COMP 478/6771 (FALL 2020)

Digital Image Processing

Digital Image Enhancement in Spatial Domain

Instructor: Prof. Yiming Xiao

Tutors:

Materials provided by Dr. T. D. Bui

Introduction

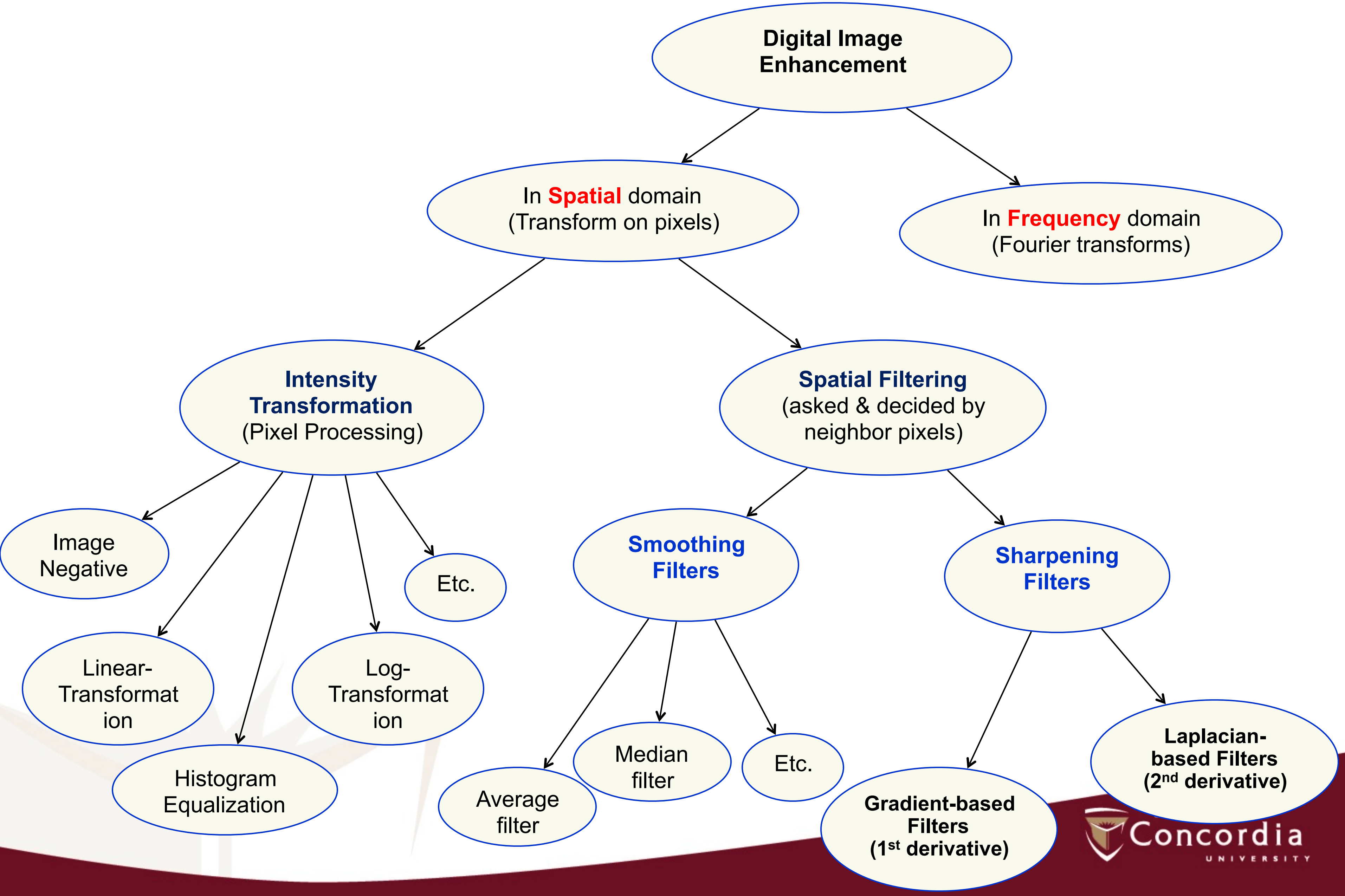
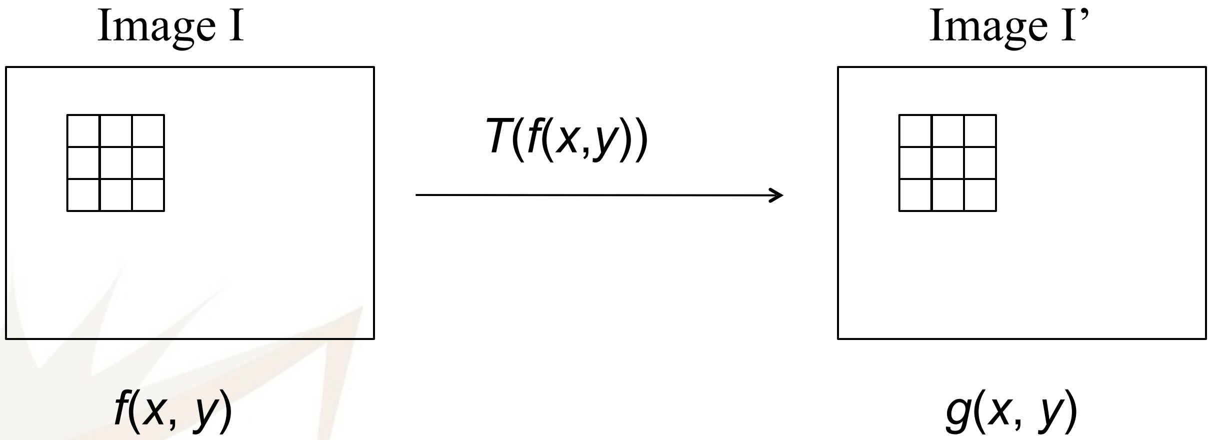

Digital Image Enhancement
In Spatial domain (Transform on pixels)
In Frequency domain (Fourier transforms)
Intensity Transformation (Pixel Processing)
Spatial Filtering (asked & decided by neighbor pixels)
Image Negative
Linear-Transformat ion
Histogram Equalization
Log-Transformat ion
Etc.
Smoothing Filters
Sharpening Filters
Average filter
Median filter
Etc.
Gradient-based Filters (1st derivative)
Laplacian-based Filters (2nd derivative)

Image Processing in Spatial Domain

Characteristics:

- Transforming on pixels (*intensity transformation*) or
- Usually asked and decided by neighbor pixels (*filtering*)

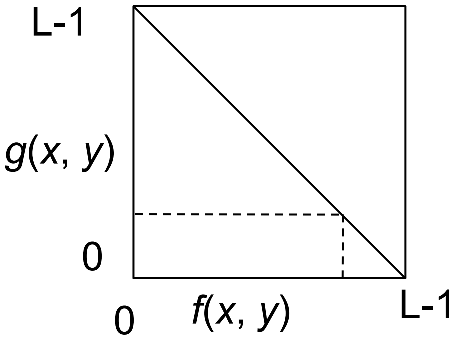
=> Spatial information is lost



Intensity Trans. – Image Negative

Purposes

- Focus on dark regions (small regions), especially when black areas are dominant in sizes.

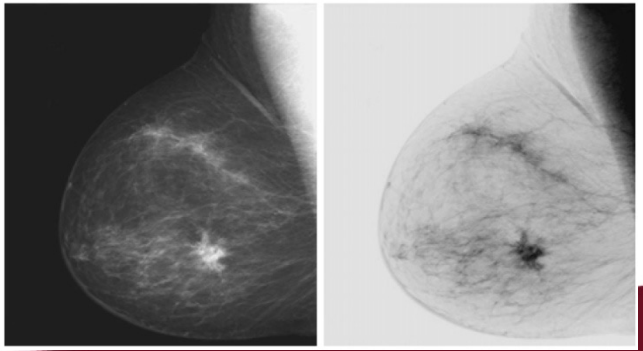


L: grayscale level,
e.g, n = 3, L = 2^3 *= 8*
n= 8, L = 2^8 *= 256*

$$g(x,y) = T(f(x,y)) = (L\text{-}1) - f(x,y)$$

$$\begin{matrix} [0, L\text{-}1] \\ [0, 255] \end{matrix} \xrightarrow[255 - f(x,y)]{L\text{-}1 - f(x,y)} g(x,y)$$

Example

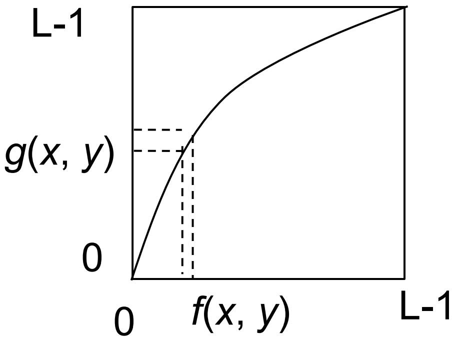


Digital Image Processing,
3rd Edition, by R. C. Gonzalez,
R. E. Woods, 2008, Prentice Hall.

Intensity Trans. – Log Transformation

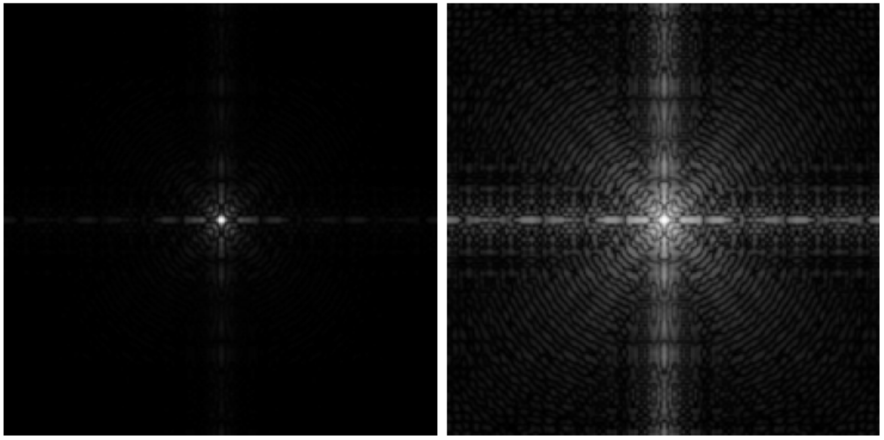
Purposes

- Enhance the values of dark pixels, while compressing the higher level values

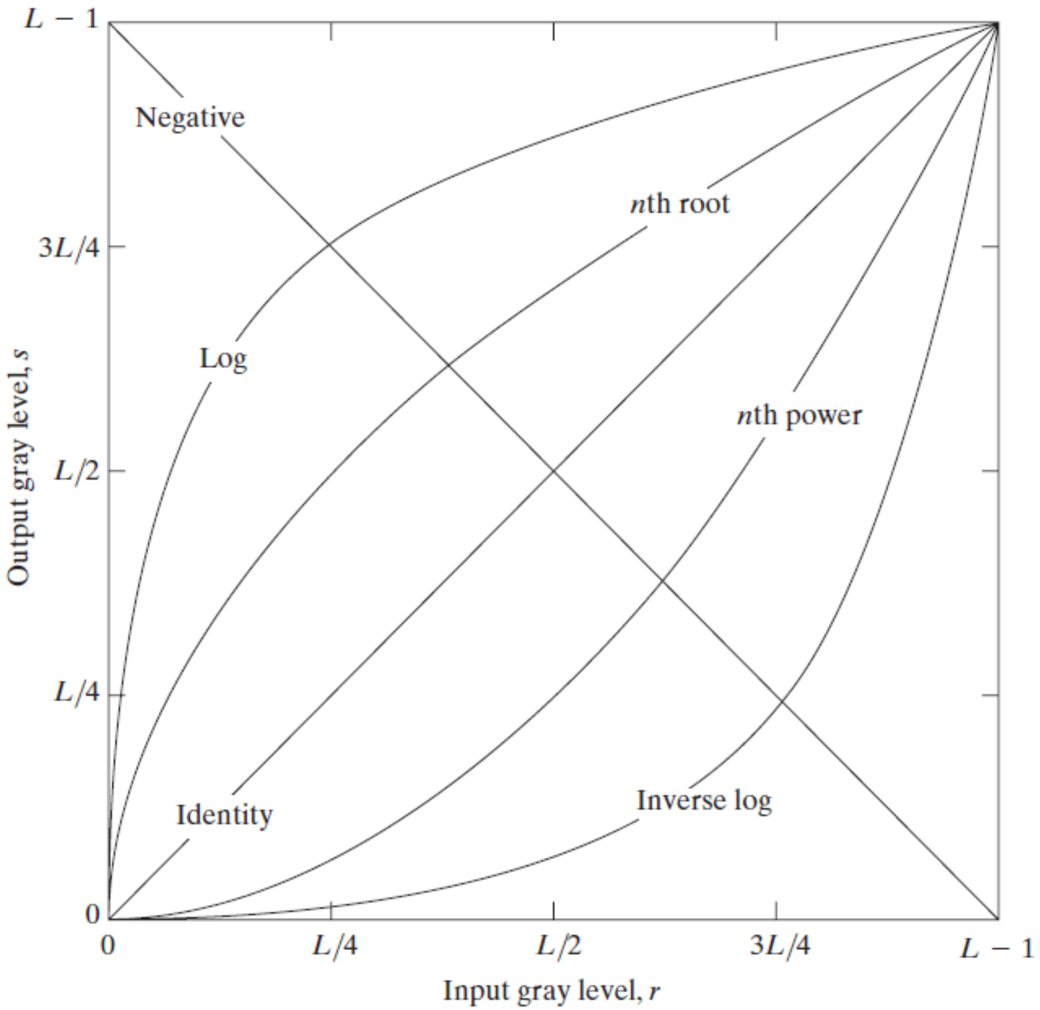


$$g(x,y) = T(f(x,y)) = c.\log(f(x,y)+1)$$

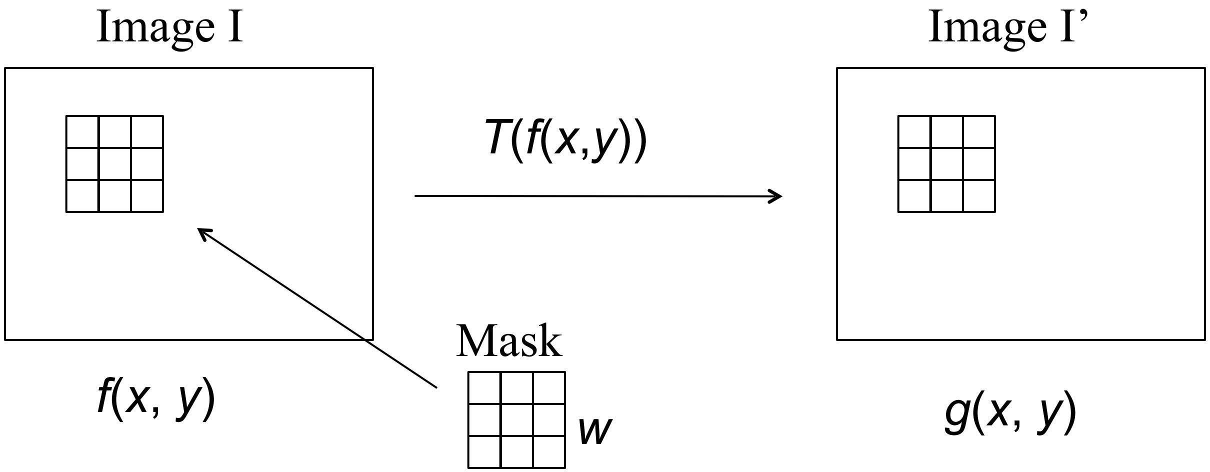
Example



Other transformations



Spatial filtering



$$g(x,y) = f(x,y) \otimes w(i,j)$$

$$\begin{aligned} g(x,y) = {} & f(x-1,y-1)*w(-1,-1) + f(x-1,y)*w(-1,0) + f(x-1,y+1)*w(-1,+1) + \\ & f(x,y-1) * w(0,-1) + f(x,y)*w(0,0) + f(x,y+1)*w(0,+1) + \\ & f(x+1,y-1)*w(1,-1) + f(x+1,y)*w(1,0) + f(x+1,y+1)*w(1,+1) \end{aligned}$$

Notes:

- $\otimes$ is the convolution operator
- Size of the mask: m × n => must be an odd value
- w(0,0) is the value of the center position of the mask
- When calculating the boundary should add the zero values outside

Smoothing filters – Average filter

In “average filter”, we have:

$$\begin{array}{|ccc|}\hline 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ \hline\end{array}$$

$w_{3\times 3}$

Meaning of the average filter:

Instead of

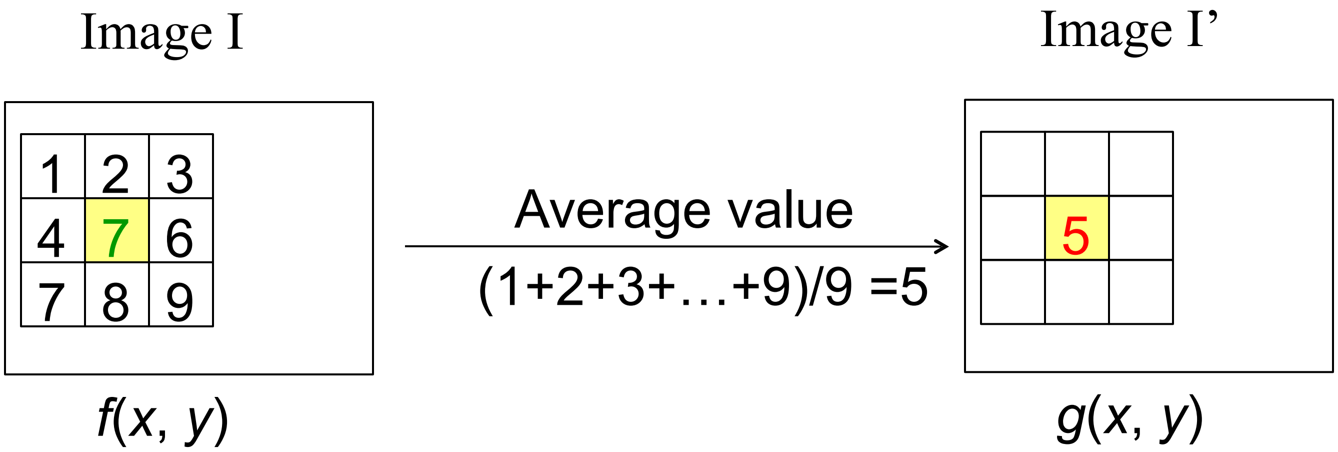


Image I

1	2	3
4	7	6
7	8	9

$f(x, y)$

$$\xrightarrow[(1+2+3+\ldots+9)/9\ =5]{\text{Average value}}$$

Image I’

	5	

$g(x, y)$

Replace by

$$\underset{w_{3\times 3}}{\begin{array}{|ccc|}\hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \hline\end{array}} \times \mathit{1/9} \;=\; \underset{w_{3\times 3}}{\begin{array}{|ccc|}\hline 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ \hline\end{array}}$$

, then we do convolution

Smoothing filters – Average filter (cont.)

Purposes (advantages)

- Reduce noise
- Smoothing

Disadvantages

- Increase the size of mask => reduce the noise but the image quality is also decreased
- Blurred images

MATLAB Code:

```
X  = imread(‘abc.bmp’);
Y = double(X)/256;

% or if color image, use:
% Y = rgb2gray(double(X/256));

W = ones(3, 3)/9;

G = conv2(Y, W);
```

Smoothing filters – Median filter

In median filter, we have:

- Select m×n neighbor pixels in *f(x,y)*
- Sorted
- Select the middle for *g(x,y)*

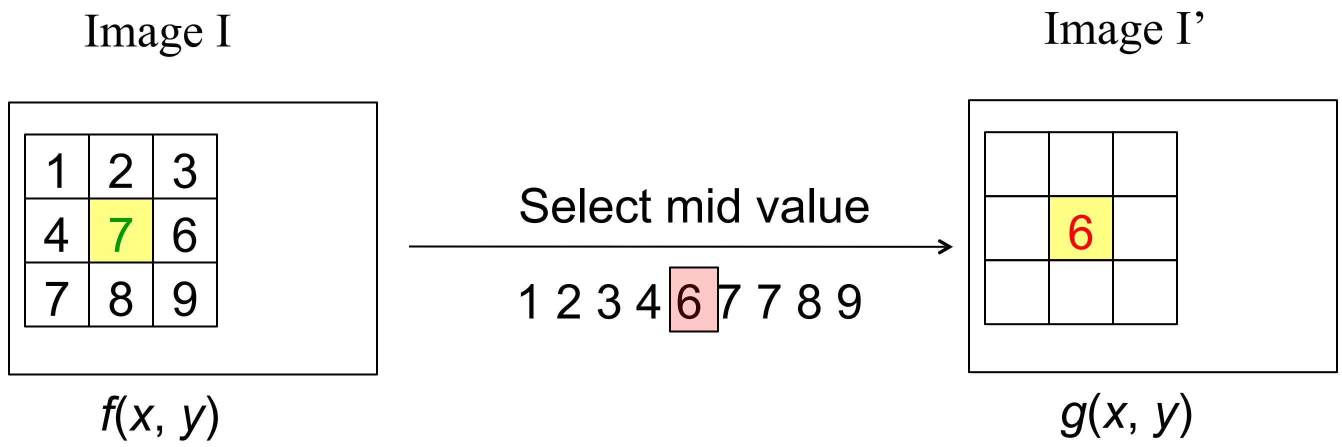


Image I

1	2	3
4	7	6
7	8	9

f(*x*, *y*)

Select mid value

1 2 3 4 6 7 7 8 9

Image I’

	6	

g(*x*, *y*)

Smoothing filters – Median filter (cont.)

Purposes (advantages)

- Reduce multiphicative noise (in X-ray images)

Disadvantages

- Blurred images

Exercise. Implement negative filter.

Exercise. Implement an N x N median filter for gray scale images.

- you should get the N (size of filter) in function input argument.
- you can use M = median(A) function for computing median.
- N is an odd number.
- reshape : change dimension of a matrix.