CSET340

Advanced Computer Vision and Video Analytics

05th Jan. to 09th Jan. 2026

Overall Course Coordinator-
Dr. Gaurav Kumar Dashondhi
Gaurav.dashondhi@bennett.edu.in

Note : Any query related to course then first connect with overall course coordinator.

CSET340

Course Coordinator & Team

Faculty Name	Faculty mail
Dr. Gaurav Kumar Dashondhi (Overall Course Coordinator)	Gaurav.dashondhi@bennett.edu.in
Dr. Shallu Sharma	Shallu.sharma@bennett.edu.in
Dr. Azad Singh	Ajad.singh@bennett.edu.in

CSET340 - Syllabus

Course Contents:

Module I: **7 lecture hours**

Introduction to Computer Vision, The Four Rs of Computer Vision, Challenges in Computer Vision, Low-level vs High-level processing, Two View Geometry, Binocular Stereopsis: Camera and Epipolar Geometry, Planar Scenes and Homography, Depth estimation and multi-camera views, Robust Correspondence Estimation, 3-D reconstruction, Auto-calibration, DLT and RANSAC, Structure from Motion, Hough Transform, Fourier Transform, Interest Point Detection, Edge Detection, Local Binary Pattern, Convolution and Filtering, Gaussian derivative filters, Gabor Filters, DWT, Pyramids, Visual Matching: Bag-of-words, Pyramid Matching, Part based recognition models, Recognition: Detectors and Descriptors, Optical Flow & Tracking.

Modulo II: **7 lecture hours**

Shape from Texture, Color, motion and edges, Face Detection, Feature Tracking, Motion Layers, SIFT & Single Object Recognition, Dense Neural Networks, Backpropagation, Convolutional Neural Networks (CNNs), AlexNet, VGG16, Image Quality Enhancement, Image Restoration, Super resolution, Residual Learning, Visual Saliency detection.

Module III: **7 lecture hours**

Evolution of CNN Architectures: AlexNet, MobileNet, InceptionNets, ResNets, DenseNets, 3D CNN for images and videos, Unsupervised image segmentation, Watershed, Level set, Active Contour, GraphCut, Supervised image segmentation, Agglomerative clustering, Segmentation as pixel classification, UNets, FCN, Deep Generative Models, GANs, VAEs, PixelRNNs, naDE, Normalizing Flows, Zero-shot, One-shot, Few-shot Learning, Self-supervised Learning, Reinforcement Learning in Vision, Video Analytics, Spatial Domain Processing, Frequency Domain Processing, Background Modelling, Crowd Analysis, Video Surveillance, Traffic Monitoring, Intelligent Transport System.

Module IV: **7 lecture hours**

Optical Character Recognition, Online Character Recognition, Visual Anomaly Detection, Anomalous action recognition, Post Estimation, Action Recognition, Graph CNN, Shape Recognition, Shape Retrieval, Content based Image retrieval, Visual Instance Recognition, Emotion Recognition from videos, Video Generation.

Studio Work / Laboratory Experiments:

In the lab work, the students will Implement the state-of-the-art computer vision and video analytics concepts to different applications.

Text Books :

1. Rajalingappaa Shanmugamani, *Deep Learning for Computer Vision* (1st ed.), Packt Publishing, 2018. ISBN 9781788295628 .
2. Nedumaan J., Prof Thomas Binford, J. Lepika, J. Tisa, J. Ruby and P. S. Jagadeesh Kumar, *Modern Deep Learning and advanced Computer Vision* (1st ed.), Intel, 2019. ISBN 9781708798641 .

Reference Books :

1. Kar Krishnendu, *Mastering Computer Vision with Tensor Flow* (1st ed.), Packt, 2020. ISBN 9781838826939.

Assessment Scheme:

Components	Internal Assessment	Mid Term Exam	End Exam	Total
Weightage (%)	45	20	35	100

Course Overview : Module 1

1. Introduction and challenges in the Computer Vision.

2. Two view Geometry, Camera and Epipolar Gometry.

3. Camera Calibration, auto calibration, 3D reconstruction and camera pipeline.

4. Discrete wavelet transform (DWT).

5. Optical flow and Tracking.

Course Overview: Module 2

1. **Evolution of CNN architectures: Alexnet, VGG16, Alexnet, Graph CNN etc.**

2. **Evolution of object detection and tracking based algorithm.**

3. **Image Quality Enhancement.**

    3.1 Image denoising
    3.2 Image deblurring
    3.3 Image Super resolution
    3.4 Image restoration.

4. **Texture Analysis: Beyond GLCM, shape from texture.**

5. **Residual Learning and Visual saliency detection.**

Course Overview: Module 3

1. **Image Segmentation**
   Supervised Segmentation
   Unsupervised Segmentation
   Semantic Segmentation
   Instance Segmentation

2. Generative Models like GAN, VAEs, PixelRNN and all.

3. Self supervised learning, Reinforcement learning in vision and how it aligns with GAN.

4. **Video analytics: Video compression, Video generation etc.**

5. Other topics: Background modeling, crowd analysis, video surveillance, traffic monitoring, intelligent transport system.

Course Overview: Module 4

1. **Optical Character Recognition and online character recognition.**

2. **Pose Estimation, Action recognition, Anomalous action recognition, visual anomaly detection.**

3. **Shape recognition, shape retrieval, Content based image retrieval.**

4. **visual Instance recognition.**

5. **Emotion recognition from videos.**

CSET340 – Course Evaluation

1. **Mid-Semester**: 20 marks
2. **End-Semester**: 40 marks
3. **Project Work**: 20 marks
4. **Laboratory Continuous Internal Assessment**: 20 marks

100 Marks

5. **Programming Environment**: All experiments will be conducted using the Python programming language with OpenCV on the Google Colab platform or Visual Studio Code.

6. **Module Coverage**: (Tentative)
    1. **Before the Mid-Semester** : Modules 1 and 2 will be covered.
    2. **After the Mid-Semester**: Modules 3 and 4 will be covered.
7. **Question Design**: All questions will emphasize logical reasoning and problem-solving.

Application of Image Processing

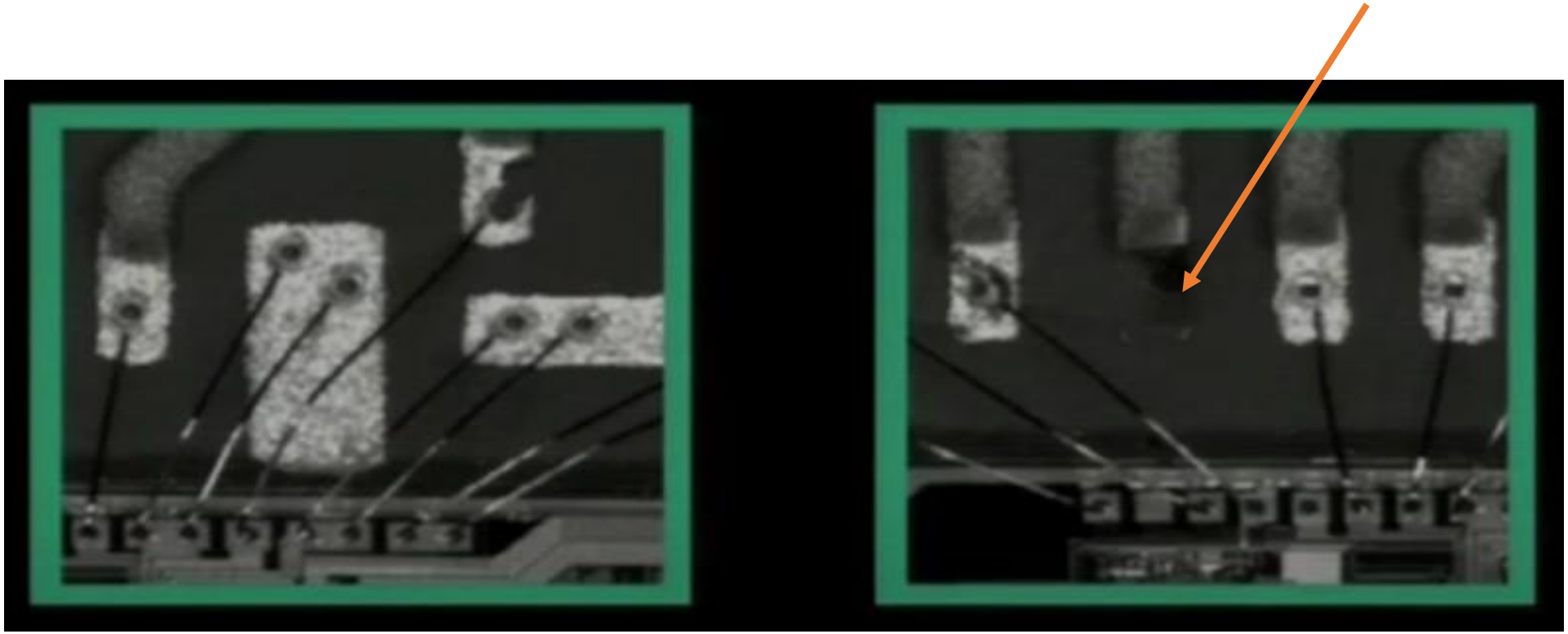
1. Remote sensing.
2. Medical Domain.
3. Security and Surveillance. Object detection and Tracking.
4. Industrial automation.
5. Film and entertainment industry. To add the special effects and create artificial environment.

Application of Image Processing

Automate the Status of Bottle Filling Inspection

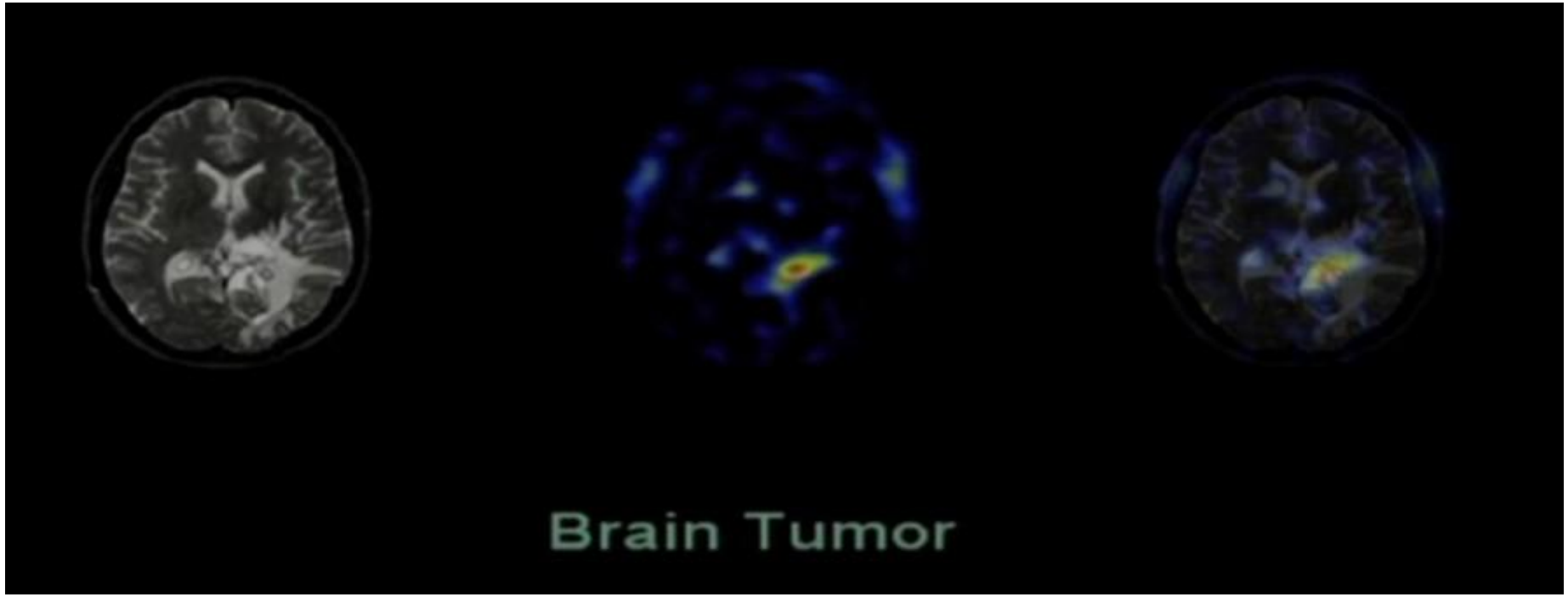
Application of Image Processing

Automate the IC Connection Inspection : Some connections are broken.



Application of Image Processing

Medical Domain



Original Image

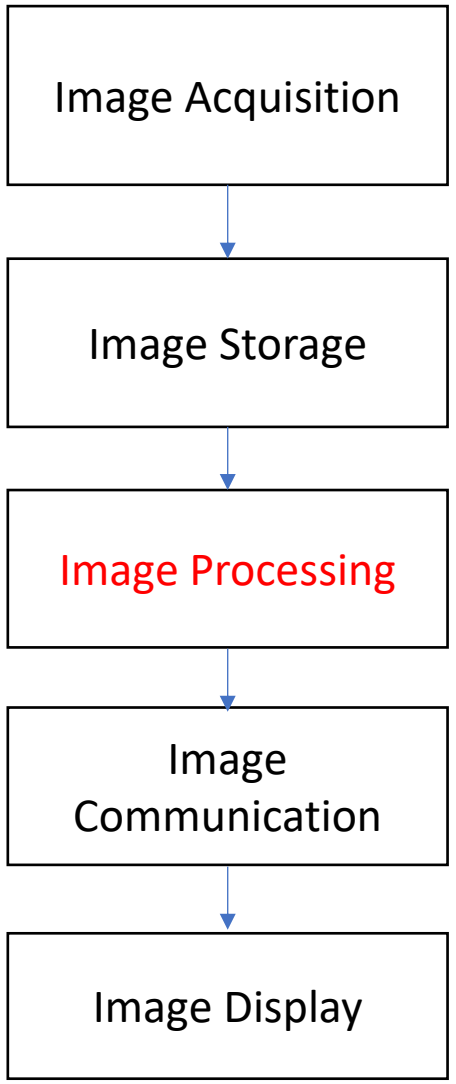
Processed Image1

Processed Image 2

Application of Image Processing

Remote Sensing

Elements of Digital Image Processing Systems



Scanners
Cameras

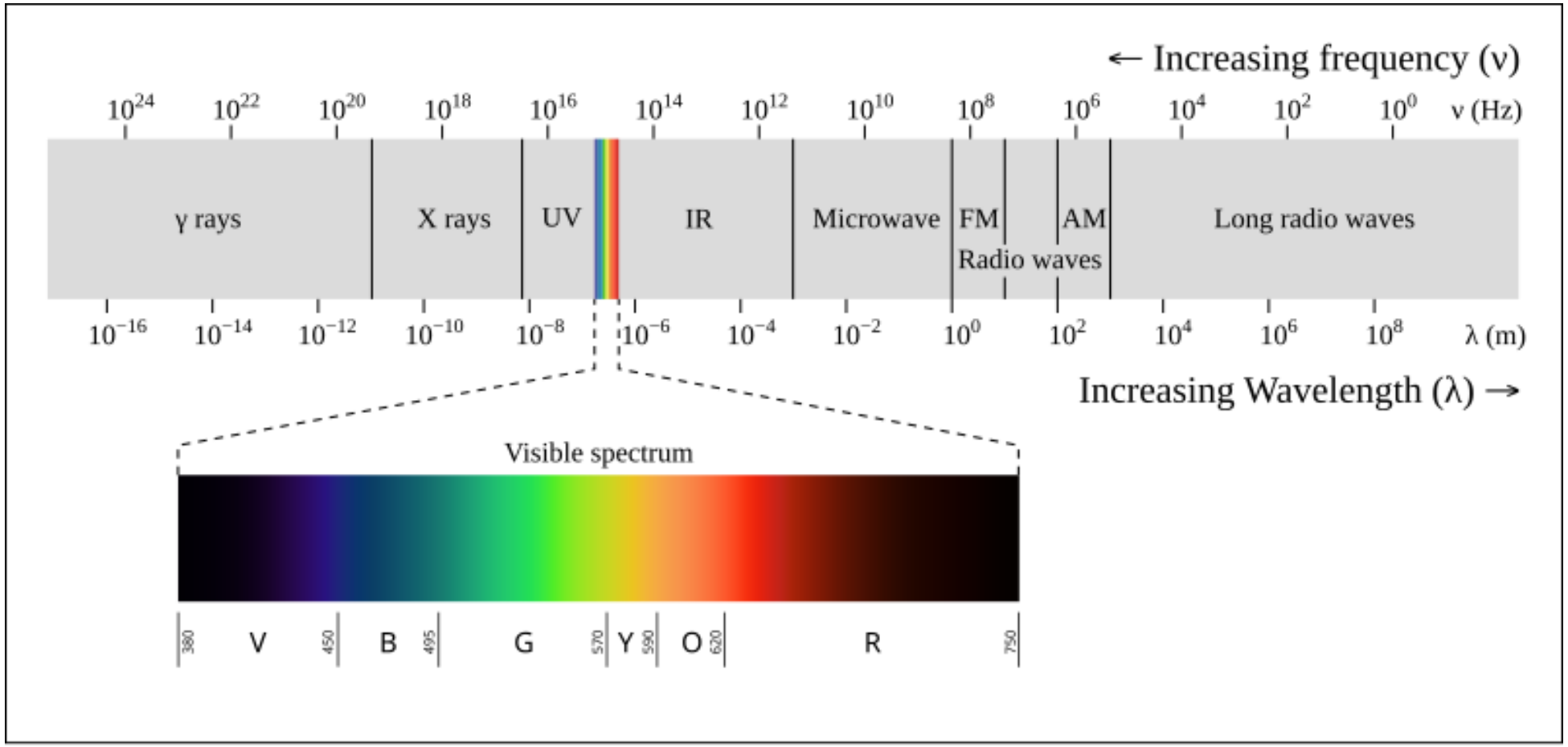
Hard Drives
Solid State Drives (SSD)

Image Enhancement
Image Restoration
Image Segmentation
Image Analysis

Image Transmission using different networks like LAN, WAN

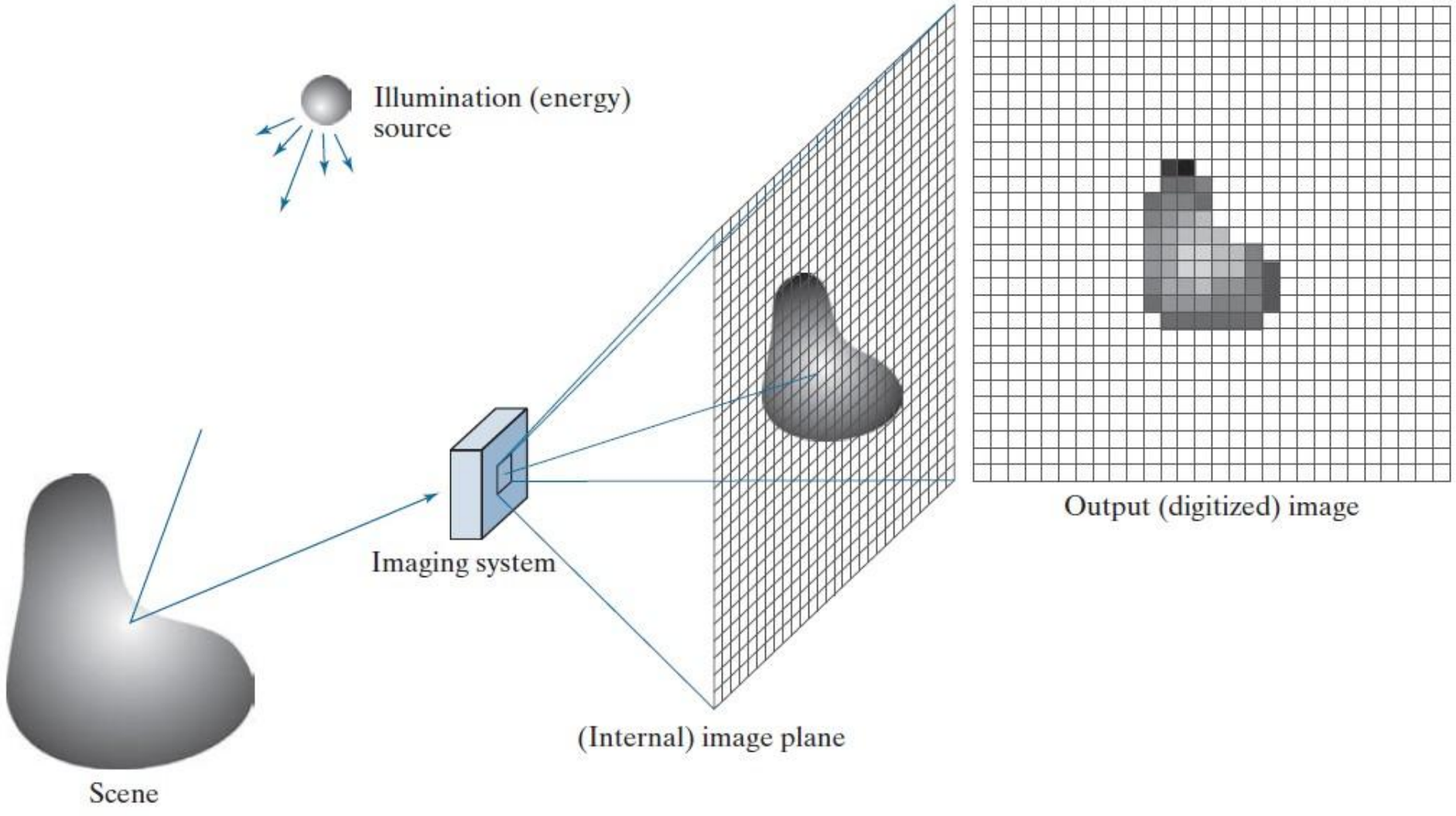
Monitors
Printers

EM Spectrum: Visible Spectrum



Refer: https://www.lumitex.com/blog/visible-light-spectrum

Digital Image Acquisition



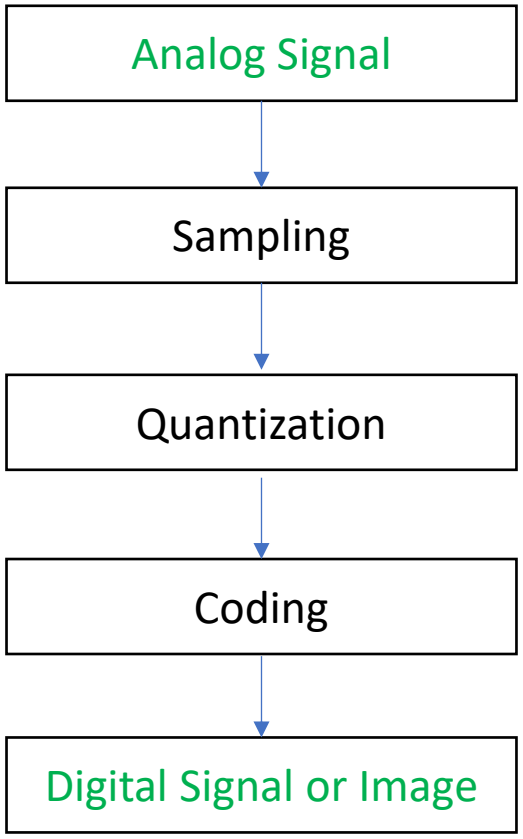
Y

Illumination of scene : Acquisition by Imaging System: Image Plane: Digital Image

Analog to Digital Image Conversion : Sampling and Quantization

The output of the sensor is continuous voltage waveform whose amplitude and time axis are continuous.

Steps



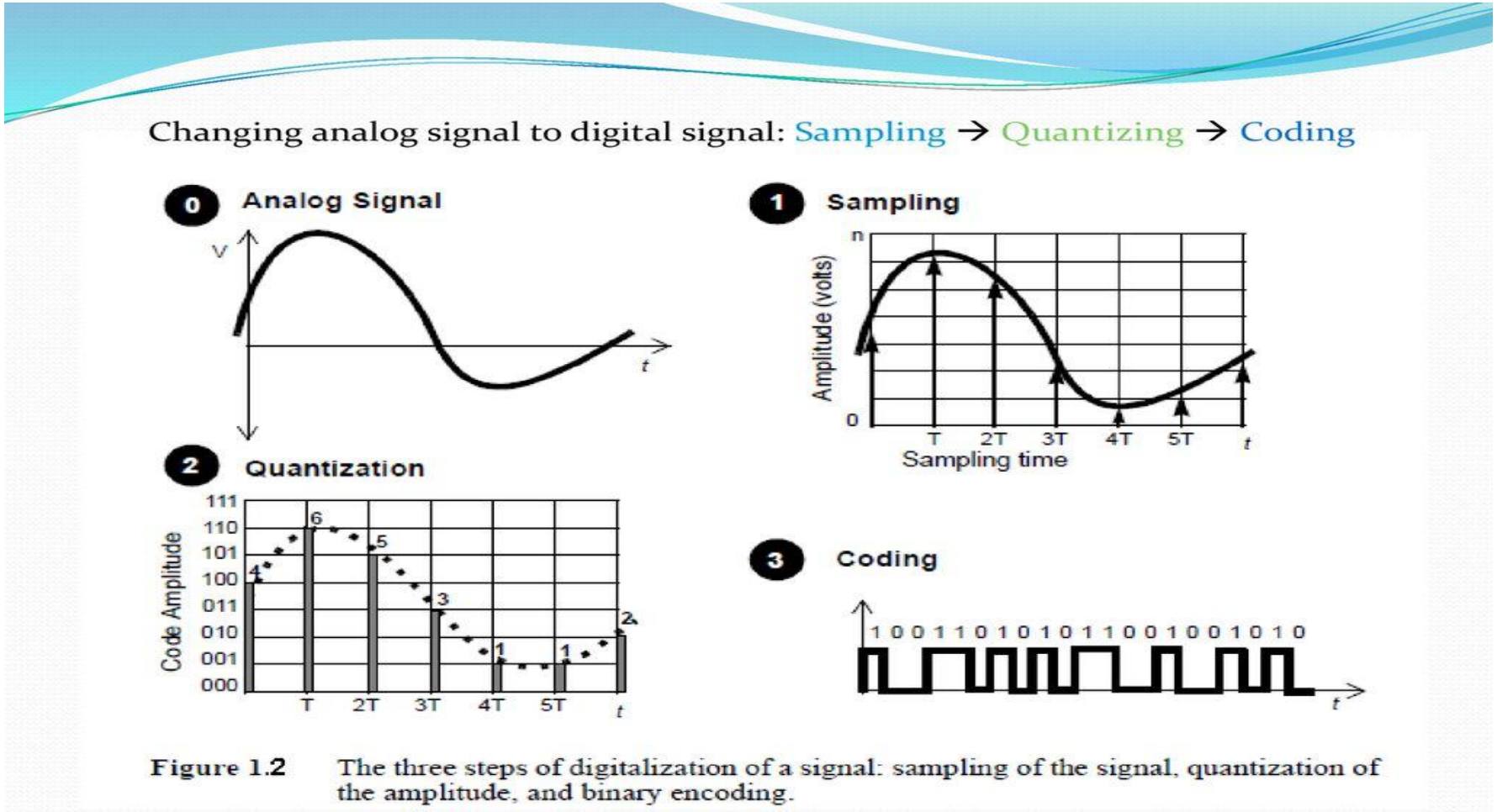
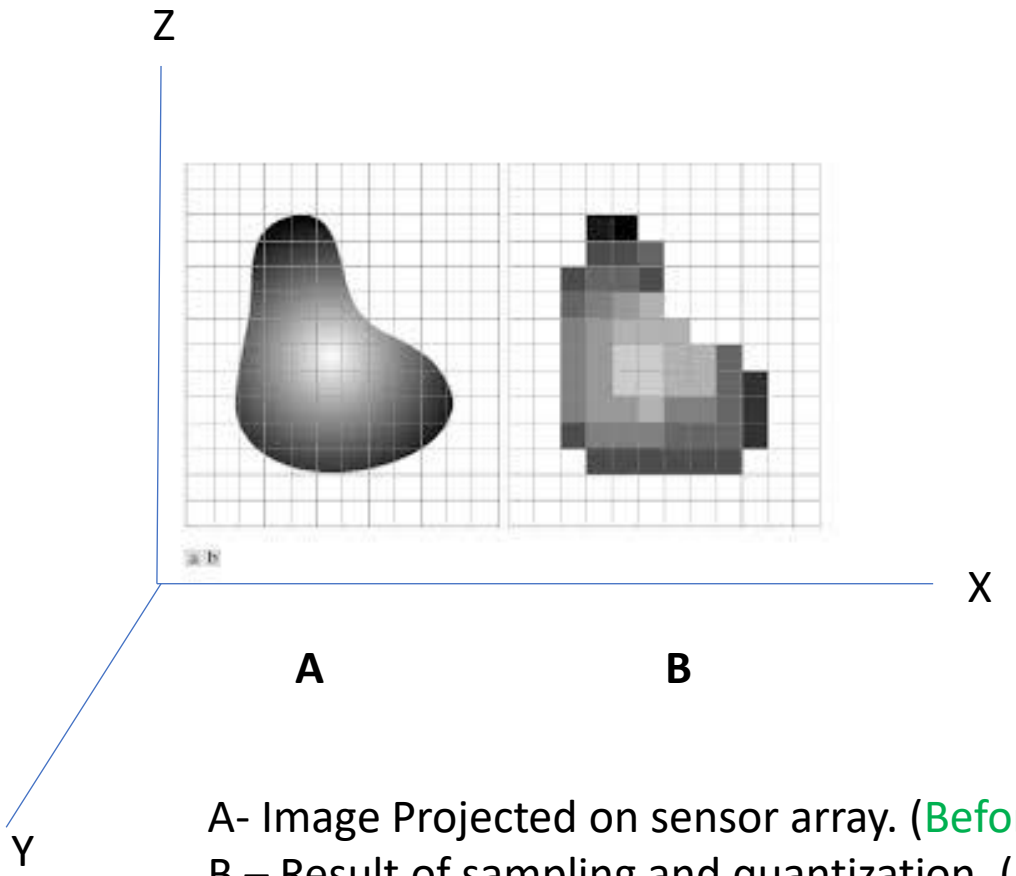


Figure 1.2 The three steps of digitalization of a signal: sampling of the signal, quantization of the amplitude, and binary encoding.

Image Sampling and Quantization



Sampling : Discretizing the time axis or coordinate.

Quantization : Discretizing the amplitude axis or coordinate.

Digital image : Sampling and Quantization

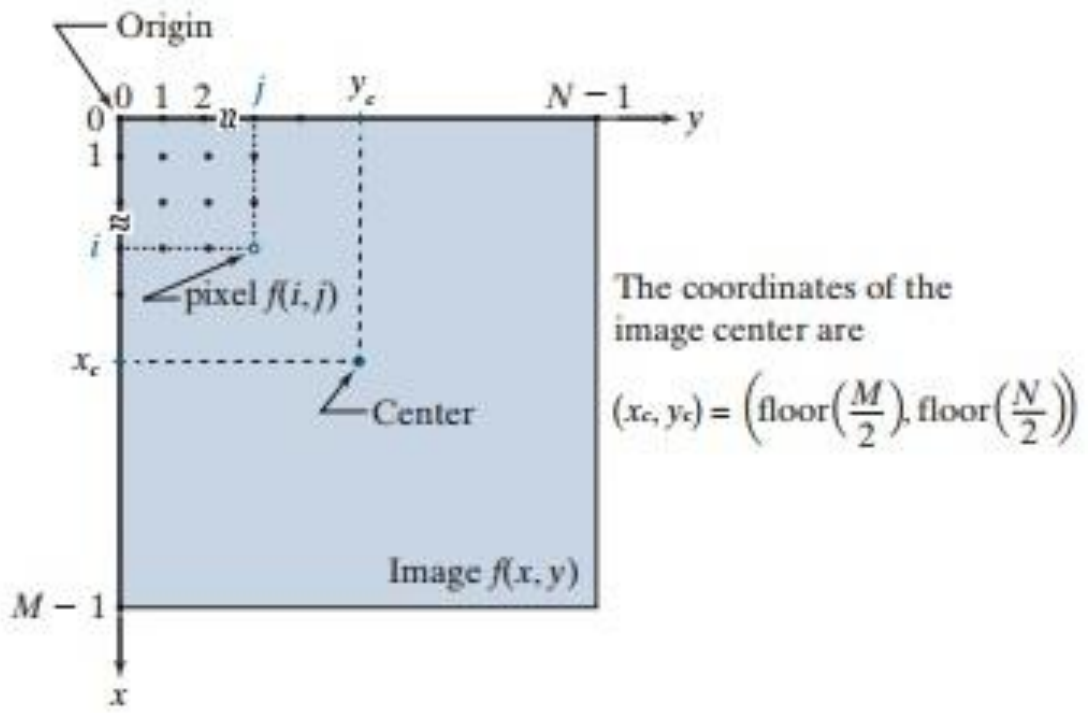
A- Image Projected on sensor array. (Before Sampling and Quantization)
B – Result of sampling and quantization. (After Sampling and Quantization)
X,Y are Coordinates.
Z is the amplitude.

Basic Terminologies used in Image Processing

- Digital Image and its type.
- Pixel, Intensity or Grey level.
- Spatial Domain.
- Image Storage and Grey Levels.
- Dynamic Range, Contrast.
- Dots per inch (DPI), Resolution.
- Spatial Resolution: Coarse and Fine Resolution.
- Intensity Resolution: Coarse and Fine Resolution.
- Basic Relationship Between Pixels : Four, Diagonal and Eight Neighbors.
- Basic Relationship Between Pixels : Adjacency, connectivity, region and boundaries.

Basic Terminology: Digital Image and its type

Digital Image: It's a two dimensional function f(x , y) where x, y are the spatial coordinate and the amplitude at that particular coordinate will be the intensity or grey level.



NOTE: Center of an Image with dimension M x N is obtained by Dividing the M and N by 2 and rounding it off to the nearest integer.

Basic Terminology: Digital Image and its type

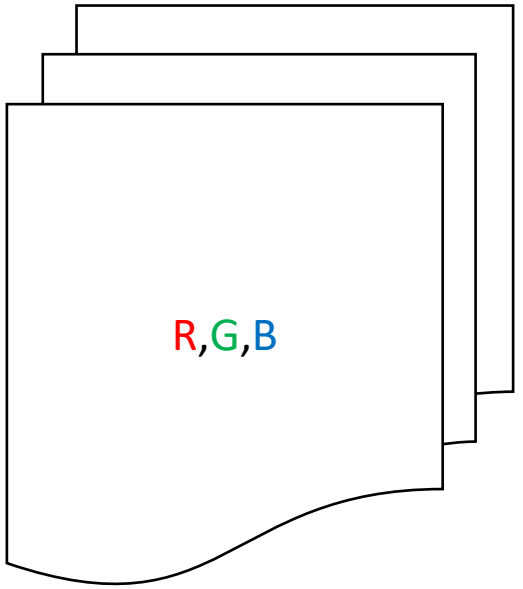
Basically, Three different types of images: Black and White, Grey Scale Image and Color Image

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

Black and White Image
Total Channel or band = 1
Pixel value varies between 0 to 1
(i.e. Range 0 to 1)

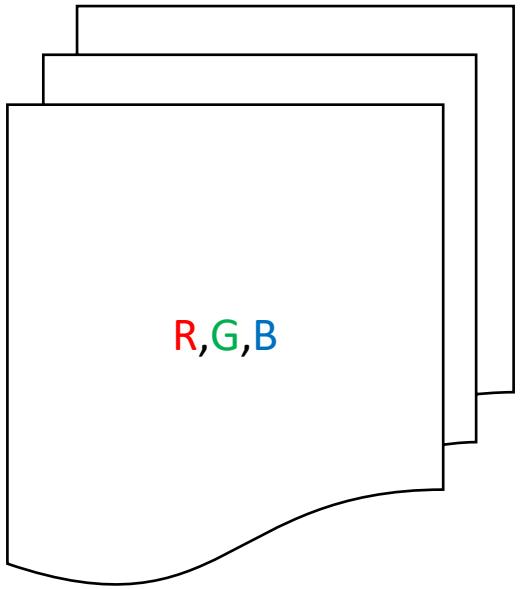
0	0	0	0	0
0	255	255	255	0
0	0	0	255	0
0	255	255	255	0
0	0	0	255	0
0	255	255	255	0
0	0	0	0	0

Gray Scale Image
Total channel or band = 1
Pixel value varies between 0 to 255
(i.e. Range 0 to 255)



Colour Image
Total Channel or Band = 3
Red Channel Range 0 to 255
Green Channel Range 0 to 255
Blue Channel Range 0 to 255

Basic Terminology: Conversion from color image to grey, BW image



Colour Image
Total Channel or Band = 3
Red Channel Range 0 to 255
Green Channel Range 0 to 255
Blue Channel Range 0 to 255

Luminance Method

This method assign different weights to RGB component. Human eyes are more sensitive to green light than red and less sensitive to blue light.

Gray Scale value = R x 0.299 + G x 0.587 + B X 0.114

For example – consider a colour pixel : (255,0,0)

Grey scale value – (0.299 x 255) + (0.587 x 0) + (0.114 x 0) = approx. 76

Grey Scale to Binary Image using thresholding
Black and white (Binary Image 0 to 1) Conversion from grey scale depends on the threshold value – For example consider threshold value 128.

```
If T <= 127
    BW = 0
Else:
     BW = 1
```

Basic Terminologies : Pixel, Intensity or Grey Level

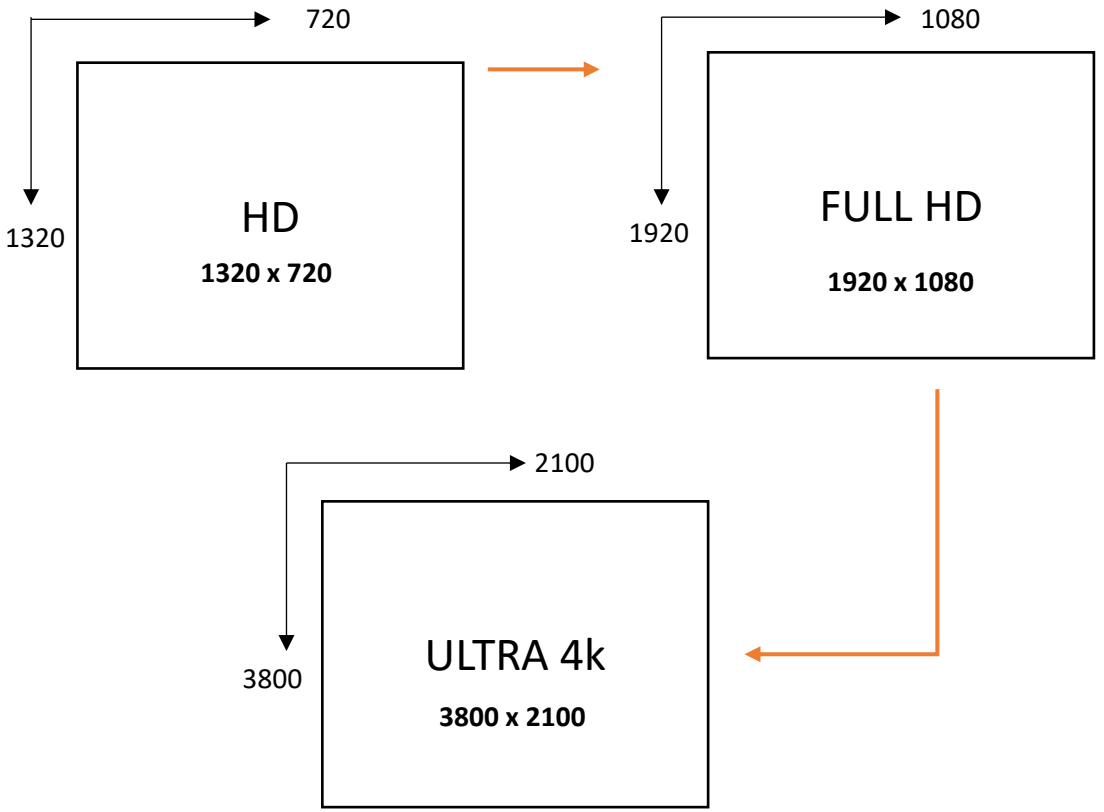
- **Pixel (dots) -** It's a smallest unit of a digital image.

0	0	0	0	0
0	255	255	255	0
0	0	0	255	0
0	255	255	255	0
0	0	0	255	0
0	255	255	255	0
0	0	0	0	0

F(4,4)

Image is a two dimensional function f(x , y). where x, y are the spatial coordinate and the **amplitude at that particular coordinate will be the intensity value or grey level.**

Number of pixels are increasing as we are moving from HD to Full HD to Ultra 4K. (Moving from coarse to fine resolution.)



Basic Terminologies: Spatial Domain and Intensity Levels

- **Spatial Domain**

Spatial means Location.
An image made by pixels. Each pixel has some intensity values.
A domain, where intensity or magnitude values are reflected based on the location.

- **Image Storage and Intensity levels**

Lets say size of the image is M x N and its a 8-bit image. Total bits required = M x N x k.

Example: M = 2048, N = 2048 and k = 16 bit image

Total no of bits = 2048 x 2048 x 16 = 67108864 bits
67108864/8 = 8388608 bytes (1 byte = 8 bits)
8388608/1024 = 8192 Kbytes (1kb = 1024 bytes)

No. of intensity levels = 2^k.
Example
if 8 bit image then 2^8 = 256 intensity levels or Grey levels (0 to 255).
if 16 bit image then 2^16 = 65536 intensity levels or Grey levels (0 to 65535).

Basic Terminologies: Dynamic Range, DPI and Resolution

- **Dynamic Range, Contrast**

Dynamic range in terms of images or image contrast : Difference between highest and lowest intensity levels in an image. **Example** For 3 bit image total intensity level 2^3 = 8 and its dynamic range is 7 (0 to 7 range.)

High dynamic range = Bright or Clear image.
Low dynamic range = Image dull

- **Dots per inch (DPI), Resolution**

Resolution directly refer to the clarity of the image or how much details clearly observed by the user.

If resolution is high,

more pixels or more dots per inch.

it means more information or details **CAN** be identified in better way.

else:

resolution is low.
less pixels or less dots per inch.
details **CANNOT** be identified in better way

Basic Terminologies: Spatial Resolution

Coarse resolution

Coarse resolution but better than 30m.

Fine resolution (better than 30m and 15m.)

- **Spatial Resolution**

Spatial Resolution: Capability of sensor to distinguish between two closely spaced objects.

Higher or fine Spatial Resolution: **Pixel size is small** and one can see more details.

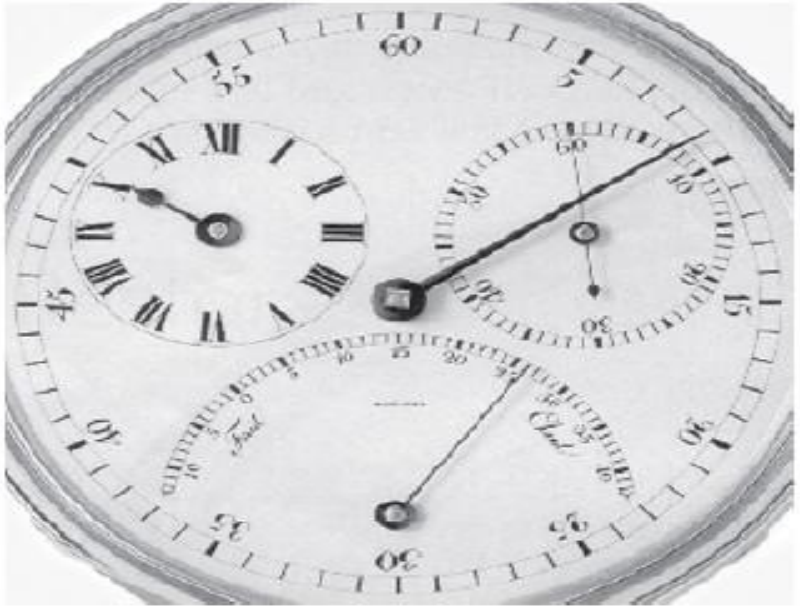
Lower or coarse Spatial Resolution: **Pixel size is big** and one can not distinguish between two closely spaced objects.

Basic Terminologies: Spatial Resolution: From 930 dpi to 72 dpi

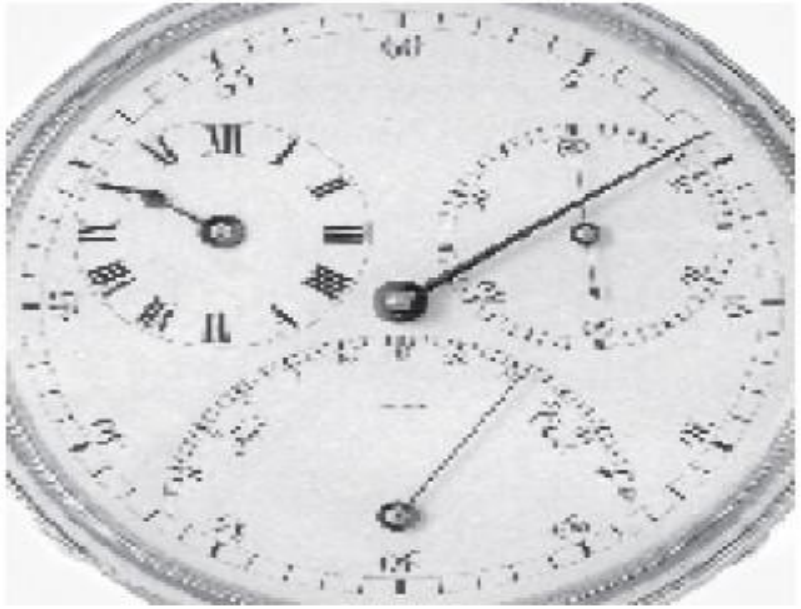
930 dpi

300 dpi

150 dpi



72 dpi



Basic Terminologies: Intensity Resolution

- **Intensity Resolution**

Capability to resolve different intensity or brightness levels or color in color image.

High or fine Intensity Resolution:

Ability to capture wide range of brightness or intensity levels.

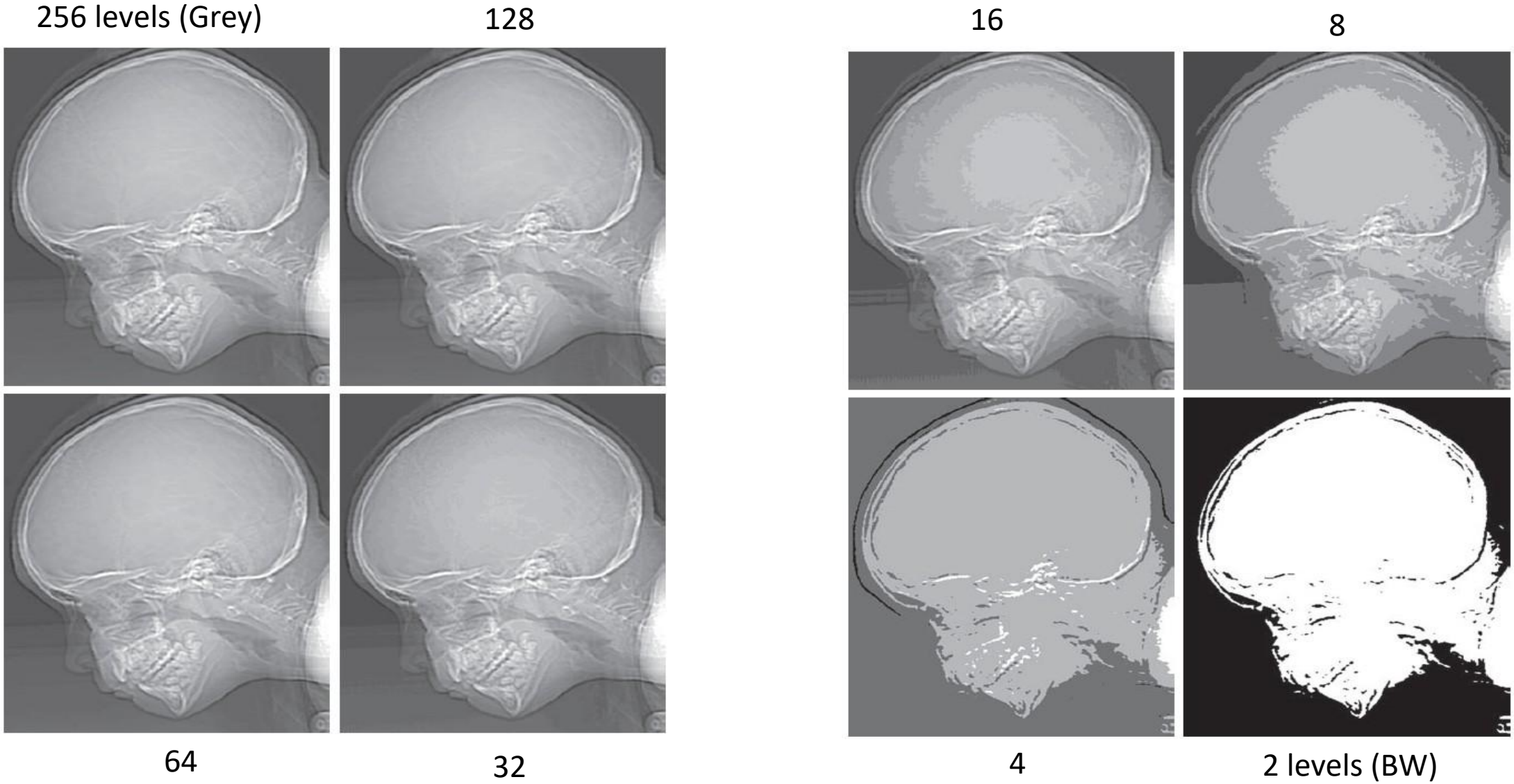
For 32 bit image 2^32 = 65536 intensity or grey levels.

Low or coarse Intensity Resolution:

Ability to capture small range of brightness or intensity levels.

For 8 bit image 2^8 = 256 intensity or grey levels.

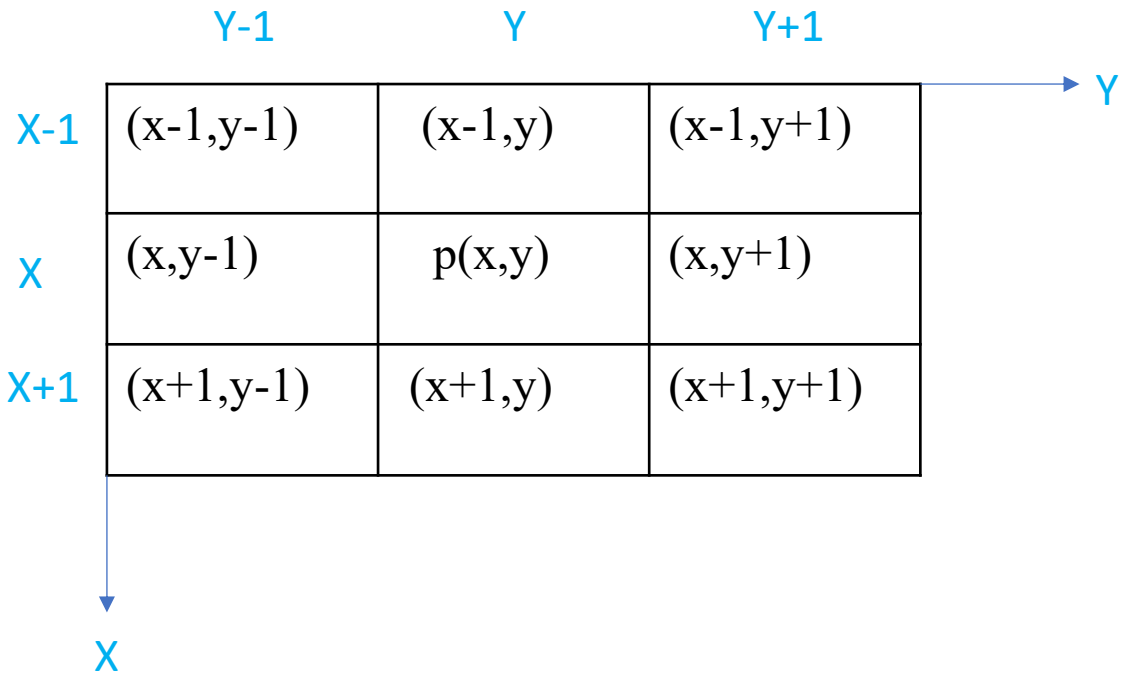
Basic Terminologies: Intensity Resolution (From 256 to 2 Levels)



Obs: As the number of levels are decreasing from 256 to 2 then details inside the image is decreasing.

Basic Terminology: Four Neighbors, Diagonal and 8 Neighbors.

Basic Relationship between pixels with respect to central pixel P(x,y).



Four Neighbors of P, $N_4(P)$

	(x-1,y)	
(x,y-1)	p(x,y)	(x,y+1)
	(x+1,y)	

Diagonal Neighbors of P,$N_D(P)$

(x-1,y-1)		(x-1,y+1)
	p(x,y)	
(x+1,y-1)		(x+1,y+1)

8 Neighbors of P $N_8(p) = N_4(P) + N_D(P)$

(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	p(x,y)	(x,y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

Basic Terminology: Adjacency, Connectivity, Region and Boundaries

Adjacency: It usually defined as spatial arrangement of pixel around the central pixel.

Three types of adjacency: 4 adjacency, 8 adjacency and mixed adjacency

4- Adjacency

If a pixel q has a value from set V and is one of these 4-neighbors of p, then p and q are 4-adjacent.

	(x-1,y)	
(x,y-1)	p(x,y)	(x,y+1)
	(x+1,y)	

Set v = {1}

Binary Image

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

8- Adjacency

If a pixel q has a value from set V and is one of these 8-neighbors of p, then p and q are 8-adjacent.

(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	p(x,y)	(x+1,y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

Set v = {1,2,3,4,5,6,7,8,9,10}

Grey Scale Image

54	10	100	8
81	150	2	34
201	200	3	45
7	70	147	56

Note: Red – pixel selected, Green – 4 adjacency, Blue – 8 adjacency, **Black** – No adjacency

Basic Terminology: Adjacency, Connectivity, Region and Boundaries

Mixed Adjacency or m – Adjacency: Motivation: To avoid the ambiguity of path or to remove the duplicate connection

Two pixels p and q with values from a set V (e.g., V=1 for foreground pixels in a binary image) are m-adjacent

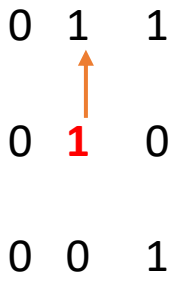
if:

1.q is in N_4(p) (i.e., q is a 4-neighbor of p),

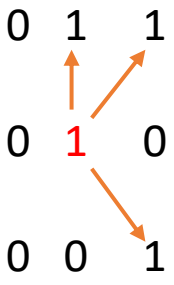
OR

1.q is in N_D(p) (i.e., q is a diagonal neighbor of p) AND the set N_4(p) Intersection N_4(q) (the intersection of their 4-neighbors) has **no pixels whose values are from V**

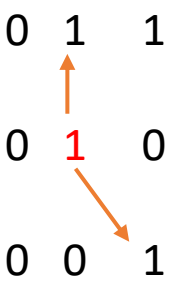
Example - Set v = {1}: **Below adjacencies are defined based on central pixel (2,2)**



4- Adjacency (1,2)



8- Adjacency (1,2), (1,3) and (3,3)



Mix- Adjacency (1,2) and (3,3)

Central pixel 1 (2,2) primarily connected With (1,2) and diagonally connected with (3,3).

Why (1,3) is not taken in m –adjacency ?

As (2,2) is connected with (1,2) and (1,2) is directly connected with (1,3).

So, to avoid the multiple path (1,3) is not included.

Basic Terminology: Adjacency, Connectivity, Region and Boundaries

Connectivity: It is established when a path exists between two pixels. This path is formed by a sequence of pixels where each successive pixel is adjacent to the previous one, and all pixels in the path share a common property (e.g., same intensity value, same object label).

Region: A region is a collection of pixels that are all connected to each other based on a chosen connectivity rule and share a common characteristic.

Boundary: The boundary of a region is identified by looking at the adjacency of pixels within the region to pixels outside the region.

Basic Terminology: Adjacency, Connectivity, Region and Boundaries

Example -

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

Lets evaluate 4 pixels at the position of (2,2), (2,3), (3,2) and (3,3).

Connectivity: All these 4 pixels shows 4 adjacency, 8 adjacency and m adjacency

Region: Given that all '1' pixels are connected (by any common definition of connectivity for this simple block), they form a single **region**.

Region R: (2,2),(2,3),(3,2),(3,3)

Basic Terminology: Adjacency, Connectivity, Region and Boundaries

Boundary: Boundary pixels are those pixels that have at least one 0 at neighbour

Using 4-adjacency for boundary definition:

- P_1=(2,2): Neighbors (2,1),(1,2) are '0'. So P_1 is a boundary pixel.
- P_2=(2,3): Neighbors (1,3),(2,4) are '0'. So P_2 is a boundary pixel.
- P_3=(3,2): Neighbors (3,1),(4,2) are '0'. So P_3 is a boundary pixel.
- P_4=(3,3): Neighbors (3,4),(4,3) are '0'. So P_4 is a boundary pixel.

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

Using 8-adjacency for boundary definition:

For P_1=(2,2): Neighbors (2,1),(1,2),(1,1) are '0'. So P_1 is a boundary pixel.

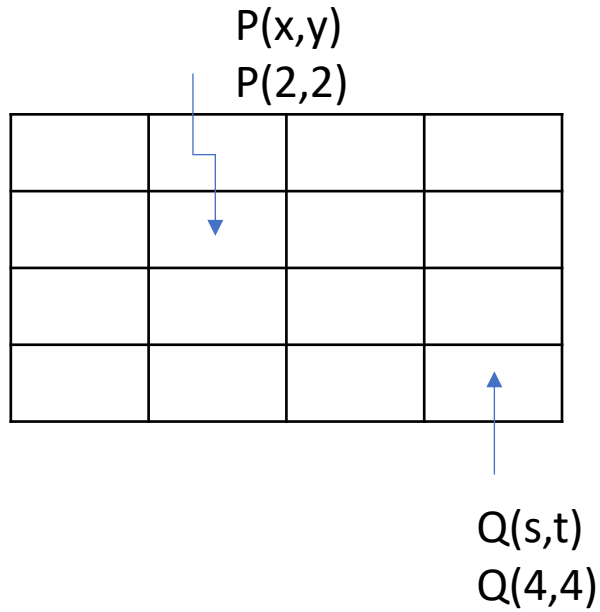
Similarly, all four '1' pixels have at least one '0' in their 8-neighborhood.

Boundary pixels (8-adjacency): All four '1' pixels are the boundary pixels.

Results : Based on above, all the four pixels are the boundary pixels.

Basic Terminology: Distance Measures

Distance Measures: In Image processing it's a basic tool to find the similarity and dissimilarity between different pixels, points or feature vectors.



Three basic distance measures.

1.Euclidean Distance: Straight line distance between two points in a Euclidean space. (2D image, 3D volume).

$$\text{Sqrt}[(x-s)^2 + (y-t)^2]$$

2.City Block Distance or Manhattan distance or L1 Distance
It calculate distance between two points by summing the absolute difference of their coordinate.

$$|x-s|+|y-t|$$

3.Chess Board or Chebyshev Distance : It calculate distance between two points by considering the maximum of absolute difference of their coordinate.

$$\text{Max}\{|x-s|,|y-t|\}$$

Basic Terminology: Distance Measures

Euclidean Distance

2.828	2.236	1.414	2.236	2.828
2.236	1.414	1	1.414	2.236
1.414	1	0	1	1.414
2.236	1.414	1	1.414	2.236
2.828	2.236	1.414	2.236	2.828

Points are equidistance from the center and form a **circle**.

Application : Image feature matching and clustering.

City Block Distance

4	3	**2**	3	4
3	**2**	1	**2**	3
2	1	0	1	**2**
3	**2**	1	**2**	3
4	3	**2**	3	4

Points are equidistance from the center and form a **diamond**.

Application: Pathfinding on grid - based systems. Used in Robotics for navigation purpose.

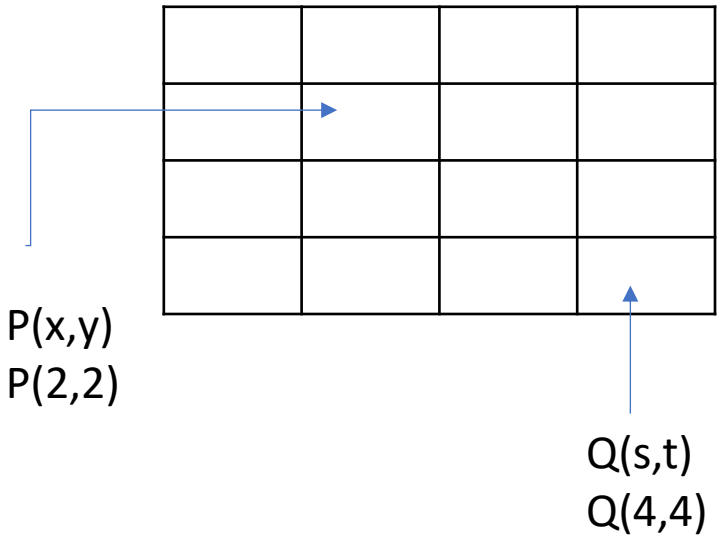
Chess Board Distance

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

Points are equidistance from the center and form a **square**.

Application: Defining neighborhood, dilation/ erosion operations..

Basic Terminology: Distance Measures



1.Euclidean Distance:

$[(x-s)^2 + (y-t)^2]1/2$

2.City Block Distance:

$|x-s|+|y-t|$

3.Chess Board Distance:

$Max\{|x-s|,|y-t|\}$

Example: Calculate the distance ?

		P(3,3)	Q1(3,4)	Q2(3,5)
			Q3(4,4)	
				Q4(5,5)

Find the distance between
(P,Q1)
(P,Q2)
(P,Q3)
(P,Q4)

CSET340

Advanced Computer Vision and Video Analytics

Module 1

12th Jan. to 16th Jan. 2026

Overall Course Coordinator-
Dr. Gaurav Kumar Dashondhi
Gaurav.dashondhi@bennett.edu.in

Note : Any query related to course then first connect with overall course coordinator.

Computer Vision

Human Eye

1. Vision is our most powerful sense. It allows us to understand the physical world, without directly making any physical contact.

2. 60 % of your brain in one way or other way involve in the process of visual perception.

3. Vision is so powerful that we can navigate in this complex world seamlessly.

4. **If our vision system is so powerful than why we need to build machines which can emulate the same task ?**

Computer Vision (CV)

Human Eye

If our vision system is so powerful than why we need to build machines which can emulate the same Task ?

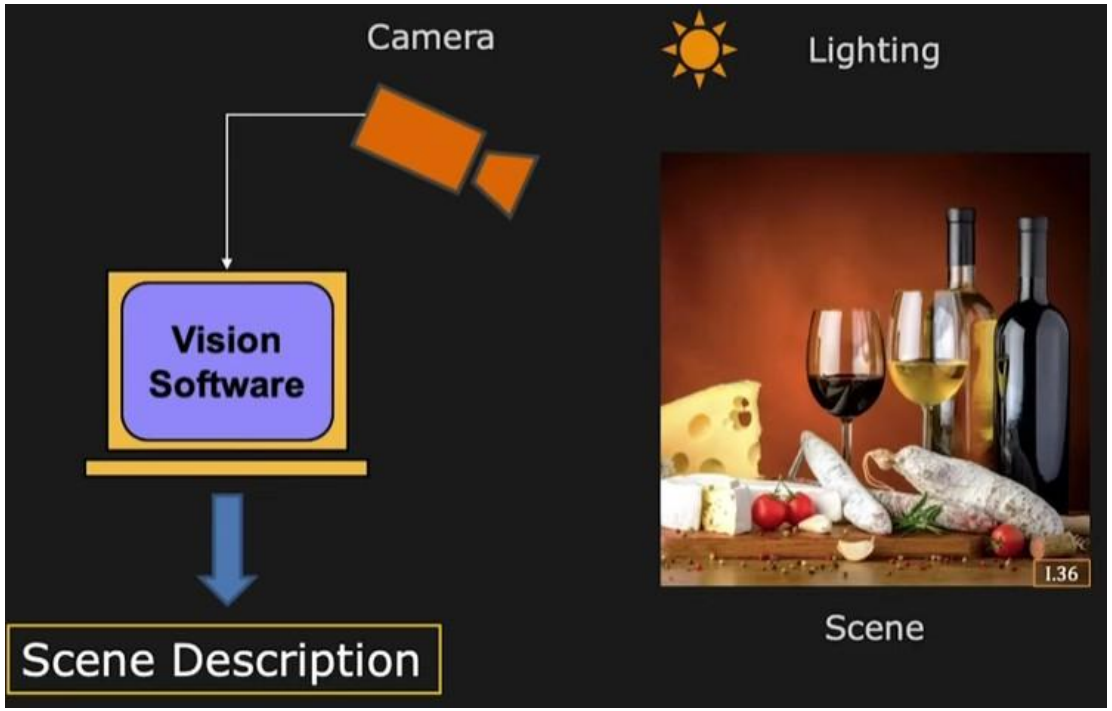
Ans-

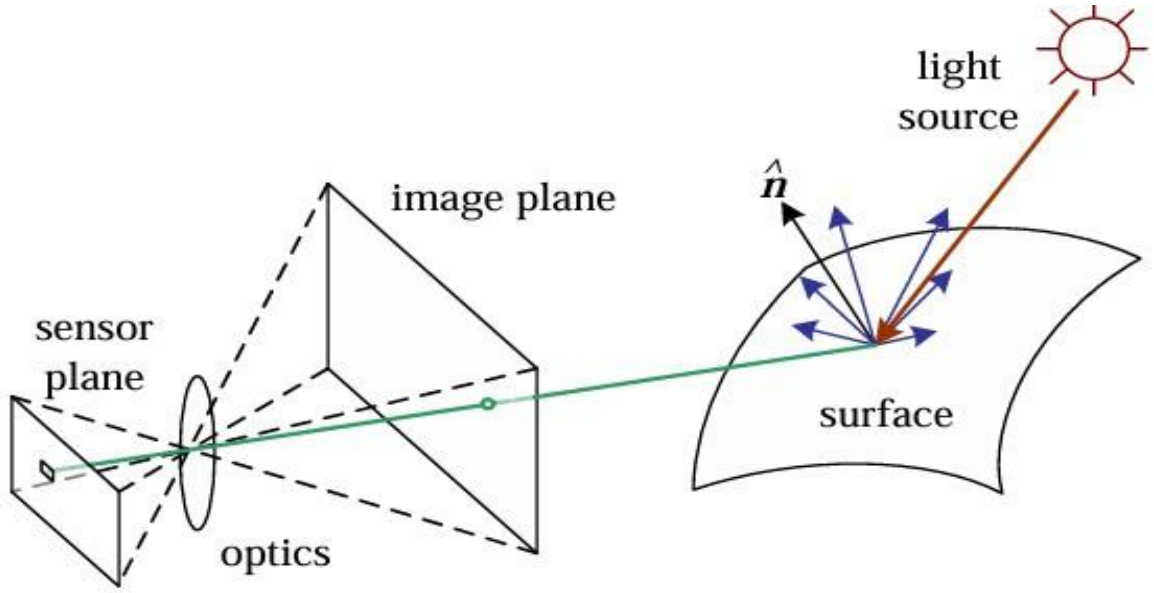
1. There are so many day to day task like -
   1.1 Driving car,
   1.2 Tiding of work.
   1.3 Some household work that machine can do. so that, one can involve himself/herself in more rewarding work.

2. Even though our vision system is so powerful. It is not good at the precise measurement of the object in the physical world.

3. A CV system should surpass the capability of human vision system that human vision system could not able to do.

Computer Vision





1. Camera or vision system takes lights from the 3D scene and pass through a vision software and goal of vision software is to provide symbolic description of the items. Like bread, cheese and two wine bottles.

2. More Sophisticated CV system tells about the

   - Freshness of the bread.
   - Quality of cheese and other items.

Computer Vision: Definition

Vision is

... automating human visual processes

... an information processing task

... inverting image formation

... inverse graphics

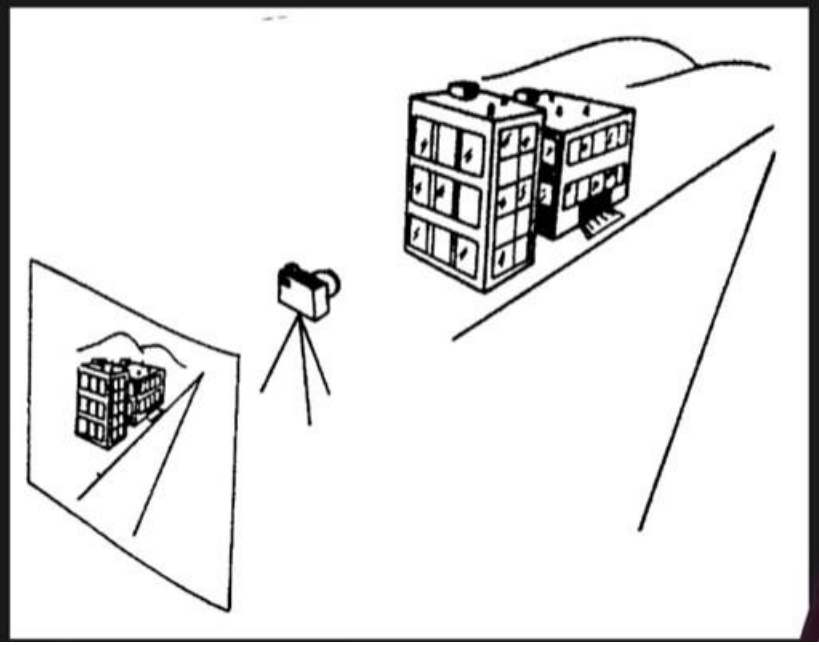
... really useful!

Vision deal with the image.

An Image is an Array of Pixels

A Pixel has Values:

- Brightness
- Color
- Distance
- Material
- ...



A pixel can disclose about brightness, color and even distance between two Objects.

In the upcoming years CV system becomes more sophisticated they can even comment on the material of the object.

Computer Vision

Images are interesting

Simply look at the image –

- Two boys are taking bath.
- What are the different vegetation around them.
- 3-dimensional structure of the scene.
- Morning time of the day.
- Even though you can understand playful mood in the scene.

Digital equivalent of the same scene

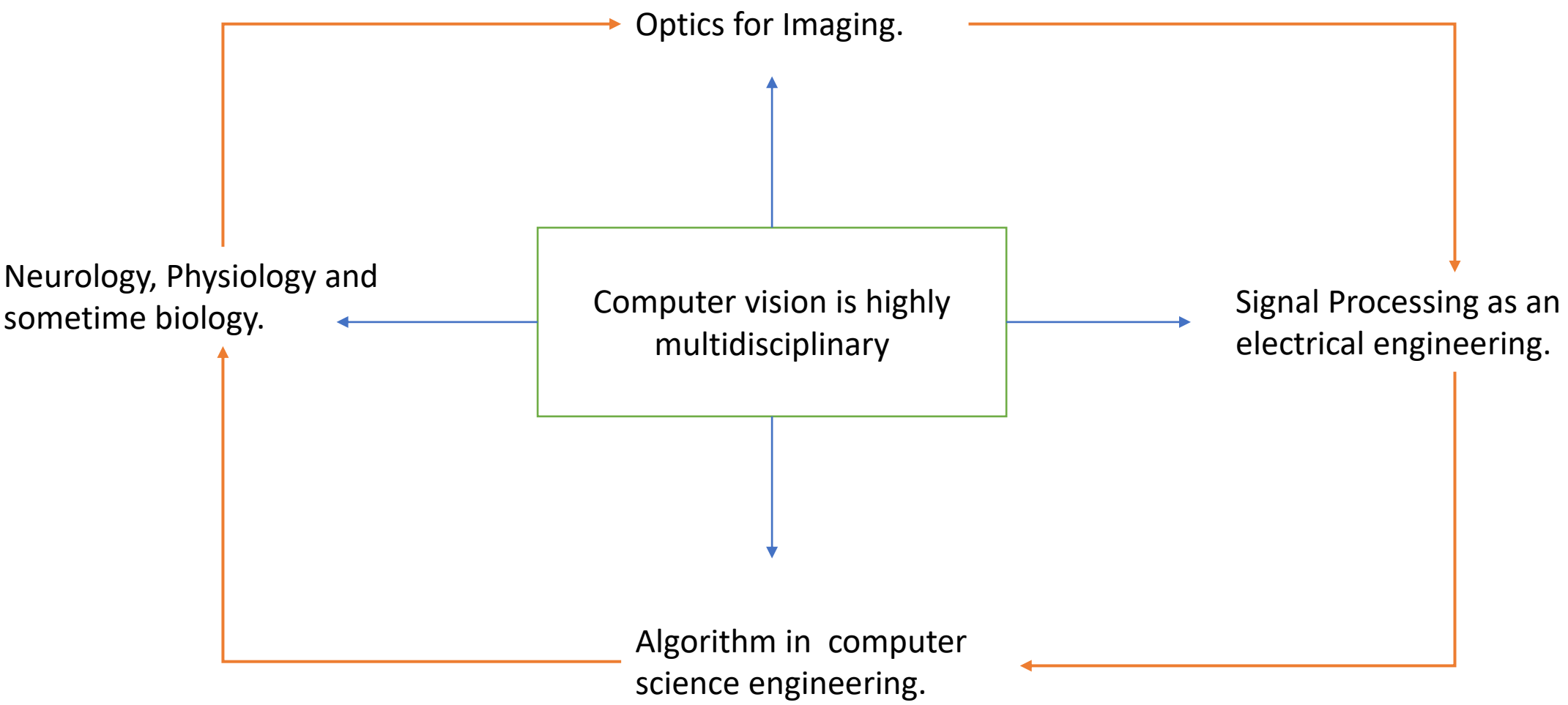
157	159	159	104	104	115	128	131	133	133	132	131	132	130	129	118	132	158	156	153	190	144	117	126	120	81
159	165	153	101	103	113	126	129	130	130	126	124	127	128	127	120	122	158	159	154	160	190	121	118	67	47
162	154	154	98	101	114	124	127	130	132	144	159	155	132	123	119	119	148	154	150	140	185	161	60	48	45
141	132	158	93	98	110	121	125	122	129	143	172	191	188	143	105	117	148	140	145	142	153	105	44	49	71
100	130	157	93	99	110	120	116	116	129	138	163	191	205	211	130	107	153	98	133	147	107	44	47	81	151
87	130	157	92	97	109	124	111	123	134	139	175	194	201	207	205	126	151	74	114	160	57	49	63	141	163
93	131	159	92	98	112	132	108	123	133	162	180	183	192	196	205	184	151	138	199	195	54	47	119	161	156
96	134	164	95	97	113	147	108	125	142	156	171	173	178	184	181	186	191	206	203	161	44	84	158	159	155
95	137	165	95	95	111	168	122	130	137	145	139	144	139	145	179	193	203	194	158	95	49	135	160	157	155
101	139	166	94	96	104	172	130	126	130	108	77	85	80	153	191	188	161	144	113	48	83	161	160	156	153
101	133	167	94	96	100	154	137	123	92	67	57	72	153	182	184	175	101	116	53	48	119	166	163	159	152
99	130	169	97	99	109	131	128	84	55	60	75	149	176	170	194	209	99	79	51	67	150	158	155	154	151
97	129	170	97	98	118	122	94	66	56	56	140	161	114	136	187	163	81	85	52	98	161	159	154	148	137
92	123	173	101	98	129	95	74	74	45	94	174	106	115	126	168	108	60	92	55	128	157	153	148	145	157
81	115	175	104	116	87	78	69	84	56	140	124	158	170	143	173	150	76	90	68	148	153	146	148	186	196
69	108	172	107	103	87	82	54	83	105	93	107	153	166	132	162	153	68	87	97	157	149	141	179	204	206
71	119	172	106	91	78	97	70	99	104	59	116	142	153	141	165	123	55	84	132	154	146	148	199	209	210
61	126	175	112	83	74	92	123	130	53	61	108	137	132	138	154	77	58	82	150	152	143	155	210	211	213
53	128	175	105	71	82	109	127	75	50	57	74	115	139	151	117	47	67	89	154	154	143	159	218	214	199
56	115	173	105	61	76	106	114	70	54	52	60	102	137	160	146	78	67	96	135	130	125	165	215	142	81
117	106	176	101	55	71	81	112	101	57	55	70	117	139	152	188	198	112	87	146	131	112	178	164	81	91
107	121	177	89	50	64	60	103	114	66	56	90	120	140	149	169	201	194	100	148	134	155	208	120	99	99

How much challenging CV is ?

Digital equivalent of the same scene is nothing but the array of the numbers.

From these array of numbers, understanding all the information embedded in the scene is challenging.

Computer Vision



Computer Vision: Applications

Factory Automation: Vision-Guided Robotics

Manufacturing is highly automated these days,
Like Cars are mostly built by vision guided robotics.

Factory Automation: Visual Inspection

Speed of manufacturing product is very high and at
the same time size of component or goods is very small.

Computer Vision : Applications

If your car is moving with high speed then by using OCR traffic system reads the number plate, and it will generate a ticket on your name.

Digitization of physical book.

Authentication of signature on the check.

Scanning envelops and packages for the postal services.

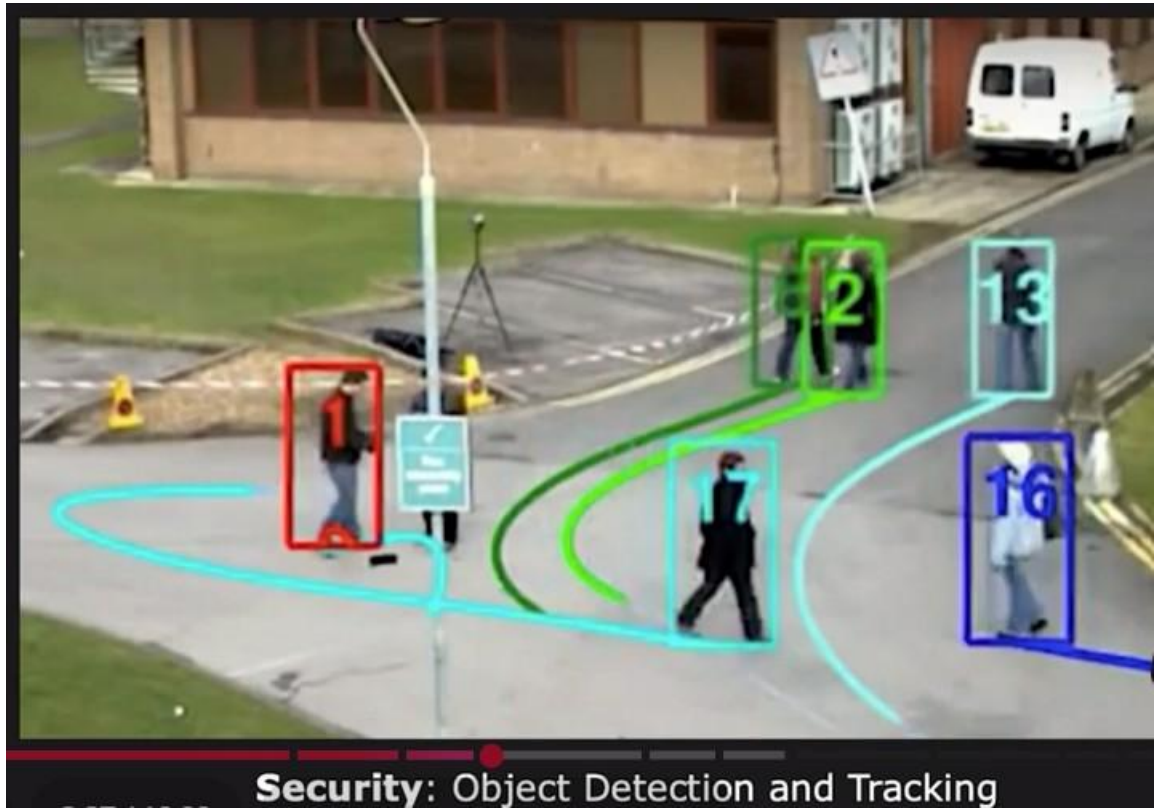
Computer Vision : Applications

Biometrics:
Iris in an eye has a particular kind of pattern.
It is found that, pattern inside the eye is exactly equal to your DNA. So, it can used exactly identify the person or may be used for access control.

Face Detection: organizing your photos or in the security purpose.

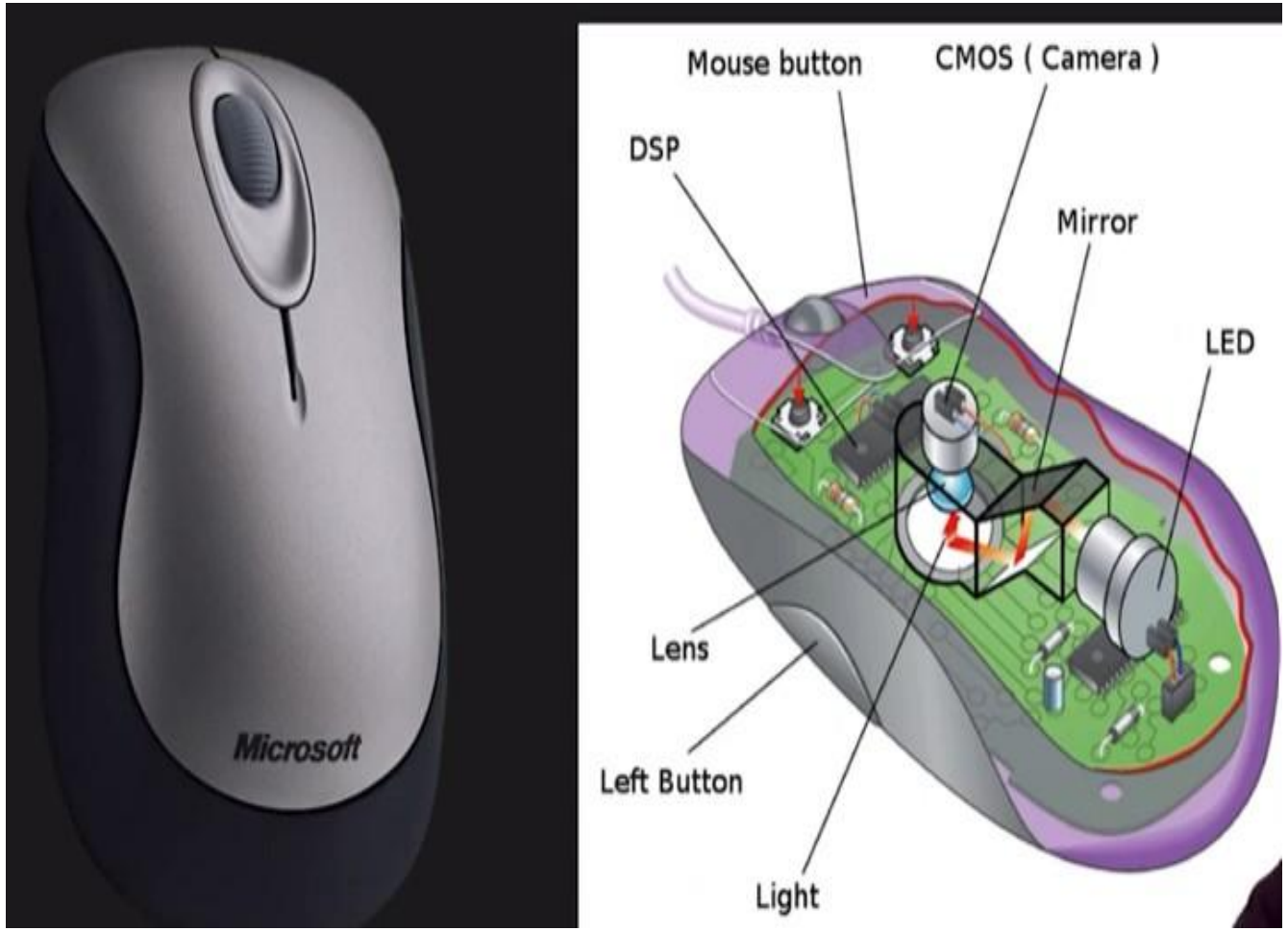
Computer Vision : Applications

Vending machine in Japan: Detect your gender and age and it will show the various product of your interest based on it.



Object Tracking: Track the person if he/she is occluded or partial visible. Now a days, if you leave the field of view of one camera then another camera will take over and track the person.

Computer Vision : Applications



Human Computer Interface: Optical Mouse

Inside the optical Mouse a complete computer vision system is integrated.

It has tiny camera which capture the image at very low resolution and with very high frame rate.

It has little lighting system which illuminate the surface on which mouse is sitting on and using the pattern this camera designed map that detects the motion of the mouse with respect to the surface that it sits on.

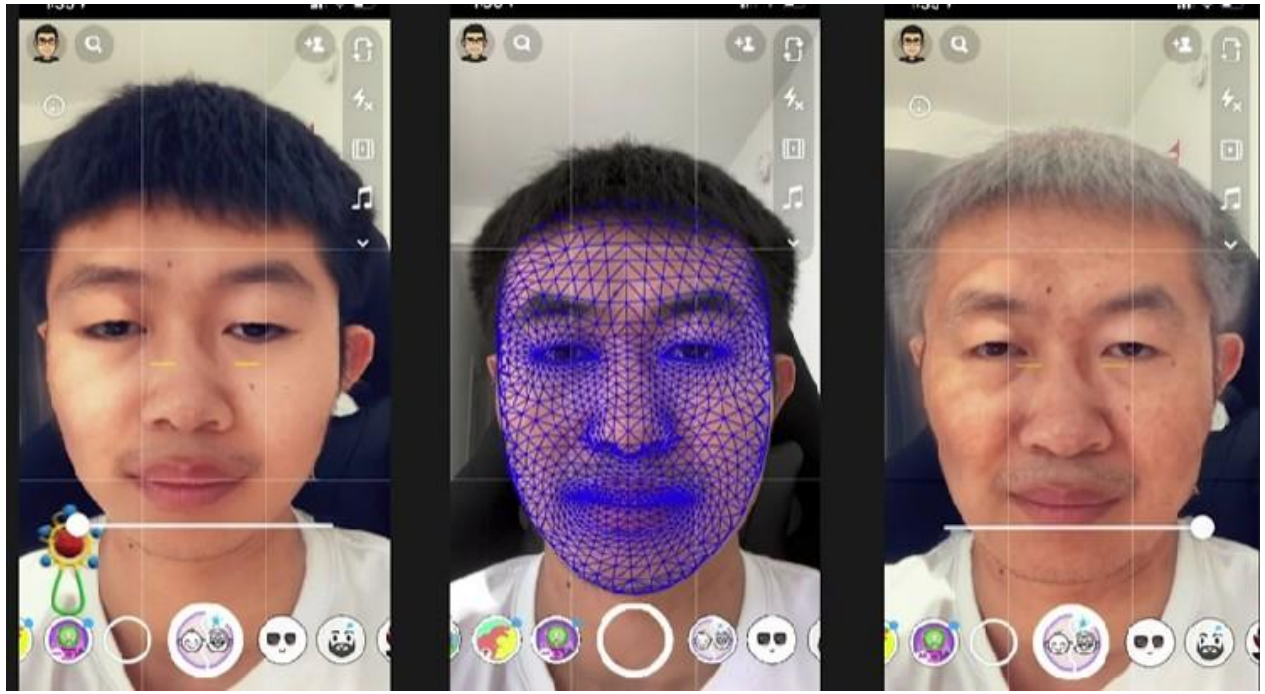
This information is passed to your computer to control the cursor on your screen or monitor.

Computer Vision : Applications

Gaming: Any gaming platform is packed with vision cameras.

Special effects: A camera is sitting on its top,
Like camera capture its expression and these expression used to form different virtual character.

Computer Vision : Applications

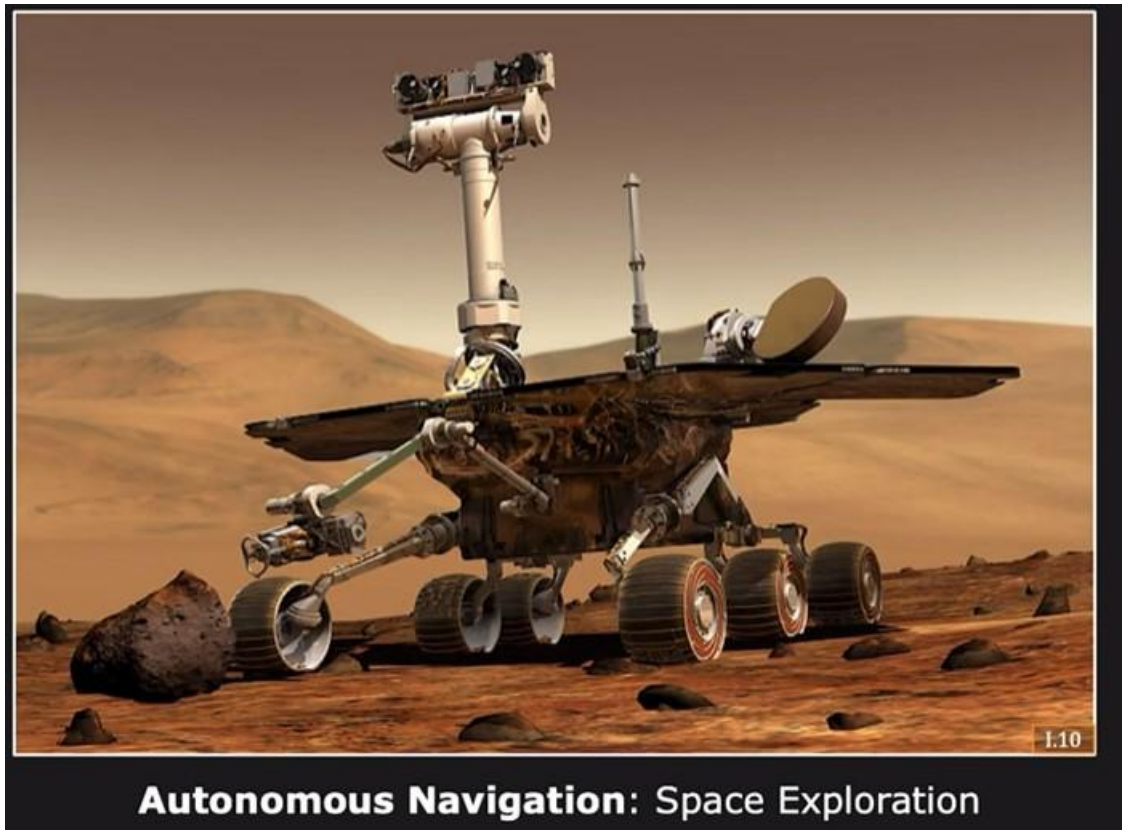


AR domain: Face Manipulation

It will extract the frame at very fast rate and able to map the face of a person.

Now based on mapping one can see the young and old version of the same person.

Computer Vision : Applications

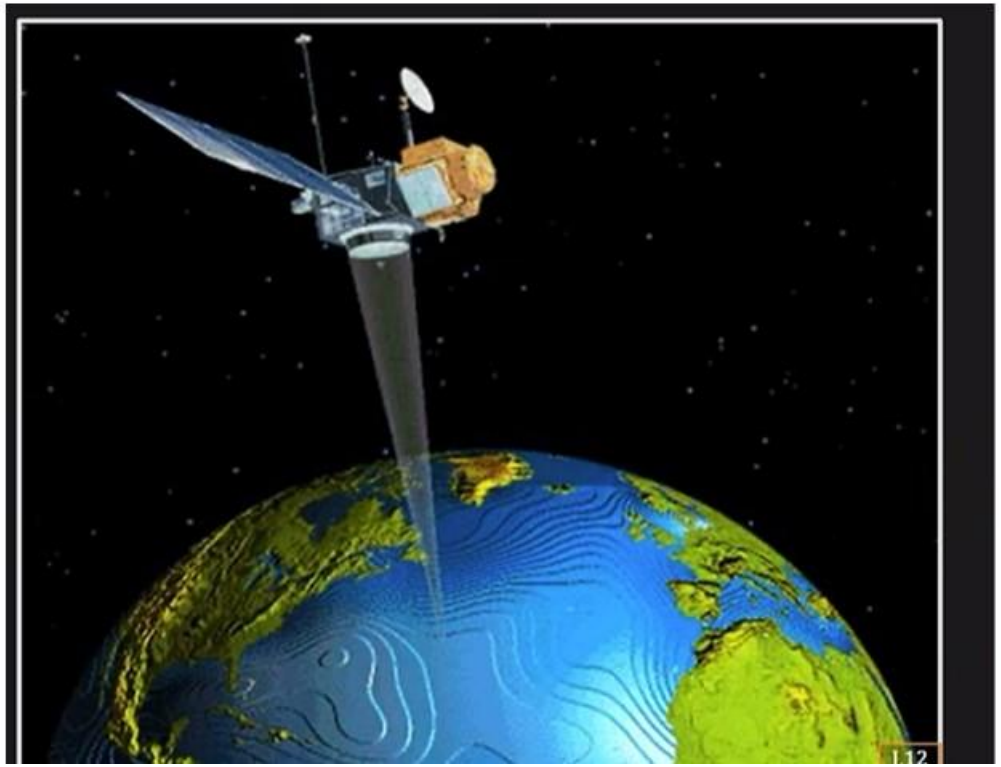


Visual Search : By internet or by your personal device like mobile. To get to know the information of it.

Autonomous Navigation and Exploration:
Mars Rover has various vision sensor.
CV algorithms needs to extract information and send this information to remote location.

Computer Vision : Applications

Driver less cars: These cars are packed with different sensors. visible light cameras, depth cameras and IR sensors.



Remote sensing.

Computer Vision: Consumer Applications

(a)

(b)

(c)

(d)

Image Stitching: Merging different views.

Exposure Bracketing: Merging different exposures.

Morphing : Blending between two photograph.

AR: Augmented Reality.

Basic Difference between Image Processing and Computer Vision

Image Processing :-

Focuses on manipulating and enhancing images to improve their quality or extract useful information.

- It is primarily about transforming images into a more desirable form.
- **Goal**: Enhance images, prepare data for further analysis, or extract low-level features.
- Examples: Noise reduction, Contrast enhancement, Histogram equalization, Image compression.

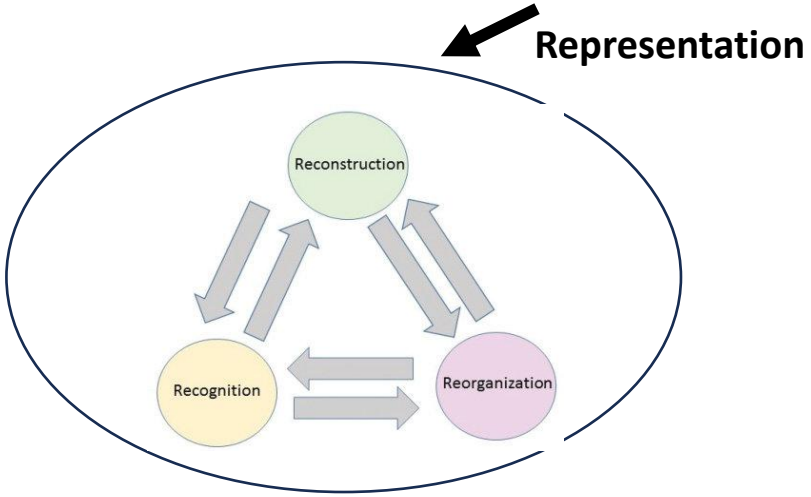
Computer Vision:-

Involves understanding and interpreting visual data to make decisions or automate tasks.

- It uses image processing as a preprocessing step but aims to achieve higher-level understanding.
- **Goal**: Mimic human vision to perform tasks like recognition, detection, and analysis.
- Example:- Object recognition, scene understanding, motion tracking, autonomous navigation.

Four Rs of Computer Vision

- **Recognition**
  - Identifying objects, patterns, or attributes within an image or video.
  - Example: Classifying an image as containing a car, a pedestrian, or a traffic signal.

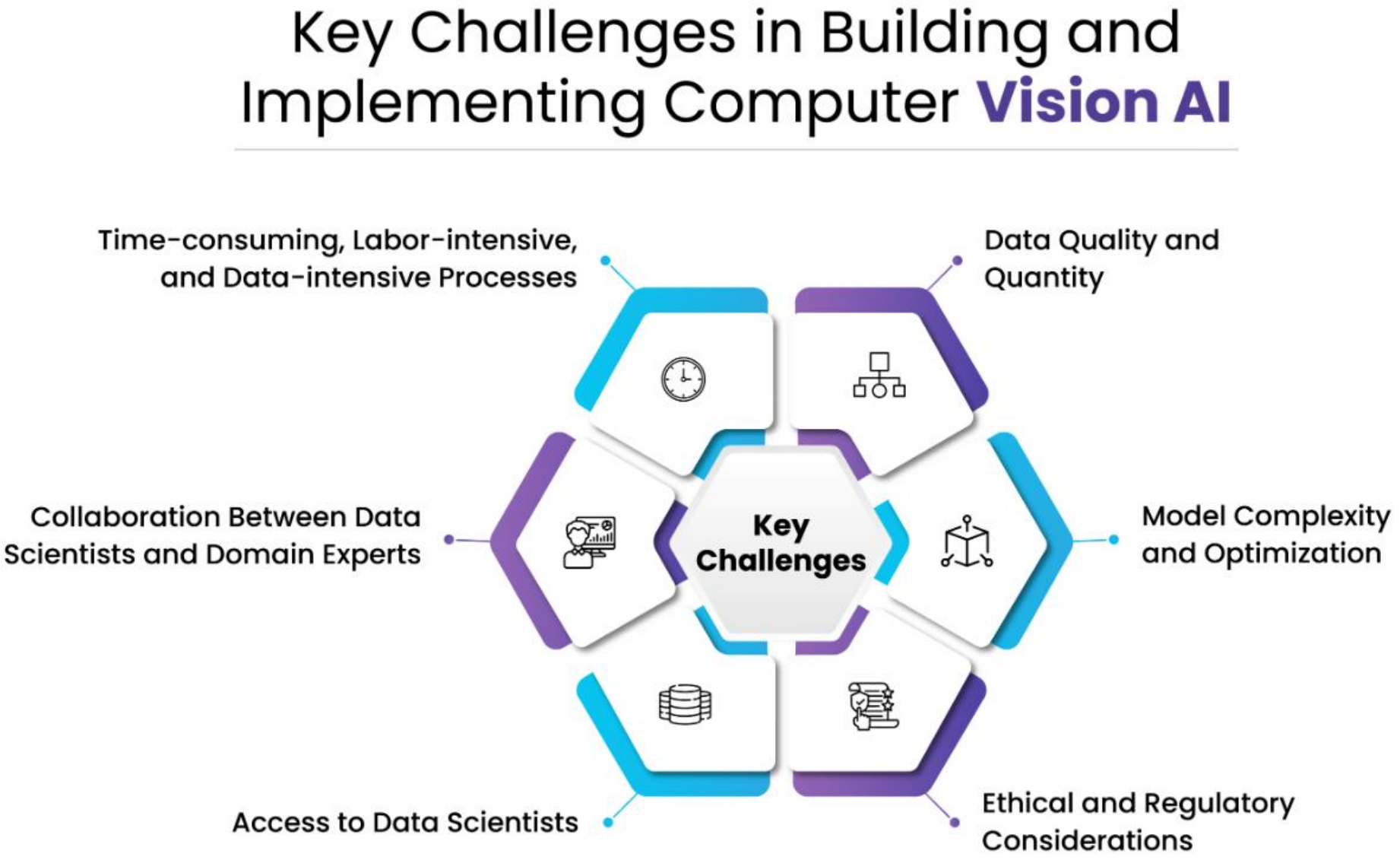


- **Reconstruction**
  - Rebuilding a 3D model or understanding the spatial structure from 2D data.
  - Example: Converting a series of 2D images into a 3D representation of a building.

- **Reorganization**
  - Rearranging or transforming visual data to better understand or represent it.
  - Example: Segmenting an image into meaningful regions (e.g., background, foreground, objects) or clustering features for analysis.

- **Representation**
  - Encoding visual data into formats or abstractions that are computationally efficient and meaningful for further tasks.
  - Example: Generating feature vectors for machine learning models or creating a semantic map of a scene.

Challenges in the Computer Vision

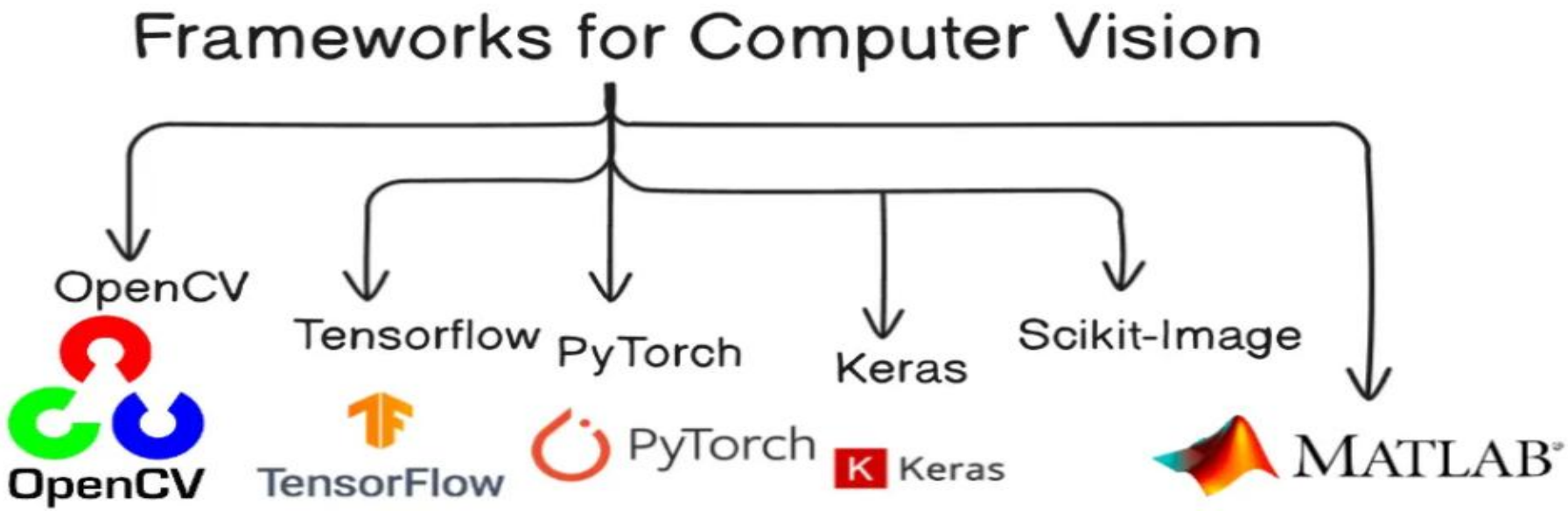


Computer Vision: Low level Vs High Level Processing

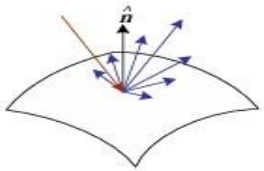
Feature	Low-Level Processing	High-Level Processing
Input Data	Raw Pixels (Intensity, Color)	Extracted Features, Segments, Models
Abstraction Level	**Low** - Local, quantitative, data-driven	**High** - Global, qualitative, knowledge-driven.
Goal	**Improve and simplify** the image; extract primitives.	**Understand and interpret** the image content.
Typical Tasks	Edge Detection, Denoising, Color Correction, Feature Points	Object Detection, Classification, Face Recognition, Scene Captioning
Question Answered	Where are the **edges**? How **bright** is this area?	**What** is this? **What** is happening un the scene?

Computer Vision: Framework

- Computer vision development and deployment are facilitated by various tools and frameworks that provide pre-built functions, libraries, and environments.

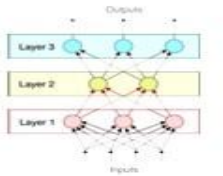


Computer Vision: Application and Reference Book


2. Image formation


3. Image processing

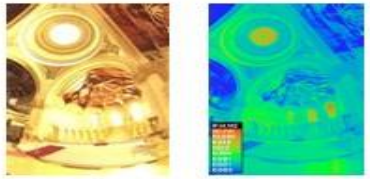

4. Optimization


5. Deep learning

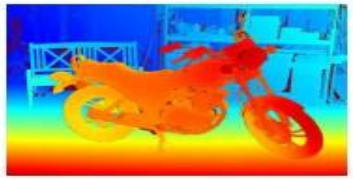

6. Recognition


7–8. Features & alignment


9. Motion estimation


10. Computational Photography


11. Structure from motion


12. Depth estimation


13. 3D reconstruction


14. Image-based Rendering

Computer Vision: Algorithms and Applications (Vision)
Szeliski, R., Springer

Computer Vision: A Modern Approach (Vision)
Forsyth, D and Ponce, J., Prentice Hall

Robot Vision (Vision)
Horn, B. K. P., MIT Press

A Guided Tour of Computer Vision (Vision)
Nalwa, V., Addison-Wesley

Digital Image Processing (Image Processing)
González, R and Woods, R., Prentice Hall

Optics (Optics)
Hecht, E., Addison-Wesley

Eye and Brain (Human Vision)
Gregory, R., Princeton University Press

Animal Eyes (Biological Vision)
Land, M. and Nilsson, D., Oxford University Press

Fourier Transform and its application:
Author: Ronald N Bracewell

Photometric Image Formation:

3D 2D

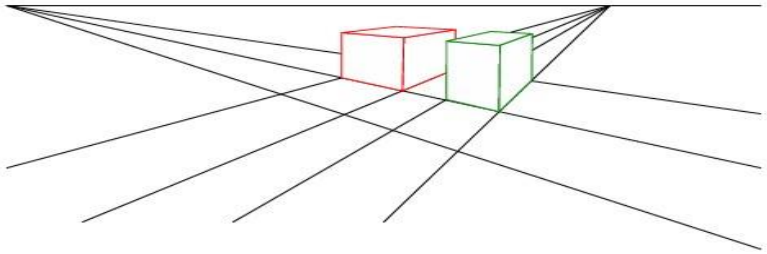
Photometric Image formation: is the process of projecting 3D scene onto a 2D plane or image.

For the above same, one need to understand the geometric and photometric relationship between the scene and its image.

Geometric: Given a point in a scene where it ends in the image.

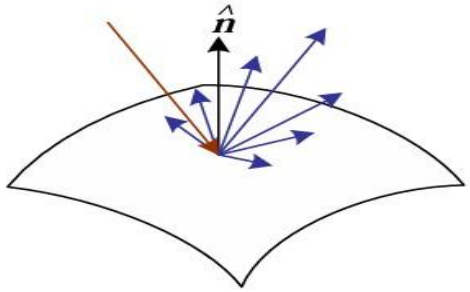
Photometric: Given a brightness and appearance of point in the scene what is the brightness and appearance of it in image.

Perspective Projections.

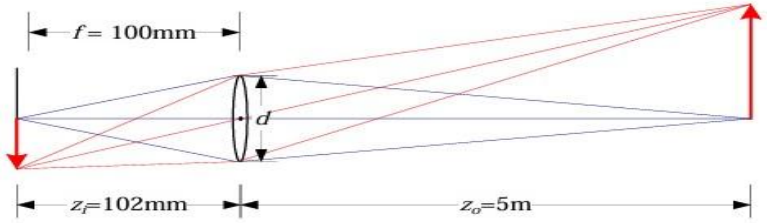


(a)

Light Scattering when hitting a surface.

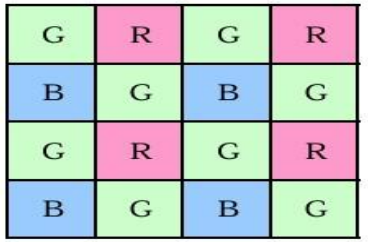


(b)



(c)

Lens Optics.



(d)

Bayer color filter array.

Overall components of image formation & Image Sensors

Perspective Projection

Is an image being formed on the screen ?

↓

Yes, but it is not a **clear** one

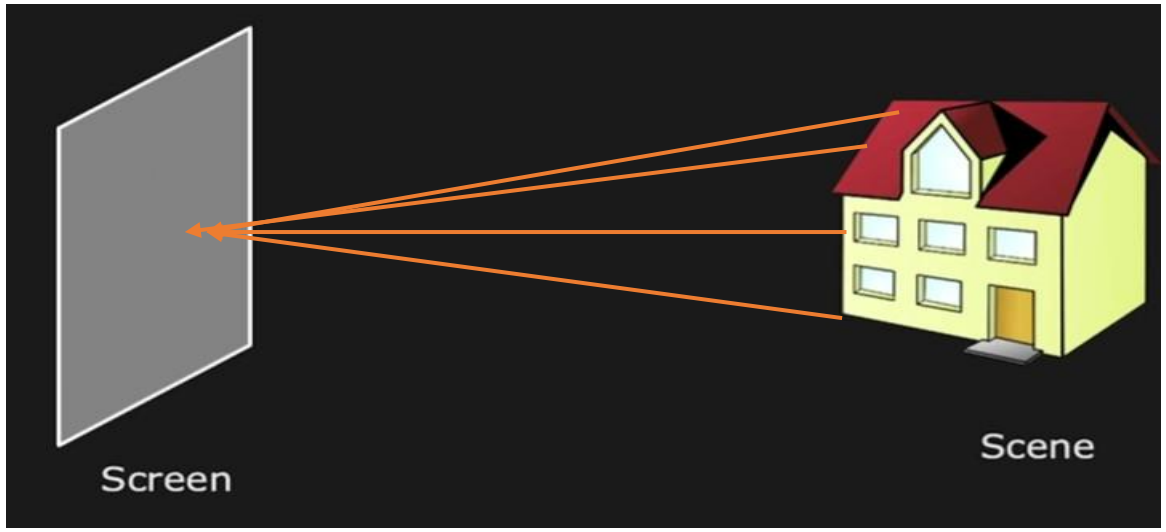
↓

Consider any point on the screen, It does receive light from Lot of points on the house (light is coming from different point On the house and focused on a single point in image plane.).

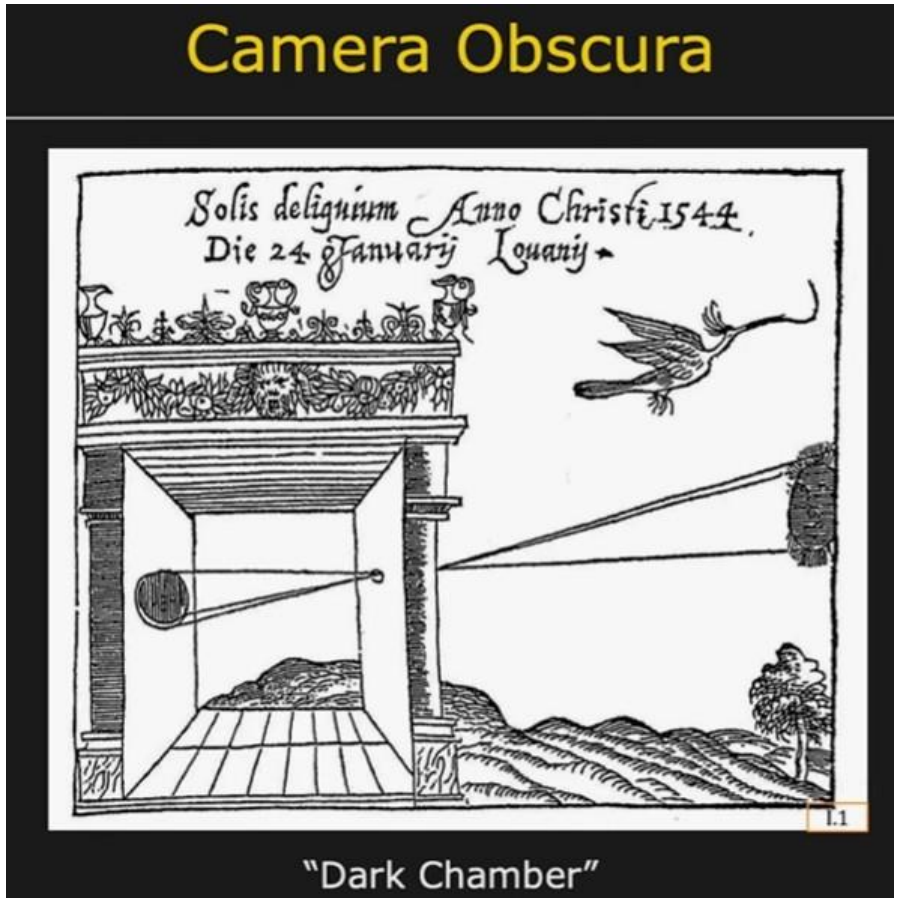
So image will be formed but it is not a clear image, it's a muddled image.

↓

Then how to get crisp and clear image on the screen.



Perspective Projection

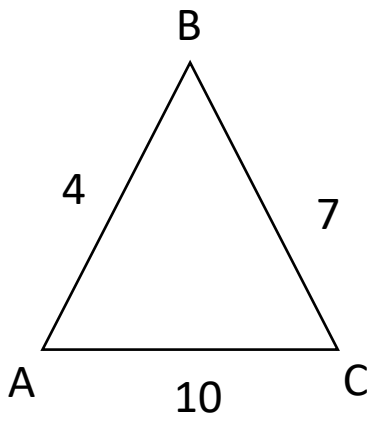


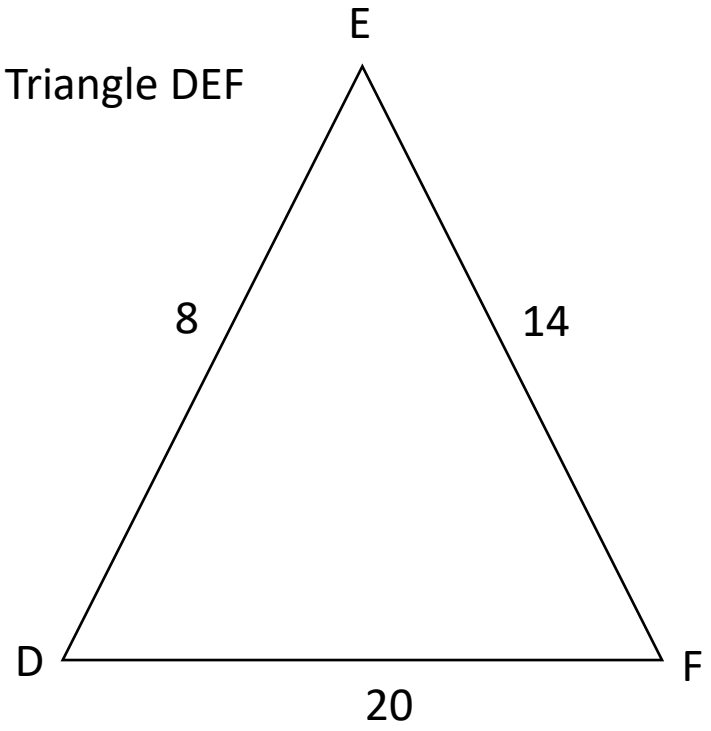
Camera Obscura or dark chamber

Dated around 500 BC.

Similar Triangles

Triangle ABC ~ Triangle DEF



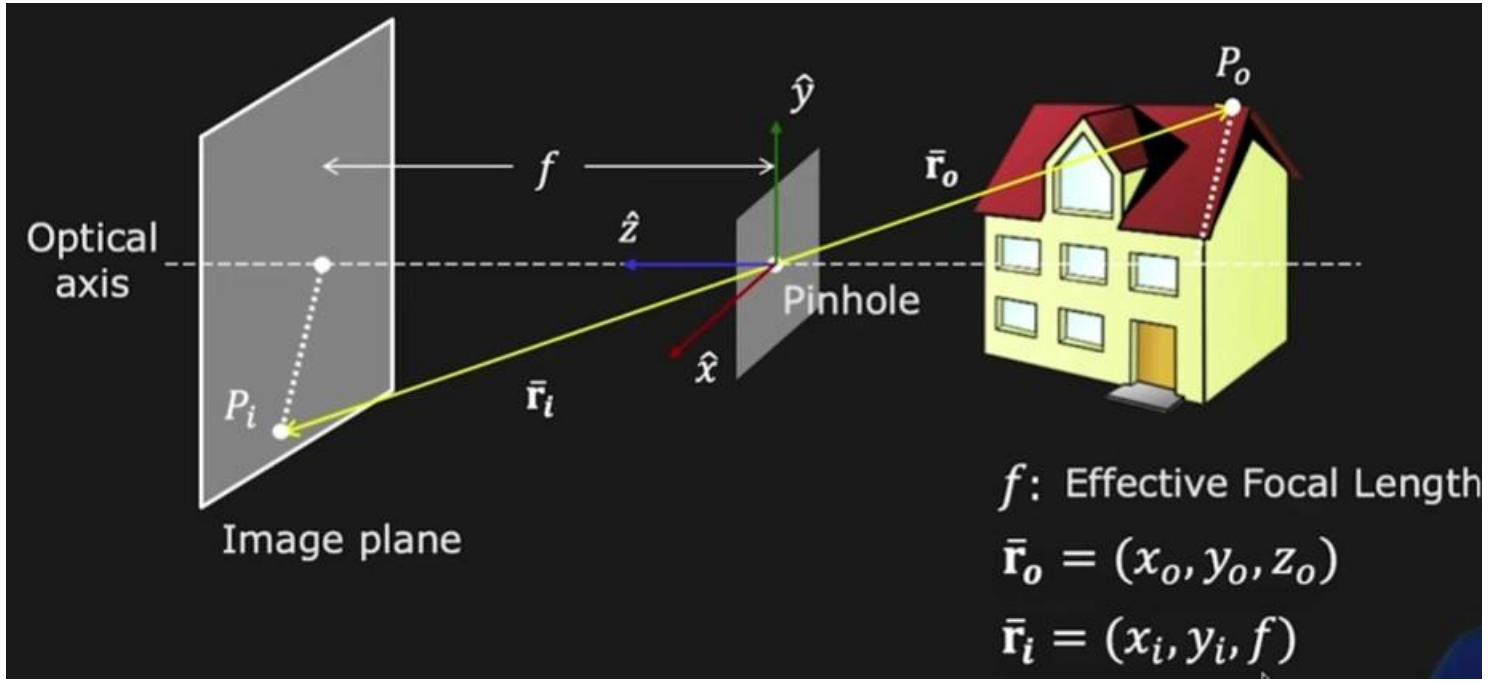


The lengths of corresponding sides are in the same ratio. This ratio is often called the scale factor (k).

$$AB/DE = BC/EF = AC/DF$$

$$4/8 = 7/14 = 10/20 = 1/2 \ (k)$$

Perspective Projection



Using similar triangles properties:

$$\frac{\bar{r}_i}{f} = \frac{\bar{r}_o}{z_o} \rightarrow \frac{x_i}{f} = \frac{x_o}{z_o}, \frac{y_i}{f} = \frac{y_o}{z_o}$$

These are the equation of perspective projection

Perspective projection is the process of projecting 3D scene or object onto a 2D plane or image.

Then how to get crisp and clear image on the screen.

↓

The answer is Pinhole

1. **Pinhole**: it's a sheet placed between scene and Image plane with the tiny hole in it.

2. Now the point Po on the scene is mapped to a single Point pi on the image plane or screen.

3. To understand the relationship between Po and Pi. First, erect a coordinate frame (xyz) placed on pinhole where z axis pointing towards the optical axis.

Optical axis: A axis perpendicular to image Plane.

4. Distance between pinhole and Image plane is known as **effective focal length (f)**.

Here in case of ri vector, z component is always equal to **f** i.e. effective focal length.

Image Magnification:

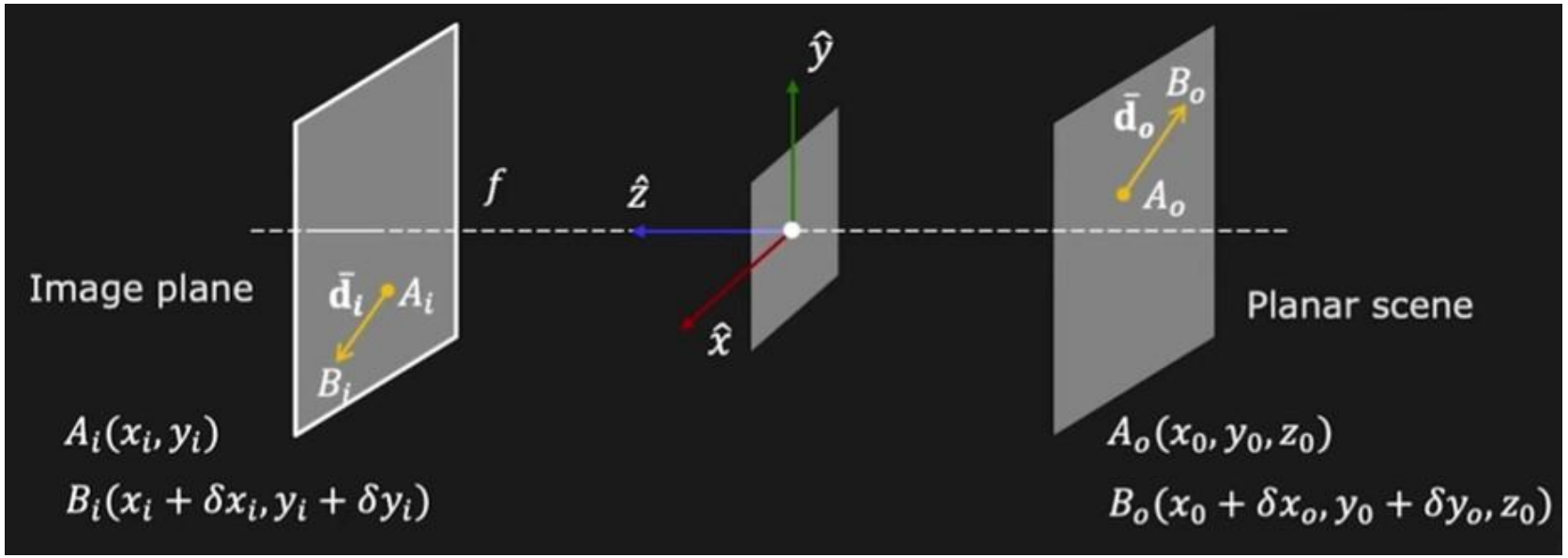


Image Magnification quantifies how much larger or smaller an image is compared to the actual object being viewed in the scene.

Consider a point (A0) and (B0) with distance (do).
Here ,Ao (xo, yo, zo) are the coordinates of it.

Image magnification = size of object in image

Size of object in Scene

Magnification: $|m| = \frac{\|\bar{\mathbf{d}}_i\|}{\|\bar{\mathbf{d}}_o\|} = \sqrt{\delta x_i^2 + \delta y_i^2} \Big/ \sqrt{\delta x_o^2 + \delta y_o^2}$

→ It can be written in terms of image displacement parameter.

From Perspective Projection:

$$\frac{x_i}{f} = \frac{x_o}{z_o} \text{ and } \frac{y_i}{f} = \frac{y_o}{z_o} \quad \text{----------} \quad \text{(A)}$$

$$\frac{x_i + \delta x_i}{f} = \frac{x_o + \delta x_o}{z} \text{ and } \frac{y_i + \delta y_i}{f} = \frac{y_o + \delta y_o}{z} \quad \text{---------} \quad \text{(B)}$$

→ To do the same, we can apply the perspective projection on point A0 and B0.

Image Magnification

From (A) and (B) we get:

$$\frac{\delta x_i}{f} = \frac{\delta x_o}{z_o} \quad \text{and} \quad \frac{\delta y_i}{f} = \frac{\delta y_o}{z_o}$$

Magnification:

$$|m| = \frac{\|\bar{\mathbf{d}}_i\|}{\|\bar{\mathbf{d}}_o\|} = \sqrt{\delta x_i^2 + \delta y_i^2} \Big/ \sqrt{\delta x_o^2 + \delta y_o^2} = \left|\frac{f}{z_o}\right|$$

Absolute value of m = $\frac{\text{Absolute value of focal length(f)}}{\text{Depth of the object in the scene (Z0)}}$

Size of the object in an image is inversely proportional to the distance of object from camera.

Example: When you take selfies then your nose is little large As compared to eyes and ears because distance between nose and camera is less as compared to other part of the face.

Image Magnification

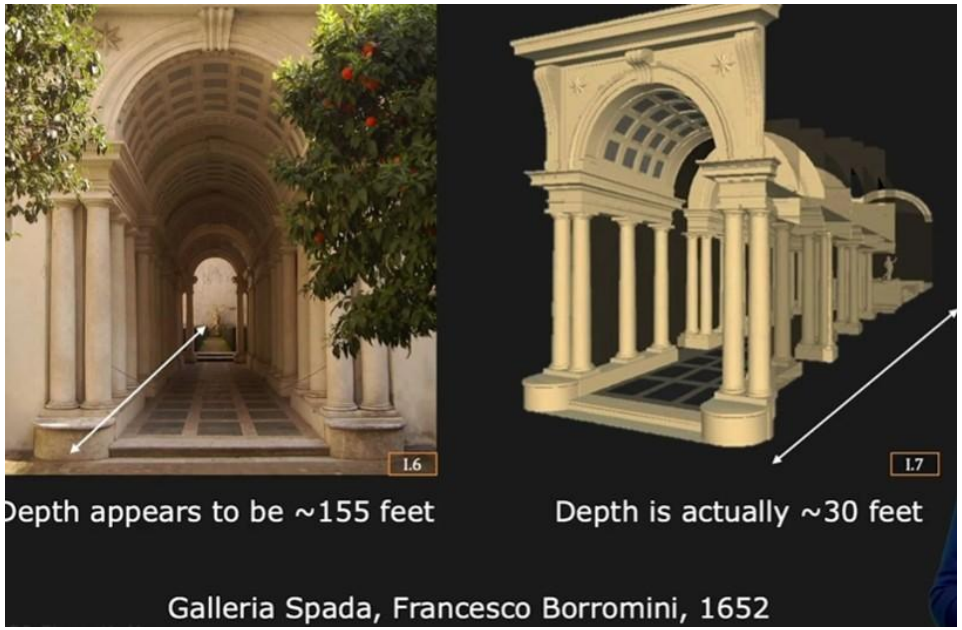
Train Tracks:
These tracks are parallel in 3D scene but in image, it seems these two tracks are intersecting to each other.

Because magnification is inversely proportional to the distance.

Here, A lot of parallel lines in scene. we can say all the lines are parallel because it's a tunnel.

In 2D image, it looks like it intersect to each other on a point known as vanishing point.

False Perspective



In perspective projection, a **vanishing point** is the specific spot in a 2D image where parallel lines in a 3D scene appear to meet or "converge" as they recede into the distance.
Location of vanishing point depends on the orientation of the parallel straight lines in 3D.

Size of the Pinhole

When the pinhole is relatively large, light from a single point on an object passes through the hole and creates a large "spot" on the sensor. This is purely geometrical.

•**The Result:** The image is blurry because the spots (pixels) are too big and overlap.

As you keep making the pinhole smaller and smaller, you eventually reach a point where the hole is comparable to the **wavelength of light**. At this stage, light waves begin to hit the edges of the pinhole and "scatter" or bend inward.

Diffraction is a physical phenomenon where waves (such as light, sound, or water) bend, spread out, and interfere with each other when they encounter an obstacle or pass through a narrow opening.

Ideal pinhole diameter: $d \approx 2\sqrt{f\lambda}$

f: effective focal length
λ: wavelength

Problem: Calculate the optimum pinhole diameter for a camera with an effective focal length of 50mm. assuming the wavelength of light is 550nm.

Steps –

Multiply focal length and wavelength

50 x 0.00055 = 0.0275 mm

Take the square root = sqrt(0.0275) = 0.16583

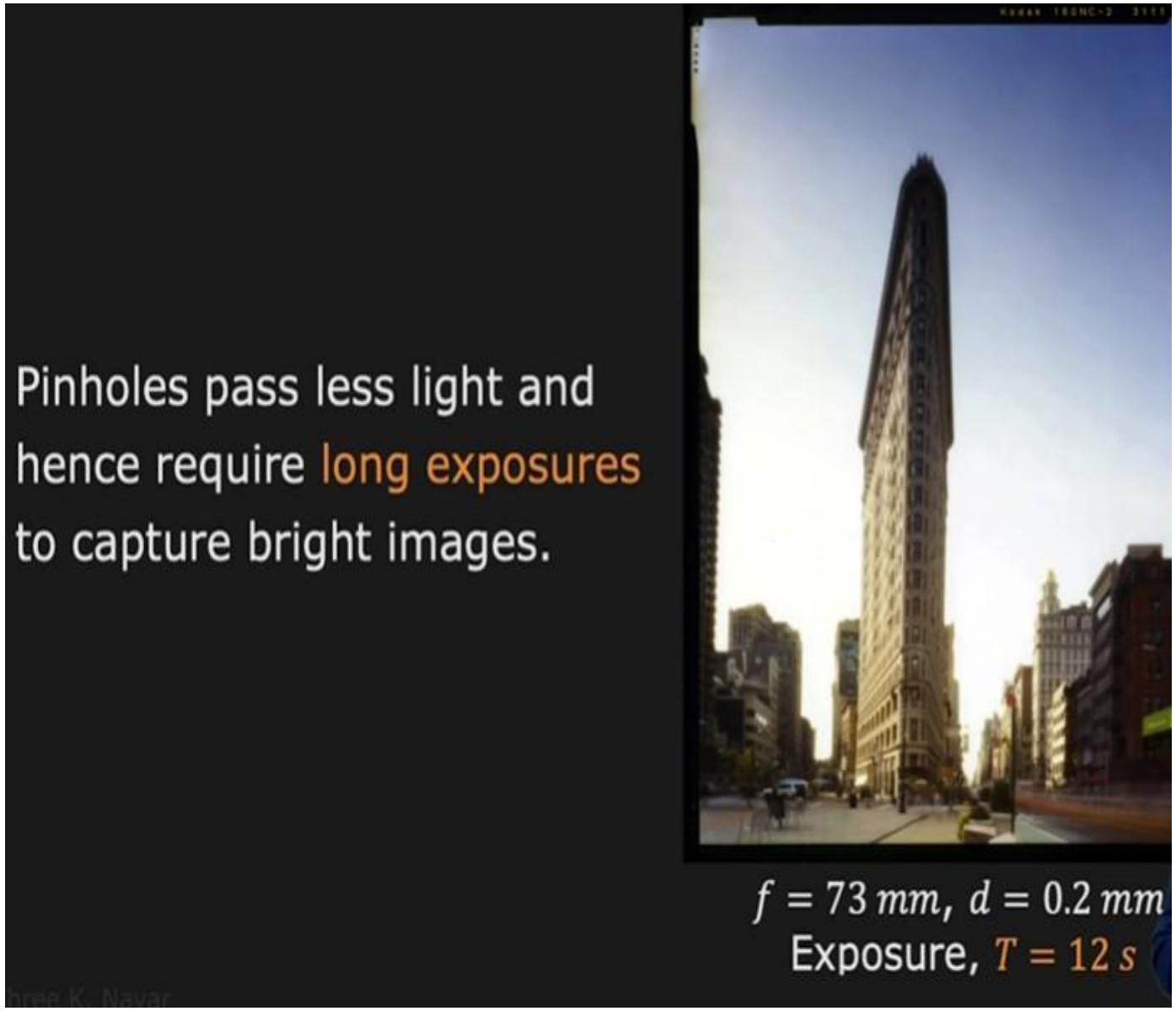
Multiply by the constant = 2 x 0.16583 = 0.33166mm

Optimum pinhole diameter is 0.33mm

If the hole is larger than 0.33 mm: Light rays from a single point on the object will spread out too much before hitting the sensor, creating a blurry "circle of confusion."

If the hole is smaller than 0.33 mm: The wave nature of light causes it to bend (diffract) around the edges of the hole, which also creates a blurry image.

Size of the Pinhole

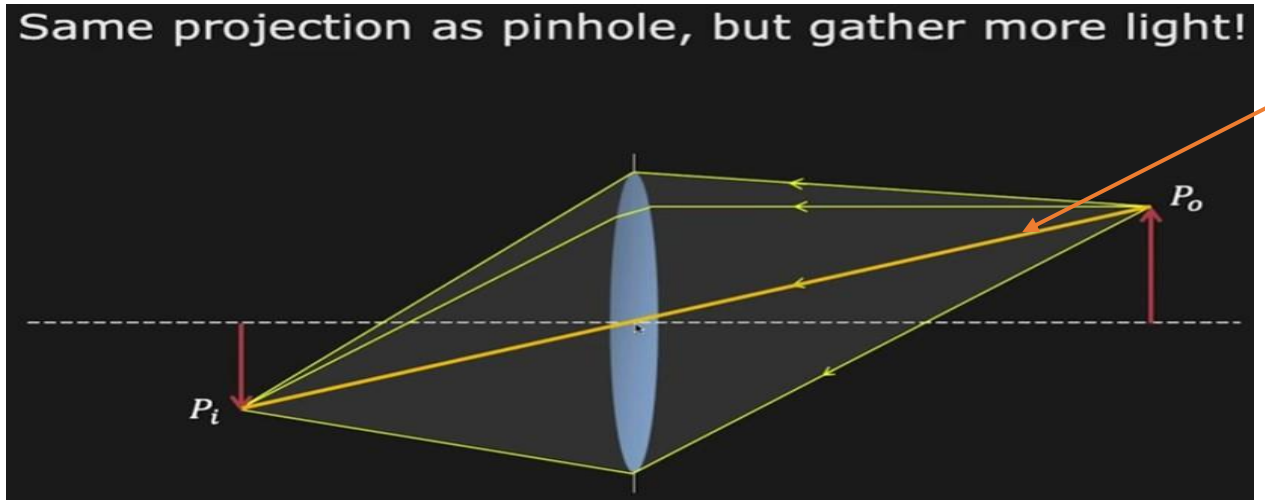


It is well focused bright image, it focused on almost all objects. Since the pinhole capture very little light. (With the **exposure time T = 12 sec**) To capture 1 frame you need to wait for at least 12 sec. Any computer vision system or mobile cameras can not wait for 12 sec to capture 1 single frame.

Exposure time : In a pinhole camera, **exposure time** is the duration for which the shutter remains open, allowing light to pass through the pinhole and hit the light-sensitive material (film or digital sensor**)**.

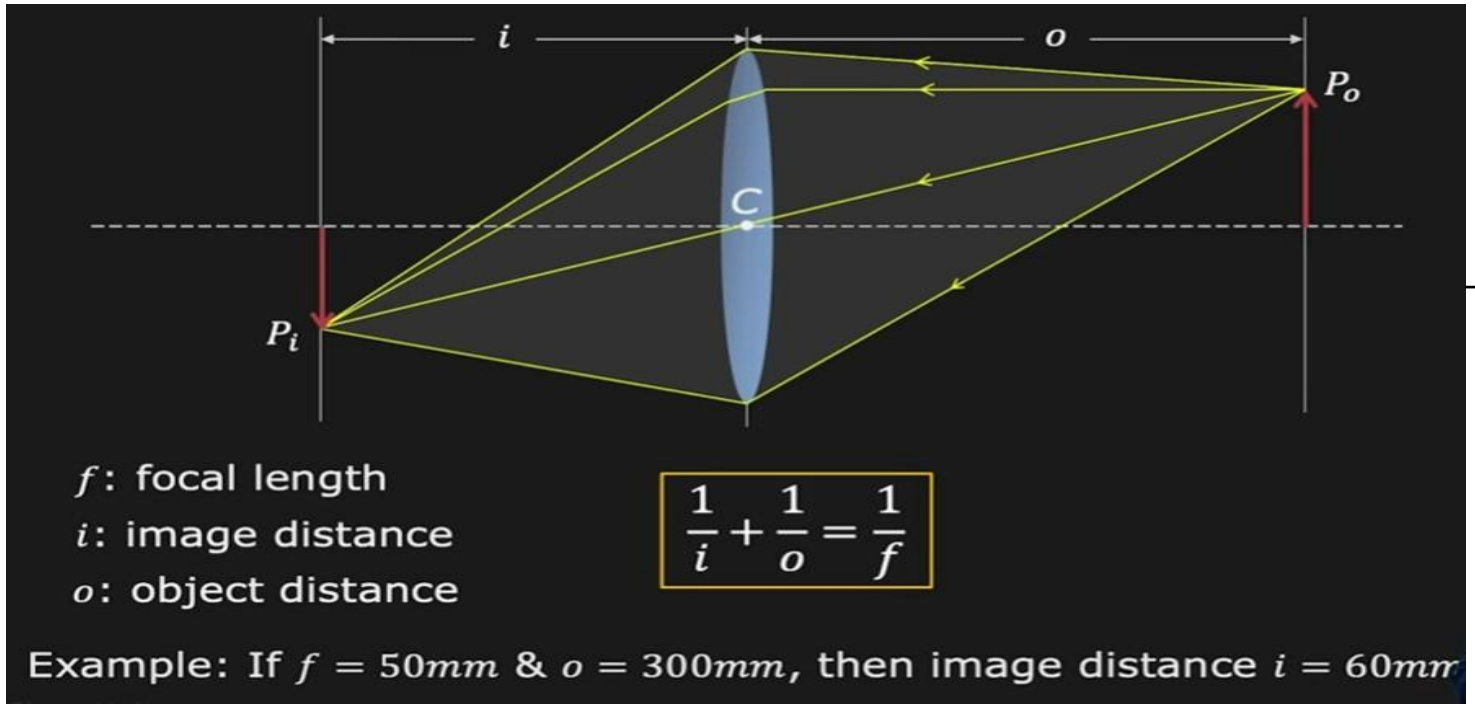
This is the problem with pinhole camera: A long exposure time then the solution is instead of using pinhole one can utilize the **lens.**

Image Formation using Lenses



Projection of Pinhole camera on the image plane.

Here focal length define by the lens bending power.



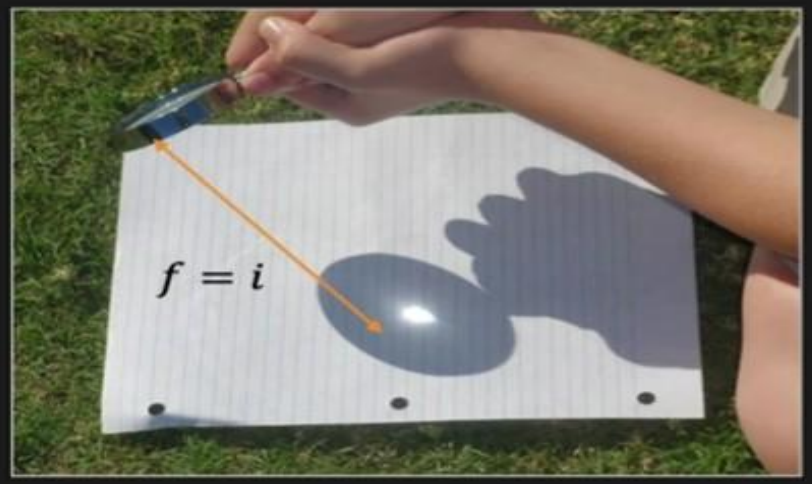
Gaussian Lens (Thin Lens) Law:

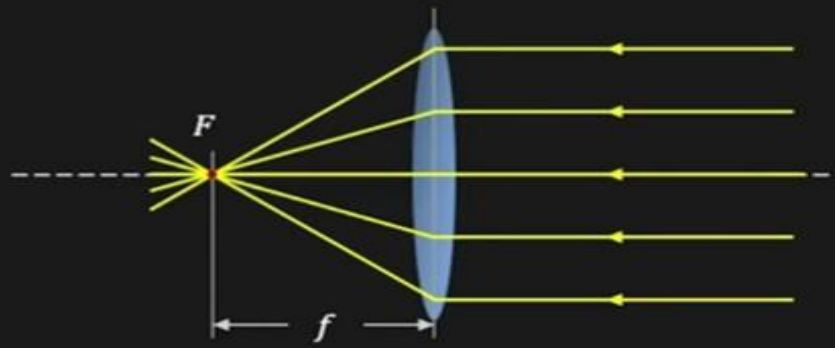
Relation between the position of P0 and the position of point Pi.

Image Formation using Lenses

How to find focal length of a lens when it is not given to you.
Lets assume object is very far away -

$$\frac{1}{i}+\frac{1}{o}=\frac{1}{f} \quad \Rightarrow \quad \text{If } o=\infty, \text{ then } f=i$$



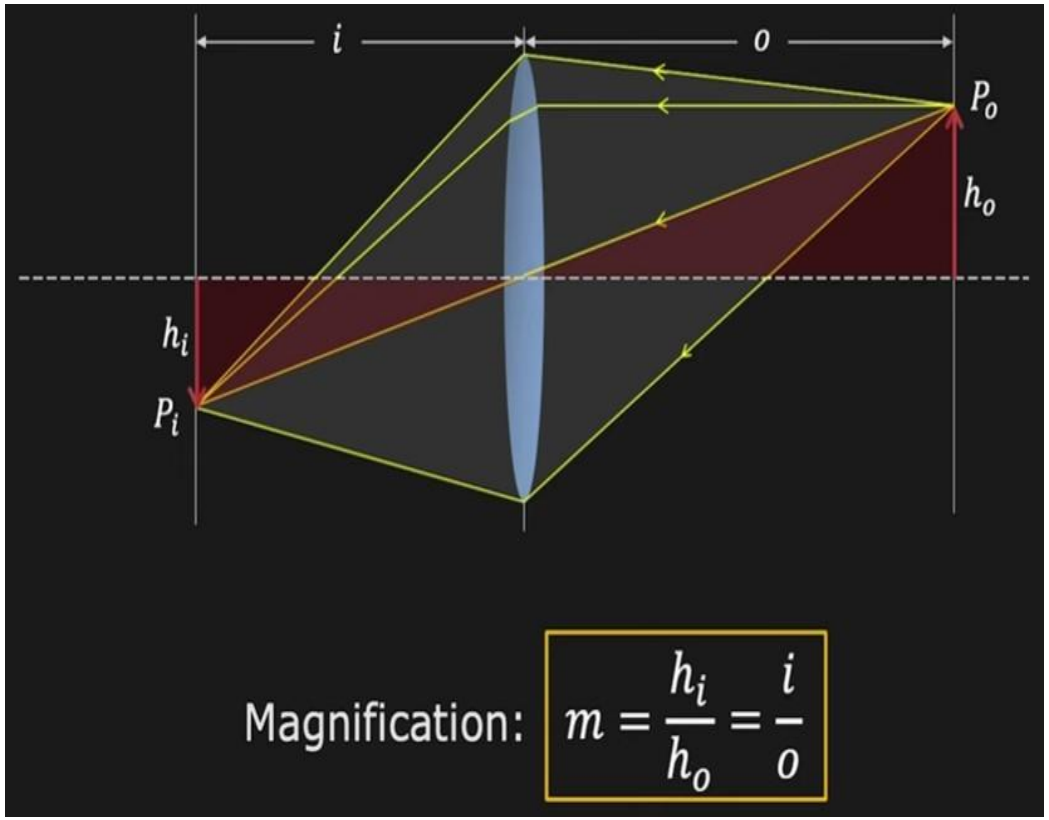


Focal length: Distance at which incoming rays that are parallel to the optical axis converge.

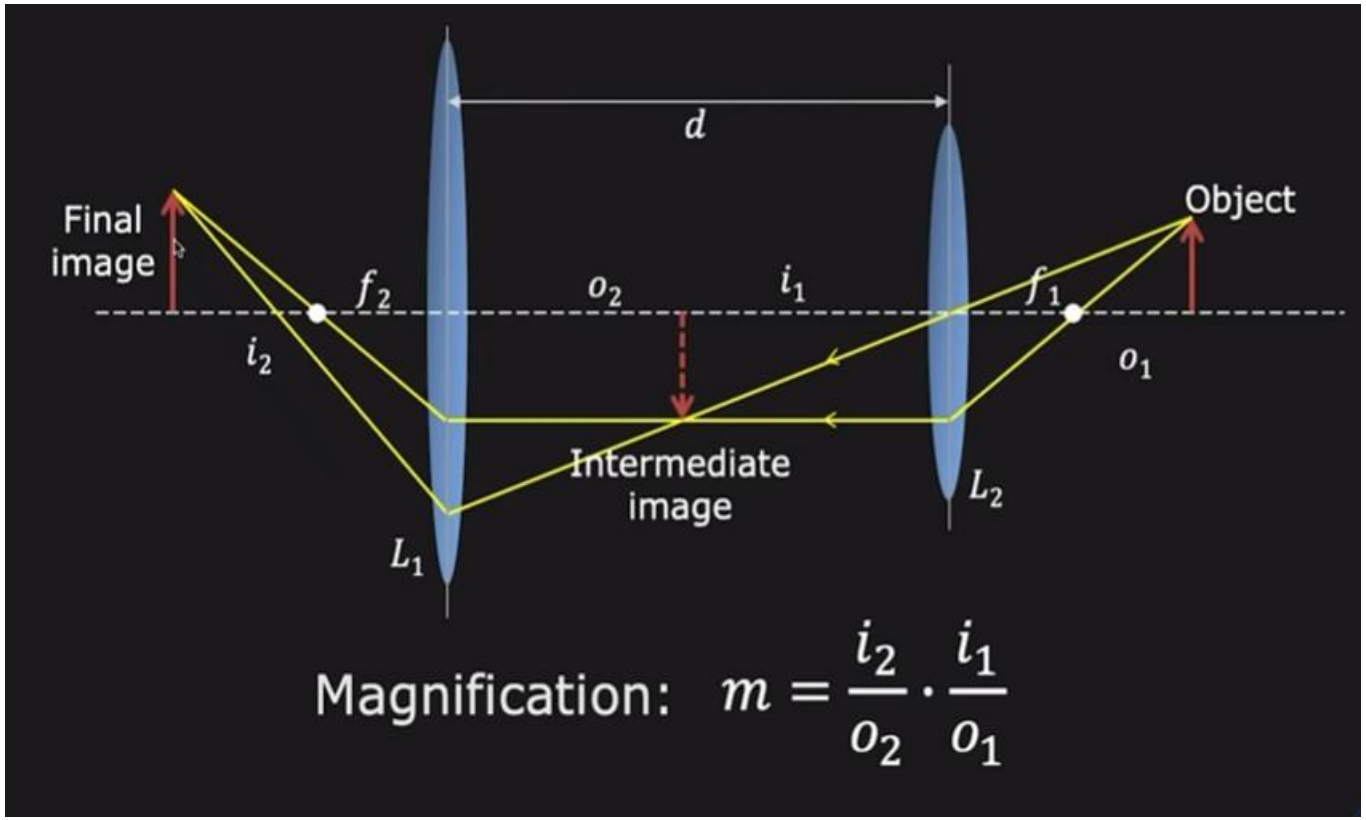
Here the distance between focused image and lens is the focal length.

Focal length of the lens is generally depend on the material by which it is made and the curve of lens.

Image magnification using Lens:



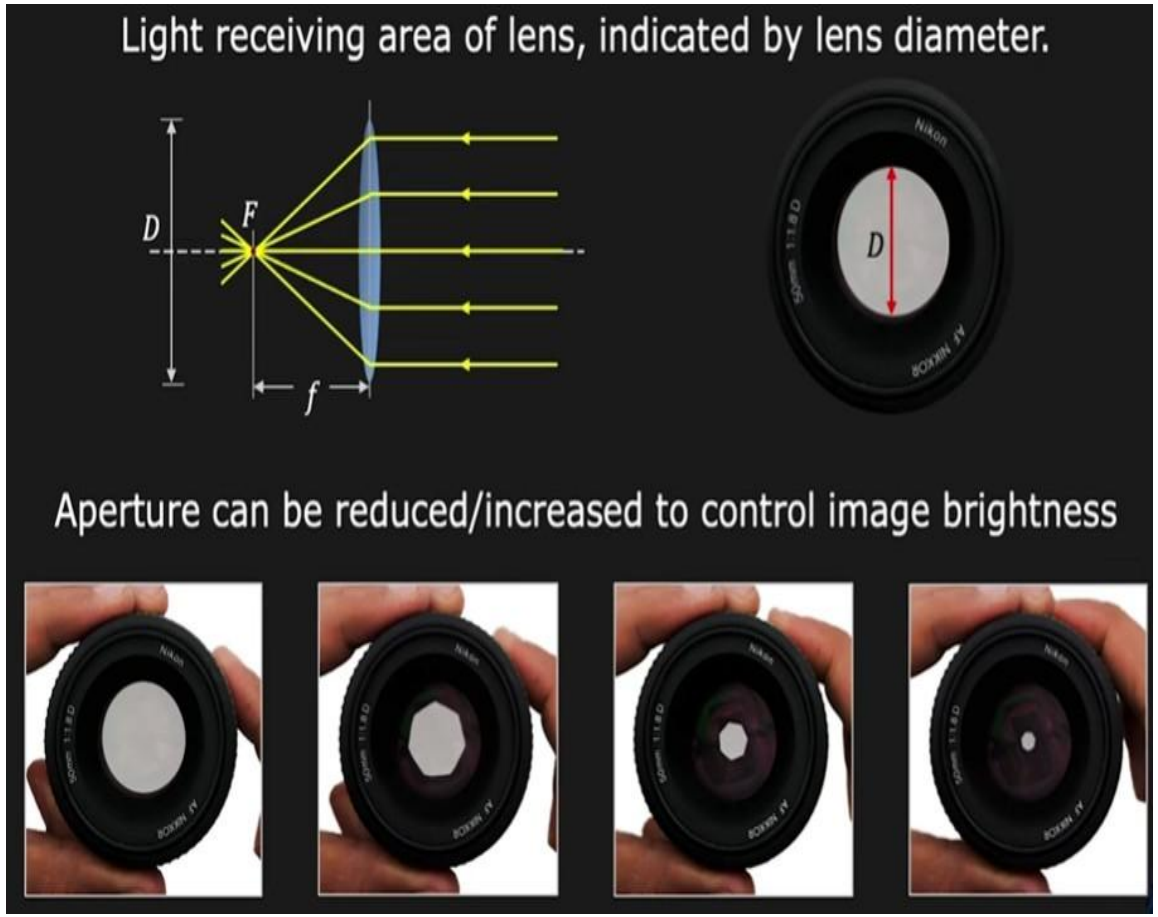
Here h0 and hi is height of the object in the scene and image respectively.



Two lens System:
Move the lens to change magnification is nothing but the Zooming.

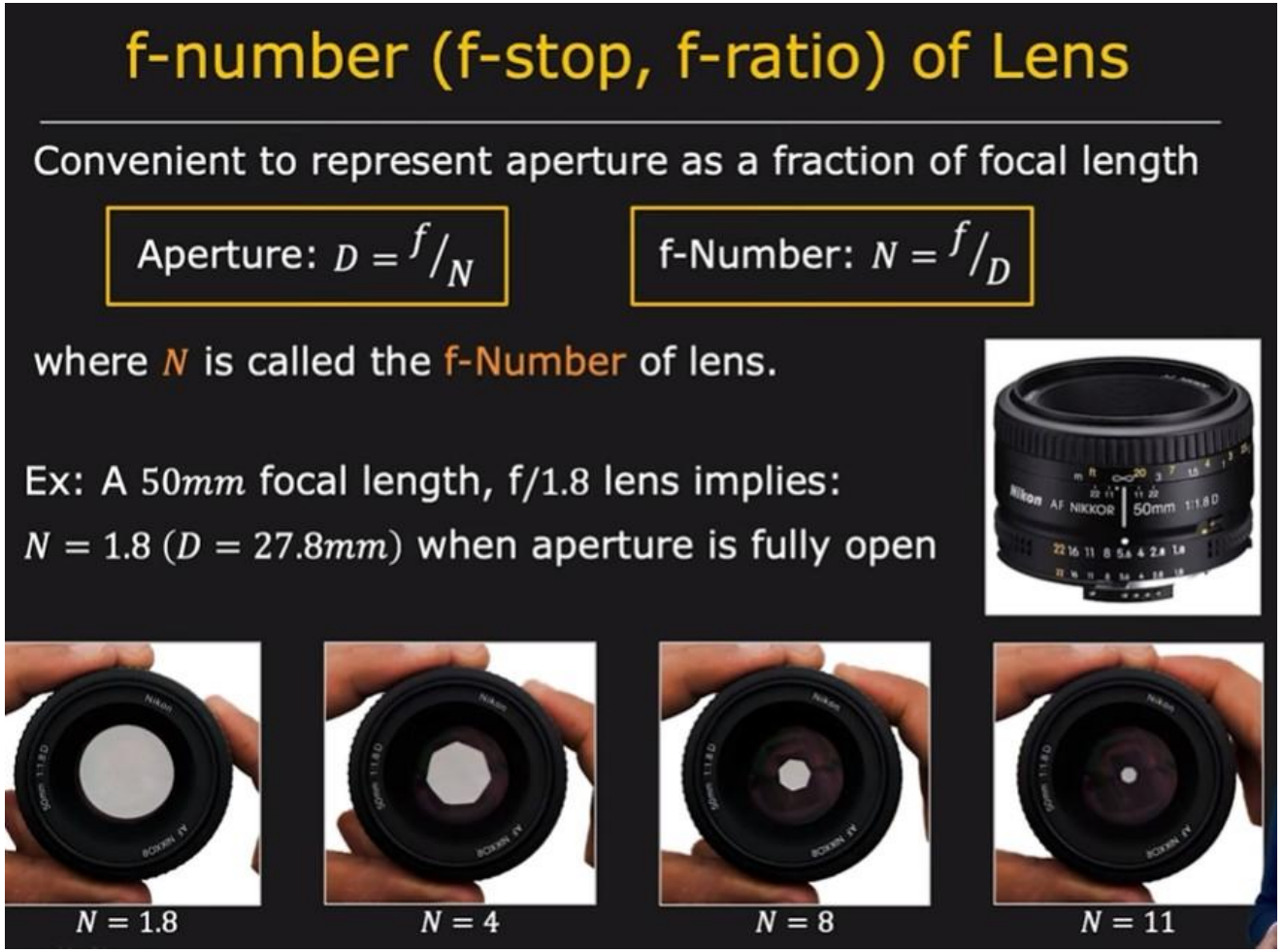
Aperture of the Lens and f-Number

Aperture of the Lens: It is the clear area of the lens that gather light from points in the scene.



D = Diameter of the aperture.

The **f-number** is defined as the ratio of the **focal length** of the lens to the **diameter of the entrance pupil** (the effective aperture opening).



Numerical: F-Number and Aperture of the lens

•**F-number:** The ratio of the focal length to the aperture diameter. N = f/D
•**Aperture Diameter (D)**: The physical width of the lens opening. D = f/N

Where:
- f = Focal length of the lens.
- D = Diameter of the entrance pupil (effective aperture).
- N = F-number (often written as f/N)

Problem: You have a telephoto lens with a focal length of **200 mm**. The effective diameter of the aperture is measured at **50 mm**. What is the f-number?
Step-by-Step Calculation:
1.Identify variables: f = 200 mm, D = 50 mm.
2.Apply formula: N = 200/50 = 4
3. Notation: This is expressed as **f/4**.

Problem: A standard portrait lens has a focal length of **85 mm**. If you set the camera to an aperture of **f/1.8**, what is the actual diameter of the lens opening?

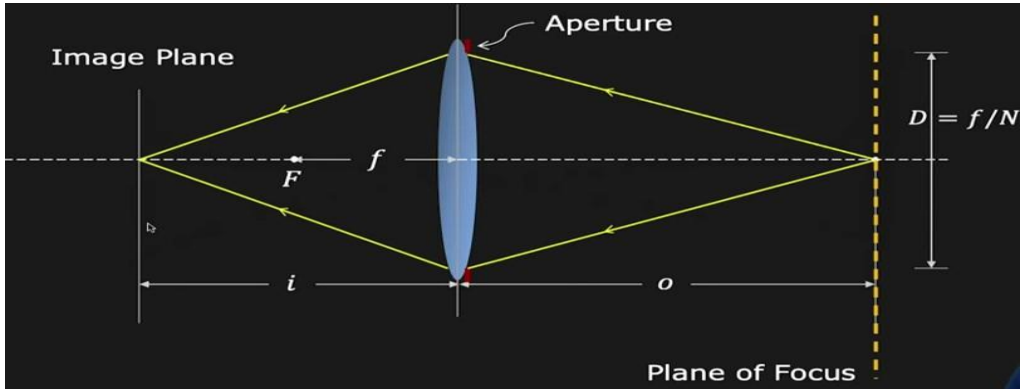
Step-by-Step Calculation:
1.Identify variables: f = 85 mm, N = 1.8.
2.Apply formula: D = 85/1.8
3.Result: D = 47.22 mm.

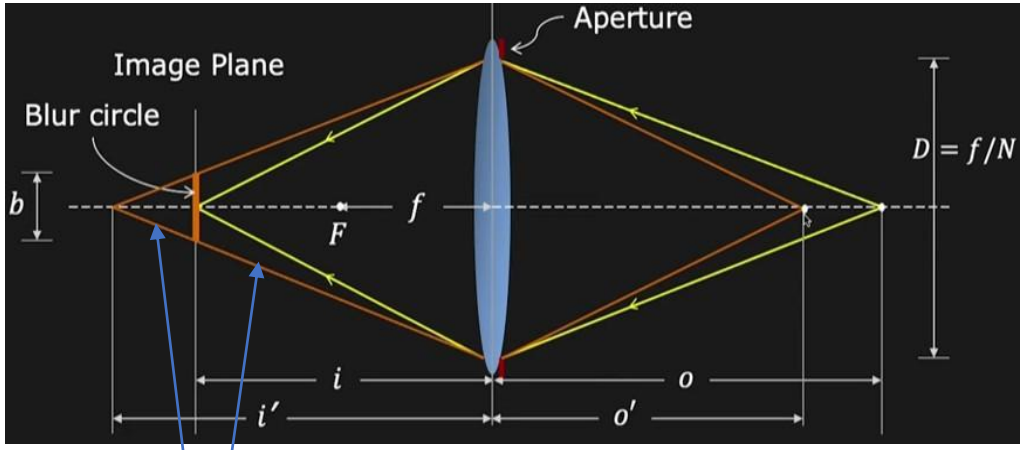
• **Light Intensity:** A smaller f-number (like f/1.8) means a larger diameter (D). A larger diameter allows more light to hit the sensor, which is why "fast" lenses have low f-numbers.

• **Aperture Area:** Light gathering depends on the **area** of the circle (pi r^2). If you change from f/4 to f/2.8, you are decreasing the f-number by a factor of sqrt(2) (Approx. 1.4), which effectively **doubles** the area and the amount of light entering the lens.

Lens Defocus.

Even though Lens gathers a lot of light but the price that we paid when we used it is, there is only one plane in the scene which is perfectly focused on the image plane.





These small and big triangle are the similar triangle

Plane of focus -
Here, this is the only one plane which is in focus. Any point on this plane is perfectly focused on the Image plane.

Here when the ray pass through it, it focuses on a single point in the image plane.

Now what happen if you lie outside this plane of focus.
Let's consider a point o', as this point is closer to the lens its image is behind to the focused image plane.

When the ray pass through it, it focuses not only a single point, but it will distribute over a circular disc on the image plane. It is going to be blur and it is known as blur circle or circle of confusion. Blur circle has a diameter equal to b.

From similar triangles:

$$\frac{b}{D} = \frac{|i' - i|}{i'}$$

Blur circle diameter:

$$b = \frac{D}{i'}|i' - i| \qquad b \propto D \propto \frac{1}{N}$$

Blur circle diameter (b) is proportional to the diameter of the aperture (D) and Inversely proportional to the f-number of the lens.

Lens Defocus.

How to calculate Diameter of a blur circle

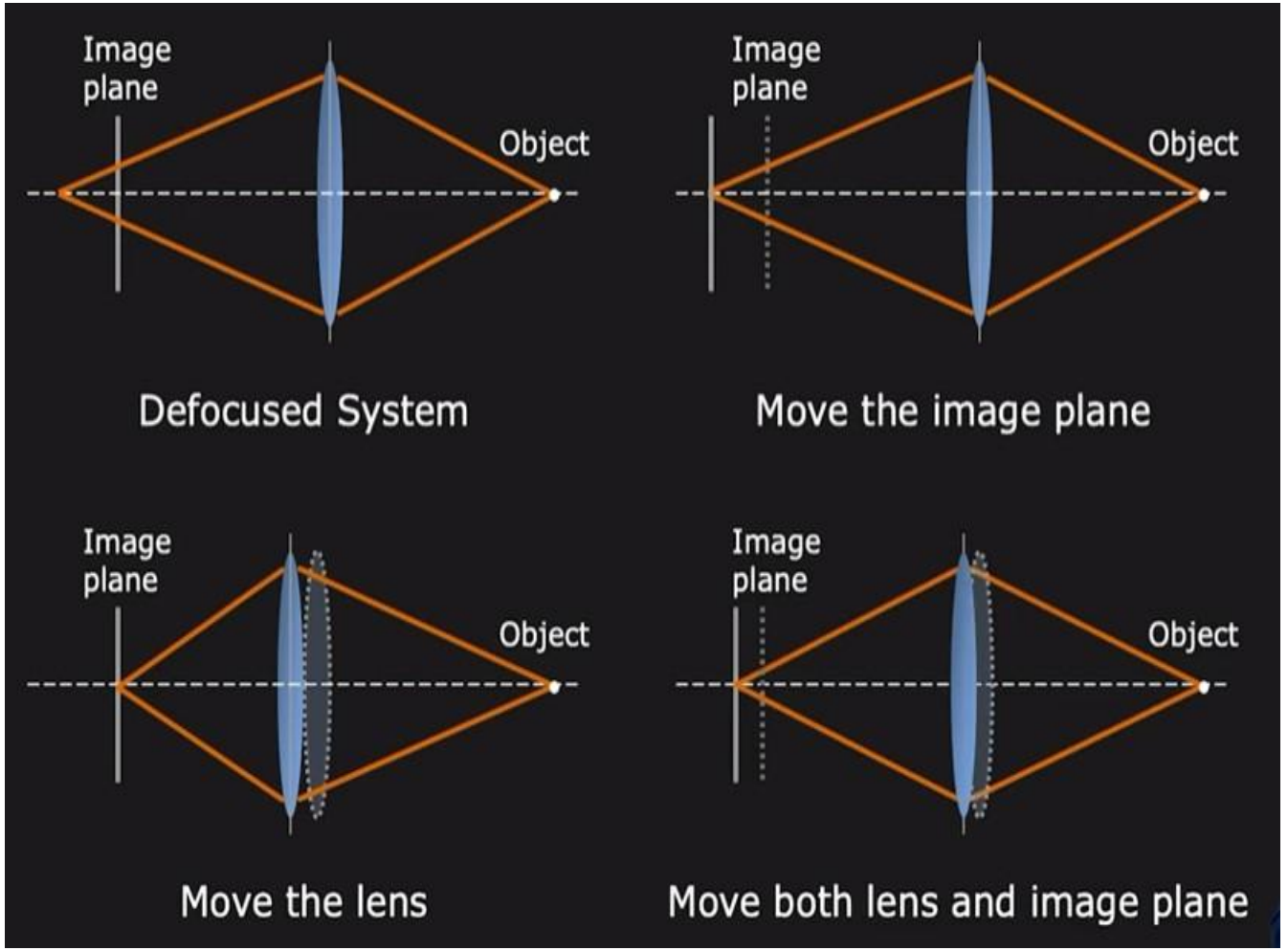
Blur Circle (Defocus)

Focused Point	Defocused Point
$\frac{1}{i}+\frac{1}{o}=\frac{1}{f}$	$\frac{1}{i'}+\frac{1}{o'}=\frac{1}{f}$ (Gaussian Lens Law)
$i=\frac{of}{o-f}$	$i'=\frac{o'f}{o'-f}$

$$i'-i=\frac{f}{(o'-f)}\cdot\frac{f}{(o-f)}\cdot(o-o')$$

$$b=Df\left|\frac{(o-o')}{o'(o-f)}\right| \qquad b=\frac{f^2}{N}\left|\frac{(o-o')}{o'(o-f)}\right|$$

Focusing



Numerical: Calculating blur circle diameter in Lens

Problem: Calculating the blur circle (or **Circle of Confusion**) for a lens is more complex than a pinhole camera because it involves a lens that is actively focusing on a specific plane. The blur occurs when an object is located at a distance different from the one the lens is currently focused on.

Problem: You are using a **50 mm** lens set to an aperture of **f/2**. You have focused the lens on a person standing **2000 mm** (2 meters) away. However, there is a light source in the background **10,000 mm** (10 meters) away. Calculate the diameter of the blur circle created by that background light.

For a lens focused at distance P, the blur circle diameter c for an object at distance S is:

$c = A \times \{f \times |S - P|\} / \{S \times (P - f)\}$

- **f = 50mm**
- **P = 2000**
- **S = 10,000 mm**
- **A** = f/f-number = 50/2 = 25 mm

$c = 25 \times \{50 \times |10{,}000 - 2000|\} / \{10{,}000 \times (2000 - 50)\}$

Answer: The blur circle diameter is approximately **0.51 mm**.

CSET340

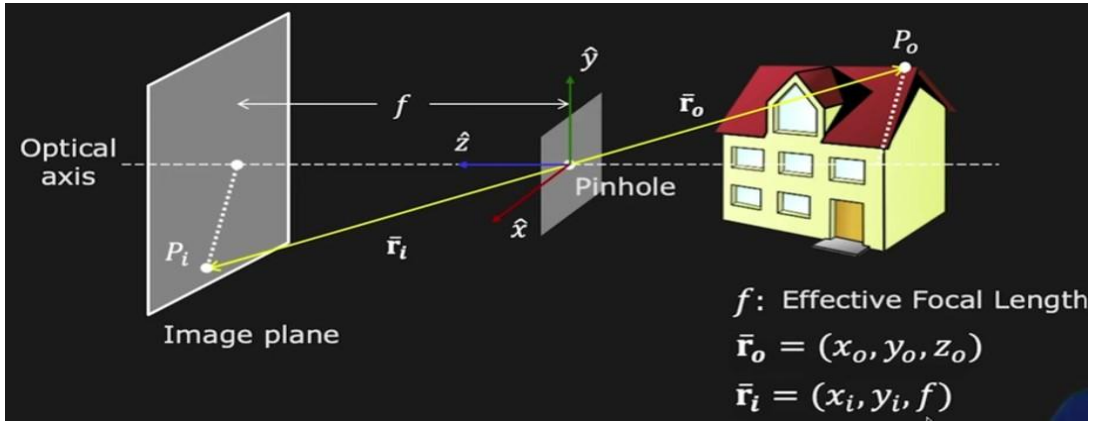
Advanced Computer Vision and Video Analytics

19th Jan. to 23th Jan. 2026

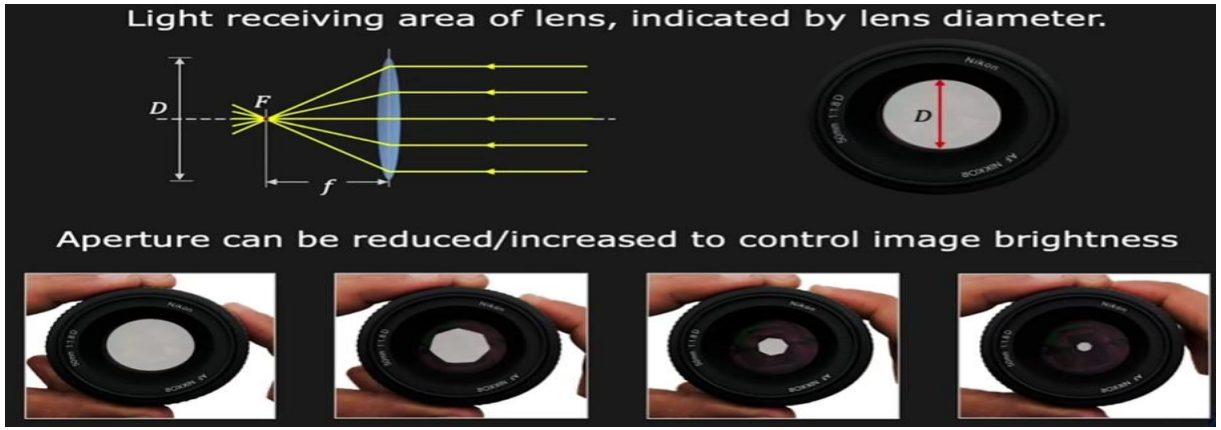
Overall Course Coordinator-
Dr. Gaurav Kumar Dashondhi
Gaurav.dashondhi@bennett.edu.in

Note : Any query related to course then first connect with overall course coordinator.

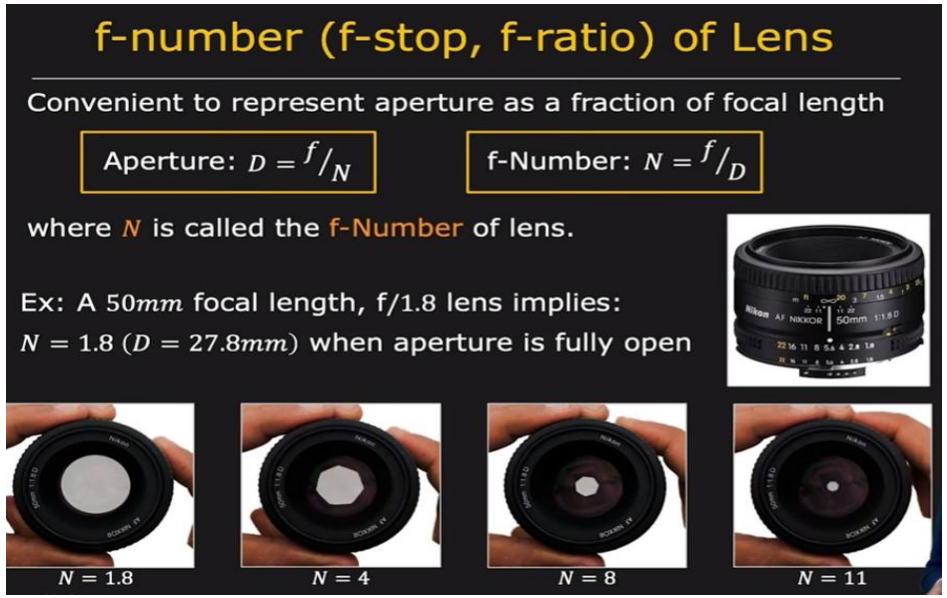
Basic Definitions



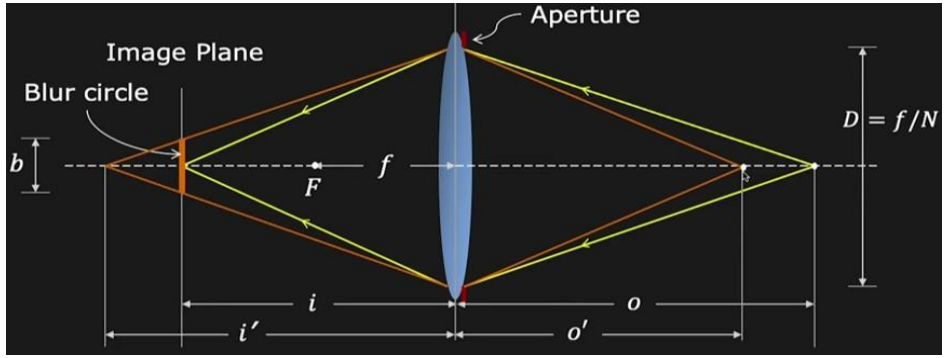
Optical axis: A axis perpendicular to image Plane. Distance between pinhole and Image plane is known as **effective focal length (f)**.



Aperture of the Lens: It is the clear area of the lens that gather light from points in the scene.



The **f-number** is defined as the ratio of the **focal length** of the lens to the **diameter of the entrance pupil** (the effective aperture opening).



When the ray pass through the lens, it not focuses only on a single point, but it will distribute over a circular disc on the image plane. It is going to be blur and it is known as **blur circle or circle of confusion**.

Depth of field (DOF)

1. As there is one plane in a scene that is perfectly focused, all the objects are lie outside the range are going to be out of focus.

2. Degree of defocus will depend on the distance of it from the plane of focus.

3. Image is made up by pixel. Now, the best focused points are those points where blur circle lies within single pixel.

Condition for focused point or object = diameter of blur circle is less than or within the pixel size.

Depth of field: The range(min-max) of object distance for which the image is sufficiently well focused i.e the range over which the blur (b) is less than the pixel size is called depth of field of th imaging system.

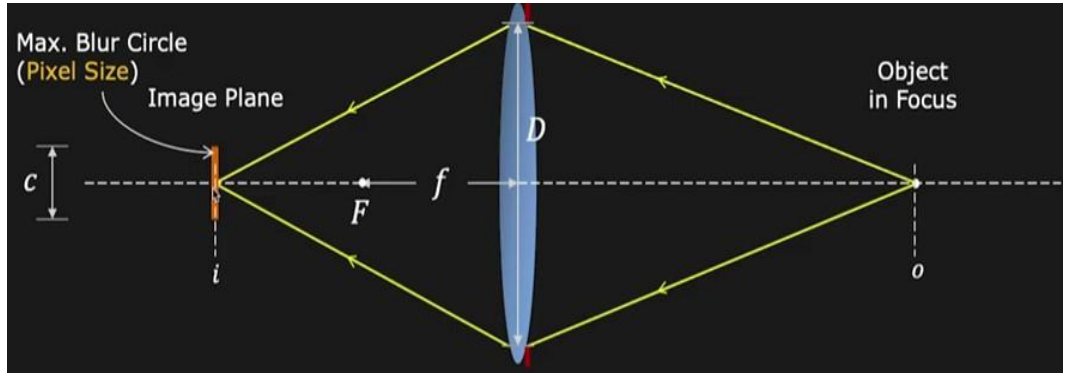
or

It is the distance between the closest and farthest objects in a picture that still look sharp and clear.

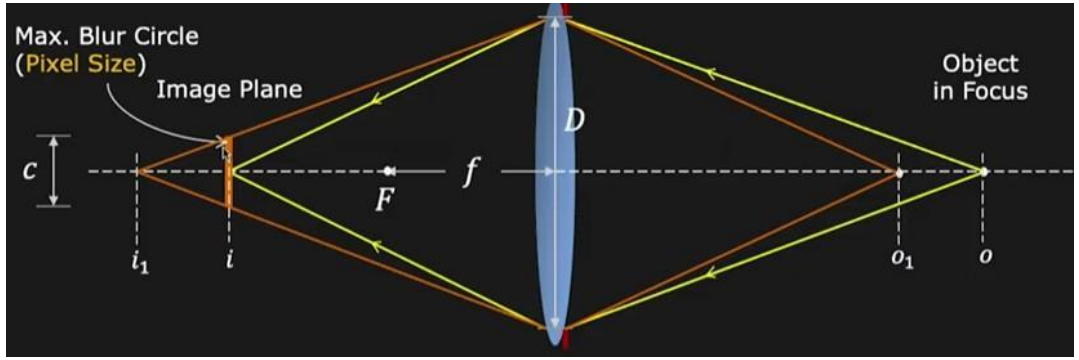
Depth of Field (DoF)

DOF for a lens System

1. First, define pixel size. lets call it C. Here O is the object which is in focus.

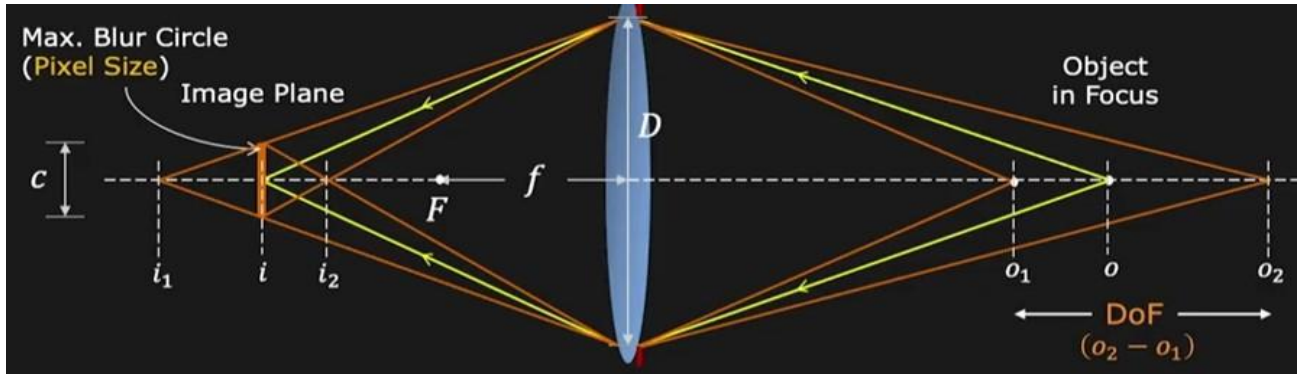


2. Now, we want to see what is the range of the Object (O) for which blur circle is smaller than C.



3.. Consider a point O1, for this point the blur circle diameter is exactly equal to the pixel size.
Here, O1 is close to the lens as compared to O, so the image is formed behind the image plane i.e. at i1.

4. Now, consider a point O2, where blur circle is exactly equal to the pixel. Here, O2 is further away to the lens as compared to O, so the image is formed before the image plane i.e. at i2.



5. Depth of field (DOF) = O2 – O1.

If o_1 and o_2 are the nearest and farthest distances respectively for which blur circle is maximum c, then:

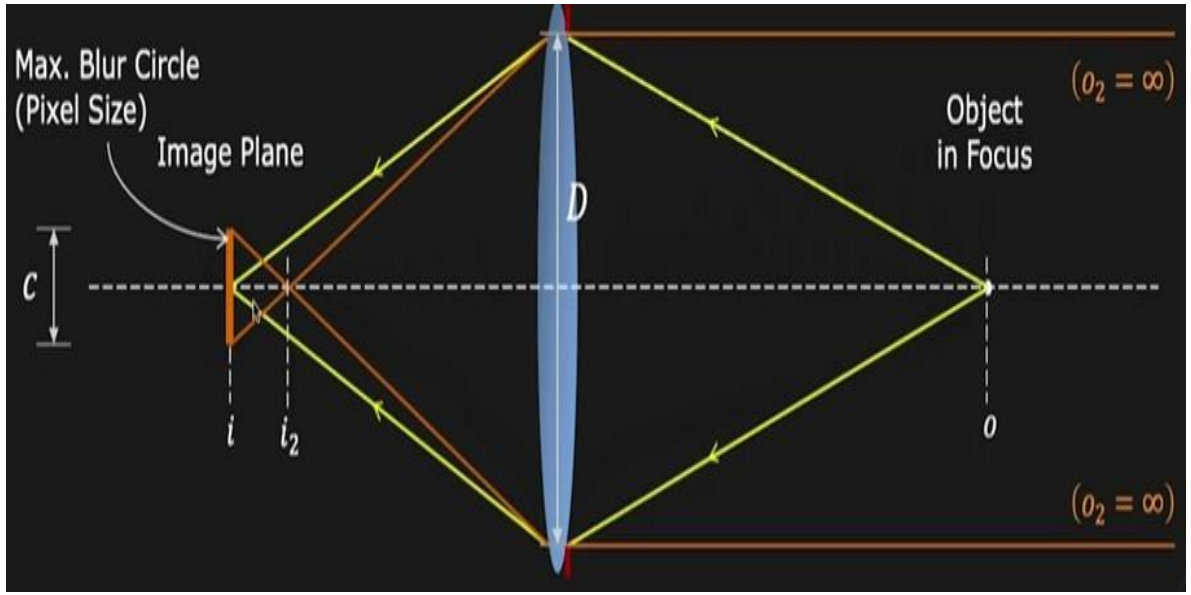
$$c = \frac{f^2(o - o_1)}{No_1(o - f)} \qquad c = \frac{f^2(o_2 - o)}{No_2(o - f)}$$

Depth of Field: $$o_2 - o_1 = \frac{2of^2cN(o - f)}{f^4 - c^2N^2(o - f)^2}$$

Here c is the diameter of the blur circle..

Now, as the object is going closer to the lens then blurriness will increase, this brings us to the concept of hyperfocal distance.

Hyperfocal Distance



Hyperfocal distance: it is the closer distance that you would focus a lens at for which all the points beyond that distance are going to be in focus. So from that depth O, all the points beyond that all the way to Infinity are going to produce a blur circle that are smaller than the size of the pixel that's call a hyperfocal distance.

If we keep O2 = infinity in the equation of depth of field, we can get this hyperfocal distance

Hyperfocal Distance: $h = \frac{f^2}{Nc} + f$

Where, f = focal length, N = f-number and c = pixel size

The "Perfect Landscape" Example

Imagine you are standing in a beautiful field of flowers.

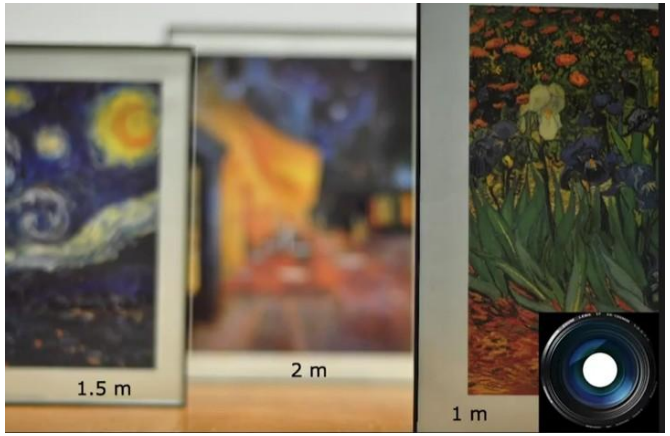
- **Foreground:** There is a single, stunning daisy just **3 feet** in front of your camera.
- **Background:** There are massive, snow-capped mountains **10 miles** away on the horizon.

The Problem:

- If you focus directly on the **daisy**, the mountains will be a blurry mess.
- If you focus on the **mountains** (infinity), the daisy at your feet will be out of focus.
- The **hyperfocal distance** is a specific spot where daisy and mountain both are perfectly in focus in one single shot.

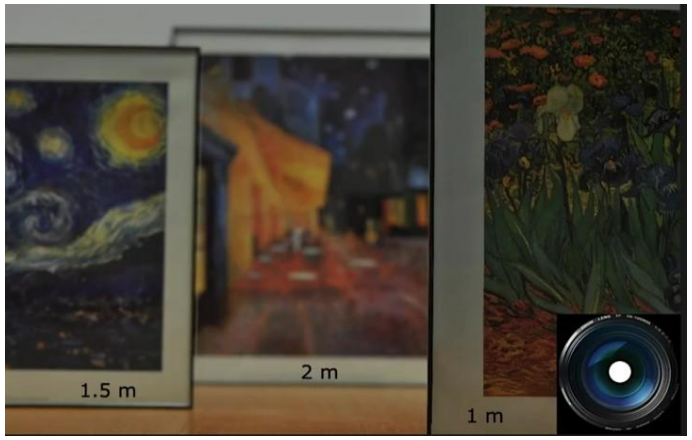
Aperture Size: DOF Vs Brightness

Here, we are using lens instead of pinhole camera because by using lens we can create the bright image.



Consider focal length f = 50mm, Aperture D = 25mm and N = 2

Consider focal length f = 50mm, Aperture D = 6.25mm and N = 8



Consider focal length f = 50mm, Aperture D = 12.5mm and N = 4

Lets consider 3 different images at different distance from lens. Where lens is focused at 1m distance.

Observations

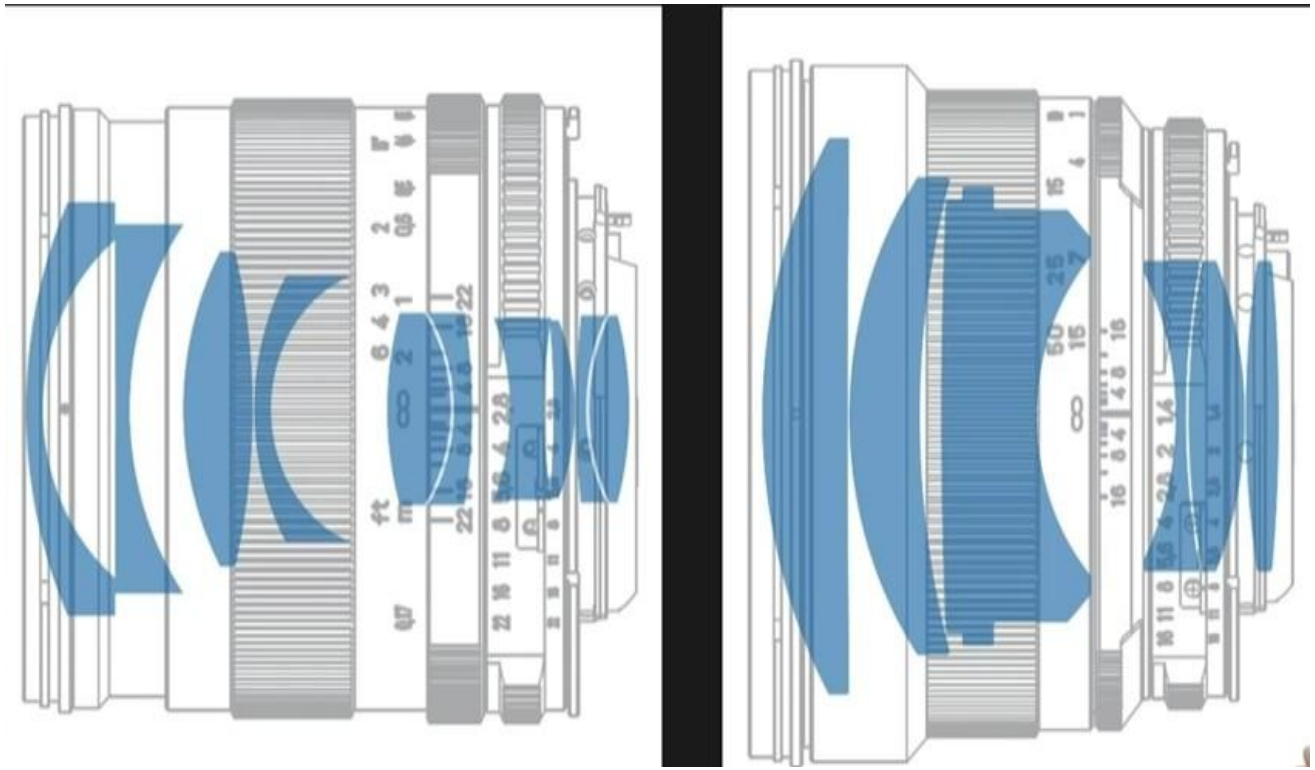
Large Aperture (Small f-Number)

- Bright Image or Short Exposure Time
- Shallow Depth of Field

Small Aperture (Large f-Number)

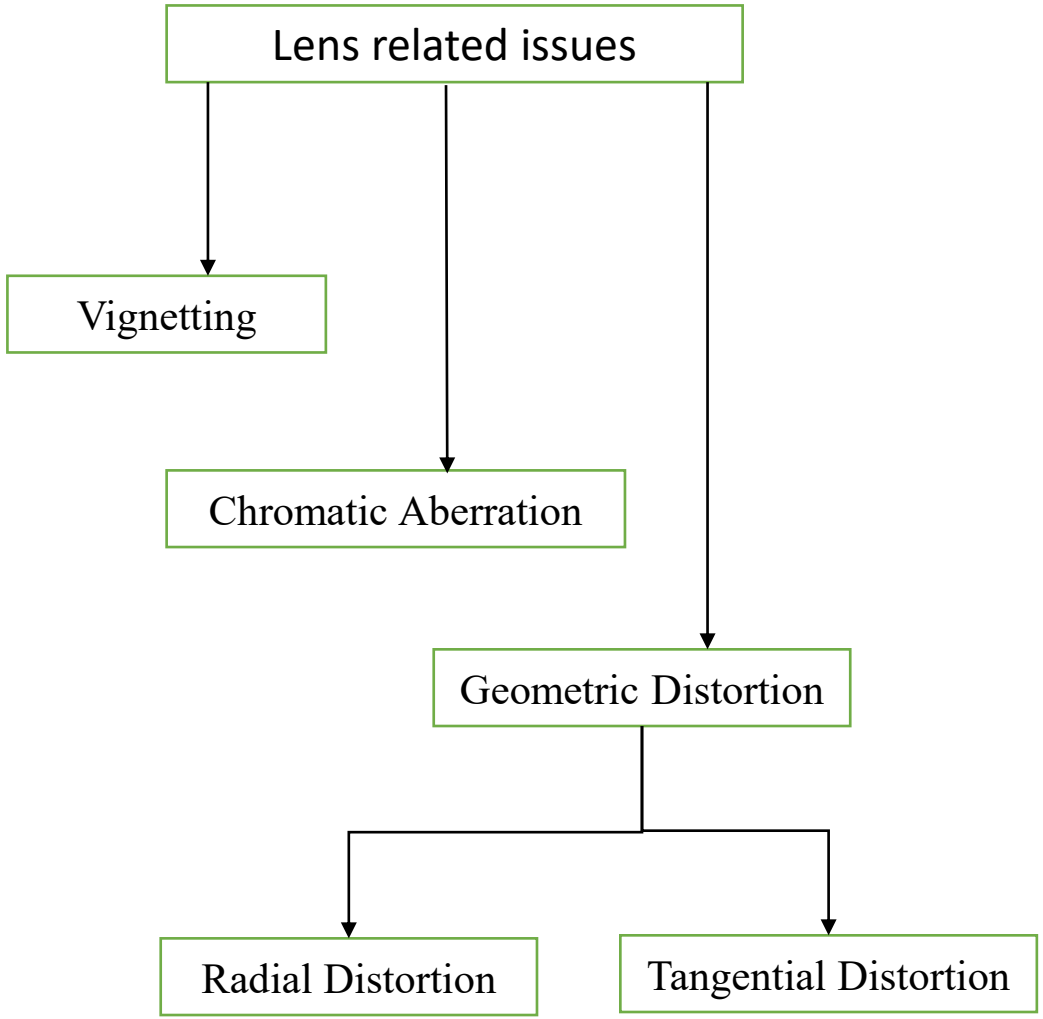
- Dark Image or Long Exposure Time
- Large Depth of Field

Lens related issues:



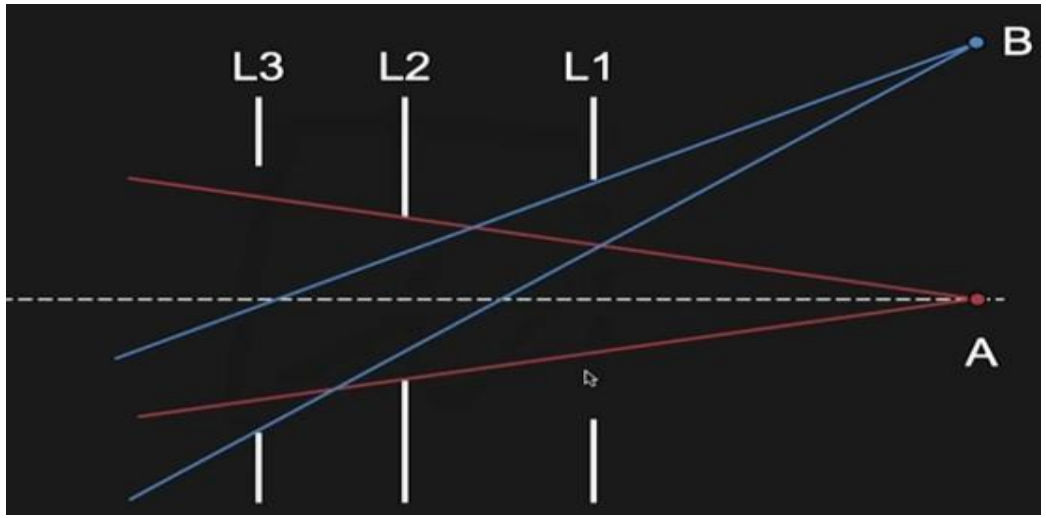
Till now we always talked about a single lens but practically, its compound of lens. Compound lens are used to compensate the undesirable effects of each other, so that quality can-not be compensated.

Design of lens = Art + science



Lens related issues

Vignetting



Here, we have three different lenses (L1, L2, l3) with different size and opening that allows different amount of light pass through it.

When you place a point on the optical axis i.e. point A, the amount of light that is reaching to the sensor is more as compared to point B (which is a point away from optical axis). Here, B is the point at the boundary of the lens.

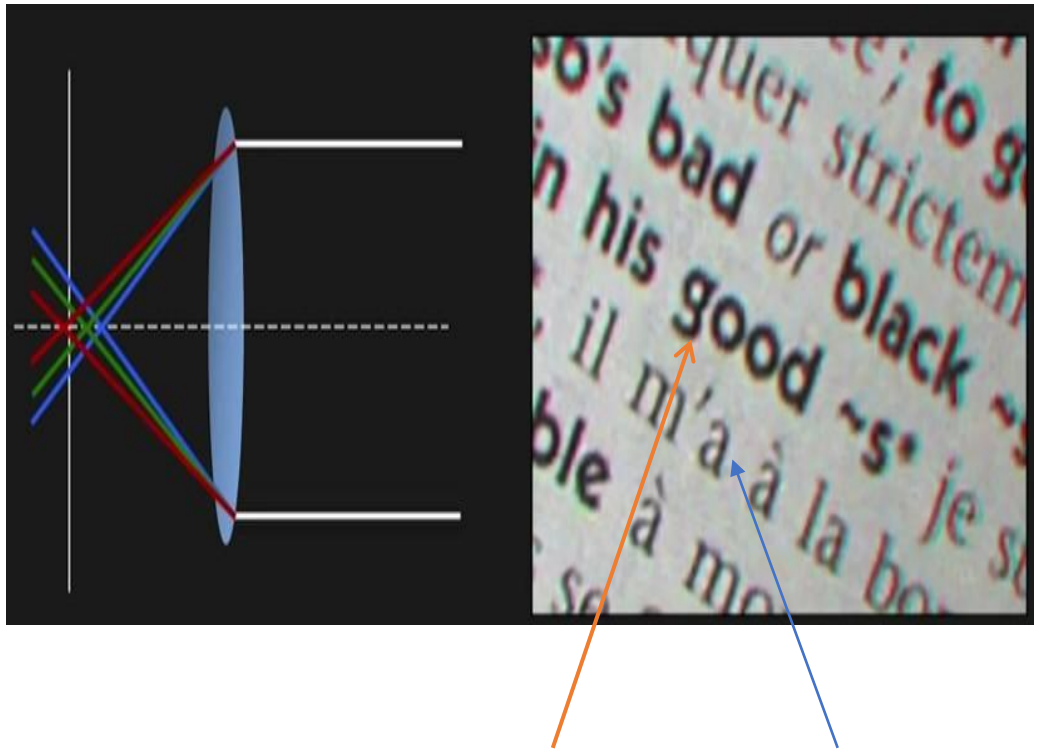
Due to this, the edges or corners of an image appear darker or less saturated than the center.

Darkening or fading of an image's corners.

Lens related issues

Chromatic Aberrations

When a lens fails to focus all colors of light to the same point. In a photograph, this looks like "color fringing".



Here, there is change in color

Lens made by certain material like glass or plastic.

↓

This material has some refractive index.

↓

Refractive index is function of wavelength
Refractive index = f (wavelength)

↓

Since R,G,B has different wavelength. Then, when white light is coming to the sensor then we can see blue bends more as compared to red.

↓

It directly implies blue color is focused at different point as compared to red color.

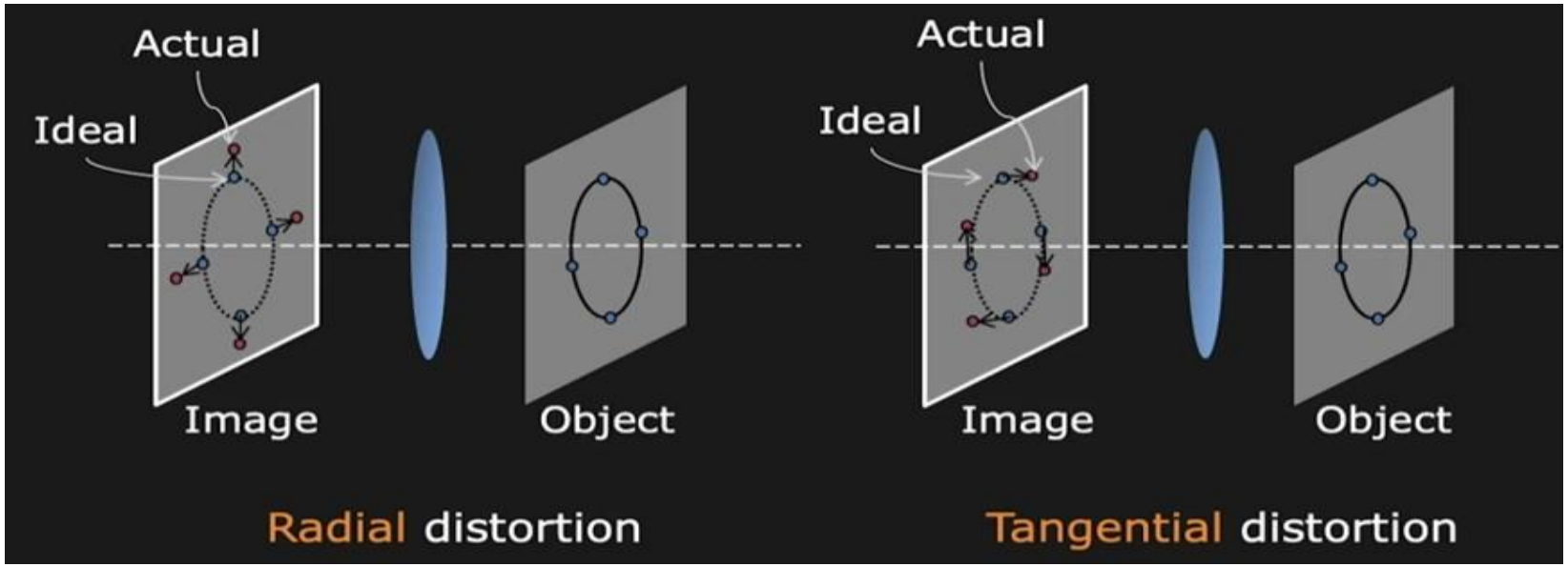
↓

As a result, there is some shift in the color in image that is noting but chromatic aberrations.

Lens related issues

Geometric Distortion

As you move away from the centre of the image Points tends to pushed Out more and more.



As you move away from the centre of the image twisting will increase.

Straight lines in the scene is not projected as straight line on the image plane.

Wide angle cameras

In case of wide angle camera, instead of reducing this kind of distortion it is always better to introduce it to get the more information or hemispherical view.

Consider a fisheye type lens that can capture the hemispherical field.

Here, Imaging system is looking up and with one shot you could able to capture the whole view.

There are two ways to do the same, one is by using lenses and another one is by using mirrors.

Now, we know how to capture the optical images. Next step is to convert **optical image to digital image.**

So that, we can provide it to computer vision system for further analysis.

Image Sensing: Converting light into electrical charges

Image sensors are made by silicon.

↓

When silicon atom hit by photon of sufficient energy.
It releases electron and a electron hole pair is generated.

It means silicon does most of the work for you.

↓

The challenging task is to read out these electrons and convert them into voltage. It will become more challenging when millions of pixel were present.

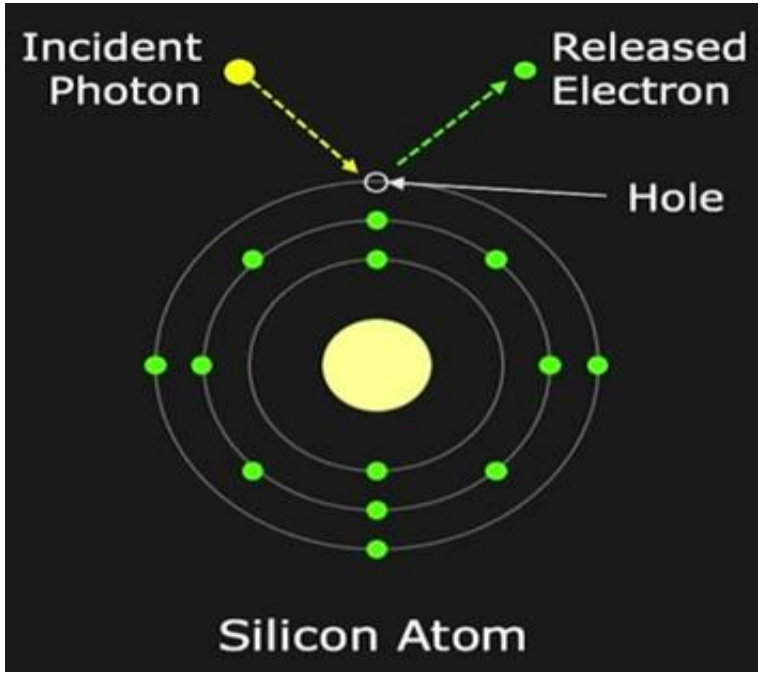
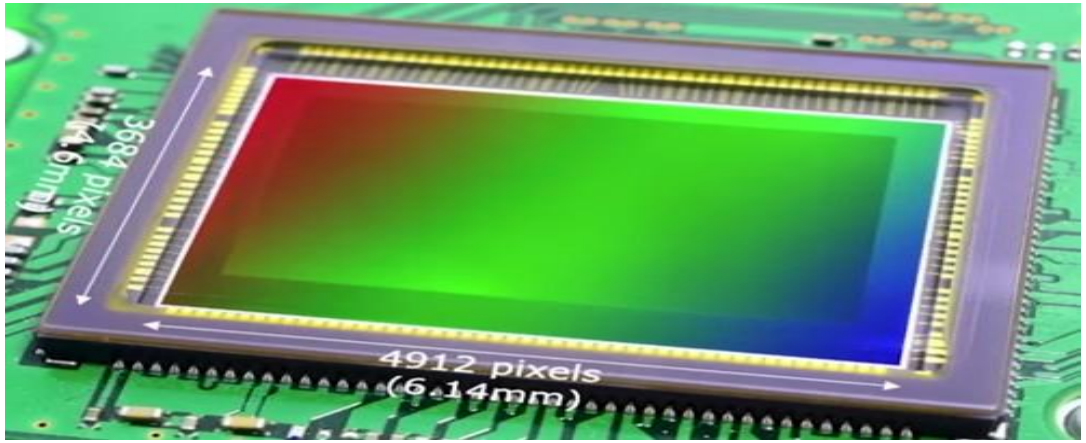
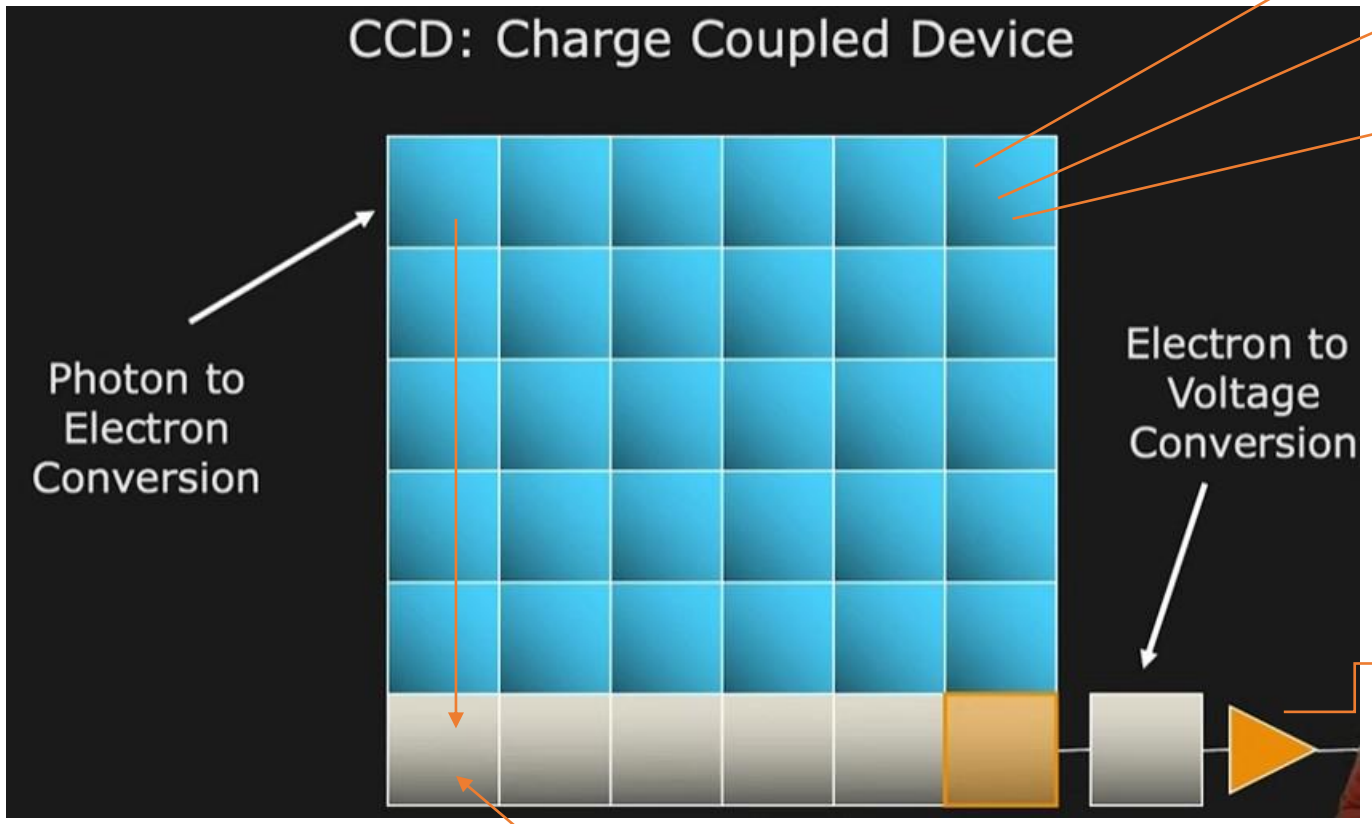


Image Sensor



Types of Image Sensors: CCD

Charge Couple Devices: CCD



Pixel

Potential well

Each pixel is looks like a bucket.

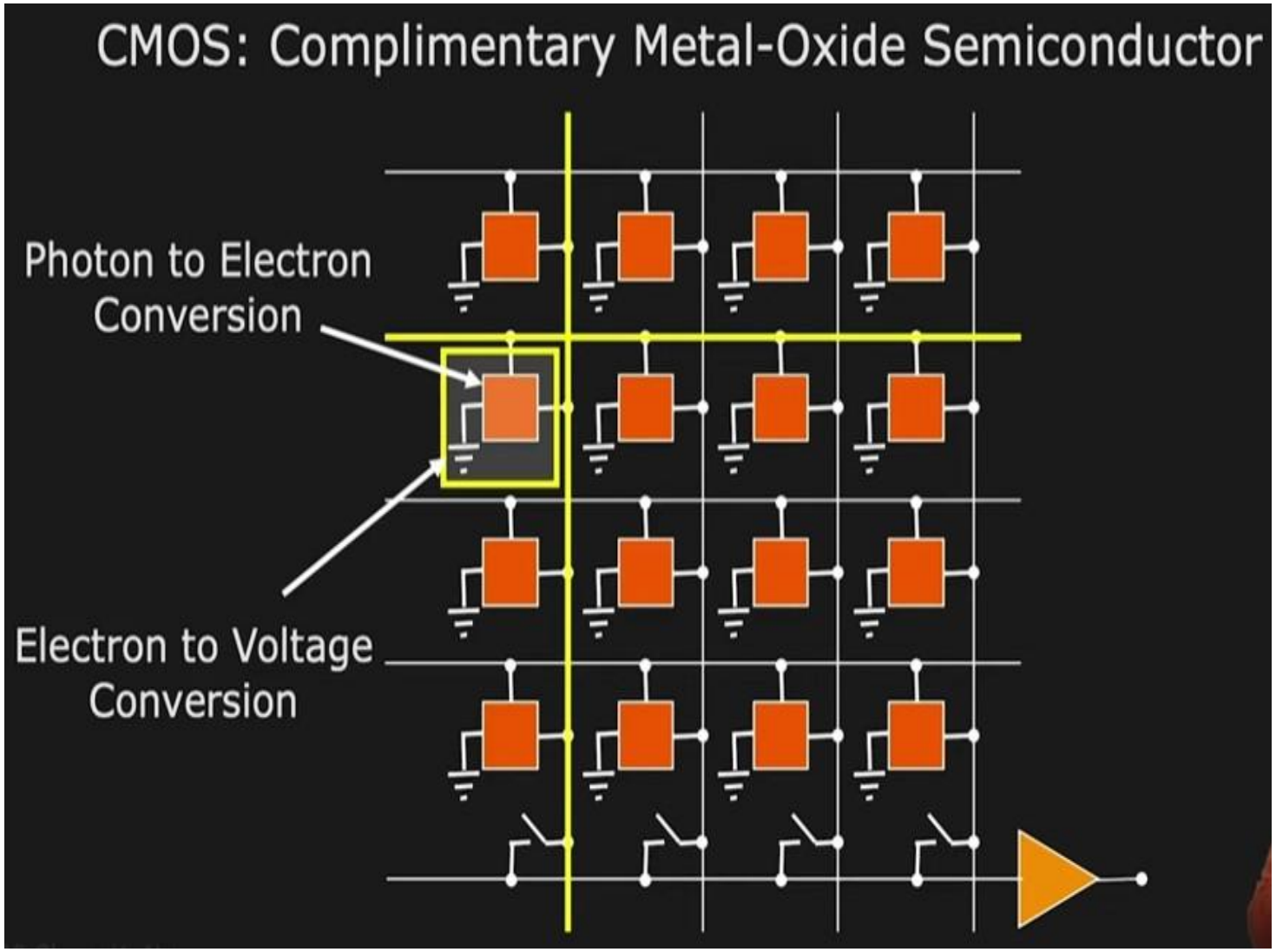
At the out we will get analog voltage then A to D convertor, convert it into digital output.

Each Row passes its electron to the next row. It called as bucket brigade.

Electric field applied beneath these bucket such that it will help to pass the electron row by row.

At bottom row, all the electron are collected and
Here, total electron presented in each pixel is converted into voltage one by one.

Types of Image Sensors: CMOS



Here each pixel has its own circuit while in CCD, one circuit is shared by the entire chip.

Next to photon to electron conversion there is a circuit which converts electron to voltage.

You can select any pixel and find its voltage.

It may be helpful when someone is interested into a particular region.

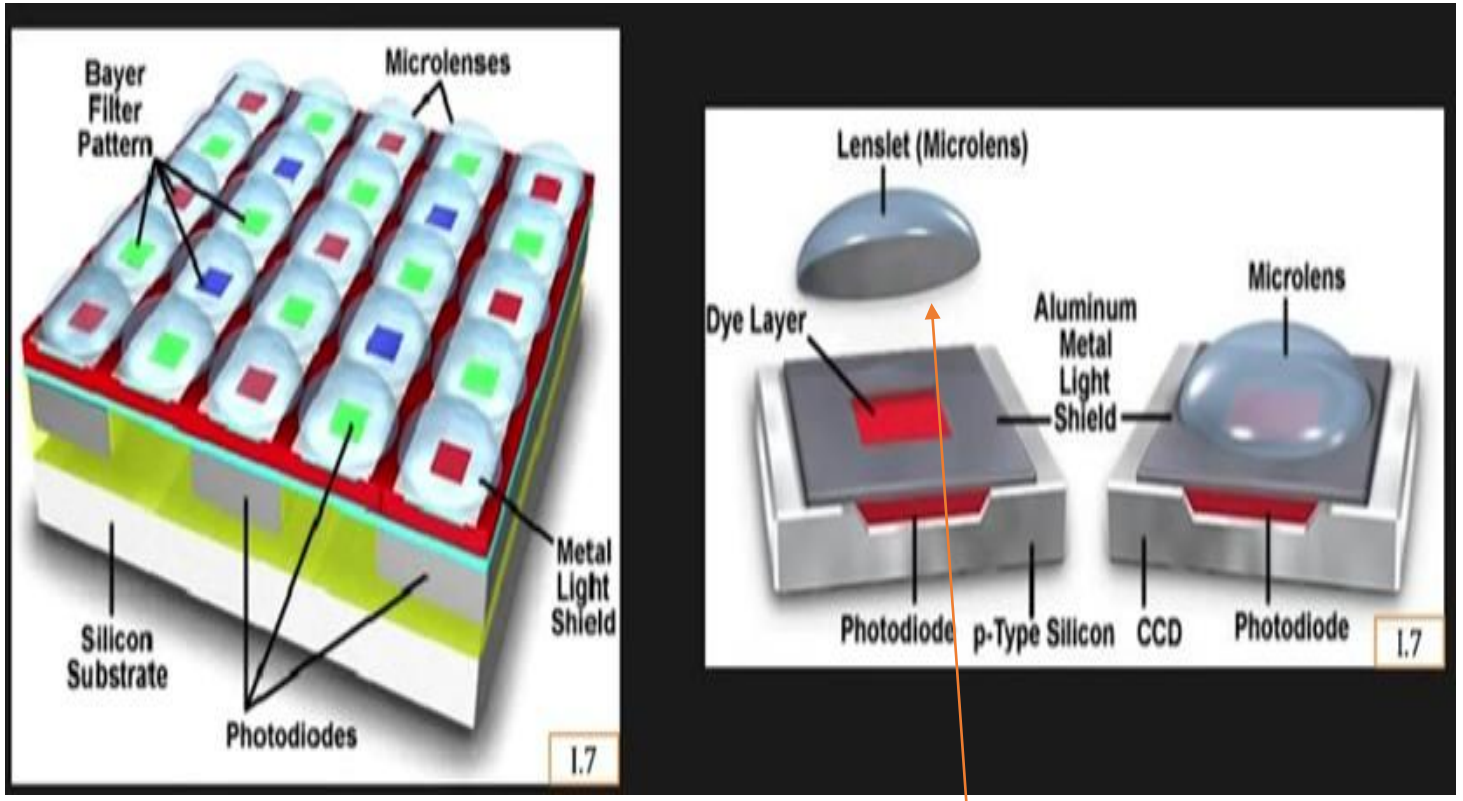
As the circuit is near to the sensor, a price need to Pay, which is total light sensitive area is smaller.

Types of Image Sensors: CMOS

Here, these are potential well.
Its array of pixels and we generally call them photodiodes.

On the top of each pixel there is a color filter.

Color filter is placed because a pixel does not know which color of light is coming, it just counting the photons.



On the top of each pixel there is a microlens
Which help to channelise all the light.

Color Filter Array (CFA) and Bayer Pattern

Color filter array (CFA)

1. Image sensors are naturally colorblind (they only measure the intensity of light not its wavelength).

2. Image sensors needs some filters to separate the light into its component color like red, green and blue.

3. CFA is mosaic of tiny filters placed over the individual pixel of image sensors and filter the light into R,G,B.

4. The most common CFA is **Bayer Pattern** developed by (Bayer 1976).

Bayer RGB Pattern

G	R	G	R
B	G	B	G
G	R	G	R
B	G	B	G

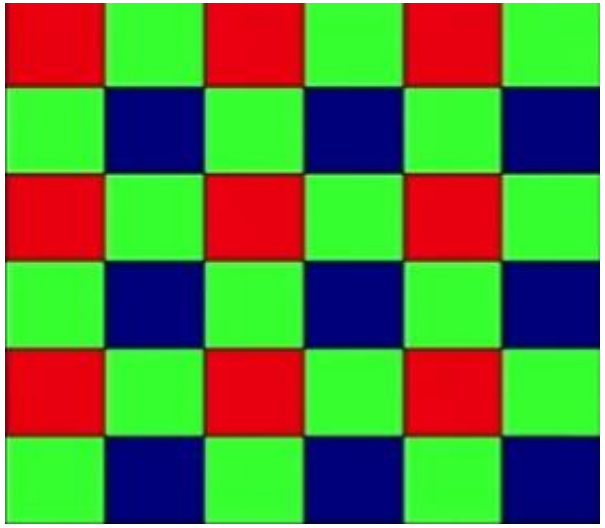
(a)
Color Filter Array (CFA) layout

rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb
rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb

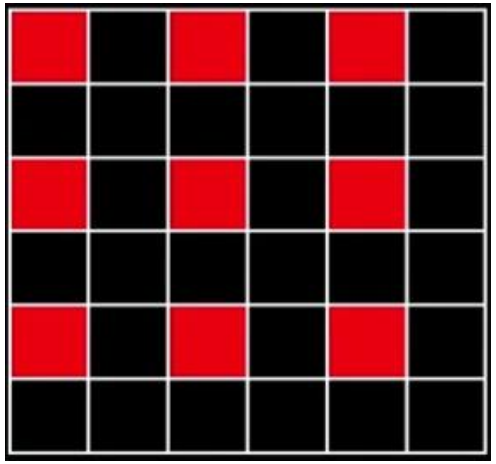
(b)
Interpolated pixel values with unknown (guessed) values shown as lower case.

4.1 It places green filters over half of the sensors and red, blue over remaining ones.
4.2 Our human eyes are more sensitive to the green color as compared to others, this is the reason to keep twice green pixel as compared to red and blue.
4.3 Because each pixel on the sensor only records one color (e.g., a "red" pixel only knows how much red light it hit but has no idea about the green or blue), the raw output of the sensor **looks like a dark, speckled mosaic**.
4.4 The process of interpolating missing color values so that we have valid RGB values for all the pixels is known as **Demosicing.**

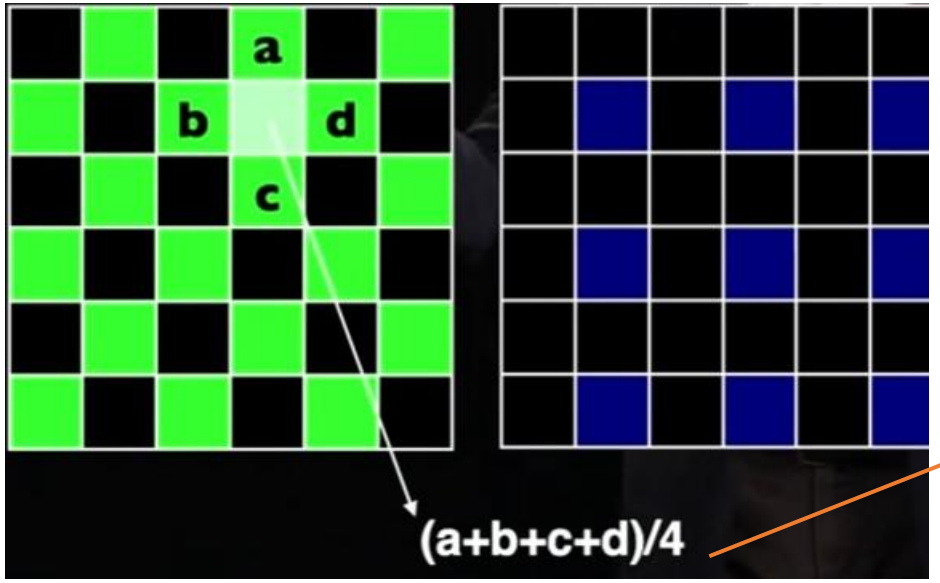
Color Filter Array (CFA) and Bayer Pattern



Red



Green

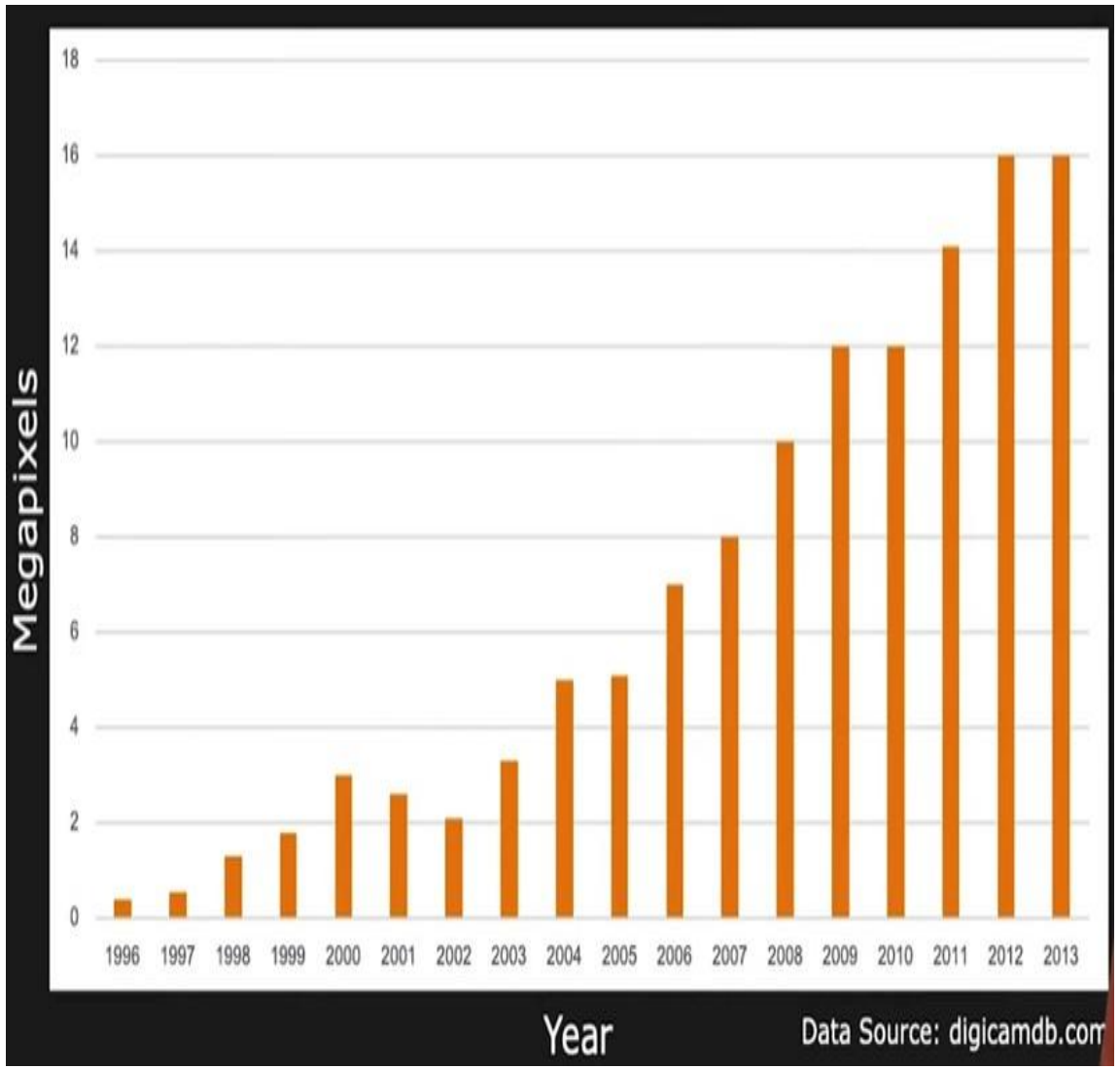


Blue

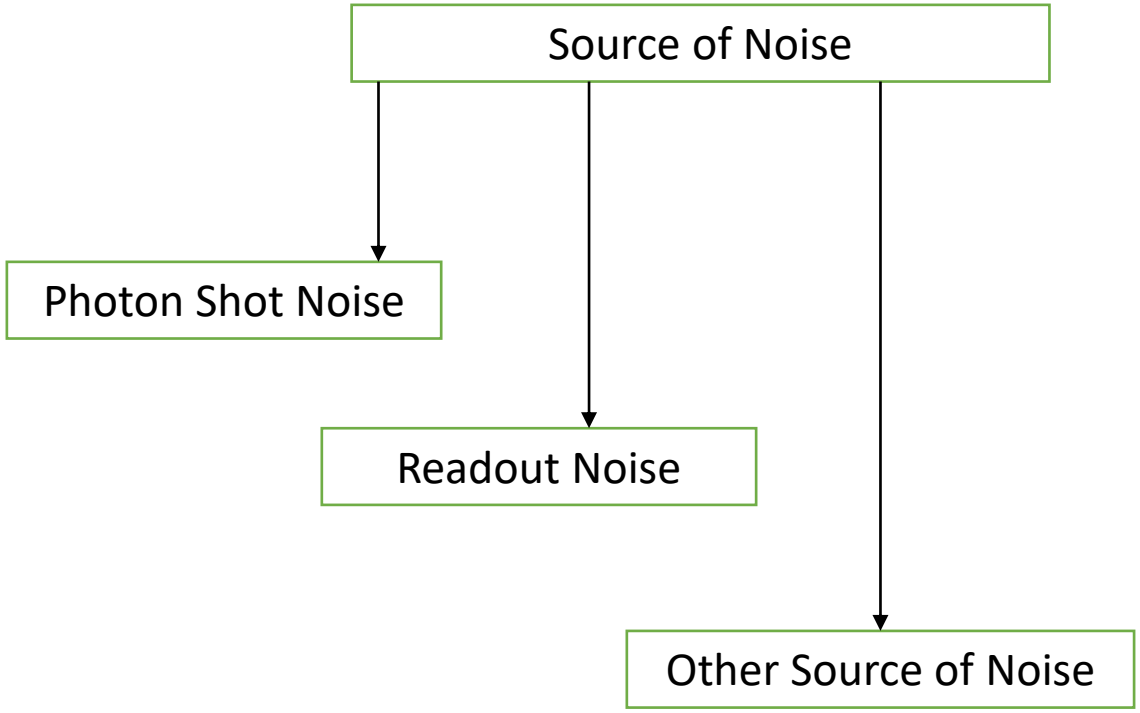
Interpolating the values.

Resolution, Noise and Sensor Dynamic Range

Resolution: Number of pixels are present inside the image.



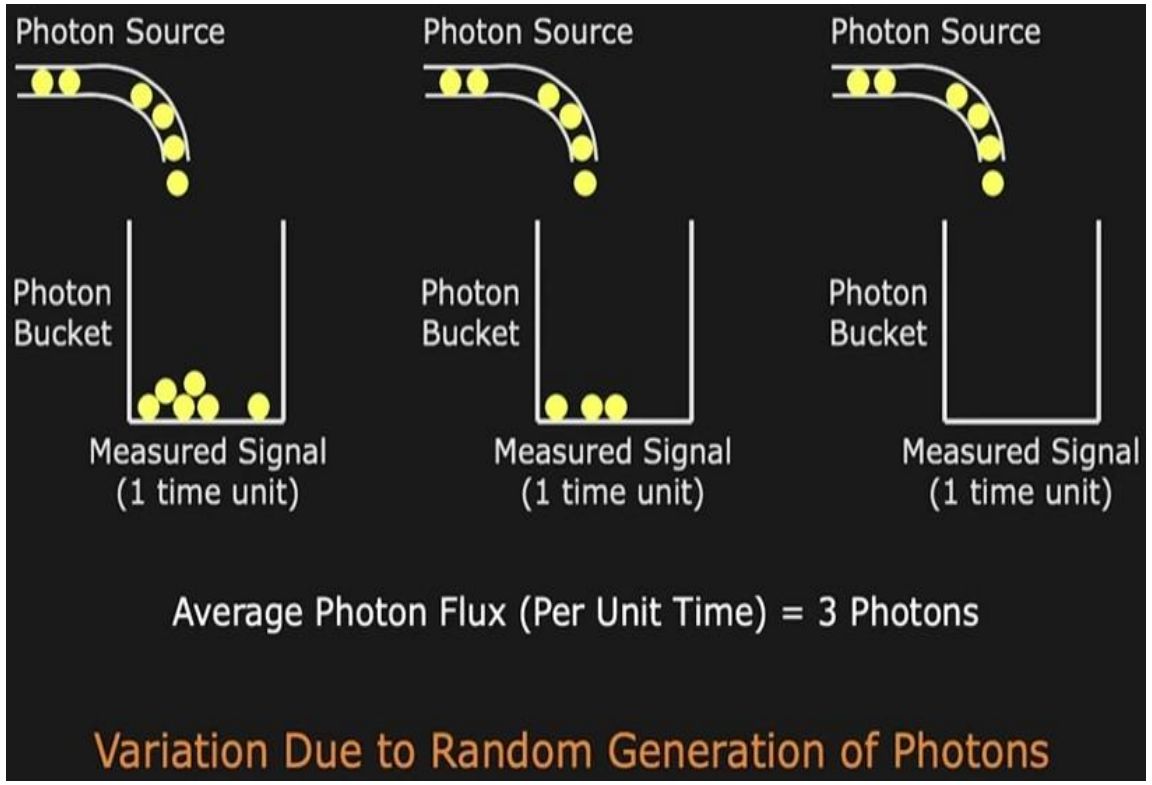
Noise: Unwanted modification of the signal during capture, conversion, transmission and Processing.

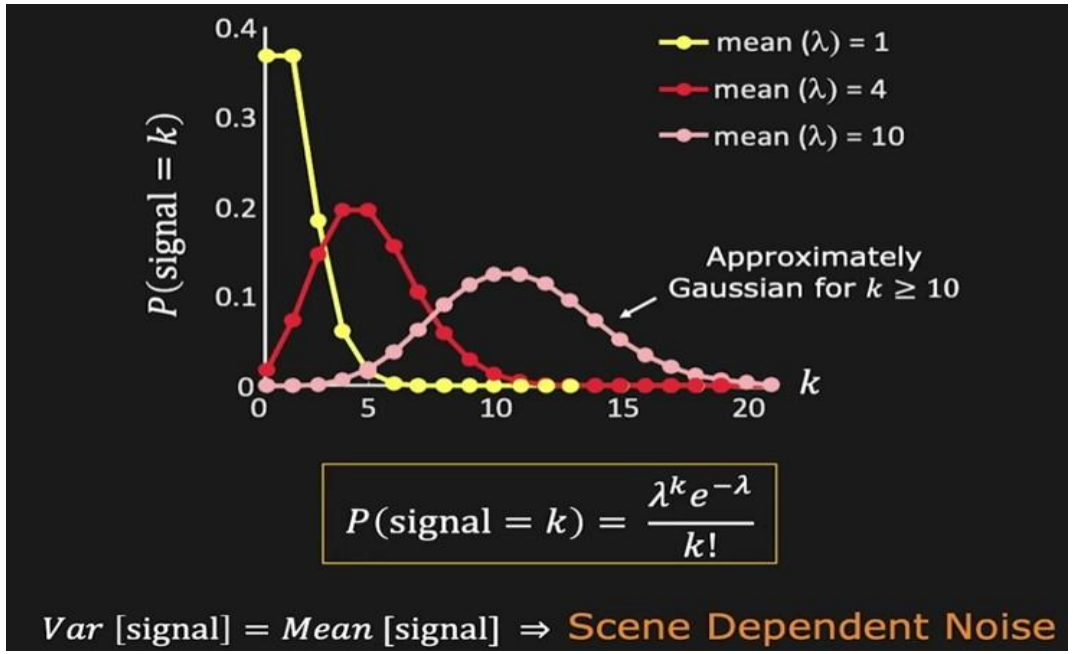


Resolution, Noise and Sensor Dynamic Range

Photon Shot Noise (Scene dependent noise)

1. It is due to random or quantum nature of the light itself.
2. Photons are arriving in a random way.
3. it is like raindrop is falling in the bucket and you are not going to get uniform distribution both in time and space.
4. some time they may arrive faster sometime slower and due to it there is some noise is introduced in image i.e. photon shot noise.





One can model this shot noise by poisson distribution, here the mean value that we are trying to get. This mean values is nothing but the brightness of the scene point.

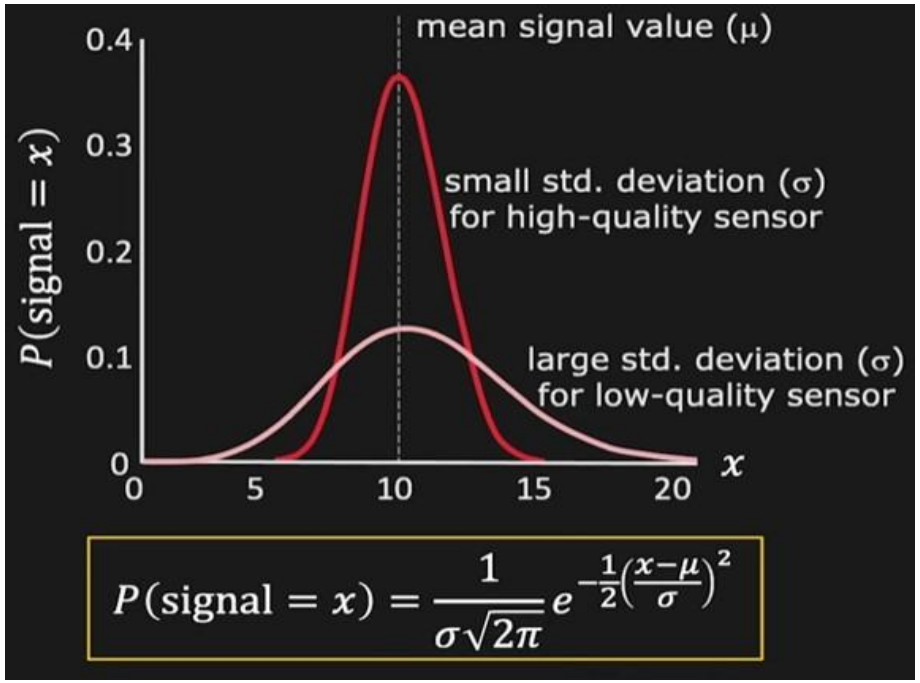
Resolution, Noise and Sensor Dynamic Range

Readout Noise (Scene Independent noise)

- Electronic Noise
- Quantization Noise

Now, photons are arrived, they need to convert into electron. Now we need to read it out using technique like CCD or CMOS.

Now, you want to compare electron into voltage. During that conversion process that circuit itself introduced noise i.e. before analog to digital conversion.

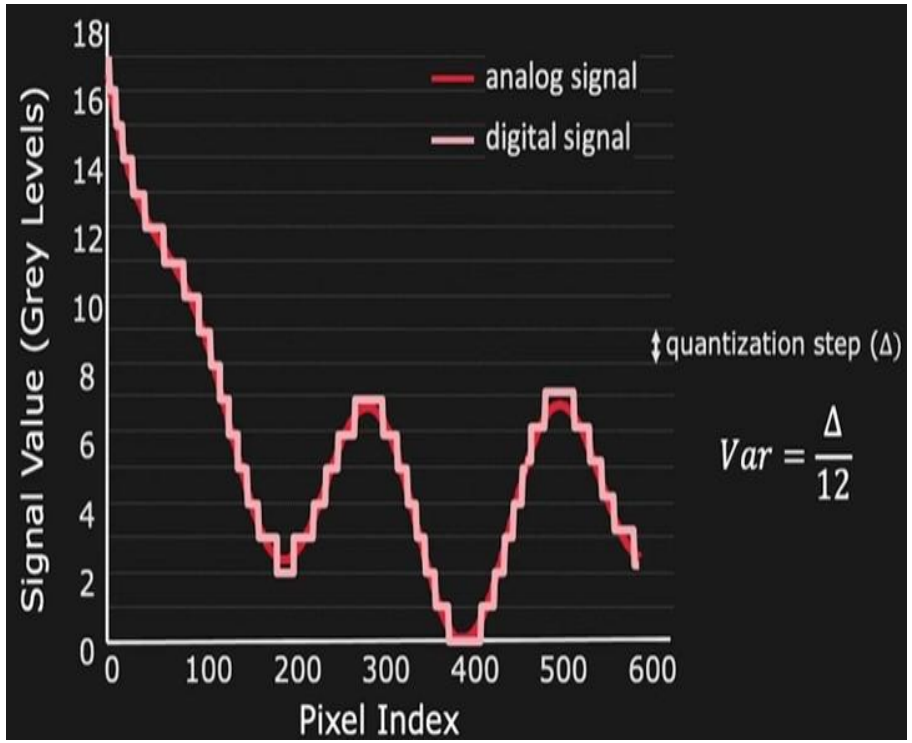


This noise is often modeled as gaussian noise. It has mean, that we are going to measure.

High quality sensor - less spread in distribution.

Low quality sensor – large spread in distribution.

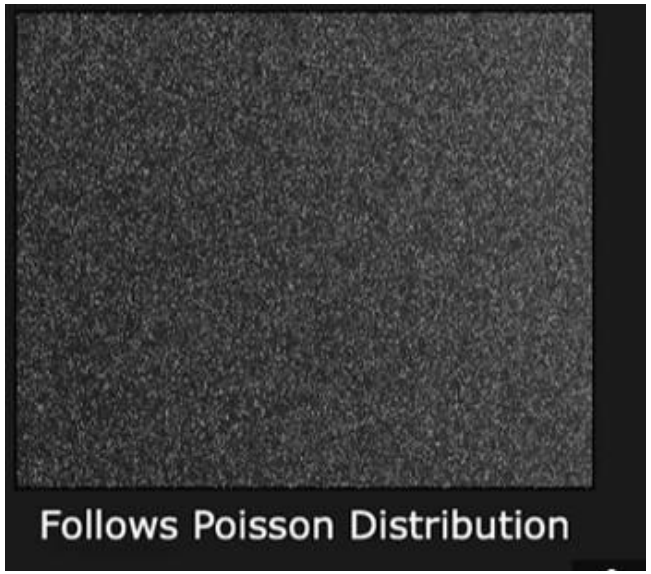
During Analog to Digital conversion, some noise is introduced.



Resolution, Noise and Sensor Dynamic Range

Other source of noise (Scene Independent Noise)

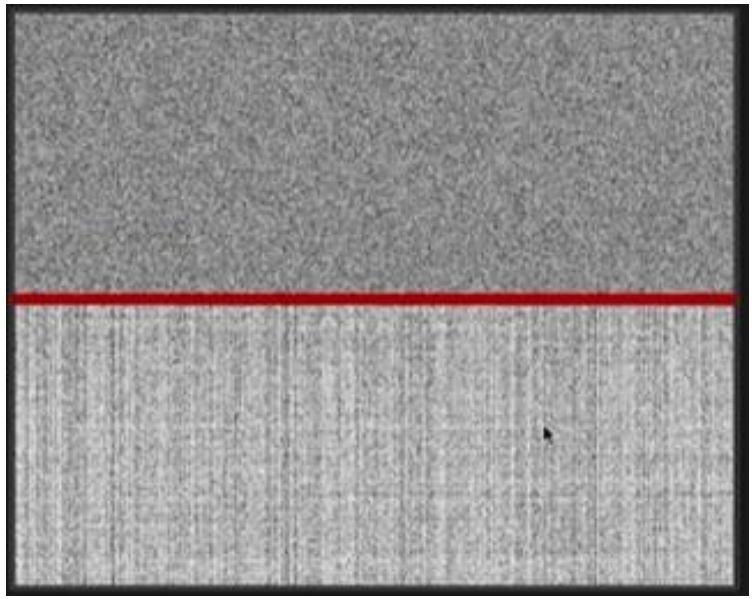
Dark Current Noise or thermal noise



If you keep a lens cap (it means no light is coming to the sensor) Then also, after capturing the photo you will end up with non zero pixel values, it is because some of the electron are ejected or released due to temperature itself. that is known as thermal noise or dark current noise.

Fix Pattern Noise

Random noise



No two pixels are identical there are some Variations due to manufacturing. Like -

1. Slight variation in the response and characteristics of the pixels
2. Variation in the well size.
3. Response from pixel during conversion.

Camera response and HDR imaging

$$\text{Dynamic Range} = 20 \log \left(\frac{B_{max}}{B_{min}} \right) \text{ decibels (dB)}$$

B_{max}: The maximum possible photon energy (full potential well)

B_{min}: The minimum detectable photon energy (in the presence of noise)

Sensor	$B_{max}:B_{min}$	dB
Human Eye	1,000,000:1	120
HDR Display	200,000:1	106
Digital Camera	4096:1	72.2
Film Camera	2948:1	66.2
Digital Video	45:1	33.1

Each pixel has a potential well.

↓

When that potential well fills up.

↓

Any photon that come in after that simply cannot measured, so there is an upper limit to the brightness level that you can measure.

Sensing Color

Color:

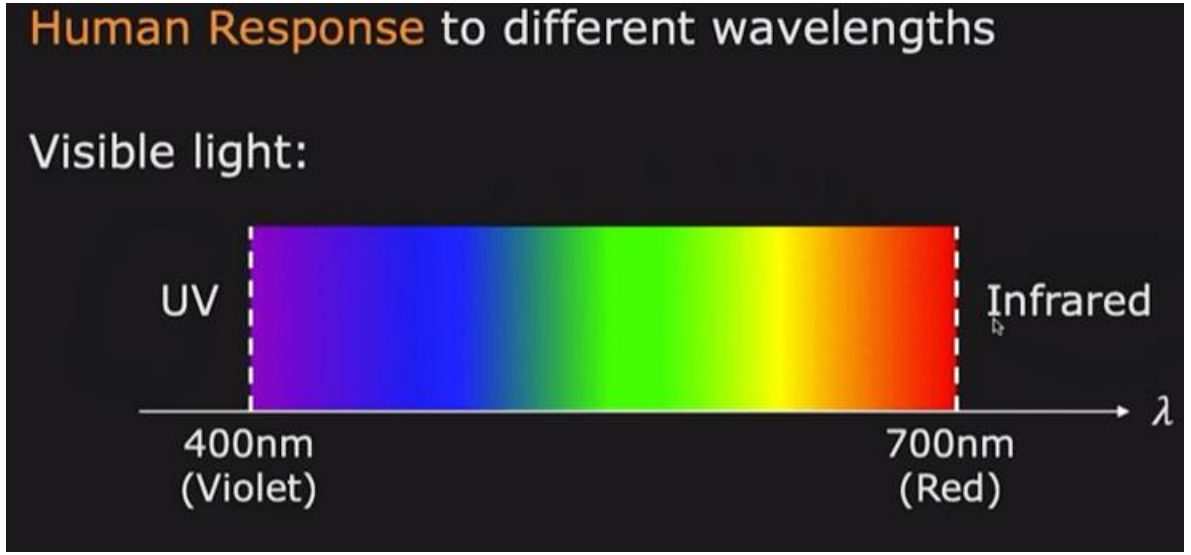
1. It is not a physical quantity that you can measure.

2. Incoming light = function (wavelength)

3. Its human response to different wavelength.

4. Our eyes can respond between 400nm to 700nm.

5. One can design a computer vision system, that can able to sense UV and IR also. (An application of CV)



Mixing of colors: As per the youngs experiments, only three colors are enough to generate all the colors present in the image.

Camera Response Function

Camera Response Function (CRF) -

Scene	Image
1. A point in a scene has a certain brightness	1. Same point in an Image has a certain brightness.
2. If we increase the brightness of a point in a scene.	2. It is assumed that brightness in image will also increases.

It means the relationship between them is Monotonic.

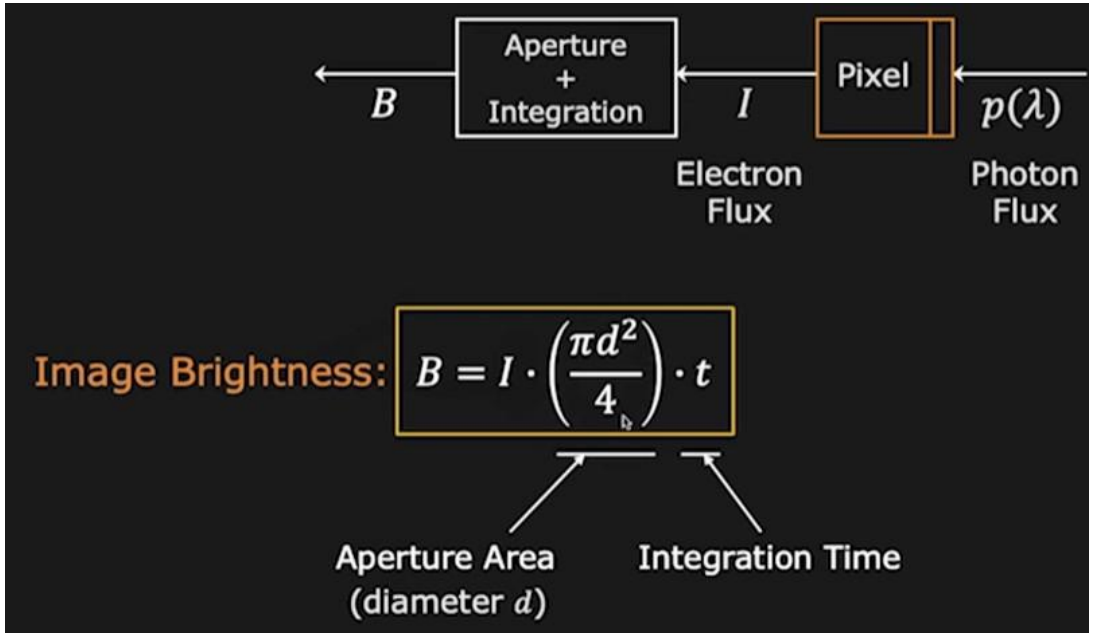
↓

But in actual scenario it is non-linear.

↓

Camera response function shows the non-linear relationship between scene brightness and image brightness.

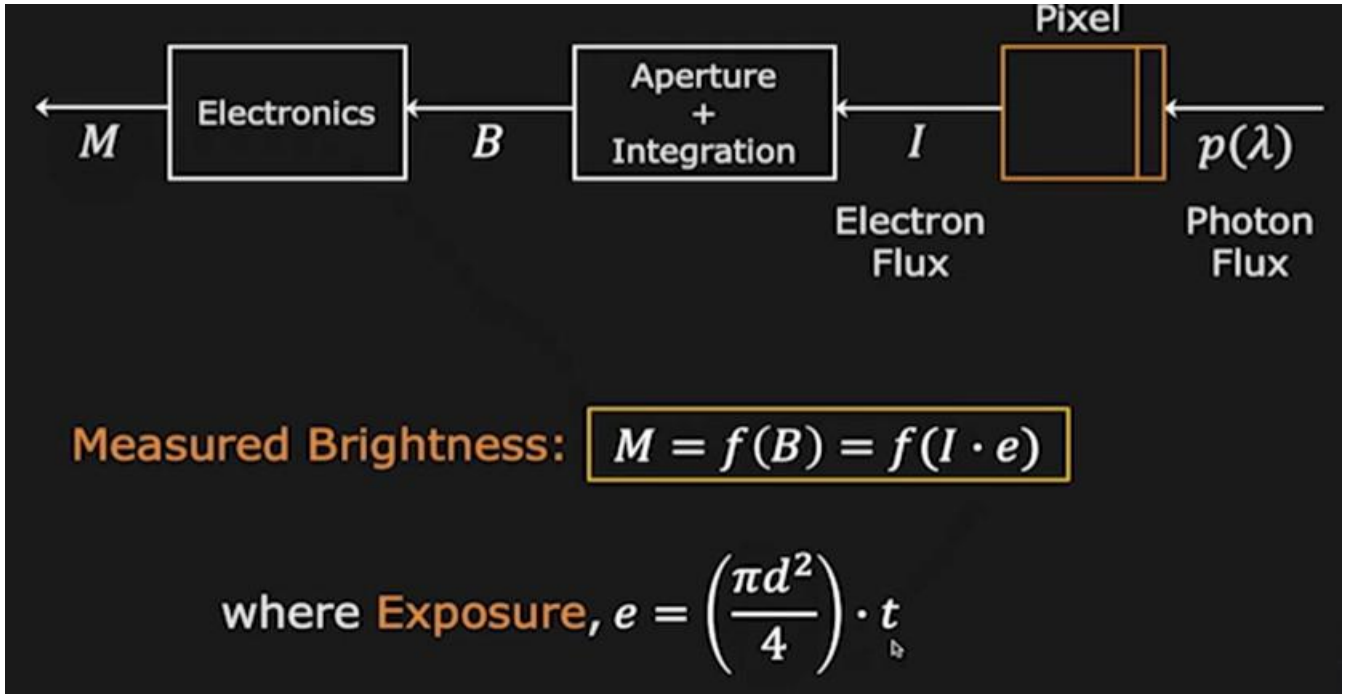
Camera Response Function



Photon flux (p) is going inside the pixel and that pixel produces the electron flux (I).

Image Brightness (B):
Electron flux (I) is modulated by aperture of the lens and the integration time of the scene.
Note:

Aperture area + Integration time = Exposure

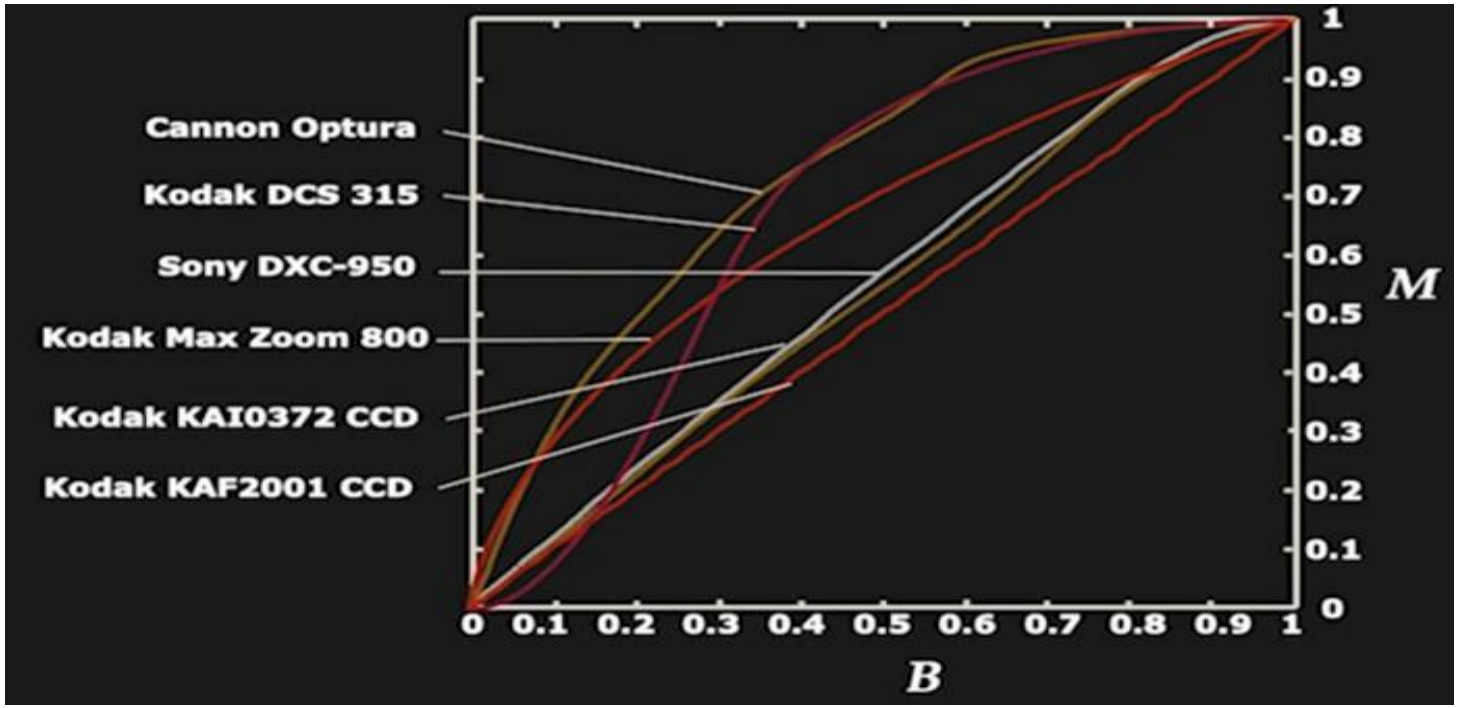


Till actual brightness (B) everything is linear. However this brightness goes through some electronics for electron to voltage conversion.

Here M shows the some non – linear relationship between measured brightness (M) and actual brightness (B).

Camera Response Function

What kind of camera response function do we find in practice ?



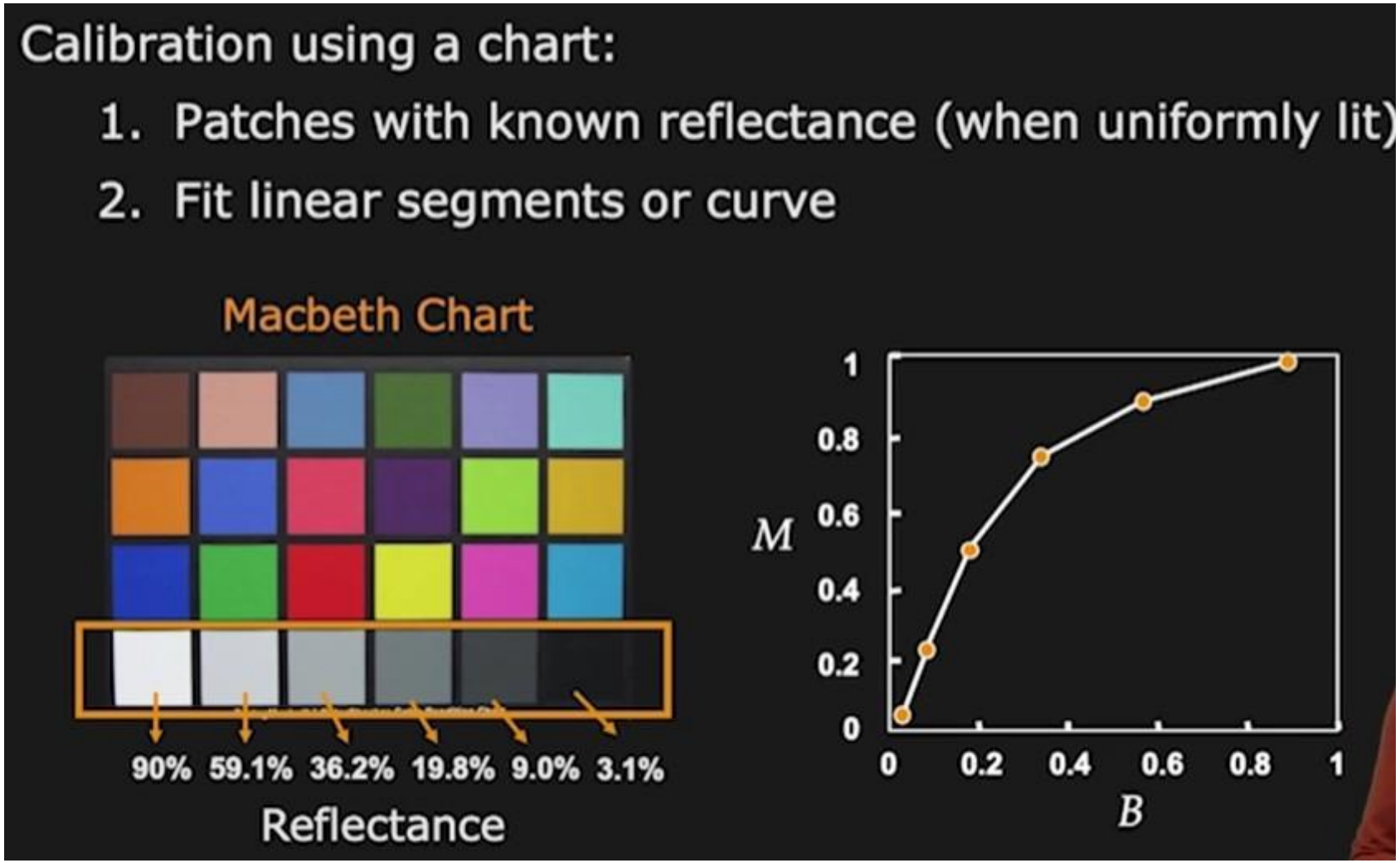
Why camera manufacturer introduce this non linearity between measured brightness and actual brightness ?

1. You have finite dynamic range in any given camera.
2. You may want to compress certain brightness value as compared to other brightness value

Camera Response Function

How we can find the camera response function of a given camera. ?

We can use the calibration chart.



Here
B is actual brightness.
M is measured brightness.

High Dynamic Range Imaging

1. **What** is Dynamic Range of a Sensor - It is the range of brightness values that it can measure.

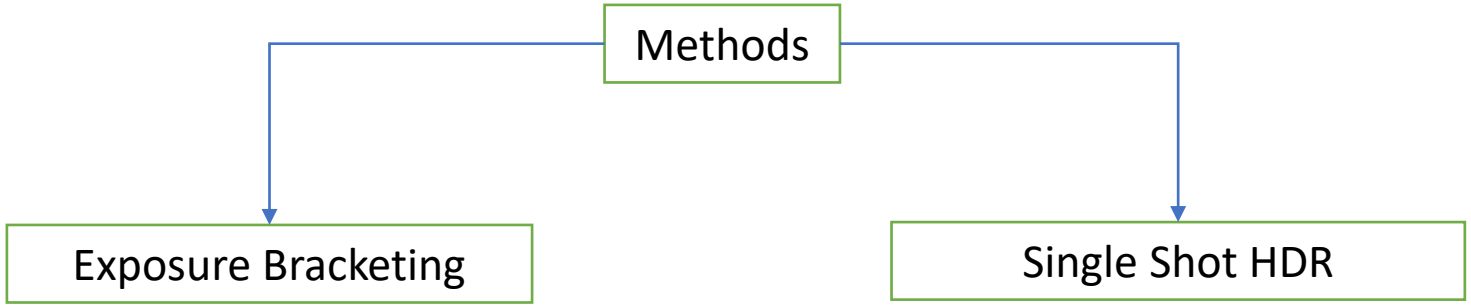
2. **Why** we need to increase the Dynamic Range of a sensor?

In real world, there is enormous amount of brightness are available and their is no camera that can capture it with high fidelity.

For example - Brightness of sky and the details present in the shadow region of a given scene can not be measured by state of art camera with limited dynamic range.

To capture all these information with high fidelity, we need to increase the Dynamic range of the sensor.

3. **How** to increase the dynamic range of the sensor ?



$$\text{Dynamic Range} = 20 \log\left(\frac{B_{max}}{B_{min}}\right) \text{ decibels (dB)}$$

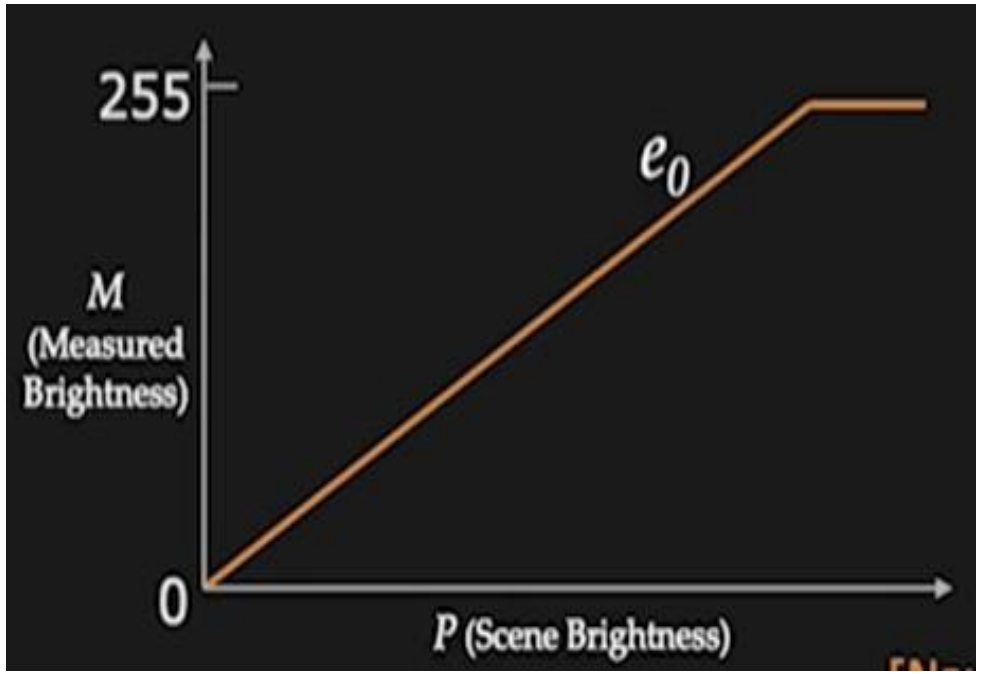
B_{max}: The maximum possible photon energy (full potential well)

B_{min}: The minimum detectable photon energy (in the presence of noise)

Sensor	$B_{max}:B_{min}$	dB
Human Eye	1,000,000:1	120
HDR Display	200,000:1	106
Digital Camera	4096:1	72.2
Film Camera	2948:1	66.2
Digital Video	45:1	33.1

High Dynamic Range Imaging

Camera response function shows the non-linear relationship between scene brightness and image brightness.



Camera Response Function

Lets assume

it's a camera response function.

camera produces 8-bit of information per pixel.

So the maximum value that it can produce which corresponds to the full capacity of the pixel i.e. 255.

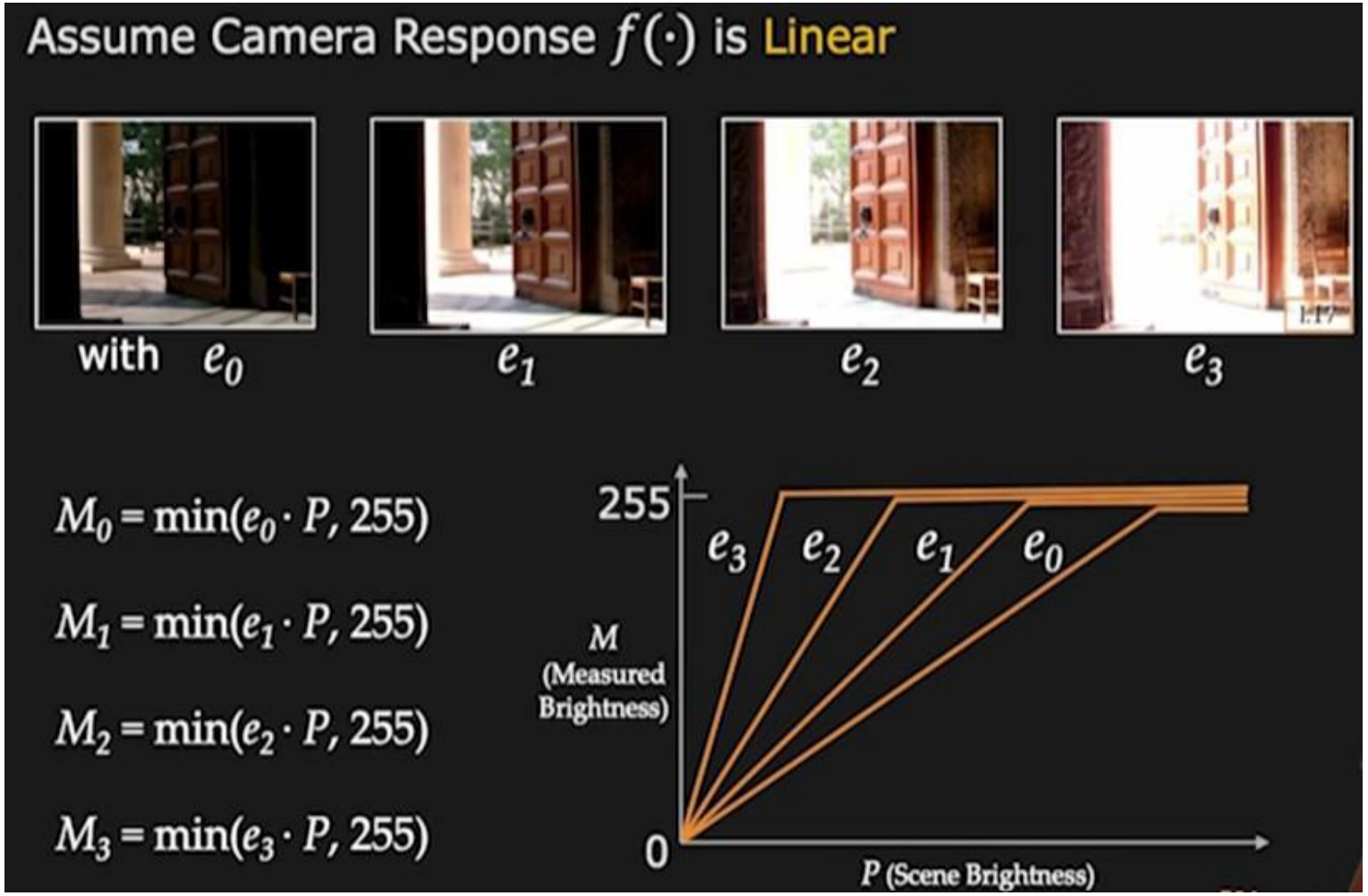
Now, even though if you will increase the scene brightness the Measured brightness can not exceed then 255.

So how to increase the dynamic range from 255 to 1020 (assume.)

Here, e0 is exposure i.e. nothing but multiplication of aperture area and integration time.

High Dynamic Range Imaging

Exposure Bracketing is a photography technique where you capture a series of three or more shots of the exact same scene, each with a different exposure setting.



Here,
eo, e1, e2, e3 are camera response fun.

As it is moving from eo to e3 measured brightness will increases.

eo = short exposure i.e. less light gathered in
i.e. less brightness i.e. high shutter speed.

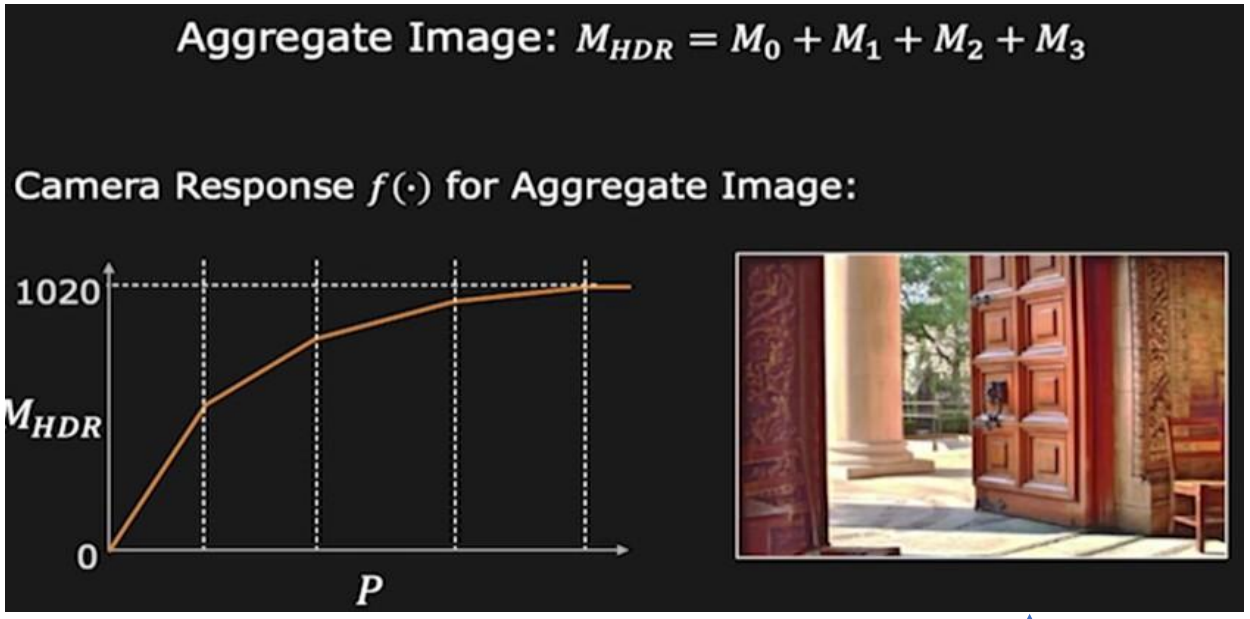
e1

e2

e3 = long exposure i.e. more light gathered in
i.e. high brightness slow shutter speed.

High Dynamic Range Imaging

By adding the all 4 response functions, at the end you will end up with a response function which has the dynamic range i.e. 1020.
By exposure bracketing, one can increase the dynamic range of the camera from 255 to 1020.



Ist image = 255
2nd image = 255
3rd image = 255
4th image = 255

1020

No one can capture the above Image contains very minor details with a camera has dynamic range of 1020.

It looks like image is captured by any Other camera.

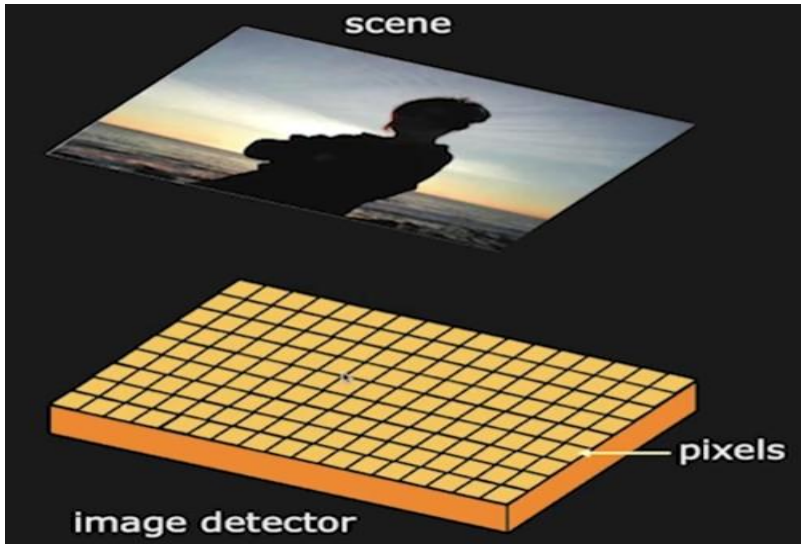
Problems with exposure bracketing

If objects is moving then there are some ghosting effect in the output image.

Note:
Exposure bracketing is an excellent method When the scene is static.

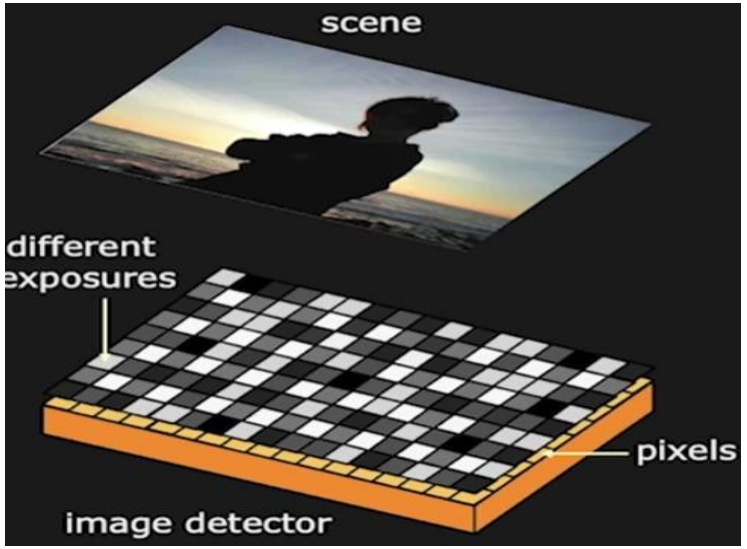
High Dynamic Range Imaging

Single-Shot HDR is a technique where a camera captures high dynamic range information in a **single shutter release** (one "click") instead of taking multiple separate photos.



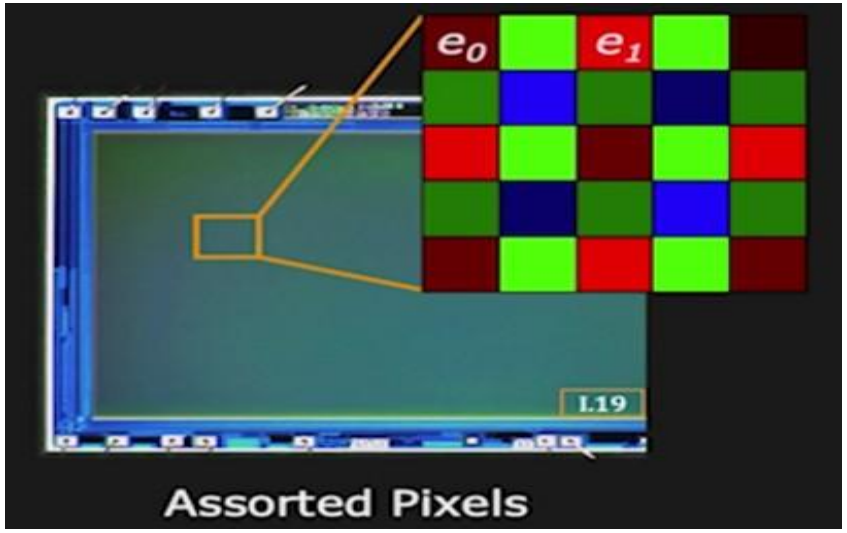
Here, all the pixels has equal sensitivity to light.

If you tune the exposure then either you can see the foreground or you can see the background (or vice versa.)



Create an image sensors with inequal sensitivity to light.
One way to do the same is by placing a little shade on the top of each pixels and you can use many different types of shades.

Here. The white pixel is very well exposed, lets say it get saturated. Its neighbour may not because there are other sensitivity around it.



Assorted pixel image sensors –
Here different pixels are has different color filter as well as it associated with different exposure.

It helps to create HDR with single shot.

CSET340

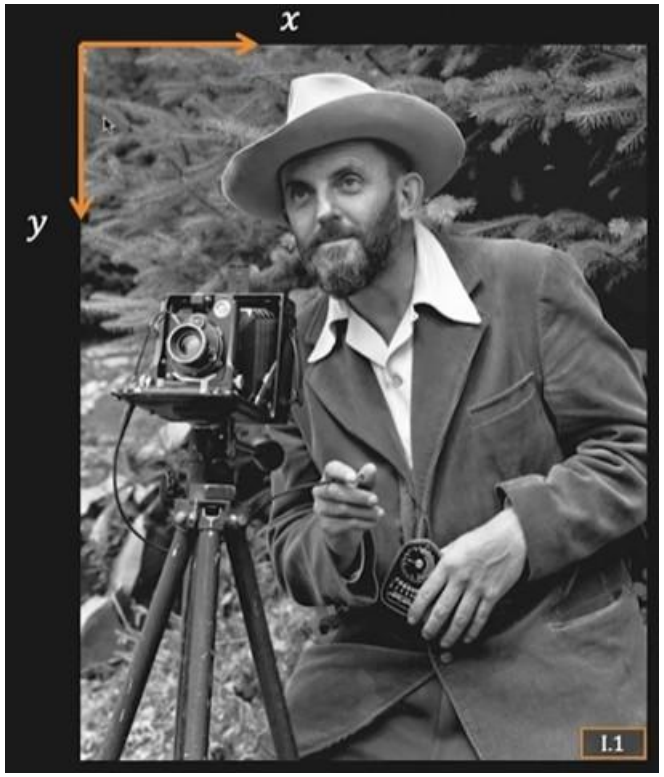
Advanced Computer Vision and Video Analytics

26th Jan. to 30th Jan. 2026

Overall Course Coordinator-
Dr. Gaurav Kumar Dashondhi
Gaurav.dashondhi@bennett.edu.in

Note : Any query related to course then first connect with overall course coordinator.

Pixel Processing



F(x,y) is the image intensity at position (x,y).

Take an image.

↓

Reach out to each pixel in the image.

↓

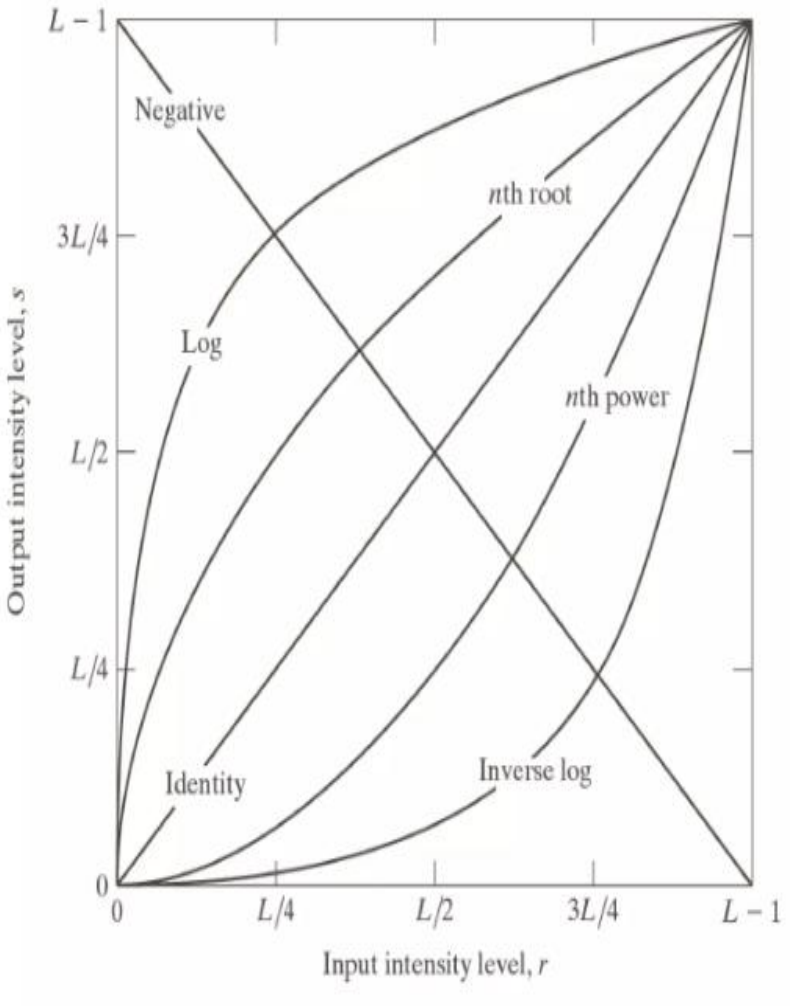
Look at the brightness or color values.

↓

Transform it, using predefined mapping

- Image Negative
- Log and Exp.
- Gamma Corr.

Note: Intensity based Transformations are not concerned about the pixel position in the image.



Basic intensity transformation function

Pixel Processing

Spatial Domain

Pixel processing or Intensity Transformation

Spatial Filtering

	P(x,y)	

(x-1,y+1)	(x,y+1)	(x+1,y+1)
(x-1,y)	p(x,y)	(x+1,y)
(x-1,y-1)	(x,y-1)	(x+1,y-1)

Intensity transformation are such kind of approaches where results are depend on the intensity at a particular point. i.e. not considering neighbours.

S = T(r), where S = output intensity.
T = Transformation function.
r = input intensity.

Smallest possible neighbourhood is 1x1, where output of transformation function depends only on a single point or single pixel.

It is applied on group or neighborhood of central pixels. These operations taking care of neighborhood or considering the neighbours.
Operations like –
Image Smoothening,
Image Sharpening, etc.

Pixel Processing: Intensity Transformation: Image Negative

Motivation : These kind of transformation is used to enhance white or grey level information embedded in the dark region of an image or it is required when black area is dominant in size as compared to white region.

$S = L - 1 - r$

where L = Maximum Intensity Level

r = Input Intensity Level

S = Output Intensity Level

Consider 8-bit (0 - 255). Where L = 255.

r	S = L-r	**S = L-1-r**	S = L-2-r	S = L-3-r	S = L-4-r
10					
20					
30					
40					
50					

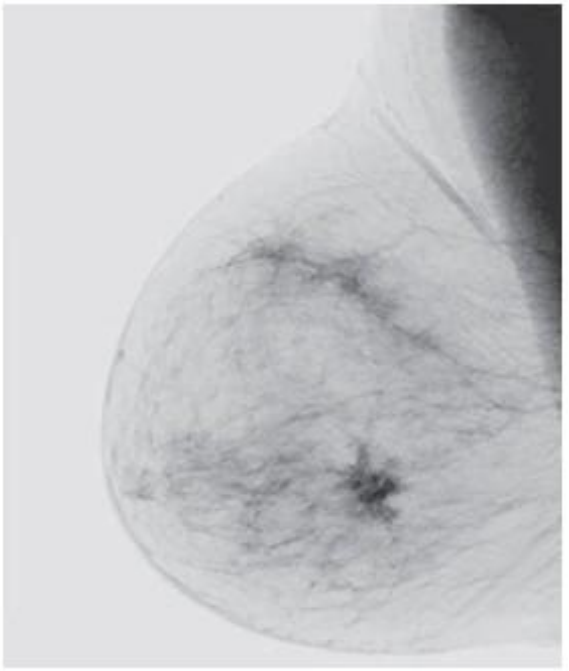
Pixel Processing: Intensity Transformation: Image Negative

r	S = L-r	**S = L-1-r**	S = L-2-r	S = L-3-r	S = L-4-r
10	**245**	**244**	**243**	**242**	**241**
20	225	224	223	222	220
30	215	214	213	212	210
40	205	204	203	202	201
50	**195**	**194**	**193**	**192**	**190**

Note: As we are moving from L-r to L-4-r then dynamic range of output image is reducing.

Application

Original mammogram of an image



Negative of same image

Pixel Processing: Intensity Transformation: Log Transformations

Motivation : These transformation used to expand the dark pixel in an image while compressing the higher-level values. This transformation maps low intensity values in input to a wider range of output levels and conversely, higher values in input levels are mapped to narrower range in output. (opposite is true for inverse log or exponential.)

$S = c\log(1+r)$, where L = Maximum Intensity Level

r = Input Intensity Level

S = Output Intensity Level, C = constant. $C = L-1/\log(1+r_{max})$

For an 8-bit image, we want the maximum input (r=255) to result in the maximum output (s=255).

$s = c \times \log_{10}(1 + r)$

$255 = c \times \log_{10}(1 + 255)$

$255 = c \times \log_{10}(256)$

$\log_{10}(256) = 2.40824$

$c = 255/ 2.40824 = 105.886$

Input Intensity (r)	Calculation: 105.886·log10(1+r)	Output Intensity (s)
0	105.886 x log10(1)	**0.00**
1	105.886 x log10(2)	**31.88**
10	105.886 x log10(11)	**110.27**
128	105.886 x log10(129)	**223.48**
255	105.886 x log10(256)	**254.78**

Note: Lower intensity values are expanding more as compared to higher intensity values.

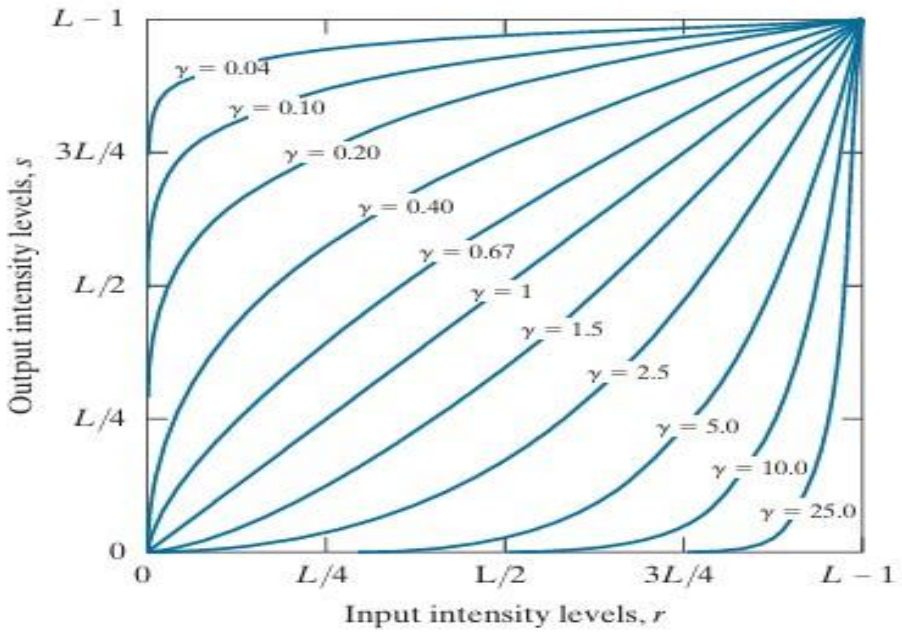
Pixel processing: Intensity Transformation: Power Law Transformations (Gamma Correction)

Motivation : Visual quality of image may be hampered by illumination condition or wrong setting of camera sensor.

To rectify the same, one can utilized power law transformation or Gamma Corrections. Basic idea is to raise the pixel value with certain power to improve the overall brightness and contrast of the image.

$$S = c\, r^{\gamma},$$

Where - r = Input Intensity Level, S = Output Intensity Level, C = constant., y = gamma value



y<1 = Overall image gets brighter.

y>1 = Overall image gets darker.

y=1 = Identity or no change in the image.

Pixel processing: Intensity Transformation: Power Law Transformations (Gamma Correction)

Case 1: Dark Pixel (r = 7) with gamma = 0.5

gamma < 1, we expect this pixel to become significantly **brighter**.

1.Normalize: 7 / 255 = 0.02745

2.Apply Gamma: (0.02745)^{0.5} = 0.16568

3.Scale back: 0.16568 x 255 = 42.25

Result: The pixel value jumps from **7 to 42**. This "stretches" the dark detail, making it much more visible.

Case 2: Bright Pixel (r = 240) with gamma = 1.5

gamma > 1, we expect this pixel to become **darker**.

1.Normalize: (240 / 255) = 0.94118$

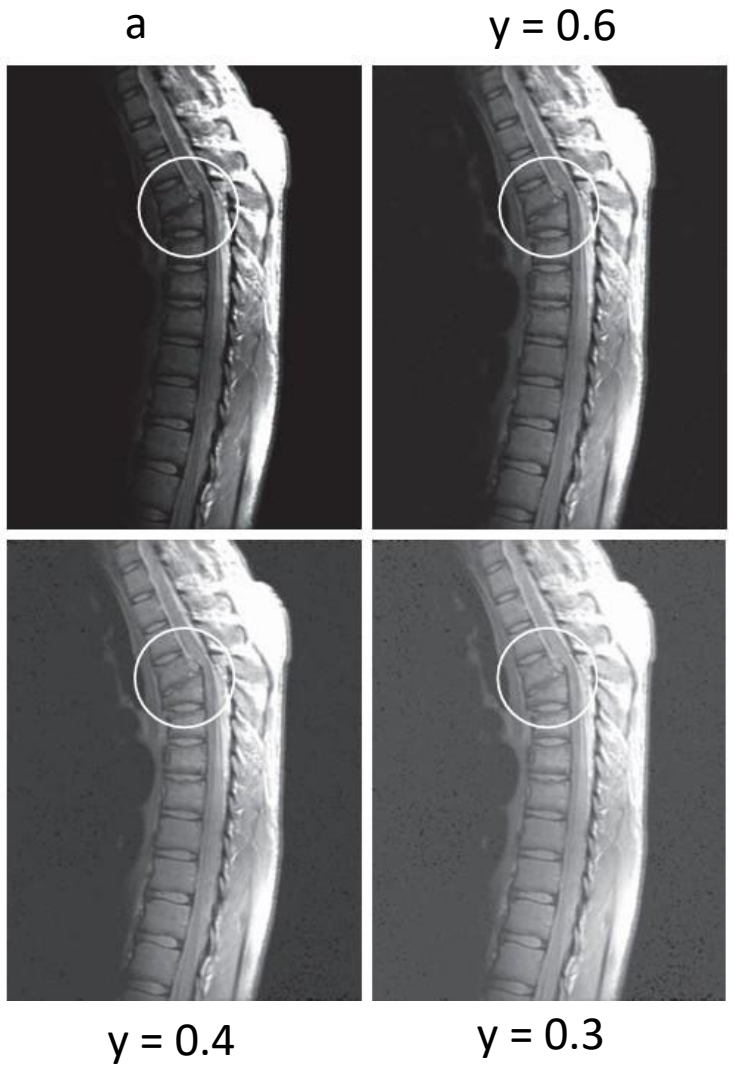
2.Apply Gamma: (0.94118)^{1.5} = 0.91354

3.Scale back: 0.91354 x 255 = 232.95

Result: The pixel value drops from **240 to 233**. Even though it is a small numerical drop, in an image, this helps reduce "blowout" and adds depth to highlight areas.

In image processing, we almost always prefer **Gamma Corrections** over Log/Exponential because of its **tunability by varying the gamma value**.

Pixel processing: Intensity Transformation: Power Law Transformations (Gamma Correction)



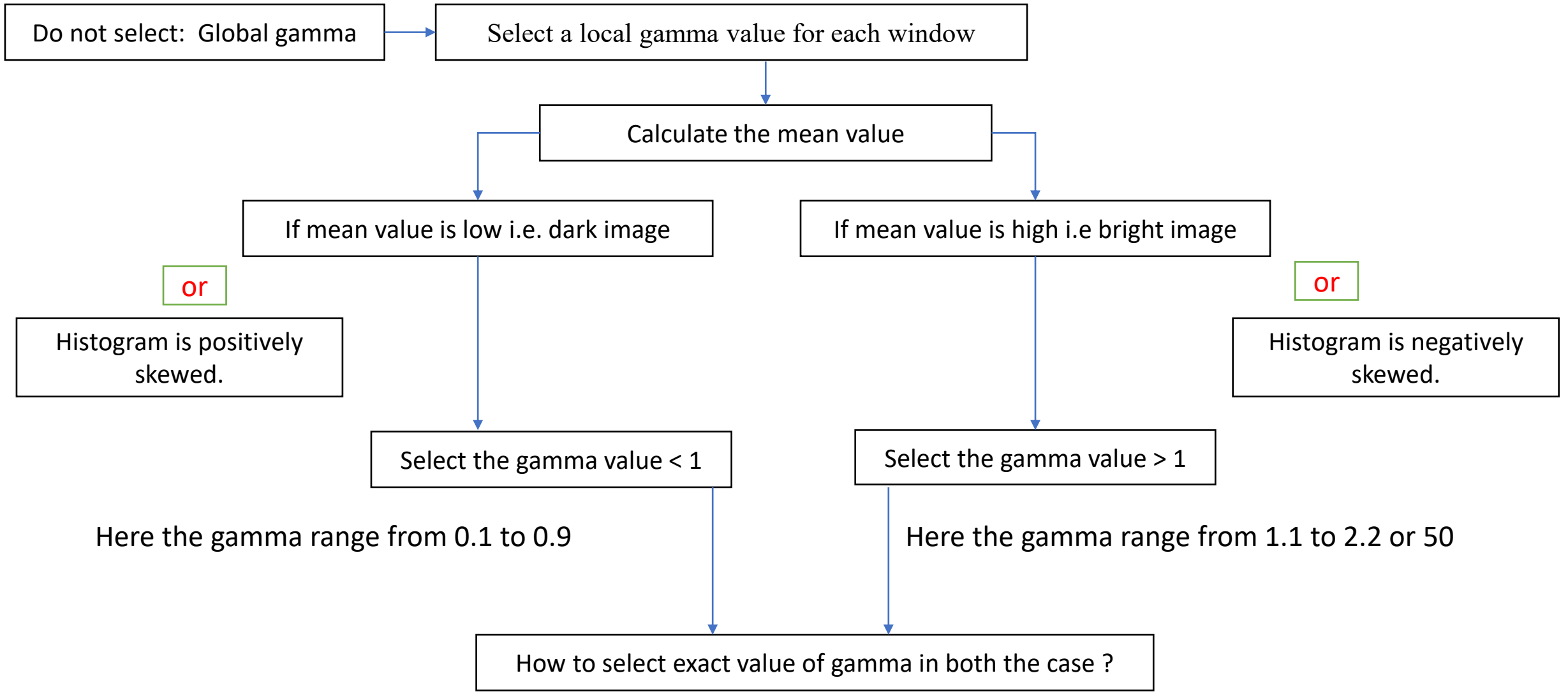
(a) MRI of fractured human spine

As gamma decreases then more details are visible in the dark area.

(a1) Aerial image of airport runway

As gamma increasing's then more details are visible in the bright area.

Local Gamma Correction or Adaptive Gamma correction.



Note: In most of the TV monitors standard gamma value is 2.2, it helps to see a lot of details.

Local Gamma Correction or Adaptive Gamma correction.

How to select exact value of gamma in both the case ?

$C = r^{y}$ Gamma correction formula

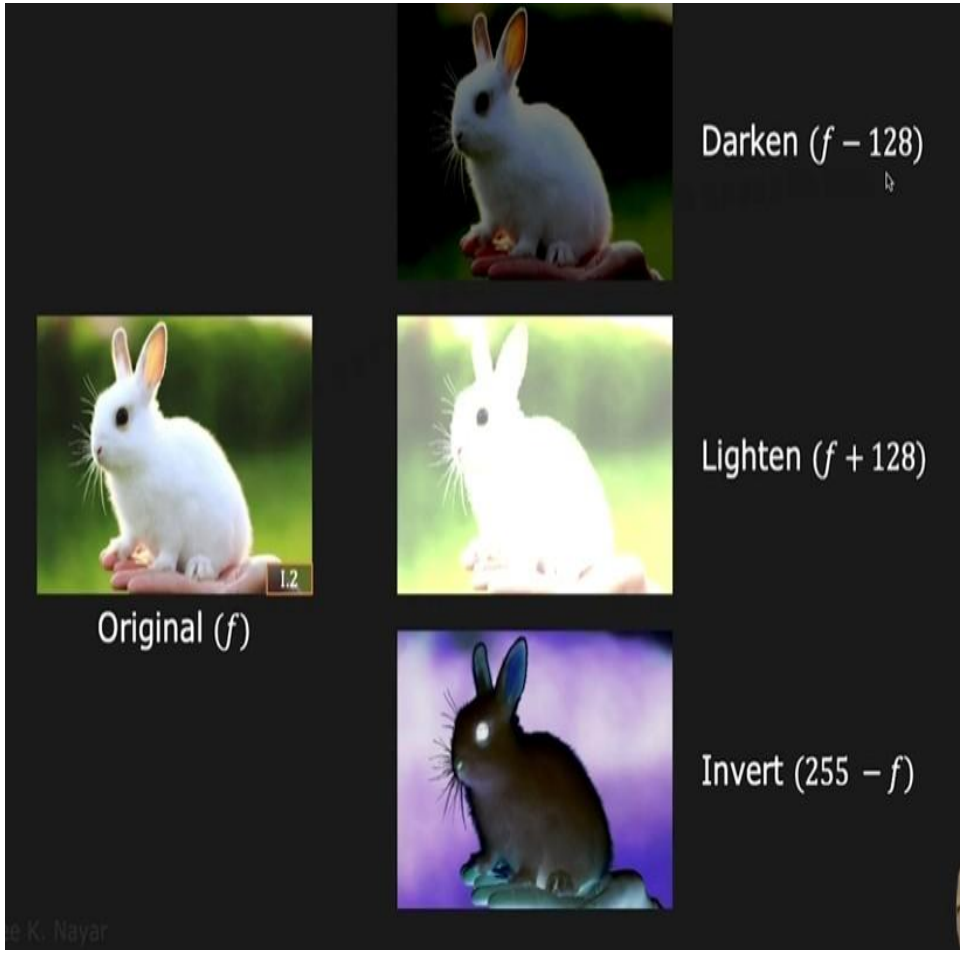
$\ln(c) = y \ln(r)$ Take log on both side.

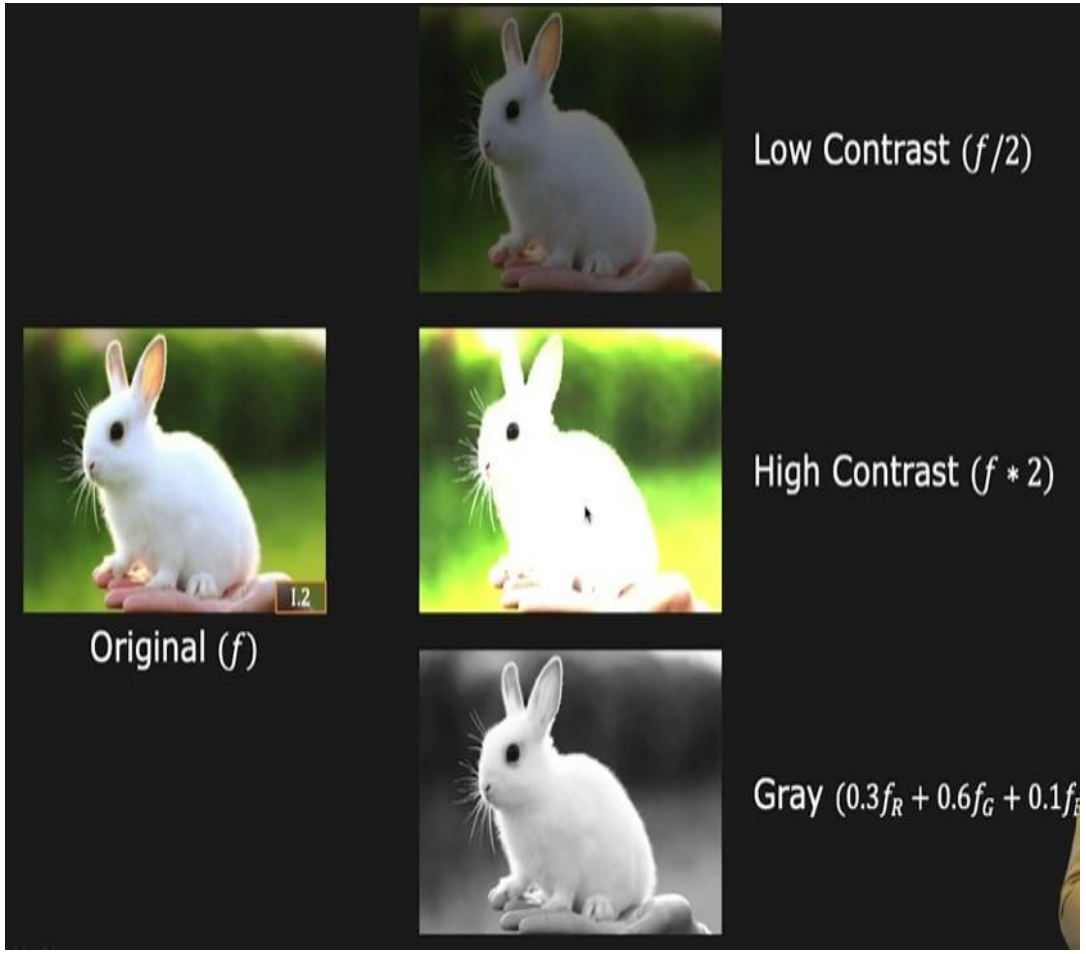
$Y = \ln(c)/\ln(r)$

$Y = \ln(0.5)/ \ln(m/255)$ Here c is mid grey level range and m is mean value Calculated in each local window.

At the end, Apply the gaussian blur on the gamma map to avoid the blocky or patchy appearance.

Pixel processing: Different transformation example





Decrease Contrast.

Increase Contrast but the maximum value can not beyond the dynamic range.

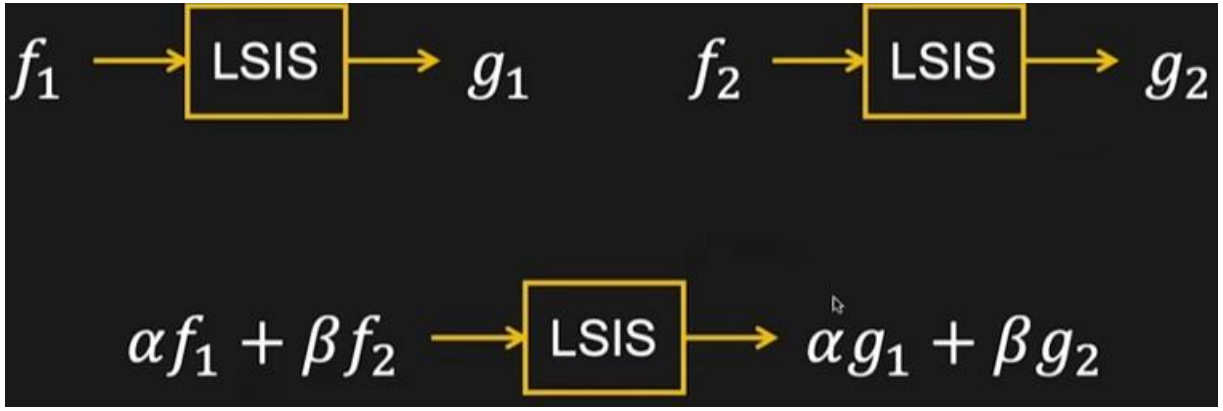
Color to Grey.

The main problem is, these techniques are **context-blind**; they process pixels in isolation without considering their neighbors, which often leads to –

1. Lost detail,
2. Amplified noise.
3. An artificial "washed-out" appearance.

Linear Shift Invariance System (LSIS) and Convolution

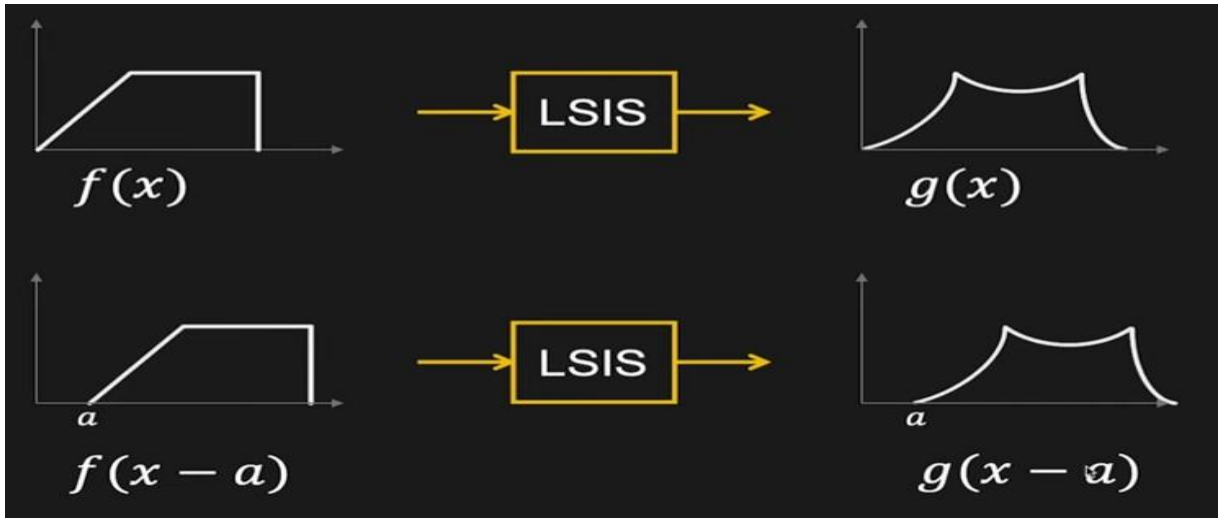
Importance of LSIS : Any system which is linear and shift invariant can be implemented as convolution.



1. Linearity:
Here, we have two different inputs. In the first (second), if input is f1(f 2) then output is g1(g2).

If you take some linear combination of f1 and f 2 then you should get same linear combination of g1 and g2 at the output.

If this condition satisfied then you can say system is linear.



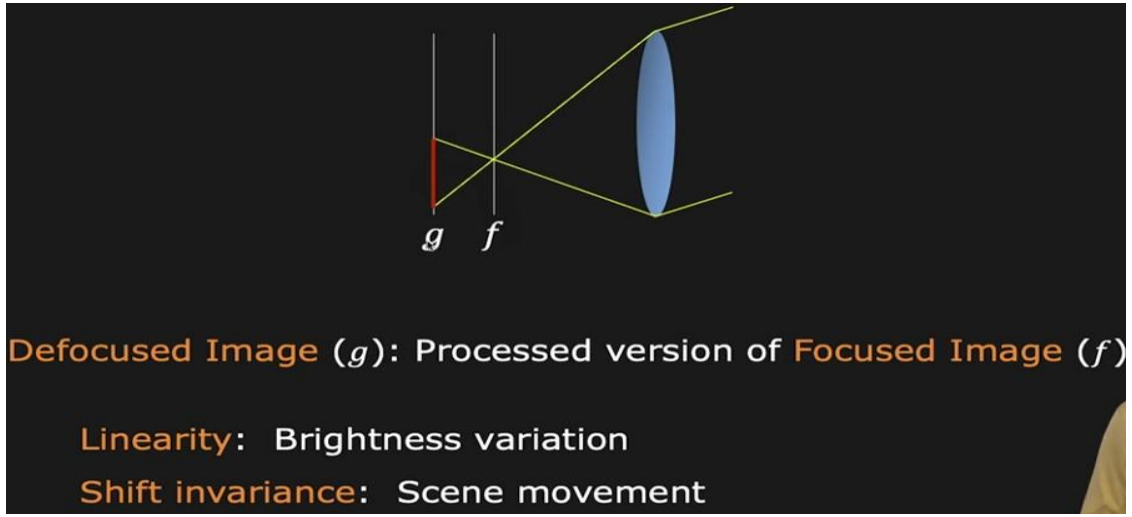
2. Shift Invariance:
Here, f(x) is input and g(x) is output.

If there is any change in input (like shift) then the same change should reflect in the output.

If this condition satisfied then you can say system is Shift Invariance.

NOTE: If any system which satisfied these two conditions is known as linear shift invariance system.

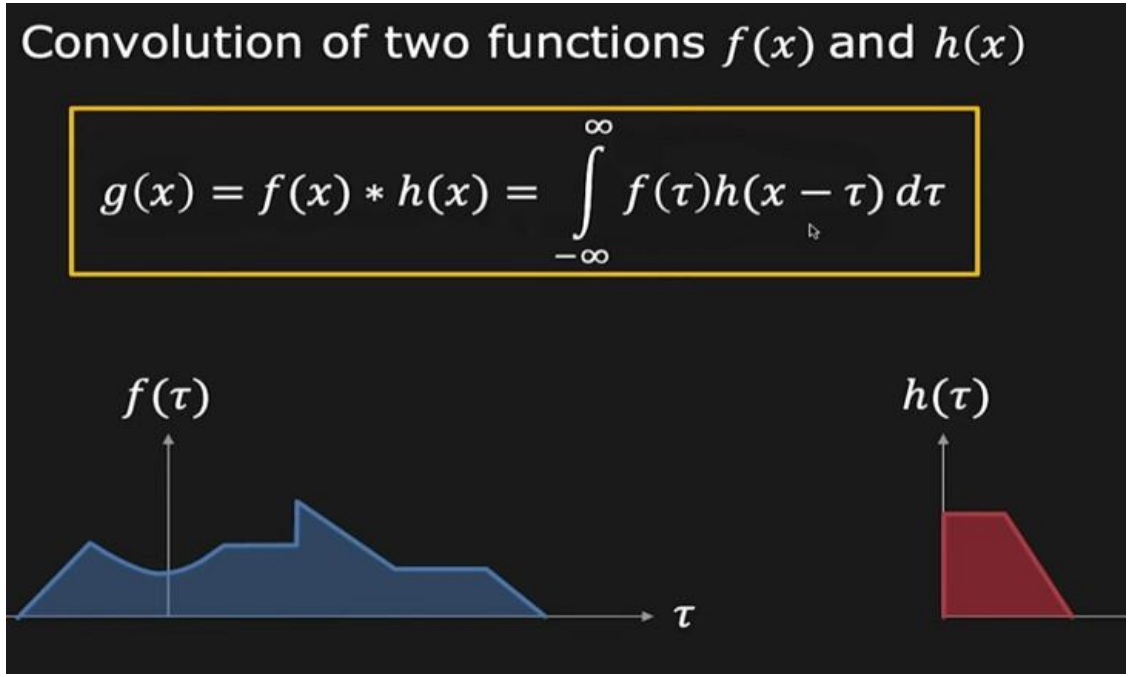
Linear Shift Invariance System (LSIS) and Convolution



How LSIS relevant to us in computer vision and imaging.

If you increase the brightness of the scene then the brightness of focused image will increases.

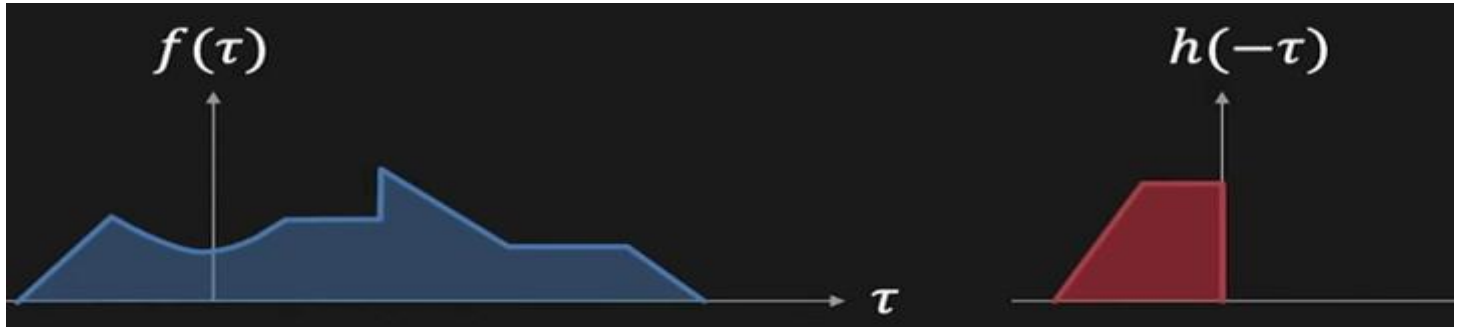
If you shift the object in the scene then their will be shift in focused and defocused image.



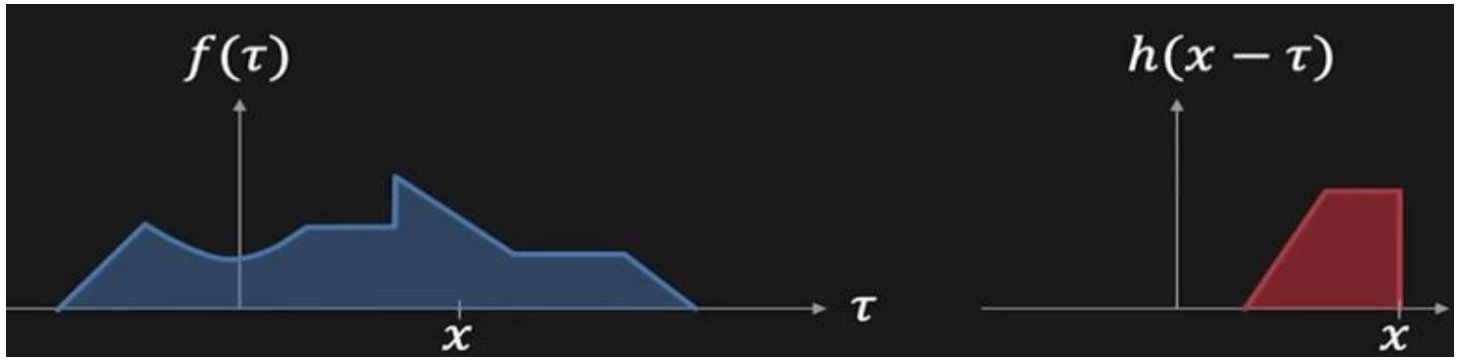
Here, you have two different function f(x) and h(x) and after convolution you will get the output g(x).

For performing convolution first flip the h(x) and then shift it.

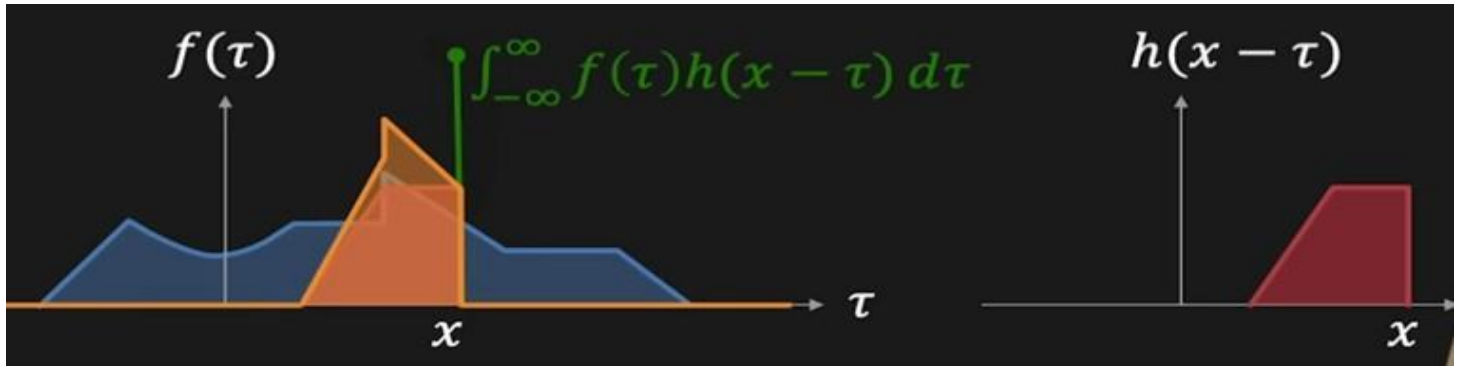
Linear Shift Invariance System (LSIS) and Convolution



Flip h(x)



shift h(x)



Overlay it on the f(x). Multiply f(x) and h(x)and perform integration, it will provide a single value to a particular point. This is know as convolution.

Linear Shift Invariance System (LSIS) and Convolution

LSIS:

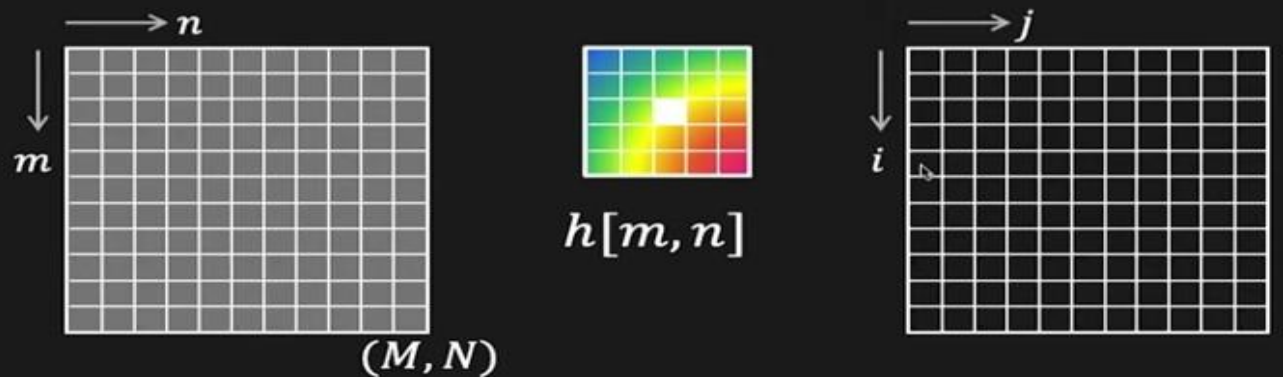
$$f(x,y) \rightarrow \boxed{h(x,y)} \rightarrow g(x,y)$$

Convolution:

$$g(x,y) = \iint_{-\infty}^{\infty} f(\tau,\mu)h(x-\tau,y-\mu)\,d\tau d\mu$$

$$f[i,j] \rightarrow \boxed{h[i,j]} \rightarrow g[i,j]$$

$$g[i,j] = \sum_{m=1}^{M}\sum_{n=1}^{N} f[m,n]h[i-m,j-n]$$

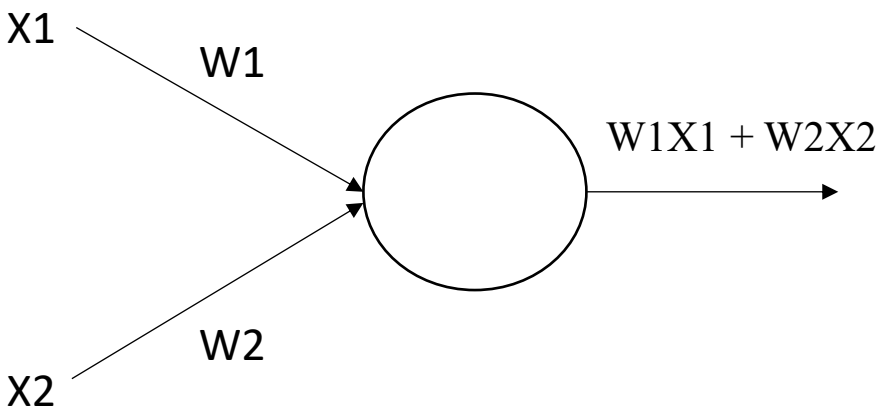


Convolution in 2D

In the Continuous Convolution flip is two times with x and y.

In the discrete convolution it is represented by m and n.

Concept of Kernal or Filter or Convolution Mask



Weights

W1	W2	W3
W4	W5	W6
W7	W8	W9

X

Inputs

X1	X2	X3
X4	X5	X6
X7	X8	X9

=

Weighted Summation

W1X1	W2X2	W3X3
W4X4	W5X5	W6X6
W7X7	W8X8	W9X9

Spatial Correlation and Convolution: Padding size (M-1)/2 or (N-1)/2

Input Image

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

Kernel

1	2	3
4	5	6
7	8	9

Padded Input Image

0	0	0	0	0	0	0
0	0	0	0	0	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	0	0	0	0	0
0	0	0	0	0	0	0

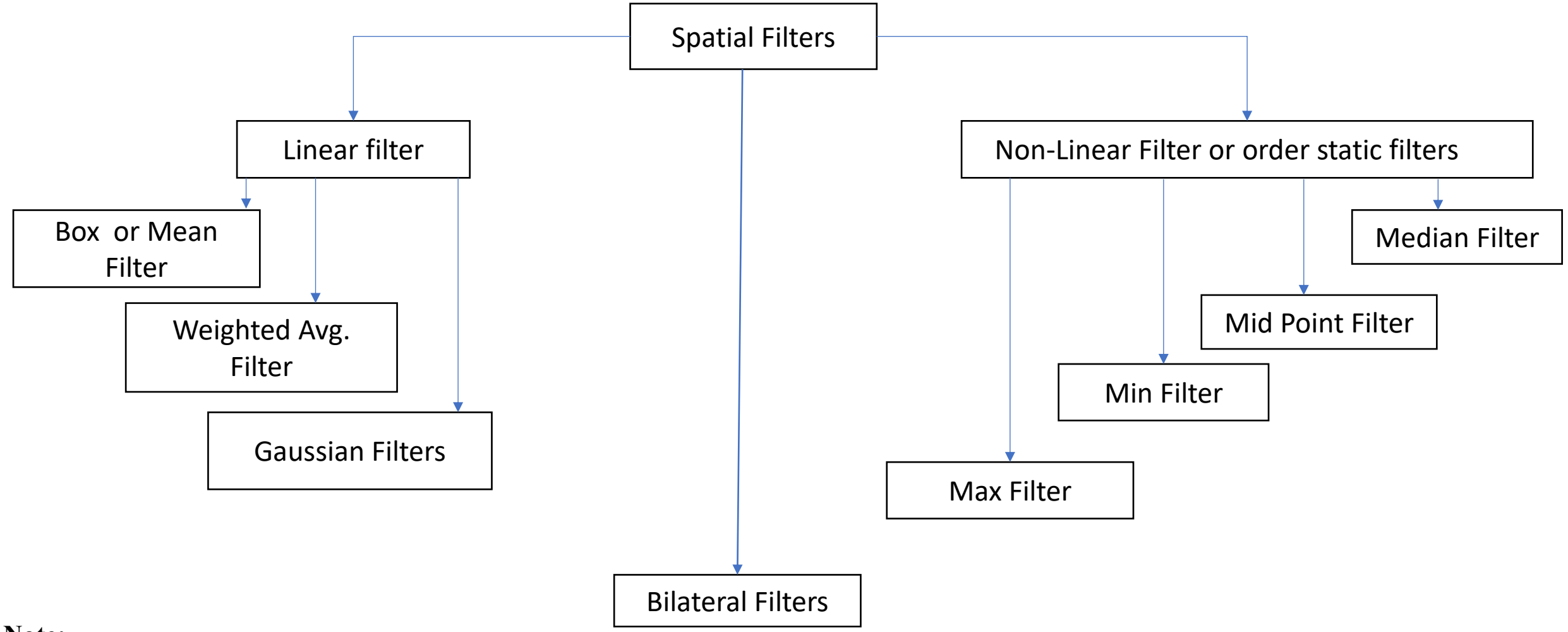
Correlation

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	0	0
0	0	3	2	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	5	6	0	0
0	0	7	8	9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Spatial Filters



Note:

Linear spatial filters means convolving an image with a linear filter or kernel.

Convolving with smoothening kernel blurs the image, with degree of blurring is determined by the size of kernel and values of its coefficients.

Image Smoothening: Box Filter

When smoothening is applied, it replaces each pixel value with the weighted average of its neighbors (This process is called as convolution.)

The simplest low pass filter kernel is box kernel whose coefficient have the same values.

Here below is a m x n box filter which is a m x n array of 1's with a normalizing constant in front, whose value is 1 divided by the sum of the values of the coefficients.

Normalizing constant

1/9 x

Kernal

1	1	1
1	1	1
1	1	1

=

Normalized Kernal

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

3 x 3 Kernel

1/16 x

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

=

0.0625	0.0625	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625
0.0625	0.0625	0.0625	0.0625

4 x 4 Kernel

Image Smoothening: Box Filter

Why Normalized kernels are required.

Consider an input image and apply normalized and not normalized kernel.

Input Image

1	2	3
4	5	6
7	8	9

Note: Not normalized means sum of the kernel elements are not equal to 1.

Normalized Kernel

Sum of kernel elements =1

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

(0.11 x 1) + (0.11 x 2) + (0.11 x 3) + (0.11 x 4)+ (0.11 x 5)+ (0.11 x 6)+ (0.11 x 7)+ (0.11 x 8)+ (0.11 x 9) = 4.95

Not Normalized Kernal (1)

Sum of kernel elements > 1

1	1	1
1	1	1
1	1	1

(1 x 1) + (1 x 2) + (1 x 3) + (1 x 4)+ (1 x 5)+ (1 x 6)+ (1 x 7)+ (1 x 8)+ (1 x 9) = 45

Not Normalized Kernal (2)

Sum of kernel elements < 1

0.01	0.01	0.01
0.01	0.01	0.01
0.01	0.01	0.01

(0.01 x 1) + (0.01 x 2) + (0.01 x 3) + (0.01 x 4)+ (0.01 x 5)+ (0.01 x 6)+ (0.01 x 7)+ (0.01 x 8)+ (0.01 x 9) = 0.45

Obs: if the sum of kernel elements are greater than 1, then new pixel value is high and overall image become brighter.
If the sum of kernel elements are less than 1, then new pixel value is low and overall image become darker.

Working Example: Padding size (M-1)/2 or (N-1)/2

Input Image

1	2	5	3	4
5	6	7	8	9
2	3	4	5	6
3	6	8	4	2
1	5	6	8	7

Kernel

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

Padded Input Image

0	0	0	0	0	0	0
0	1	2	5	3	4	0
0	5	6	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

Apply Kernel on padded input image.

Working Example: Padding size (M-1)/2 or (N-1)/2

Kernel

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

Padded Input Image

0	**0**	**0**	0	0	0	0
0	**1**	**2**	5	3	4	0
0	**5**	**6**	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

Calculation for pixel 1 -

0.11 x 0 + 0.11 x 0 + 0.11 x 0 + 0.11 x 0 + 1 x 0.11 + 2 x 0.11 + 0 x 0.11 + 5 x 0.11 + 6 x 0.11 = 1.54

1.54				

Note : Size of original image should not change. It should be 5 x 5.

Working Example: Padding size (M-1)/2 or (N-1)/2

0	**0**	**0**	0	0	0	0
0	**1**	**2**	5	3	4	0
0	**5**	**6**	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

0	**0**	**0**	0	0	0	0
0	**1**	**2**	5	3	4	0
0	**5**	**6**	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

0	**0**	**0**	0	0	0	0
0	**1**	**2**	5	3	4	0
0	**5**	**6**	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

0	**0**	**0**	0	0	0	0
0	**1**	**2**	5	3	4	0
0	**5**	**6**	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

Note : center is moving one pixel right.

Working Example: Padding size (M-1)/2 or (N-1)/2

Input Image

1	2	5	3	4
5	6	7	8	9
2	3	4	5	6
3	6	8	4	2
1	5	6	8	7

Kernel

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

Padded Input Image

0	0	0	0	0	0	0
0	1	2	5	3	4	0
0	5	6	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

Output Image

1.54	2.86	3.41	3.96	2.64

Limitation of Box Car Filter or avg. filter

Output is Blurred.

Solution

Circularly Symmetric or isotropic kernel.

Weighted average filter

Weighted average is an isotropic kind of filter which give more importance to the central pixels than four neighbors and than diagonal neighbors of the central pixel.

Input Image

1	2	5	3	4
5	**6**	**7**	**8**	9
2	**3**	**4**	**5**	6
3	**6**	**8**	**4**	2
1	5	6	8	7

Weighted Average Filter

1/16 x

1	2	1
2	4	2
1	2	1

or

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

Padded Input Image

0	0	0	0	0	0	0
0	1	2	5	3	4	0
0	5	6	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

Weighted average filter

Weighted Average Filter

1/16 x

1	2	1
2	4	2
1	2	1

=

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

Padded Input Image

0	0	0	0	0	0	0
0	1	2	5	3	4	0
0	5	6	7	8	9	0
0	2	3	4	5	6	0
0	3	6	8	4	2	0
0	1	5	6	8	7	0
0	0	0	0	0	0	0

Output Image

1.5	2.75	3.625	3.87	3

Comparison between box and weighted average filter.

Input Image

1	2	5	3	4
5	6	7	8	9
2	3	4	5	6
3	6	8	4	2
1	5	6	8	7

Box Car Kernel

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

Output Image

1.54	2.86	3.41	3.96	2.64

Input Image

1	2	5	3	4
5	6	7	8	9
2	3	4	5	6
3	6	8	4	2
1	5	6	8	7

Weighted average Kernel

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

Output Image

1.5	2.75	3.625	3.87	3

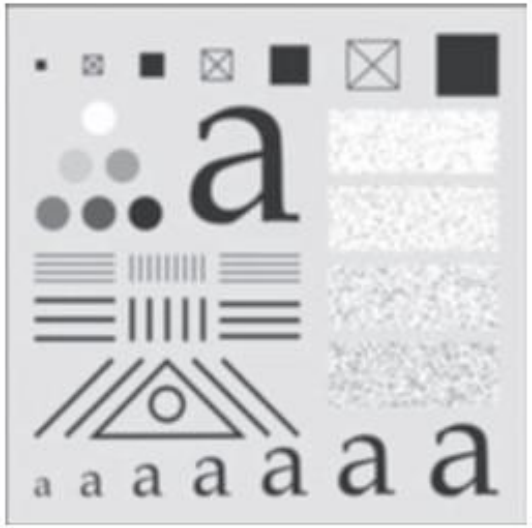
Observation : Output of weighted average is near to the original image i.e. less blurring as compared to box kernel or weighted avg filter.

Image Smoothening: Box filter

Original Image

Kernel 3 x 3

Kernel 11 x 11



Kernel 21 x 21

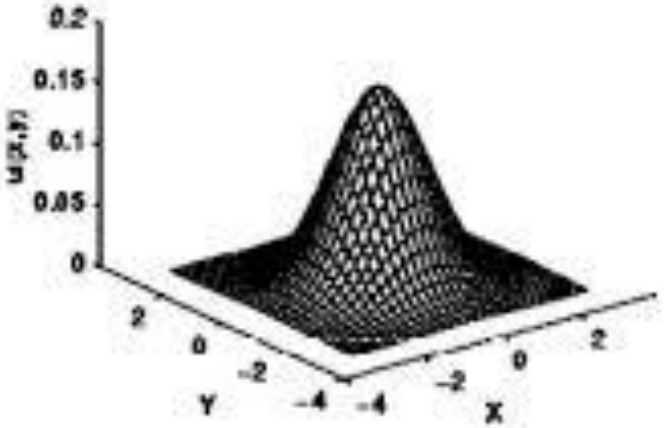
Obs: As the size of kernel is increased then smoothening increases (i.e boundaries becomes more blur.)

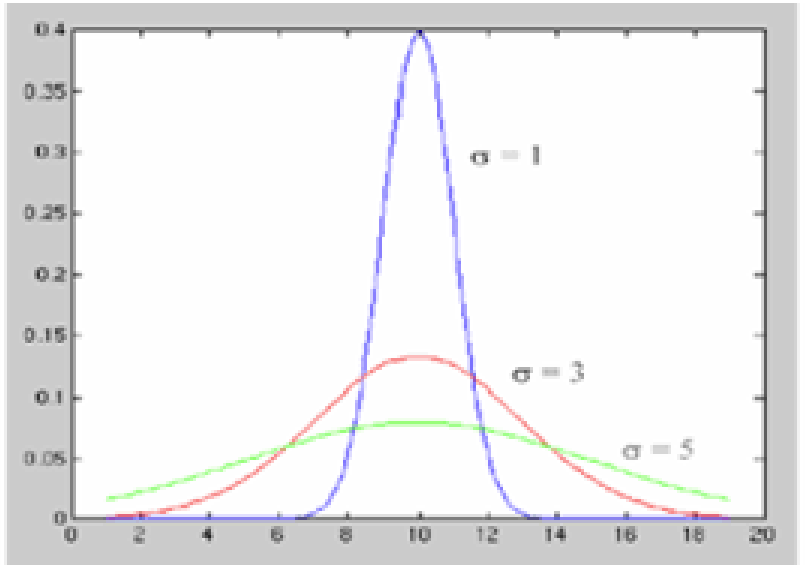
Image Smoothening: Gaussian Filter

Limitation of Box Car Filter or avg. filter
Output is Blurred.
Solution
Circularly Symmetric or isotropic kernel.

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

A graphical representation of the 2D Gaussian distribution with mean(0,0) and σ = 1 is shown to the right.





Gaussian or Normal function with different value of sigma.

Gaussian kernel or filter is circularly symmetric or isotropic, it means their response are independent of orientation of image features.

Image Smoothening: Gaussian Filter

Calculate the Gaussian Kernel for different value of sigma.

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}$$

Consider r = (s^2 + t^2)^1/2

Pixel positions for reference

(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	p(x,y)	(x,y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

-1,-1	-1,0	-1,1
0,-1	0,0	0,1
1,-1	1,0	1,1

Sigma =1

0.3678	0.6065	0.3678
0.6065	1	0.6065
0.3678	0.6065	0.3678

Sigma =2

0.7788	0.8825	0.7788
0.8825	1	0.8825
0.7788	0.8825	0.7788

Sigma =3

0.8949	0.9464	0.8949
0.9464	1	0.9464
0.8949	0.9464	0.8949

Image Smoothening: Gaussian Filter

After Normalization

Sigma =1

0.0780	0.1286	0.0780
0.1286	0.2121	0.1286
0.0780	0.1286	0.0780

As moving from central pixel to four neighbour to diagonal neighbour pixel values are decreasing.

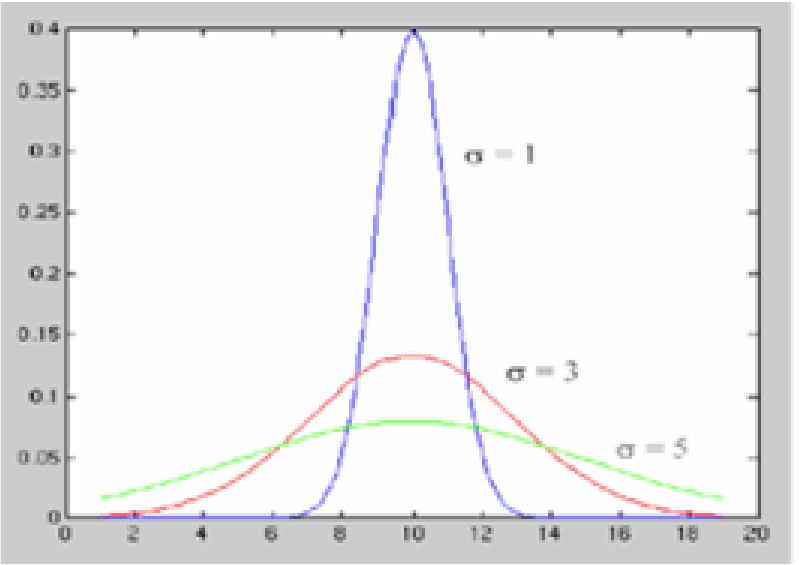
Sigma =2

0.1018	0.1154	0.1018
0.1154	0.1308	0.1154
0.1018	0.1154	0.1018

In case of sigma =2 and 3, there is not significant difference between central pixel and other neighbours.

Sigma =3

0.1069	0.1131	0.1069
0.1131	0.1195	0.1131
0.1069	0.1131	0.1069



Neighbourhood size increases as sigma is increasing from 1 to 3.

Gaussian filter working example

Calculate the value for the central pixel

Input Image

1	2	5	3	4
5	**6**	**7**	**8**	9
2	**3**	**4**	**5**	6
3	**6**	**8**	**4**	2
1	5	6	8	7

Sigma =1

0.0780	0.1286	0.0780
0.1286	0.2121	0.1286
0.0780	0.1286	0.0780

Sigma =2

0.1018	0.1154	0.1018
0.1154	0.1308	0.1154
0.1018	0.1154	0.1018

Sigma =3

0.1069	0.1131	0.1069
0.1131	0.1195	0.1131
0.1069	0.1131	0.1069

Gaussian filter working example

Input Image

1	2	5	3	4
5	**6**	**7**	**8**	9
2	**3**	**4**	**5**	6
3	**6**	**8**	**4**	2
1	5	6	8	7

Sigma =1

0.0780	0.1286	0.0780
0.1286	0.2121	0.1286
0.0780	0.1286	0.0780

Sigma =2

0.1018	0.1154	0.1018
0.1154	0.1308	0.1154
0.1018	0.1154	0.1018

Sigma =3

0.1069	0.1131	0.1069
0.1131	0.1195	0.1131
0.1069	0.1131	0.1069

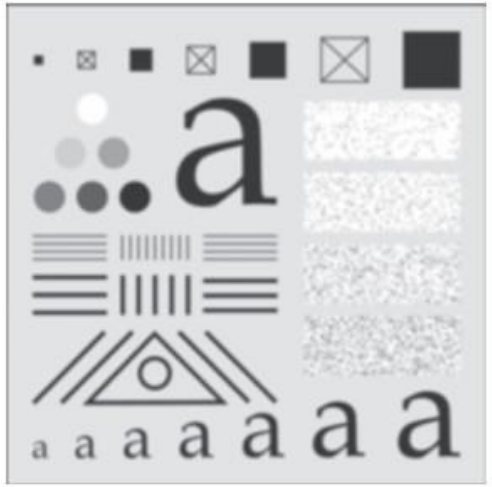
		5.1554		

		5.6209		

		5.6449		

Observation: Output of sigma =1,2,3 is near to the original value respectively

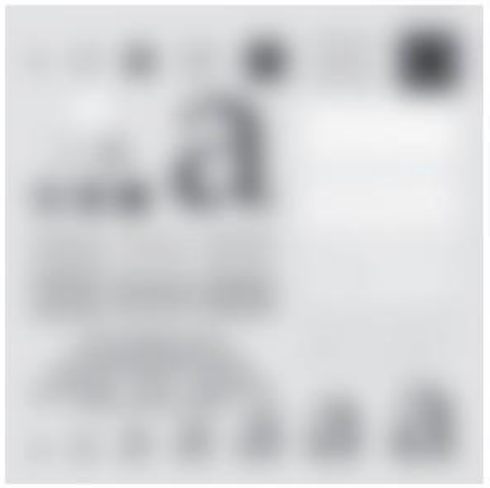
Original image



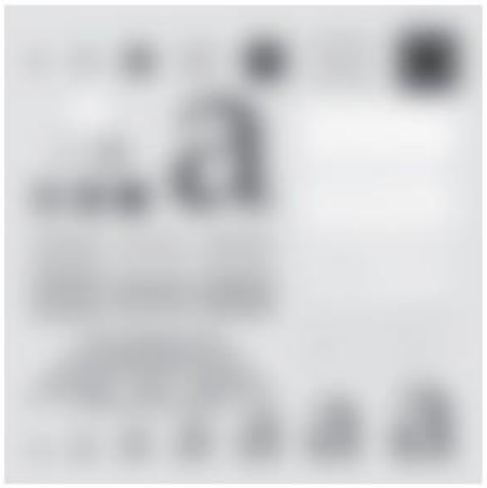
sigma = 3.5

sigma = 7

Zero Padding



Mirror Padding



replicate padding

k =1 in all three cases.
As sigma will increase from 3.5 to 7, it will incorporate more neighbourhood and image will become blur.

Sigma = 31 and k =1

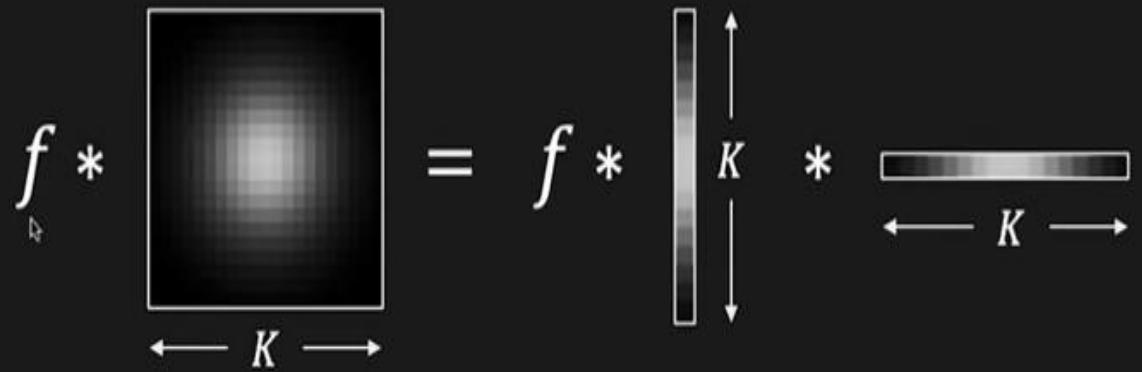
Observation: in case of zero padding borders are dark. Other two paddings are eliminating the dark border and provide more visual appealing result.

Gaussian Smoothening is separable

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f[i-m, j-n]$$

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2}{\sigma^2}\right)} \cdot \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{n^2}{\sigma^2}\right)} f[i-m, j-n]$$

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters



Convolution with 2-D gaussian mask.

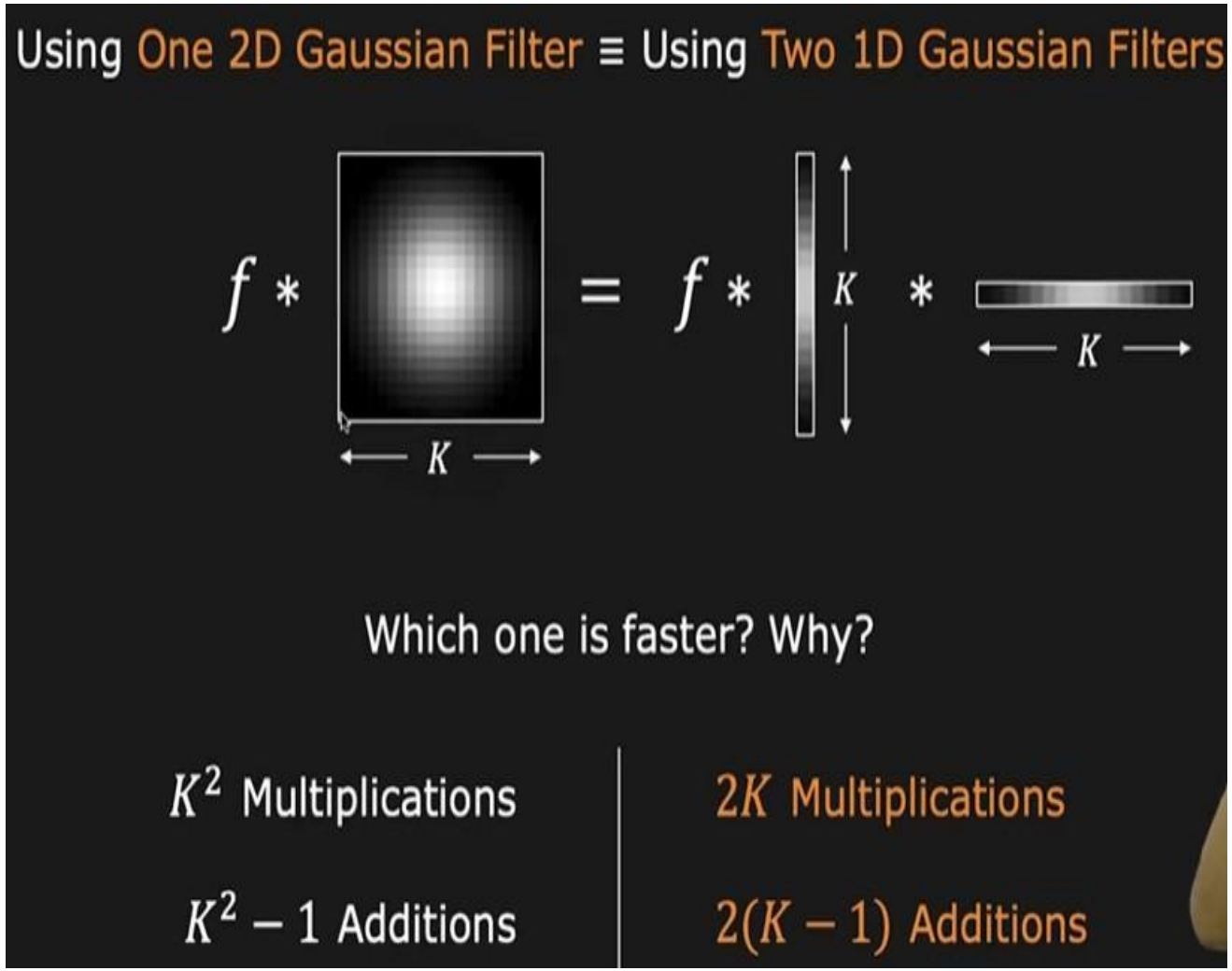
Convolution with two 1-D gaussian mask separately.

Here, lets say image size is K x K.

F convolve with Kx K image is exactly equal to the F convolve with vertical (1-D) K and the output of it convolve with the horizontal (1-D) K.

Gaussian Smoothening is separable

Cost of calculation at one pixel



Why we want to perform separate convolution in 1-D ?

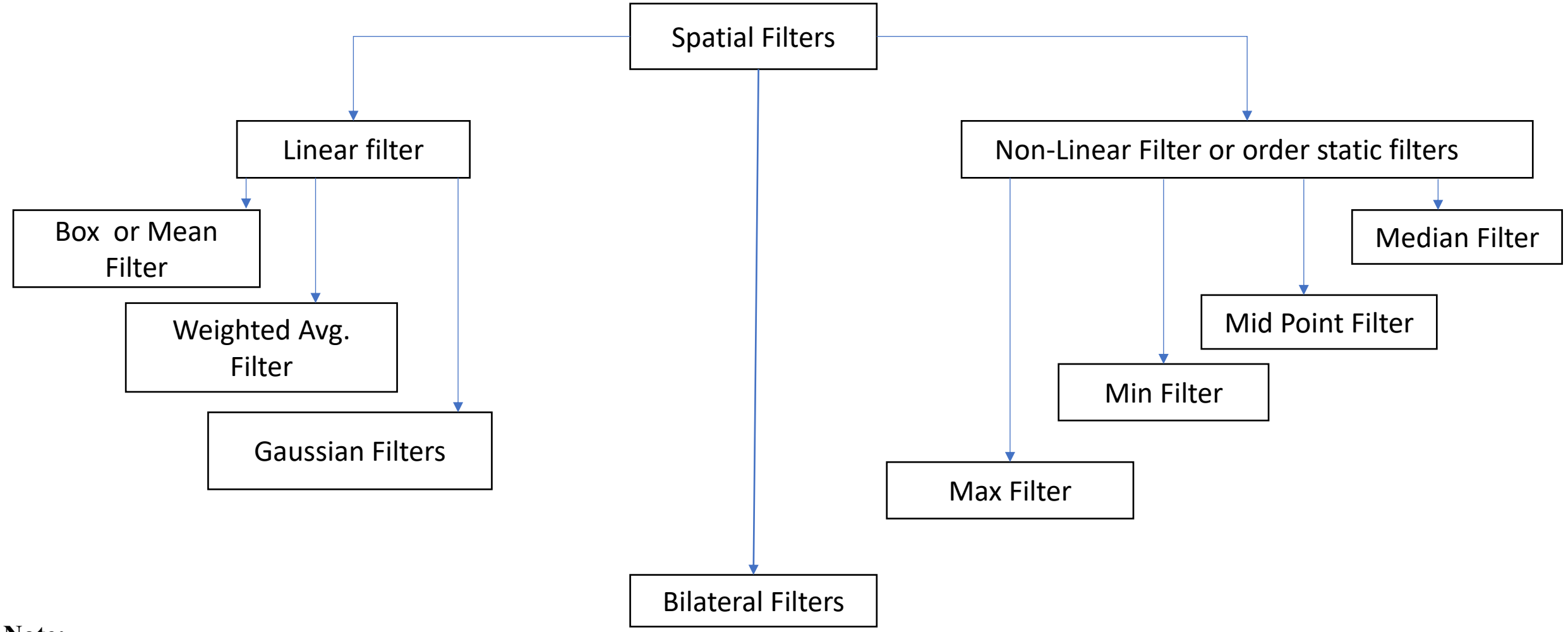
Have a look at the cost of calculation for one pixel in both the cases.

In case of 2-D convolution, number of multiplication and addition is very high as compared to two separate 1-D convolution.

Computation will increase in 2-D case, as the size of sigma or gaussian filter will increase.

NOTE: If any kernel is separable then exploit this property to decrease the computation complexity.

Spatial Filters



Note:

Linear spatial filters means convolving an image with a linear filter or kernel.

Convolving with smoothening kernel blurs the image, with degree of blurring is determined by the size of kernel and values of its coefficients.

Max Filter, Min Filter, Mid-Point and Median Filter

Max and Min Filter

Max Filter:

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

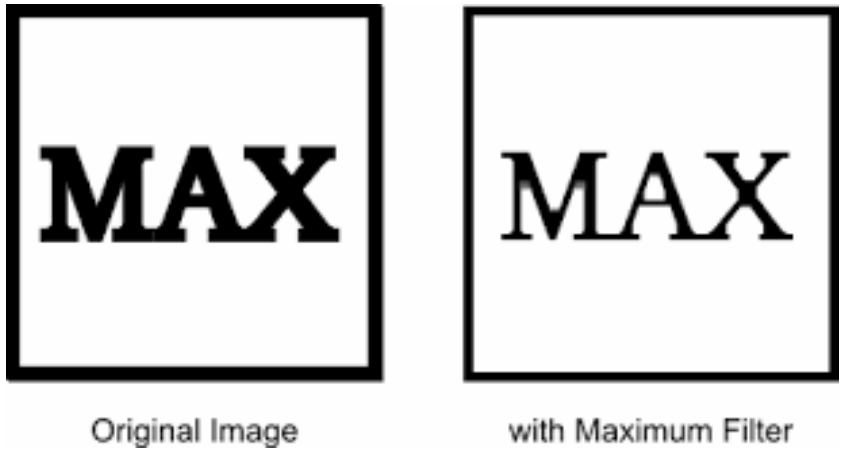
Min Filter:

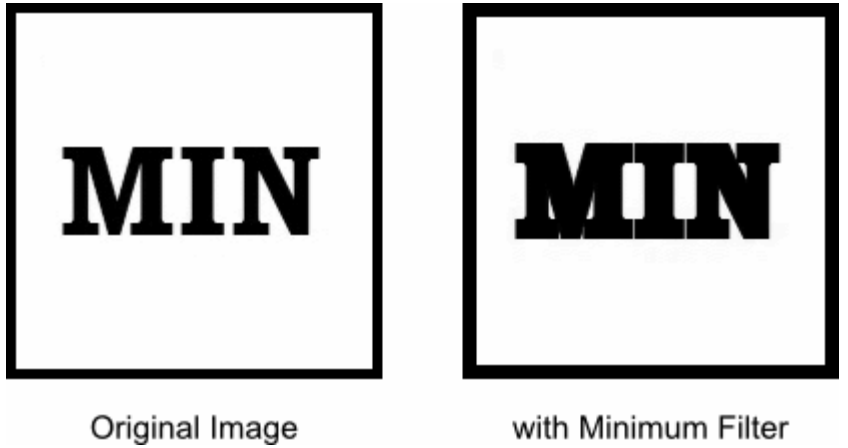
$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Max filter is good for pepper noise and Min filter is good for salt noise.

Median filter

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$





Note: Non linear filter where smoothening or filtering is independent of the size of kernel.

Max Filter, Min Filter, Mid-Point, Median filter. Assume Kernel size is 3 x 3

Input Image

3	7	17	18	13
10	5	2	20	5
9	8	13	1	7
16	8	7	20	19
14	19	3	30	10

Input Image After Padding Zeros

0	0	0	0	0	0	0
0	3	7	17	18	13	0
0	10	5	2	20	5	0
0	9	8	13	1	7	0
0	16	8	7	20	19	0
0	14	19	3	30	10	0
0	0	0	0	0	0	0

First Step

0	0	0	0	0	0	0
0	3	7	17	18	13	0
0	10	5	2	20	5	0
0	9	8	13	1	7	0
0	16	8	7	20	19	0
0	14	19	3	30	10	0
0	0	0	0	0	0	0

Second Step

0	0	0	0	0	0	0
0	3	7	17	18	13	0
0	10	5	2	20	5	0
0	9	8	13	1	7	0
0	16	8	7	20	19	0
0	14	19	3	30	10	0
0	0	0	0	0	0	0

Max Filter, Min Filter, Mid-Point, Median filter. Assume Kernel size is 3 x 3

Input Image After Padding Zeros

0	0	0	0	0	0	0
0	3	7	17	18	13	0
0	10	5	2	20	5	0
0	9	8	13	1	7	0
0	16	8	7	20	19	0
0	14	19	3	30	10	0
0	0	0	0	0	0	0

Median Filter Output

Max. Filter Output

Min. Filter Output

Mid Point Filter = ½(Min filter + Max Filter)

Max Filter, Min Filter, Mid-Point, Median filter. Assume Kernel size is 3 x 3

Input Image After Padding Zeros

0	0	0	0	0	0	0
0	3	7	17	18	13	0
0	10	5	2	20	5	0
0	9	8	13	1	7	0
0	16	8	7	20	19	0
0	14	19	3	30	10	0
0	0	0	0	0	0	0

Median Filter Output

0	3	5	5	0
5	8	8	13	5

Max. Filter Output

10	17	20	20	20
10	17	20	20	20

Min. Filter Output

0	0	0	0	0
0	2	1	1	0
0	2	1	1	0
0	3	1	1	0
0	0	0	0	0

Mid Point Filter = ½(Min filter + Max Filter)

5	8.5	10	10	10
5	8.5	10	10	10

Comparison Between Max., Min. and Median Filters output

Original Image Maximum Filter output Minimum Filter output Median Filter output

Adaptive Median filters

All the previously defined filters are applied to an image without regard for how much image characteristics vary from one point to another point.

z_{min} = minimum intensity value in S_{xy}

z_{max} = maximum intensity value in S_{xy}

z_{med} = median of intensity values in S_{xy}

z_{xy} = intensity at coordinates (x, y)

S_{max} = maximum allowed size of S_{xy}

The adaptive median-filtering algorithm uses two processing levels, denoted level A and level B, at each point (x, y):

Level A: If $z_{min} < z_{med} < z_{max}$, go to Level B

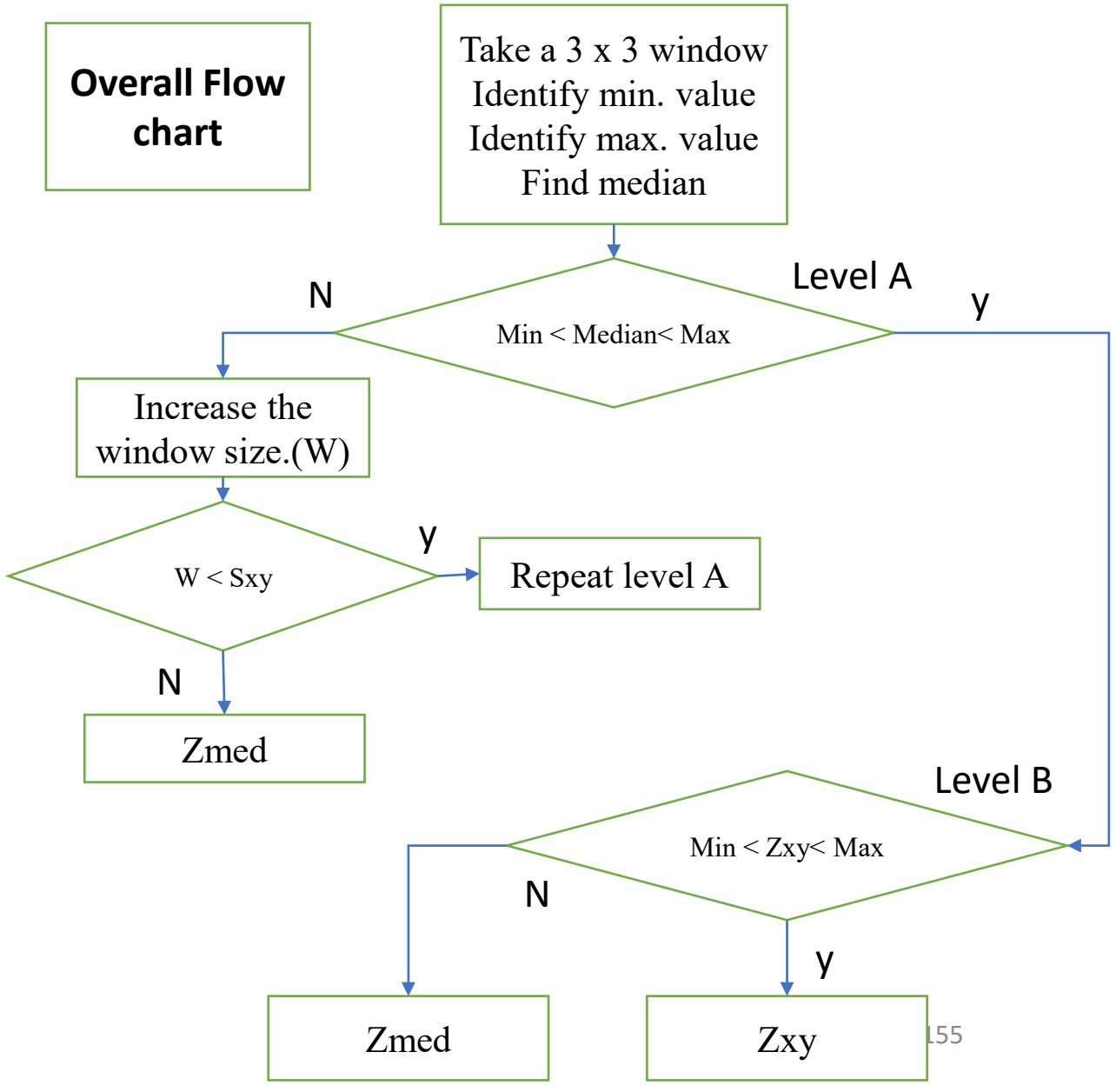
Else, increase the size of S_{xy}

If $S_{xy} \leq S_{max}$, repeat level A

Else, output z_{med}.

Level B: If $z_{min} < z_{xy} < z_{max}$, output z_{xy}

Else output z_{med}.



Adaptive Median filter working example

1	2	3
4	5	6
7	8	9

Z min = 1, Z max = 9

Arrange them in ascending order = 1,2,3,4,5,6,7,8,9
Z median = 5
Z median is in between Zmin and Zmax. (Level A first condition true.)

Arrange them in ascending order = 1,2,3,4,5,6,7,8,9
Zxy = 5
Zxy is in between Zmin and Zmax. (Level B first condition true.)

Output will be Zxy i.e Zxy = 5

z_{min} = minimum intensity value in S_{xy}
z_{max} = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity at coordinates (x, y)
S_{max} = maximum allowed size of S_{xy}

The adaptive median-filtering algorithm uses two processing levels, denoted level A and level B, at each point (x, y):

Level A: If $z_{min} < z_{med} < z_{max}$, go to Level B
Else, increase the size of S_{xy}
If $S_{xy} \leq S_{max}$, repeat level A
Else, output z_{med}.

Level B: If $z_{min} < z_{xy} < z_{max}$, output z_{xy}
Else output z_{med}.

Adaptive Median filter working example

5	2	3
4	1	6
7	8	9

Z min = 1, Z max = 9

Arrange them in ascending order = 1,2,3,4,5,6,7,8,9
Z median = 5
Z median is in between Zmin and Zmax. (Level A first condition true.)

Arrange them in ascending order = 1,2,3,4,5,6,7,8,9
Zxy = 1
Zxy is not in between Zmin and Zmax. (Level B first condition fail.)

Output will be Zmedian i.e Zmedian = 5

z_{min} = minimum intensity value in S_{xy}
z_{max} = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity at coordinates (x, y)
S_{max} = maximum allowed size of S_{xy}

The adaptive median-filtering algorithm uses two processing levels, denoted level A and level B, at each point (x, y):

Level A: If $z_{min} < z_{med} < z_{max}$, go to Level B
Else, increase the size of S_{xy}
If $S_{xy} \leq S_{max}$, repeat level A
Else, output z_{med}.

Level B: If $z_{min} < z_{xy} < z_{max}$, output z_{xy}
Else output z_{med}.

Adaptive Median filter working example

1	1	255
1	1	255
1	1	255

Z min = 1, Z max = 255

Arrange them in ascending order = 1,1,1,1,1,1,255,255,255
Z median = 1
Z median is not in between Zmin and Zmax. (Level A first condition fail.)

Check the second condition of level A

In this situation, increase the size of the window like it is 3 by 3 then take a size of 7 by 7 and repeat the process.

z_{min} = minimum intensity value in S_{xy}
z_{max} = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity at coordinates (x, y)
S_{max} = maximum allowed size of S_{xy}

The adaptive median-filtering algorithm uses two processing levels, denoted level A and level B, at each point (x, y):

Level A: If $z_{min} < z_{med} < z_{max}$, go to Level B
Else, increase the size of S_{xy}
If $S_{xy} \leq S_{max}$, repeat level A
Else, output z_{med}.

Level B: If $z_{min} < z_{xy} < z_{max}$, output z_{xy}
Else output z_{med}.

Comparison between median filter and adaptive median filter.

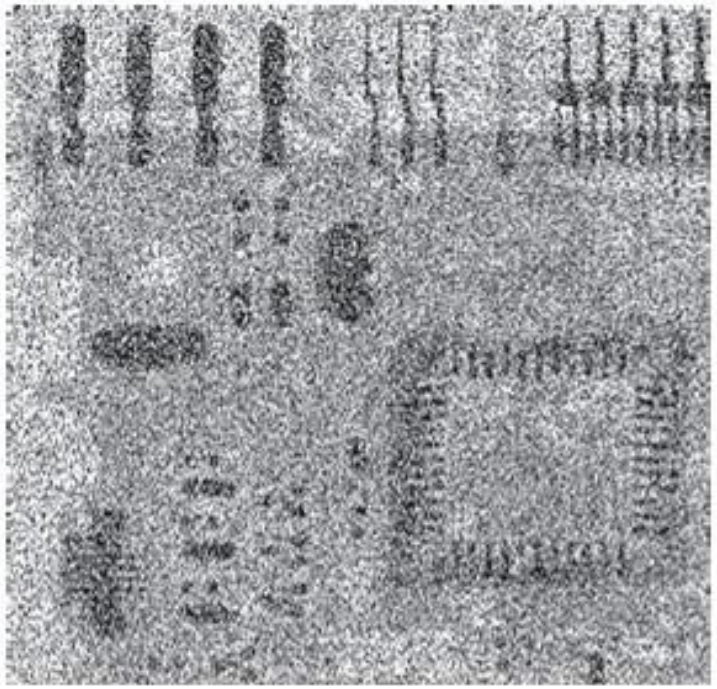
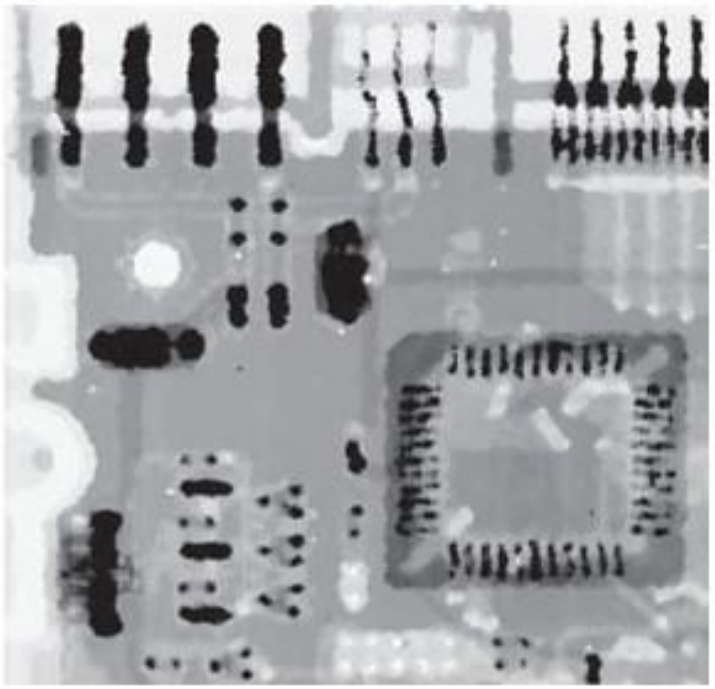
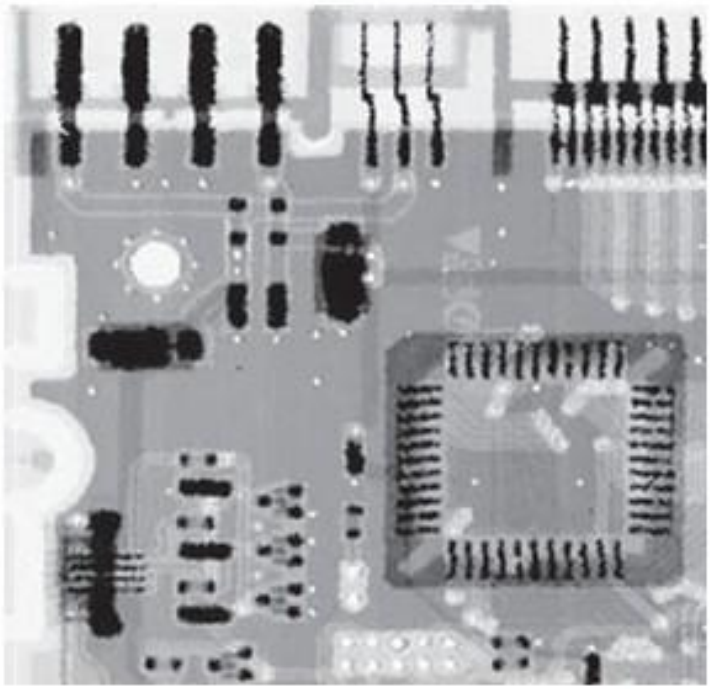


Image corrupted by salt and pepper noise

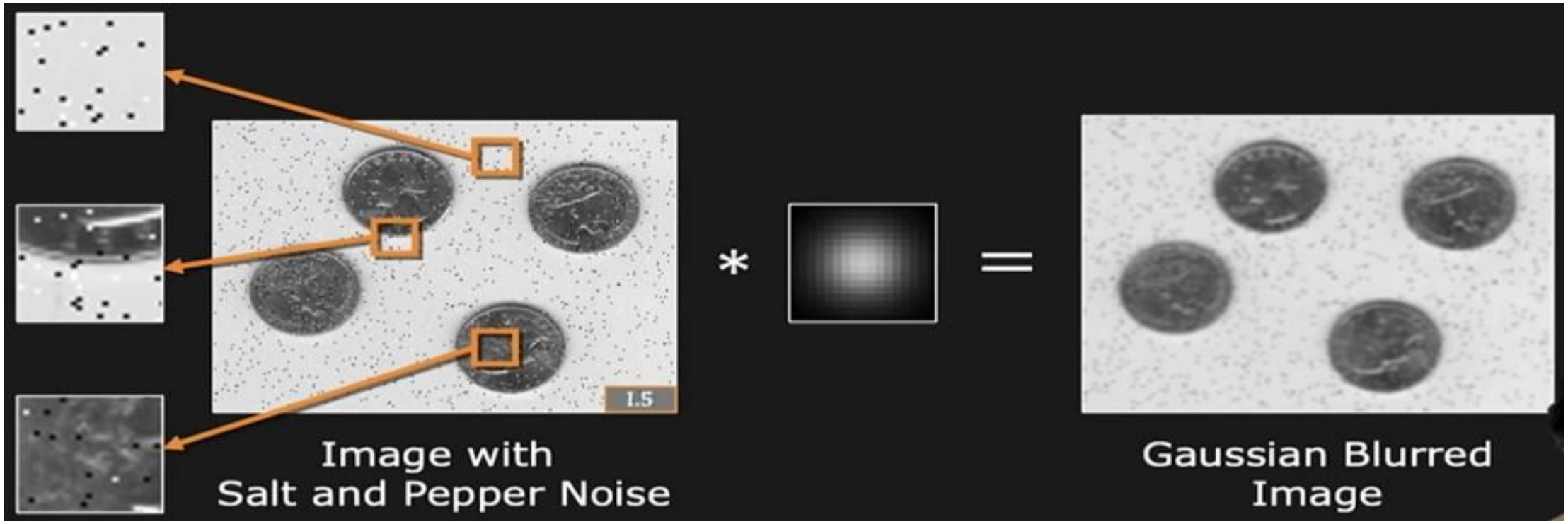


Filtering with 7 x 7 median filter



Adaptive median filter with $S_{max} = 7$

Why Non –linear image filters are required ?



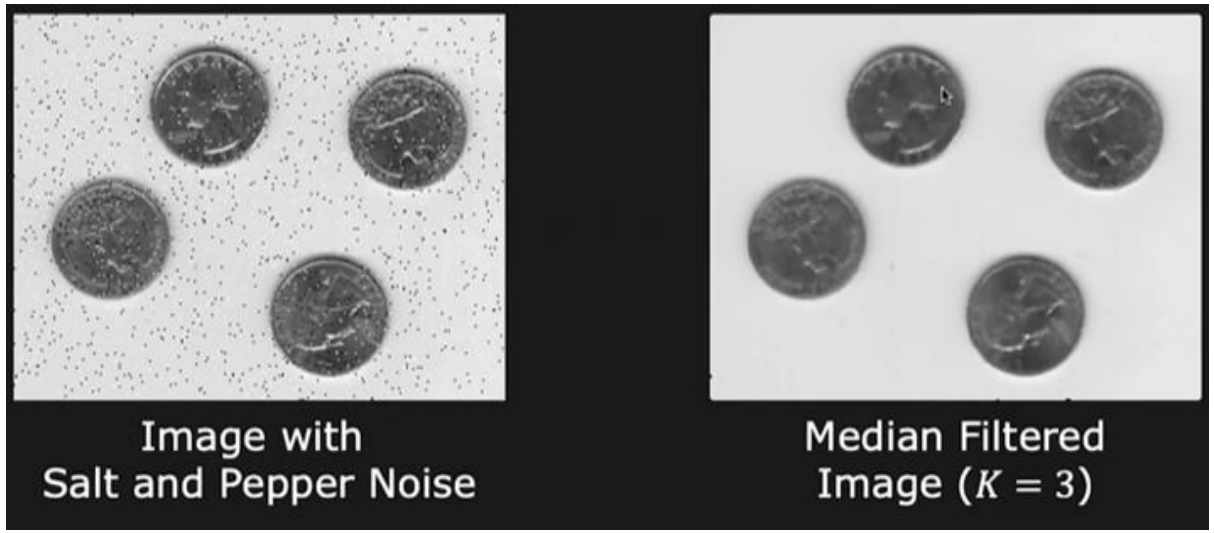
Here, an image with salt and pepper noise.

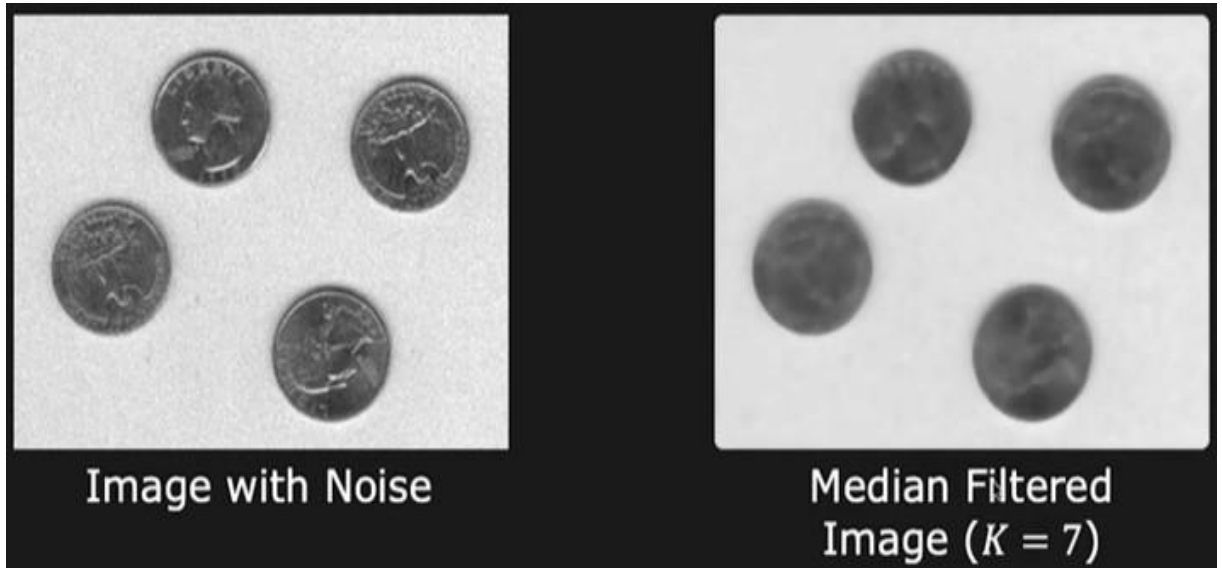
1. We want to preserve these information during noise reduction.
2. If we apply the gaussian blur then -
   - 2.1 It does not remove the outliers. (Noise.)
   - 2.2 It smooth the edges. (Blur)

Non- Linear Filter -

Convolution or the linear filter could not able to remove the noise then one can use non linear filter.

Median filter





Steps followed in Median filter –

1. Take a window.
2. Arrange all the pixels in the ascending or descending order.
3. Calculate the median. Repeat the process.

It turns out, median is a **magical filter** when salt and pepper noise is present in the image.

Median filter is not effective -

When the image contains a type of noise which impact all the pixels

or

Noise which added in all the pixels.

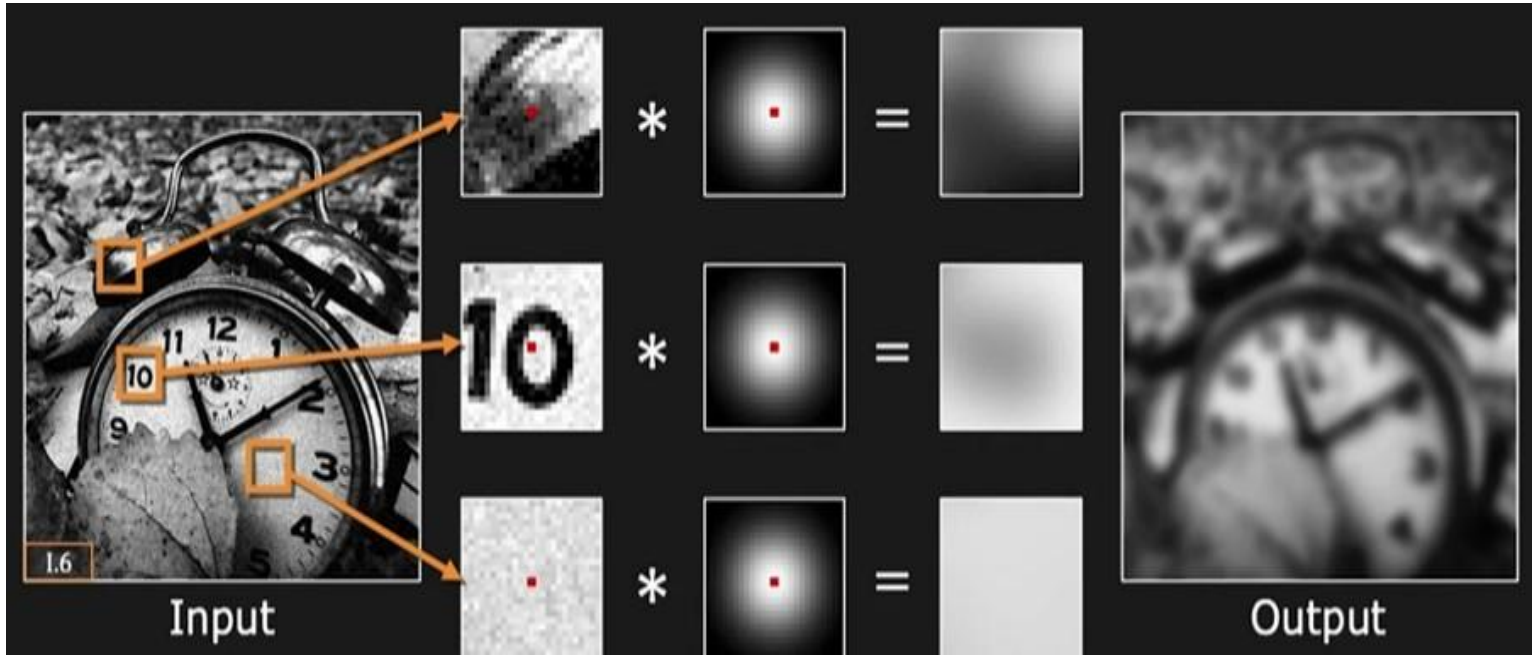
At the same time blurriness in image increased or details inside the image is lost.

As the type of noise changes, then median filters is not working perfectly.

Filter = Gaussian filter + Median filter.

Bi-lateral Filter.

What is the issue with gaussian smoothening



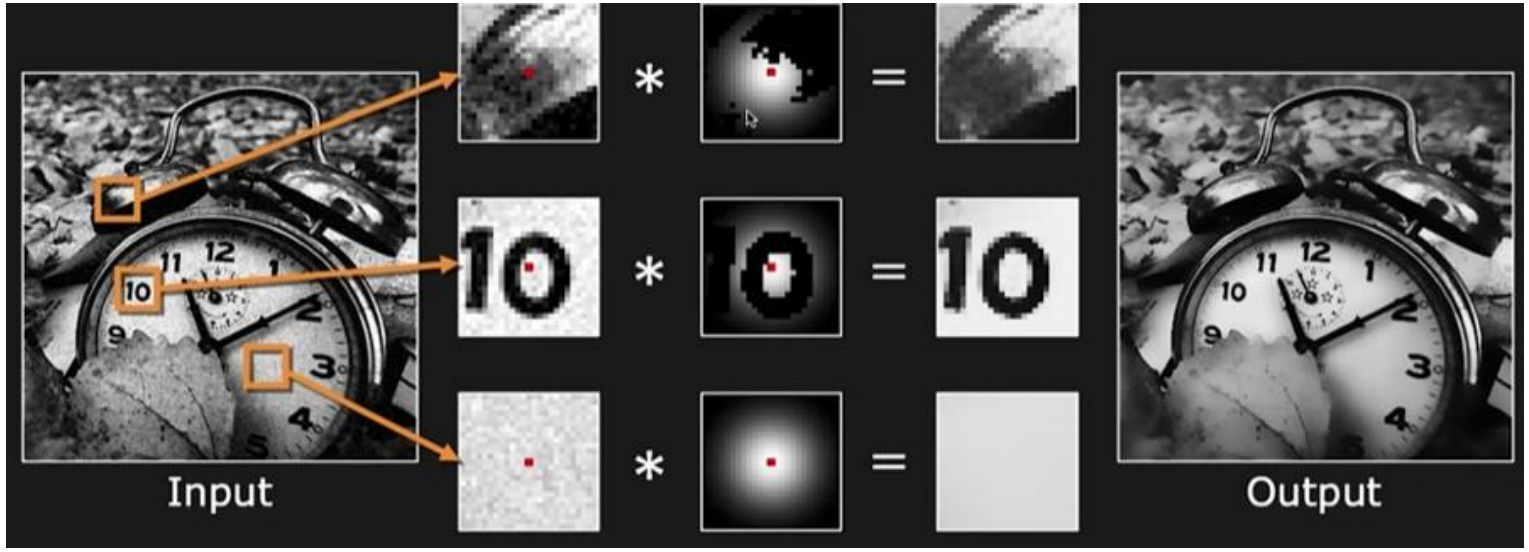
Here, we have Input image with some noise.

1. Three different ROI of image is selected with three different contextual information.

2. Same gaussian kernel apply to all these ROI.

3. In the last case, where the input image is homogenous or the flat region, gaussian filter is performing better as compared to both the ROI where some edge and text information present respectively.

Solution: Use adaptive kernel based on the context.

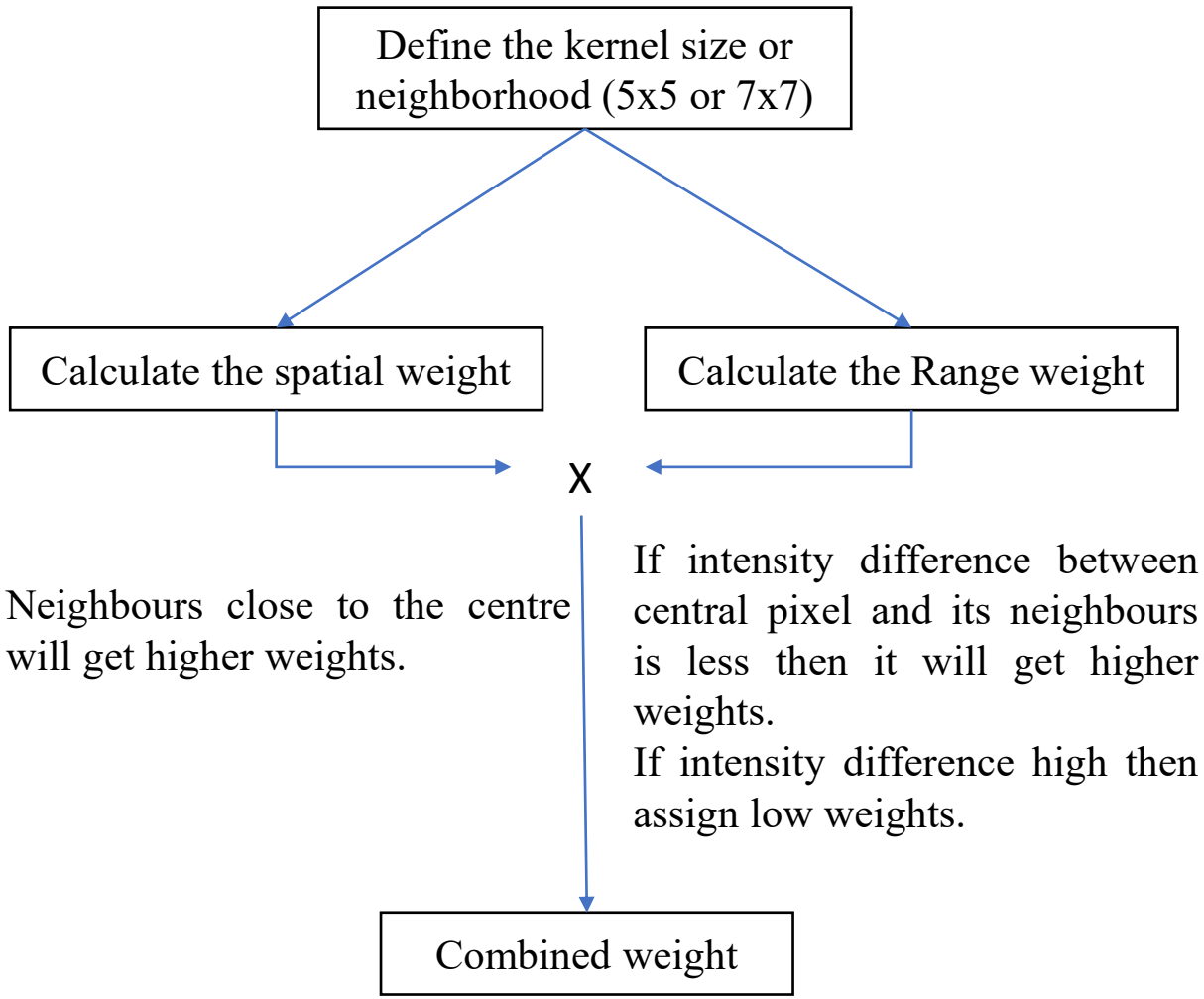


Here, gaussian kernel designed in such a way that pixels not similar in intensity to the centre pixel receive a lower weight.

It means, blur the similar pixels or use a certain part of the gaussian kernel or bias the gaussian kernel.

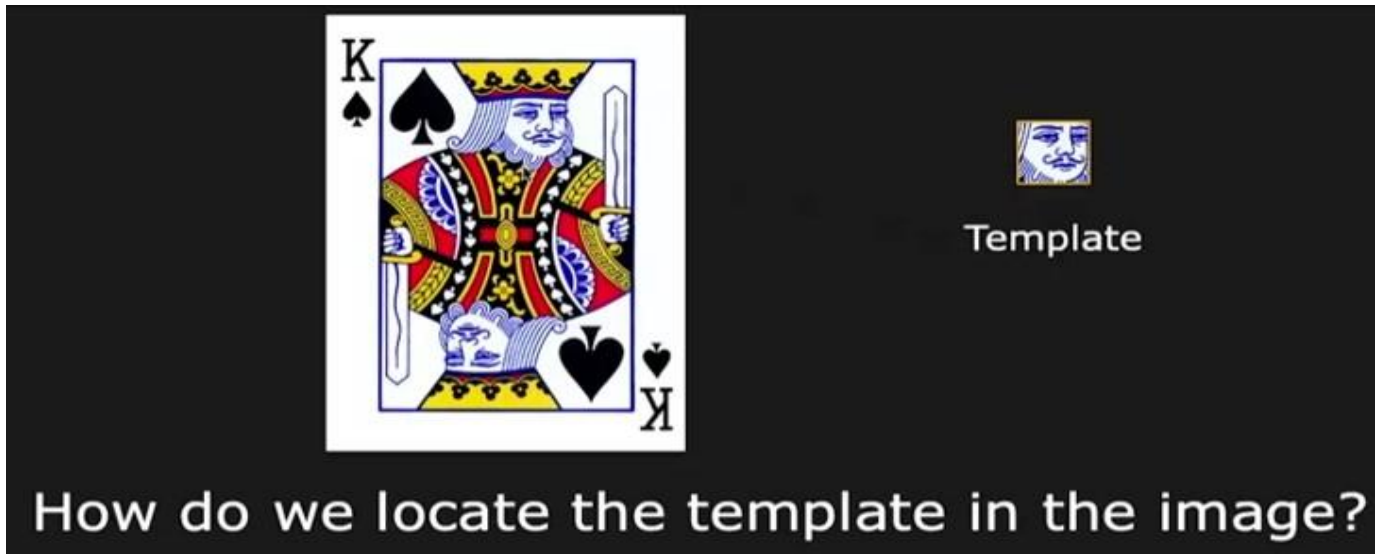
Bi-lateral Filter or Edge Preserving filter

Flow Chart



This combined weight ensures that a pixel only influences the result. if it is **both** physically close **and** visually similar.

Template Matching: Correlation



Take the templet

↓

Slide it over the image

↓

Find the difference between overlapping part of image and template.

↓

Wherever you will get the minimum difference that will be the position of template in the image.

Formulate it in mathematical way :

Minimize:

$$E[i,j] = \sum_m \sum_n (f[m,n] - t[m-i,n-j])^2$$

$$E[i,j] = \sum_m \sum_n (f^2[m,n] + t^2[m-i,n-j] - 2f[m,n]t[m-i,n-j])$$

Maximize

It is sum of square differences.
Here f(m,n) = original image and then t[m-i, n-j] is Template.

Template Matching: Convolution and Correlation

Convolution:

$$g[i,j] = \sum_m \sum_n f[m,n]t[i-m,j-n] = t * f$$

Correlation:

$$R_{tf}[i,j] = \sum_m \sum_n f[m,n]t[m-i,n-j] = t \otimes f$$

No Flipping in Correlation

Problem with cross correlation ?

$$R_{tf}[i,j] = \sum_m \sum_n f[m,n]t[m-i,n-j] = t \otimes f$$

$$R_{tf}(C) > R_{tf}(B) > R_{tf}(A)$$

We need $R_{tf}(A)$ to be the maximum!

Solution: use normalized cross correlation to overcome the problem of uneven illumination.

Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]} \sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$