Digital Image Processing
CS390S

Feng Jiang

METROPOLITAN STATE UNIVERSITY OF DENVER

Welcome!

- CS390S
- Introduction (instructor, students, course)
- What is digital image processing
- Basics of an image

Introduction - instructor

- Dr. Feng Jiang
- Email: fjiang@msudenver.edu
- Office: AES 200U
- Phone: 303-615-1258

Introduction - instructor

- My research work

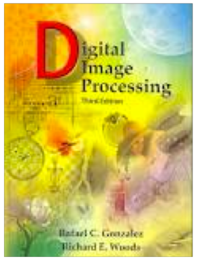
Introduction - students

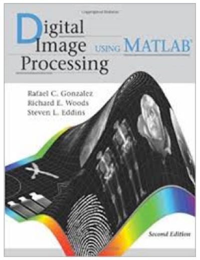
Introduction - course

- Course Syllabus available on blackboard
- Course Policies
- Waitlist procedure

Resources

- Books:
- Digital Image Processing 3rd Ed
- Digital Image Processing Using Matlab 2nd Ed





- Links:
- http://www.imageprocessingplace.com/index.htm
- https://web.stanford.edu/class/ee368/handouts.html
- Dataset
- http://www.imageprocessingplace.com/root_files_V3/image_databases.htm
- http://www.image-net.org/
- Labs
- http://vision.stanford.edu/

Software

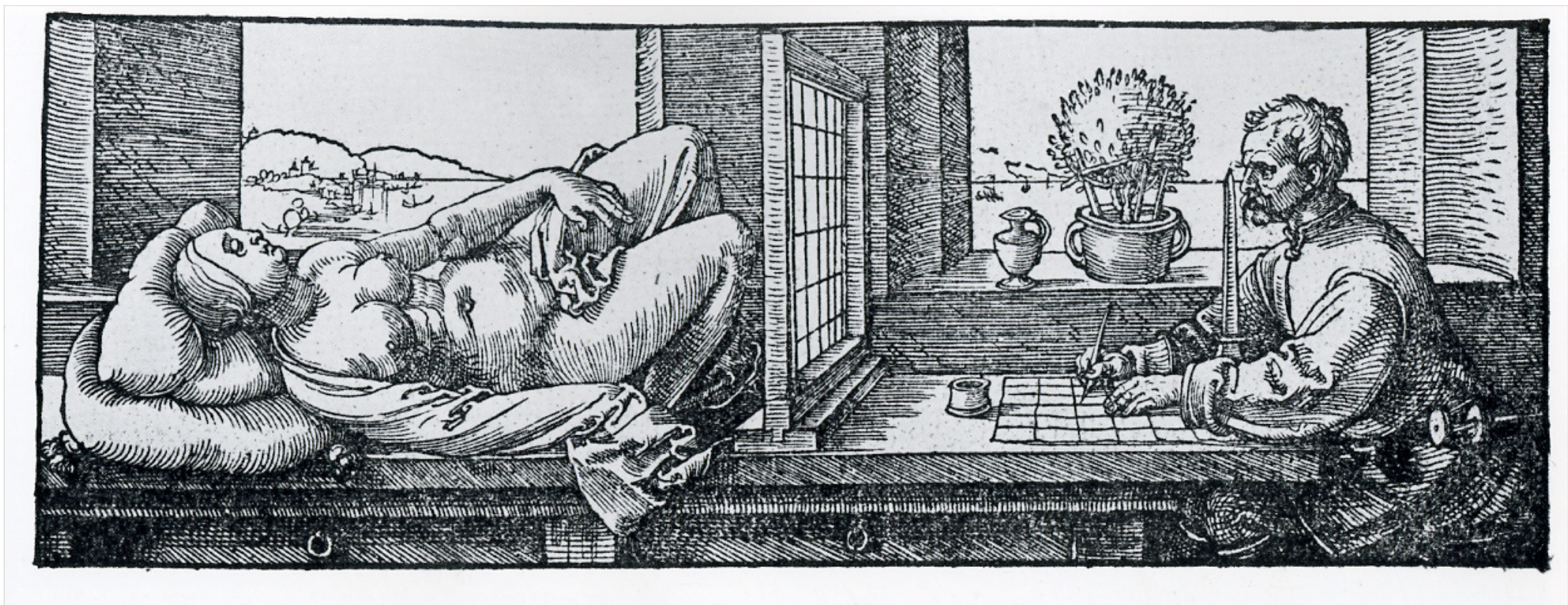
- Matlab

  https://www.mathworks.com/academia/student_version.html?s_tid=products_stu_version

- https://www.mathworks.com/pricing-licensing.html?intendeduse=student&prodcode=ML

  Please add “image processing toolbox”

What is an image?



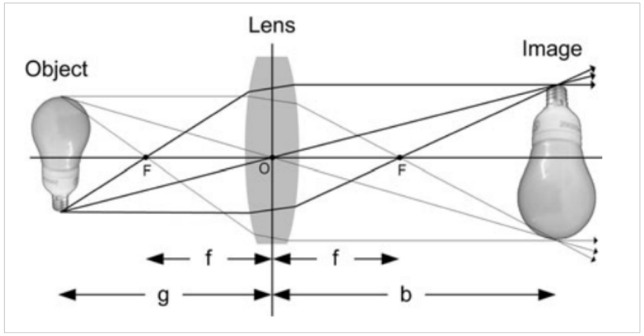
Digital Image

Image: a visual representation in form of a function $f(x,y)$ where f is related to the brightness (or color) at point (x,y)
Most images are defined over a rectangle

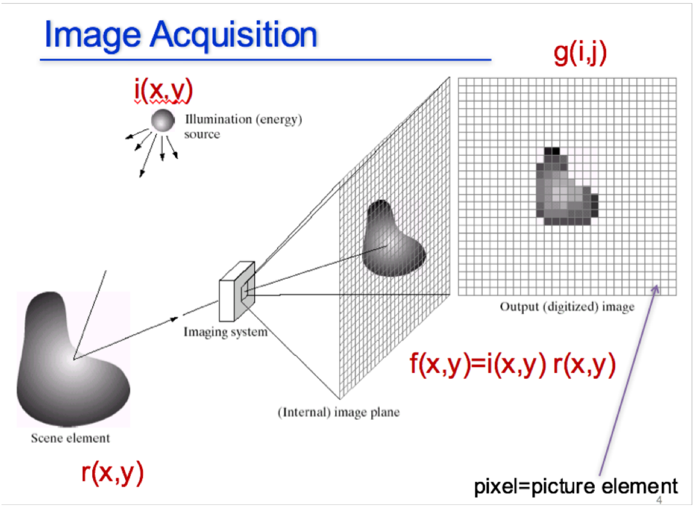
Digital image: discrete samples $f[x,y]$ representing continuous image $f(x,y)$
Each element of the 2-d array $f[x,y]$ is called a **pixel** or **pel**
(from "picture element")

200x200 100x100 50x50 25x25

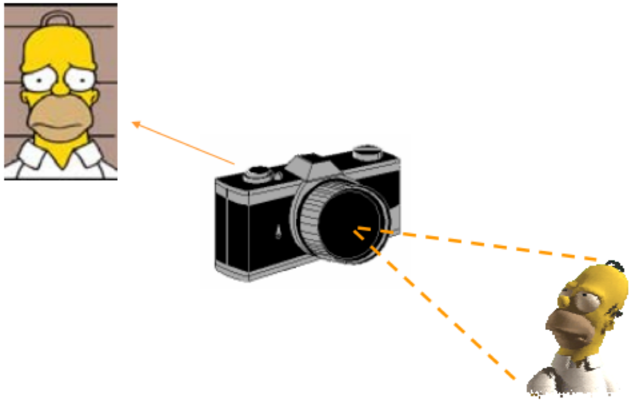
Imaging System



- Image acquisition
- Real scenery/object
- Sensor



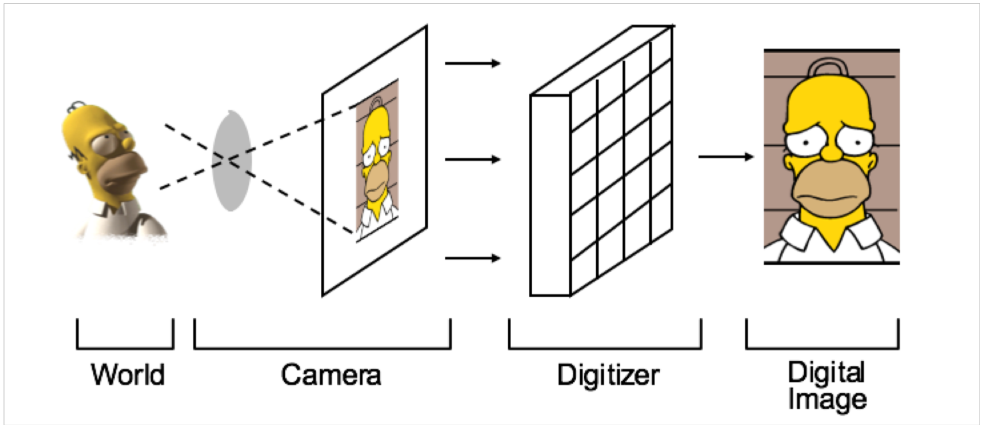
- An image is a projection of a 3D scene into a 2D *projection plane*.

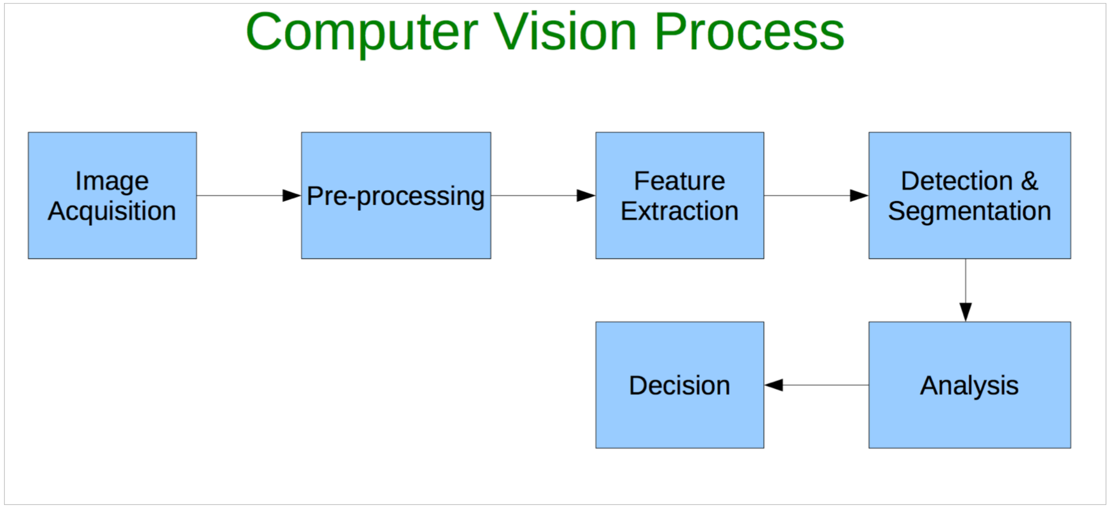


- An image can be defined as a 2 variable function I(x,y) , where for each position (x,y) in the projection plane, I(x,y) defines the light intensity at this point.

Imaging System

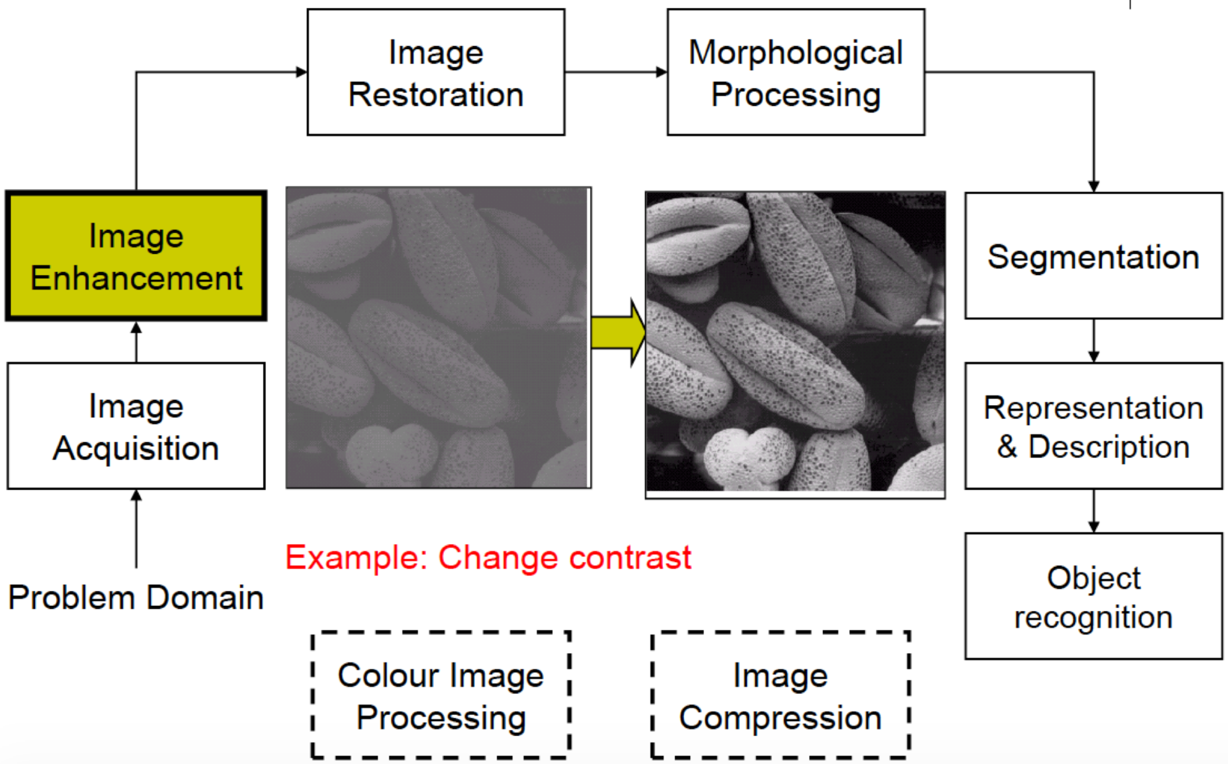
- Image acquisition





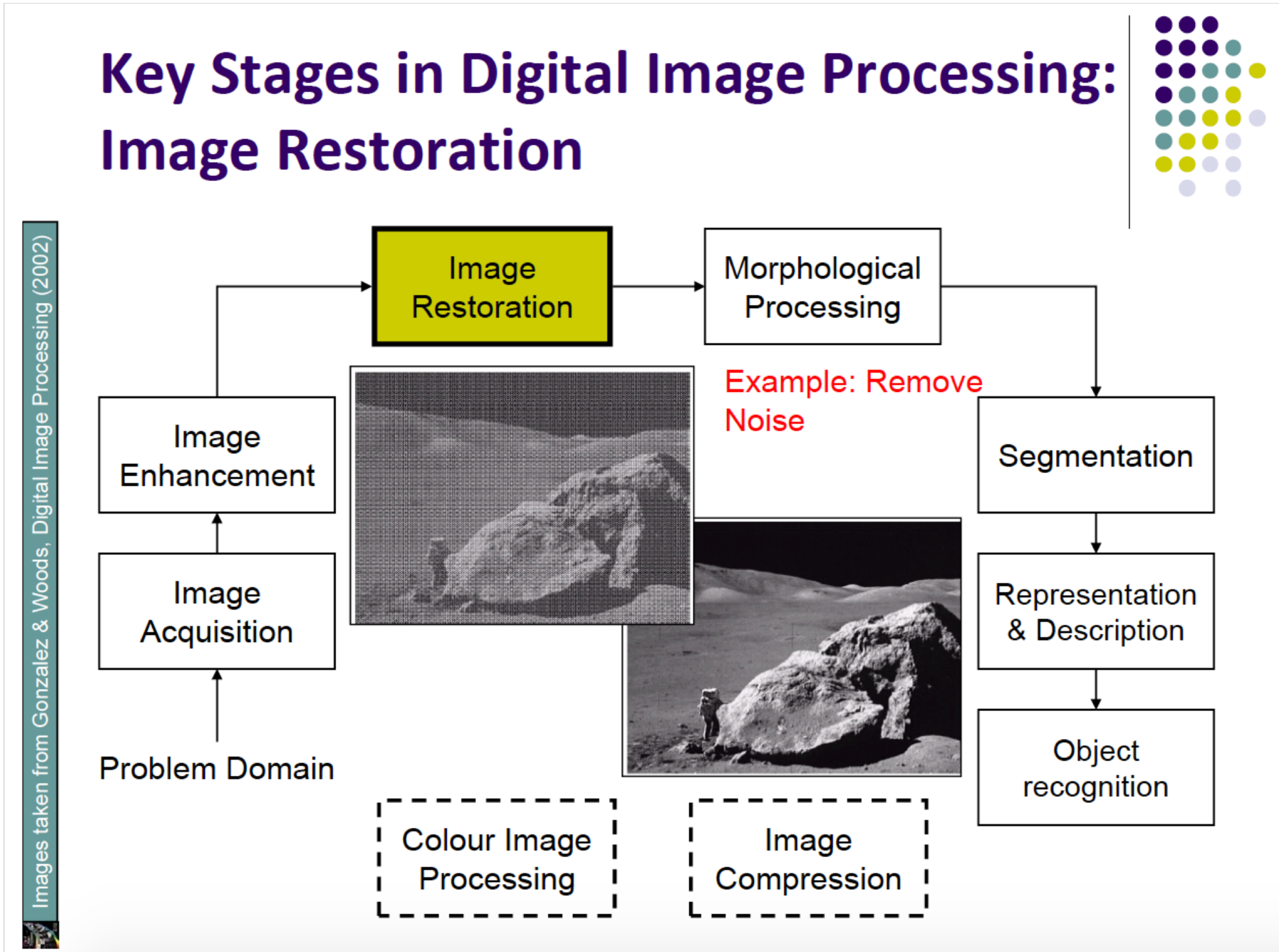
Digital Image Processing

Key Stages in Digital Image Processing: Image Enhancement

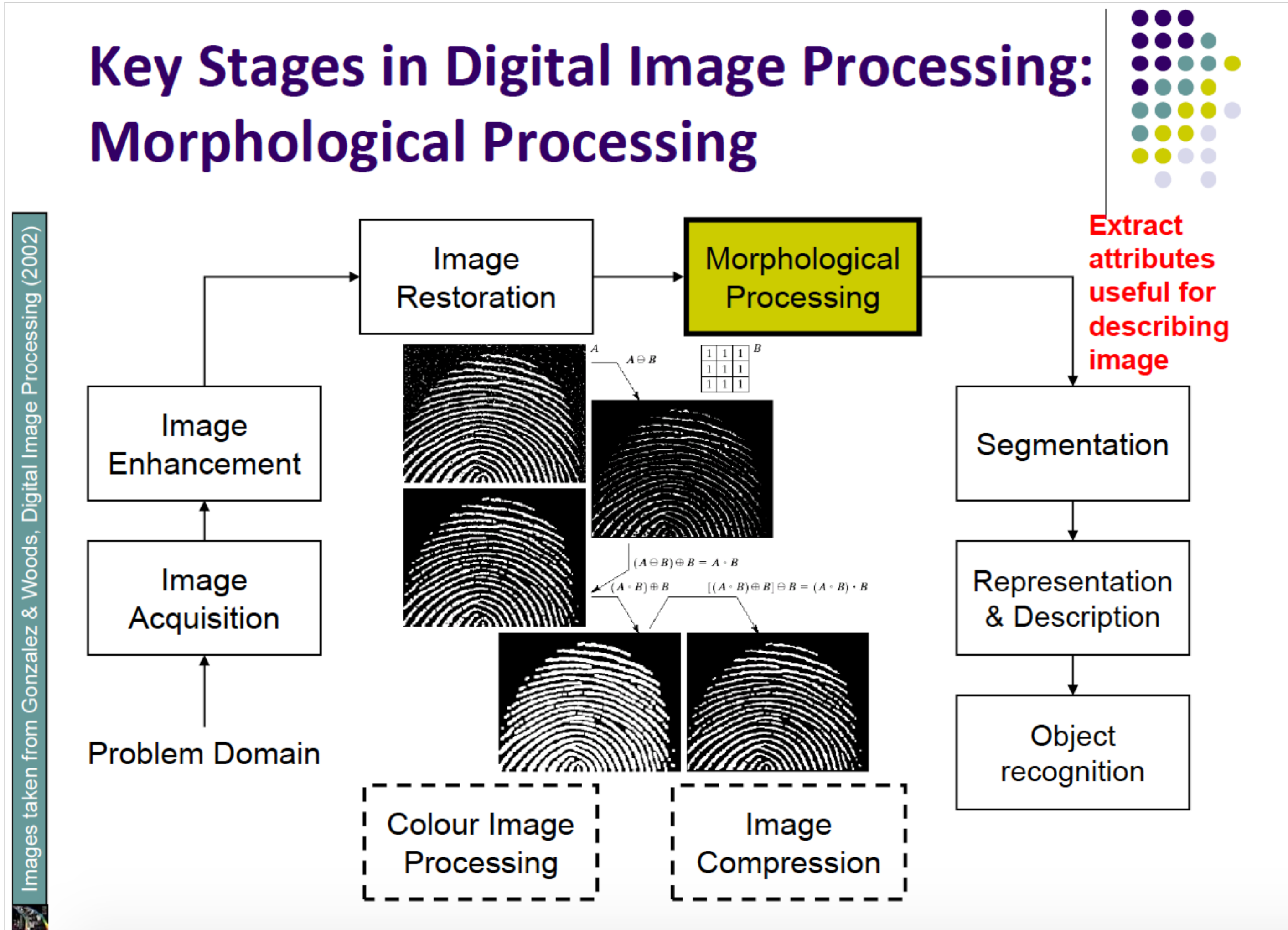


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

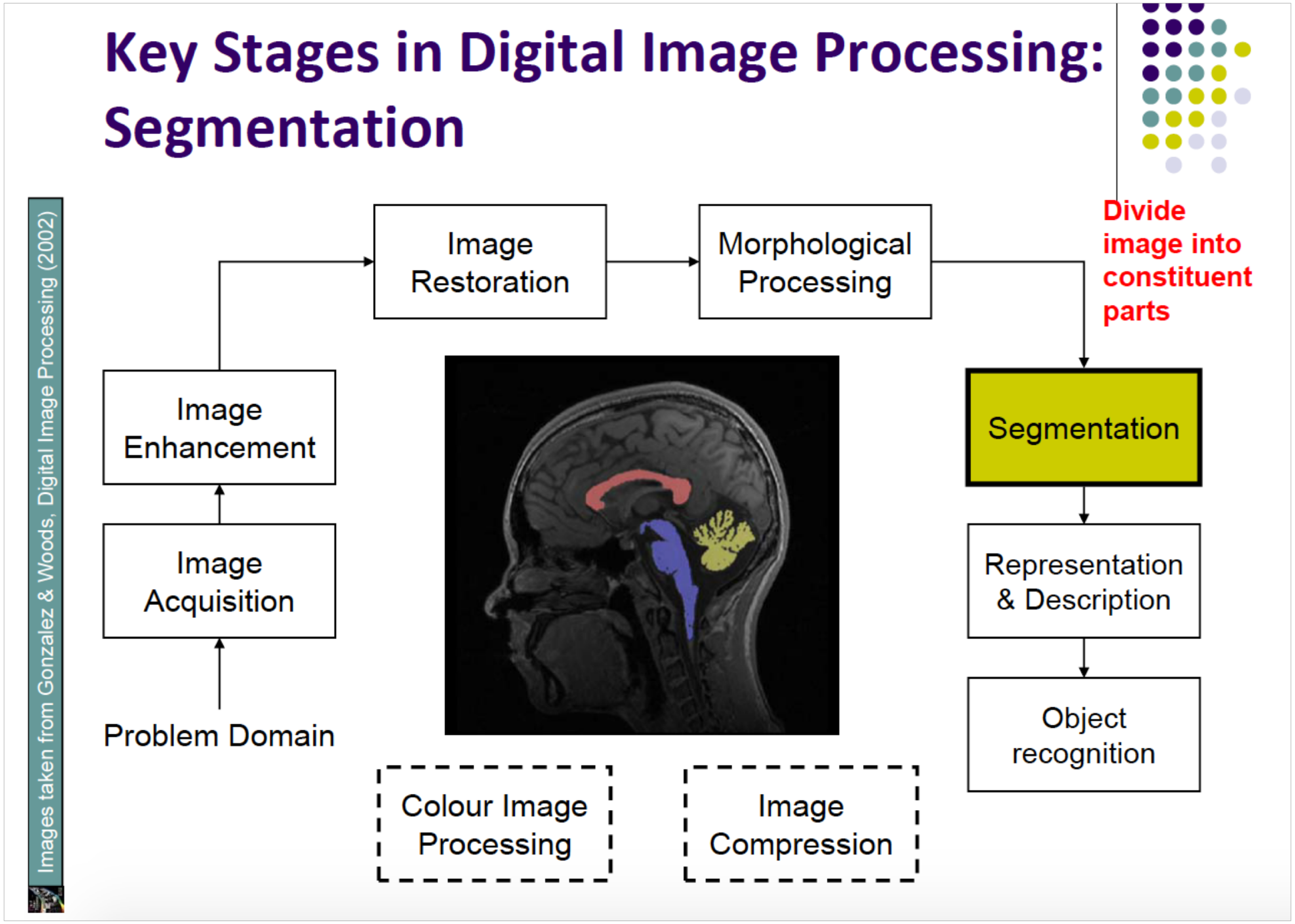
Digital Image Processing



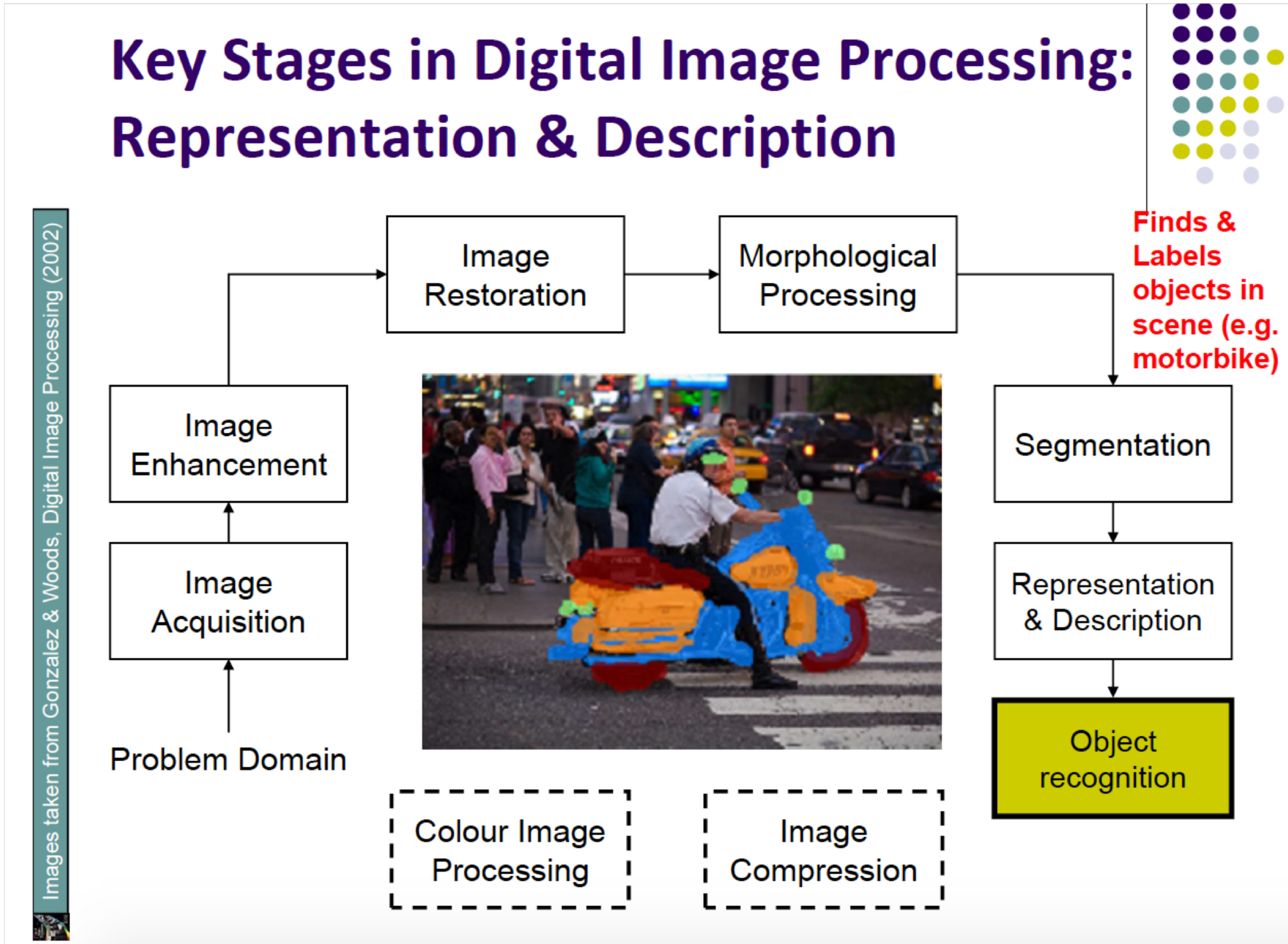
Digital Image Processing



Digital Image Processing

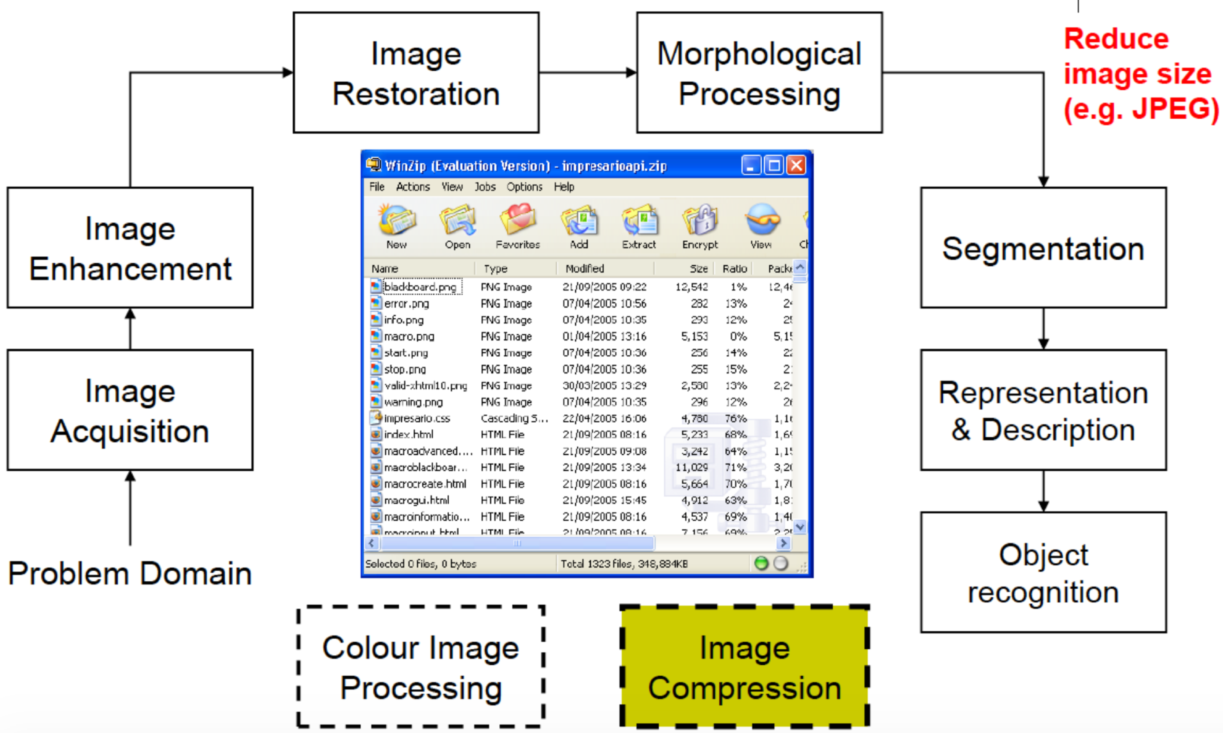


Digital Image Processing



Digital Image Processing

Key Stages in Digital Image Processing: Image Compression



Applications of DIP(digital image processing)

Why do we process images?


Acquire an image
- *Correct aperture and color balance*
- *Reconstruct image from projections*


Prepare for display or printing
- *Adjust image size*
- *Color mapping, gamma-correction, halftoning*

Facilitate picture storage and transmission
- *Efficiently store an image in a digital camera*
- *Send an image from space*

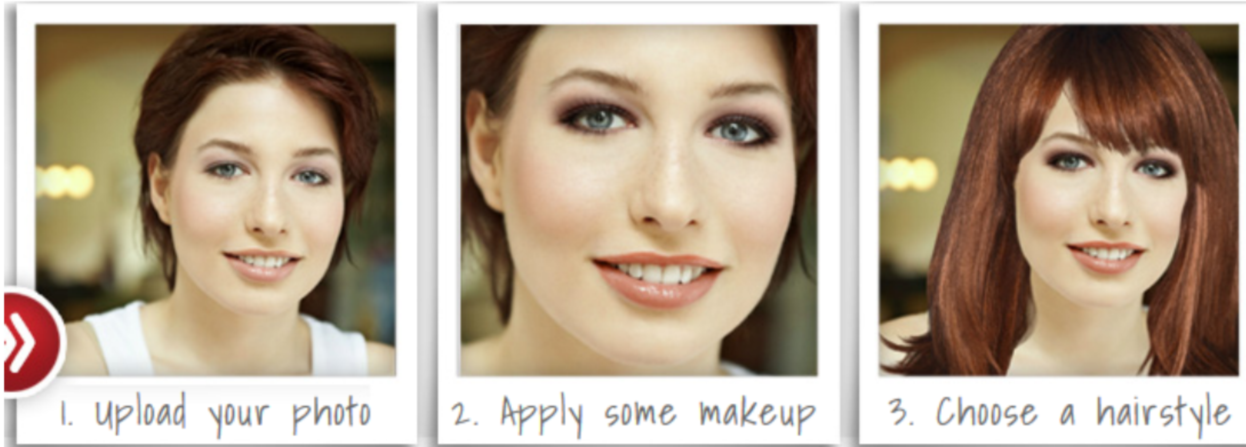


Enhance and restore images
- *Touch up personal photos*
- *Color enhancement for security screening*

Extract information from images
- *Read 2-d bar codes*
- *Character recognition*

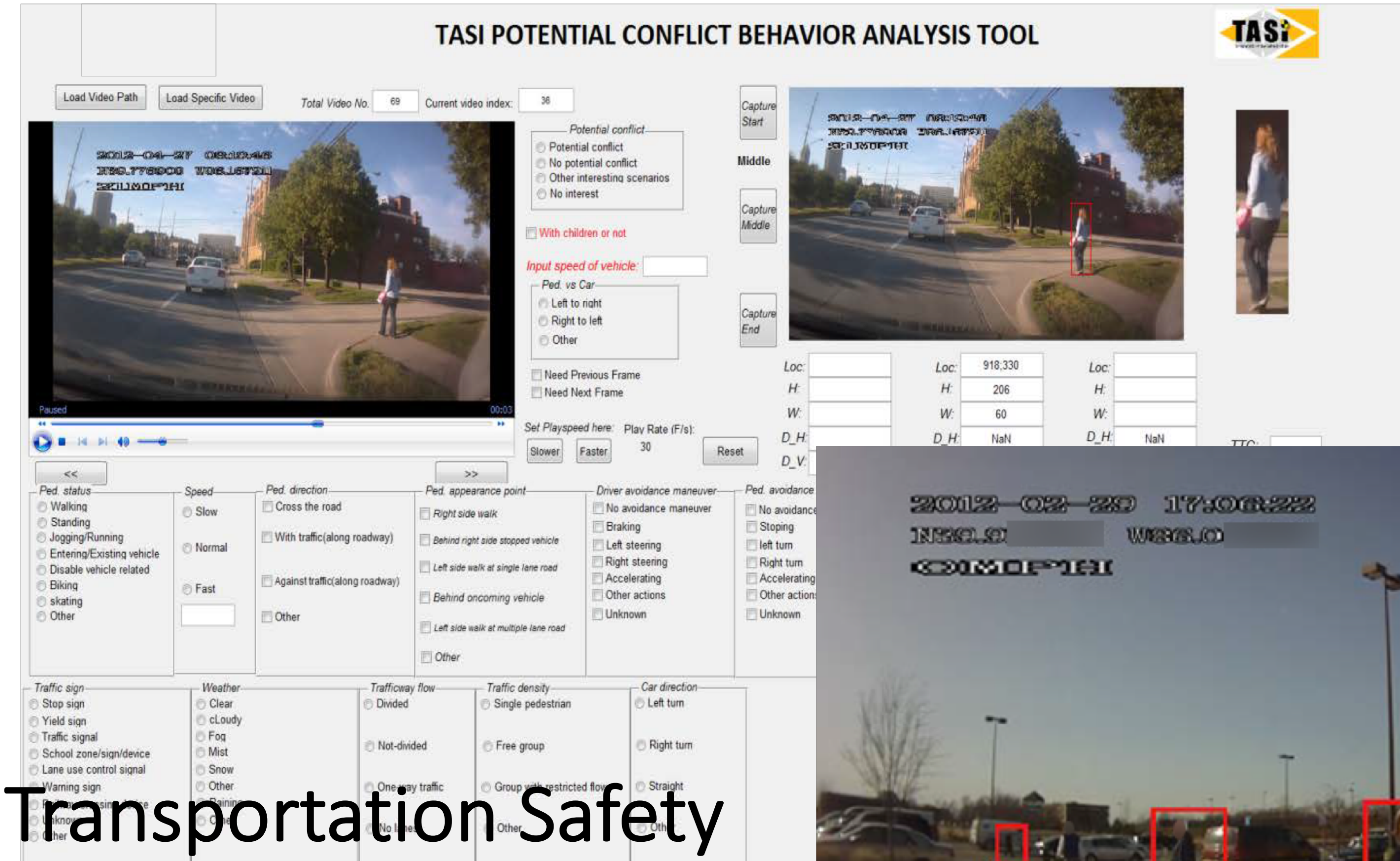
Many more ... image processing is ubiquitous

Applications of DIP

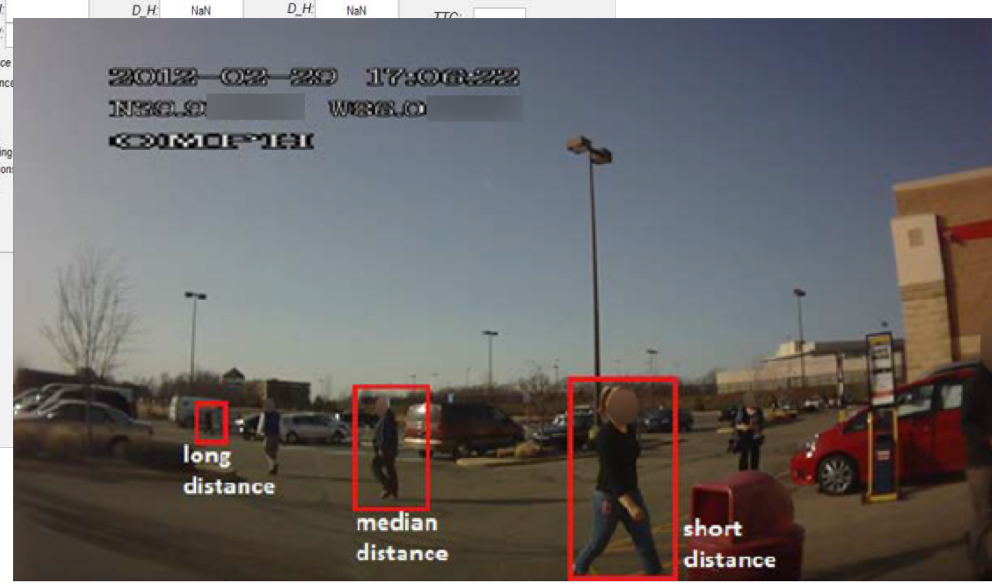


Face Detection & Virtual Makeover

Applications of DIP



Transportation Safety



Pedestrian Detection and Behavior Analysis

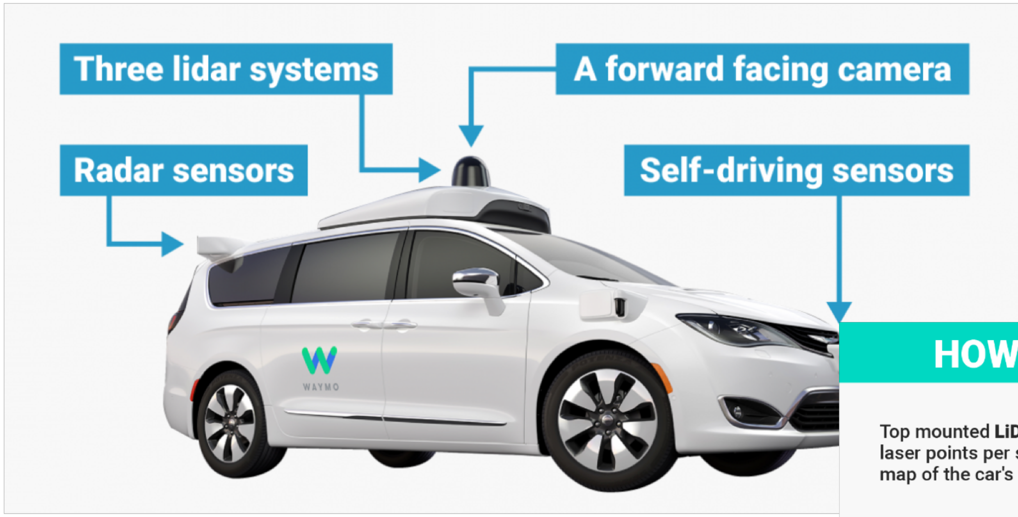
METROPOLITAN STATE UNIVERSITY™ OF DENVER

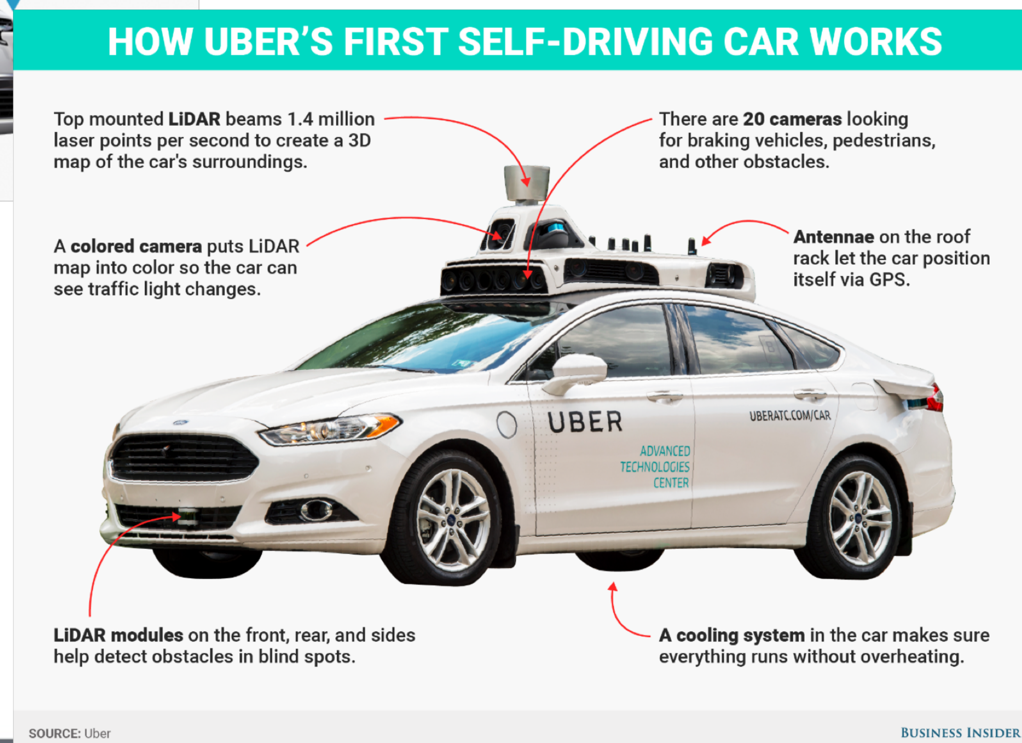
Applications of DIP

Advanced Driver Assistance System (ADAS)

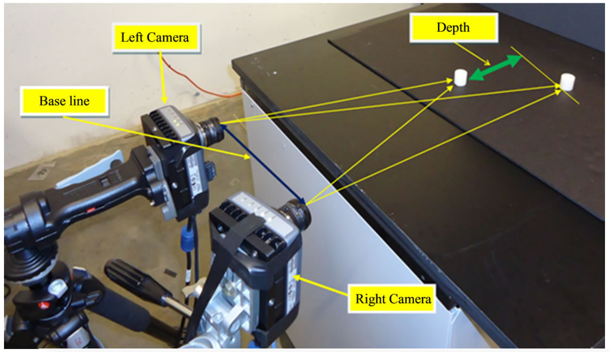
Applications of DIP

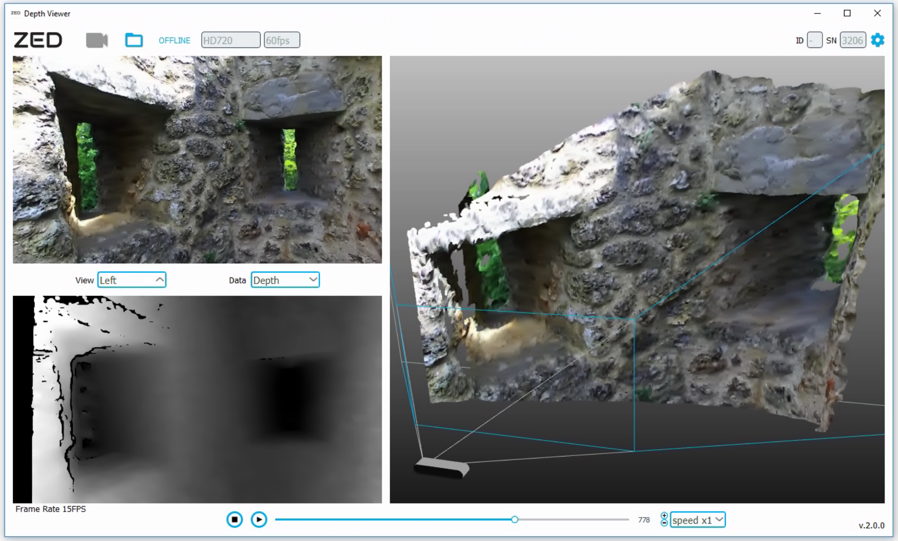
Self-Driving Car





Applications of DIP





Stereo Imaging

Applications of DIP

Image Guided Surgery

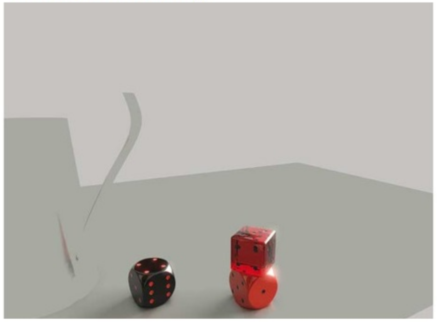
https://www.youtube.com/watch?v=N6Xn--e90oQ

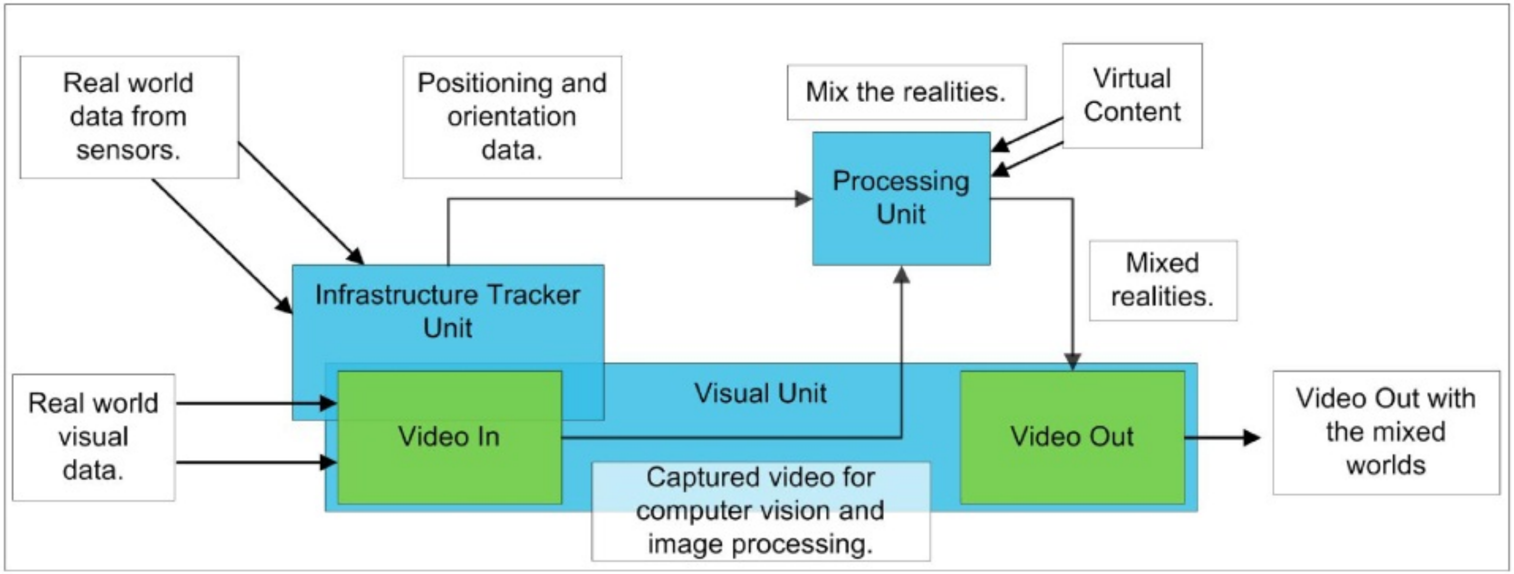
Applications of DIP

Augmented Reality

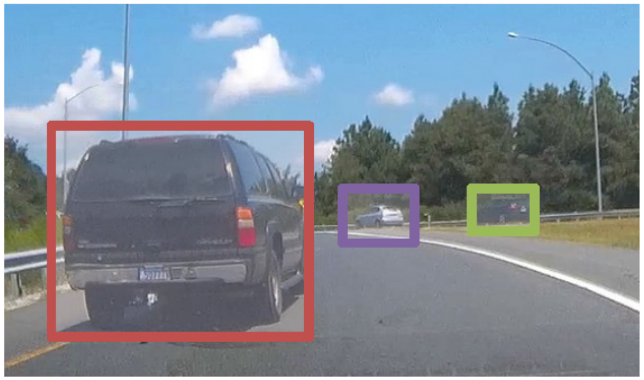
Reality

Virtual Reality





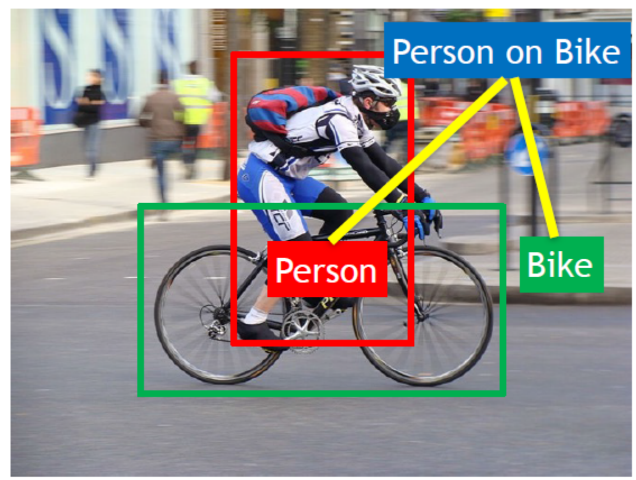
Applications of DIP



- Object detection
- Action classification
- Image captioning
- …

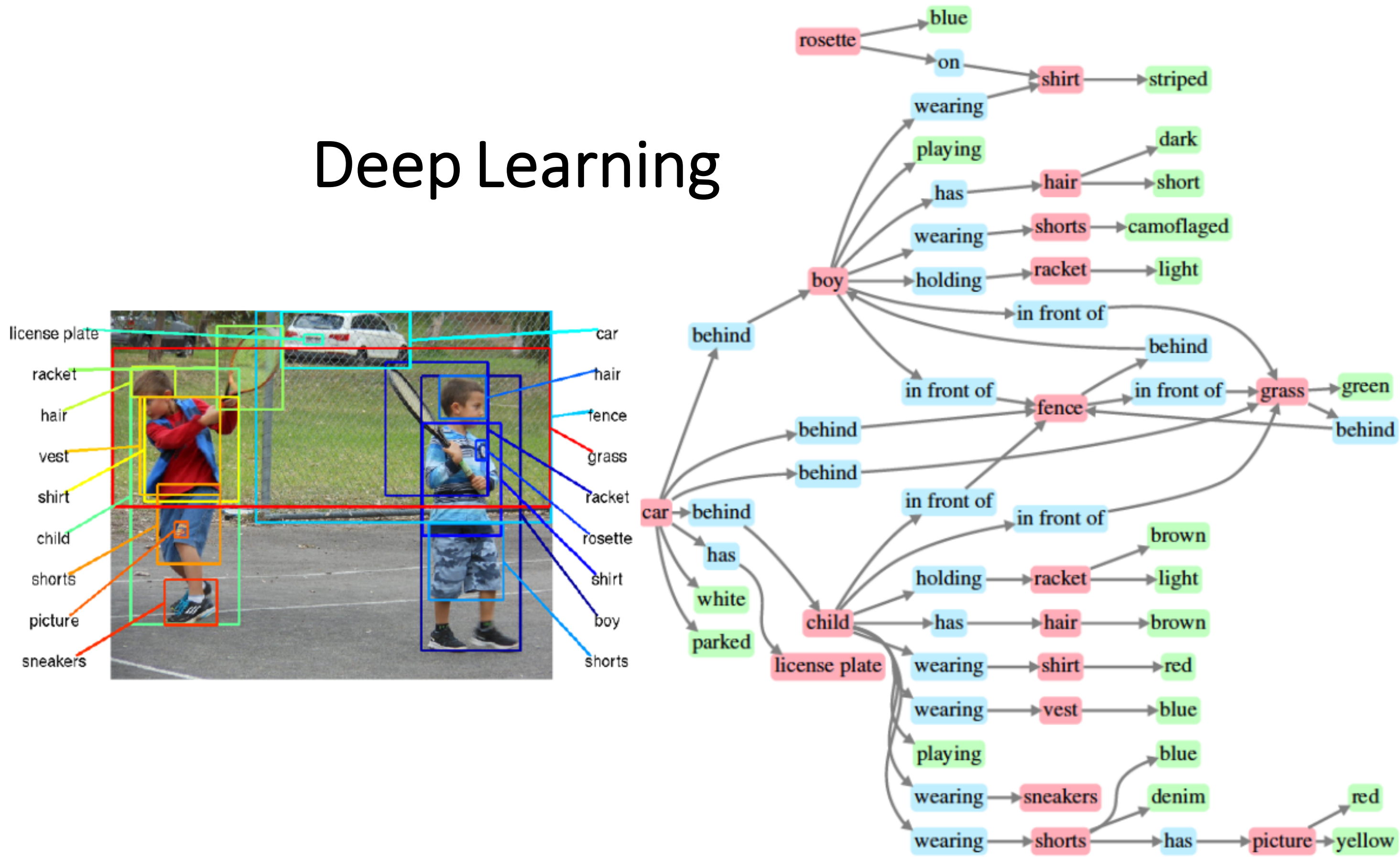
Person

Hammer



Applications of DIP

Deep Learning



Johnson *et al.*, “Image Retrieval using Scene Graphs”, CVPR 2015

Day 2

- Image (color, grayscale, binary), dimension
- Pixel, intensity
- Pixel relationship, distance
- Matlab, Matlab tutorial
- https://www.tutorialspoint.com/matlab/matlab_environment.htm

- Digital Image Processing using Matlab: Chapter 2 2.1-2.8

- Start Basic Image Enhancement Tools (pdf)

Basics of Digital Image Processing

y = \ x =	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	50
44	216	206	211	193	202	207	208	57	69	60	55	77	49	62	61
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

Basics of Digital Image Processing

- Pixel
- Basic relationships between pixels
  - Neighborhood
  - Adjacency
  - Connectivity
  - Regions and boundaries

Grayscale Image

y = \ x =	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	50
44	216	206	211	193	202	207	208	57	69	60	55	77	49	62	61
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

Exercise

- Select your own working path
- Download image “lena_gray_256.tif” from blackboard
- Download the demo code “CS390S_day1_demo.m” from blackboard and run it
- Test the pixel value at pixel location (x,y)=(100 100)
- Check with your neighbor

Basics of Digital Image Processing

- Pixel

- **Image Intensity -Light energy emitted from a unit area in the image.**
- **Image Gray-Level – The relative intensity at each unit area. Between the lowest intensity (Black value) and the highest intensity (White value).**

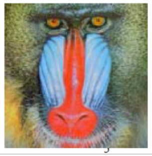
- Three types of images:
  - Binary images

    $I(x,y) \in \{0\ ,\ 1\}$
  - Gray-scale images

    $I(x,y) \in [a\ ,\ b]$
  - Color Images

    $I_R(x,y) \quad I_G(x,y) \quad I_B(x,y)$





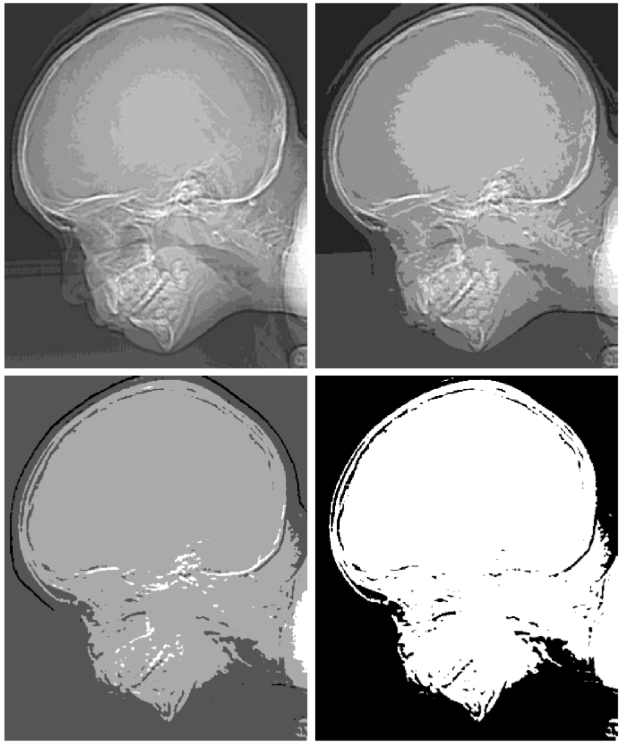
Color Image


Grayscale Image

y = \ x =	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	50
44	216	206	211	193	202	207	208	57	69	60	55	77	49	62	61
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

Basics of Digital Image Processing

- **Image Gray-Level – The relative intensity at each unit area. Between the lowest intensity (Black value) and the highest intensity (White value).**



e f
g h

FIGURE 2.21 (*Continued*) (e)–(h) Image displayed in 16, 8, 4, and 2 intensity levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

Basics of Digital Image Processing

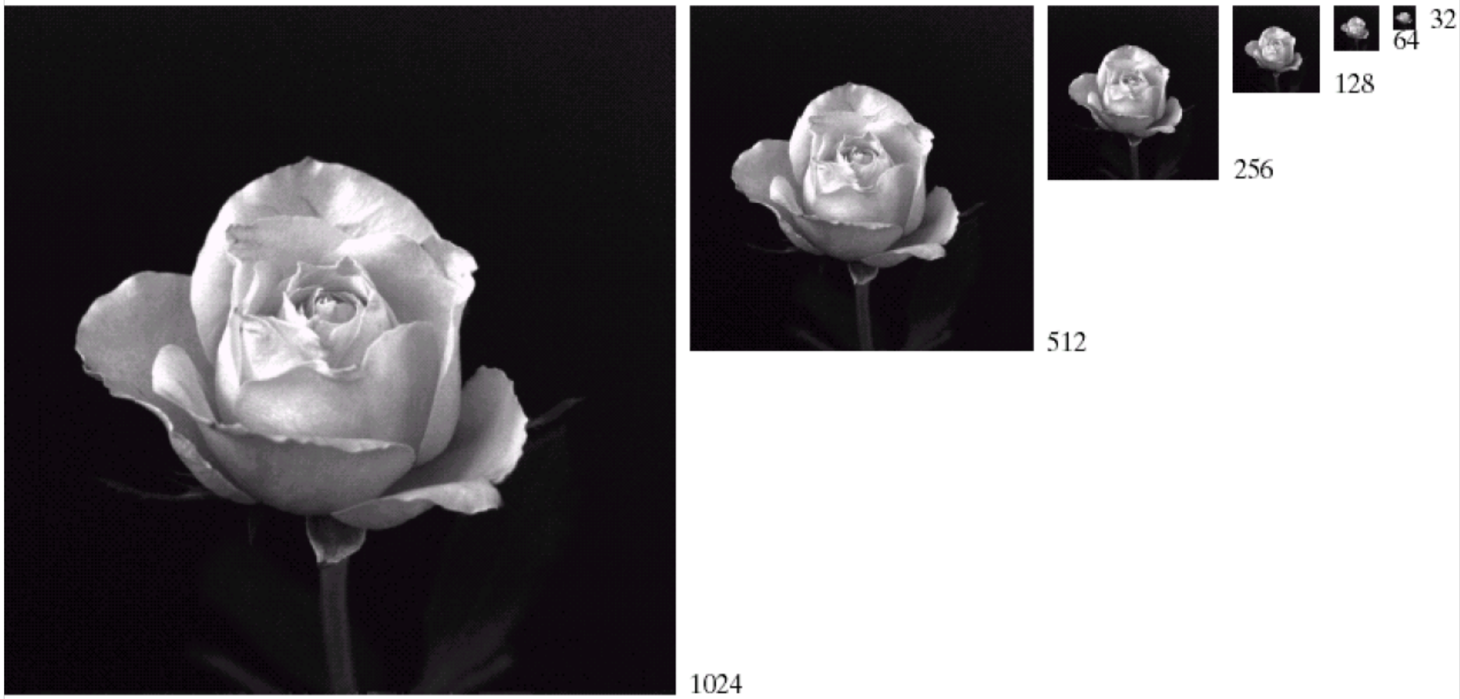
- **Image Gray-Level – The relative intensity at each unit area. Between the lowest intensity (Black value) and the highest intensity (White value).**

The number, b, of bits required to store a digitized image is

$$b = M \times N \times k \tag{2.4-4}$$

Basics of Digital Image Processing

Spatial Resolution



METROPOLITAN STATE UNIVERSITY™ OF DENVER

Basics of Digital Image Processing

Intensity Level Resolution

Low Detail | Medium Detail | High Detail

Basics of Digital Image Processing

How many Bits Per Image Element?

Grayscale (Intensity Images):

Chan.	*Bits/Pix.*	*Range*	*Use*
1	1	0...1	Binary image: document, illustration, fax
1	8	0...255	Universal: photo, scan, print
1	12	0...4095	High quality: photo, scan, print
1	14	0...16383	Professional: photo, scan, print
1	16	0...65535	Highest quality: medicine, astronomy

Color Images:

Chan.	*Bits/Pix.*	*Range*	*Use*
3	24	$[0...255]^3$	RGB, universal: photo, scan, print
3	36	$[0...4095]^3$	RGB, high quality: photo, scan, print
3	42	$[0...16383]^3$	RGB, professional: photo, scan, print
4	32	$[0...255]^4$	CMYK, digital prepress

Special Images:

Chan.	*Bits/Pix.*	*Range*	*Use*
1	16	$-32768...32767$	Whole numbers pos./neg., increased range
1	32	$\pm 3.4 \cdot 10^{38}$	Floating point: medicine, astronomy
1	64	$\pm 1.8 \cdot 10^{308}$	Floating point: internal processing

Basic Relationships Between Pixels

- **Neighbors** of a pixel p at coordinates (x,y)

➢ **4-neighbors of p**, denoted by $\mathbf{N_4(p)}$:
(x-1, y), (x+1, y), (x,y-1), and (x, y+1).

➢ **4 diagonal neighbors of p**, denoted by $\mathbf{N_D(p)}$:
(x-1, y-1), (x+1, y+1), (x+1,y-1), and (x-1, y+1).

➢ **8 neighbors of p**, denoted $\mathbf{N_8(p)}$
$N_8(p) = N_4(p) \cup N_D(p)$

Basic Relationships Between Pixels

- **Adjacency**

  Let S be the set of intensity values

- **4-adjacency**: Two pixels p and q with values from S are 4-adjacent if q is in the set $N_4(p)$.

- **8-adjacency**: Two pixels p and q with values from S are 8-adjacent if q is in the set $N_8(p)$.

Examples: Adjacency and Path

$0_{1,1}$	$1_{1,2}$	$1_{1,3}$		0	1	1		0	1	1
$0_{2,1}$	$2_{2,2}$	$0_{2,3}$		0	2	0		0	2	0
$0_{3,1}$	$0_{3,2}$	$1_{3,3}$		0	0	1		0	0	1

S = {1, 2}

(1,3) and (3,3)
Path : (1,3), (1,2), (2,2), (3,3)

Basic Relationships Between Pixels

- **Connected in S**

  Let S represent a subset of pixels in an image. Two pixels p with coordinates (x_0, y_0) and q with coordinates (x_n, y_n) are said to be **connected in S** if there exists a path

  $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

  Where $\forall i, 0 \le i \le n, (x_i, y_i) \in S$

Basic Relationships Between Pixels

Let S represent a subset of pixels in an image

- For every pixel *p* in S, the set of pixels in S that are connected to *p* is called a ***connected component*** of S.

- If S has only one connected component, then S is called ***Connected Set*.**

- We call R a **region** of the image if R is a connected set

- Two regions, R_i and R_j are said to be ***adjacent*** if their union forms a connected set.
- Regions that are not to be adjacent are said to be ***disjoint.***

Basic Relationships Between Pixels

- **Boundary (or border)**

➢ The ***boundary*** of the region R is the set of pixels in the region that have one or more neighbors that are not in R.

➢ If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns of the image.

- **Foreground and background**

➢ An image contains K disjoint regions, R_k, $k = 1, 2, \ldots, K$. Let R_u denote the union of all the K regions, and let $(R_u)^c$ denote its complement.

All the points in R_u is called **foreground**;

All the points in $(R_u)^c$ is called **background**.

Question 1

- **In the following arrangement of pixels, are the two regions (of 1s) adjacent? (if 8-adjacency is used)**

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

Region 1

Region 2

Question 2

- **In the following arrangement of pixels, are the two parts (of 1s) adjacent? (if 4-adjacency is used)**

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

Part 1

Part 2

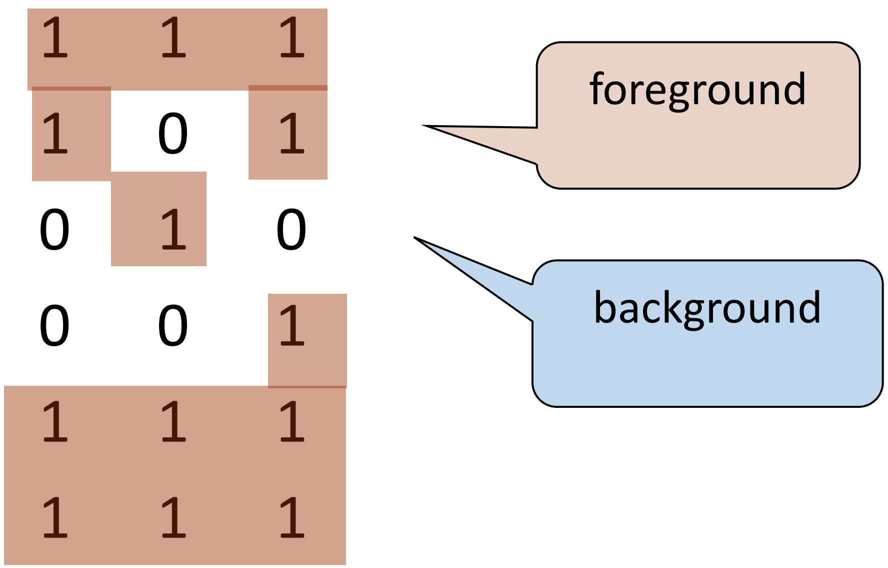
- **In the following arrangement of pixels, the two regions (of 1s) are disjoint (if 4-adjacency is used)**

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

Region 1

Region 2

- **In the following arrangement of pixels, the two regions (of 1s) are disjoint (if 4-adjacency is used)**



Question 3

- **In the following arrangement of pixels, the circled point is part of the boundary of the 1-valued pixels if 8-adjacency is used, true or false?**

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

Question 4

- **In the following arrangement of pixels, the circled point is part of the boundary of the 1-valued pixels if 4-adjacency is used, true or false?**

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

Distance Measures

- Given pixels *p, q* and *z* with coordinates (x, y), (s, t), (u, v) respectively, the distance function D has following properties:

a. $D(p, q) \geq 0$ [$D(p, q) = 0$, iff $p = q$]

b. $D(p, q) = D(q, p)$

c. $D(p, z) \leq D(p, q) + D(q, z)$

Distance Measures

The following are the different Distance measures:

a. **Euclidean Distance** :

$D_e(p, q) = [(x-s)^2 + (y-t)^2]^{1/2}$

b. City Block Distance:

$D_4(p, q) = |x-s| + |y-t|$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

c. Chess Board Distance:

$D_8(p, q) = \max(|x-s|, |y-t|)$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Question 5

- **In the following arrangement of pixels, what's the value of the chessboard distance between the circled two points?**

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

Question 6

- **In the following arrangement of pixels, what's the value of the city-block distance between the circled two points?**

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

Matlab basics

- Image (color gray binary), dimension
- Pixel, intensity
- Pixel relationship, distance
- **Matlab, Matlab tutorial**
- https://www.tutorialspoint.com/matlab/matlab_environment.htm

- **Digital Image Processing using Matlab**: Chapter 2 2.1-2.8

- Start Basic Image Enhancement Tools (pdf)

Matlab basics

Some Basic Matlab Commands

- Some basic image related commands: imshow; image; imagesc; imfinfo; imread; imwrite; axis off; axis image;
- Some basic program related commands:
  - if..else..end; for..end; while..end; switch..case..otherwise..end; continue; break;
  - Basic logic commands: ==, |, &, >, <, >=
  - function
  - Other useful commands: find; length; dir
- Some basic math commands:
  - Sum; max; min; mean; mean2; std; std2
  - Matrix calculations: 2 different ways depend on the applications

```
A+B
A-B
A*B      A.*B
A/B      A./B
A\B      A.\B
A^B      A.^B
A'       A.'
```

http://matlab.izmiran.ru/help/techdoc/ref/arithmeticoperators.html

Matlab matrix/image operation

- **Array vs. Matrix Operation**

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Array product operator

$$A \;.*\; B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Array product

Matrix product operator

$$A * B = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Matrix product

Exercise

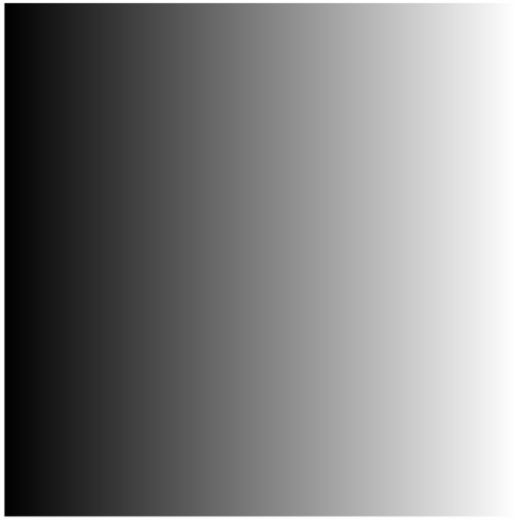
```
A=[1 2 3; 1 2 3];
A(2,2)
A(:)
A(:,1)
```

Imshow(im); % if integer expecting [0 255]
Imshow(im); % if double, expecting [0 1]
Figure, index -> intensity , settings
Image(im); % color image works fine.
Image(im); colormap(gray(256)) ; % for grayscale
imagesc(im);

Exercise

- The image of a ramp (256×256):

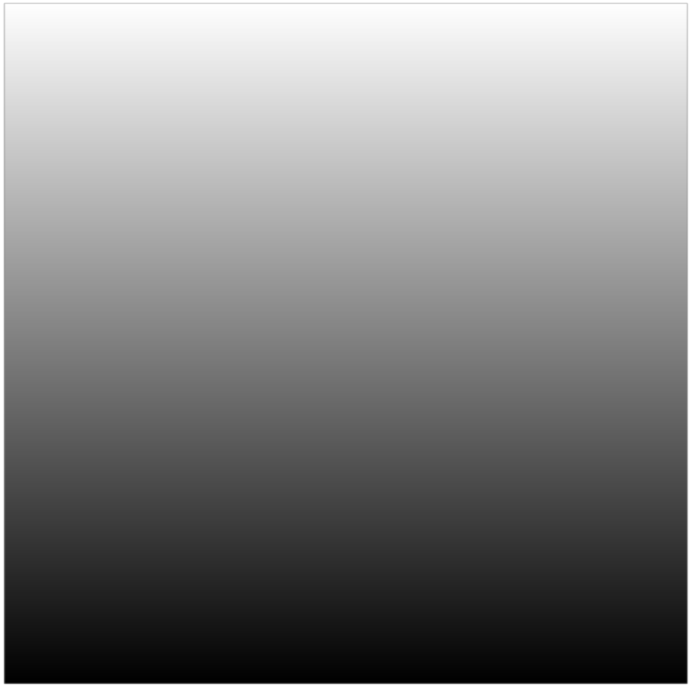
$$\mathbf{A} = \left.\begin{bmatrix} 0 & 1 & 2 & \dots & 255 \\ 0 & 1 & 2 & \dots & 255 \\ \vdots & & & & \\ 0 & 1 & 2 & \dots & 255 \end{bmatrix}\right\} 256 \text{ rows}$$

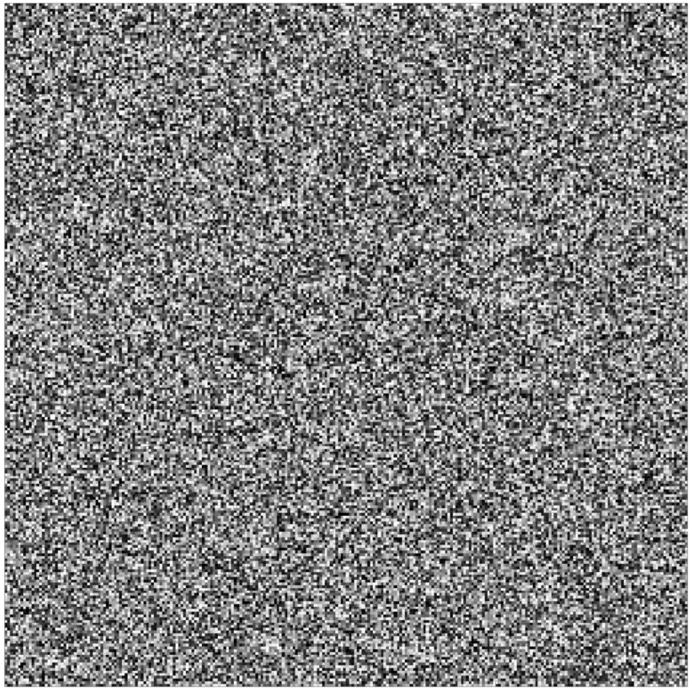


$>>$ for $i = 1 : 256$
  for $j = 1 : 256$
    $A(i, j) = j - 1;$
    end
  end
$>>$ image(A);
$>>$ colormap(gray(256));
$>>$ axis('image');

Exercise rand() randn()

How about this?

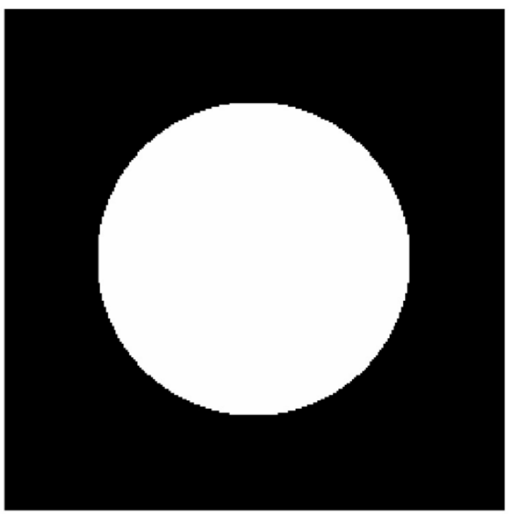




Exercise

The image of a circle (256×256) of radius 80 pixels centered at $(128, 128)$:

$$B(i,j) = \begin{cases} 255 & \text{if } \sqrt{(i-128)^2 + (j-128)^2} < 80 \\ 0 & \text{otherwise} \end{cases}$$

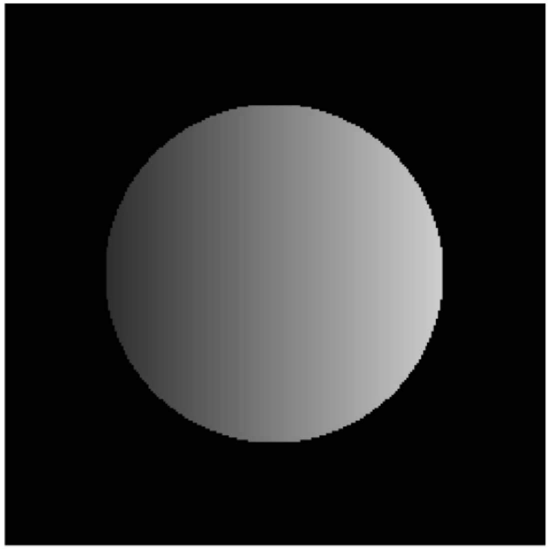


```
>> for i = 1 : 256
     for j = 1 : 256
       dist = ((i - 128)^2 + (j - 128)^2)^(.5);
       if (dist < 80)
         B(i, j) = 255;
       else
         B(i, j) = 0;
         end
       end
     end
>> image(B);
>> colormap(gray(256));
>> axis('image');
```

Exercise

The image of a "graded" circle (256×256):

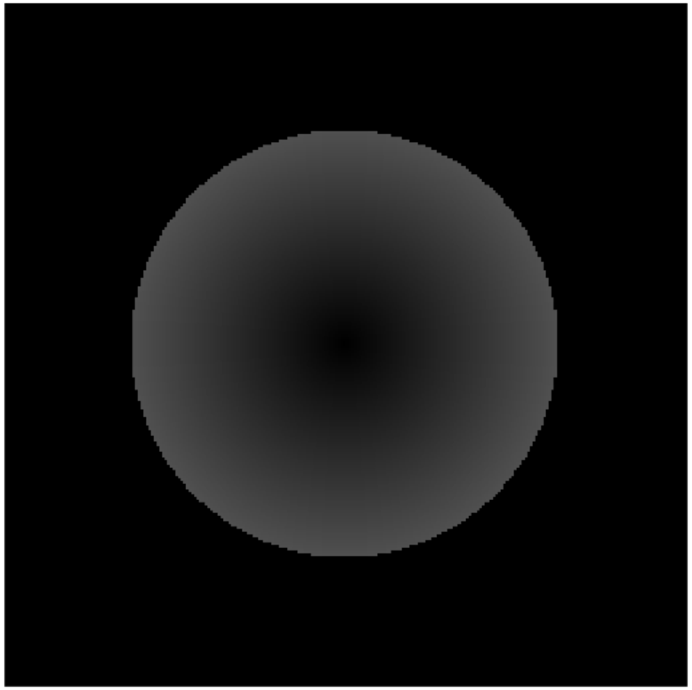
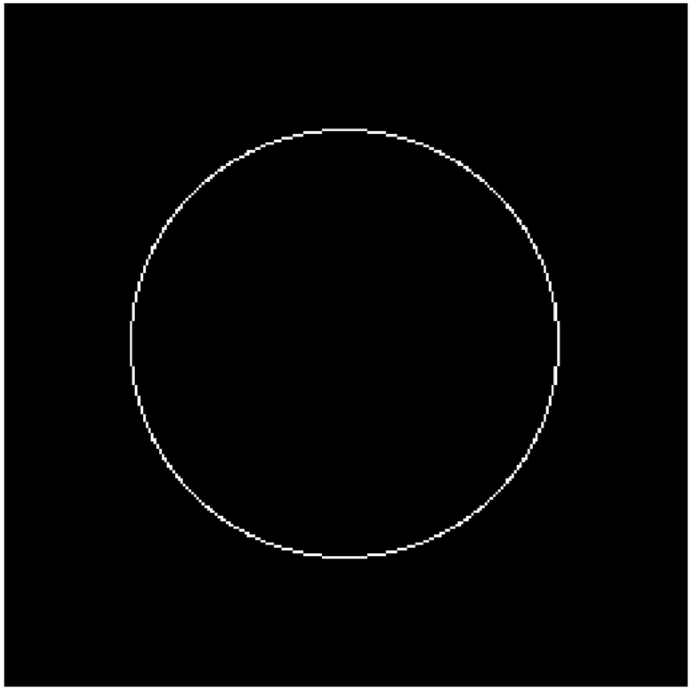
$$C(i,j) = A(i,j) \times B(i,j)/255$$



$$
\begin{array}{ll}
>> & \text{for } i = 1:256 \\
& \quad \text{for } j = 1:256 \\
& \quad\quad C(i,j) = A(i,j) * B(i,j)/255; \\
& \quad\quad \text{end} \\
& \quad \text{end} \\
>> & \text{image}(C); \\
>> & \text{colormap(gray(256));} \\
>> & \text{axis('image');}
\end{array}
$$

Exercise

- How about this?

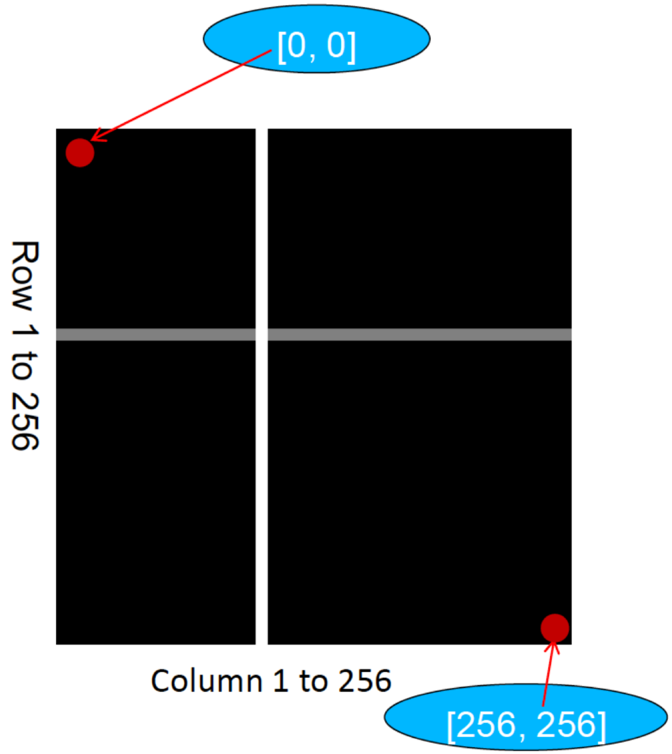
 

Exercise

- How about this?

How to build a matrix (or image)?

Answer…

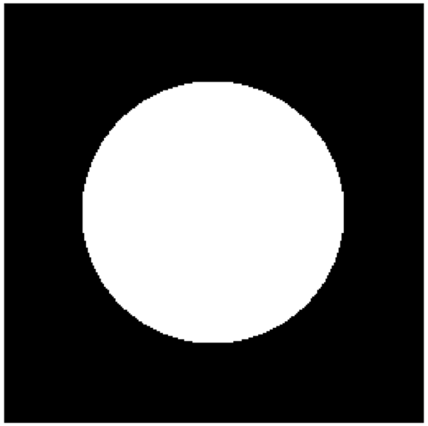


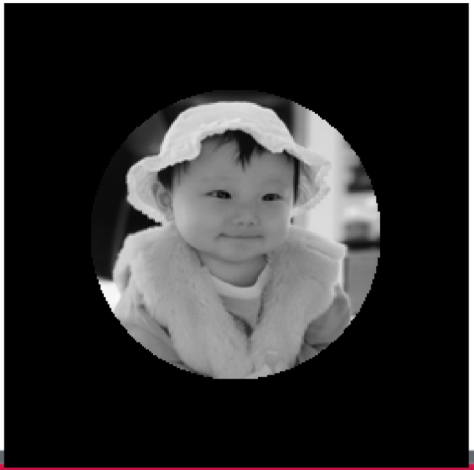
Day 2

- Image (color gray binary), dimension
- Pixel, intensity
- Pixel relationship, distance
- **Matlab, Matlab tutorial**
- https://www.tutorialspoint.com/matlab/matlab_environment.htm
- **Start Digital Image Processing using Matlab**: Chapter 2 2.1-2.8

Assignment 1

- Assignment 1 (get to know matlab and show one image)
  - Download and install matlab
  - Practice the exercises
  - Change the “graded circle” image by inserting your own picture into the center part. Or simply modify one your own image, emphasis the center part by making the pixels outside the circle to be black. You could set circle radius to any value.





Exercise

A=[1 2 3; 1 2 3];
A(2,2)
A(:)
A(:,1)

Imshow(im); % if integer expecting [0 255]
Imshow(im); % if double, expecting [0 1]
Figure, index -> intensity, color; settings
Image(im); % color image works fine.
Image(im); colormap(gray(256)) ; % for grayscale
Imagesc(im); aixs equal; colormap(gray(256)) ; % for grayscale

Day 3

- Image (color gray binary), dimension
- Pixel, intensity
- Pixel relationship, distance
- **Matlab, Matlab tutorial**
- https://www.tutorialspoint.com/matlab/matlab_environment.htm

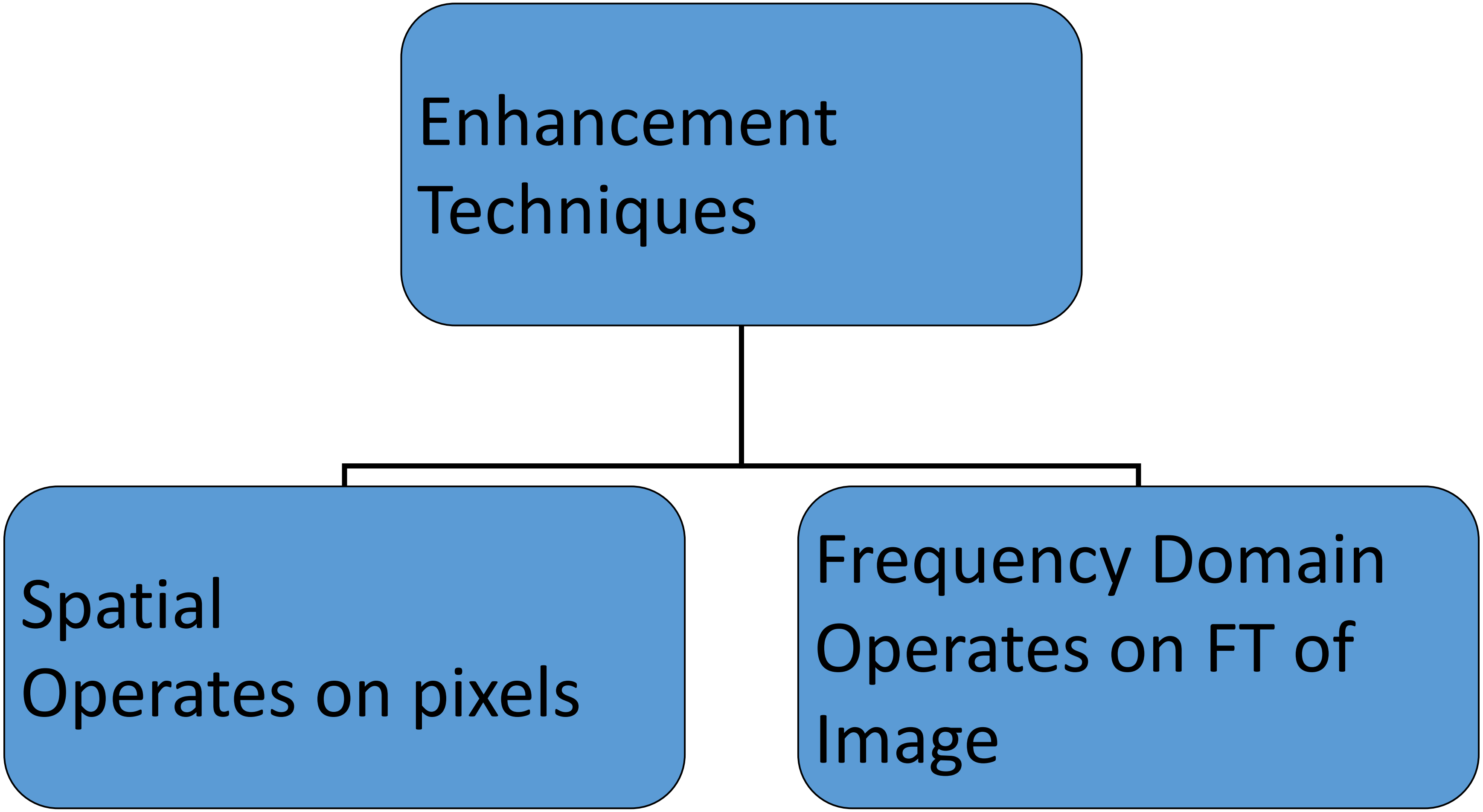
- **Digital Image Processing using Matlab: Chapter 2 2.1-2.8**

- **Basic Image Enhancement Tools (pdf)**
- **Assigment 1**

Image Enhancement in Spatial Domain

- Image Histogram
- Image Negative
- Contrast Stretching
- Power-Law Transformation
- Histogram Equalization
- Local Enhancement
- Image Subtraction
- Image Averaging
- Image Smoothing
- Image Sharpening

Image enhancement

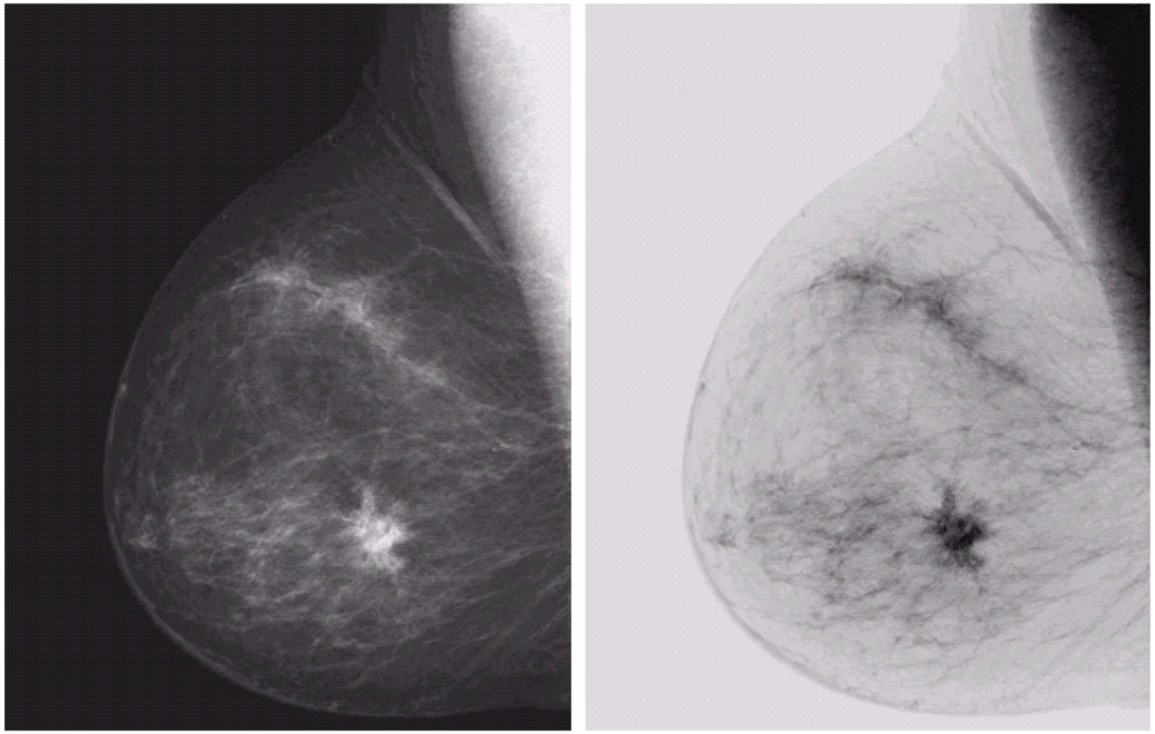


Spatial Domain Methods

- In these methods a operation (linear or non-linear) is performed on the pixels in the neighborhood of coordinate (x,y) in the input image F, giving enhanced image F'
- Neighborhood can be any shape but generally it is rectangular (3x3, 5x5, 9x9 etc)

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

Image negative



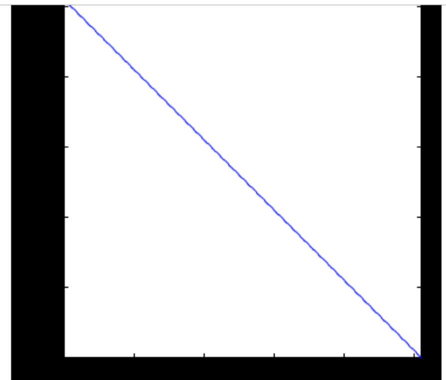
a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Image Negative: $s = L - 1 - r$

Image negative

Demo

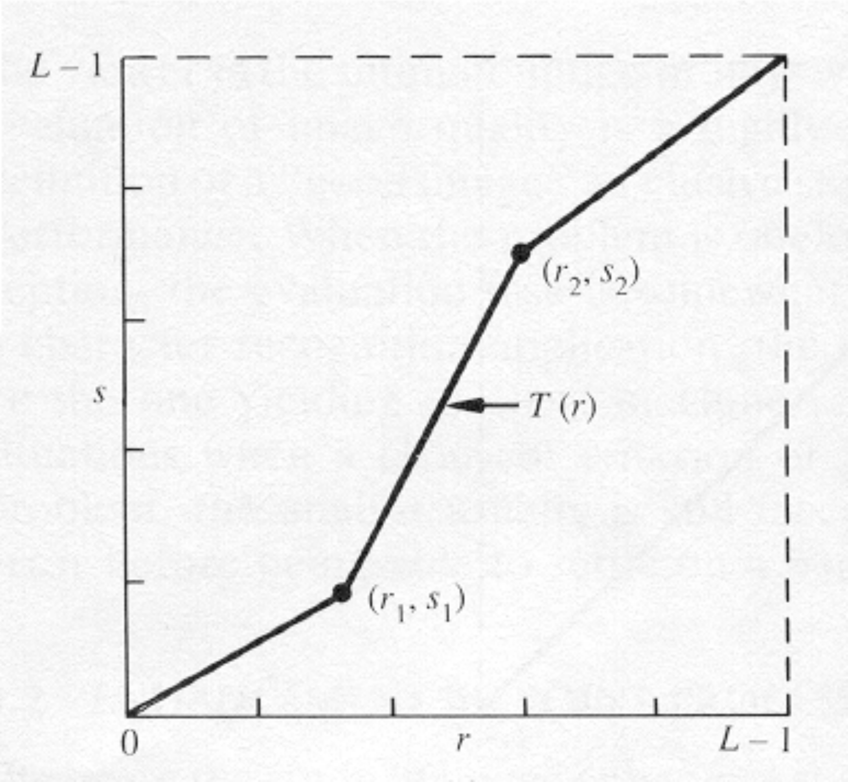


S=255-r

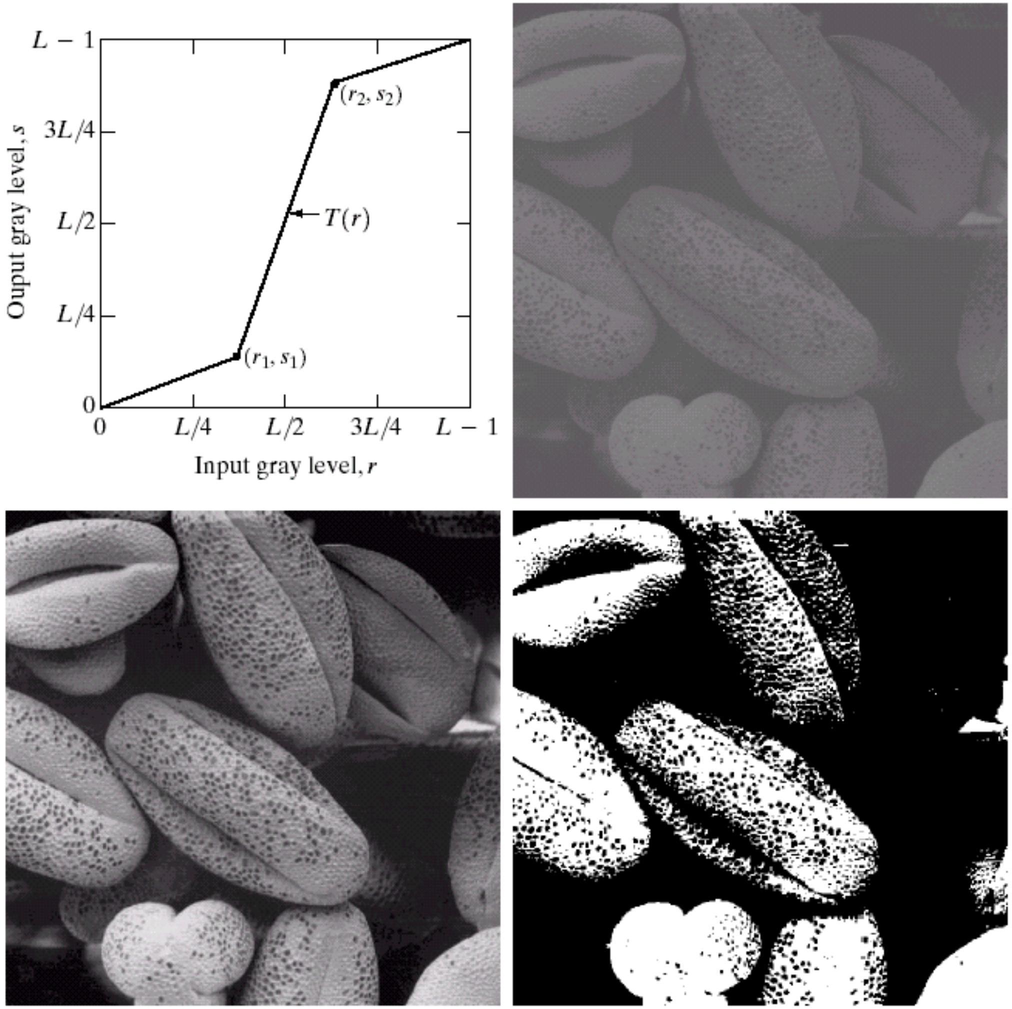
Image Negative: $s = L - 1 - r$

Contrast Stretching

- To increase the dynamic range of the gray levels in the image being processed.



Example



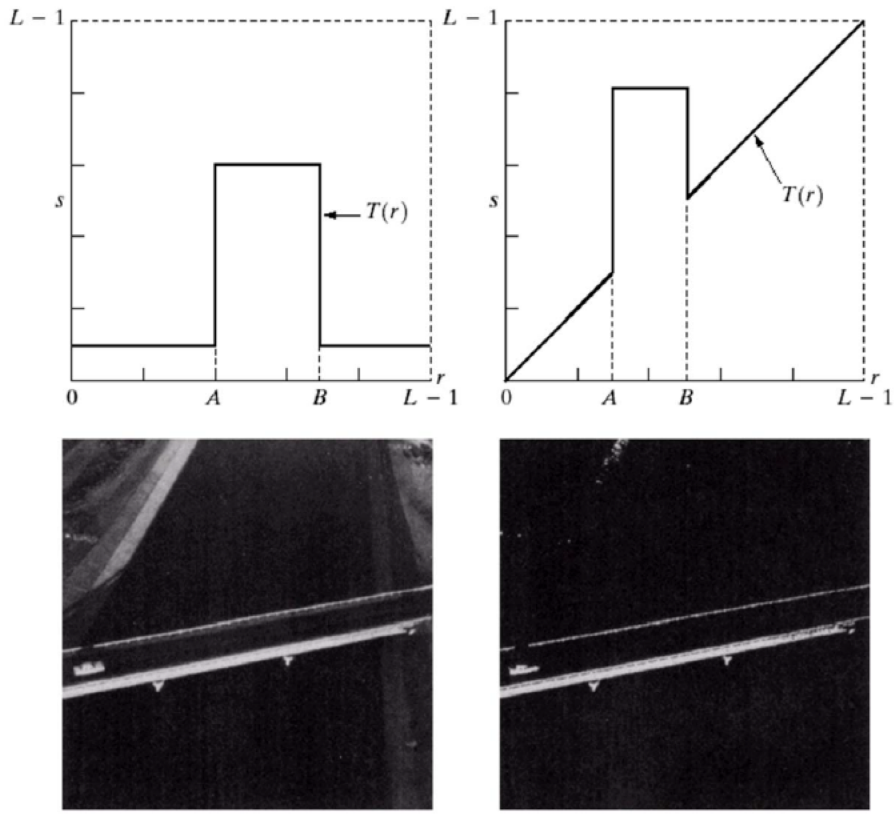
a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Example

- The locations of (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.
  - If $r_1= s_1$ and $r_2= s_2$ the transformation is a linear function and produces no changes.
  - If $r_1=r_2$, $s_1=0$ and $s_2=L-1$, the transformation becomes a thresholding function that creates a binary image.
  - Intermediate values of (r_1,s_1) and (r_2,s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
  - Generally, $r_1 \le r_2$ and $s_1 \le s_2$ is assumed.

Contrast Stretching for image highlighting



Power Law Transformation

- $s = cr^{\gamma}$
- C, γ : positive constants
- Gamma correction

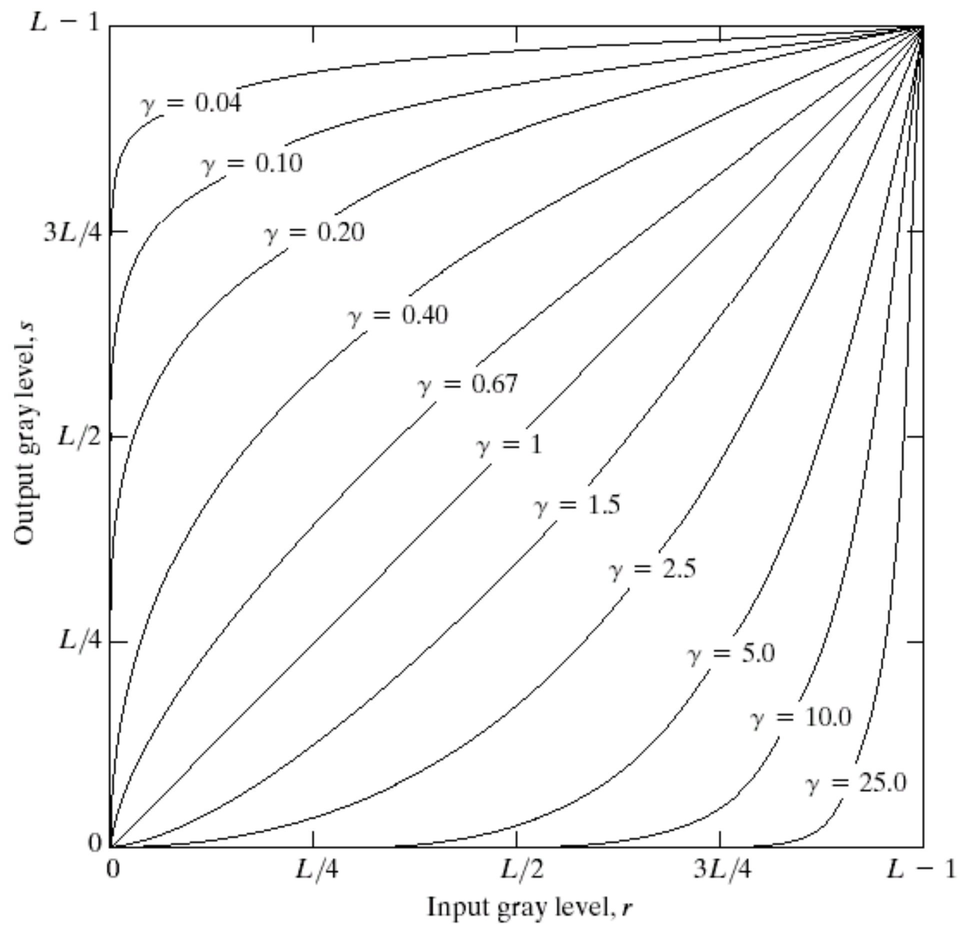
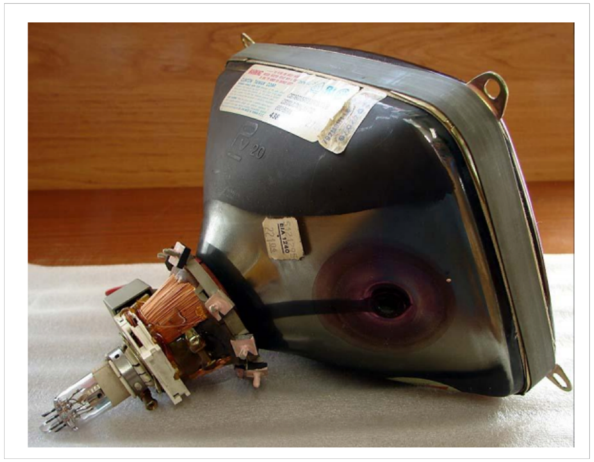


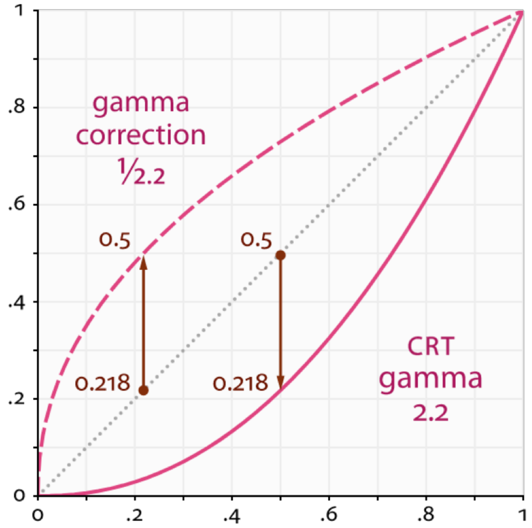
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Power Law Transformation

https://www.siggraph.org/education/materials/HyperGraph/color/gamma_correction/gamma_intro.html



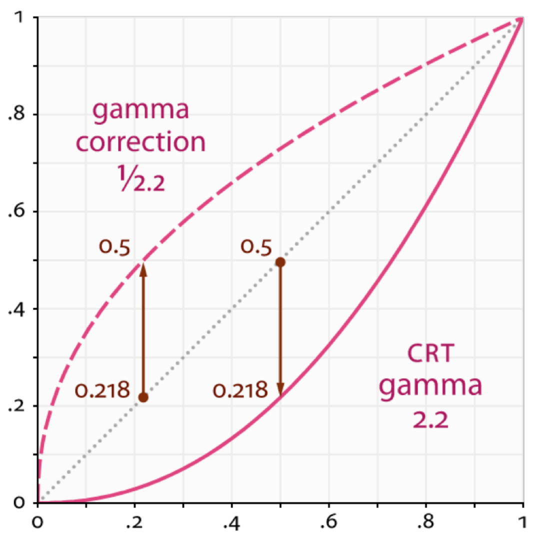
The effect can be overcome by applying an end-to-end power function whose exponent is about 1.1 or 1.2.



Almost every computer monitor, from whatever manufacturer, has one thing in common. They all have a intensity to voltage response curve which is roughly a 2.5 power function.

Power Law Transformation

Gamma correction function is a function that maps luminance levels to compensate the non-linear luminance effect of display devices (or sync it to human perceptive bias on brightness).



https://www.epaperpress.com/monitorcal/gamma.html

https://www.eizo.com/library/basics/lcd_display_gamma/

Bit-Plane Slicing

- To highlight the contribution made to the total image appearance by specific bits.
    - i.e. Assuming that each pixel is represented by 8 bits, the image is composed of 8 1-bit planes.
    - Plane 0 contains the least significant bit and plane 7 contains the most significant bit.
    - Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details.

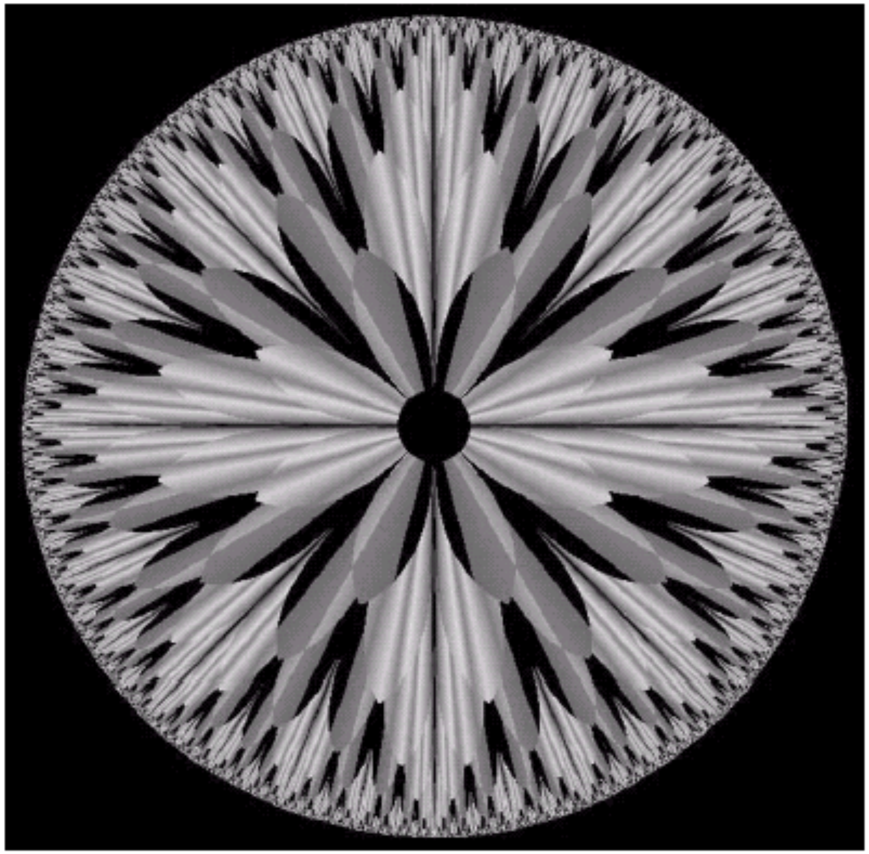


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

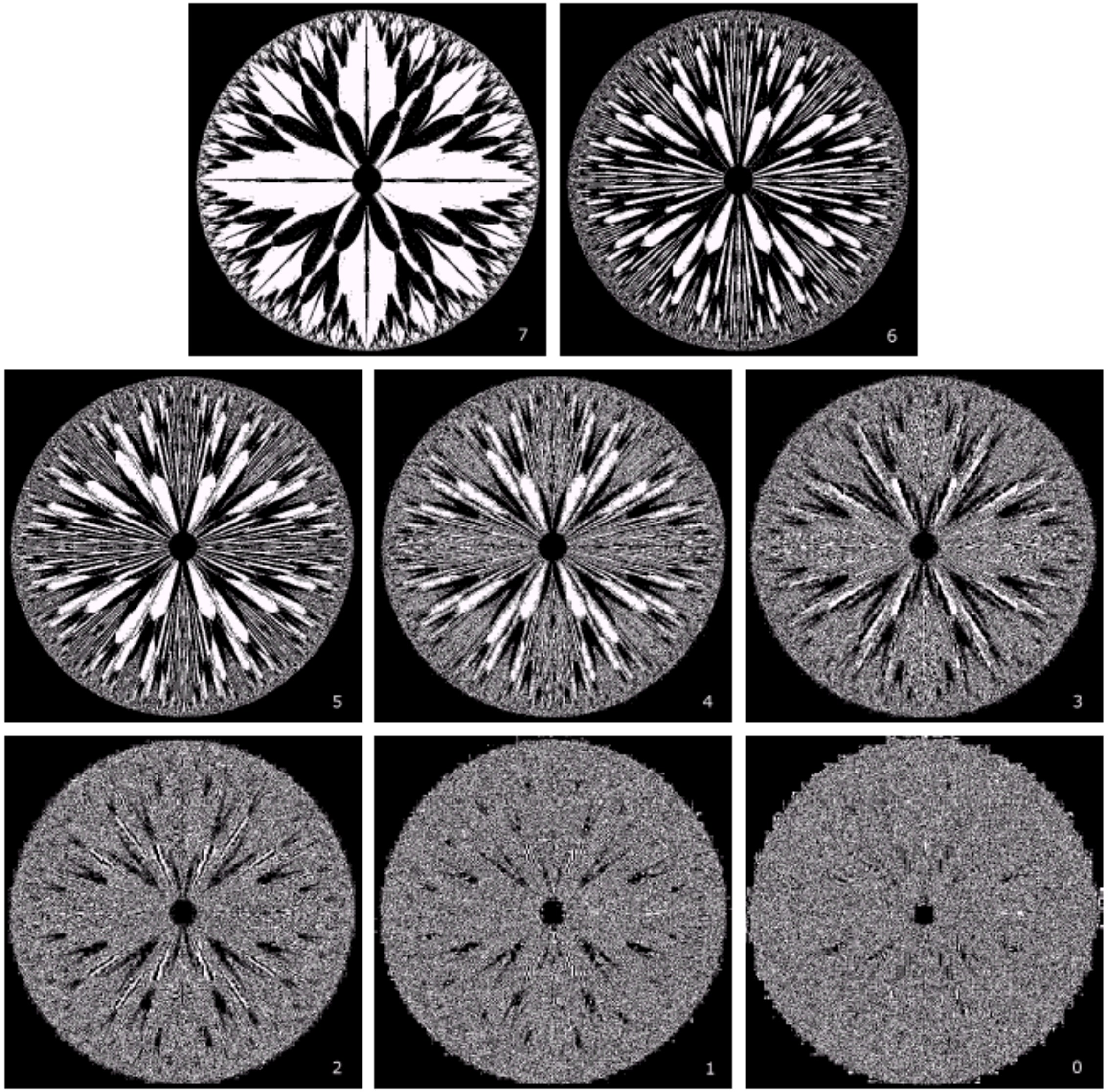
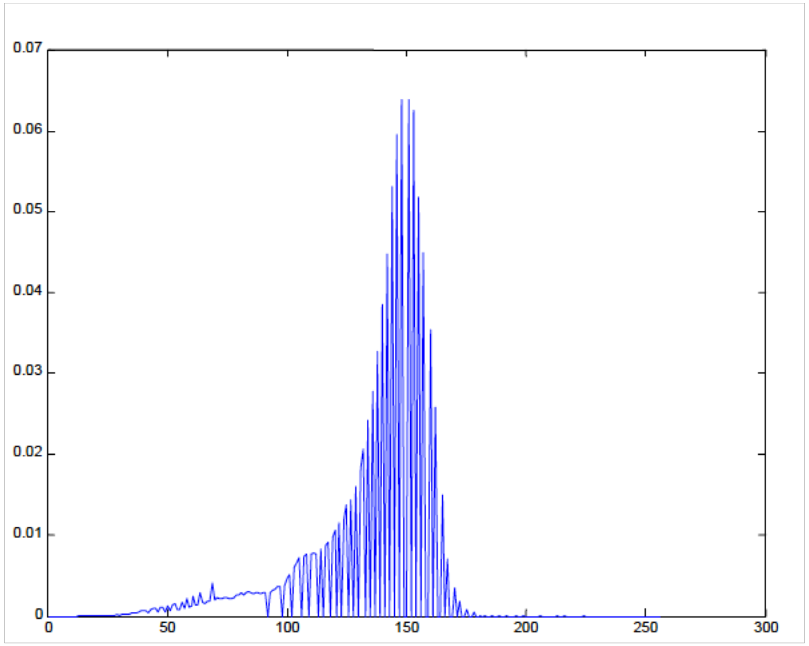


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

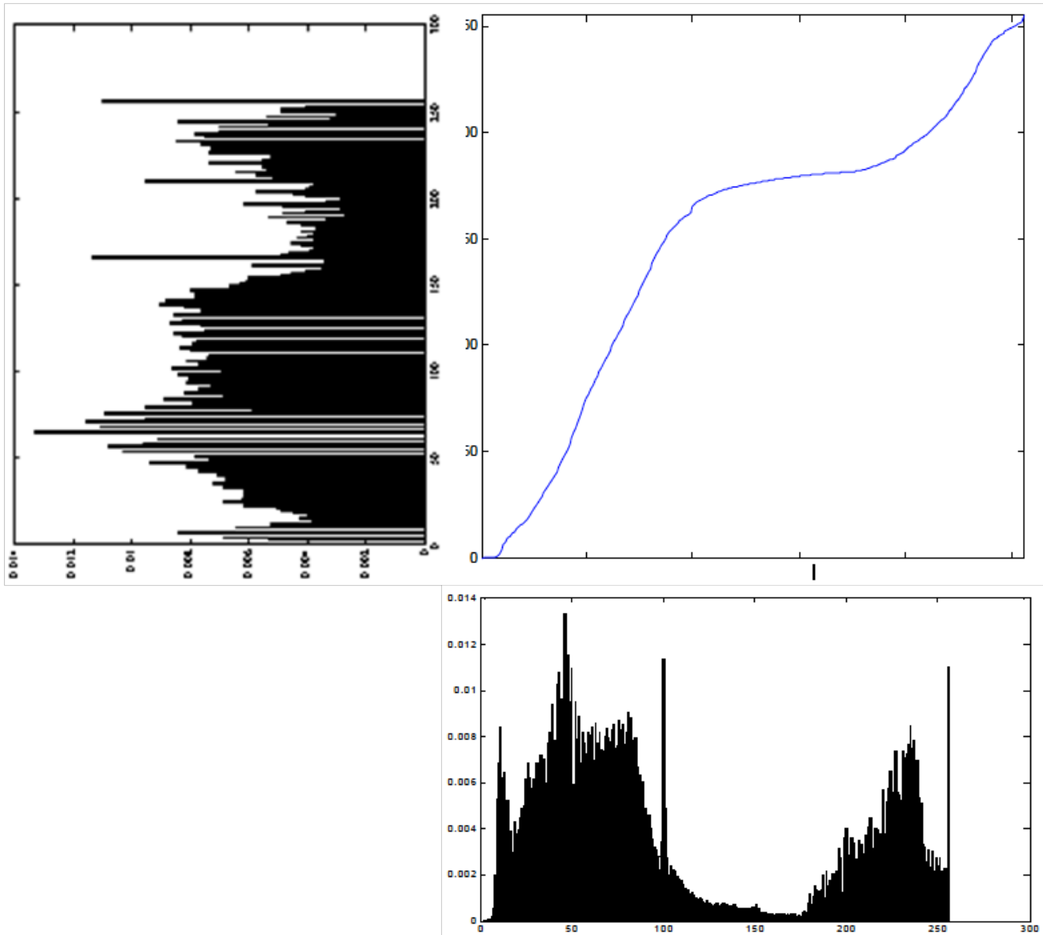
Histogram Processing

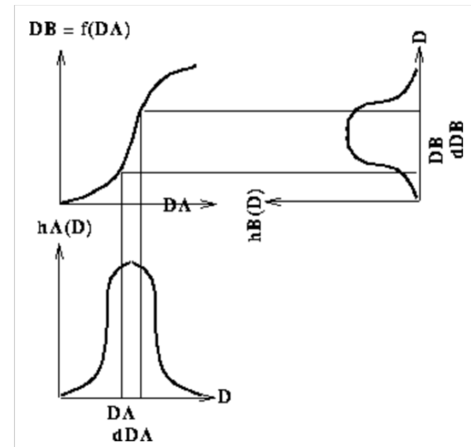
- The histogram of a digital image with gray levels from 0 to L-1 is a discrete function $h(r_k)=n_k$, where:
  - r_k is the kth gray level
  - n_k is the # pixels in the image with that gray level
  - n is the total number of pixels in the image
  - k = 0, 1, 2, …, L-1
- Normalized histogram: $p(r_k)=n_k/n$
  - sum of all components = 1

Histogram Processing



Histogram Equalization

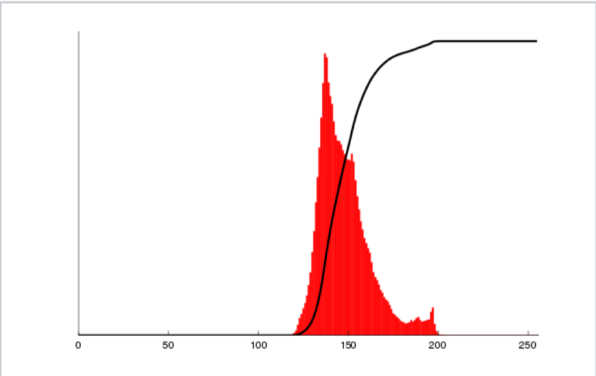




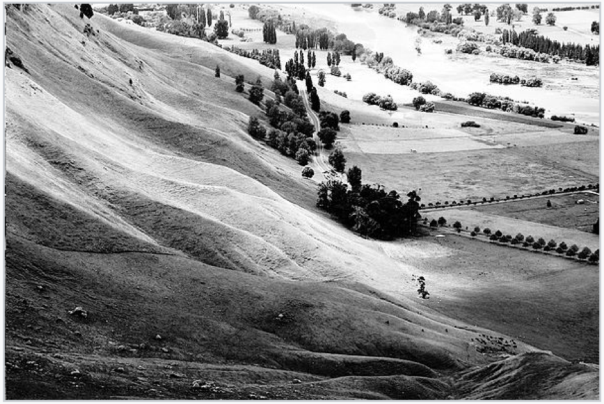
http://homepages.inf.ed.ac.uk/rbf/HIPR2/histeq.htm

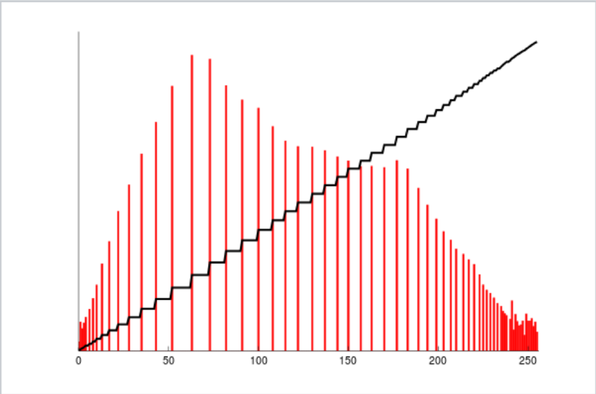
Histogram Equalization


Before Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)


After Histogram Equalization



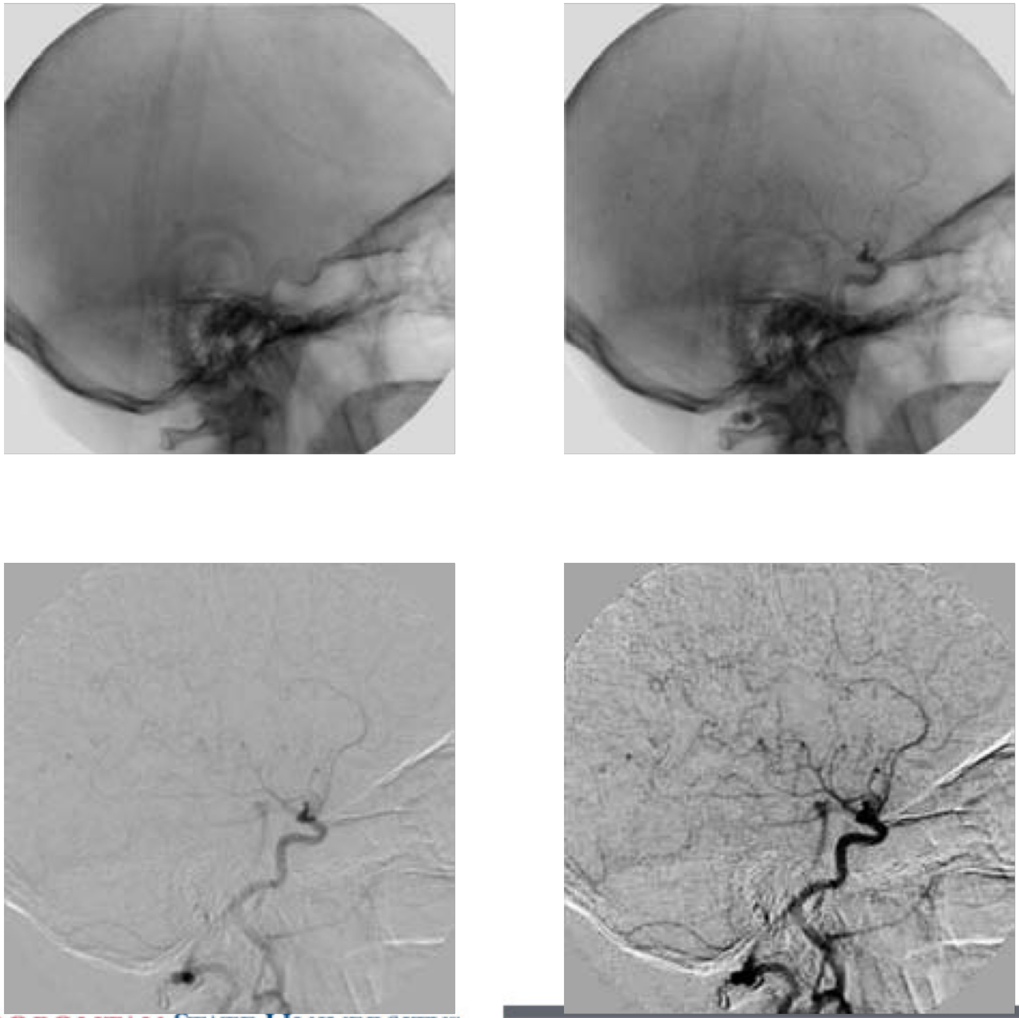
Corresponding histogram (red) and cumulative histogram (black)

Histogram Equalization

Original Image

After Histogram Equalization

Image subtraction



(a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht,The Netherlands.)

Image Smoothing or Averaging

- A noisy image:

$$g(x, y) = f(x, y) + n(x, y)$$

Let $g(x, y)$ denote a corrupted image formed by the addition of noise, $\eta(x, y)$, to a noiseless image $f(x, y)$; that is,

$$g(x, y) = f(x, y) + \eta(x, y) \tag{2.6-4}$$

EXAMPLE 2.5: Addition (averaging) of noisy images for noise reduction.

Image Smoothing or Averaging

- A noisy image:

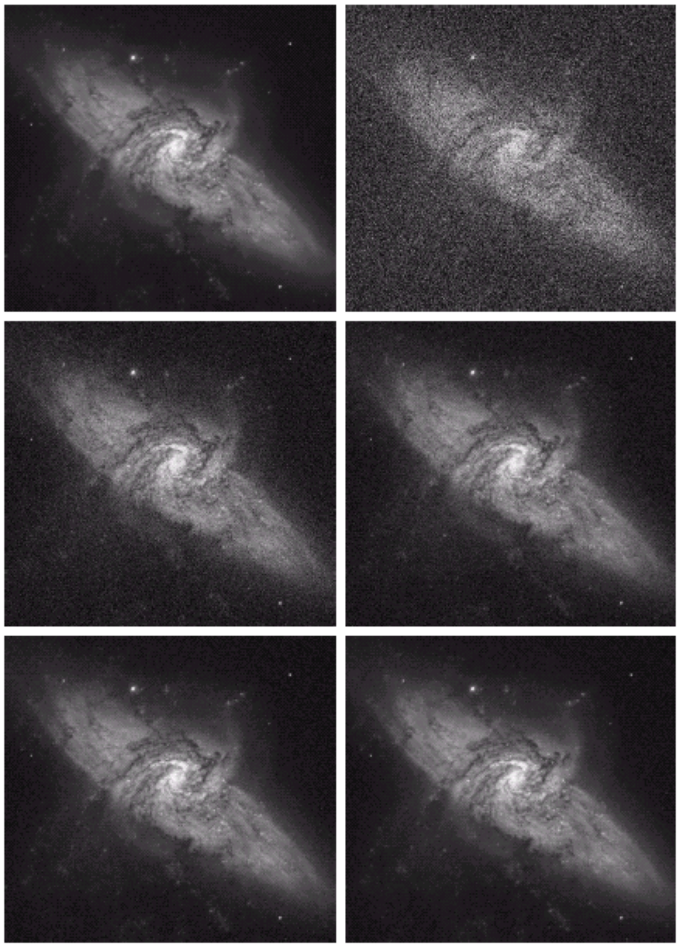
$$g(x,y) = f(x,y) + n(x,y)$$

- Averaging M different noisy images:

$$\overline{g}(x,y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x,y)$$

- As M increases, the variability of the pixel values at each location decreases.
  - This means that g(x,y) approaches f(x,y) as the number of noisy images used in the averaging process increases.

Example



a	b
c	d
e	f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)

Spatial Filtering

- Use of spatial masks for image processing (spatial filters)
- Linear and nonlinear filters
- Low-pass filters eliminate or attenuate high frequency components in the frequency domain (sharp image details), and result in image blurring.

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

$a=(m-1)/2$ and $b=(n-1)/2$,
m x n (odd numbers)

For x=0,1,…,M-1 and y=0,1,…,N-1

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:

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 \qquad \text{(for a 3 x 3 filter)}$$

Neighborhood Averaging

Each point in the smoothed image, $\hat{F}(x, y)$ is obtained from the average pixel value in a neighbourhood of (x, y) in the input image.

For example, if we use a 3×3 neighbourhood around each pixel we would use the mask

$$\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$

General Spatial Filter

FIGURE 3.33
Another representation of a general 3×3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

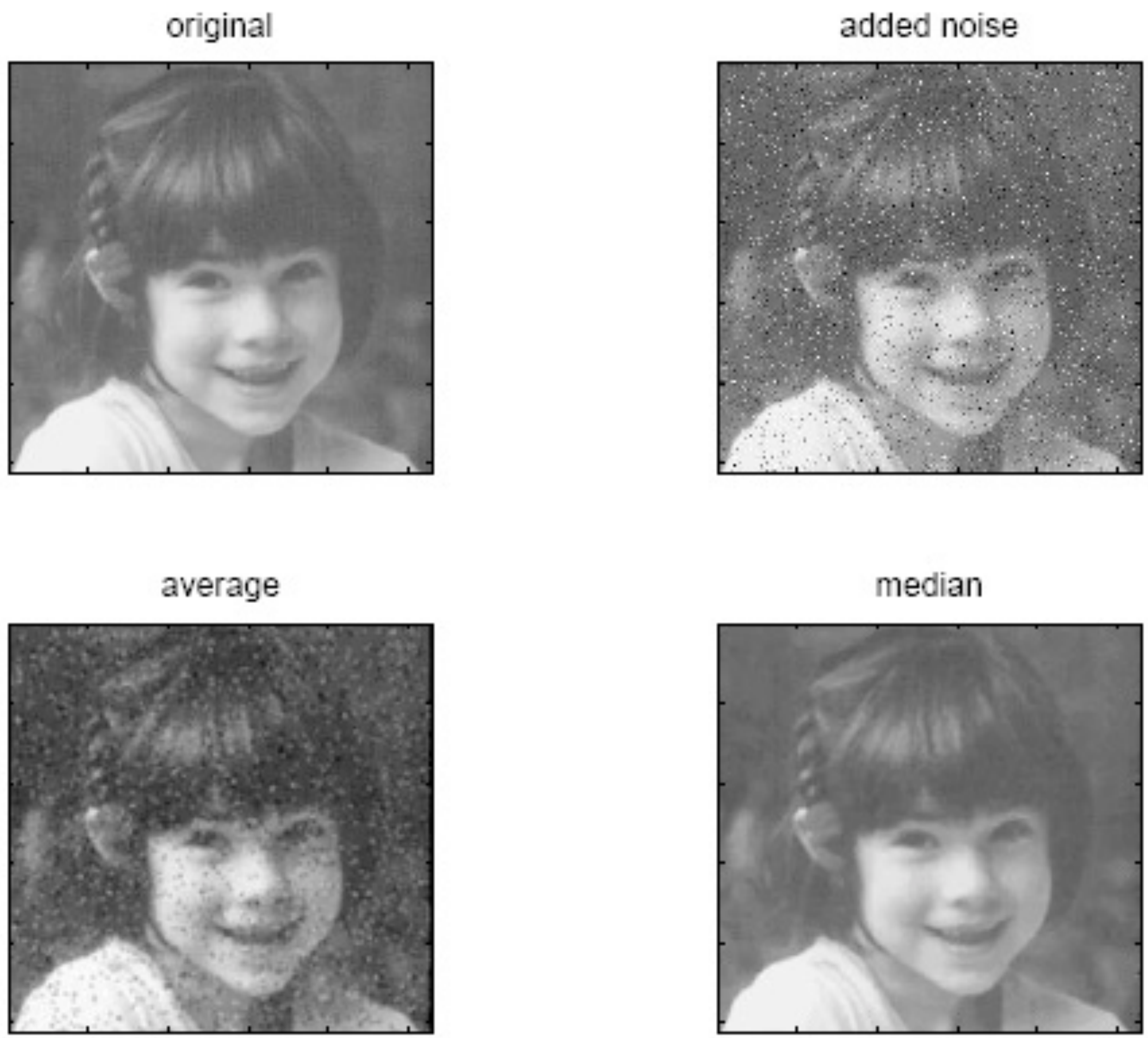
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

FIGURE 3.34 Two 3×3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Non-linear Filter

- Median filtering (nonlinear)
  - Used primarily for noise reduction (eliminates isolated spikes)
  - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).


original
added noise
average
median

Sharpening Filters

- The main aim in image sharpening is to highlight fine detail in the image
- With image sharpening, we want to enhance the high-frequency components; this implies a spatial lter shape that has a high positive component at the centre

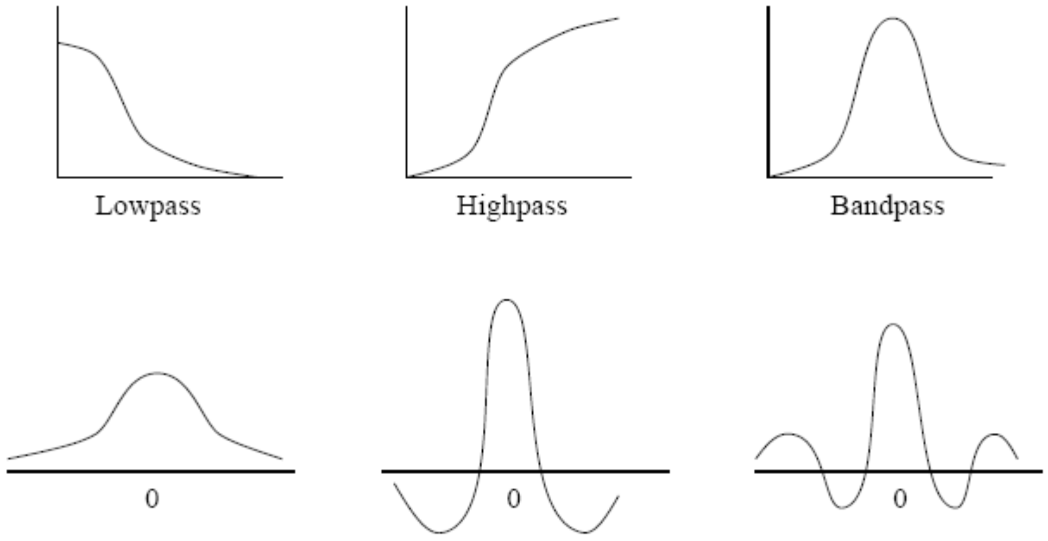


Figure 4: Frequency domain filters (top) and their corresponding spatial domain counterparts (bottom).

Derivatives

- First derivative

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

- Second derivative

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

Observations

- 1st order derivatives produce thicker edges in an image
- 2nd order derivatives have stronger response to fine detail
- 1st order derivatives have stronger response to a gray lever step
- 2nd order derivatives produce a double response at step changes in gray level

A simple spatial filter that achieves image sharpening is given by

$$\begin{array}{ccc} -1/9 & -1/9 & -1/9 \\ -1/9 & 8/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{array}$$

- Since the sum of all the weights is zero, the resulting signal will have a zero DC value

Frequency Domain Methods

- We simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter (rather than convolve in the spatial domain), and take the inverse transform to produce the enhanced image.
- Low pass filtering involves the elimination of the high frequency components in the image. It results in blurring of the image

Frequency Domain Methods

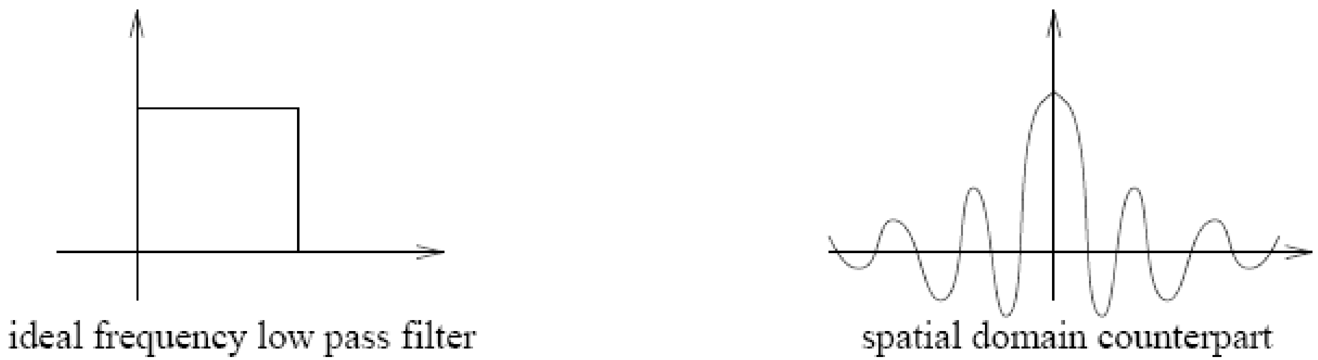


Figure 5: Transfer function for an ideal low pass filter.