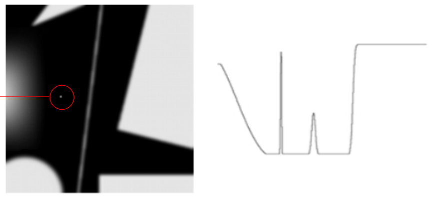
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- 2nd order derivative:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Convolution (Sharpening)

noise



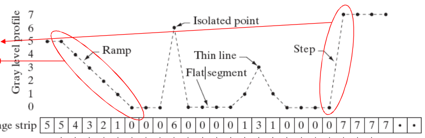


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First Derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0		
Second Derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0		

Convolution (Sharpening)

- 1st order derivative produces thick edges.
- 2nd order derivative give better response especially in image regions with details such as small edges and noisy pixels.
- Since the 2nd order derivative produce better response at the edges and image regions with details, it is preferred in sharpening mostly.

Utilization of 2nd order derivatives - Laplacian filter:

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

In the discrete form:

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

Convolution (Sharpening)

- A compact form of 2-D Laplacian filter :

$$\nabla^2 f = \left[f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \right]$$

$$\downarrow$$

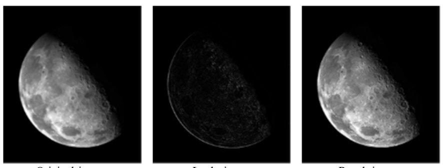
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Convolution (Sharpening)

- After 2-D Laplacian filtering, the sharper image can be obtained:

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{, if the center value in kernel is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{, if the center value in kernel is positive} \end{cases}$$



Original image | Laplacian filtered image | Result image

Convolution – Edge Detection

Utilization of 1st order derivative (The Gradient):

- Gradient as 2-D column vector:

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

Amplitude of this vector:

$$\nabla f = \mathrm{mag}(\nabla \mathrm{f}) = \left[G_x^2 + G_y^2 \right]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

- In order to eliminate computation cost of square and square-root operations absolute value operation is generally used in practice.

$$\nabla f \approx |G_x| + |G_y|$$

Convolution – Edge Detection

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix} \qquad \begin{aligned} G_x &= z_8 - z_5 \\ G_y &= z_6 - z_5 \end{aligned} \qquad \begin{aligned} \nabla f &\approx |G_x| + |G_y| \\ &\approx |z_8 - z_5| + |z_6 - z_5| \end{aligned}$$

- Robert cross gradient operator:

$$\begin{aligned} G_x &= z_9 - z_5 \\ G_y &= z_8 - z_6 \end{aligned} \qquad \begin{aligned} \nabla f &\approx |G_x| + |G_y| \\ &\approx |z_9 - z_5| + |z_8 - z_6| \end{aligned}$$

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

- In stead of even numbered kernels such as 2x2, let us use a 3x3 kernel:

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

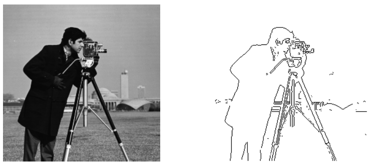
Sobel operators

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

Convolution – Edge Detection

- Prewitt operator:

$$h_{yatay} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad h_{düşey} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$



- Note that, edge image is constructed the thresholding of filtered image with the kernel described above.

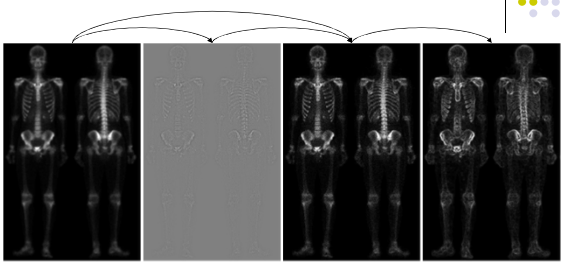
Convolution – Edge Detection

Kirsch operator:

- Used in template matching for pattern recognition.
- Very sensitive to edge directions.

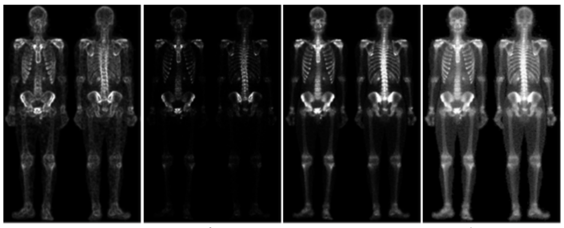
$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \qquad h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}$$

Application



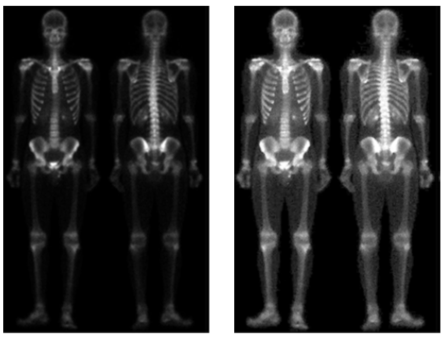
a- Original image | b- Laplacian | c- Original + Laplacian | d- Sobel

Uygulama-devam



e- 5x5 mean | f- (c x e) | g- (a+f) | h- Image enhancement

Uygulama-devam



Correlation

- Aims to find relation between two signals or image.
- It can be used to find an image block within a larger image/image set.

$$r(x,y) = \frac{\sum_{i=-m}^{m}\sum_{j=-n}^{n}\left(h(i,j)-\bar{h}\right)\left(f(x+i,y+j)-\bar{f}\right)}{\sqrt{\left(\sum_{i=-m}^{m}\sum_{j=-n}^{n}\left(h(i,j)-\bar{h}\right)^2\right)\left(\sum_{i=-m}^{m}\sum_{j=-n}^{n}\left(f(x+i,y+j)-\bar{f}\right)^2\right)}}$$

h : the block to be searched

f : search region (can be whole image)

$\bar{h}$: block mean

$\bar{f}$: image mean

(x,y) : pixel position

Correlation