Introduction to Image Processing

Lecture 2
21st Sept. 2015

Guillaume Lemaître
guillaume.lemaitre@udg.edu

Université de Bourgogne

1. Intensity Transformation
    - Foreword
    - Basic transformation
    - Histogram Equalization
    - Histogram specification - matching
    - From global to local

2. Spatial Filtering
    - Preface
    - Linear filtering
    - Smoothing linear filtering
    - Smoothing non-linear filtering
    - Sharpening filter

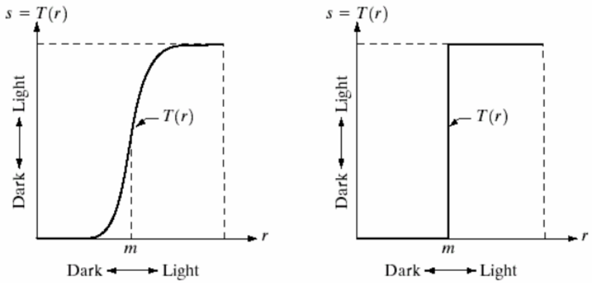
Intensity Transformation
Foreword

Definition

$$g(x, y) = T[f(x, y)] \quad (1)$$

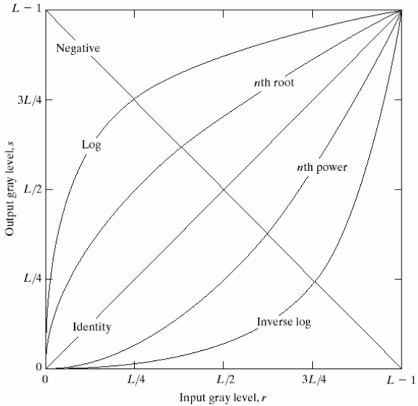
In this part, g depends *only* on the value of f at a single point (x, y)

$$s = T[r] \quad (2)$$



Intensity Transformation
Foreword

Gray-scale transform



Intensity Transformation
Basic transformations

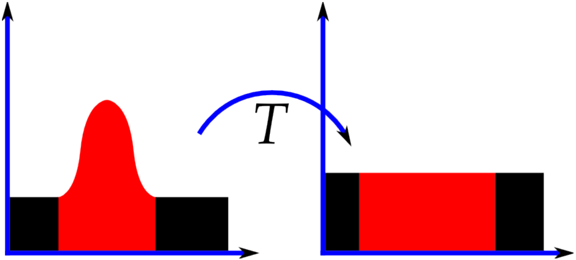
Gray-scale transform

- Image negatives
- Log transformations
- Power-law (Gamma) Transformations
- Contrast stretching - Intensity-level slicing
- Bit-plane slicing

→ Check the Ipython notebook

Intensity Transformation
Histogram equalization

Principle



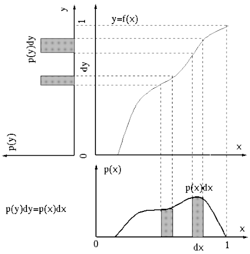
- Equally make use of all the available gray levels in the dynamic range

Intensity Transformation
Histogram equalization

Formulation

- Find the gray level mapping $y = f(x)$
- All pixels in dx will be mapped into dy

$$p(y)dy = p(x)dx \ . \tag{3}$$



Intensity Transformation
Histogram equalization

Formulation

- Equalizing is equivalent to force $p(y) = 1$ so that

$$\int_0^1 p(x)dx = \int_0^1 dy = 1 \ , \tag{4}$$

$$p(x)dx = dy \text{ or } p(x) = \frac{dy}{dx} \ , \tag{5}$$

- By integrating both sides

$$y = f(x) = \int_0^x p(u)du = P(x) - P(0) = P(x) \ , \tag{6}$$

Intensity Transformation
Histogram equalization

Intuitive aspects

- If $p(x)$ is low, $P(x)$ shallow slope, dy will be narrow, causing $p(y)$ to be high
- If $p(x)$ is high, $P(x)$ steep slope, dy will be wide, causing $p(y)$ to be low

Intensity Transformation
Histogram equalization

Exercise

r_k	n_k	$p_r(r_k) = \frac{n_k}{MN}$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Intensity Transformation
Histogram specification matching

Principle

- Similar to histogram equalization
- Involve an extra step to match a given histogram instead of a uniform histogram

Algorithm

- Perform histogram equalization to find $s = T(r)$
- Perform histogram equalization to find $s = G(z)$
- Find the inverse transform $z = G^{-1}(s)$
- Apply the successive mapping $r \xrightarrow{T(\cdot)} s \xrightarrow{G^{-1}(\cdot)} z$

Intensity Transformation
Histogram specification matching

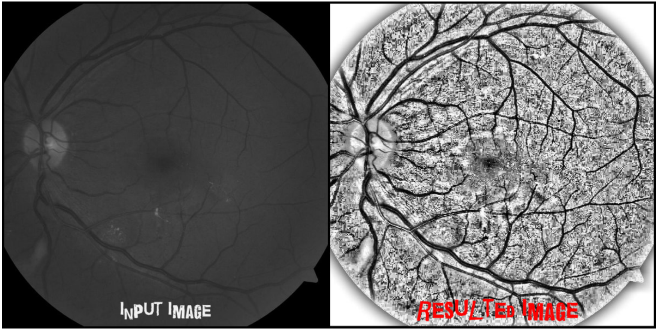
Exercise

z_k	$p_z(z_k) = \frac{n_k}{MN}$
$z_0 = 0$	0.00
$z_1 = 1$	0.00
$z_2 = 2$	0.00
$z_3 = 3$	0.15
$z_4 = 4$	0.20
$z_5 = 5$	0.30
$z_6 = 6$	0.20
$z_7 = 7$	0.15

Intensity Transformation
From global to local

Local enhancement

- Define a square neighbourhood
- Move pixel to pixel
- Apply any processing



Spatial Filtering
Preface

How-to

- Use of spatial masks for image processing
- Linear and Nonlinear

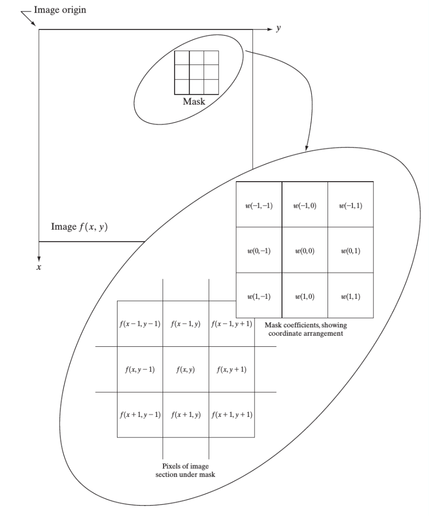
Different types

- Low-pass filters
- High-pass filters
- Band-pass filters

Spatial Filtering

- **Low-pass** filters eliminate or attenuate high frequency component (sharp image details) in the frequency domain, and result in image blurring.
- **High-pass** filters eliminate and attenuate the low frequency components and result in sharpening edges and other sharp details.
- **Band-pass** filters remove or select a frequency region between low and high frequencies.

Spatial Filtering
Linear filtering



Spatial Filtering
Linear filtering

Correlation

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \tag{7}$$

Convolution

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \tag{8}$$

Spatial Filtering
Linear filtering

Computation example

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

1	2	3
4	5	6
7	8	9

Spatial filter
Linear filtering

Vector representation

The basic approach is to sum products between the mask coefficients and the intensities of the pixels under the mask at a specific location in the image.
For 3×3 filter:

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$

Spatial Filtering
Smoothing linear filter

- Replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.
- Averaging filter
- Blurring the edges
- Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

$\frac{1}{9} \times$

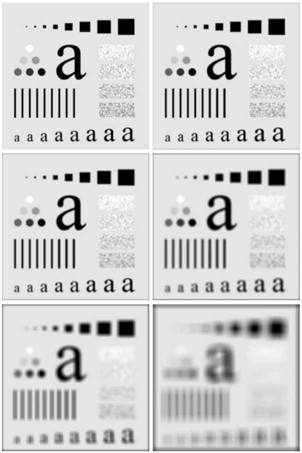
1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

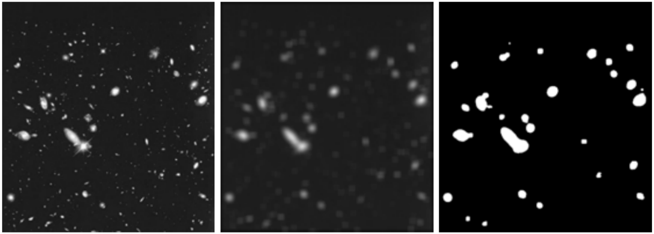
1	2	1
2	4	2
1	2	1

Spatial Filtering
Smoothing linear filter

Results of smoothing with square averaging filter masks of sizes $n = \{3, 5, 9, 15, 35\}$



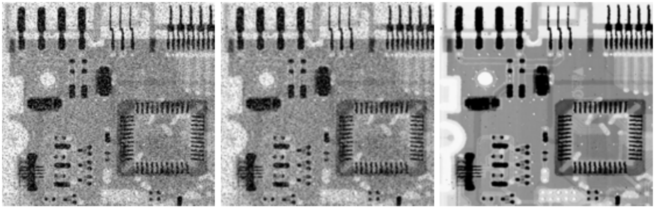
Spatial Filtering
Smoothing linear filter



Spatial Filtering
Smoothing non-linear filter

Median filtering also used for noise elimination.

- The gray level of each pixel is replaced by the median gray levels in the neighborhood of the pixel instead of the average.
- X-ray image of the circuit board corrupted by salt-and-pepper noise. Noise reduction with 3×3 average and median filter, respectively.



Spatial Filtering
Sharpening filter

Aim

$\rightarrow$ To highlight fine detail or to enhance blurred detail.

- smoothing $\approx$ integration
- sharpening $\approx$ differentiation

Categories of sharpening filters

- Derivative operator
- Basic high-pass spatial filter
- High-boost filtering

Spatial Filtering
Sharpening filter - Derivative filter

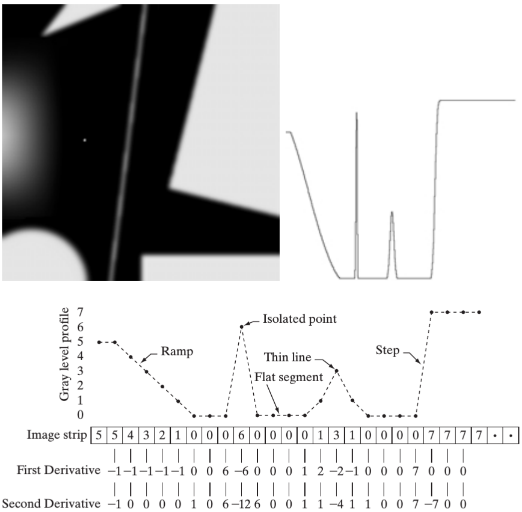
First-order derivative

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

Second-order derivative

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

Spatial Filtering
Sharpening filter - Derivative filter



Spatial Filtering
Sharpening filter - Derivative filter

First-order derivative

- 0 in constant gray segments
- Non-zero at the onset of steps or ramps
- Non-zero along ramps

Second-order derivative

- 0 in constant gray segments
- Non-zero at the onset and end of steps or ramps
- 0 along ramps of constant slope.

Spatial Filtering
Sharpening filter - Derivative filter

First-order derivative

- produce thicker edges in an image
- have a stronger response to a gray-level step

Second-order derivative

- have a stronger response to fine detail, such as thin lines and isolated points
- produce a double response at step changes in gray level
- have stronger response to a line than to a step and to a point than to a line

Spatial Filtering
Sharpening filter - 2D, Second order derivatives

- Isotropic filters, rotation invariant
- Laplacian (linear operator)

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

- Discrete version:

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

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

Spatial Filtering
Sharpening filter - 2D, Second order derivative

- Digital implementation

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

- Two definitions, one is negative of the other

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{Center of the mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{Center of the mask is positive} \end{cases}$$

Spatial Filtering
Sharpening filter - 2D, Second order derivative

- Filtering and recovering the original part:

$$g(x,y) = f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] + 4f(x,y)$$

$$g(x,y) = 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

Spatial Filtering

Sharpening filter - 2D, Second order derivative

0	1	0
1	−4	1
0	1	0

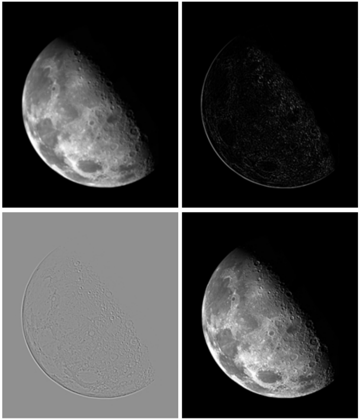
1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Spatial Filtering
Sharpening filter - 2D, Second order derivative

Image of the north pole of the moon, laplacian filtered image, laplacian image scaled for display and image enhanced by laplacian, respectively.



Spatial Filtering
Sharpening filter

High-boost filter

1. Blur the image (low-pass filtering)
2. Subtract the blurred image from the original (high frequency image)
3. Add the previous mask

- $k = 1$ –> Unsharp masking
- $k > 1$ –> Highboost filtering

Spatial Filtering
Gradient filter - first derivative

- The most common method of differentiation in image processing

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

- It is non-isotropic
- Its magnitude (often call the gradient) is rotation invariant

$$\nabla f = |G_x| + |G_y| = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial x})^2]^{1/2}$$

Spatial Filtering
Gradient filter - first derivative

Different masks use to calculate the gradient of a region of interest with z_5 as a central pixel, Robert cross gradient masks middle row and Sobel filters in the last row.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Spatial Filtering
Gradient filter - first derivative

- Computation:
- Cross differences as used in early development of digital image processing: $G_x = (z_9 - z_5), \quad G_y = (z_8 - z_6)$
- Robert cross gradient:

$$\nabla f \approx [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

- Sobel filter

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