Computação Visual e Multimédia

10504: Mestrado em Engenharia Informática

Chap. 4 — Image Filters

Image Filters

Outline

….:

- Spatial filter: definition
- Image neighborhoods versus masks
- Linear and non-linear filtering methods
- Spatial filter classification
- Smoothing filters (low-pass filters)
  - Mean filter, Gaussian filter, median Filter
- Sharpening filters (high-pass filters)
  - Unsharp mask filter, gradient operator, Roberts cross operator, Sobel operator, Prewitt operator, Laplacian operator

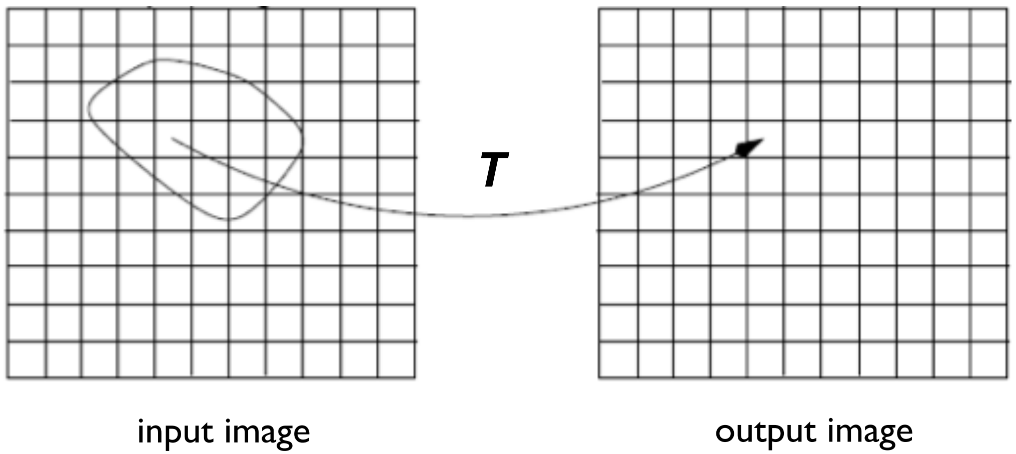
SPATIAL FILTERING

(or mask processing)

Spatial filtering

Spatial filtering is defined by:

- A neighborhood.
- An operation that is performed on the pixels inside the neighborhood



$g(x,y)=\boldsymbol{T}[f(x,y)]$

T operates on a neighborhood of pixels

Neighborhoods vs. masks

http://www.cambridgeincolour.com/tutorials/dynamic-range.htm

Filtering:

– Filter an image by replacing each pixel in the input image with a weighted sum of its neighbors

Neighborhood:

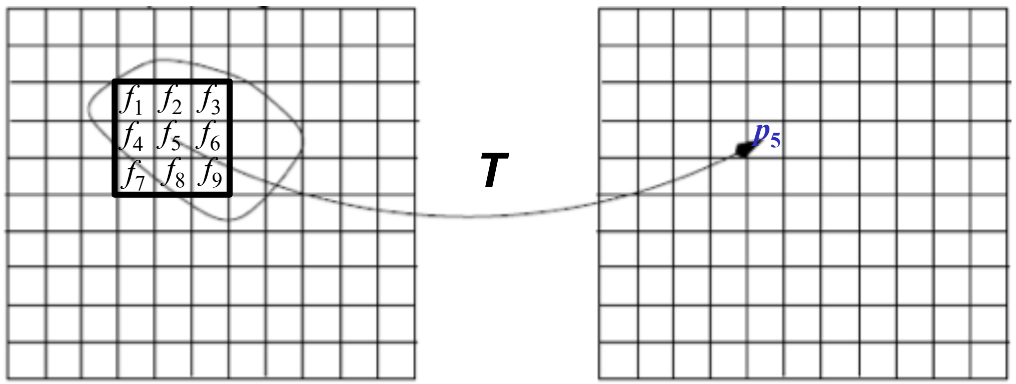
f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

– It is rectangular and its size is much smaller than that of f(x,y) - e.g., 3x3 or 5x5

Mask (or kernel):

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

– Apply a "mask" of weights



input image output image

$$p_5 = \sum_{i=1}^{9} w_i f_i$$

Neighbourhoods vs. masks (cont'd)

Filter weights or coefficients

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

In more general terms, and assuming that the origin of the mask is the center of the mask, we have:

- 3x3 mask:

$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w(i,j) f(x+i, y+j)$$

- KxK mask:

$$g(x,y) = \sum_{i=-K/2}^{K/2} \sum_{j=-K/2}^{K/2} w(i,j) f(x+i, y+j)$$

- A filtered image is generated as the center of the mask moves to every pixel in the input image.

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

Pixels of image section under filter

Linear / Non-linear spatial filtering methods

Linear filtering:

- A filtering method is linear when the output is a weighted sum of the input pixels.

Example:

$$p_5 = \sum_{i=1}^{9} w_i f_i$$

Non-linear filtering:

- Methods that do not satisfy the above property are called non-linear.

Example:

$$p_5 = \max(f_i, i = 1,2,\ldots,9)$$

Linear spatial filtering methods

Main methods:

- Correlation (studied above): $g(x,y) = w(x,y) \bullet f(x,y) = \sum_{i=-K/2}^{K/2} \sum_{j=-K/2}^{K/2} w(i,j) f(x+i, y+j)$
  - Application: Often used in applications where we need to measure the similarity between images or parts of images (e.g., pattern matching).
- Convolution: $g(x,y) = w(x,y) \otimes f(x,y) = \sum_{i=-K/2}^{K/2} \sum_{j=-K/2}^{K/2} w(i,j) f(x-i, y-j)$
  - Similar to correlation except that the mask is first **flipped** both horizontally and vertically
  - Note: if w(x,y) is symmetric, that is w(x,y)=w(-x,-y), then convolution is equivalent to correlation!

Spatial filter classification and setting

The word "filtering" has been borrowed from the frequency domain

Spatial filter classification:

- Low-pass (i.e., preserve low frequencies)
- High-pass (i.e., preserve high frequencies)
- Band-pass (i.e., preserve frequencies within a band)
- Band-reject (i.e., reject frequencies within a band)

How to choose a mask as spatial filter?:

- By sampling specific functions, namely:
  - *Examples*: Gaussian function, 1^{st} derivative of Gaussian, 2^{nd} derivative of Gaussian
- and, then, assigning function samples (the so-called weights) to pixels of the mask.

SMOOTHING FILTERS

(or low-pass filters)

Smoothing filters (low-pass filters)

Characteristics:

- Reduce noise and small details.
- The elements of the mask must be positive.
- Sum of mask elements is 1

Categories:

- Mean filter
- Gaussian filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Mean filter

5x5 mean mask:

$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm

Definition:

- The idea of mean filtering is simply to replace each pixel value in an image with the mean (or average) value of its neighbors, including itself.

Effect on the image:

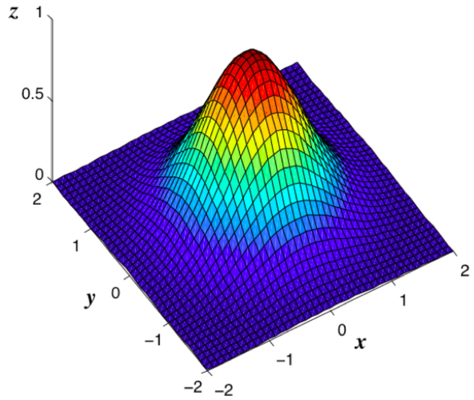
- Mean filtering is a simple, intuitive and easy to implement method of *smoothing* images, i.e. reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images.

input image   output filtered image

Gaussian filter

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

σ is the standard deviation of the distribution



$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

5x5 mean mask:

http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

Definition:

– The Gaussian smoothing operator is a 2-D convolution operator that is used to 'blur' images and remove detail and noise.

– The weights are samples of the Gaussian function.

– Mask (kernel) size:

height = width = 5 σ (subtends 98.76% of the area)

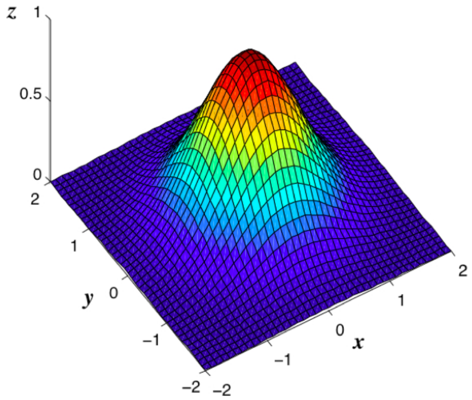
Effect on the image:

– It is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian ('bell-shaped') hump.

Gaussian filter (cont'd)

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

σ is the standard deviation of the distribution



$$\frac{1}{273}\begin{bmatrix}1 & 4 & 7 & 4 & 1\\ 4 & 16 & 26 & 16 & 4\\ 7 & 26 & 41 & 26 & 7\\ 4 & 16 & 26 & 16 & 4\\ 1 & 4 & 7 & 4 & 1\end{bmatrix}$$

5x5 mean mask:

http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

How does it work?:

- As σ increases, more samples must be obtained to represent the Gaussian function accurately.
- Consequently, the size of the mask increases.
- Therefore, σ controls the amount of smoothing.

input image

limited smoothing
small σ

strong smoothing
large σ

Median filter

Neighbor values:
115, 119, 120, 123, 124, 125, 126, 127, 150

Median value: 124

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Use of a median filter to improve an image severely corrupted by defective pixels

http://en.wikipedia.org/wiki/Median_filter

http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm

Definition:

- Replace each pixel by the median in a neighborhood around the pixel.
- It is a smooth, but not linear, filter.

How does it work?

- It replaces the value of a pixel with the *median of its neighbor pixel values,* including itself.
- The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.)

Effect on the image:

- Somewhat like the mean filter, it is normally used to reduce noise in an image.
- It is very effective for removing "salt and pepper" noise (i.e., random occurrences of black and white pixels).

SHARPENING FILTERS

(or high-pass filters)

Sharpening filters (high-pass filters)

Examples of sharpening masks/kernels:

$$\begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & +1 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix} \quad \begin{bmatrix} 0 & -1/4 & 0 \\ -1/4 & +2 & -1/4 \\ 0 & -1/4 & 0 \end{bmatrix}$$

Characteristics:

- Sharpening an image increases the contrast between bright and dark regions to bring out features. These filters emphasize fine details in the image – exactly the opposite of the low-pass filter.
- While low-pass filtering smoothes out noise, high-pass filtering does just the opposite: it *amplifies noise.*
- Useful for emphasizing transitions in image intensity (e.g., edges).
- Note that the response of high-pass filtering might be negative, so that values must be re-mapped to [0, 255].

Categories:

- Unsharp masking
- Gradient
- Sobel / Prewitt
- Laplacian

input image

output filtered image

http://www.cyanogen.com/help/maximdl/High-Pass_Filtering.htm

Unsharp mask filter

Definition:

- A sharp image is generated by subtracting a lowpass filtered (i.e., smoothed) image from the original image:

Highpass = **_Original_** – **_Lowpass_**
Image Image Image

- The "unsharp" of the name derives from the fact that the technique uses a blurred, or "unsharp," positive to create a "mask" of the original image
- The unsharped mask is then combined with the negative, creating the illusion that the resulting image is sharper than the original.

input image

output filtered image

http://en.wikipedia.org/wiki/Unsharp_masking

High Boost Unsharp mask filter

input image

output filtered image (k=1.4)

http://en.wikipedia.org/wiki/Unsharp_masking

Definition:

- Image sharpening emphasizes edges but details (i.e., low frequency components) might be lost.
- **High boost filter**: amplify input image, then subtract a lowpass image.

Highboost = k . ***Original*** – ***Lowpass***
= (k-1) . ***Original*** + ***Original*** – ***Lowpass***
= (k-1) . ***Original*** + ***Highpass***

- If **k=1**, we get a high pass filter
- If **k>1**, part of the original image is added back to the high pass filtered image

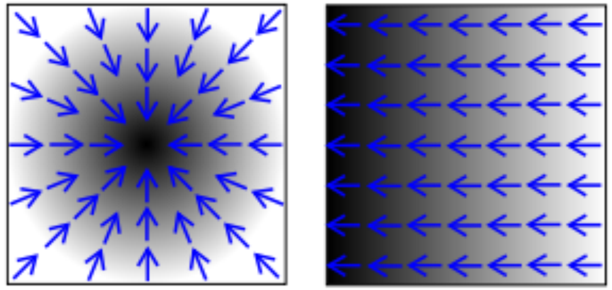
k>=1
w=9k-1

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & w & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm

Image gradient

At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction.



Two types of gradients, with blue arrows to indicate the direction of the gradient. Dark areas indicate higher values.

http://en.wikipedia.org/wiki/Image_gradient

Definition:

- Also called *color progression* or *gradation*, an image gradient is a directional change in the intensity or color in an image.

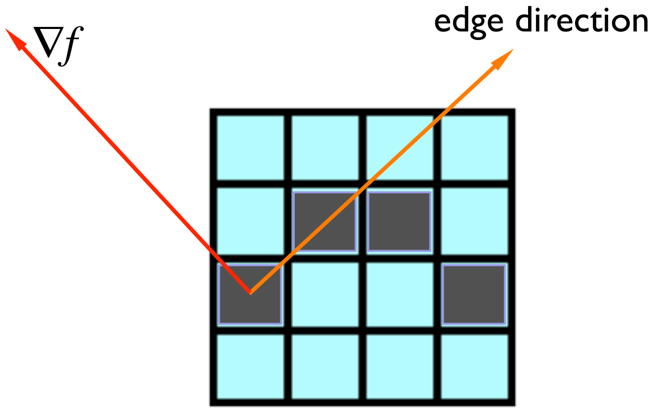
Maths:

- The gradient is the derivative of image: $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$
- $\frac{\partial f}{\partial x}$ is the gradient in the x-direction
- $\frac{\partial f}{\partial y}$ is the gradient in the y-direction.

What is it for?

- Image gradients may be used to extract information from images.
- Taking the derivative of an image results in sharpening the image.

Image gradient (cont'd)



http://en.wikipedia.org/wiki/Image_gradient

Definition:

- The gradient is a **vector** which has:
  - magnitude: $|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$ or $\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$

    approximation
  - direction: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$
- It conveys edge sharpening information as follows:
  - magnitude: provides information about edge strength
  - direction: perpendicular to the direction of the edge.

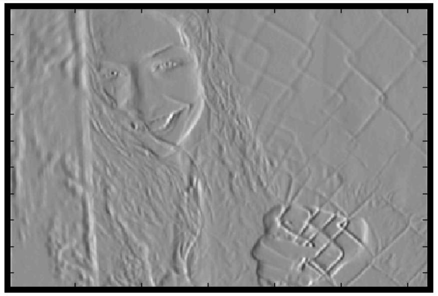
Image gradient computation

input image

X-direction gradient:

- Continuous domain: $$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h,y) - f(x,y)}{h}$$
- Image domain: $$G_X = f(x+1,y) - f(x,y), \qquad \Delta x = 1$$

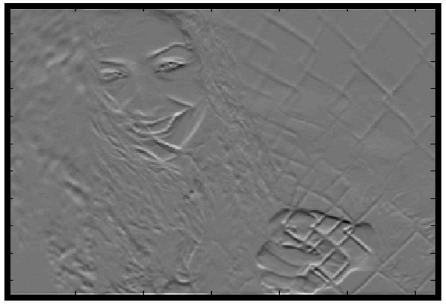
Sensitive to vertical edges



Y-direction gradient:

- Continuous domain: $$\frac{\partial f}{\partial y} = \lim_{h \to 0} \frac{f(x,y+h) - f(x,y)}{h}$$
- Image domain: $$G_Y = f(x,y+1) - f(x,y), \qquad \Delta y = 1$$

Sensitive to horizontal edges

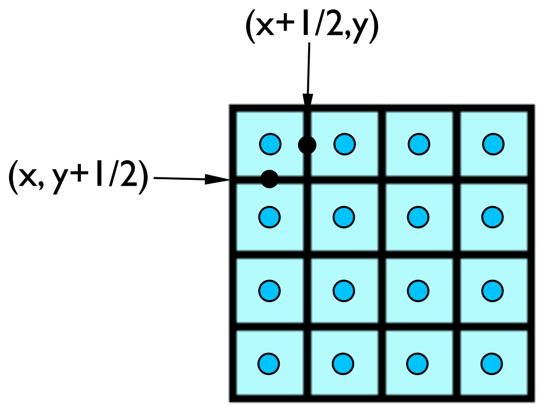


http://en.wikipedia.org/wiki/Image_gradient

Gradient operator

$$|G| = \sqrt{G_X^2 + G_Y^2}$$

The image will be shifted by half a pixel



X-direction gradient mask: $\begin{bmatrix} -1 & 1 \end{bmatrix}$

– Good approximation at (x+1/2, y): $G_X = f(x+1,y) - f(x,y)$

Y-direction gradient mask: $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

– Good approximation at (x, y+1/2): $G_Y = f(x,y+1) - f(x,y)$

Example:

- Approximate gradient at f_5

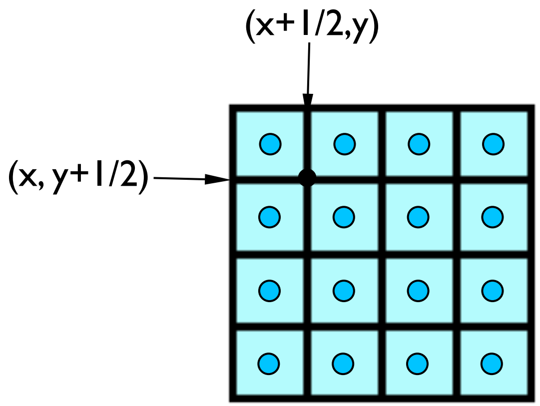
f_1	f_2	f_3
f_4	**f_5**	f_6
f_7	f_8	f_9

$$G_X = f_6 - f_5$$

$$G_Y = f_5 - f_8$$

Roberts cross operator

$$|G| = \sqrt{G_X^2 + G_Y^2}$$



X-direction gradient mask:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad G_X = f(x,y) - f(x+1,y+1)$$

Y-direction gradient mask:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad G_Y = f(x+1,y) - f(x,y+1)$$

Example**:**

- Approximate gradient at f_5

f_1	f_2	f_3
f_4	**f_5**	f_6
f_7	f_8	f_9

$$G_X = f_5 - f_9$$

$$G_Y = f_6 - f_8$$

Sobel operator

$$|G| = \sqrt{G_X^2 + G_Y^2}$$

The Prewitt operator is identical to Sobel operator, except that the constant k=2 takes on the value k=1 instead.

http://euclid.ii.metu.edu.tr/~ion528/demo/lectures/6/2/2/index.html

The Sobel operator is one of the most commonly used edge detectors

X-direction gradient mask:

$$\begin{bmatrix} -1 & 0 & 1 \\ -k & 0 & k \\ -1 & 0 & 1 \end{bmatrix} \quad (k=2)$$

$$G_X = \left[f(x+1,y-1) + k.f(x+1,y) + f(x+1,y+1) \right] - \left[f(x-1,y-1) + k.f(x-1,y) + f(x-1,y+1) \right]$$

Y-direction gradient mask:

$$\begin{bmatrix} 1 & k & 1 \\ 0 & 0 & 0 \\ -1 & -k & -1 \end{bmatrix} \quad (k=2)$$

$$G_Y = \left[f(x-1,y-1) + k.f(x,y-1) + f(x+1,y-1) \right] - \left[f(x-1,y+1) + k.f(x,y+1) + f(x+1,y+1) \right]$$

Example:

– Approximate gradient at f_5

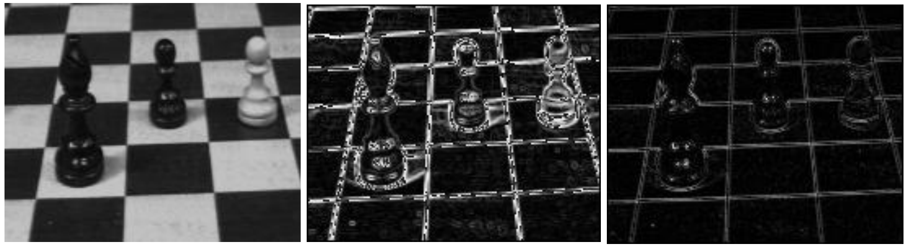
f_1	f_2	f_3
f_4	**f_5**	f_6
f_7	f_8	f_9

$$G_X = \left[f_3 + k.f_6 + f_9 \right] - \left[f_1 + k.f_4 + f_7 \right]$$

$$G_Y = \left[f_1 + k.f_2 + f_3 \right] - \left[f_7 + k.f_8 + f_9 \right]$$

http://en.wikipedia.org/wiki/Sobel_operator

Image Laplacian



Sobel Laplacian

• dot product

$$\nabla^2 f = \nabla f \bullet \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \bullet \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

Definition:

– The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image.

Approximate 2nd derivatives and Laplacian:

– The gradient in the x-direction:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad L_X = f(x,y+1) - 2f(x,y) + f(x,y-1)$$

– The gradient in the y-direction

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad L_Y = f(x+1,y) - 2f(x,y) + f(x-1,y)$$

– The Laplacian:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad L = L_X + L_Y$$

Effect on the image:

– It is a sharpening operator because it highlights image regions of rapid intensity change and is therefore often used for edge detection.

Summary:

…:

- Images are discrete digital signals subjected to additive, random and impulse noise
- Noise can be attenuated through a variety of filters (mean, Gaussian, median, etc.) by convolving the original image with a discrete kernel generated from one of these smoothing filters
- Convolution associativity can be used to speed up processing when multiple convolutions are required.
- Sharpening filters (gradient, Sobel, Laplacian, etc.) can be used to detect edges in images.
- Results likely are imperfect because we are operating in a discrete world.