ECE 470: Introduction to Robotics
Lecture 22

Liangjing Yang
Assistant Professor, ZJU-UIUC Institute
liangjingyang@intl.zju.edu.cn

Wechat ID: Liangjing_Yang

Our Learning Roadmap

- **Schedule Check on our Learning Roadmap**

O. Overview
- Science & Engineering in Robotics

I. Spatial Representation & Transformation
- Coordinate Systems; Pose Representations; Homogeneous Transformations

II. Kinematics
- Multi-body frame assignment; D-H Convention; Joint-space; Work-space; Forward/Inverse Kinematics

III. Velocity Kinematics and Static Forces
- Translational/Rotational Velocity; Joint torque; Generalized Force Coordinates; Jacobian; Singularity

Fundamentals
Week 1-4
Revision/ Quiz on Week 5

IV. Dynamics
- Acceleration of Body; Newton-Euler Equations of Motion; Lagrangian Formulation

V. Control
- Closed-Loop Control and Feedback, Control of 2nd order system, Independent Joint Control, Force Control

VI. Planning
- Joint-Based Scheme; Cartesian-Based Scheme; Collision Free Path Planning

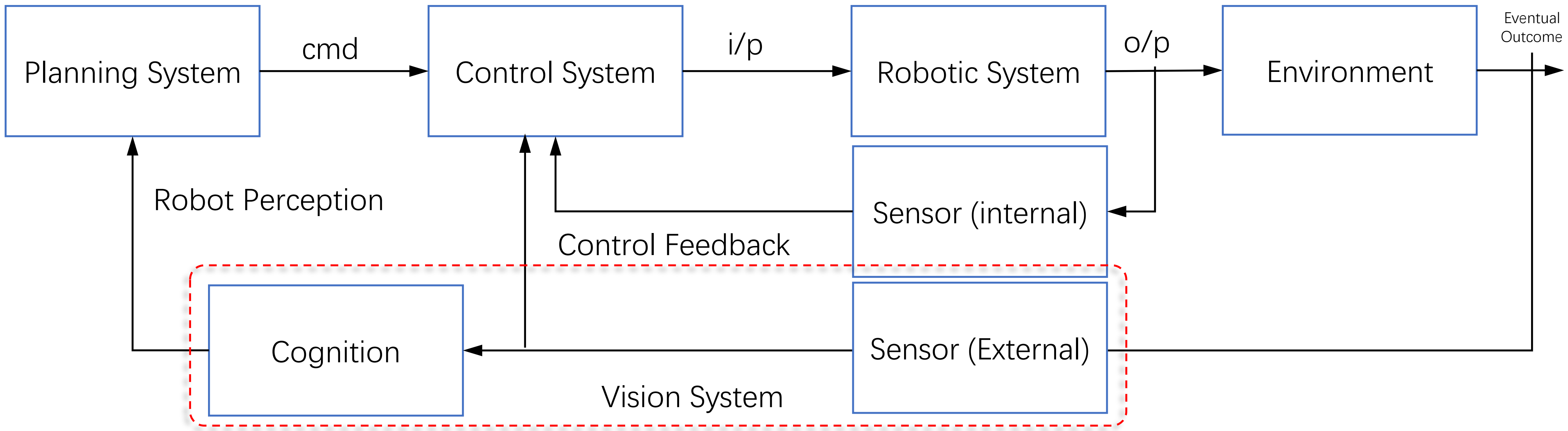
Essentials
Week 6-9
Revision/ Quiz on Week 10

VII. Robot Vision (Perception)
- Image Formation; Image Processing; Visual Tracking & Pose Estimation; Vision-based Control & Image-guided robotics

Applied
Week 11-14
Reading Wk/ Exam on Week 15-16

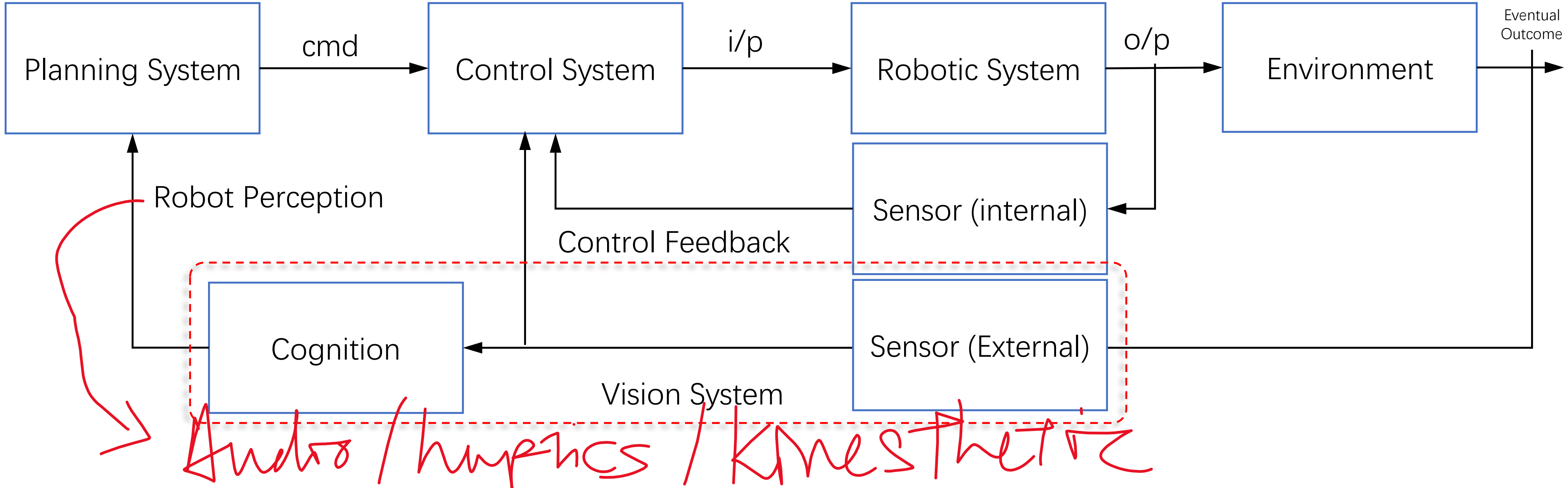
Robot Vision: Closing the final loop

- Model **kinematics** and **dynamics** of the robotic system
- Design **control** for appropriate input to achieve desired outcome
- **Planning system** to send the command to **control** system
- **Perceive** and interact with environment to achieve goal



Robot Vision: Closing the final loop

- Model **kinematics** and **dynamics** of the robotic system
- Design **control** for appropriate input to achieve desired outcome
- **Planning system** to send the command to **control** system
- **Perceive** and interact with environment to achieve goal



Overview of Robot Vision

0. Introduction to Robot Vision
   - What is Robot Vision? Imparting robots the capability of making sense of the scene
I. Image Formation
   - The science behind computer/machine vision
II. Image Processing
   - Common techniques to manipulate, enhance & analyse images
III. Robot Vision Applications
   - 3D Vision; Photogrammetry; Vision-based techniques in robotics- visual servo, pose estimation, localization, mapping, navigation

obtaining info on physical objects or environment using photography

Image Formation

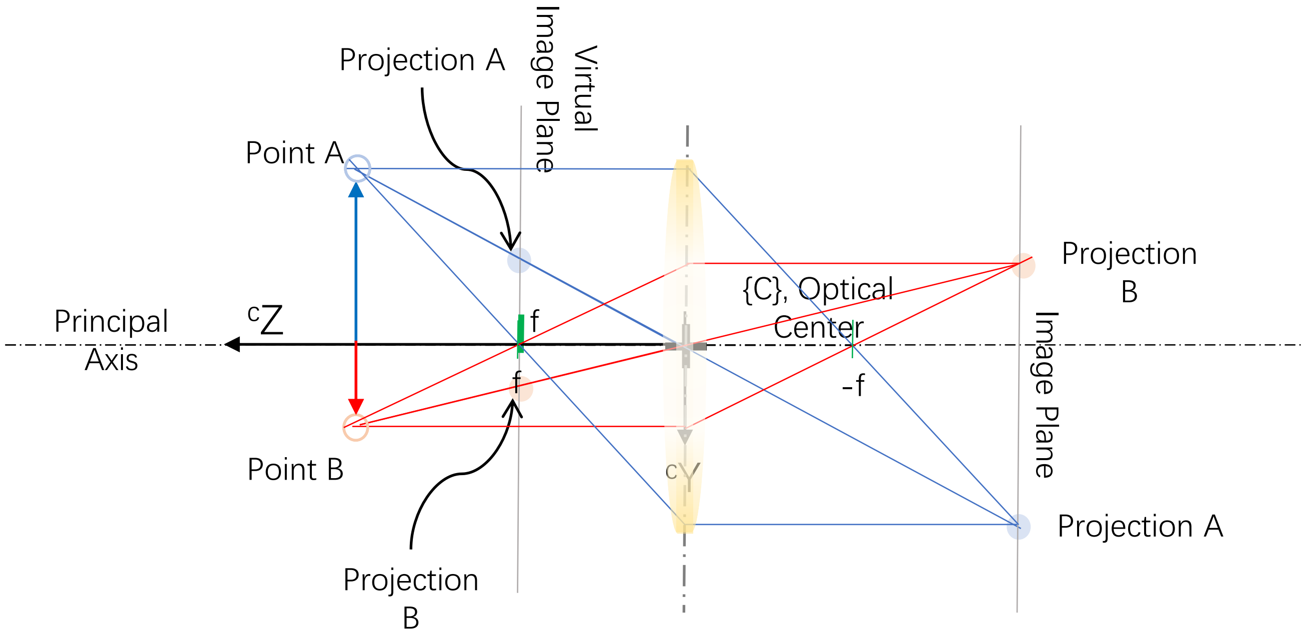
ECE 470 Introduction to Robotics

Image Formation

- Camera Model
  - How is image formed?
- Imaging System
  - How is it acquired?
- Digital Image Representation
  - How is it represented in computer?

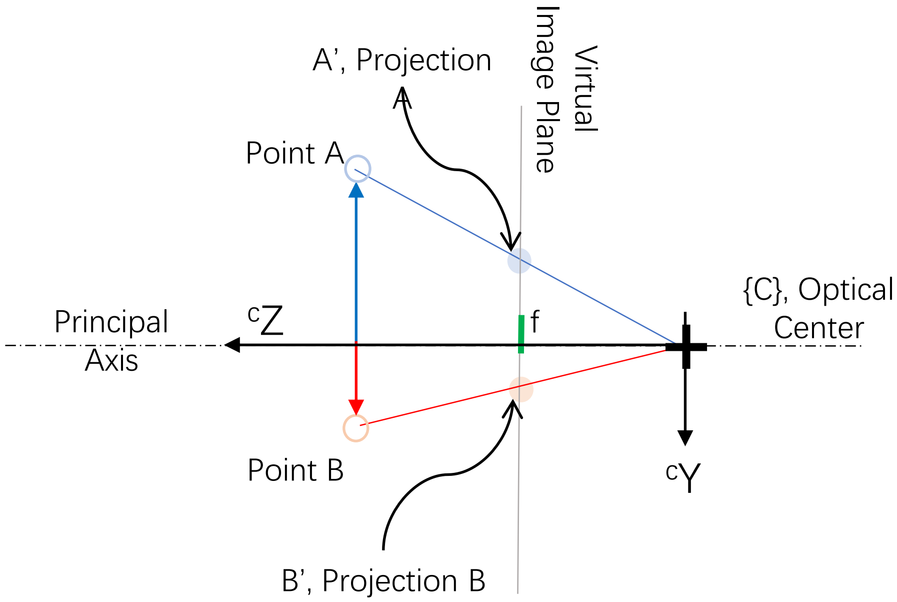
Camera Model

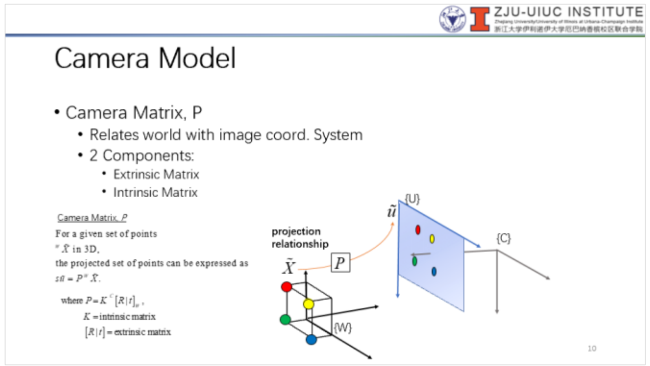
- Central Projection Camera Model
  - A simplified model for camera geometry



Camera Model

- Central Projection Camera Model
  - A simplified model for camera geometry





Computer Vision System Components

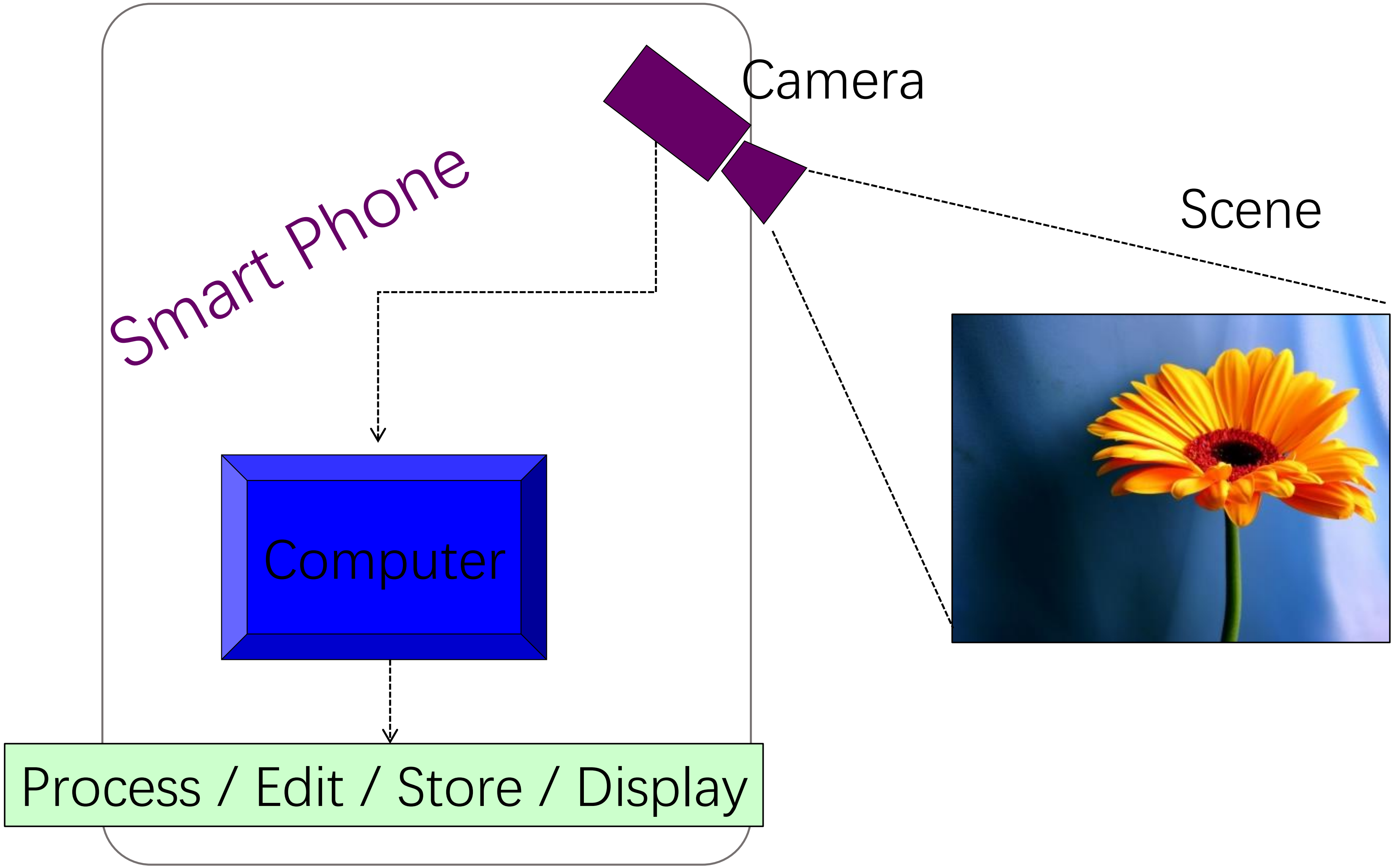
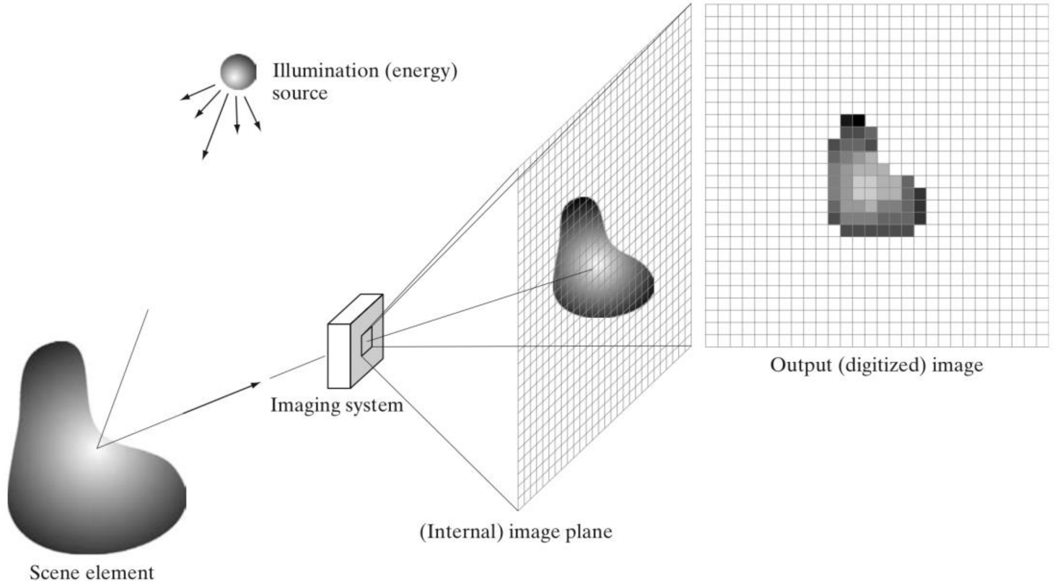
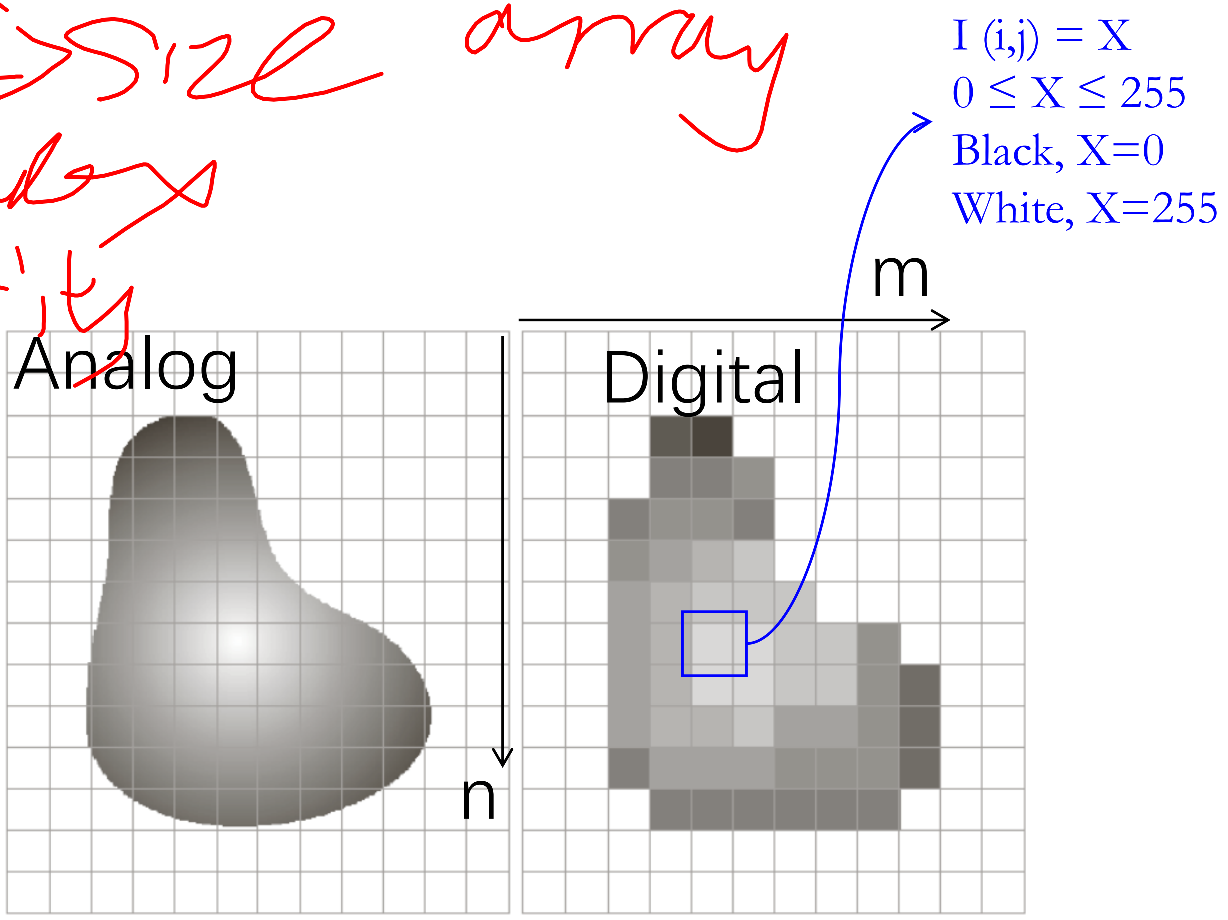


Image Representation

pixel (picture elem)
→ elements in array

- Digital Image
- N X M Array of Pixels ( ) → size
  Matrix
  - Pixel Position ( ) → index
  - Pixel Values ( ) → intensity

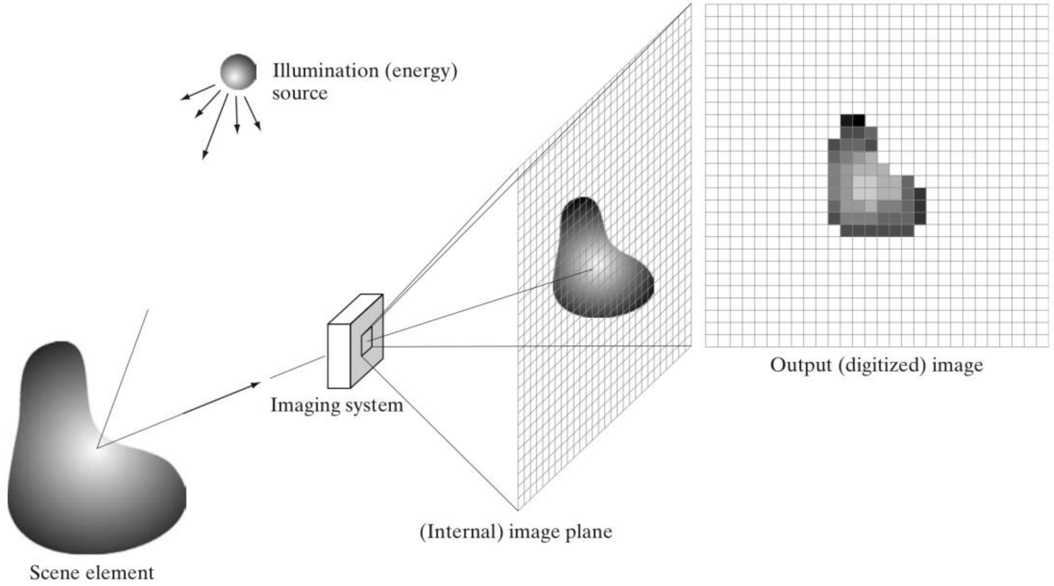


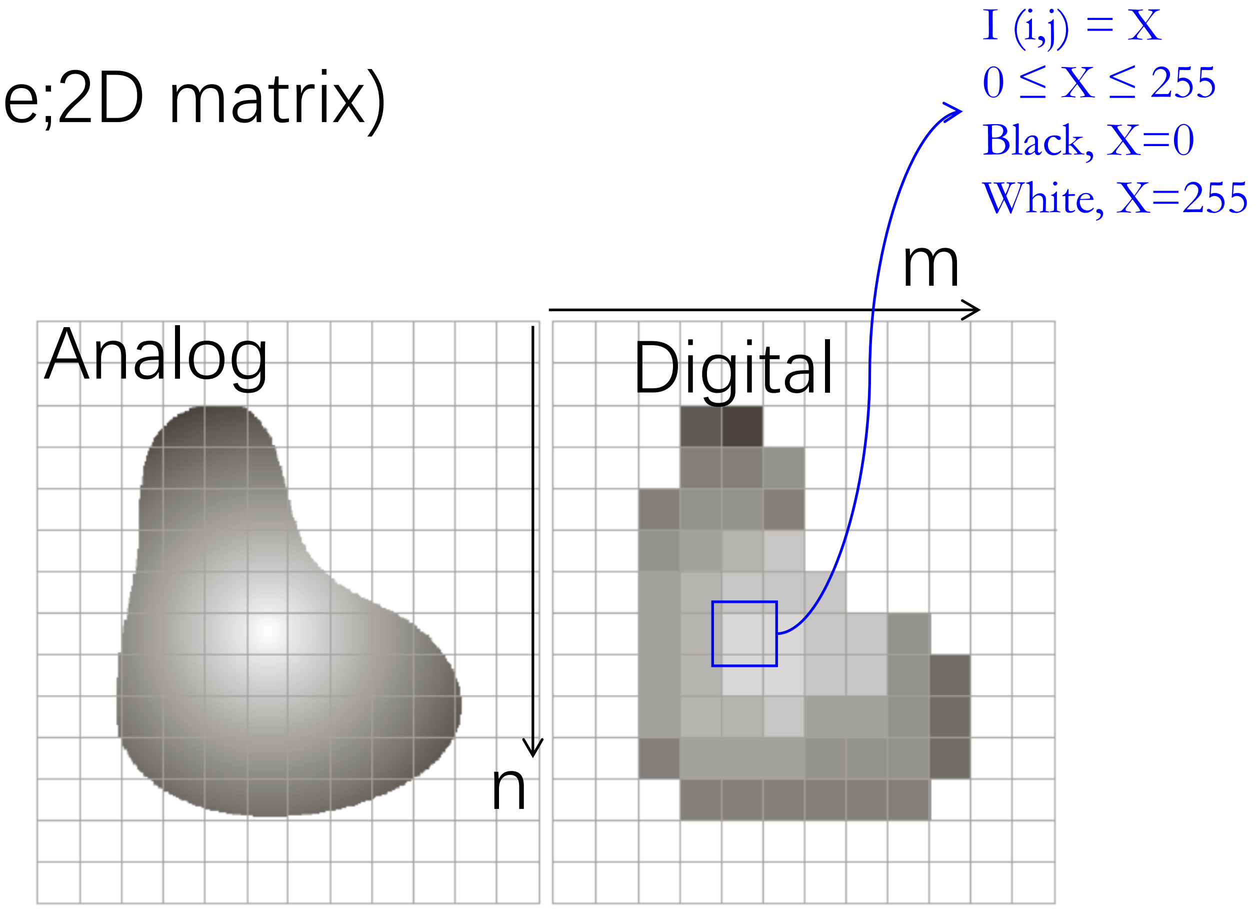


Ref: António R. C. Paiva, 2010

Image Representation

- Digital Image
- *N* X *M* Array of Pixels (Image size;2D matrix)
  - Pixel Position (Coordinates)
  - Pixel Values (Intensity)

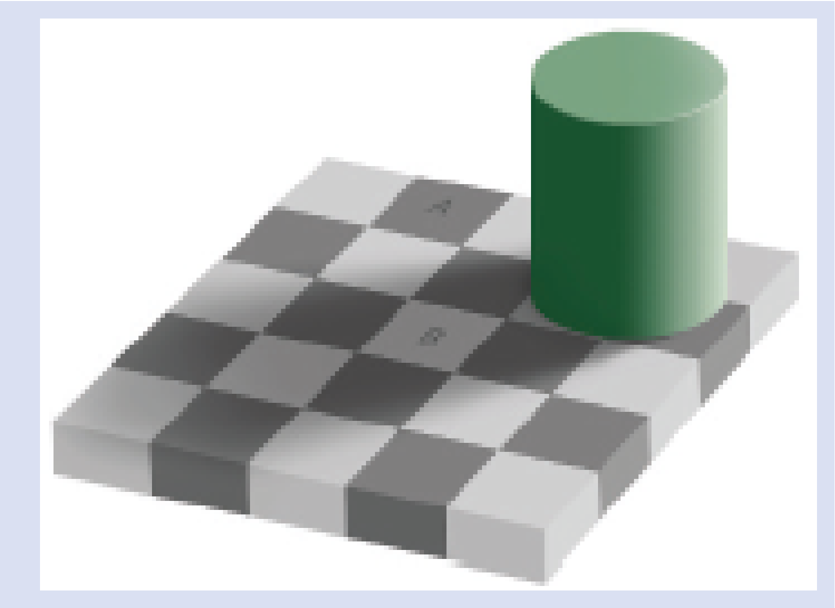




Ref: António R. C. Paiva, 2010

Image Representation

Which region of pixels have higher intensity values?
A or B?



Computer Interpretation of Images

x

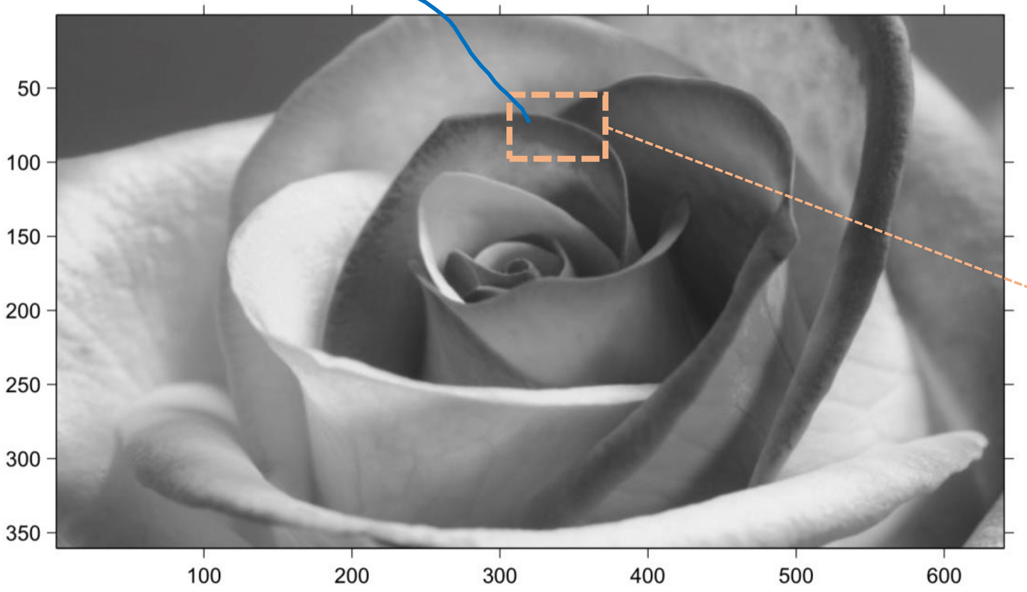
y

Region of Interest

ROI

Gray scale Images

Image



0 = black
gray
255 = white

Intensity , I(*i*,*j*)

i \ *j*	345	346	347	348	349	350	351	352	353	354	355
70	67	64	61	61	60	59	60	61	60	59	59
71	74	69	65	63	63	61	60	60	61	60	59
72	86	79	72	70	68	66	64	63	63	62	61
73	96	97	94	87	81	77	72	67	66	64	62
74	75	83	91	97	98	95	88	81	69	65	63
75	72	74	75	78	84	93	101	104	92	84	75
76	68	68	68	69	73	81	88	92	101	101	96
77	67	67	67	69	71	72	73	74	81	94	102
78	69	67	65	66	67	68	71	74	73	80	85
79	67	67	68	69	70	68	67	68	74	74	71
80	70	69	68	68	67	66	66	69	70	70	73

(70 80, 345, 355)

Matlab Commands:

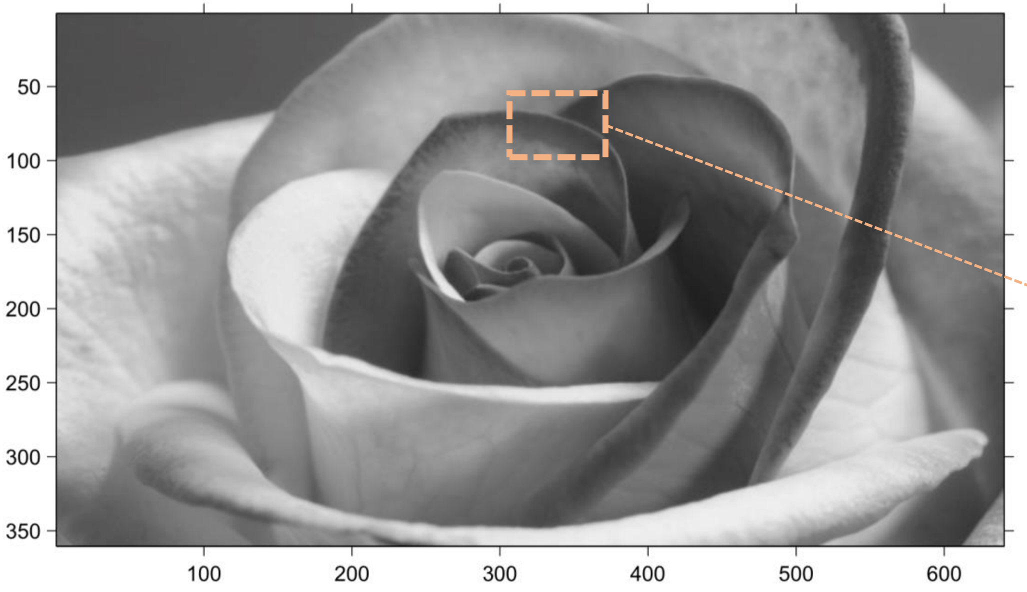
```
img_gray=rgb2gray(img1);
imshow(img_gray)
imsub_gray=img_gray(          ,:)
```

– for printing intensity matrix

Computer Interpretation of Images

Gray scale Images

Image



0 = black
gray
255 = white

Intensity , I(*i*,*j*)

j →

i ↓

	345	346	347	348	349	350	351	352	353	354	355
70	67	64	61	61	60	59	60	61	60	59	59
71	74	69	65	63	63	61	60	60	61	60	59
72	86	79	72	70	68	66	64	63	63	62	61
73	96	97	94	87	81	77	72	67	66	64	62
74	75	83	91	97	98	95	88	81	69	65	63
75	72	74	75	78	84	93	101	104	92	84	75
76	68	68	68	69	73	81	88	92	101	101	96
77	67	67	67	69	71	72	73	74	81	94	102
78	69	67	65	66	67	68	71	74	73	80	85
79	67	67	68	69	70	68	67	68	74	74	71
80	70	69	68	68	67	66	66	69	70	70	73

Matlab Commands:

```
img_gray=rgb2gray(img1);
imshow(img_gray)
imsub_gray=img_gray(70:80,345:355,:)
```
– for printing intensity matrix

Example 1: Spatial & Intensity Information

- A gray scale image $\mathbf{I}$ of our campus has a dimension of 945x1680 shown. A region of interest (ROI) from row 391 to 420 and column 1071 to 1100 is specified to encompass a window in the scene with an array of intensity values. This ROI is denoted as I_ROI= $\mathbf{I}$ ($\boxed{}$).

Example 1: Spatial & Intensity Information

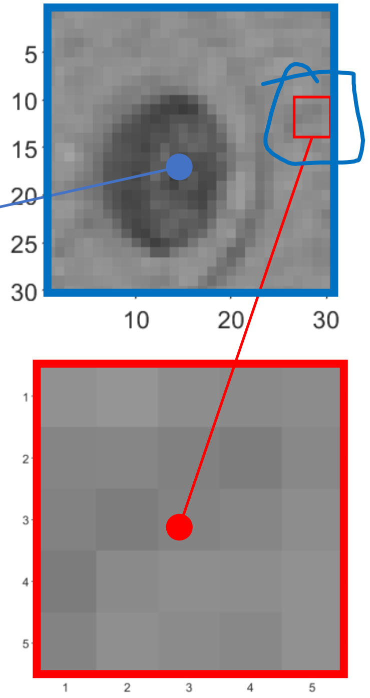
- A gray scale image $\mathbf{I}$ of our campus has a dimension of 945x1680 shown. A region of interest (ROI) from row 391 to 420 and column 1071 to 1100 is specified to encompass a window in the scene with an array of intensity values. This ROI is denoted as I_ROI= $\mathbf{I}$ (391:420, 1071:1100).

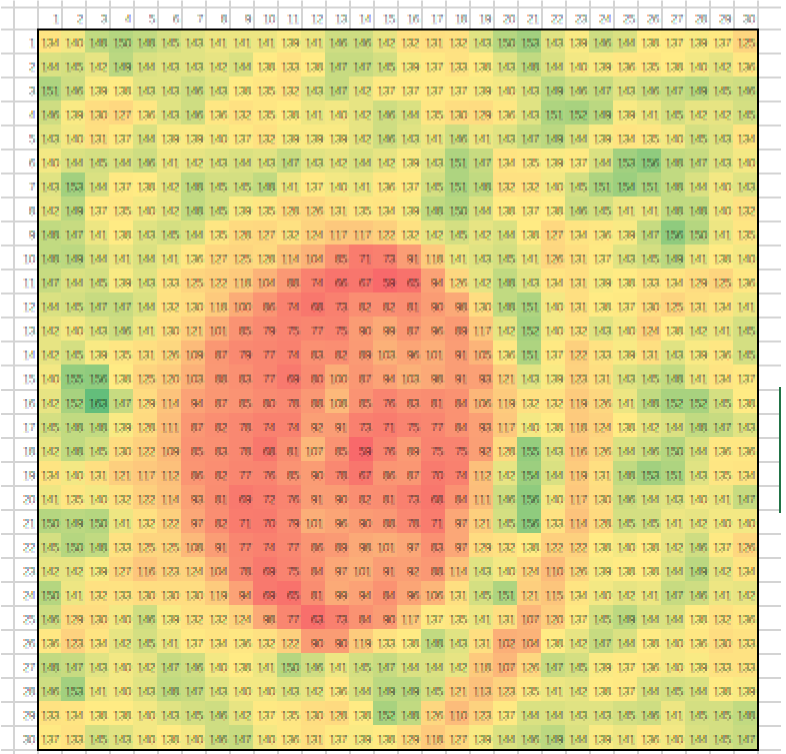
Example 1: Spatial & Intensity Information

- A gray scale image **I** of our campus has a dimension of <u>945x1680</u> shown. A region of interest (ROI) from <u>row 391 to 420</u> and <u>column 1071 to 1100</u> is specified to encompass a window in the scene with an array of intensity values. This ROI is denoted as I_ROI= **I** (391:420, 1071:1100).

column

row





Computer Interpretation of Images

Let us see how computer handles this image

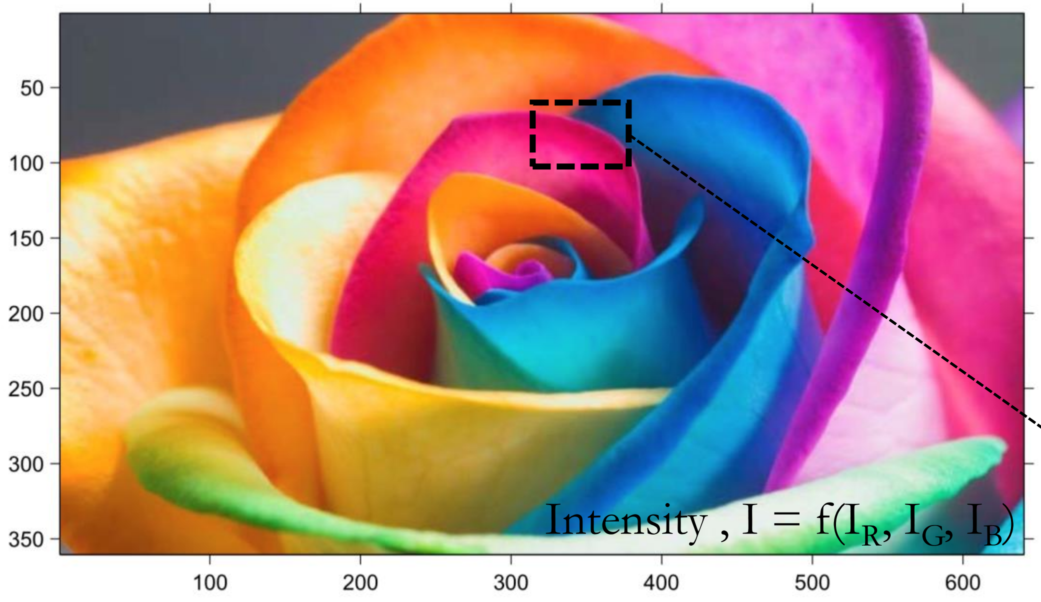
Matlab Commands:

```
img1=imread('filename.ext', 'ext');
imshow(img1)
```

Computer Interpretation of Images

NXMX3

Color Images (RGB)



$I_R(i,j)$

$j \rightarrow$

$i \downarrow$

	345	346	347	348	349	350	351	352	353	354	355
70	57	50	43	40	36	34	33	32	26	24	25
71	87	70	58	49	45	39	36	31	25	22	25
72	134	114	94	81	69	59	51	42	31	26	29
73	191	181	164	140	117	98	80	61	45	35	34
74	197	198	193	185	171	152	133	109	77	62	56
75	201	200	197	194	193	195	198	185	151	125	103
76	197	197	197	197	201	207	213	208	199	181	159
77	194	194	193	195	199	201	205	204	207	210	202
78	199	194	190	191	194	197	203	209	208	213	207
79	197	196	198	199	199	196	193	194	203	204	201
80	205	202	199	199	195	191	189	194	199	202	208

$I_G(i,j)$

	345	346	347	348	349	350	351	352	353	354	355
70	64	63	61	62	62	62	63	65	66	66	65
71	60	60	60	62	63	63	62	65	69	69	65
72	55	54	53	56	59	60	61	64	69	70	67
73	42	48	52	53	55	58	59	61	67	69	66
74	8	20	34	46	54	58	57	58	56	57	57
75	3	6	9	14	23	35	45	55	54	55	52
76	0	0	0	0	4	12	19	27	44	53	57
77	0	0	0	2	3	3	1	3	11	29	45
78	1	0	0	0	0	0	0	2	0	7	18
79	0	0	0	1	2	1	0	1	5	4	1
80	0	0	0	0	0	0	1	3	2	0	1

	345	346	347	348	349	350	351	352	353	354	355
70	110	108	107	109	111	112	115	116	115	114	116
71	113	110	108	107	111	111	113	113	114	113	114
72	121	116	111	111	112	114	114	114	114	115	114
73	126	127	126	122	120	120	118	115	116	115	113
74	98	108	118	127	132	132	129	125	115	113	111
75	90	92	96	103	112	125	136	141	133	127	118
76	79	81	83	87	96	106	116	124	136	136	133
77	76	77	79	83	89	91	96	99	109	124	134
78	78	75	73	76	78	83	89	93	94	105	110
79	73	71	73	78	81	80	81	84	92	93	90
80	79	74	73	74	75	74	77	81	84	84	89

$I_B(i,j)$

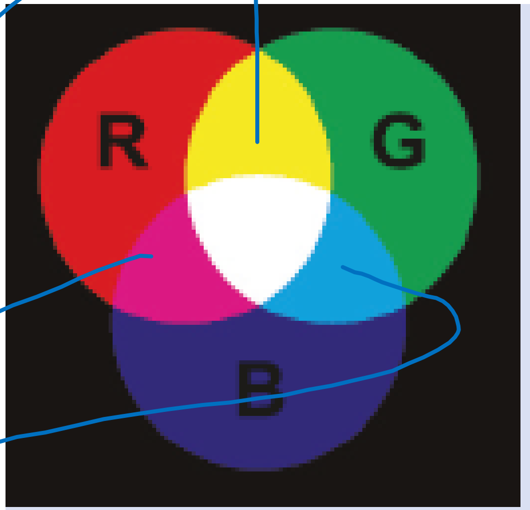
Matlab Commands:

```
imshow(img1)
red=img1(:,:,1)
Imsub_red=red(70:80,345:355,:) – for printing intensity matrix of red
```

Computer Interpretation of Images

primary colors

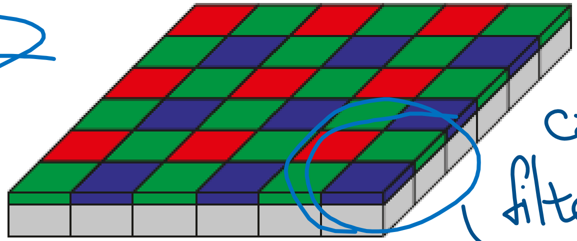
Color Images (RGB)



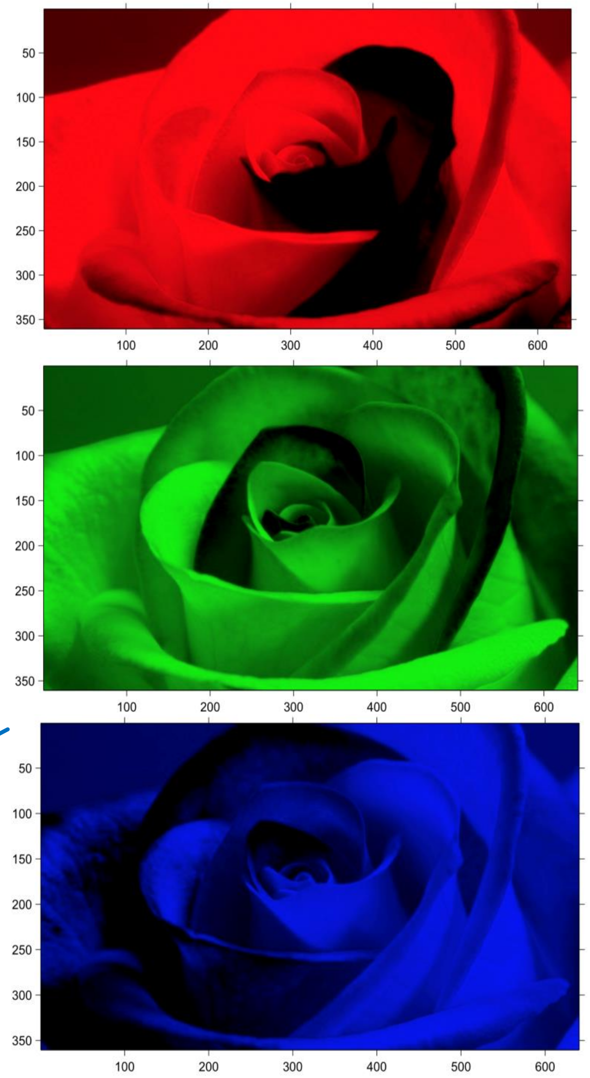
secondary colors

array of sensor

light



color filters



Computer Interpretation of Images

red layer

Color Images (RGB)

channel

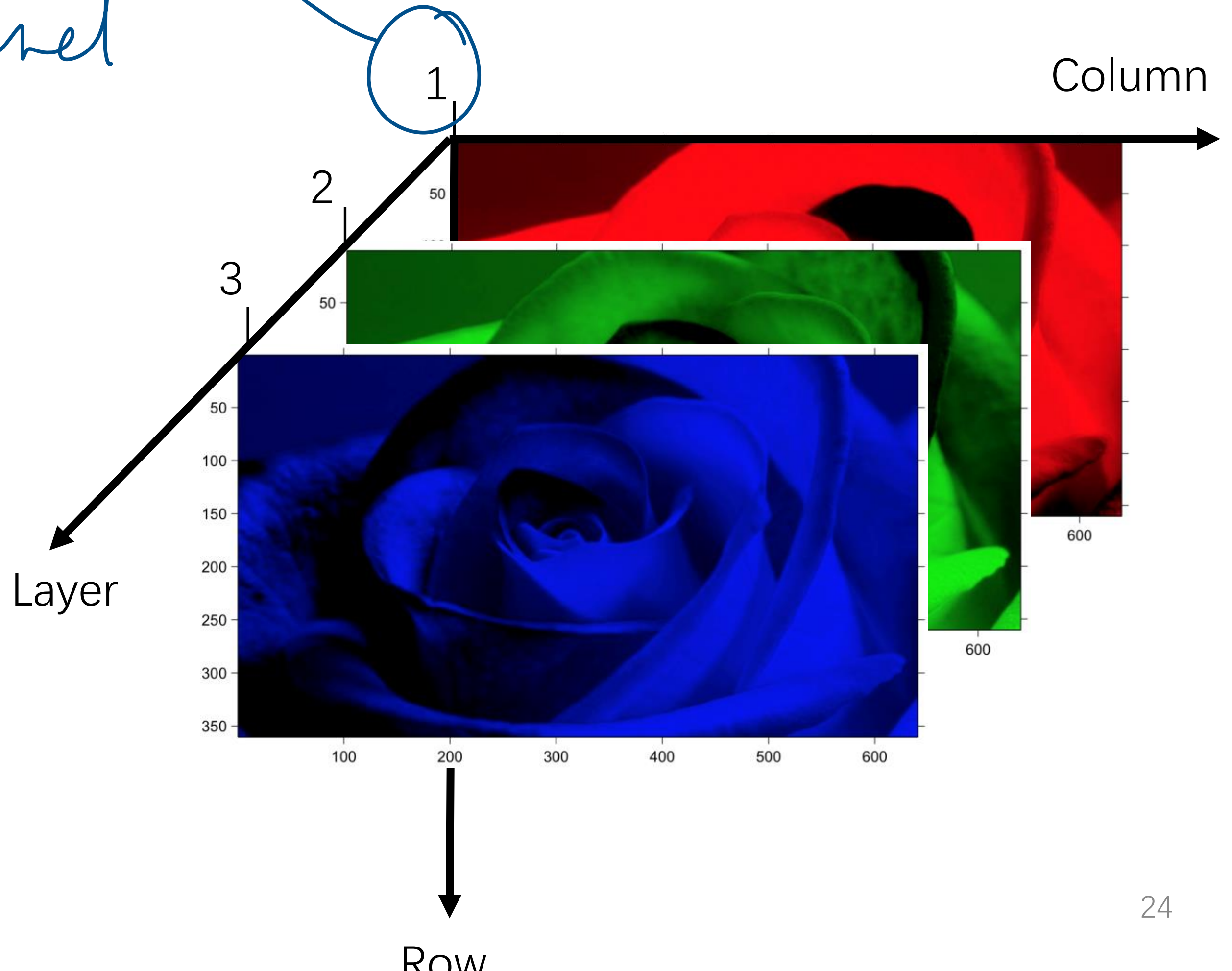
Matlab Commands:

```
imshow(img1)
red=img1(:,:,1)
green=img1(:,:,2);
blue=img1(:,:,3);
a=zeros(size(img1,1),size(img1,2));
only_red=cat(3,red,a,a)
imshow(only_red)
only_green=cat(3,a,green,a)
imshow(only_green)
only_blue=cat(3,a,a,blue)
imshow(only_blue)
```

red layer

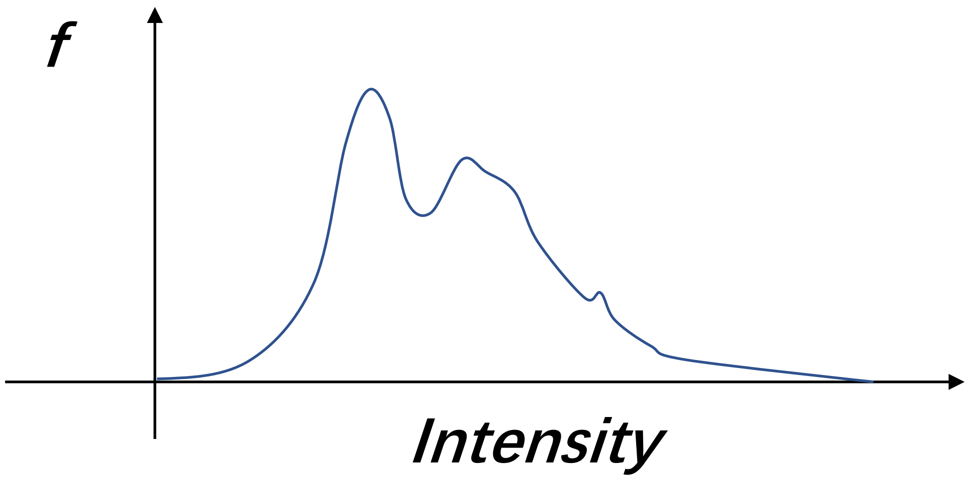
green layer

blue layer



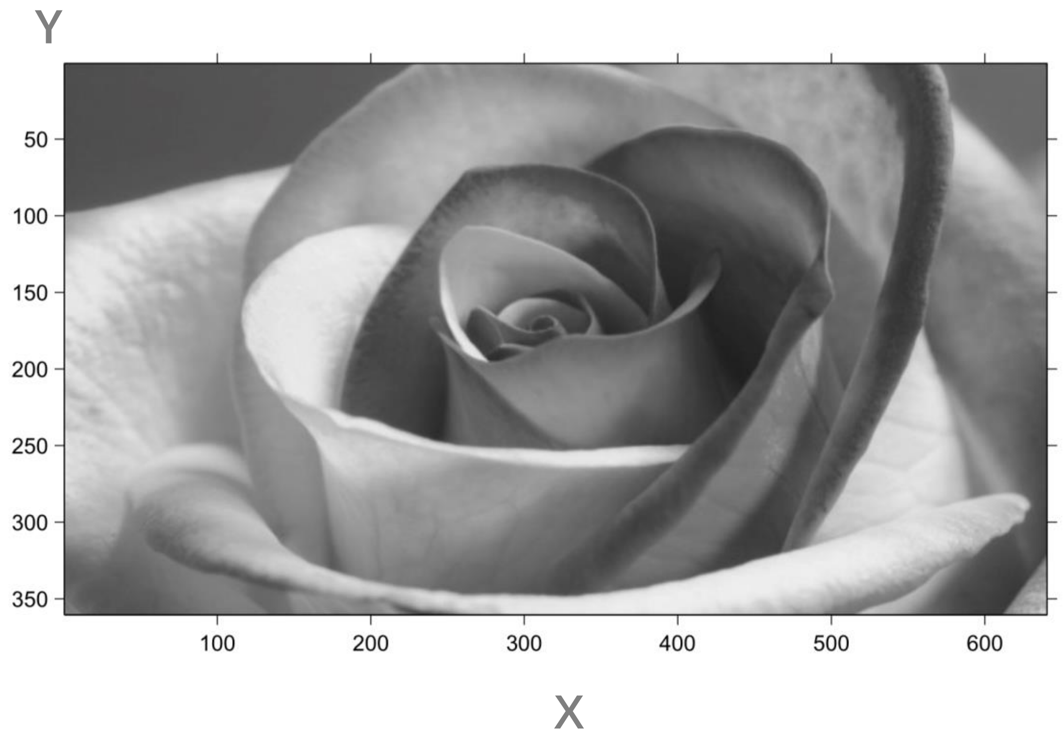
Histogram

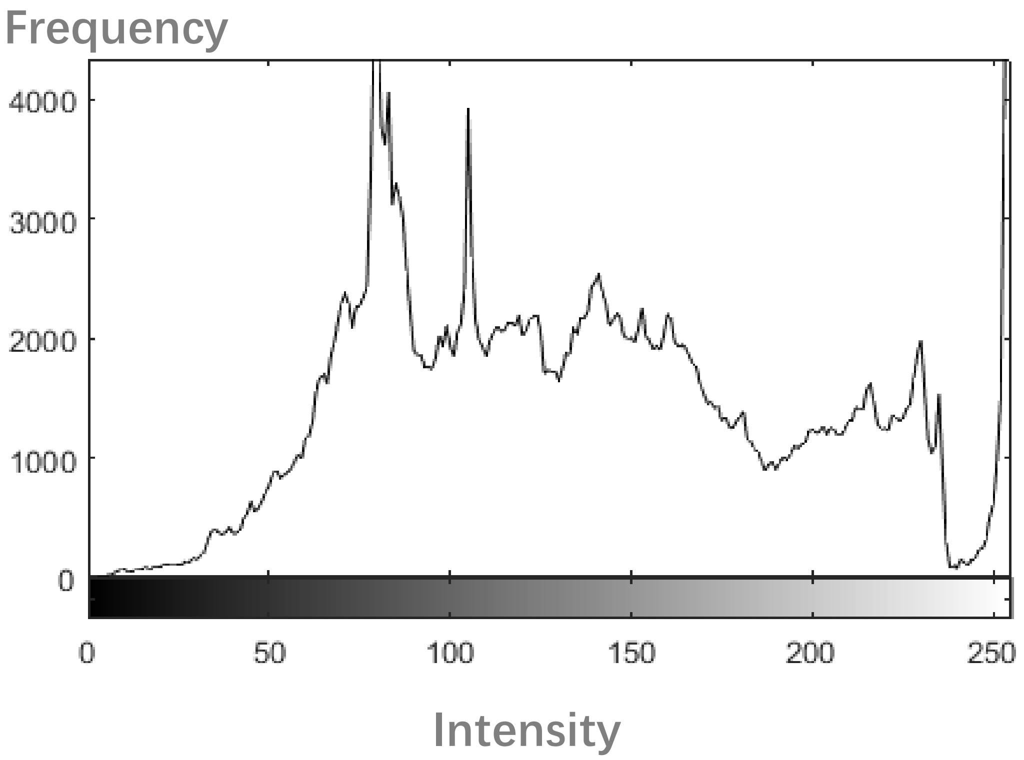
- Distribution of the intensity levels



Histogram

- Distribution of the intensity levels

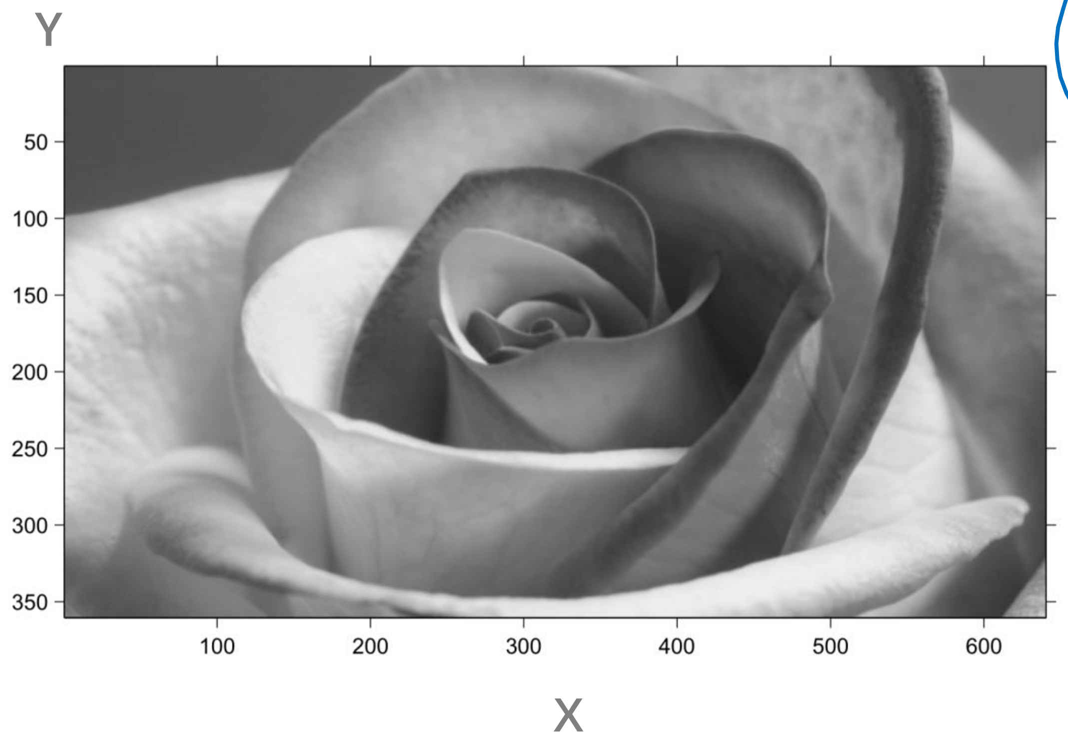


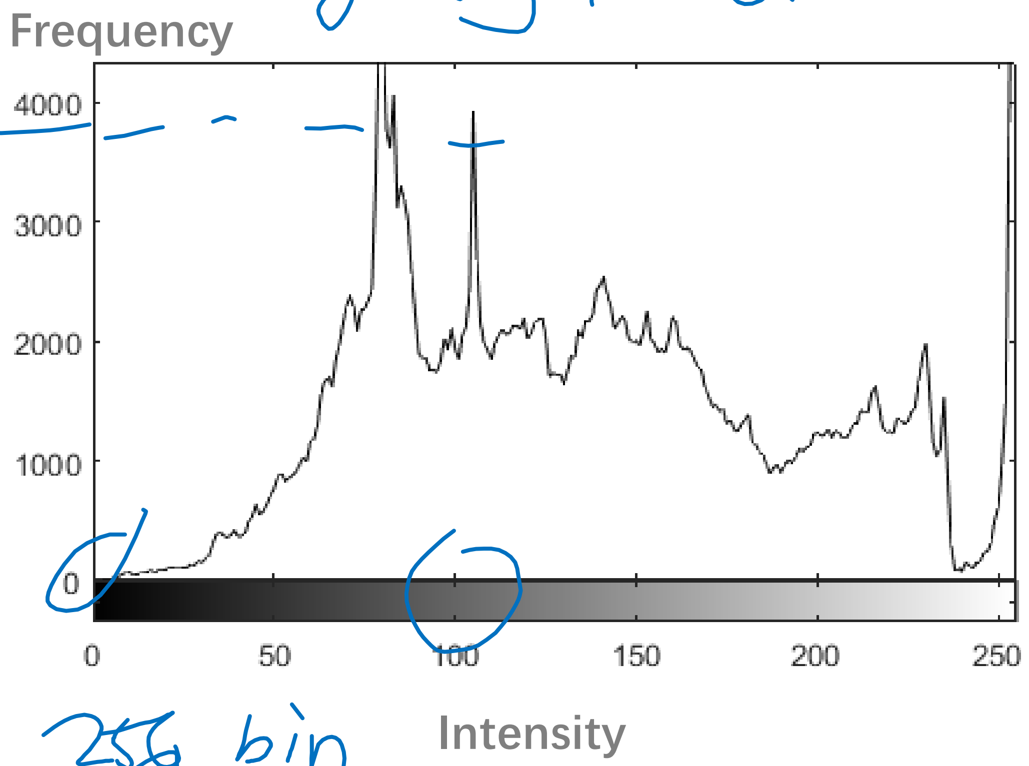


Histogram

- Distribution of the intensity levels

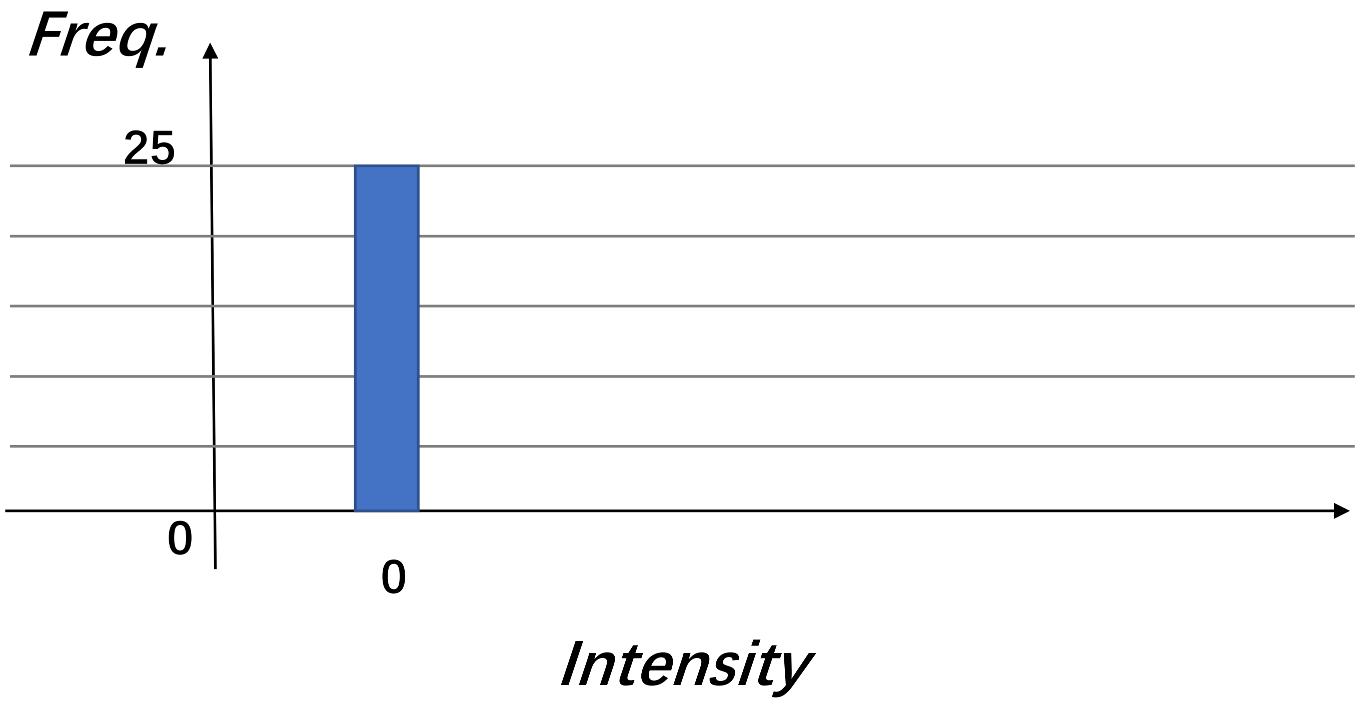
# of pixel with gray level 100

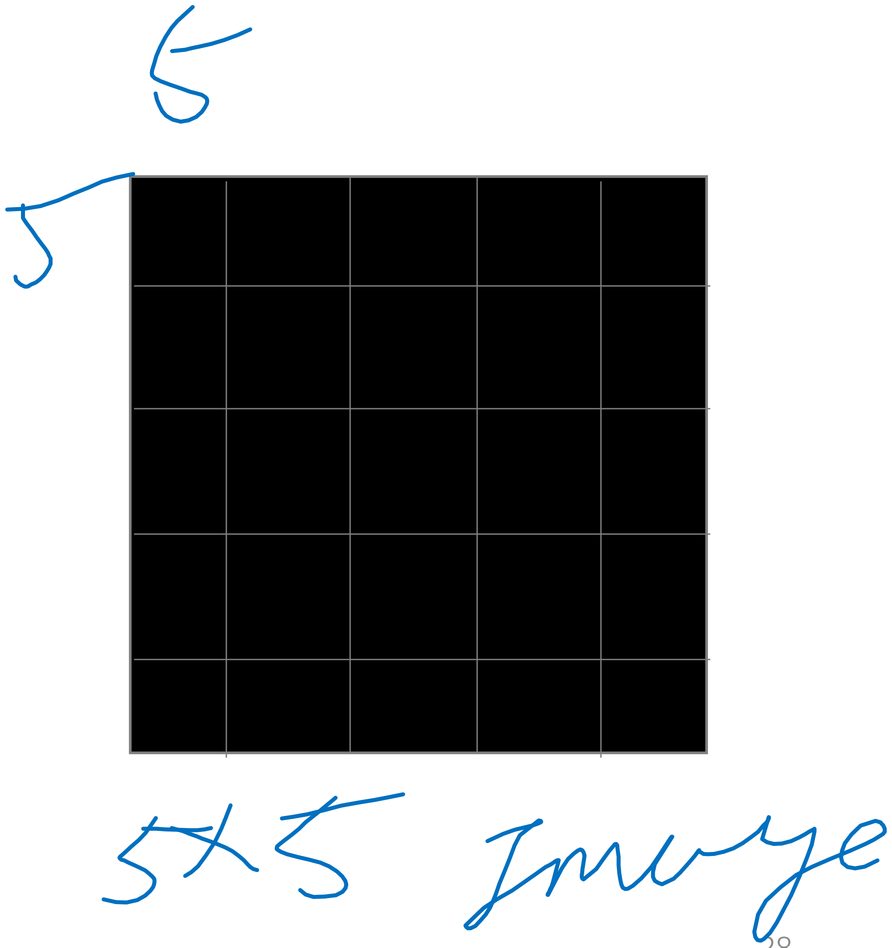




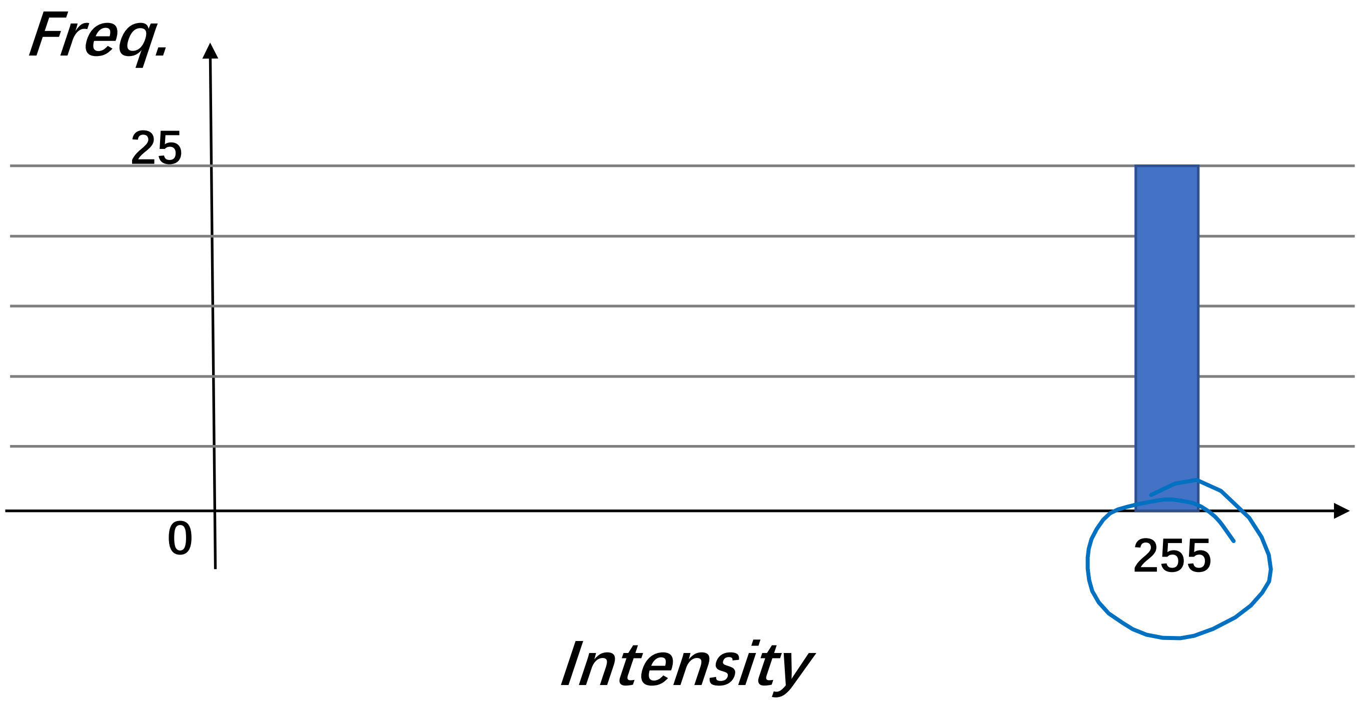
256 bin

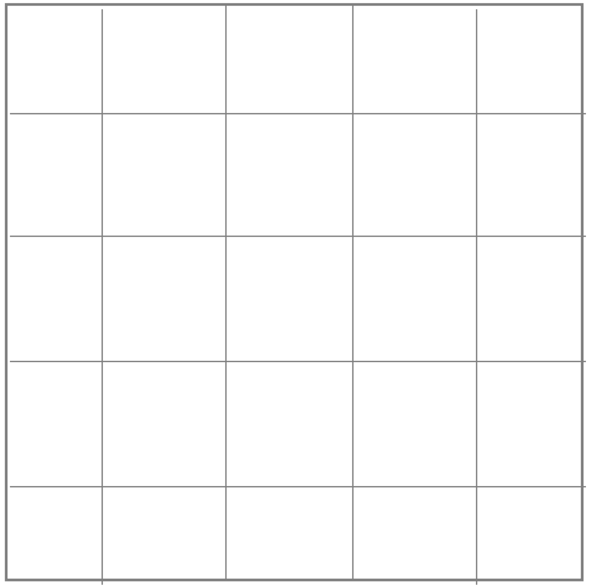
Histogram



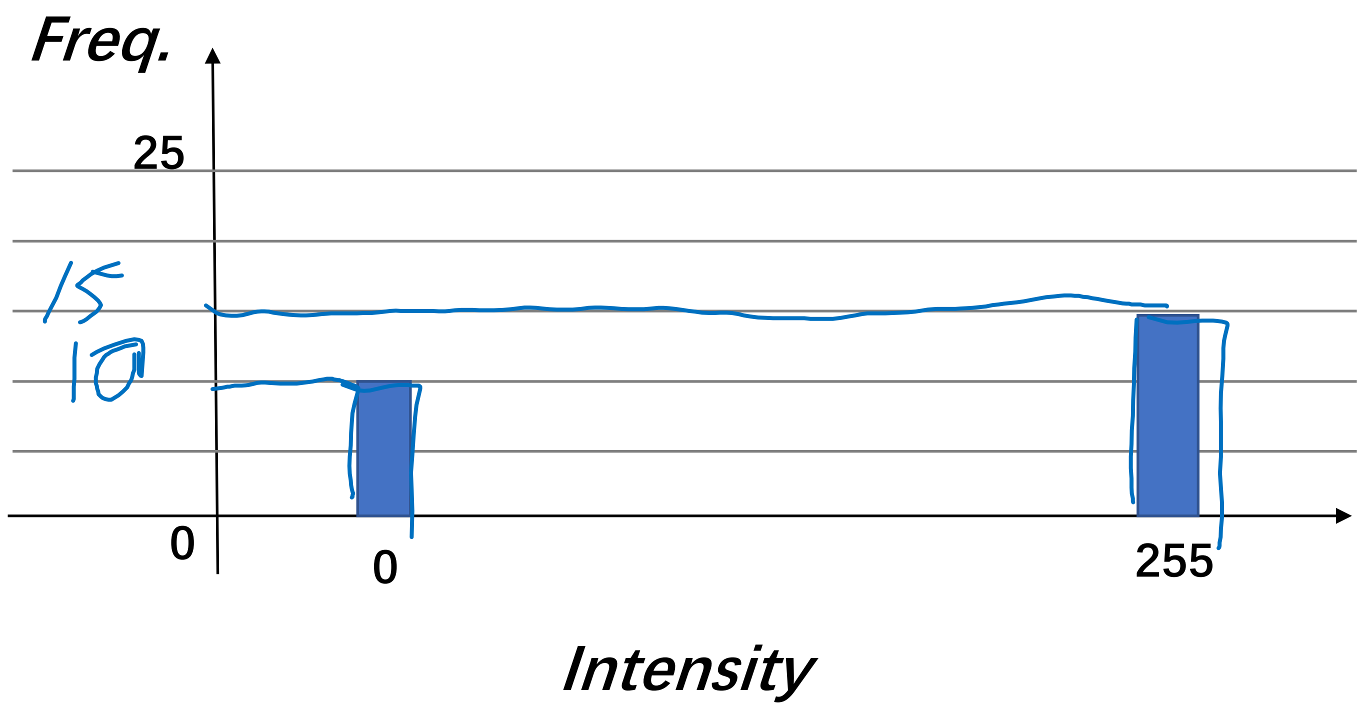


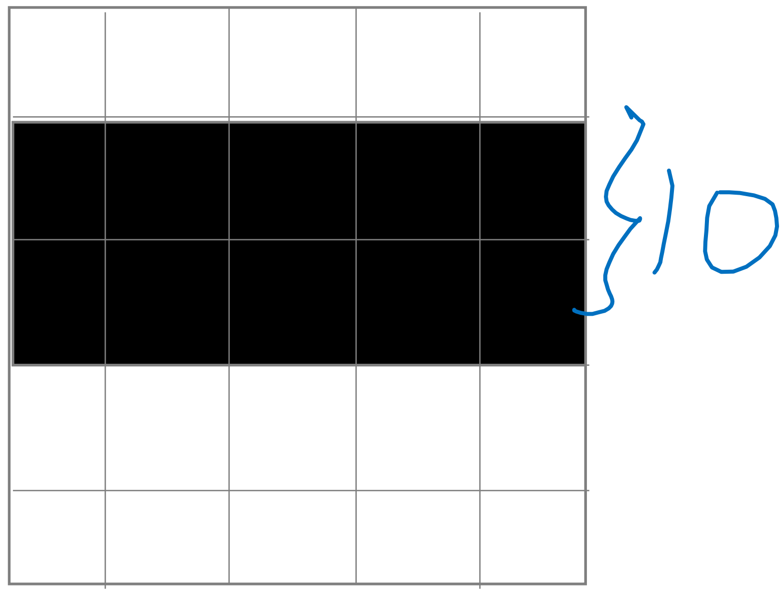
Histogram



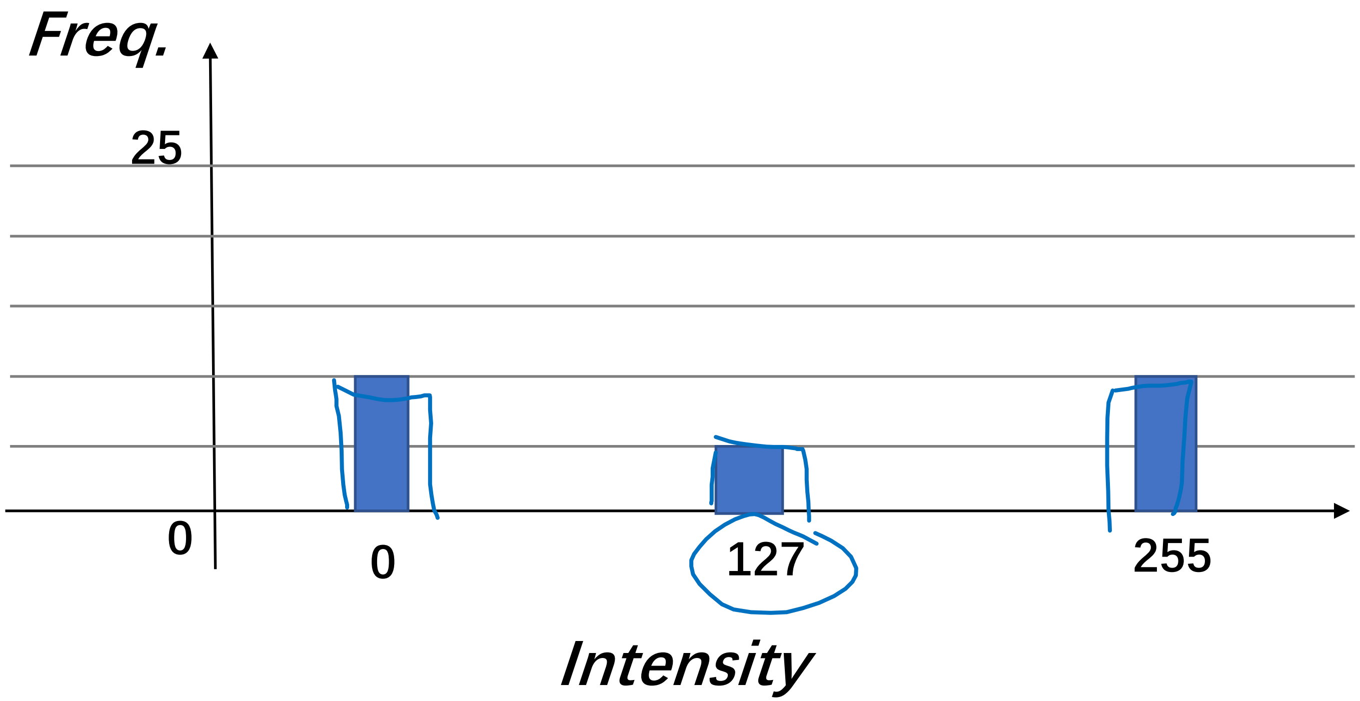


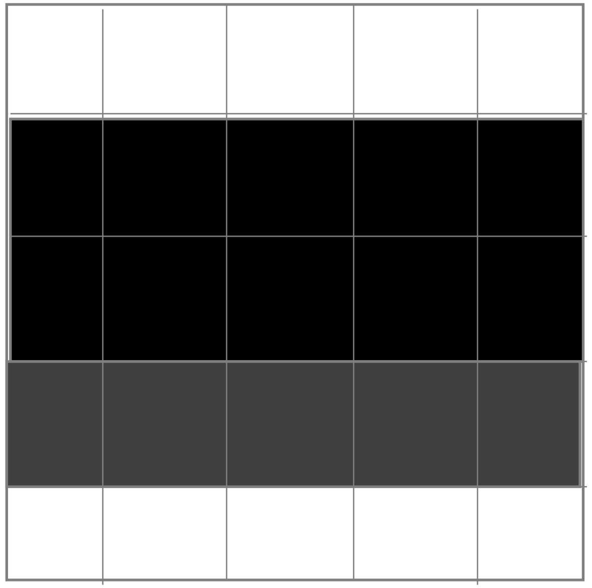
Histogram





Histogram





Example 2: Histogram Representation

For a 5x5 sub-region I_subROI= I_ROI (10:14,26:30),

5

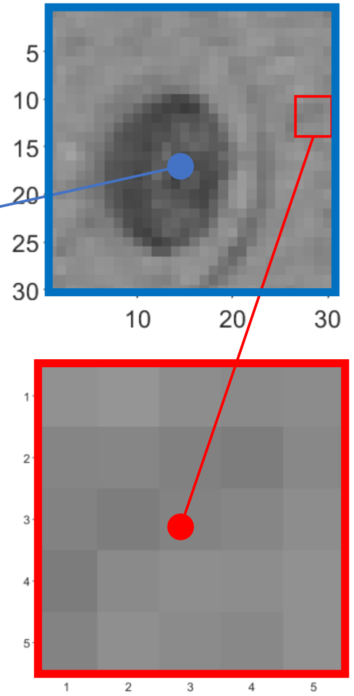
5

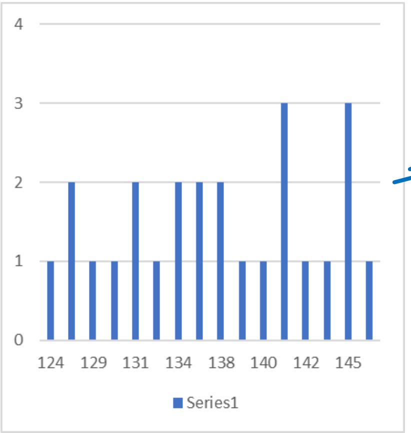
145	149	141	138	140
133	134	~~129~~	~~125~~	136
130	~~125~~	131	134	141
124	138	142	141	145
131	143	139	136	145

Gray Value	124	125	129	130	131	133	134	136	138	139	140	141	142	143	145	149
# of occurrence	1	2	1													
cumulative # of occurrence	1	3	4													

Example 2: Histogram Representation

For a 5x5 sub-region I_subROI= I_ROI (10:14,26:30),

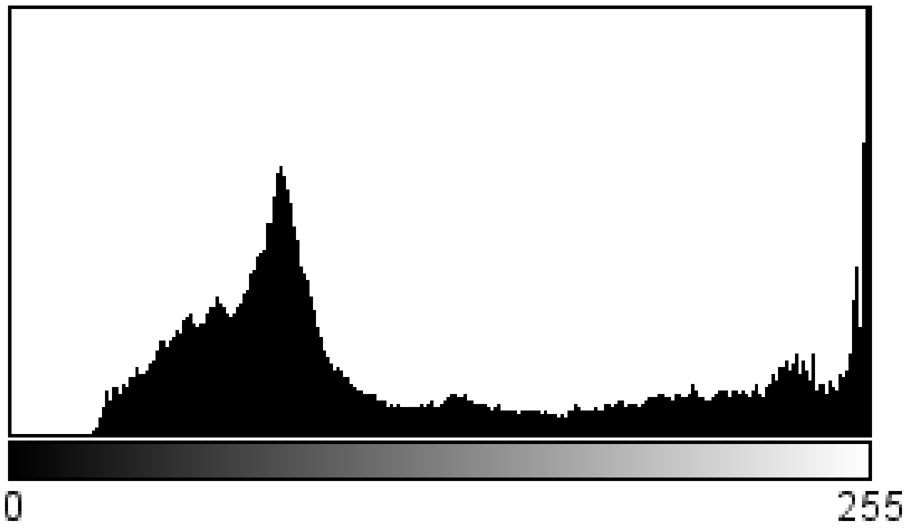




145	149	141	138	140
133	134	129	125	136
130	125	131	134	141
124	138	142	141	145
131	143	139	136	145

Gray Value	124	125	129	130	131	133	134	136	138	139	140	141	142	143	145	149
# of occurrence	1	2	1	1	2	1	2	2	2	1	1	3	1	1	3	1
cumulative # of occurrence	1	3	4	5	7	8	10	12	14	15	16	19	20	21	24	25

Histogram

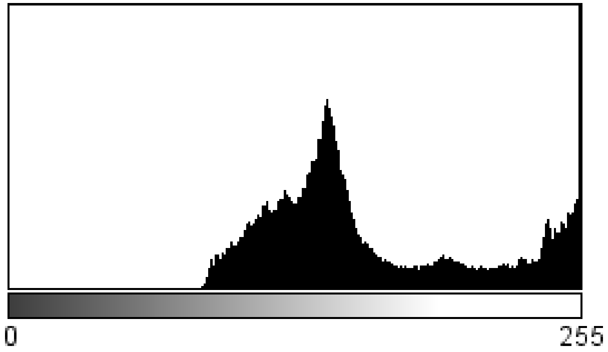


Count: 153436
Mean: 124.213
StdDev: 72.364

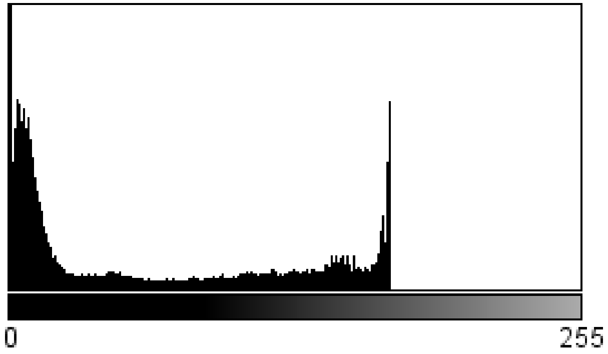
Min: 21
Max: 255
Mode: 255 (4159)

Histogram: Infer Brightness

H1



H2

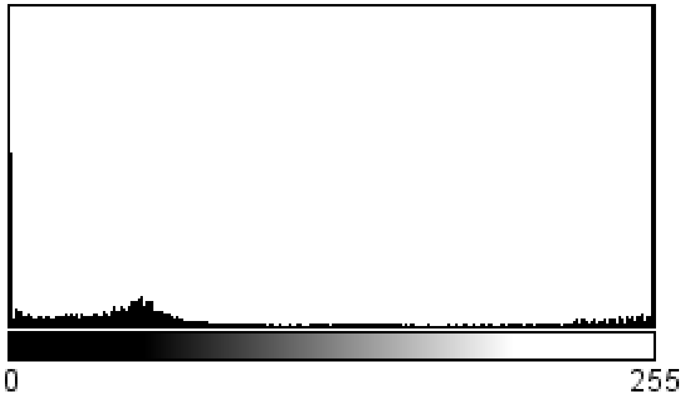


A

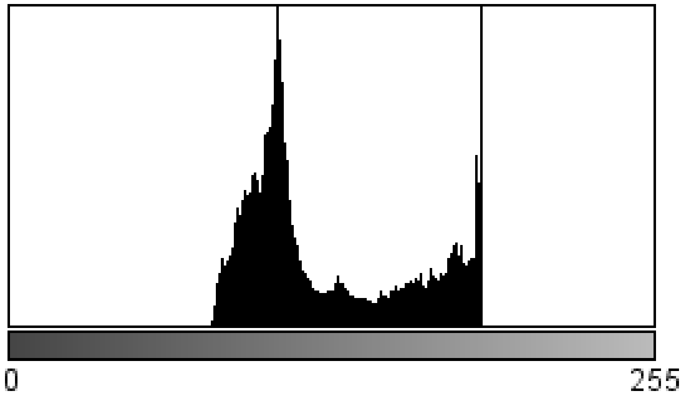
B

Histogram: Infer Contrast

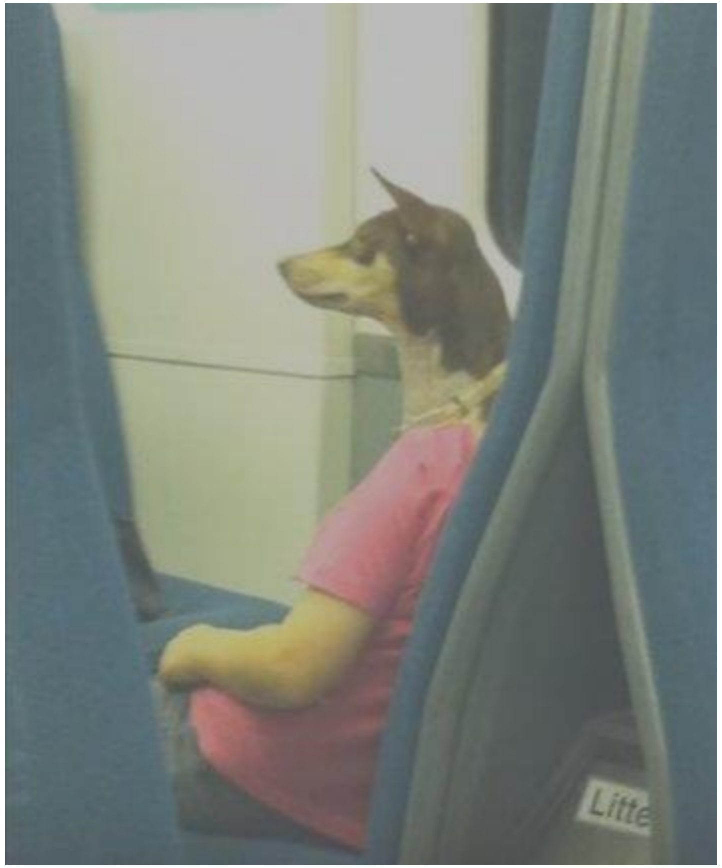
H1



H2



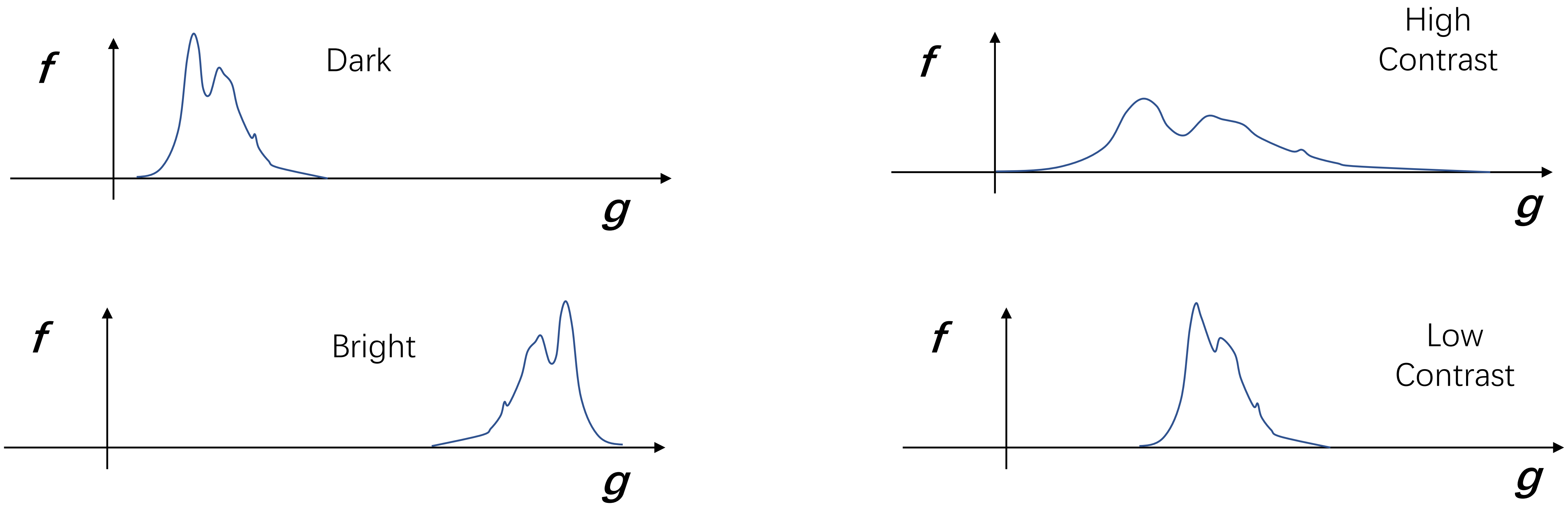
A



B

Histogram (Information available)

- Distribution of the intensity levels



What can you say about the image given its histogram?

Image Processing

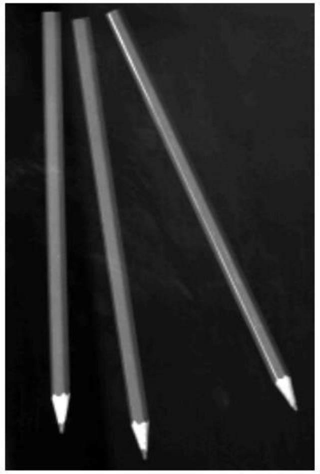
ECE 470 Introduction to Robotics

Image Processing

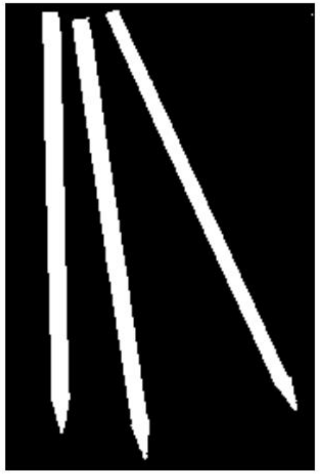
- Image Enhancement
  - Thresholding & Histogram Processing
  - Filtering
- Image Analysis
  - Feature Detection
    - Edges
    - Lines & Shapes
    - Interest points- Corners
  - Target Tracking

Thresholding

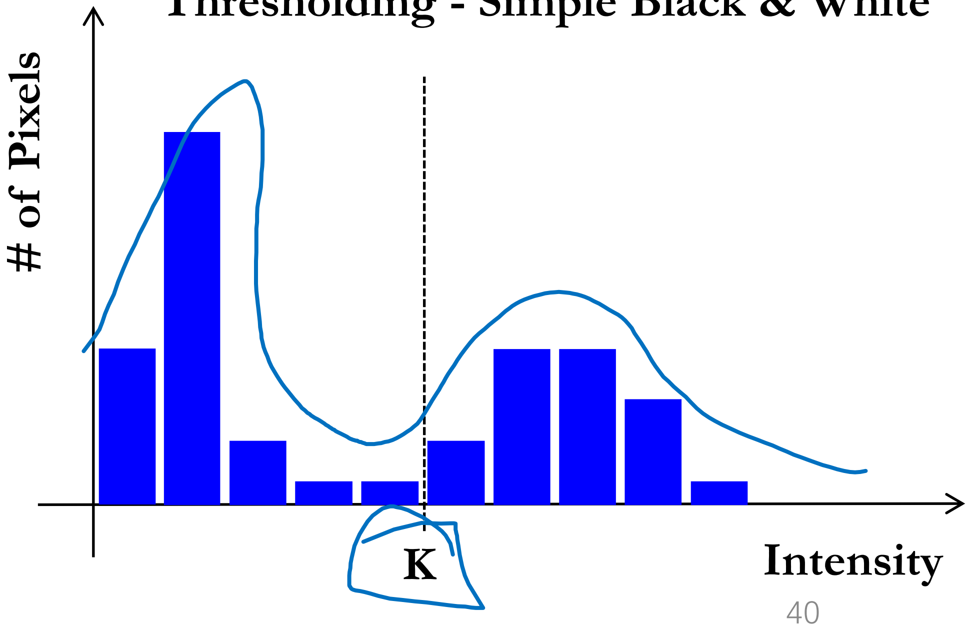
- A threshold parameter "K" is defined using intensity histogram
- Intensity of each pixel is changed to completely dark, "0" or completely bright "1" based on the "K" value
- If intensity of an image pixel, p(m,n)>= K, then p(m,n) = 1
- Else, p(m,n)=0



Original



After Thresholding

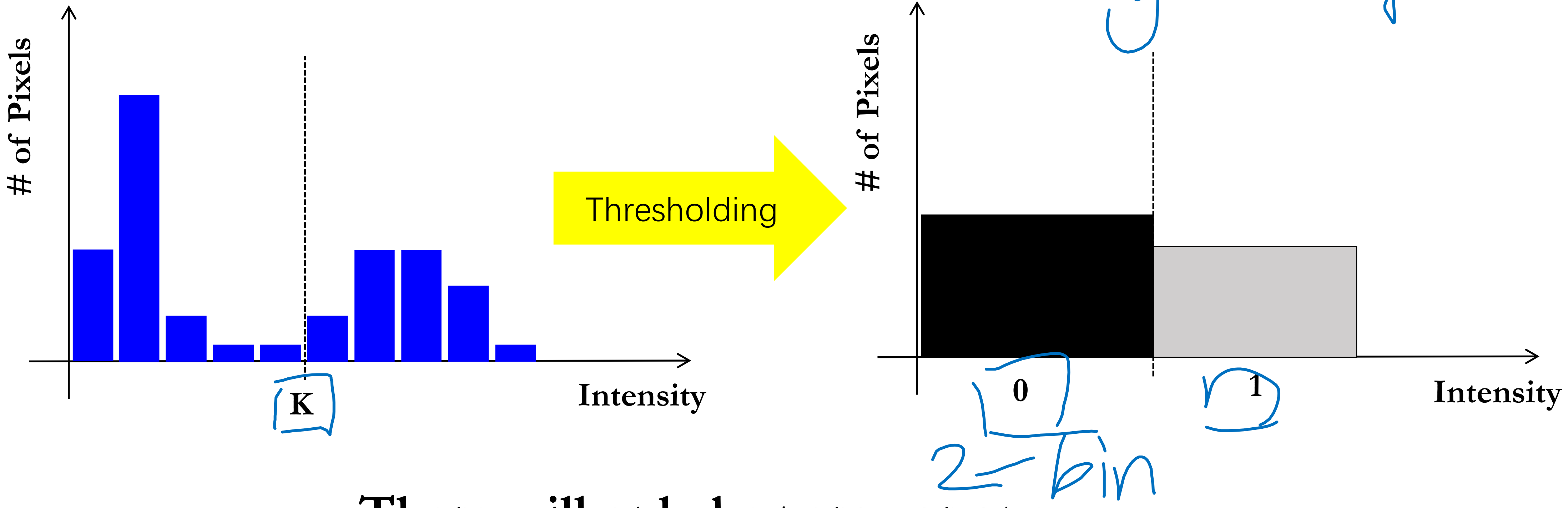


Ref: *Twan Maintz, Digital & Medical Image Processing, 2005*

Recap: Thresholding

What will the new histogram look like?

binary image



There will only be zeros or ones

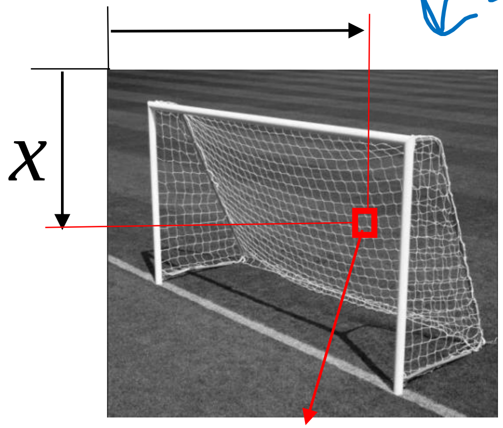
Background Removal

Static

A difference image between two images taken at time i and time j may be defined as

$$d_{ij}(x,y) = \begin{cases} 1; & \text{if } \left| f(x,y,i) - f(x,y,j) \right| > T \\ 0; & \text{Otherwise} \end{cases}$$

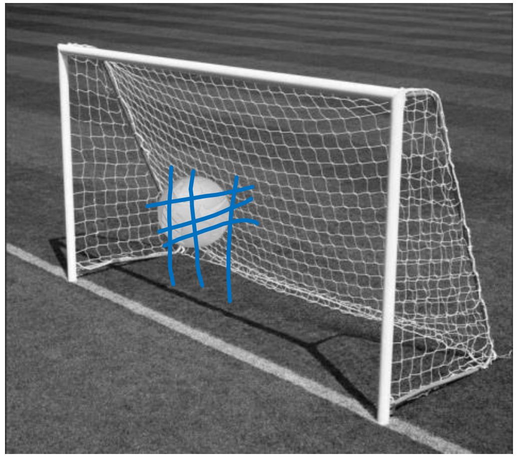
where T is a threshold value, and $f(x,y,i)$ is the intensity of pixel at x, y, in the i-th image.



$f(x,y,i)$

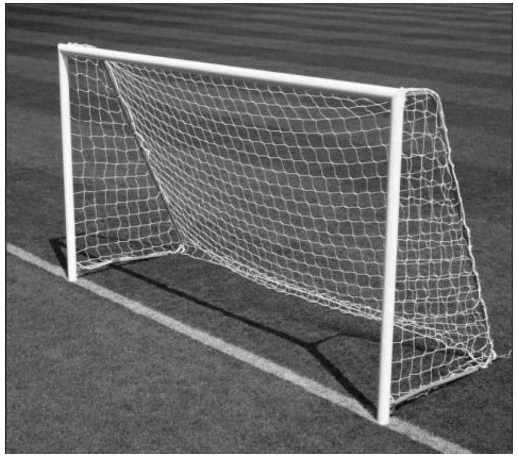
$f = 0$; Black, $f = 1$; White

Current Image



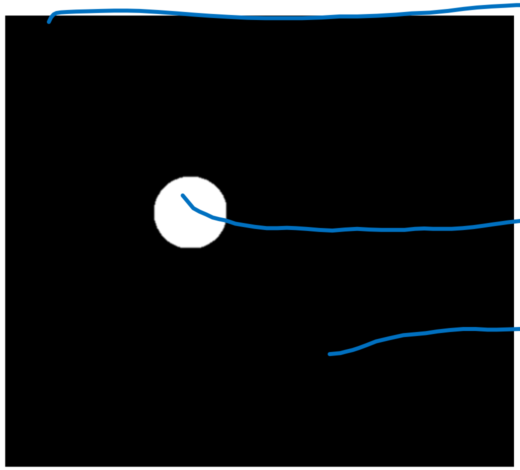
−

Background Image



=

Difference Image



binary

1

0

Ref: *Computer Vision Handouts, Harry Asada, MIT*

Background Removal - Activity

Given:

Two images in Gray scale, namely

BackgroundImage.jpeg and CurrentImage.jpeg

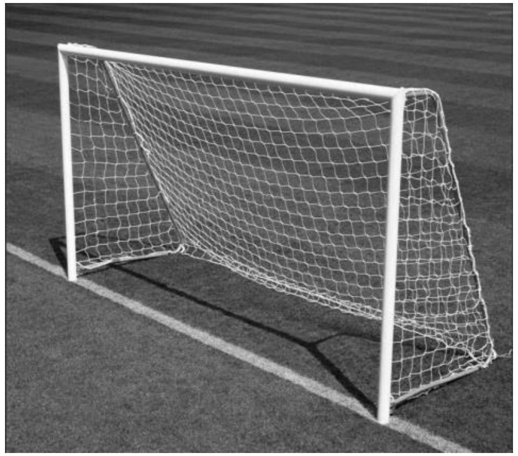
Write a code to determine the difference image DifferenceImage.

Compare with the original image and comment

CurrentImage.jpeg

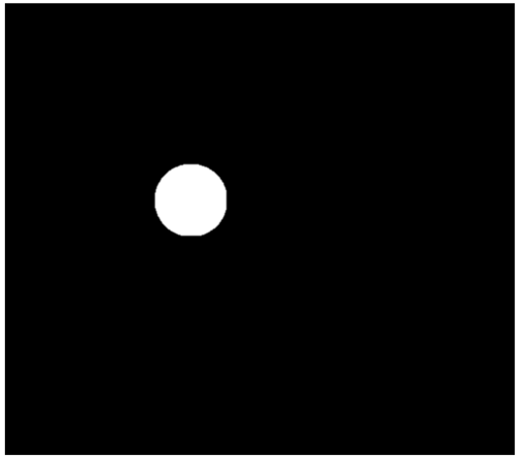
–

BackgroundImage.jpeg



=

DifferenceImage



Ref: *Computer Vision Handouts, Harry Asada, MIT*

Background Removal – Activity

Matrix variable

image file

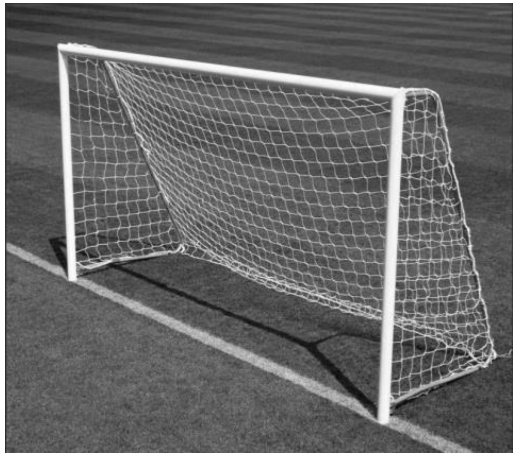
In Matlab,

```
CurrentImage = imread('CurrentImage.jpg');
BackgroundImage = imread('BackgroundImage.jpg');
DifferenceImage = CurrentImage - BackgroundImage;
DifferenceImage = im2bw(DifferenceImage, T);
```

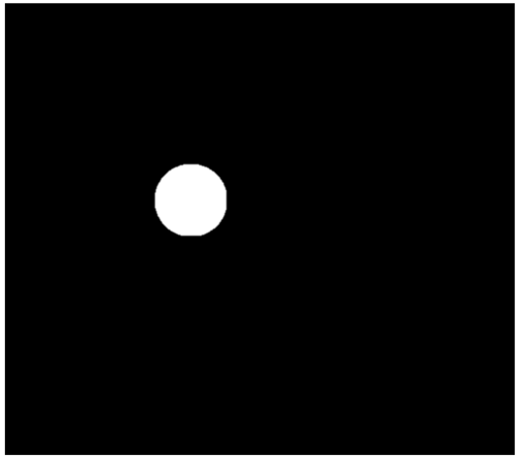
binarized

CurrentImage.jpeg

BackgroundImage.jpeg



DifferenceImage



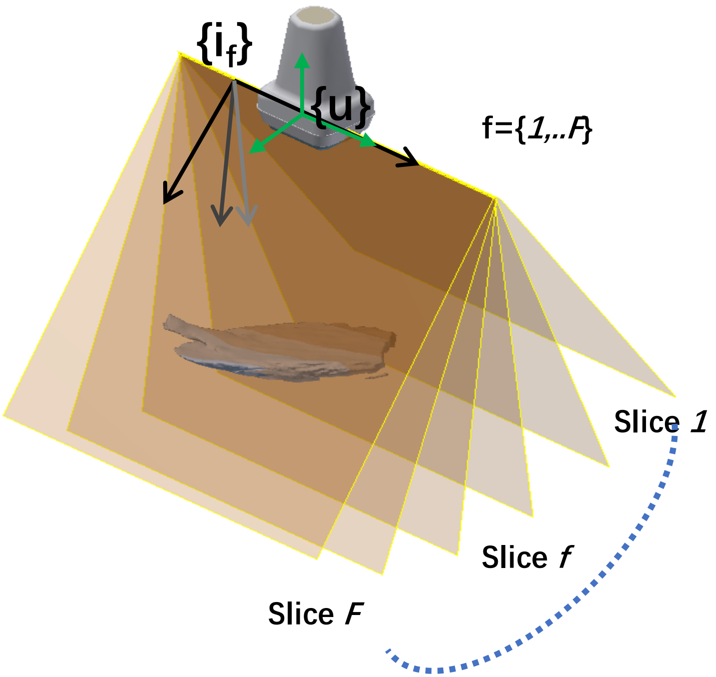
Ref: *Computer Vision Handouts, Harry Asada, MIT*

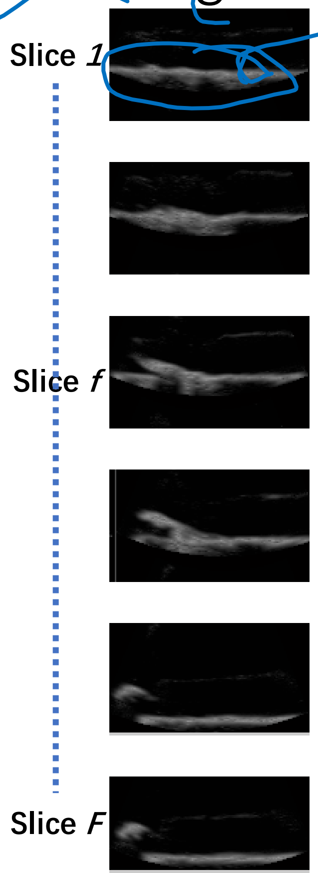
Thresholding

→ background

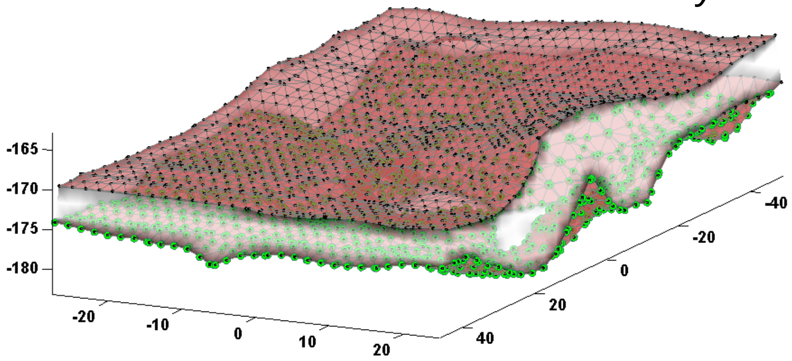
- Ultrasound Images Example
  - 3D reconstruction of hyperechoic organ from 2D images

organ





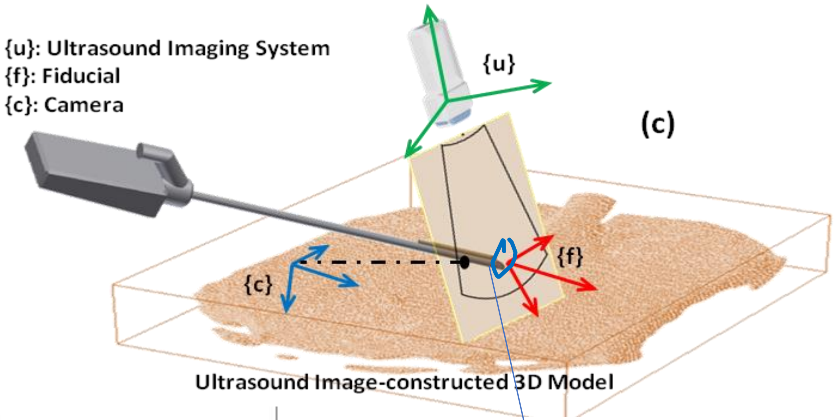
Interested in 3D Surface Geometry

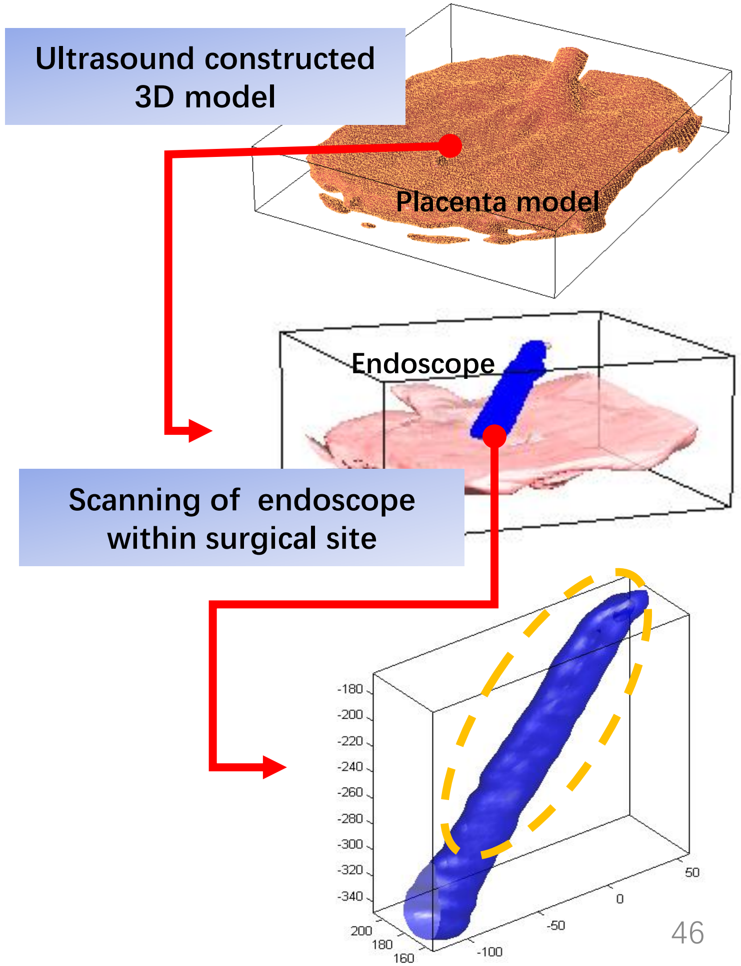


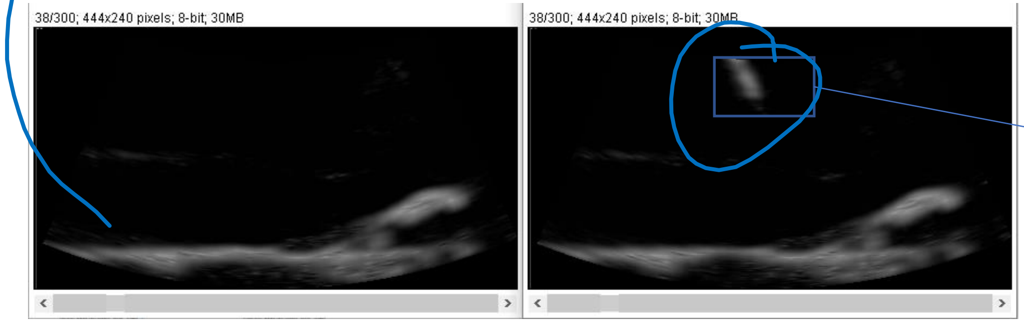
Background Removal

- Ultrasound Images Example
  - 3D reconstruction of moving surgical instrument

static







background image

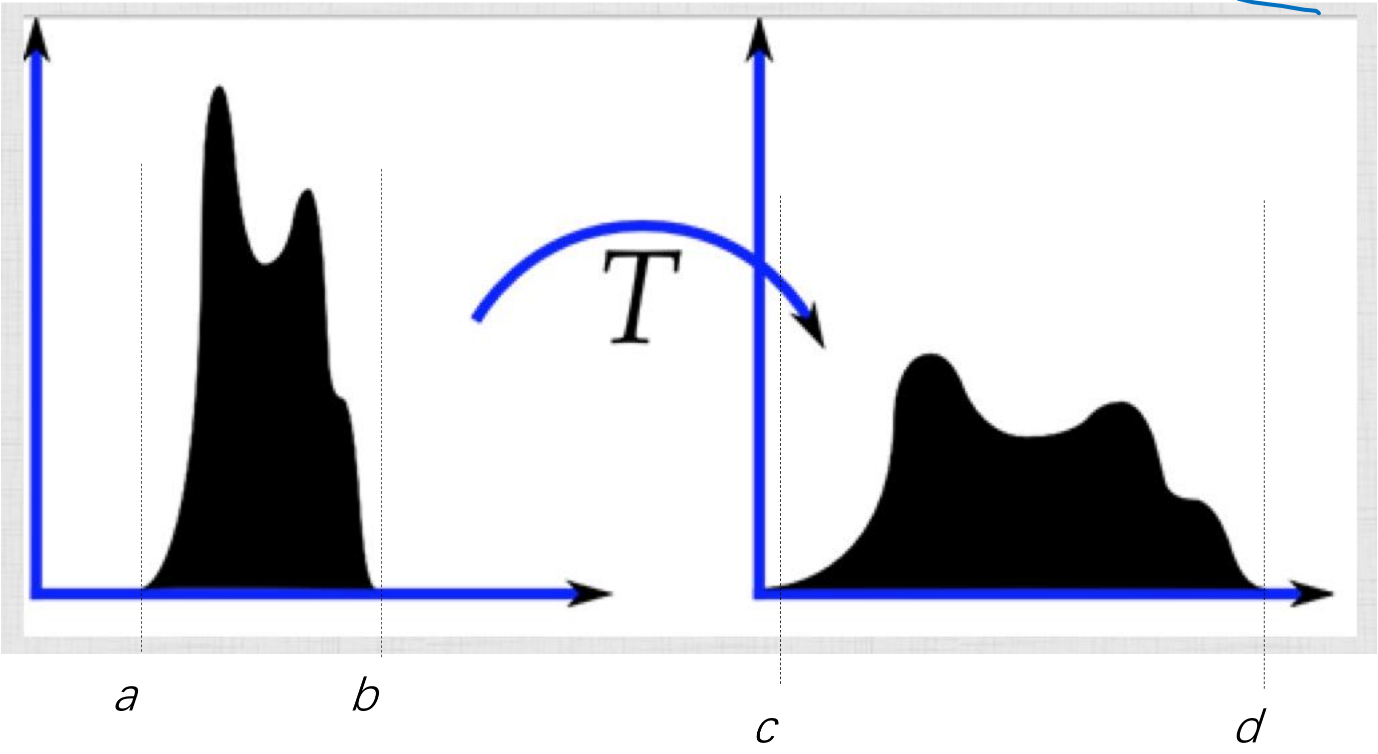
current image

Interested in Localizing the Tool

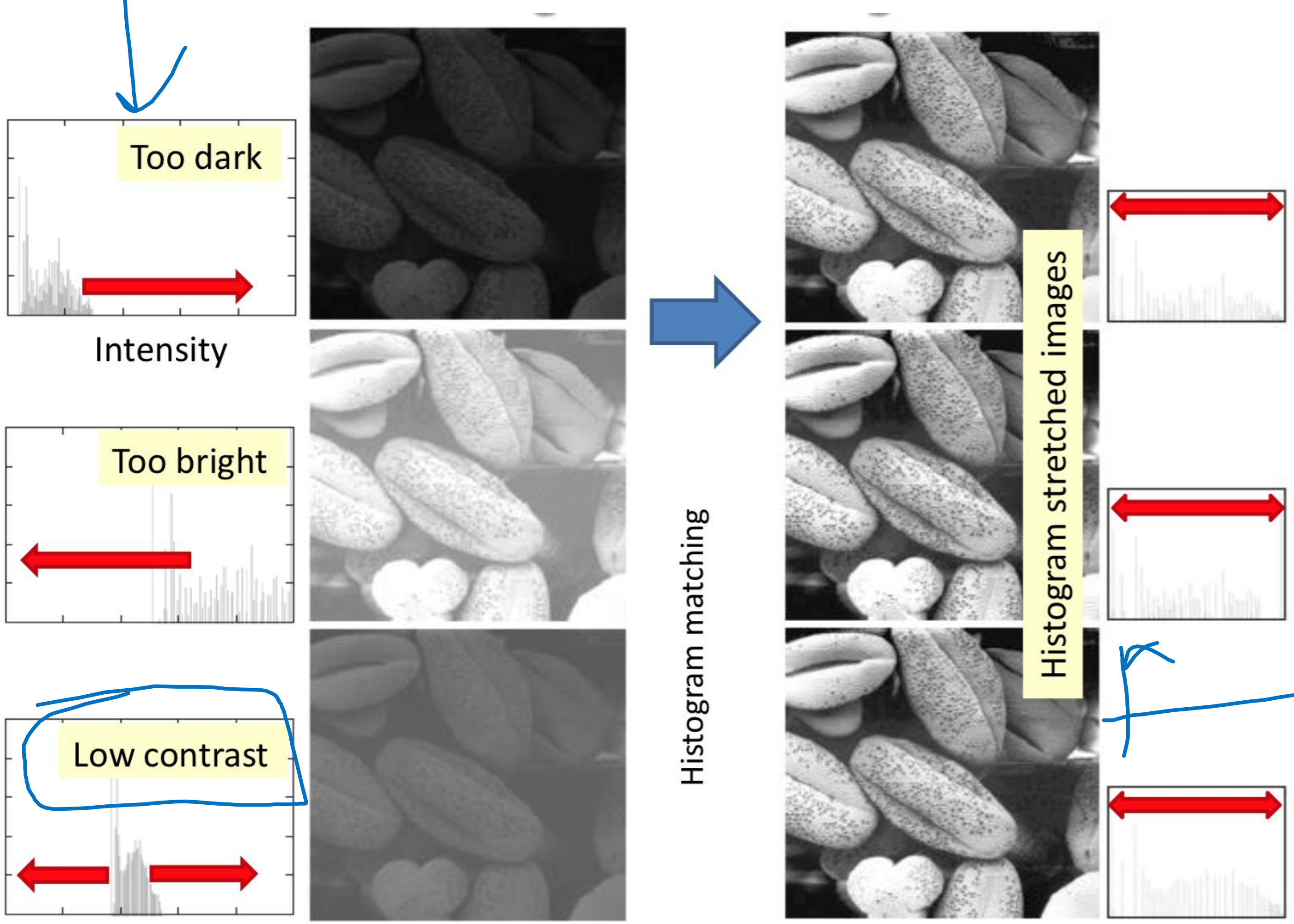
Histogram Processing - Stretching

Matlab Function for Histogram stretching:

EnhancedImage=Imadjust(Image, [a,b],[c,d], gamma)



Histogram Processing - Stretching



Slide From: *Computer Vision Handouts, Harry Asada, MIT*

spread over larger range

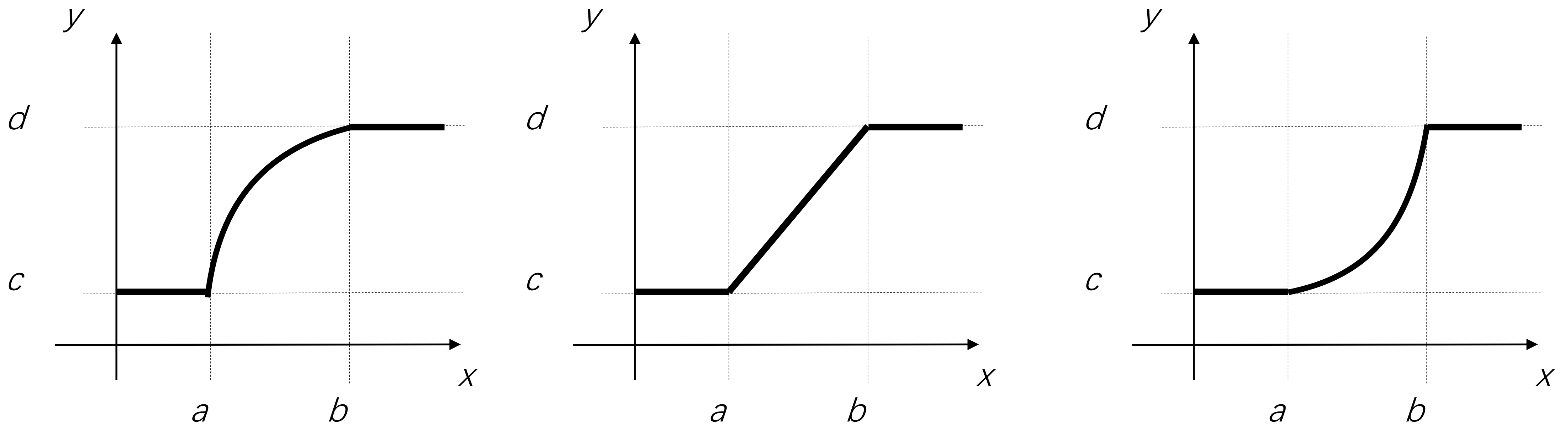
Histogram Processing - Stretching

$$y = \left(\frac{x-a}{b-a}\right)^{\gamma}(d-c)+c$$

x: input pixel

y: output pixel

- Pure stretching when γ =1
- Combined stretching & gamma transform when γ ≠ 0 and 1.



γ <1 γ =1 γ >1

Histogram Processing - Stretching

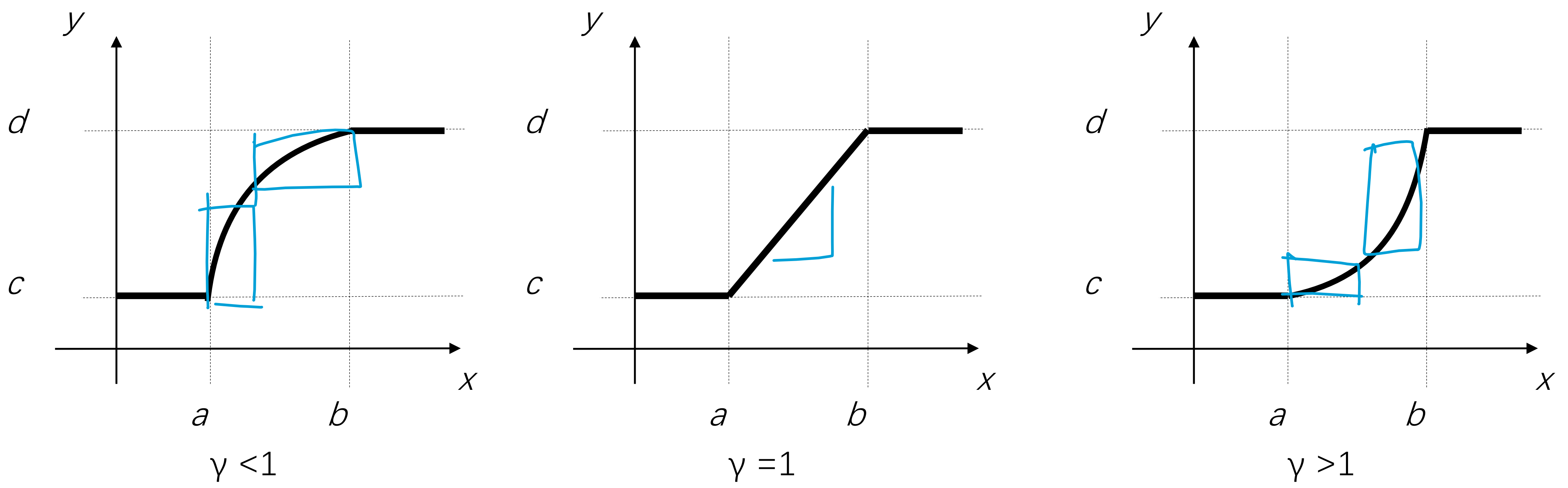
imadjust(Im, , ,)

$$y = \left(\frac{x-a}{\boxed{b}-\boxed{a}}\right)^{\boxed{\gamma}}(\boxed{d}-\boxed{c})+c$$

x: input pixel

y: output pixel

- Pure stretching when γ =1
- Combined stretching & gamma transform when γ ≠ 0 and 1.



Histogram Processing - Stretching

- Is there a way to automatically determine the extent of histogram stretch?

Especially relevant for Robotics

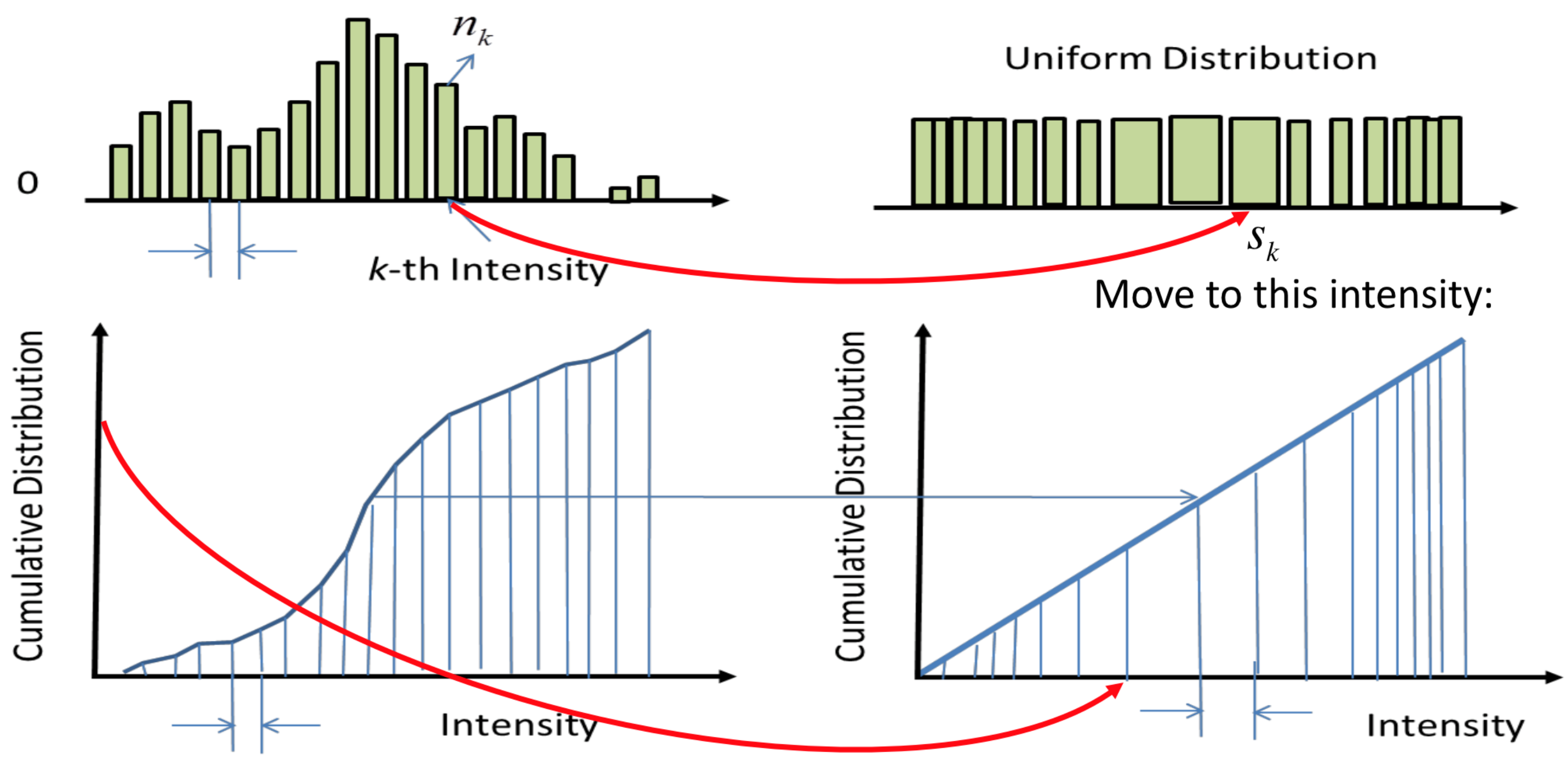
Histogram Processing - Equalization

- Histogram stretching **require user input**.
- Histogram equalization is an entirely **automatic.**
- Histogram equalization can Stretch/Compress an image such that:
    - Pixel values (intensity) that occur more frequently get stretched and become more visible.
    - Pixel values that occur infrequently get compressed and become less visible.

Ref: *Computer Vision Handouts, Harry Asada, MIT*

Histogram Processing - Equalization

Treat it as a type of probability: $p_k \approx \frac{n_k}{n}, \quad n = MN$



It transforms the histogram to be flat keeping area of individual blocks same

$$s_k = \frac{n_1 + n_2 + \cdots + n_k}{n}(L-1)$$

Typically L = 256

Ref: *Computer Vision Handouts, Harry Asada, MIT*

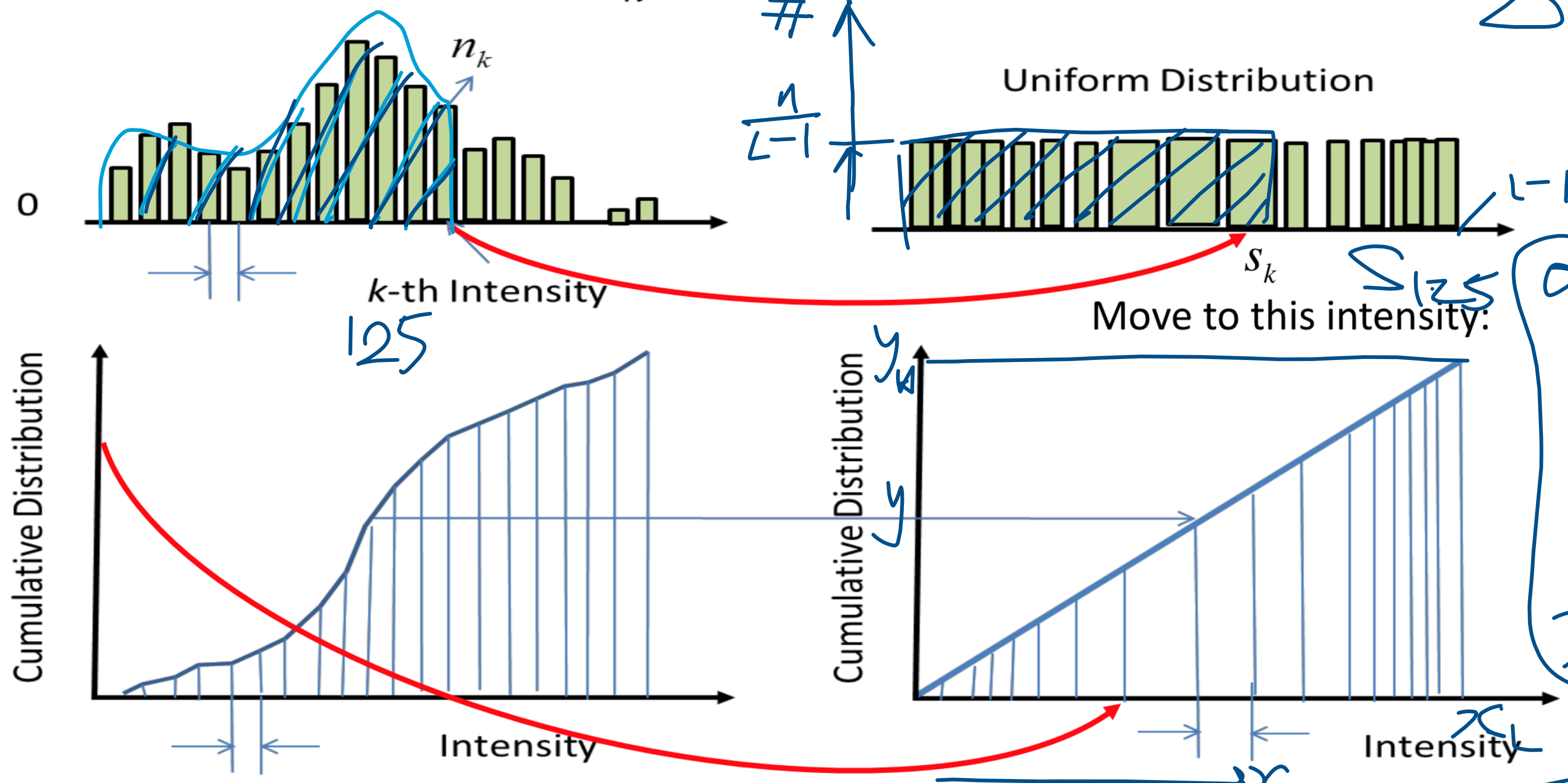
Histogram Processing - Equalization

$A_1 = A_2$

$\sum n_k = \left(\frac{n}{L-1}\right) s_k$

$\therefore s_k = \frac{\sum n_k}{n}(L-1)$

Treat it as a type of probability: $p_k \approx \frac{n_k}{n}, \quad n = MN$



or

$\frac{x}{x_L} = \frac{y}{y_H}$

$x = \frac{y}{y_H} x_L$

It transforms the histogram to be flat keeping area of individual blocks same

$$s_k = \frac{n_1 + n_2 + \cdots + n_k}{n}(L-1)$$

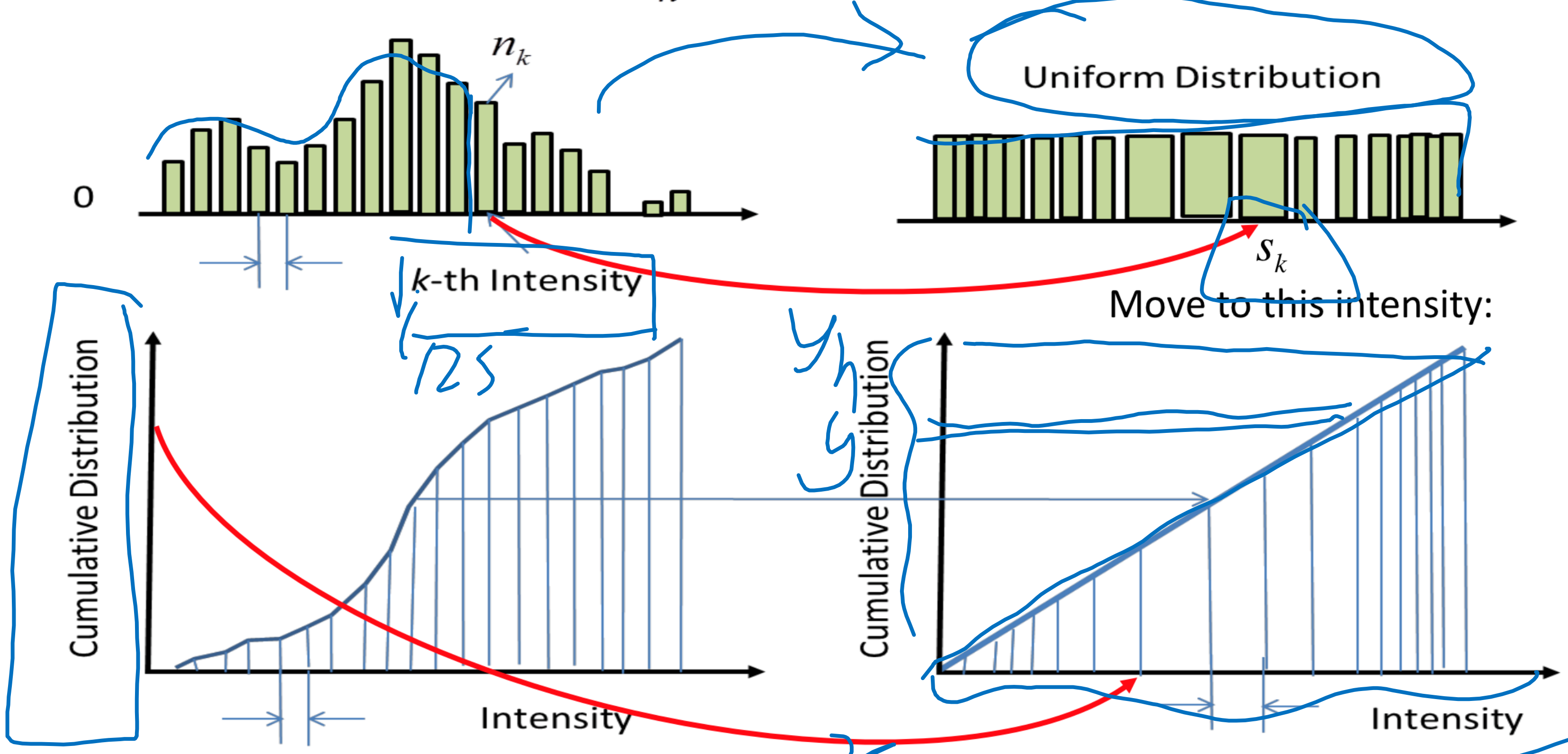
$x \qquad \frac{y}{y_H} \qquad x_L$

Typically L = 256

Ref: *Computer Vision Handouts, Harry Asada, MIT*

Histogram Processing - Equalization

Treat it as a type of probability: $p_k \approx \frac{n_k}{n}, \quad n = MN$



It transforms the histogram to be flat keeping area of individual blocks same

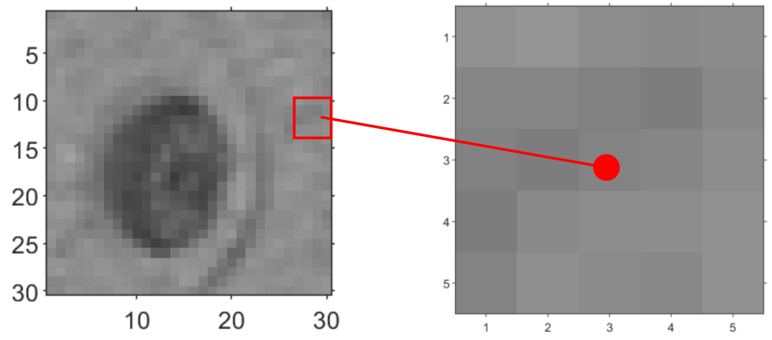
$$s_k = \frac{n_1 + n_2 + \cdots + n_k}{n}(L-1)$$

Typically L = 256

Ref: *Computer Vision Handouts, Harry Asada, MIT*

Example 3: Histogram Equalization

Perform histogram equalization and plot the new histogram



Transformation for histogram equalization:

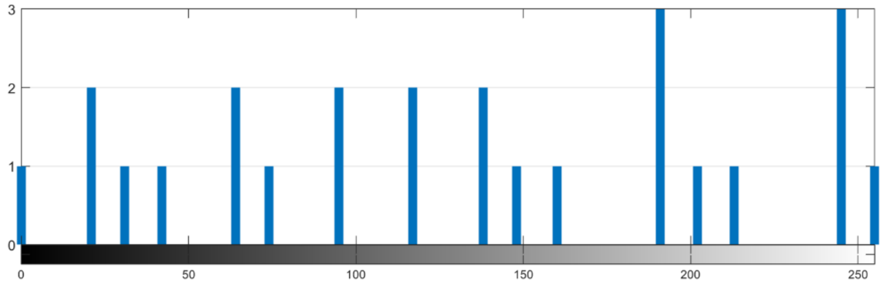
$$S_k = \left(\frac{n_1 + n_2 + \cdots n_k}{n}\right)(L-1)$$

Substitute L-1=255, n=5x5, $S_k = \left(\frac{n_1+n_2+\cdots n_k}{25}\right)(255)$

Rounding off to nearest integer,

n	**1**	**2**	**1**	**1**	**2**	**1**	**2**	**2**	**2**	**1**	**1**	**3**	**1**	
lup	**11**	**31**	**41**	**51**	**72**	**82**	**102**	**123**	**143**	**153**	**164**	**194**	**204**	**21**
wn	**10**	**30**	**40**	**51**	**71**	**81**	**102**	**122**	**142**	**153**	**163**	**193**	**204**	**21**

**Rounding up or down are both acceptable



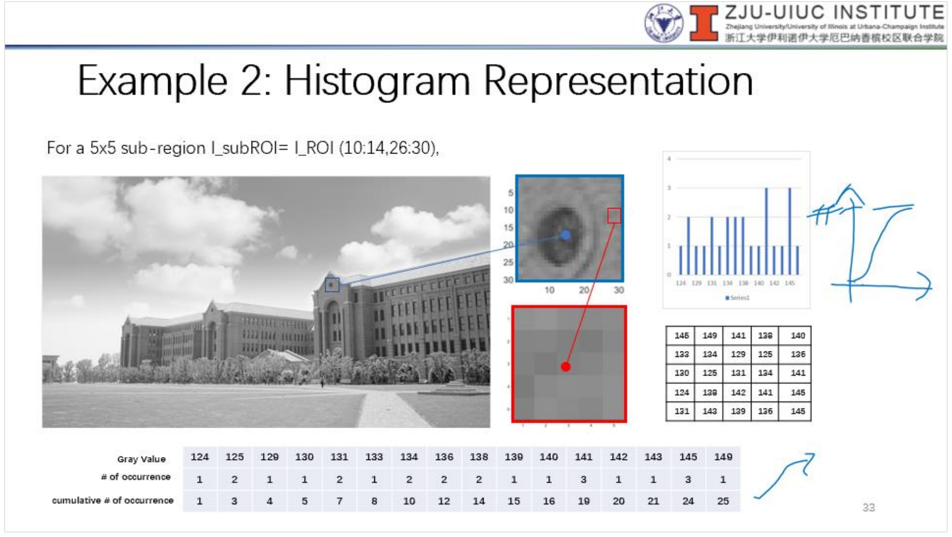


Image Processing (Next Lecture)

- Image Enhancement
  - Thresholding & Histogram Processing
  - Filtering
- Image Analysis
  - Feature Detection
    - Edges
    - Lines & Shapes
    - Interest points- Corners
    - Target Tracking

Filtering (Next Lecture)

- Operation that modify pixels based on their neighbourhood values