MAEG 5720:
Computer Vision in Practice

Lecture 2:

Image Formation

Dr. Terry Chang

2021-2022

Semester 1

香港中文大學
The Chinese University of Hong Kong

Department of Mechanical and Automation Engineering
機械與自動化工程學系

Today's Agenda

- Image Formation and camera fundamentals
  - What is digital image
  - Image Formation
  - What is pin-hole camera
  - Camera Models: Lenses, Depth of Field, View Angle
  - Geometric transformation
  - Accidental Camera

- Image warping
  - Interpolation
  - Resampling
  - Applications

Today's Agenda

- Image Formation and camera fundamentals
  - What is digital image
  - Image Formation
  - What is pin-hole camera
  - Camera Models: Lenses, Depth of Field, View Angle
  - Geometric transformation
  - Accidental Camera

- Image warping
  - Interpolation
  - Resampling
  - Applications

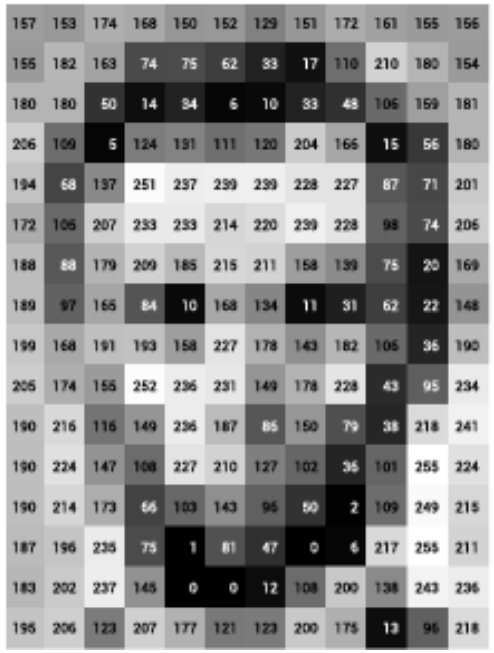
What is Digital Image?

http://en.wikipedia.org/wiki/Lenna

From **Lawrence G. Roberts's** master's thesis at MIT 1961

What is Digital Image

- An image is a 2D array of ***pixels***



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Source: http://techundred.com/how-snapchat-filter-work/

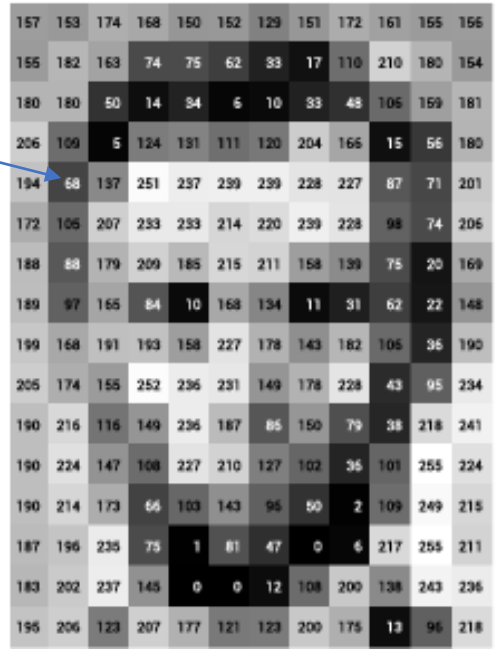
What is a Pixel

- ***Pixel*** is the basic element of a digital photo which is a sample of continuous function at a position

Pixel

Color at (x,y)

Intensity at (x,y)



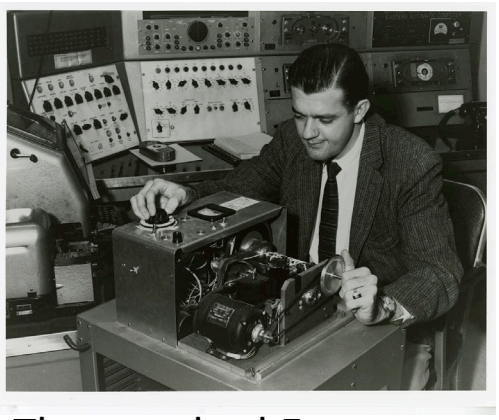
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

Source: http://techundred.com/how-snapchat-filter-work/

First Digital Image

- The first picture to be ***scanned***, stored, and recreated in digital pixels in 1957 by Russell Kirsch
- 176×176 pixel digital image by scanning a photograph of his three-month-old son



The standard Eastern Automatic Computer Scanner

By Russell A. Kirsch - National Institute of Standards and Technology, Public Domain, https://commons.wikimedia.org/w/index.php?curid=2437615

Types of Images

- Three Basic Types of images

Black and White
Just two colours

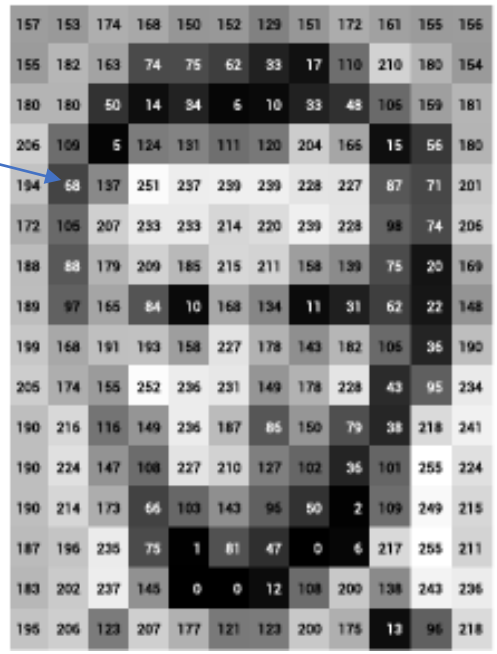
Grey Scale
0-255 diff. levels

Colour Image

Gray Scale Image

- ***Pixel*** is the basic element of a digital photo which is a sample of continuous function at a position

Pixel at x,y
f(x,y)=0~255



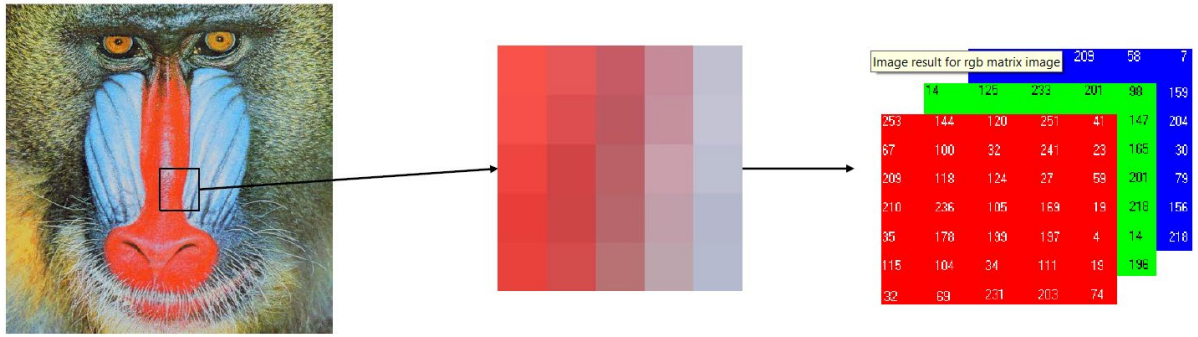
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

Source: http://techundred.com/how-snapchat-filter-work/

Colour Image

- An image is a ***2D*** array of pixels
- Color Image is a ***3D*** array
- Common Color mode
  - RGB: [R,G,B]
  - CMY [C,M,Y] usually with K
  - HSV: [H, S, V]
  - LAB: [L,A,B]

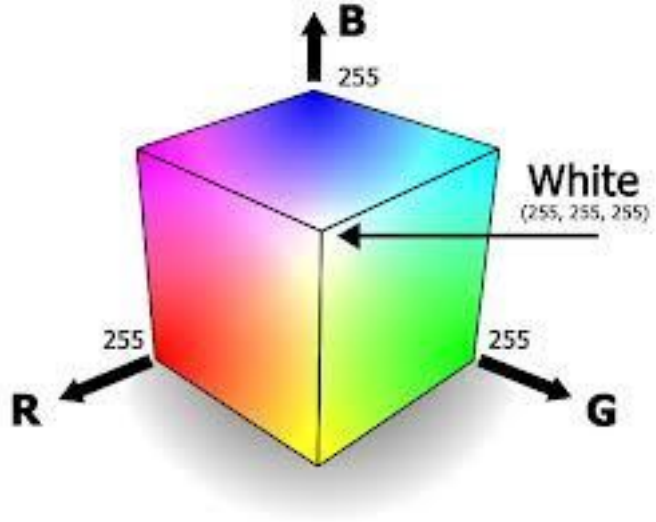


Source: R. Dockter

Colour spaces: RGB

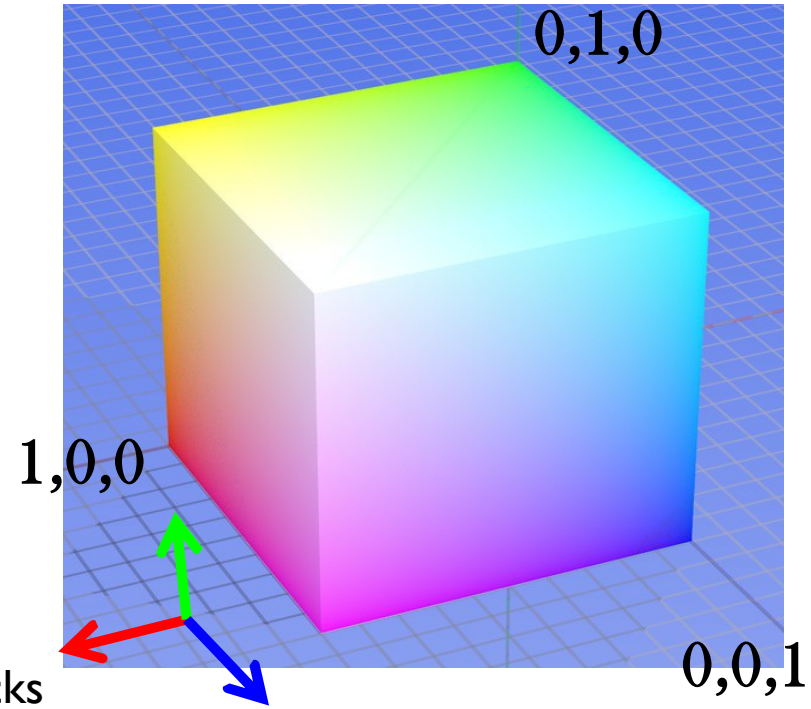
- Single wavelength primaries
- makes a ***particular monitor*** RGB standard
- Good for devices (e.g., phosphors for monitor)

- Applications
  - Widely used in representation and display of image in ***computer*** and ***television***
  - Adopted in ***photography***
  - Common in ***web graphics***



Colour spaces: RGB

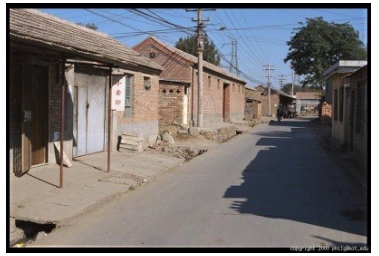
Default color space



R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)



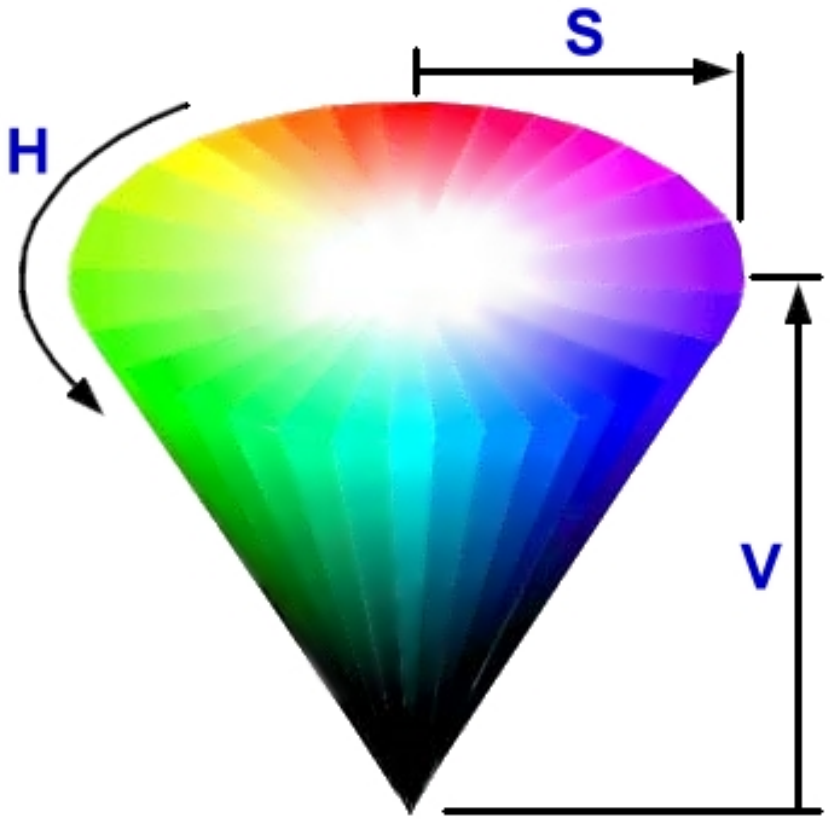
Some drawbacks

- Strongly correlated channels
- Non-perceptual

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

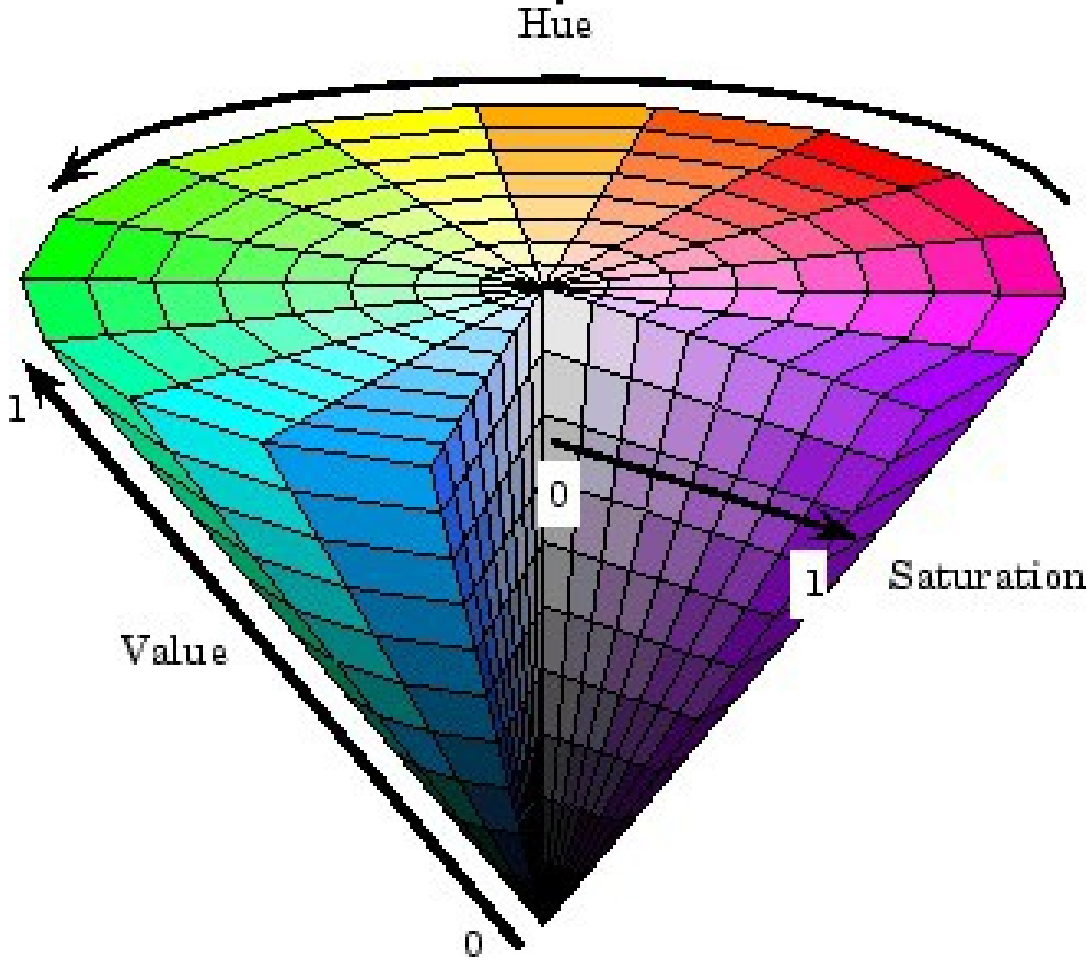
Slide credit: D. Hoiem

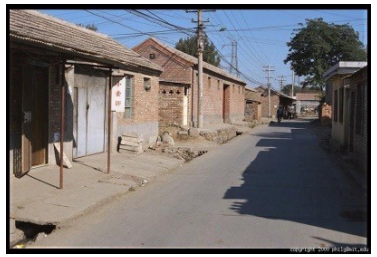
HSV Colour Space

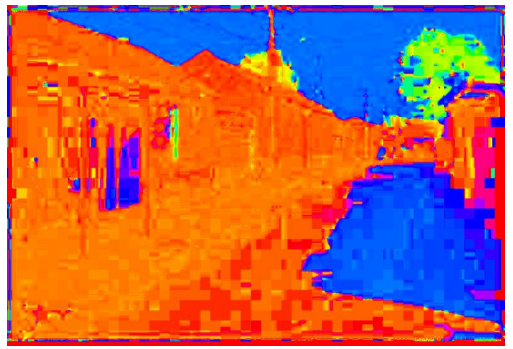


- *H (hue)* **:** is the dominant wavelength of the colour, e.g. red, blue, green
- *S (saturation)* **:** is the 'purity' of colour (in the sense of the amount of white light mixed with it)
- *V (value)* **:** is the brightness of the colour (also known as luminance).
- MATLAB: `hsv2rgb`, `rgb2hsv`.

HSV Color Space







H

(S=1,V=1)

S

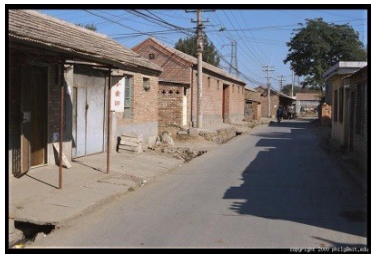
(H=1,V=1)

V

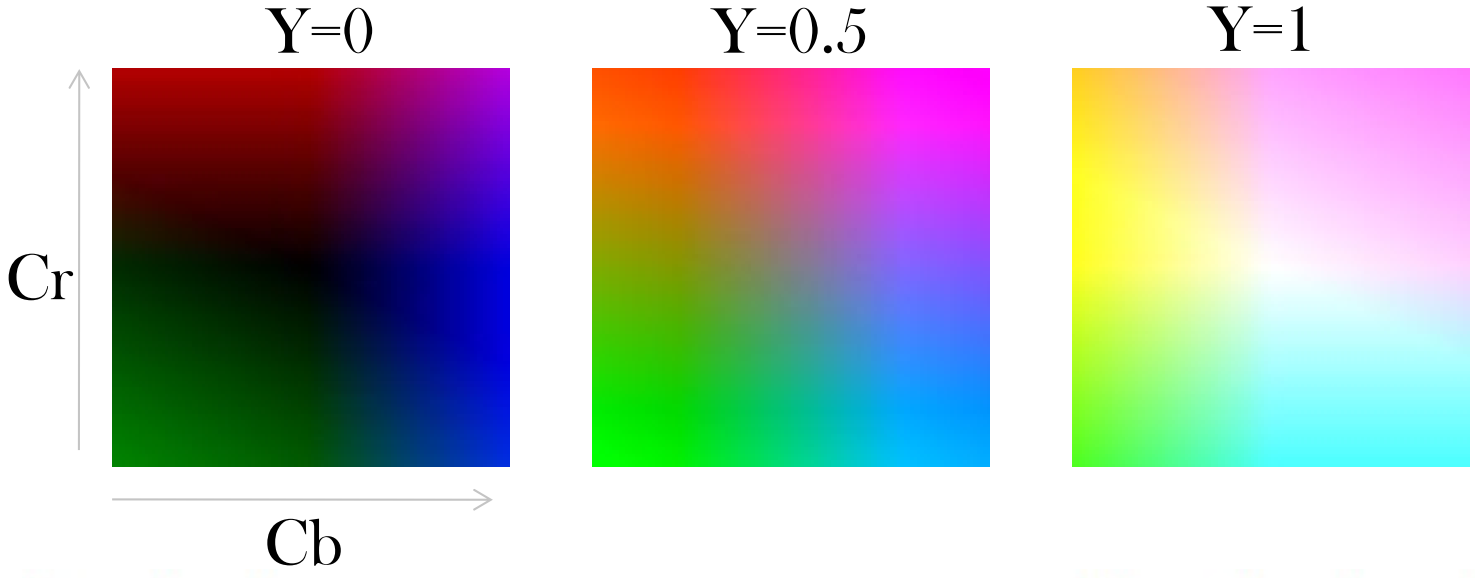
(H=1,S=0)

Slide credit: D. Hoiem

YCbCr Color Space



Fast to compute, good for compression, used by TV



$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

[MS-RDPRFX]: Color Conversion (RGB to YCbCr) | Microsoft Docs

Y

(Cb=0.5,Cr=0.5)

Cb

(Y=0.5,Cr=0.5)

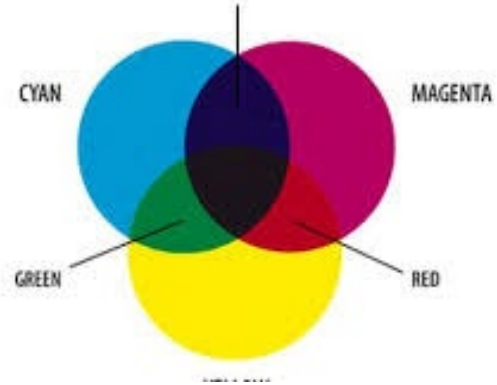
Cr

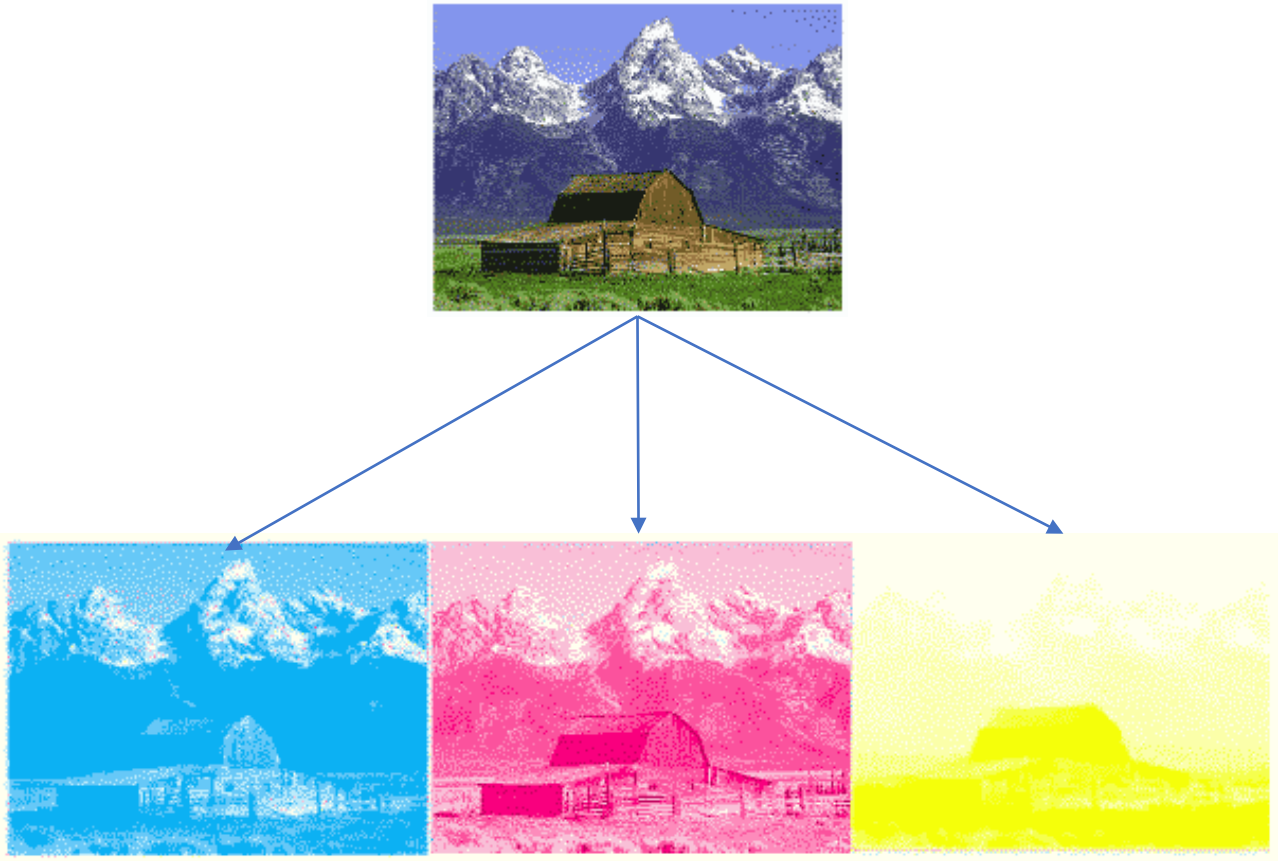
(Y=0.5,Cb=05)

Slide credit: D. Hoiem

CMY Colour

- Contains secondar colours
- $\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = 1 - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$

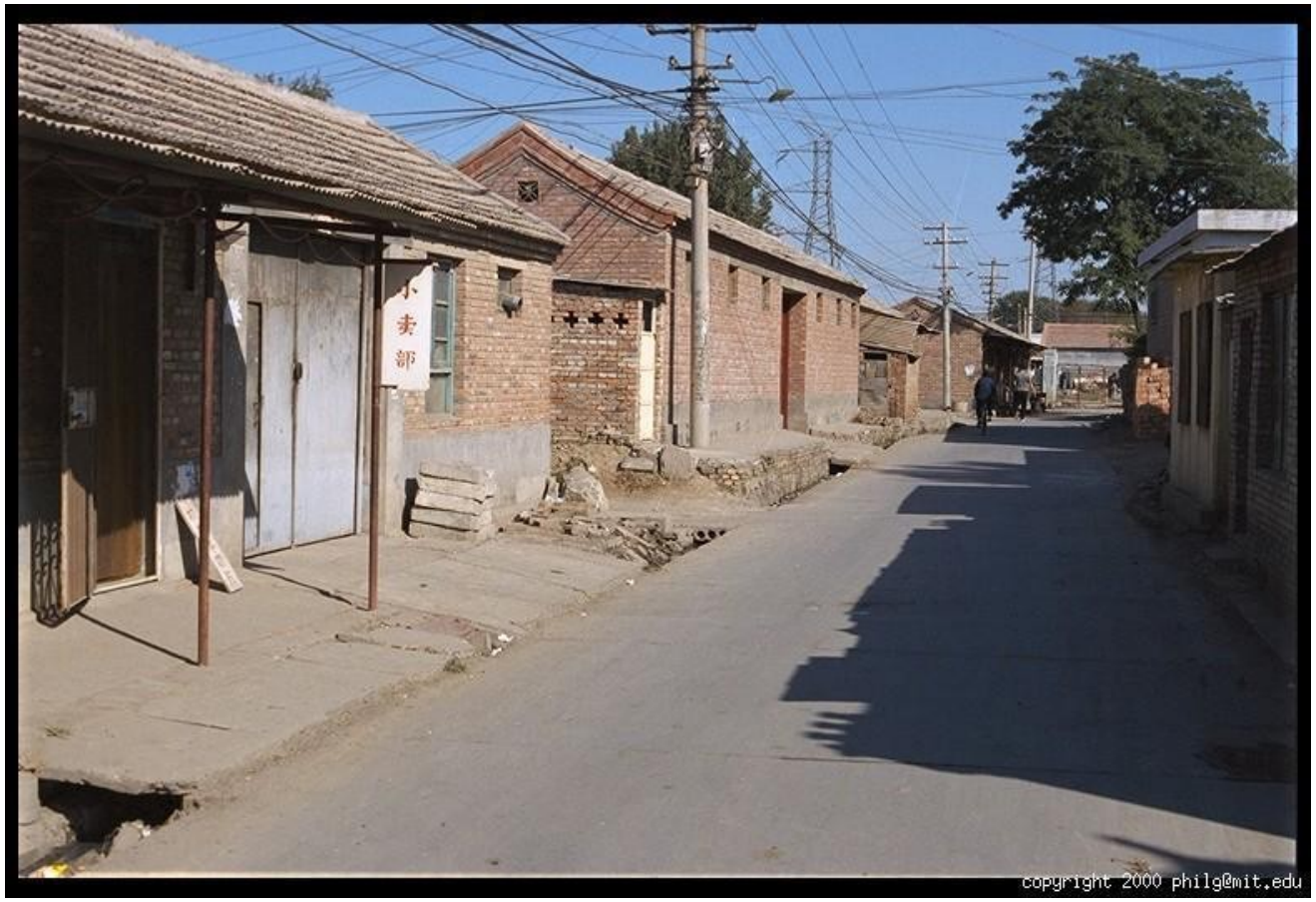




- Commonly used in colour printing

https://www.youtube.com/watch?v=YtH9eXWuf3Y

However, most information in intensity



Original image

Slide credit: D. Hoiem

Most information in intensity

Only intensity shown – constant color

Slide credit: D. Hoiem

Back to grayscale intensity

Slide credit: D. Hoiem

Today's Agenda

- Image Formation and camera fundamentals
  - What is digital image
  - Image Formation
  - What is pin-hole camera
  - Camera Models: Lenses, Depth of Field, View Angle
  - Geometric transformation
  - Accidental Camera
  - Other types of cameras

- Image warping
  - Interpolation
  - Resampling
  - Applications

Image Formation

How an image is formed?

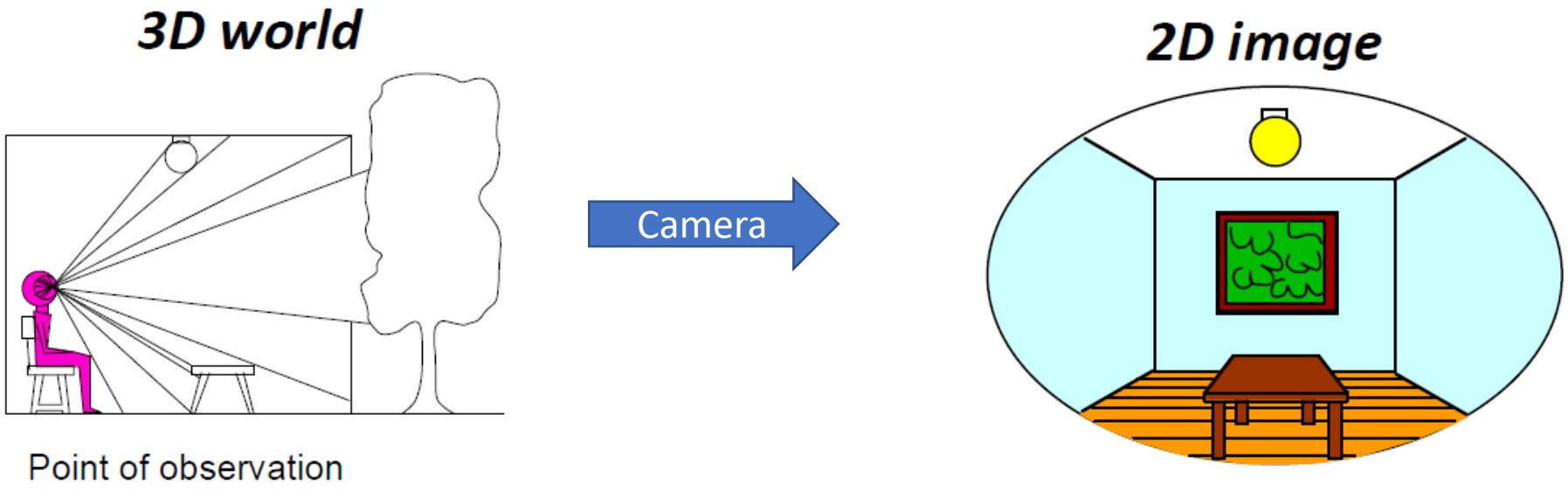
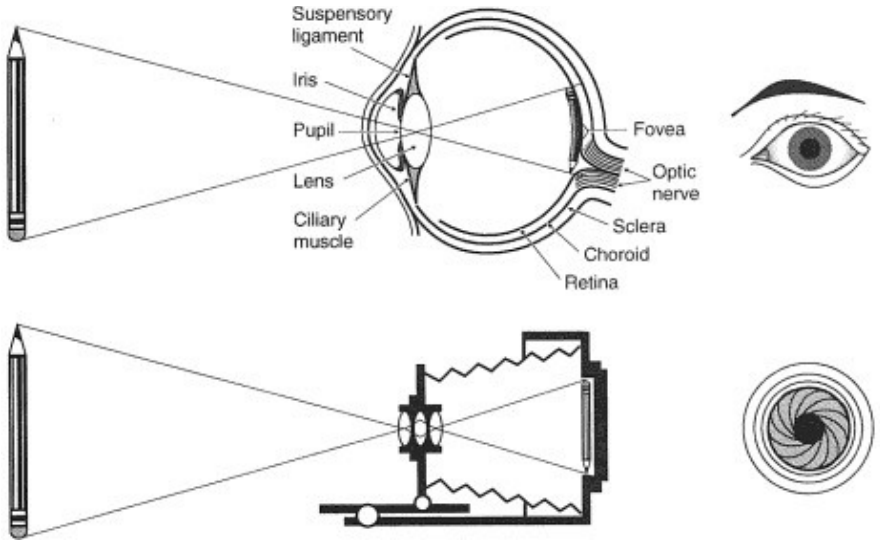


Image Formation



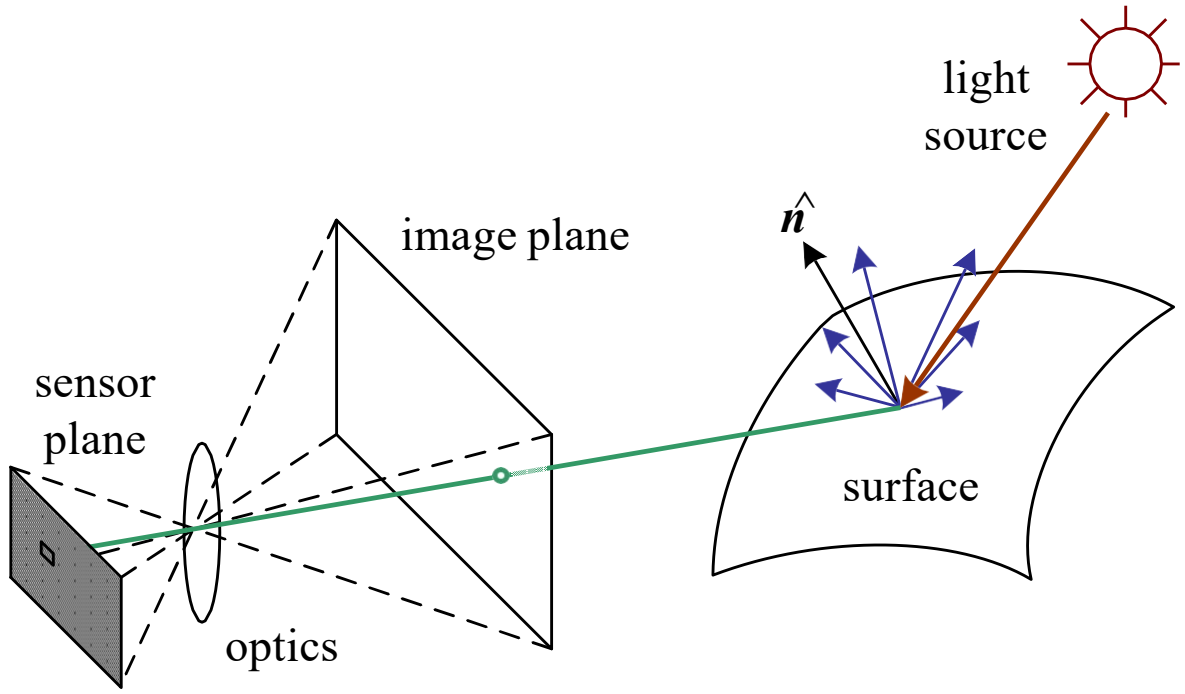
Three Dimensional World → Two Dimensional Image Space

- What is measured in an image location?
  - brightness
  - color

<< illumination conditions
local geometry
local material properties

Figures: Francis Crick, The Astonishing Hypothesis,

Image Formation

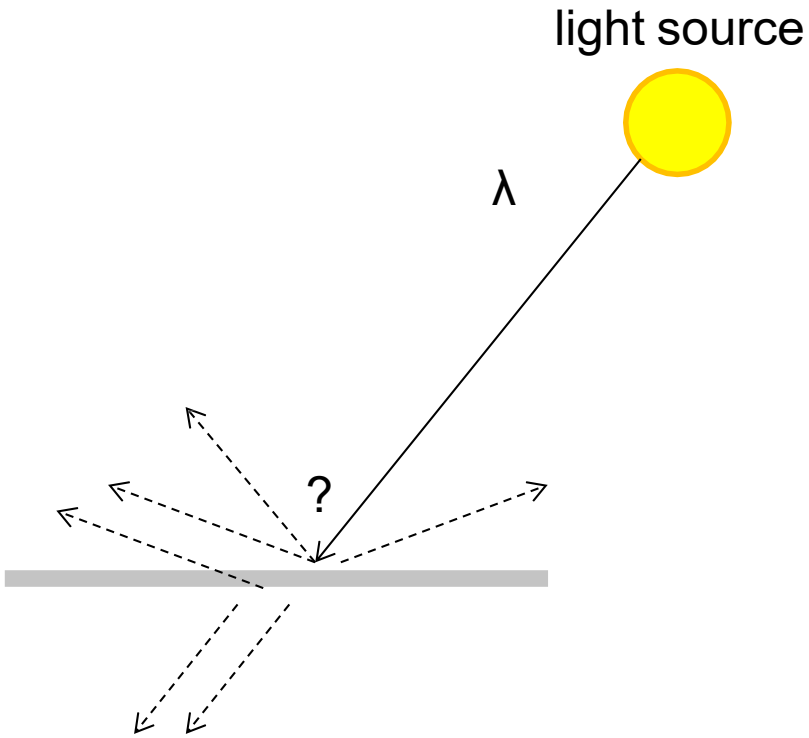


Three Dimensional World → Two Dimensional Image Space

Figures: Francis Crick, The Astonishing Hypothesis

A photon's life choices

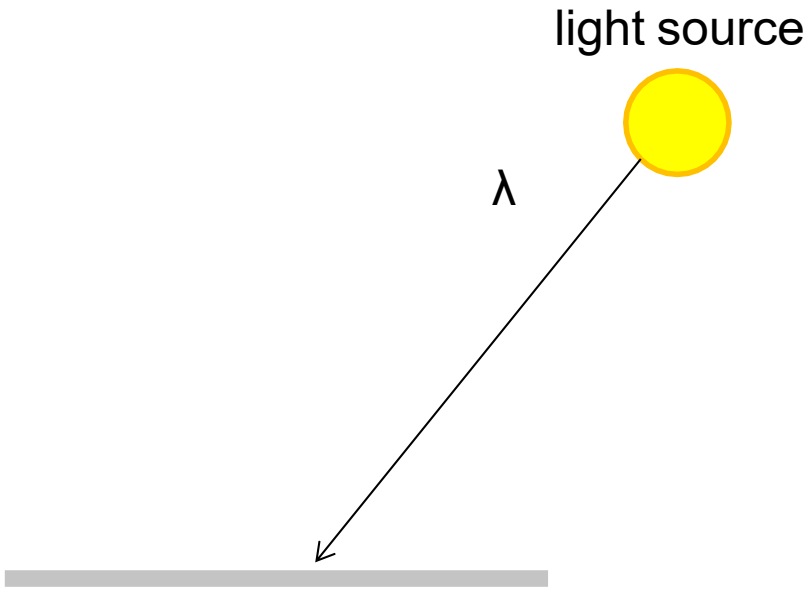
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

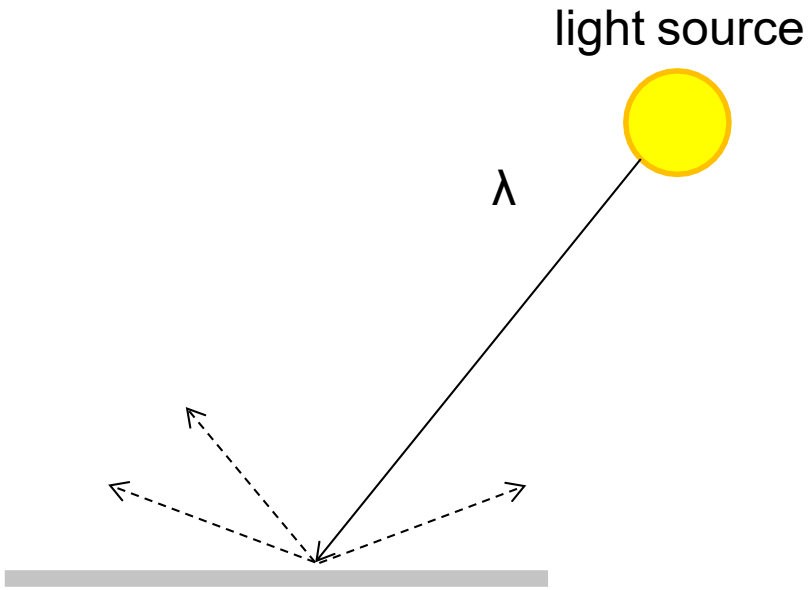
A photon’s life choices

- **Absorption**
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



A photon’s life choices

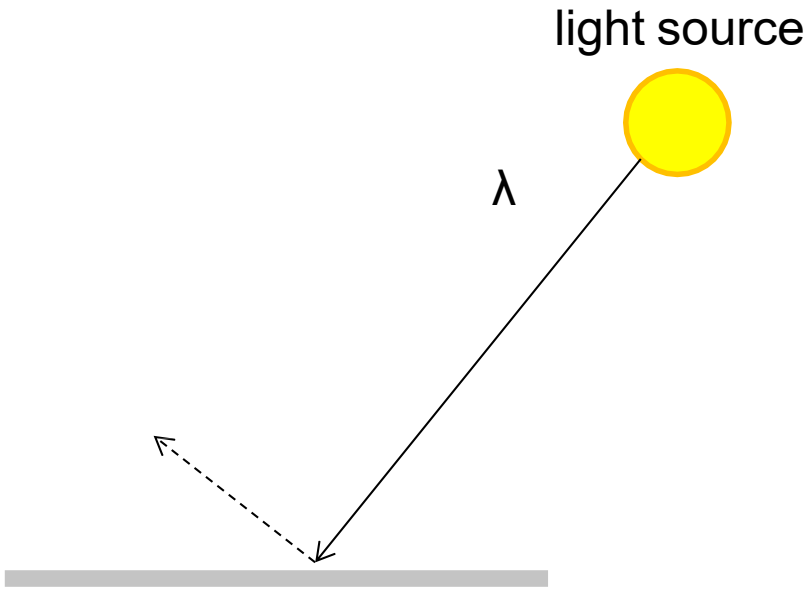
- Absorption
- **Diffusion**
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

A photon’s life choices

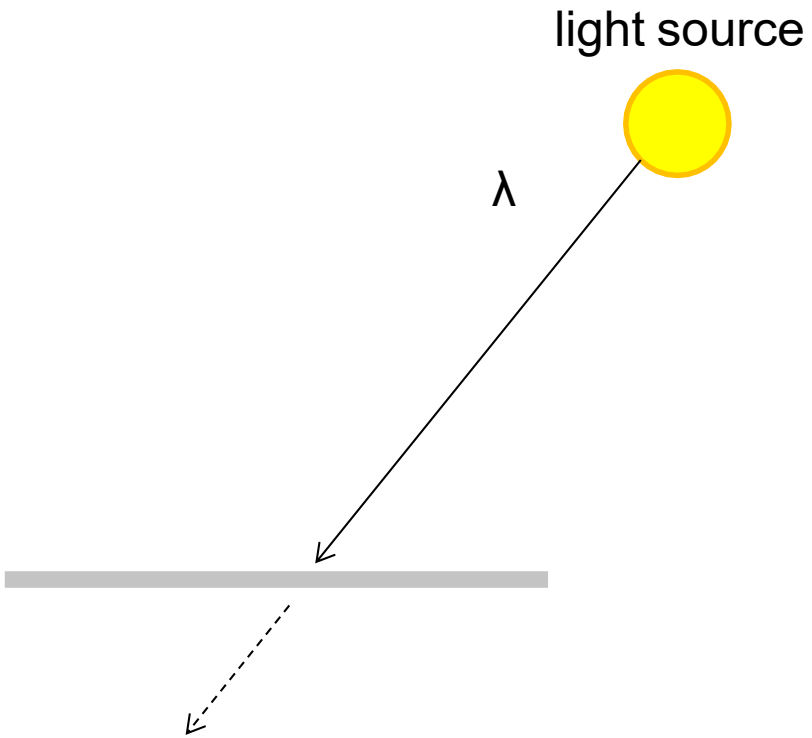
- Absorption
- Diffusion
- **Reflection**
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

A photon’s life choices

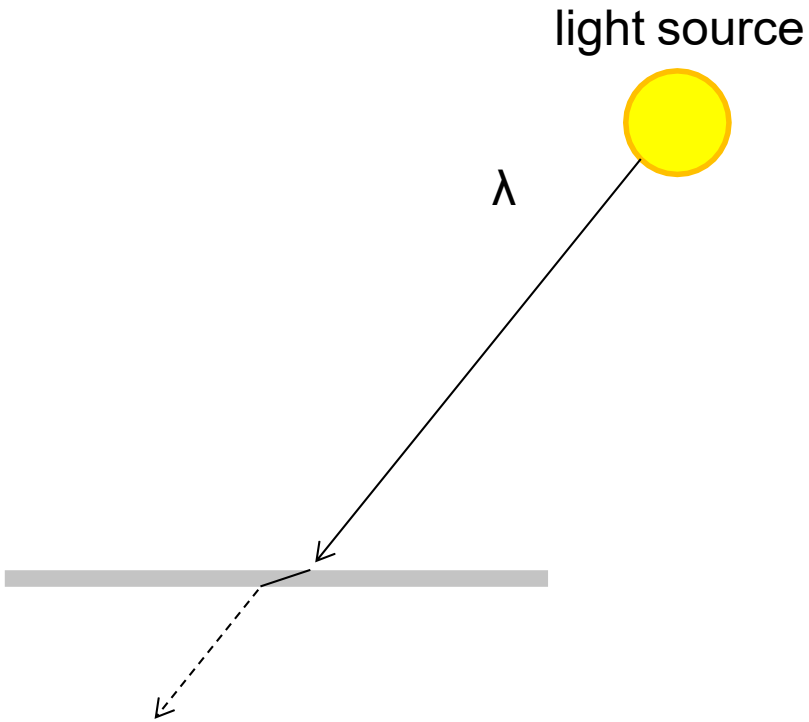
- Absorption
- Diffusion
- Reflection
- **Transparency**
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

A photon’s life choices

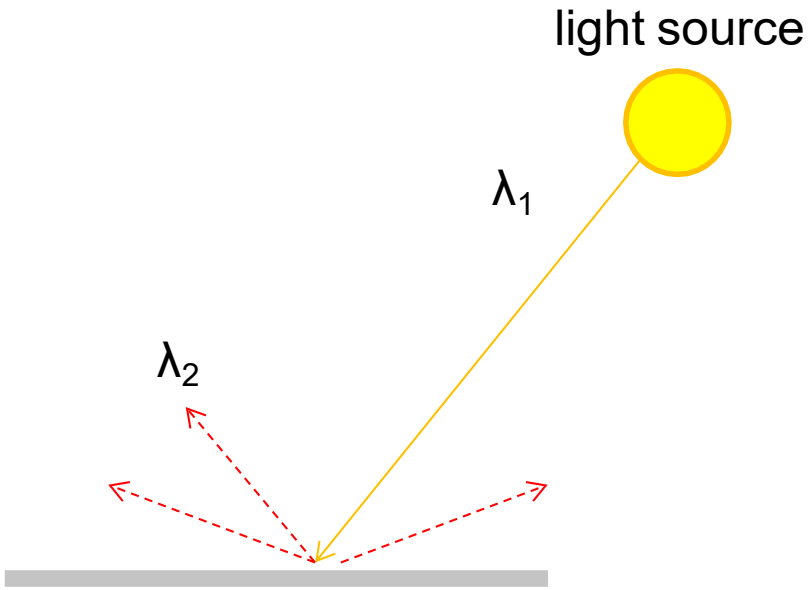
- Absorption
- Diffusion
- Reflection
- Transparency
- **Refraction**
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

A photon's life choices

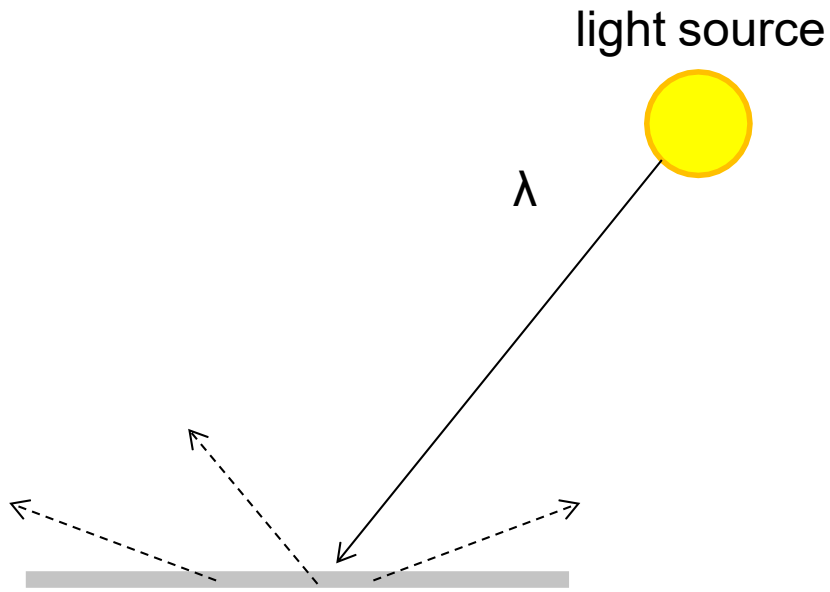
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- **Fluorescence**
- Subsurface scattering
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

A photon’s life choices

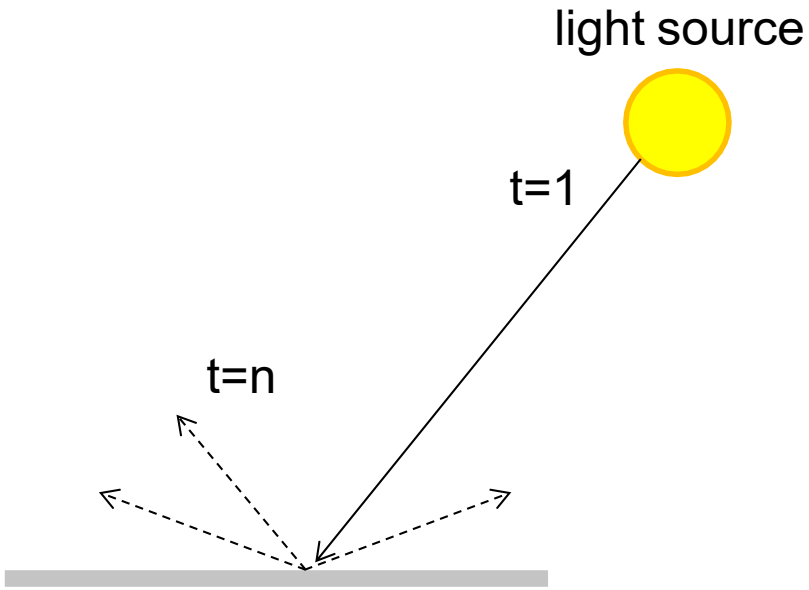
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- **Subsurface scattering**
- Phosphorescence
- Interreflection



Slides adopted from E. Erdem.

A photon’s life choices

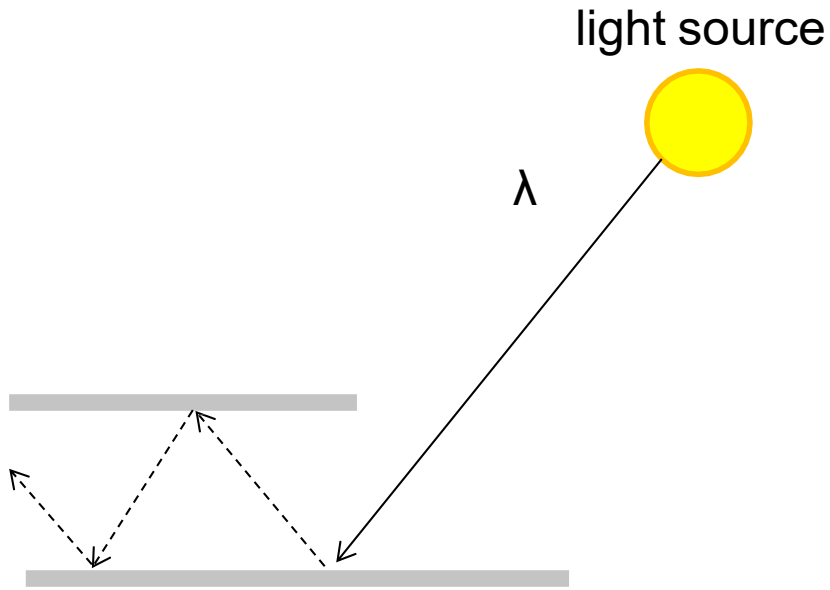
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- **Phosphorescence**
- Interreflection



Slides adopted from E. Erdem.

A photon’s life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- **Interreflection**

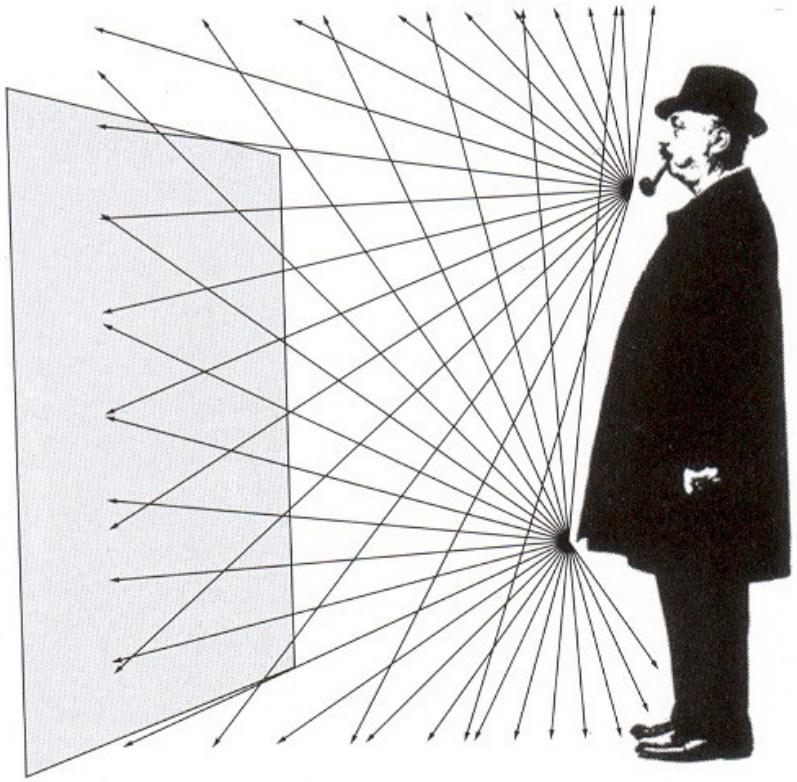


(Specular Interreflection)

Slides adopted from E. Erdem.

Image Formation

- Images cannot exist without light!
- Why is there no image on a white piece of paper?
- It receives light from all directions



From Photography, London et al.

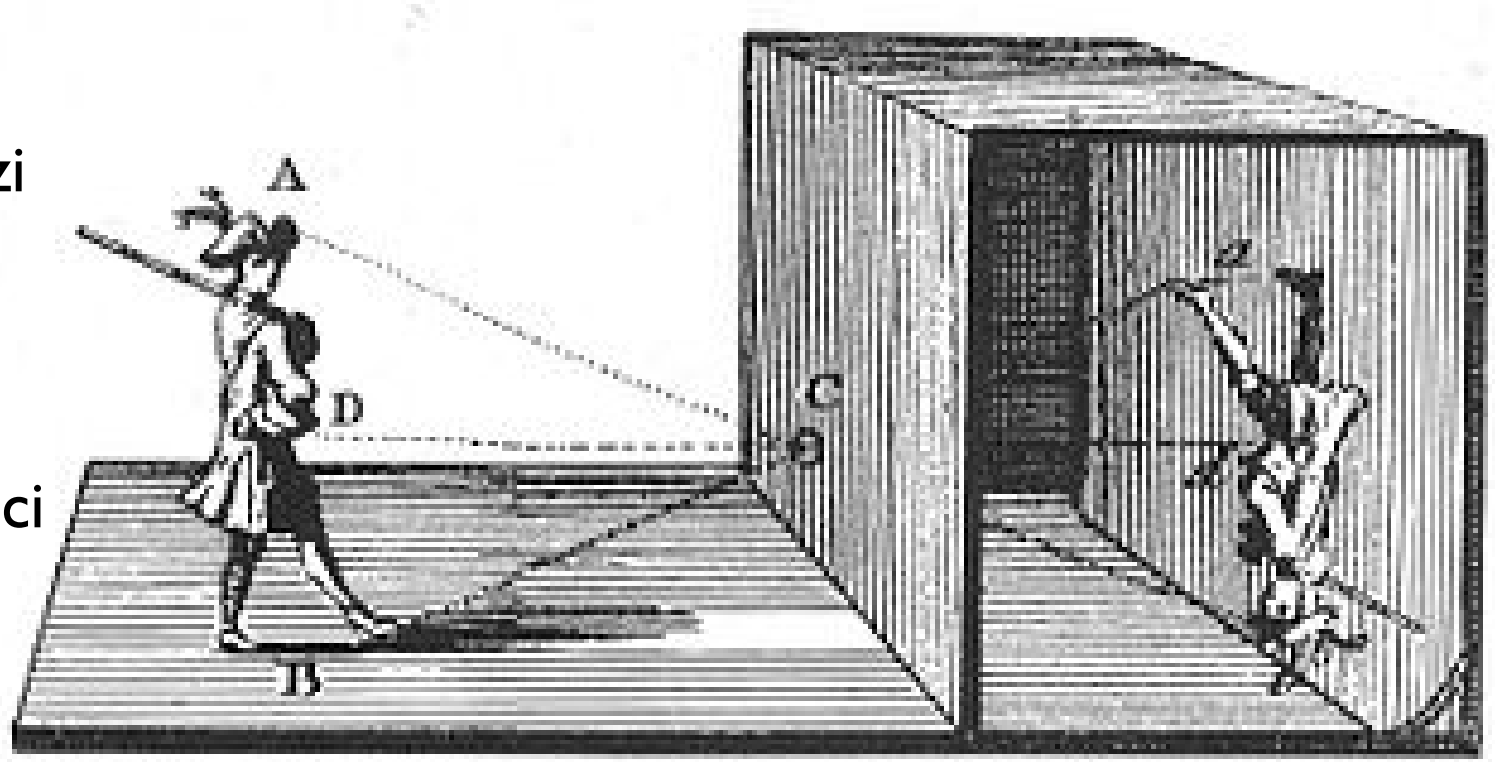
Today's Agenda

- Image Formation and camera fundamentals
  - What is digital image
  - Image Formation
  - What is pin-hole camera
  - Camera Models: Lenses, Depth of Field, View Angle
  - Geometric transformation
  - Accidental Camera
- Image warping
  - Interpolation
  - Resampling
  - Applications

Camera obscura

- **Camera obscura** means dark room *camera* chamber also refer to a Pinhole Camera

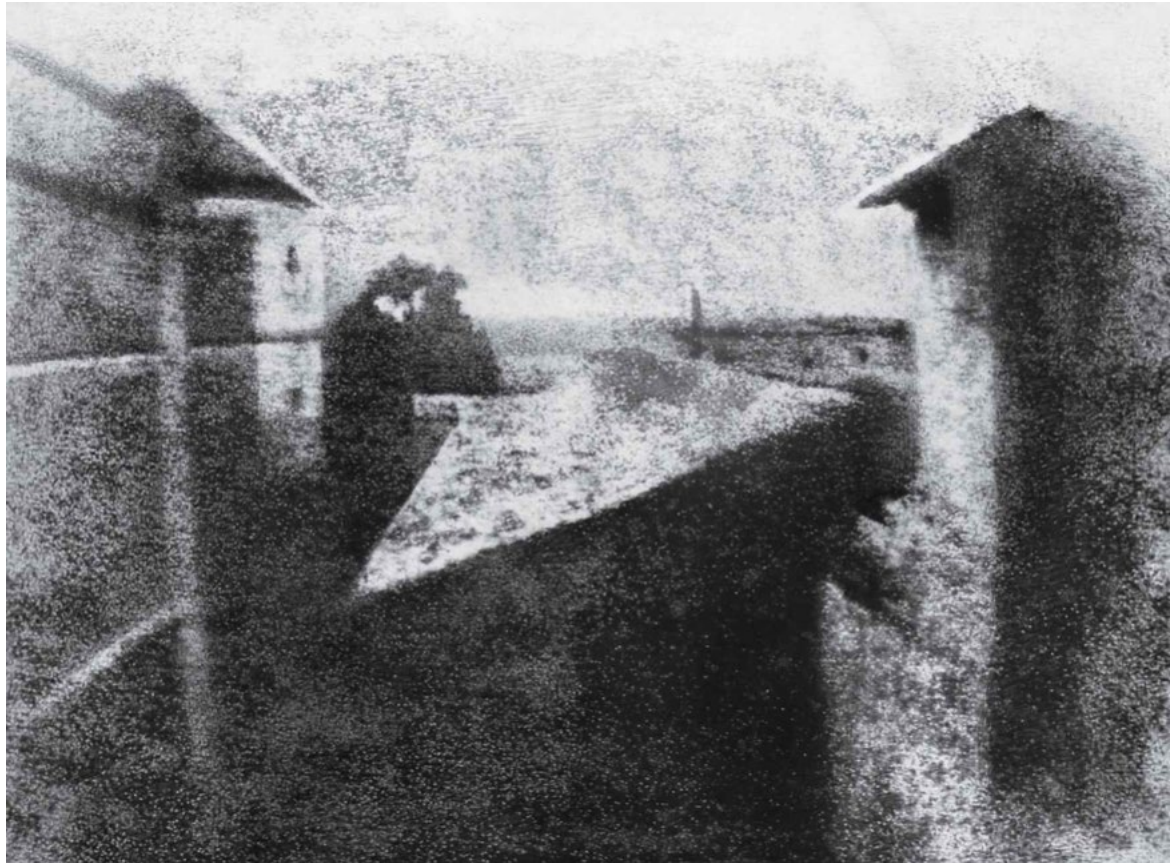
- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo daVinci (1452-1519)



https://en.wikipedia.org/wiki/Camera_obscura

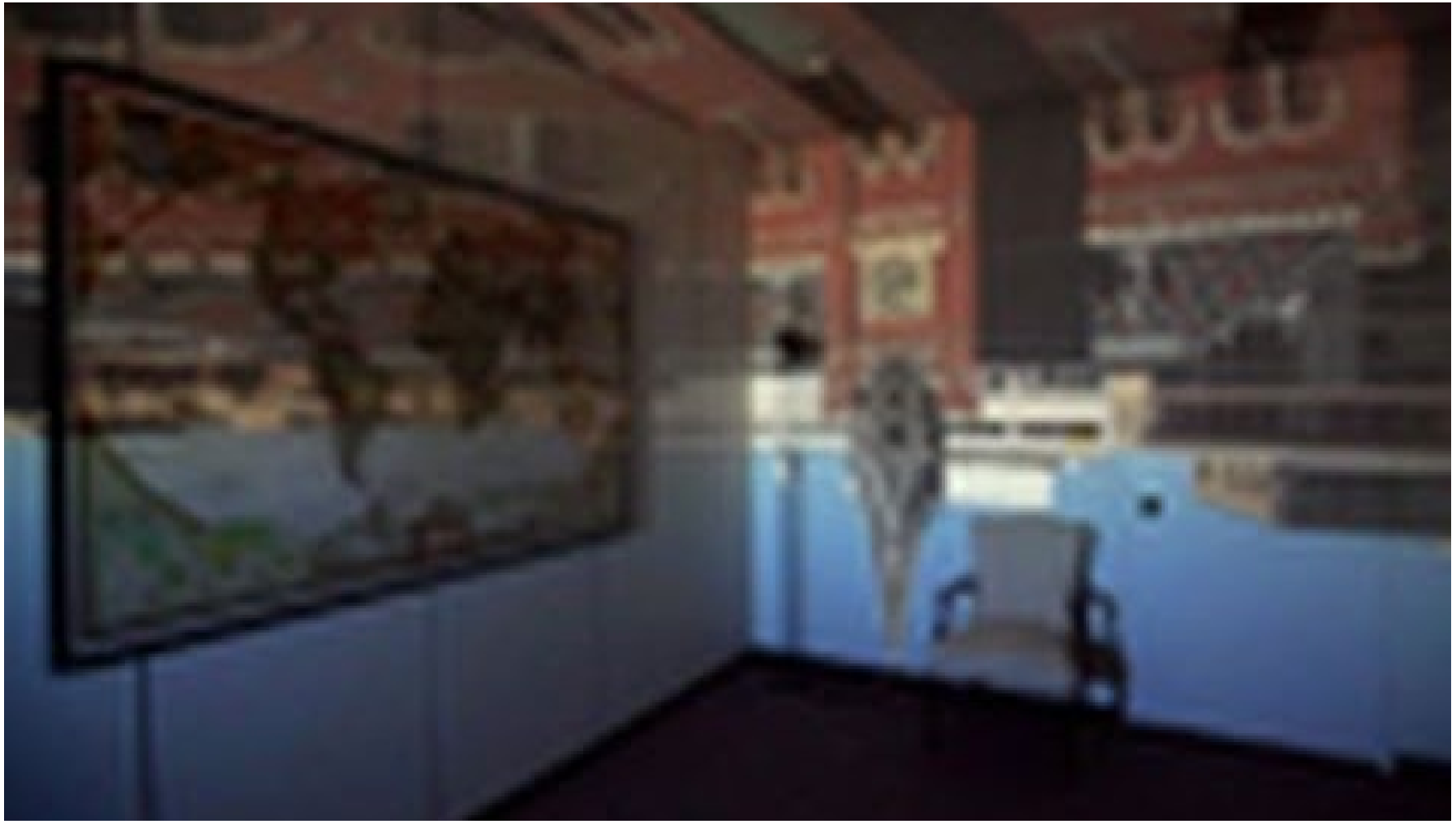
Camera obscura

- Joseph Nicéphore Niépce made the first camera at 1814
- First image was taken by shutter open for 8 hours



First image taken by Nicéphore Niépce

Demo



Today's Agenda

- Image Formation and camera fundamentals
  - What is digital image
  - Image Formation
  - What is pin-hole camera
  - Camera Models: Lenses, Depth of Field, View Angle
  - Geometric transformation
  - Accidental Camera

- Image warping
  - Interpolation
  - Resampling
  - Applications

Pinhole Camera Model

- The centre of project is ***camera centre***, also known as the ***optical centre***.
- The line from the camera centre perpendicular to the image plane is called the ***principal axis*** or ***principal ray*** of the camera
- The point where the principal axis meets the image plane is called ***principal point***
- The plane through the camera centre parallel to the image plane is called ***principal plane***

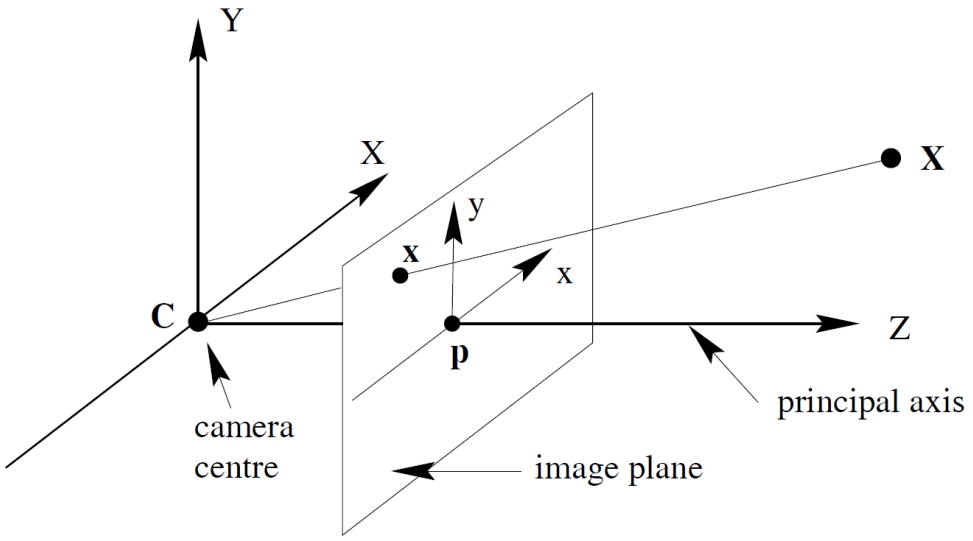


Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

Camera Models – Pinhole Camera

- *By Similar Triangle*

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

- One can quickly computes the point

$$(X, Y, Z)^{\mathrm{T}} \rightarrow \left(\frac{f\mathrm{X}}{\mathrm{Z}}, \frac{f\mathrm{Y}}{\mathrm{Z}}, f\right)^{\mathrm{T}}$$

- Describes the ***central projection*** mapping from ***world*** to ***image coordinates***

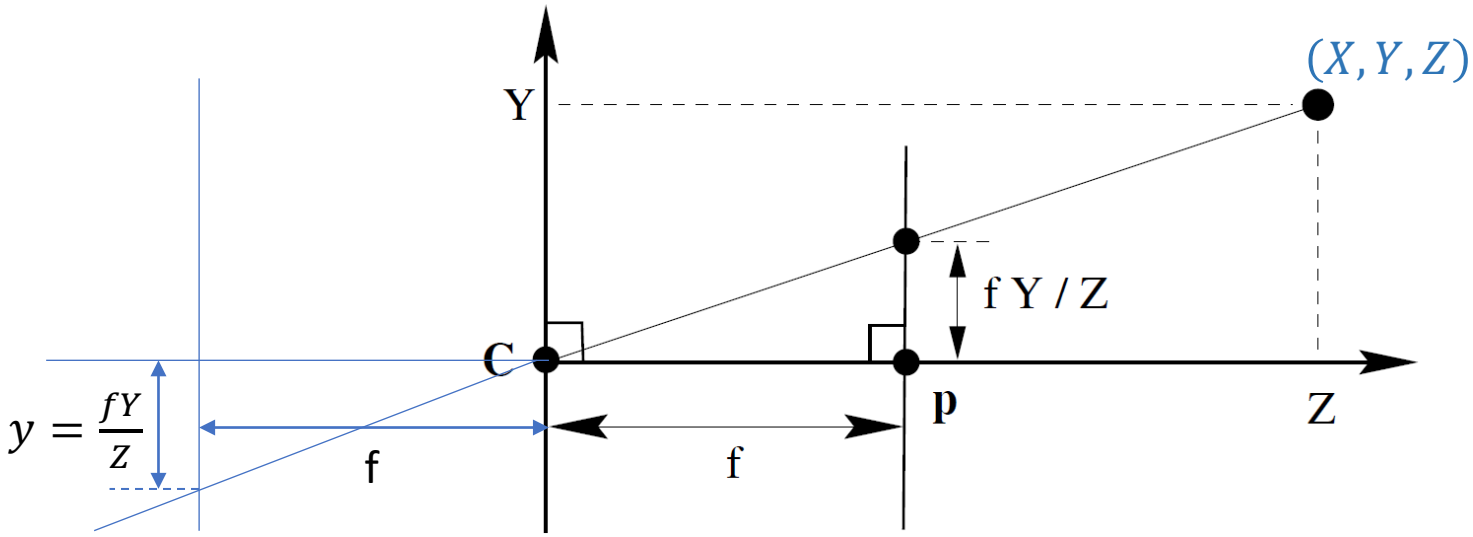


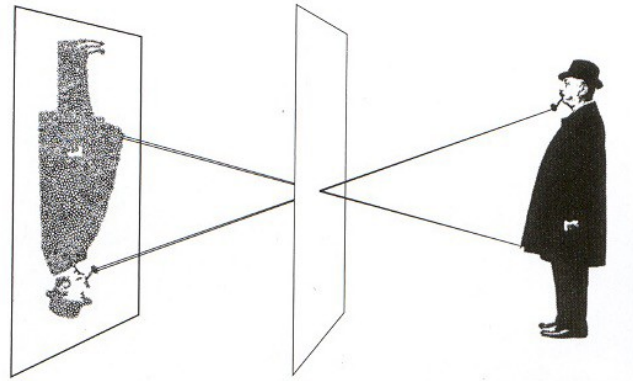
Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

Factors affecting the image formation - Pinhole

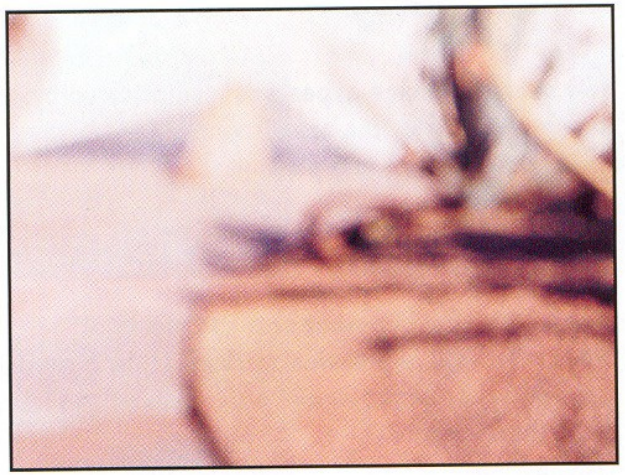
- Small pinhole
  - Long exposure time
  - High intensity

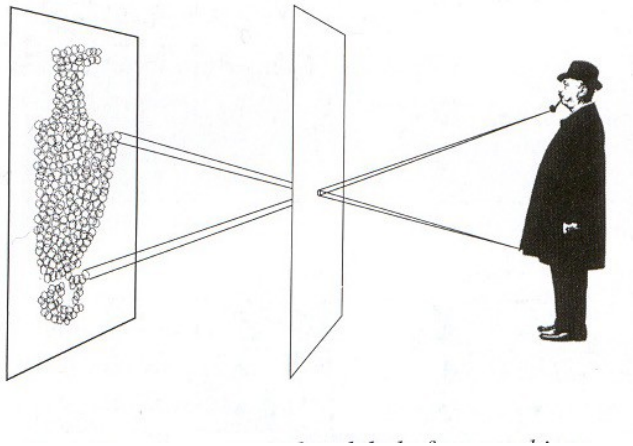
- Large pinhole
  - Blurry image

Photograph made with small pinhole



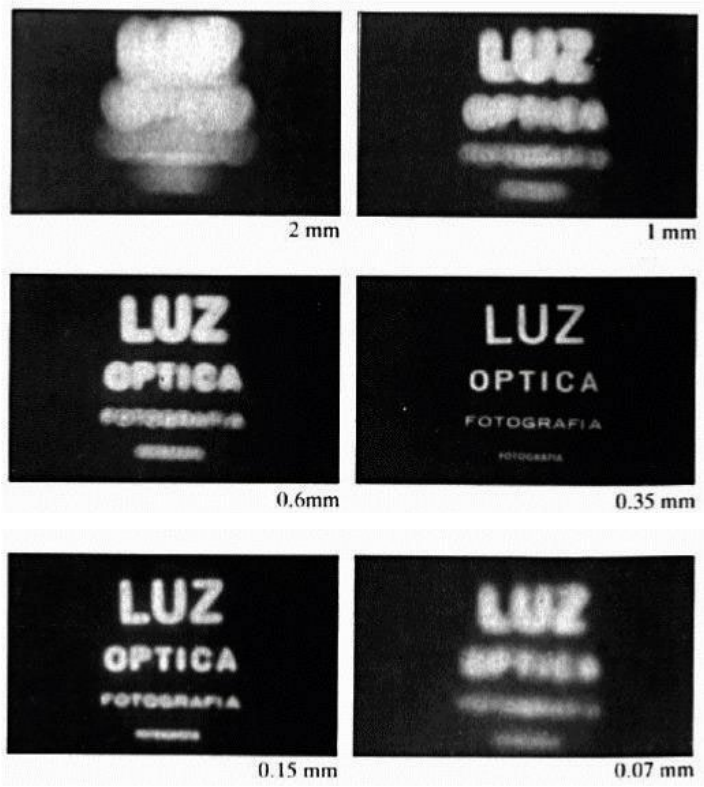
Photograph made with larger pinhole





Effect of different pinhole size.

- Very Large pinhole – multiple rays projected onto the same point -> ***blurry image***
- Very Small pinhole – diffraction effect -> ***blurry image***
- ***Optimal*** pinhole size
  - Pinhole diameter $d = 2\sqrt{f'\lambda}$
  - Example f' = 50mm, λ=600nm(red) d= 0.36mm (Source: https://en.wikipedia.org/wiki/Pinhole_camera#Selection_of_pinhole_size)



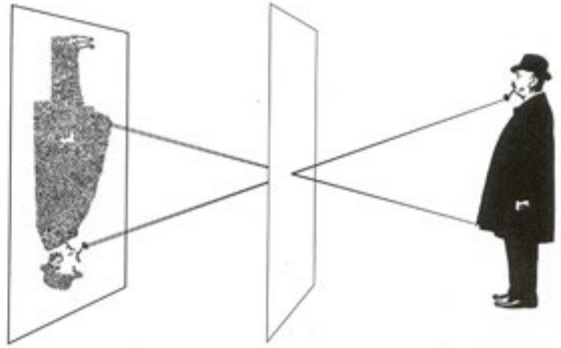
Source: R. Dockter

Lenses VS Pinhole?

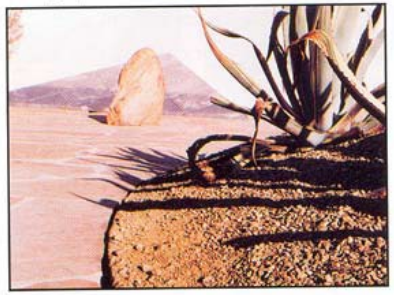
- Allow more light rays pass through
- Align multiple light rays
- Alignment works for one distance only

Photograph made with small pinhole

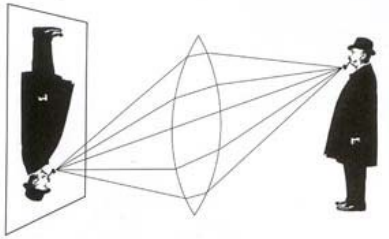
To make this picture, the lens of a camera was replaced with a thin metal disk pierced by a tiny pinhole, *equivalent in size to an aperture of f/182. Only a few rays of light from each point on the subject got through the tiny opening, producing a soft but acceptably clear photograph. Because of the small size of the pinhole, the exposure had to be 6 sec long.*



Photograph made with lens



This time, using a simple convex lens with an f/16 aperture, *the scene appeared sharper than the one taken with the smaller pinhole, and the exposure time was much shorter, only 1/100 sec.*

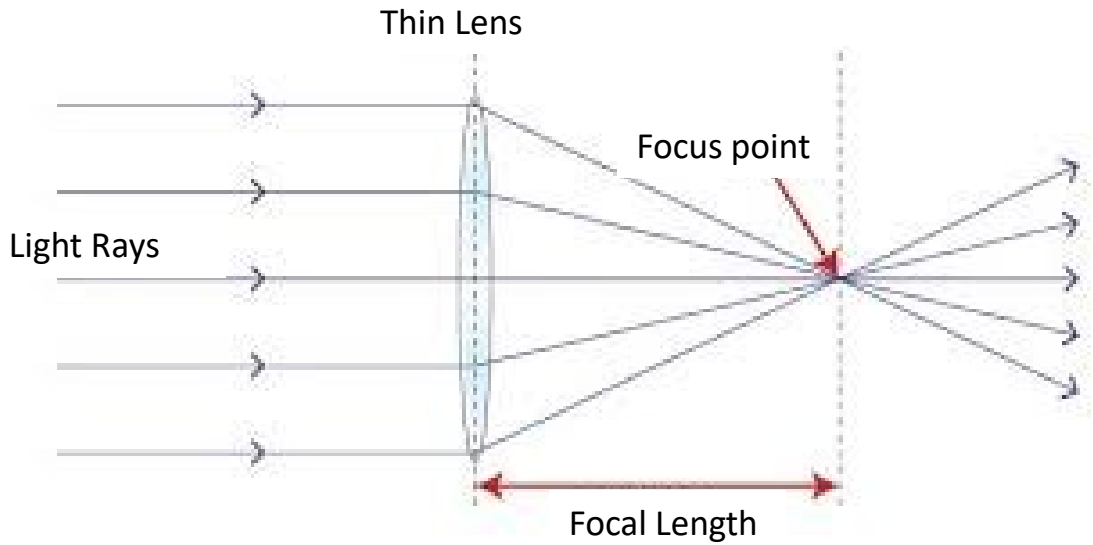


The lens opening was much bigger than the pinhole, letting in far more light, but it focused the rays from each point on the subject precisely so that they were sharp on the film.

From Photography, London et al.

Lenses

- A Lens focuses parallel rays onto a single point – focal point
  - Focal point is at distance f’ behind the lens
  - F’ is function of shape and refractive index of the lens
- Lenses are typically spherical



https://www.miniphysics.com

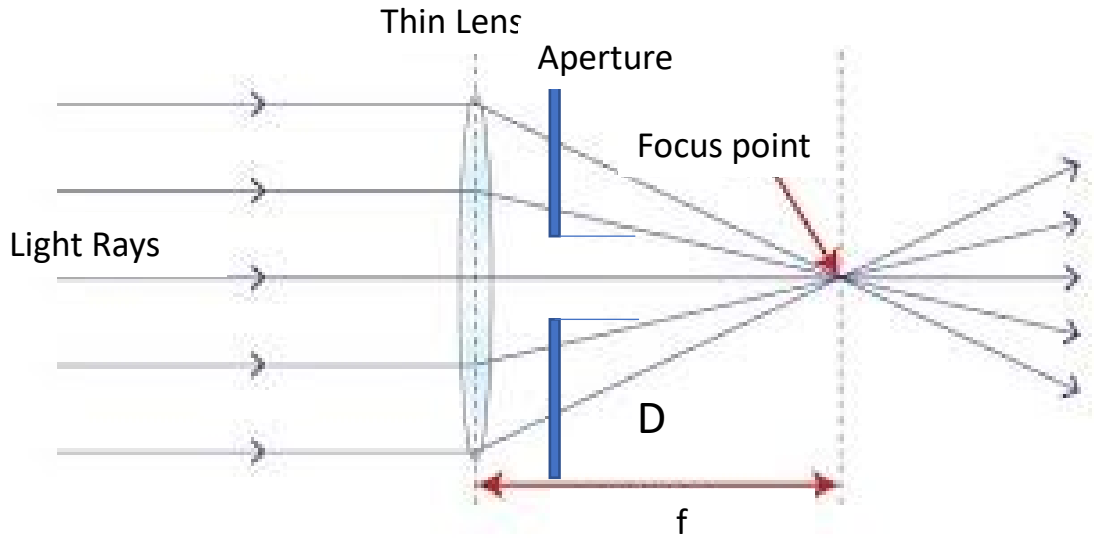
Factors affecting the image formation - Aperture

- In optics, an ***aperture*** is a hole or an ***opening*** through which light travels
- More specifically, the ***aperture*** of an optical system is the opening which determines the ***cone angle of a bundle of rays*** that come to a focus in the image plane

Factors affecting the image formation - Aperture

- ***Aperture F-Number*** is measured by
  - $N = \frac{f}{D}$

- ***Lens Power P*** is the degree to which a lens ***converges*** or ***diverges*** light.
  - $P = \frac{1}{f}$ unit is dioptre (D) or m^-1



https://www.miniphysics.com

Aperture

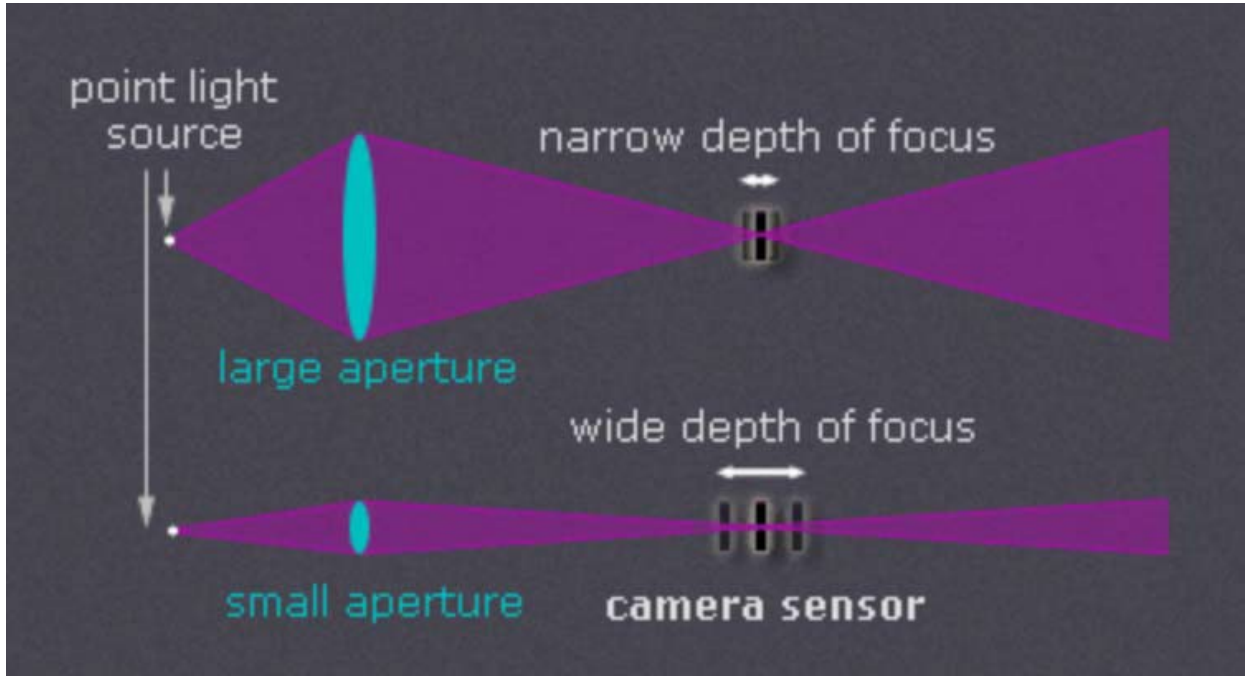
- Aperture controls the radius of hole through which light can pass

f/1.4 f/5.6 f/16

- F-number is diameter of aperture relative to focal length

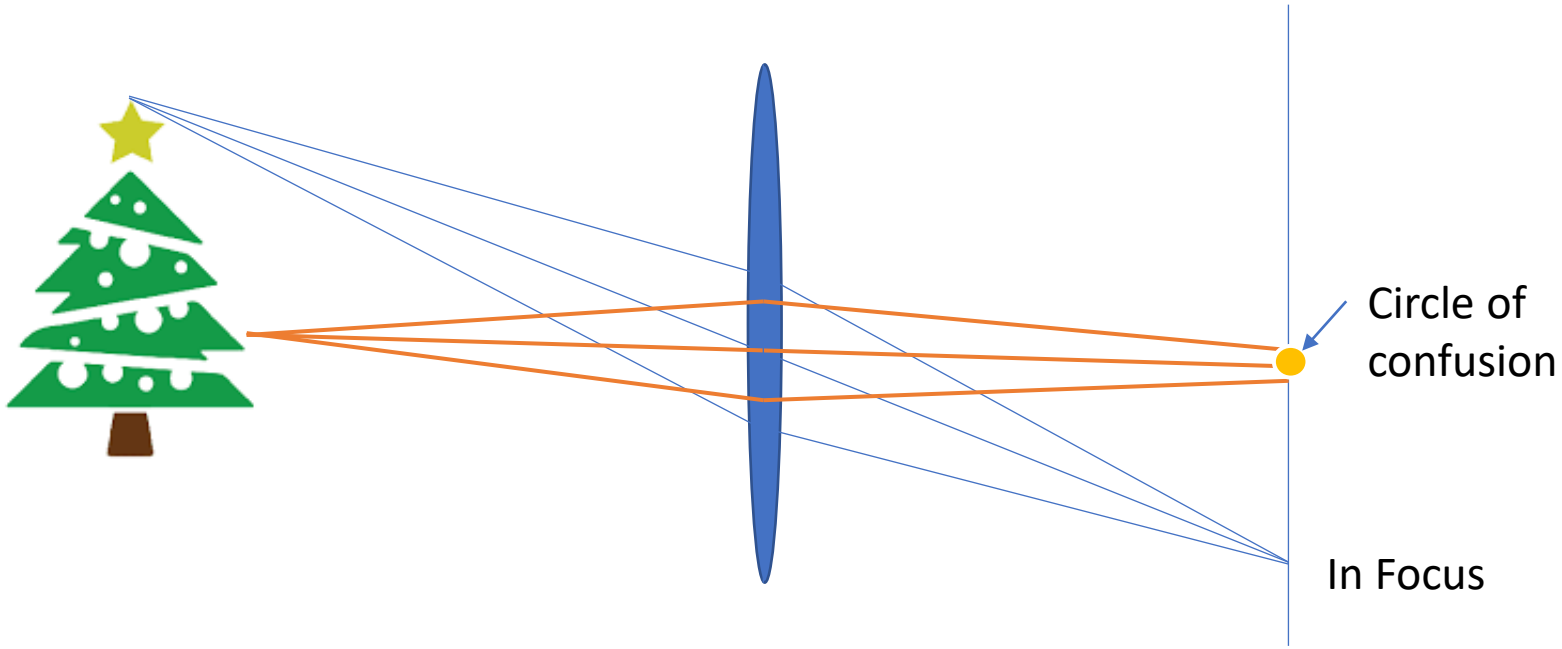
Effect on Different Aperture Sizes

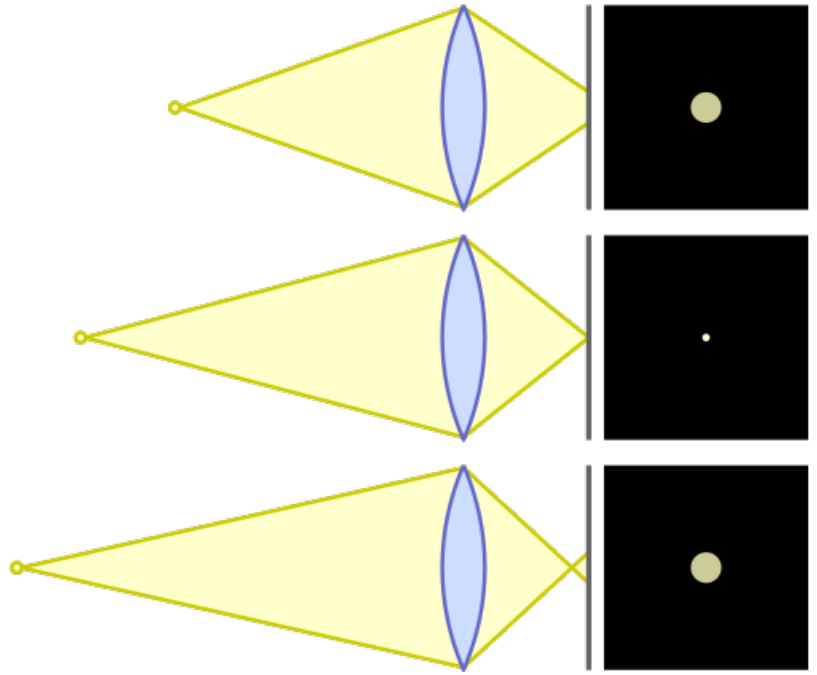
- Smaller apertures
- Less light in
- Larger depth of field



Depth of Field

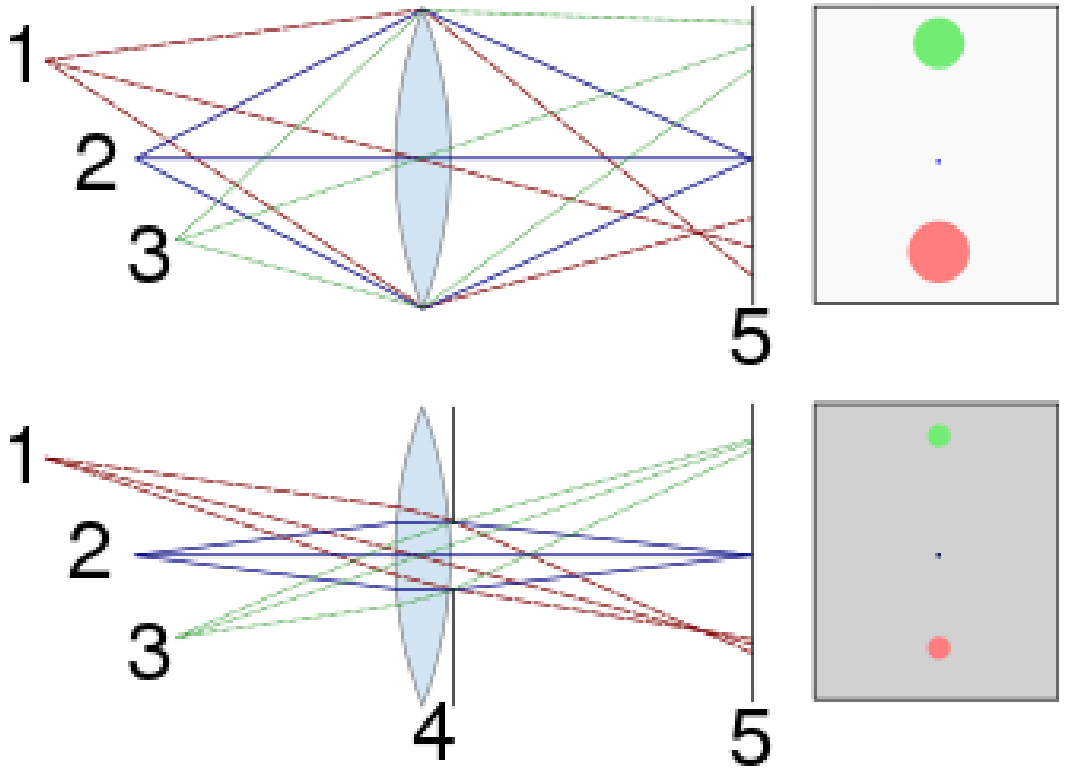
- Depth of Field (DOF) is the distance between the nearest and the farthest object that are in acceptably sharp focus in an image
- DOF varies will lens shape, distance to subject and aperture size





https://en.wikipedia.org

Depth of Field



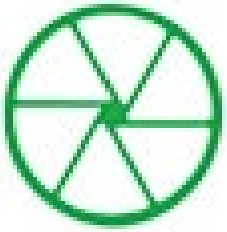
https://en.wikipedia.org

- Point 2 is in focus
- Point 1&3 project burry images
- Decreasing Aperture size reduces the blur spots
- Decreasing Aperture size reduces amount of light -> need longer exposure time

Depth of Field

f/1.4

f/4



f/22

Aperture = f/1.4. DOF=0.8 cm

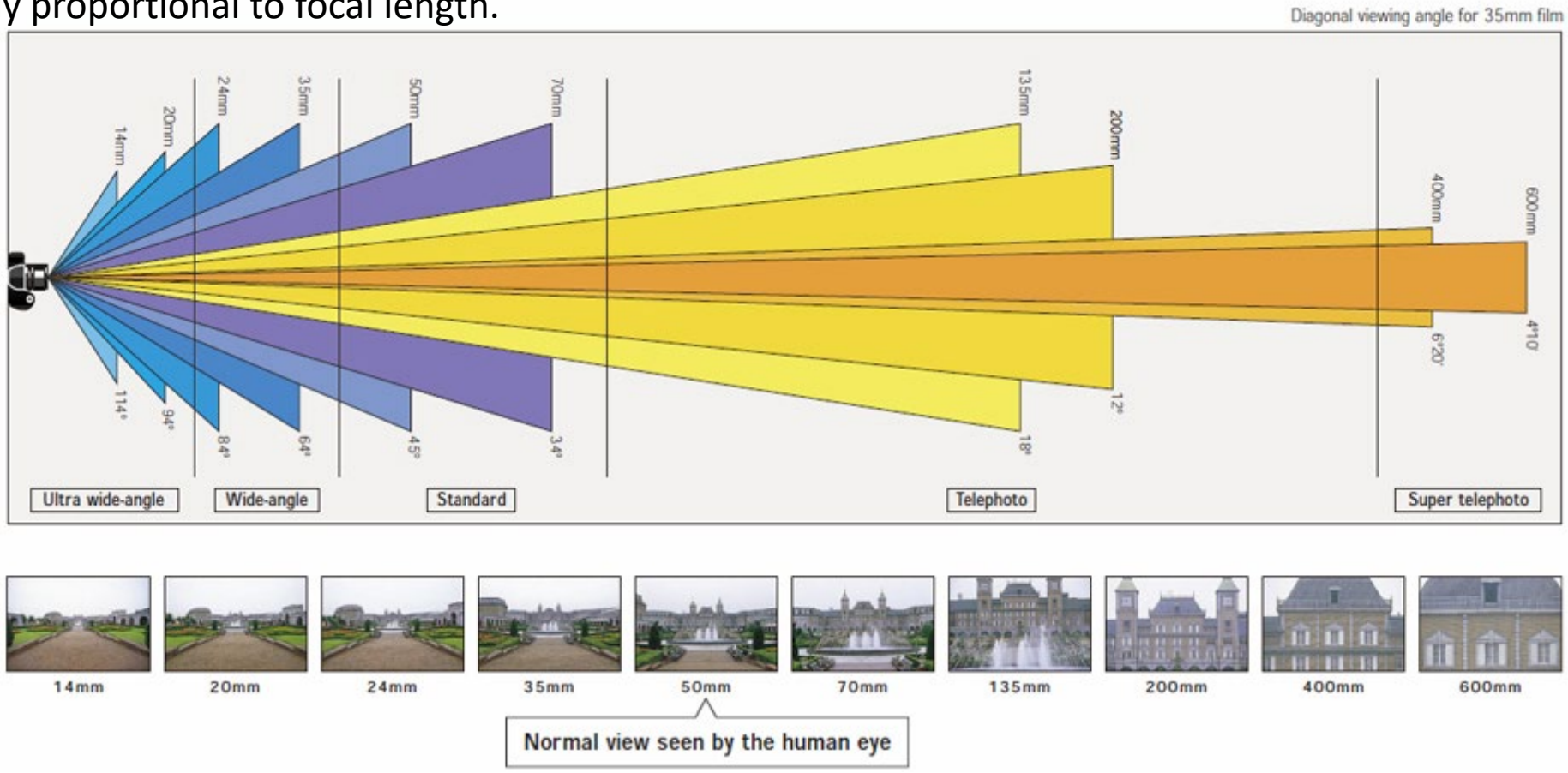
Aperture = f/4.0. DOF=2.2 cm

Aperture = f/22. DOF=12.4 cm

https://en.wikipedia.org

Field of View

- The cone of viewing directions of the camera
- Inversely proportional to focal length.

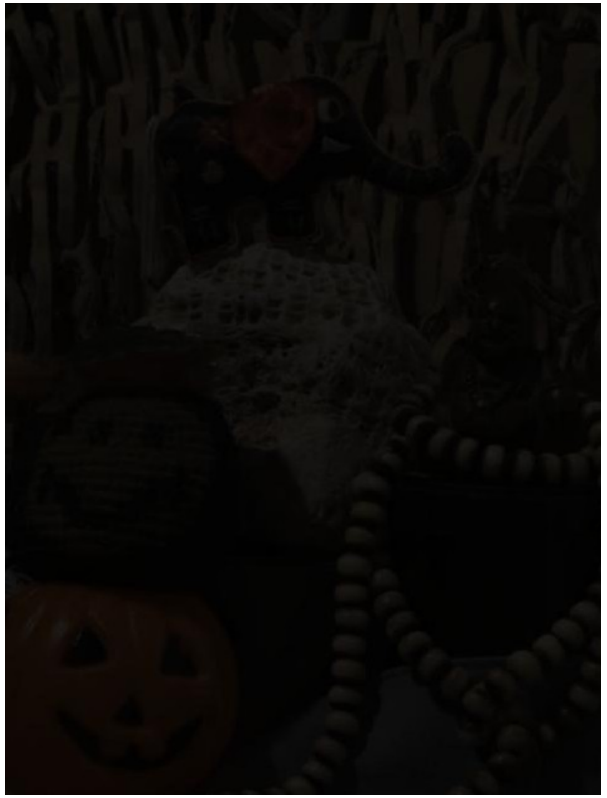


https://av.jpn.support.panasonic.com/

Factors affecting the image formation - Shutter speed

- Controls ***how long*** the film/sensor is ***exposed***, i.e. the amount of light reaching the sensor
- Pretty much ***linear effect*** on exposure
- Usually in ***fraction*** of a second:
  - 1/30, 1/60, 1/125, 1/250, 1/500
  - Get the pattern ?
- ***Faster*** shutter (e.g. 1/500th sec) = ***less*** light
- ***Slower*** shutter (e.g. 1/30th sec) = ***more*** light
- On a normal lens, normal humans can hand-hold down to 1/60
  - In general, the rule of thumb says that the limit is the inverse of focal length, e.g. 1/500 for a 500mm

Shutter Speed

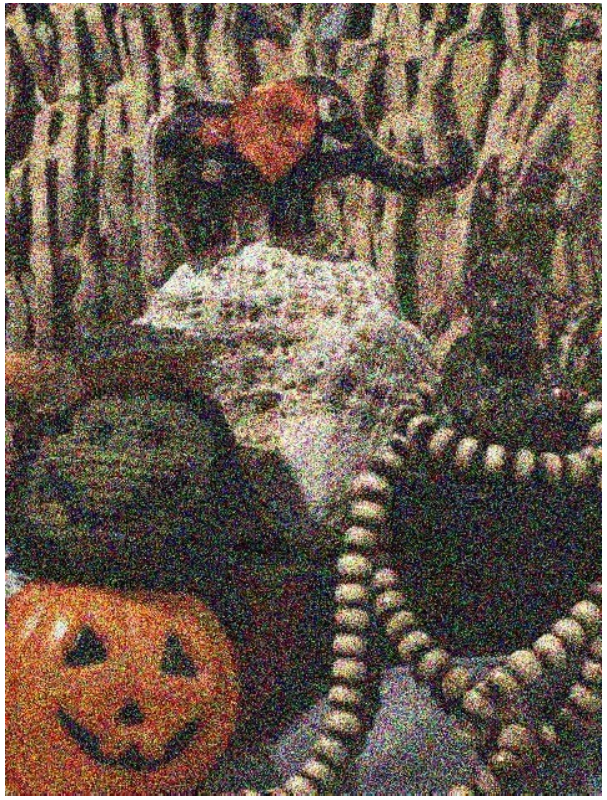


Short exposure- dark

medium exposure

long exposure- saturation

Shutter Speed



Short exposure after contrast adjustment- noise

medium exposure

long exposure- saturation

Main effect of slower shutter speed

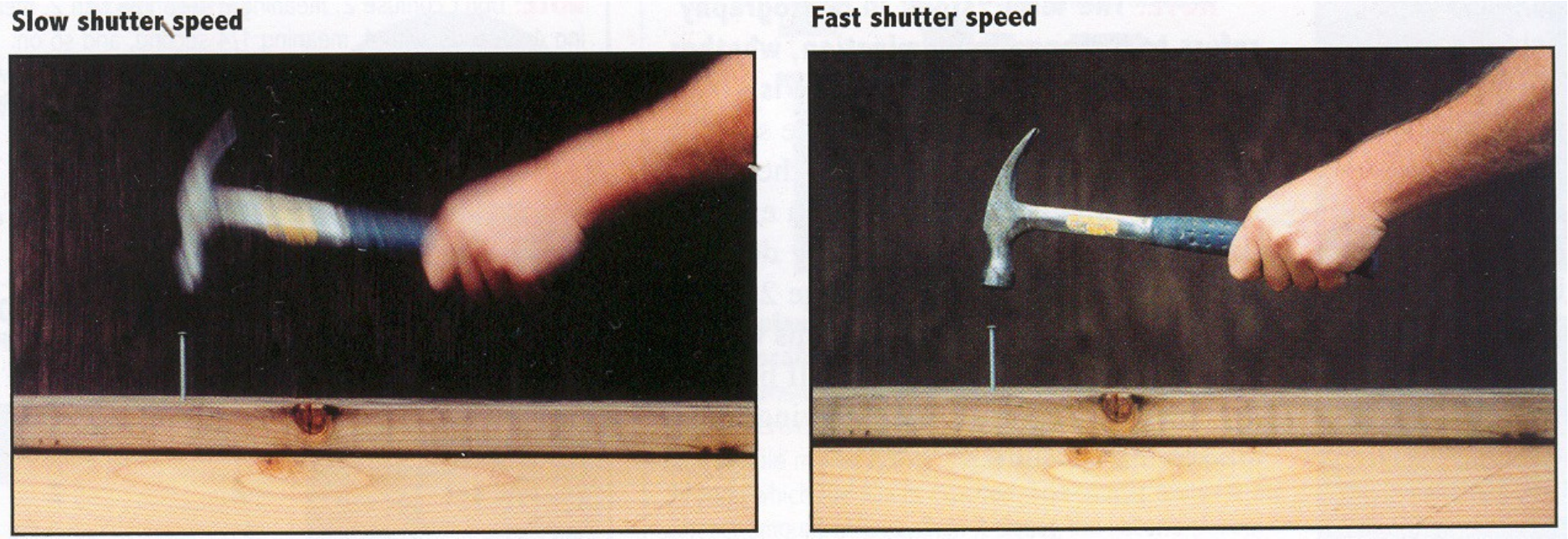
- For dynamic scenes, the shutter speed also determines the amount of ***motion blur*** in the resulting picture.
- Camera shake

Image taken with a tripod

Image taken with a hand held camera

Main effect of slower shutter speed

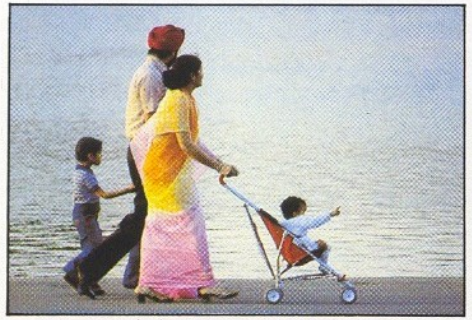
- For dynamic scenes, the shutter speed also determines the amount of *motion blur* in the resulting picture.
- Scene motion



From Photography, London et al.

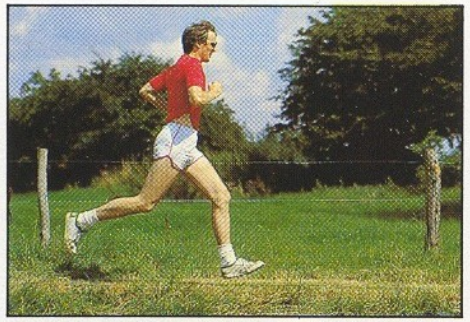
Effect of Shutter Speed

Walking People



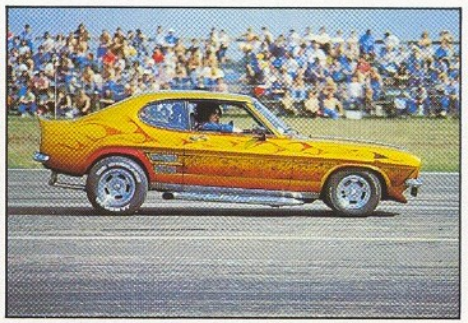
1/125

Running people



1/250

Car



1/500

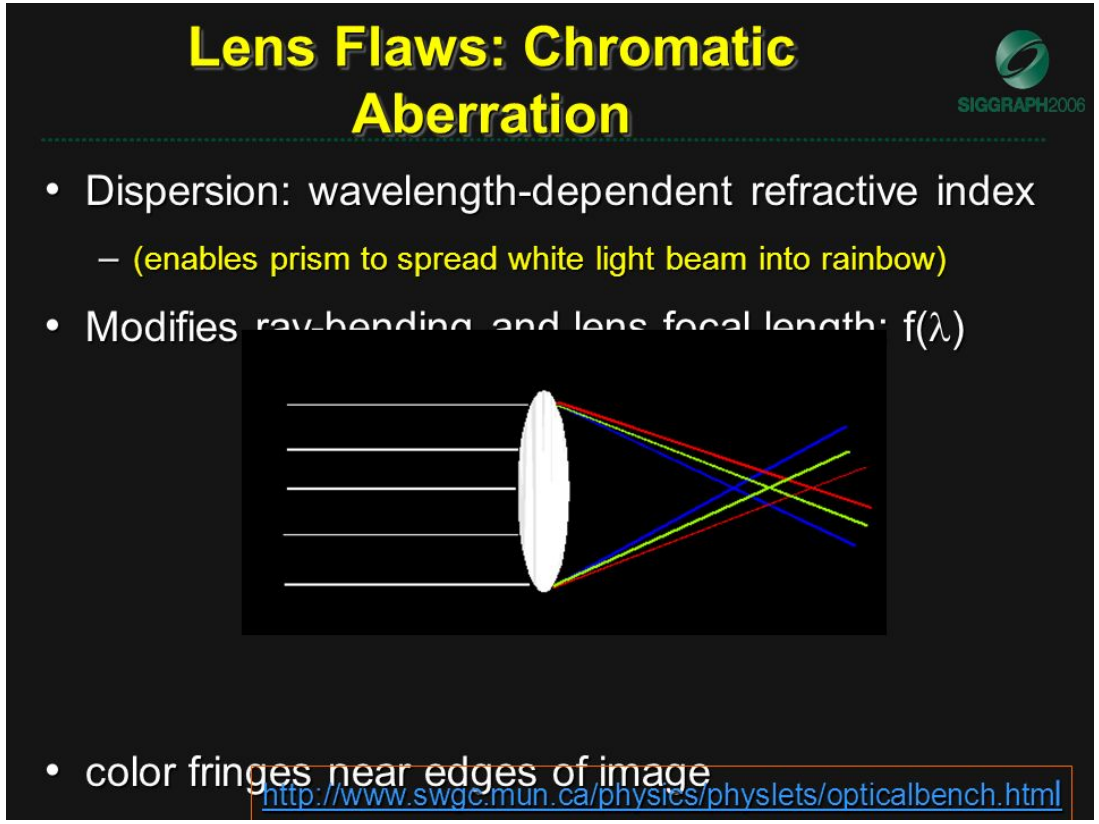
Fast train

1/1000

Lens Flaws

- In real life, lens has flaws

- Common lens flaws
  - Chromatic Aberration
  - Different refractive indices for different wavelengths
  - Fringing: Different colors not focus on the same point



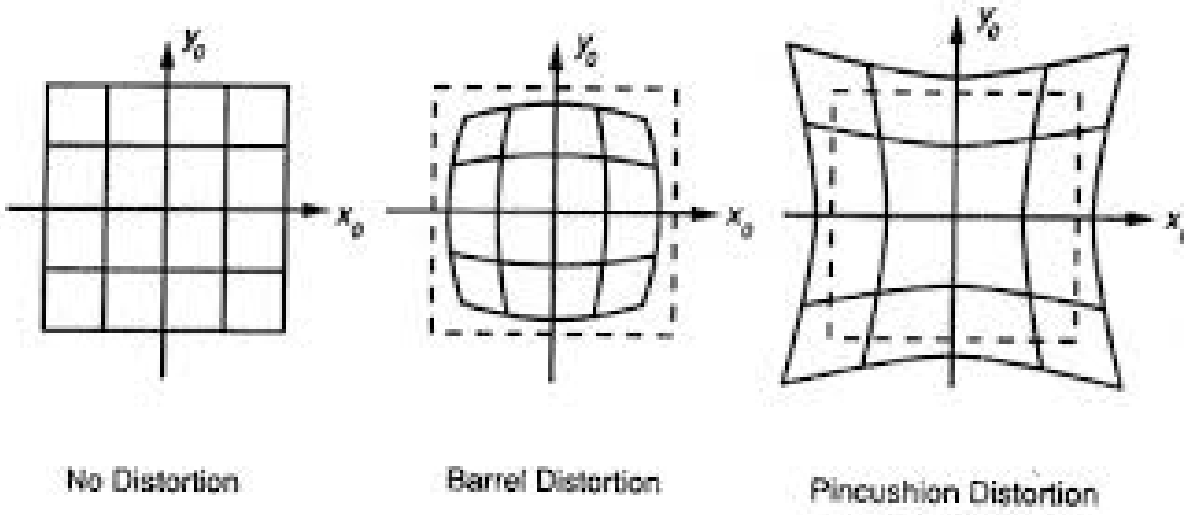
Lens Flaws – Chromatic Aberration

https://www.cambridgeincolour.com

www.photoeditingindia.com

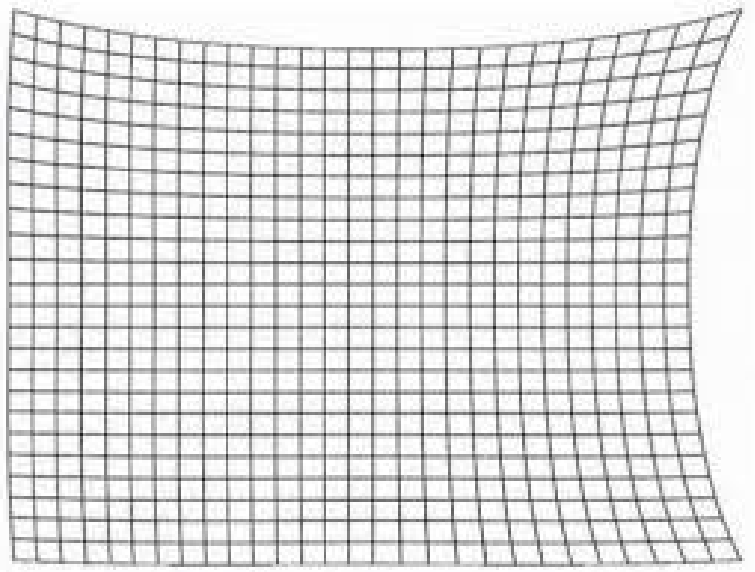
Lens Flaws – Radial Distortion

- Straight lines distorted more when is it further away from the centre of the image.



Lens Flaw – Tangential Distortion

- Lens is not parallel to the image plane
- Image plane is skewed 歪斜的



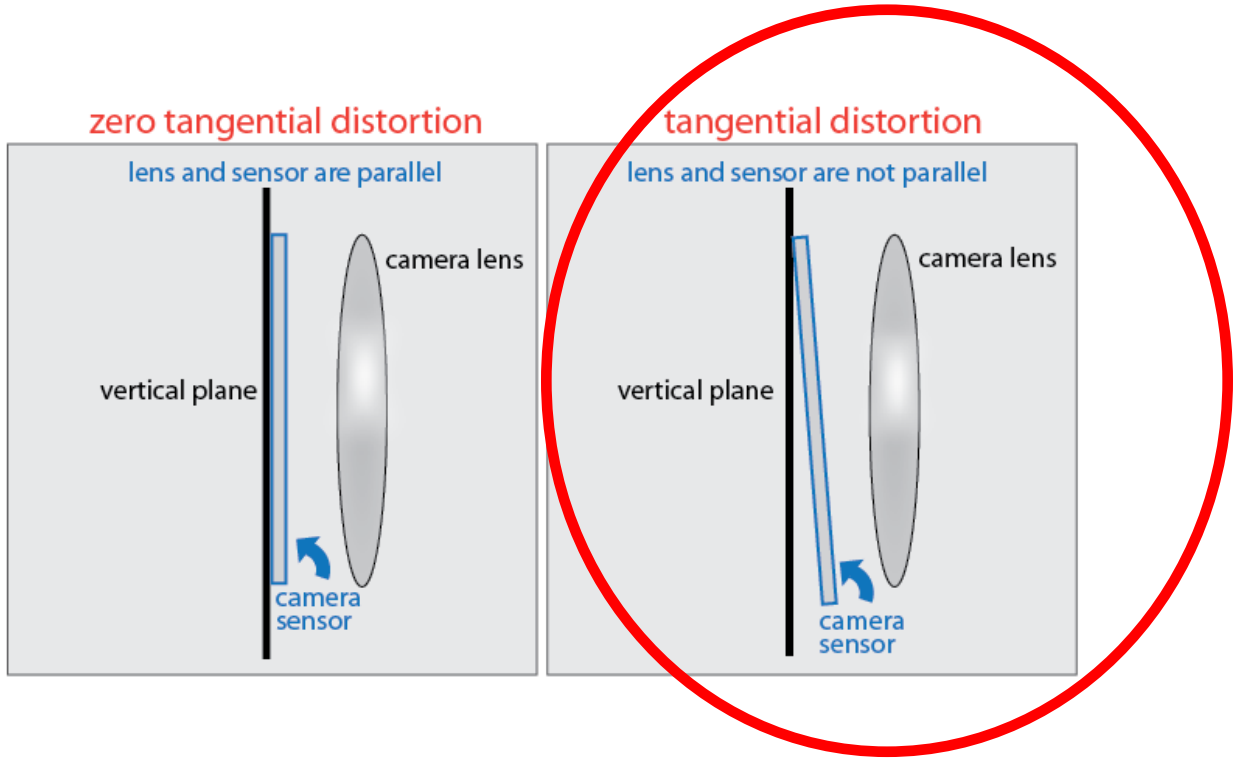


Image Source: https://www.mathworks.com

Accidental Camera

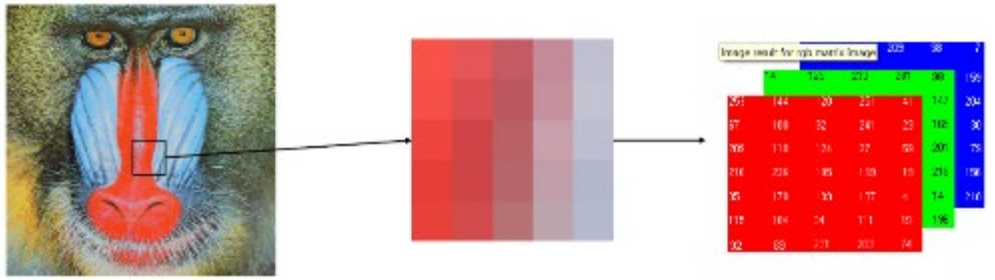
https://www.reddit.com

https://www.reddit.com

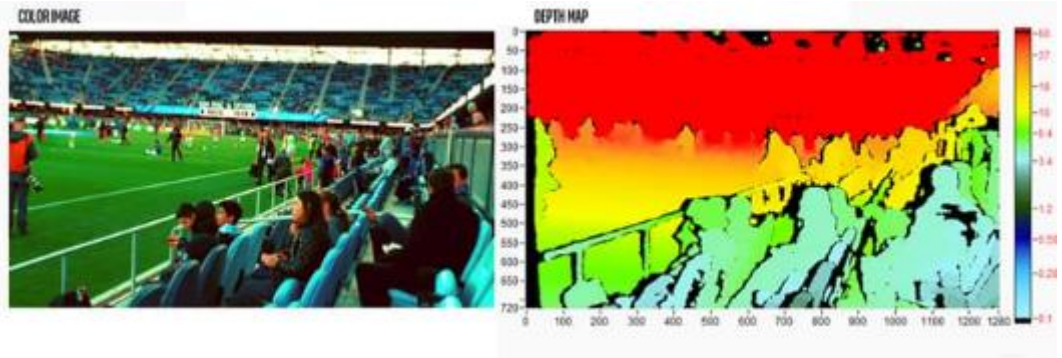
Window turned into a pinhole

Other Types of Cameras

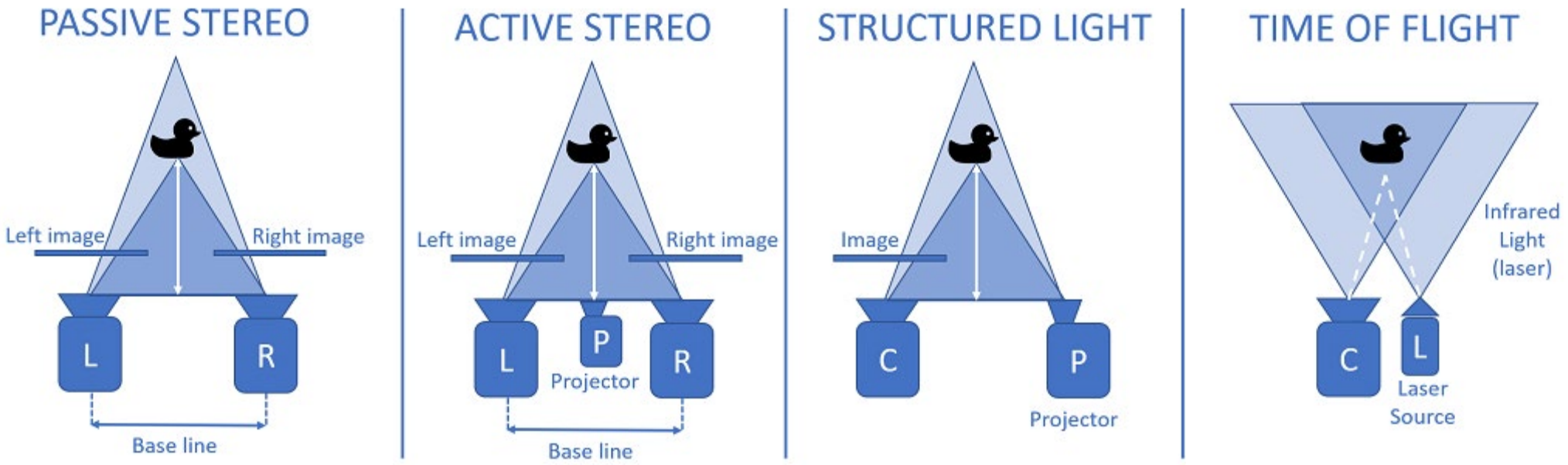
- Traditional cameras give only 2D array (2D information) of the captured object. No 3D information can be extracted.



- A depth camera has pixels with distance information (depth) with them. Some camera both RGB information and depth system, called RGBD camera



Types of depth cameras

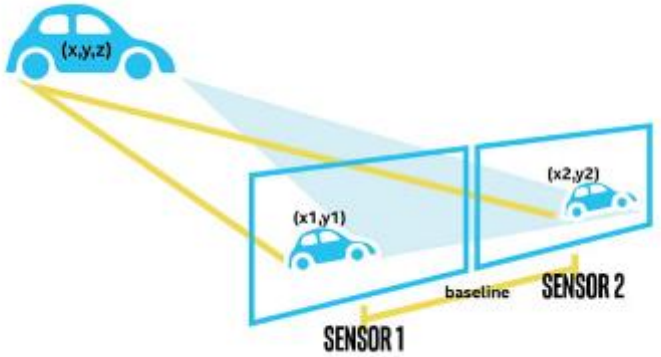


https://community.arm.com/

Stereo Vision – Passive Stereo

- Formed by a ***pair of cameras*** with know distance apart (***baseline***). Depth is calculated by the correspondence between images through a triangulation process.

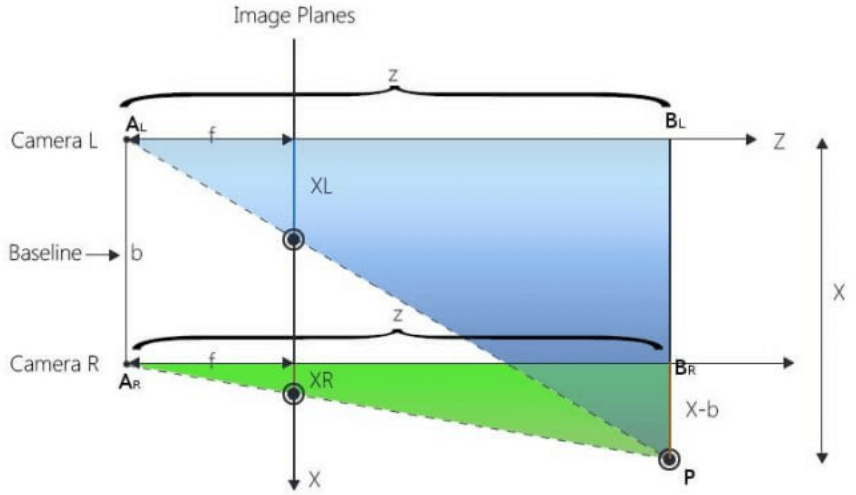
```
cannot reconstr wall
```



https://www.intelrealsense.com/

- Quality of result depends on the density of ***visually distinguishable feature points*** (Harris corner detection) or ***SIFT/ORB*** features

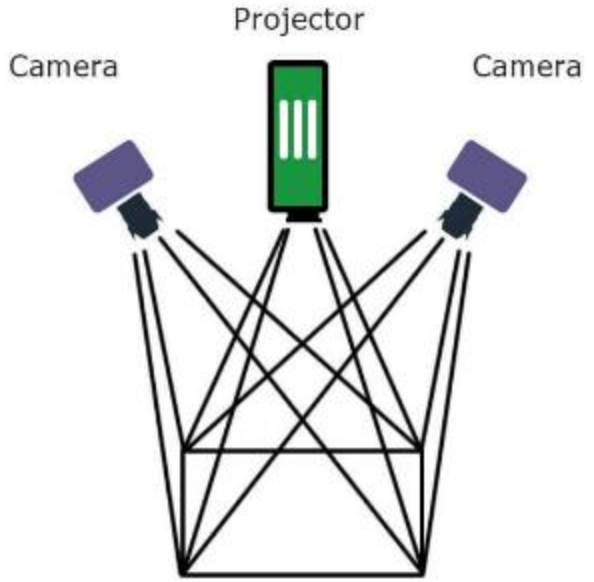
- Any source of natural or artificial ***texture*** will help in significantly ***improve*** the accuracy



https://www.e-consystems.com/

Stereo Vision – Active Stereo

- Similar to passive stereo by using two cameras
- Addition ***projector*** is used to provide ***extra texture*** for the triangulation process.
- Active stereo is useful in region with ***low light*** or ***texture***.
- Same as passive stereo under sunlight / long range



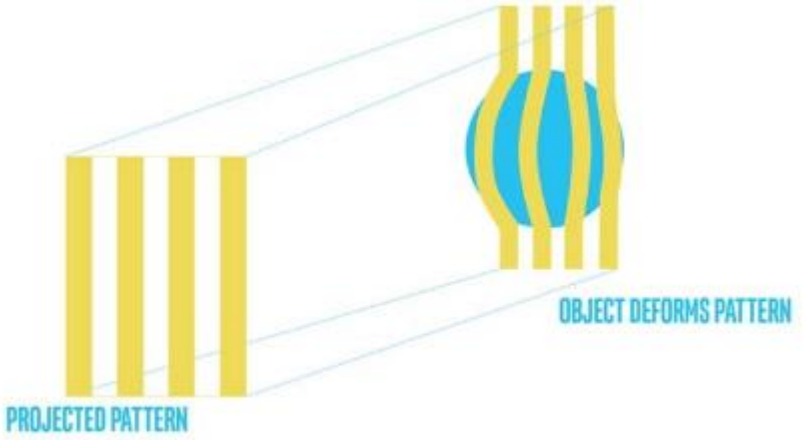
https://www.e-consystems.com/

Stereo Vision – Structure lighting

- Structured light relies on ***projecting light*** (usually infrared light) from emitter onto the scene

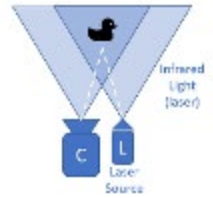
- Since the ***pattern is known***, ***depth*** informed is reconstructed by the deformed pattern.

- Best for ***indoor*** and relatively ***short*** range.



Structure Lighting Pros and Cons

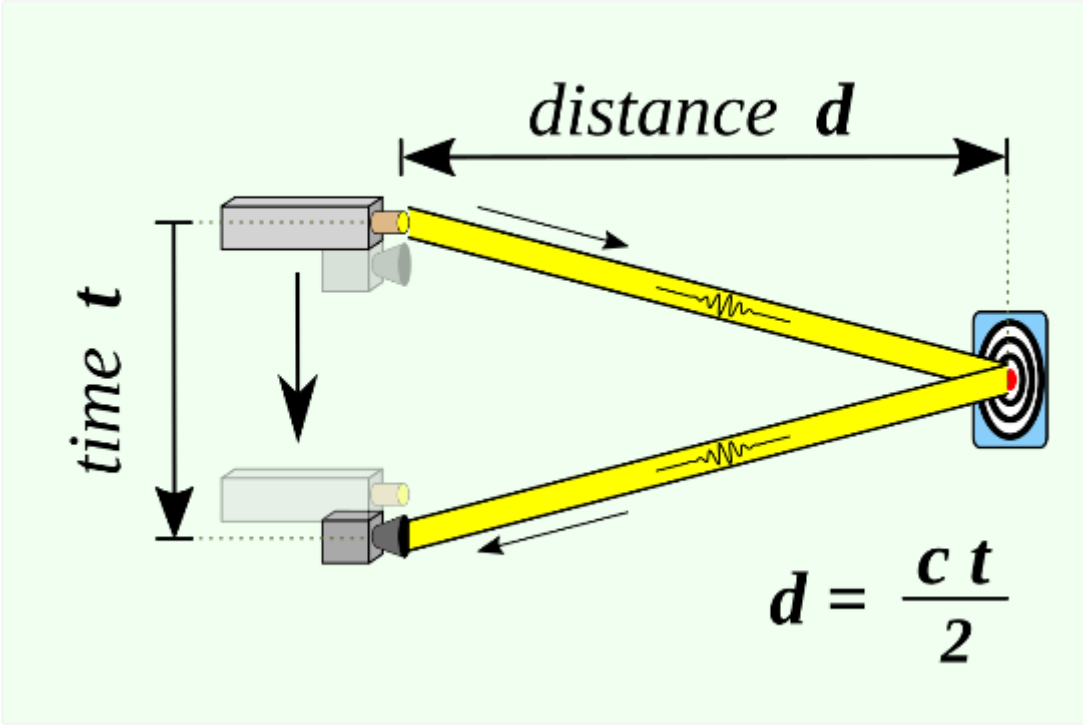
- Pros
  - Usually very easy to get ***good results*** using commercial scanners
  - ***High-precision*** model
- Cons
  - ***Cannot*** apply to ***outdoors*** environment or ***brightly*** lit environment
  - Object size is ***limited***
  - Scanner can be ***expensive***



Time of Flight Camera (Laser scanner)

- TOF camera consist of one camera and one ***laser pulse generator***.
- The direct ***time-of-flight*** required for a single ***laser pulse*** to ***leave*** the camera and ***reflect back*** onto the focal plane array
- Pros
  - Compact system
  - Fast
- Cons
  - Several TOF camera may cause interference

Indoor Only



Today's Agenda

- Image Formation and camera fundamentals
  - What is digital image
  - Image Formation
  - What is pin-hole camera
  - Camera Models: Lenses, Depth of Field, View Angle
  - Accidental Camera
  - Other Types of Cameras
  - Geometric transformation
- Image warping
  - Interpolation
  - Resampling
  - Applications

Review of Matrix Operations

- A matrix $\mathrm{A} \in \mathbb{R}^{m \times n}$ is an array of number
- The array has m rows and n columns

- $\mathrm{A} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m4} & \cdots & a_{mn} \end{bmatrix}$

- A is a square matrix when $m = n$

Matrix Operation Review

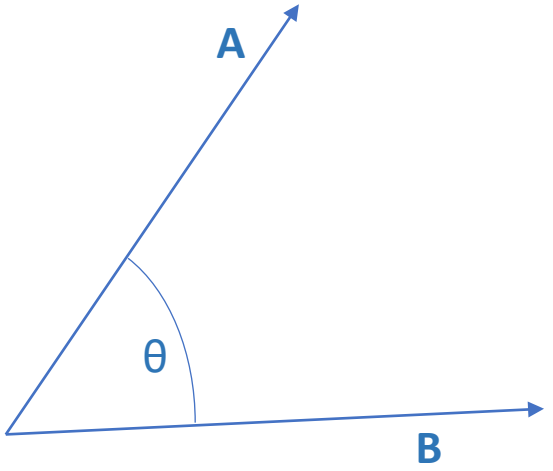
- Matrix Operations
  - Addition
  - Scaling
  - Dot product
  - Multiplication
  - Transpose
  - Determinant
  - Inverse

Matrix Operations Review

- Addition
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix} + e = \begin{bmatrix} a+e & b+e \\ c+e & d+e \end{bmatrix}$
- Scaling
  - $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times k = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix}$

Matrix Operations Review

- Inner product/dot product
  - $A^T \cdot B$ where $A \in \mathbb{R}^{n \times 1}$ & $B \in \mathbb{R}^{n \times 1}$
  - $A^T \cdot B = [a_1 \quad \dots \quad a_n] \cdot \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \sum_{i=1}^{n} a_i \cdot b_i$
  - $A^T \cdot B = |A||B|cos\theta$ where θ is angle between A and B



Matrix Operations Review

- Matrix Multiplication
- Two matrices $A \in \mathbb{R}^{3\times 2}$ $B \in \mathbb{R}^{2\times 3}$

- $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$ $B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{33} \end{bmatrix}$

- $A \times B = \begin{bmatrix} a_{11} \times b_{11} + a_{12} \times b_{21} & \dots & a_{11} \times b_{13} + a_{12} \times b_{23} \\ \vdots & \vdots & \vdots \\ a_{31} \times b_{11} + a_{32} \times b_{21} & \dots & a_{31} \times b_{13} + a_{32} \times b_{23} \end{bmatrix}$

Matrix Operations Review

- ***Transpose*** – flip along the diagonal
- $A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$ $\quad A^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$
- $(AB)^T = B^T A^T$

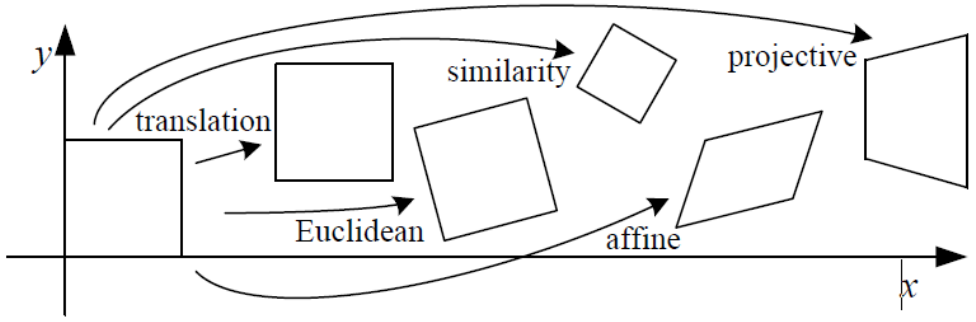
- ***Determinant***
- Det(A) is a scalar
- $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
- $\det(A) = ad - bc$

Trace for nxn matrix

- $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$
- $tr(A) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{12}$

Transformations

- Matrices can be used for transform vectors: $x' = Tx$
  - Scaling
  - Rotation
  - Shear
  - Mirror
  - Translation



Transformation - Scaling

- Scaling
- $x' = Sx$
- *For uniform scale*
  - $x' = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
  - $x' = \begin{bmatrix} sx \\ sy \end{bmatrix}$
- For non-uniform scale
  - $x' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
  - $x' = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$

Uniform scale: Degree of freedom = 1

Transformation - Rotation

- We want to rotate $(x', y') \rightarrow (x", y")$
- We have
  - $x' = Rcos\emptyset$
  - $y' = Rsin\emptyset$
- And we have
  - $x" = Rcos(\theta + \emptyset)$
  - $y" = Rsin(\theta + \emptyset)$

Expanding

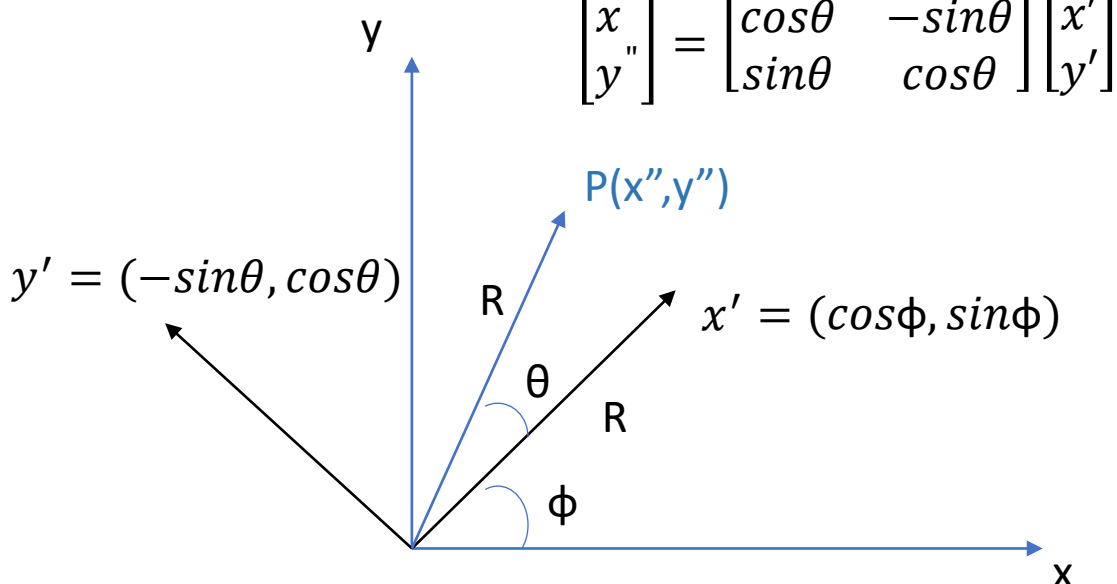
$$x" = Rcos\theta cos\emptyset - Rsin\theta sin\emptyset$$
$$y" = Rsin\theta cos\emptyset + Rcos\theta sin\emptyset$$

Then

$$x" = x'cos\theta - y'sin\theta$$
$$y" = x'sin\theta + y'cos\theta$$

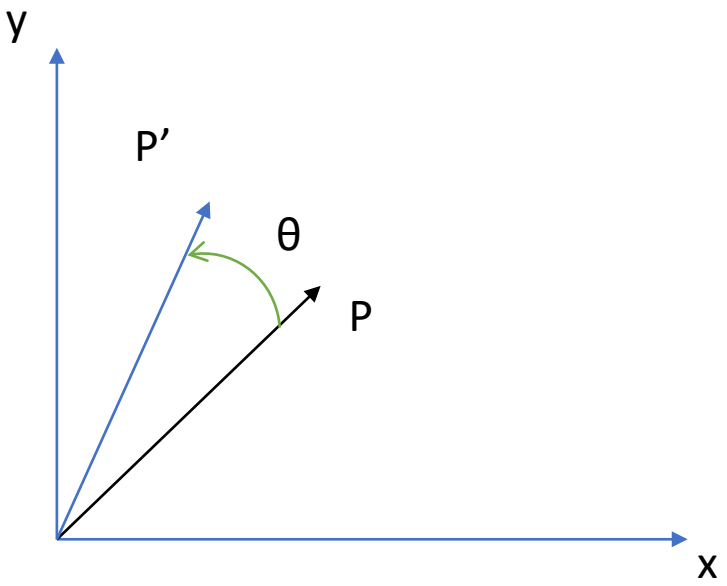
In matrix form

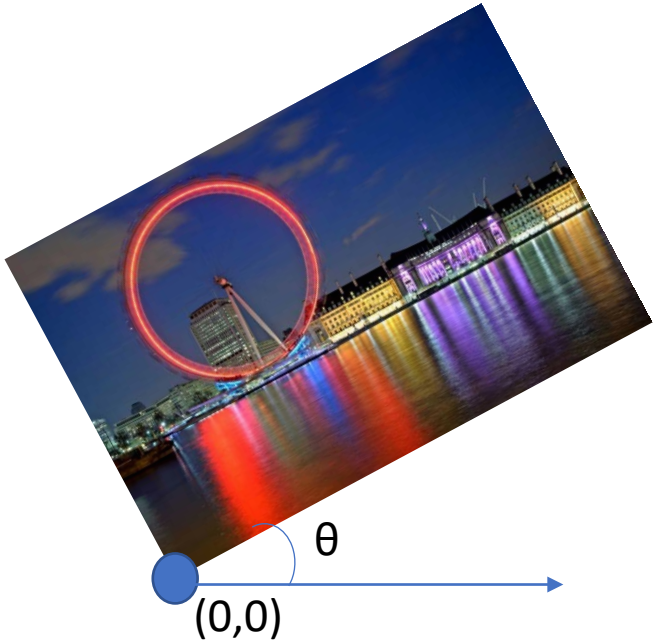
$$\begin{bmatrix} x" \\ y" \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$



Transformation - Rotation

- $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
- $R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$

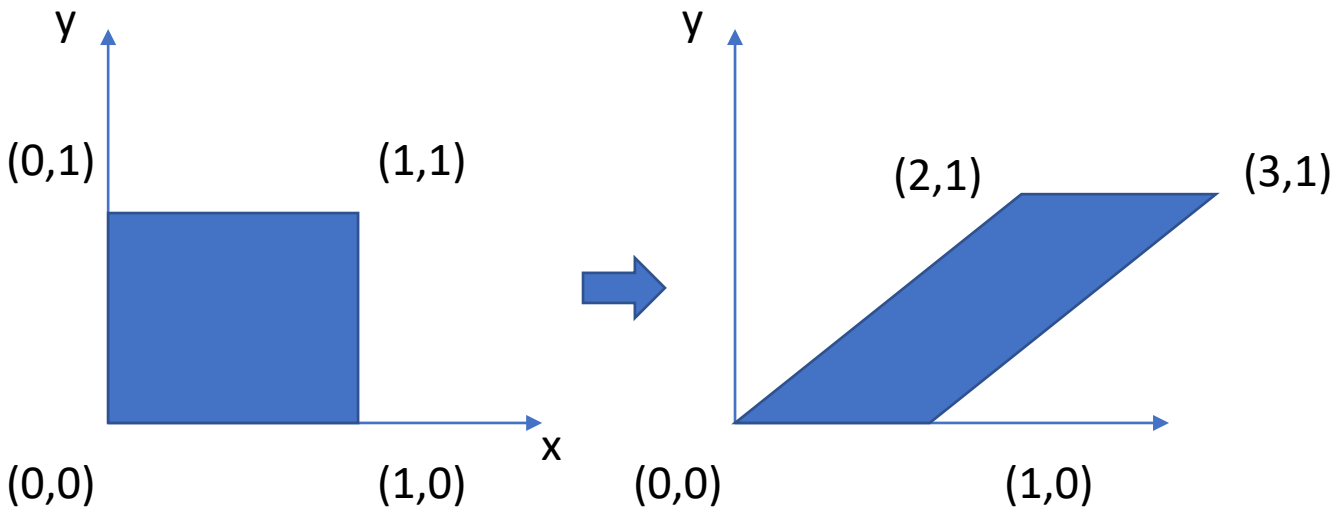




Degree of freedom = 1

Transformation - Shear

- 2D Shear
- $x' = x + shx \times y$
- $y' = shy \times x + y$
- $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & shx \\ shy & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$



shx

Degree of freedom = 2

Transformations - Mirror

- Other types of transformation represented by 2x2 matrix
- 2D Mirror about Y axis
  - $x' = -x$
  - $y' = y$
  - $T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
- 2D mirror access line y=x
  - $x' = y$
  - $y' = x$
  - $T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Multiple Transformation

- Multiple Transformation can be performed
  - $p' = R_2 R_1 S p$
- The operation is carried out from right to left
- The result is same as we multiply the matrices first
  - $p' = (R_2 R_1 S) p$

How to handle translation?

- In general, 2D transformation can be represented as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- This 2x2 matrix is sufficient for scale, rotation, shear and mirror

- How about translation?
  - $x' = x + t_x$
  - $y' = y + t_y$

- No! we cannot add a constant!

Homogenous Coordinates

- Definition: ***Homogenous coordinate*** $\mathbf{x}$ of a geometric entity x are *invariant* with respect to *multiplication* by a scalar $\lambda \neq 0$

$$\mathbf{x} = \lambda\mathbf{x} \qquad \text{for} \qquad \lambda \neq 0$$

- Example
-

$$\mathbf{x} = \lambda\mathbf{x} \qquad\qquad x \neq \lambda x$$

Homogenous

Euclidian

Homogenous Coordinates

- Notation for ***homogenous*** and ***inhomogeneous*** vector and matrices

	2D	3D	transformations
inhomogeneous	x	X	R
homogeneous	l, x	**A**, **L**, **X**	H

Homogenous Coordinates

- ***Homogenous Coordinates*** add ***one*** extra dimension to the vector.
- i.e. use *n+1* dimensional vector to represents *n* dimensions point

- Transforming ***Euclidean coordinates*** to ***Homogenous Coordinates***

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

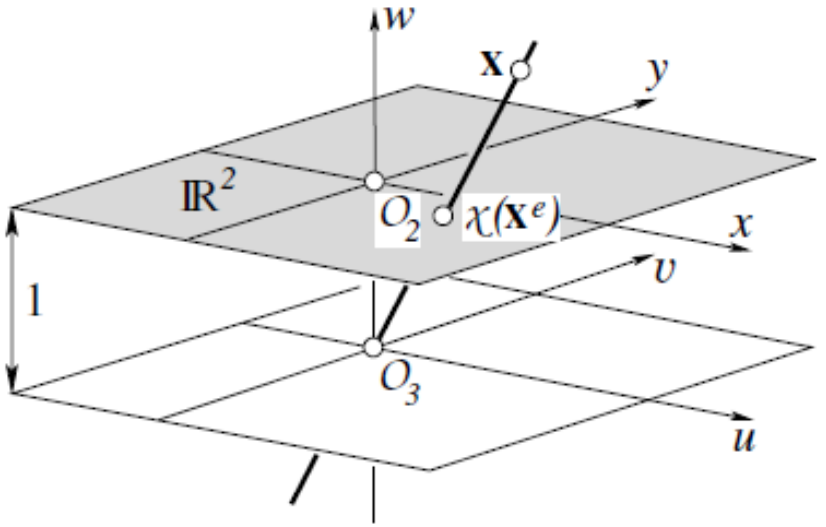
$$\mathbf{x} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} = w \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix} = \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \text{ in Euclidean coordinates}$$

Homogenous coordinates

- *Homogeneous Coordinates* of a point x in the plane in $\mathbb{R}^2$ is a 3-dimensional vector

x: $\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$ where

$$u^2 + v^2 + w^2 \neq 0$$



- In Euclidian coordinates

x: $\boldsymbol{x} = \begin{bmatrix} u/w \\ v/w \end{bmatrix}$ where $w \neq 0$

Example

- All point χ of Euclidian plane $\mathbb{R}^2$

$$\boldsymbol{x} = [x, y]^T$$

- Represented in homogenous coordinates

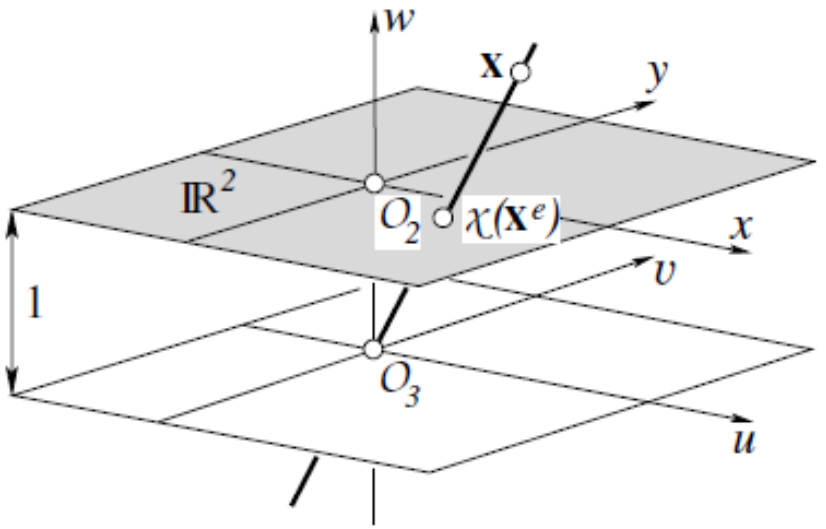
$$\mathbf{x} = [x, y, 1]^T$$

How about point at infinity?

$$\mathrm{x} = [x, y, 0]^T$$

except

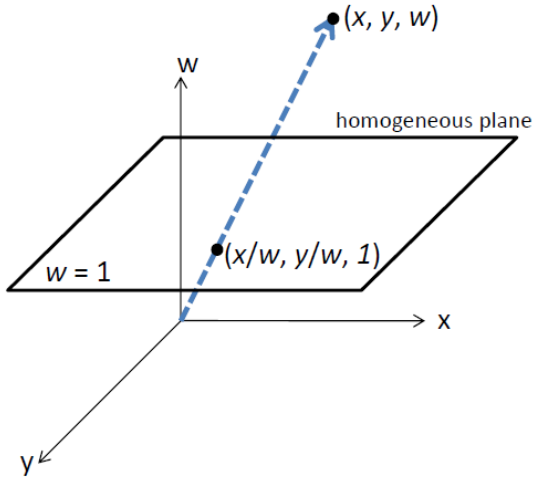
$$\mathrm{x} = [0,0,1]^T$$



Homogenous coordination

- The solution is to stick a “1” at the end of the vector

$$\begin{bmatrix} x \\ y \end{bmatrix} \Longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



- Converting from homogenous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$$

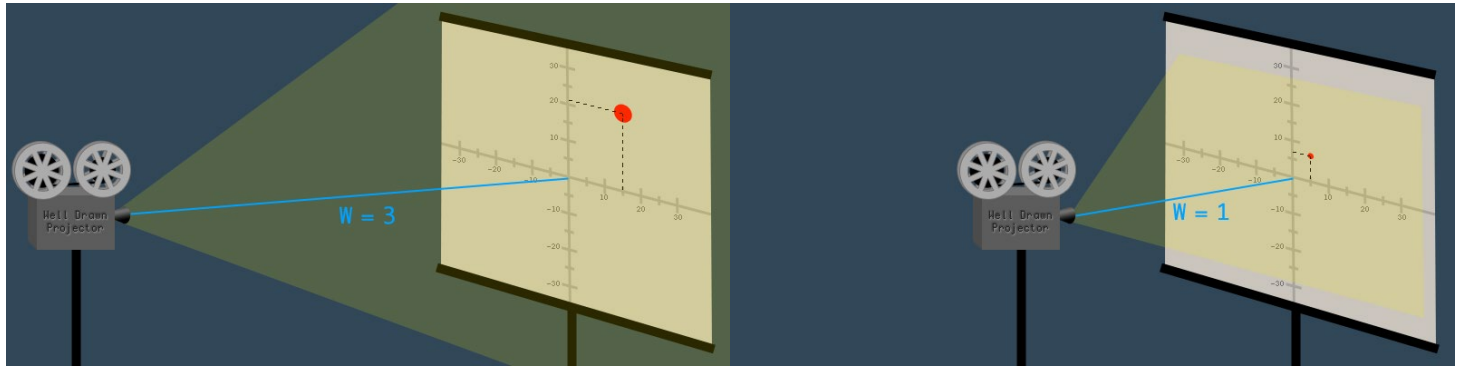
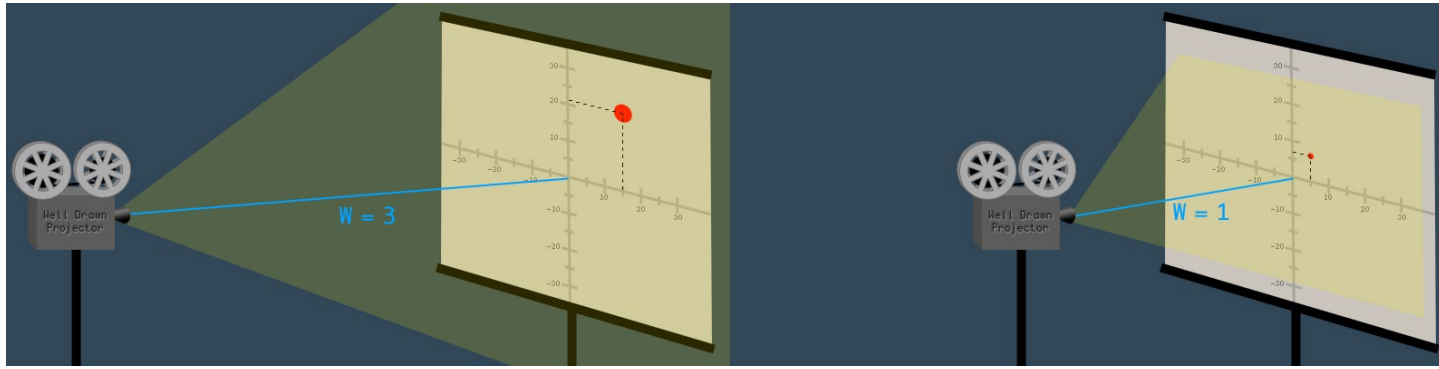


Image Source: https://www.tomdalling.com

In homogenous coordinate

- *Homogenous to Euclidean*

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$$



- Therefore

$$\begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}, \begin{bmatrix} 4 \\ 8 \\ 2 \end{bmatrix} \text{and} \begin{bmatrix} 8 \\ 16 \\ 4 \end{bmatrix} \text{representing the same point}$$

$\mathbf{x} = \boldsymbol{\lambda}\mathbf{x}$ for $\boldsymbol{\lambda} \neq 0$

Advantages of Homogenous coordinates

- Allow to represent ***entities at infinity*** which occurs frequently, e.g. when dealing with vanishing points.

- Allows us to easily represent ***straight line-preserving transformations***, not only ***translations***, ***rotations*** or ***affine transformations*** but also ***projective transformation*** such as mapping from 3D object space to 2D image space in a pinhole camera

- They simplify ***concatenation*** and ***inversion*** of straight line-preserving transformations since all transformation are represented as a matrix vector product.

Translation with Homogenous Coordinate

- In general, 2D transformation represented below ***CANNOT*** cater for translation. ***(We cannot add a constant)***

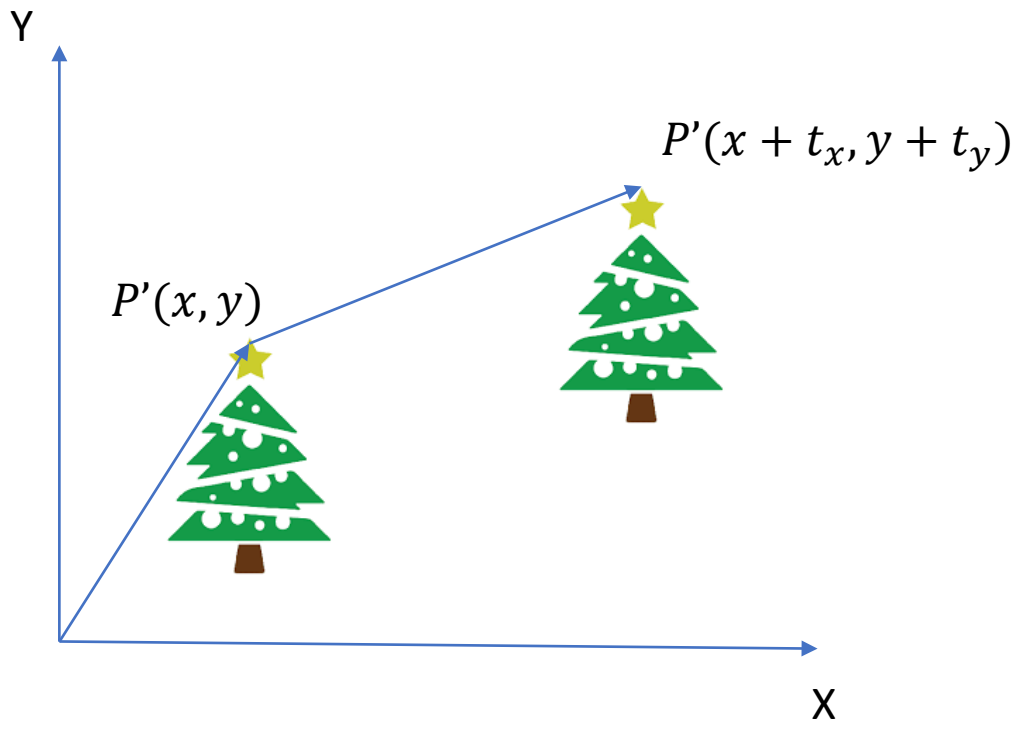
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Using ***Homogenous Coordinate*** system, the transformation matrix now becomes

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

- Now we can do ***rotation***, ***scale***, ***shear***, ***mirror*** and ***translation***

Translation in Homogenous Coordinates (HS)



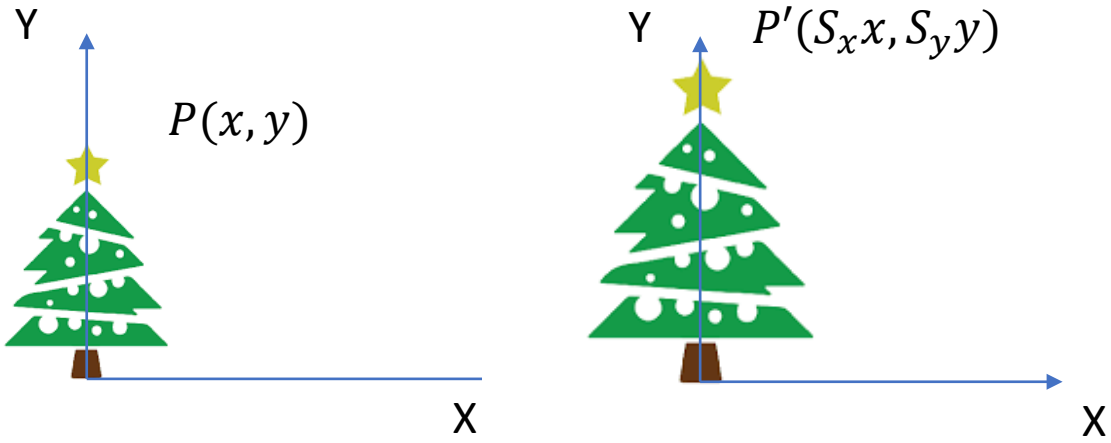
- Homogenous Coordinates
  - $P = (x, y) \rightarrow (x, y, 1)$
  - $t = (t_x, ty) \rightarrow (t_x, t_y, 1)$

- $T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

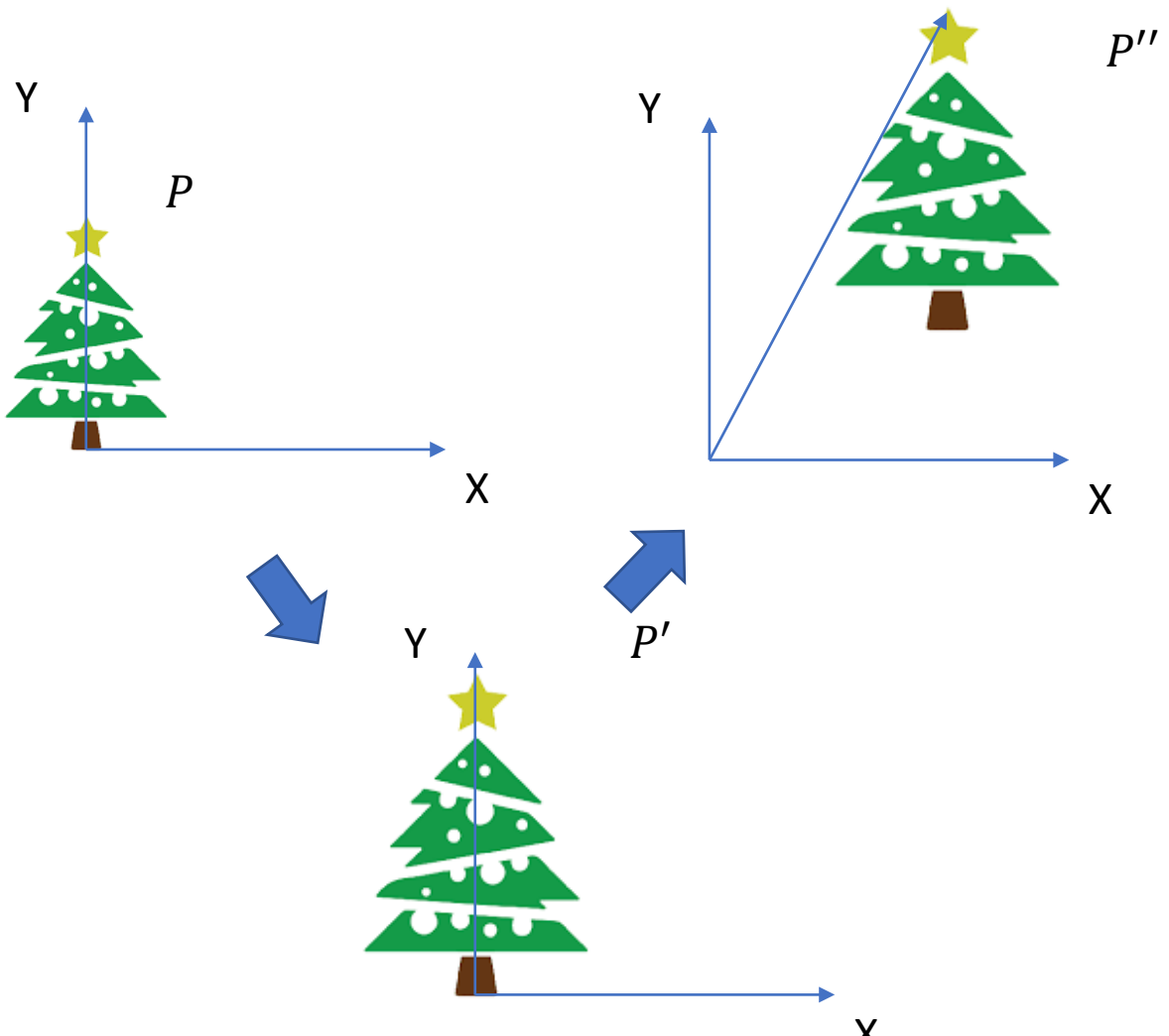
- $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$

Scaling in HS

- Homogenous Coordinates
  - $P = (x, y) \rightarrow (x, y, 1)$
- $S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S_x x \\ S_y y \\ 1 \end{bmatrix}$



Scaling and Translation in HS



- $P' = S \cdot P$
- $P'' = T \cdot P' = T \cdot S \cdot P$
- $P'' = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $P'' = \begin{bmatrix} s_x & 0 & t_x \\ 0 & S_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

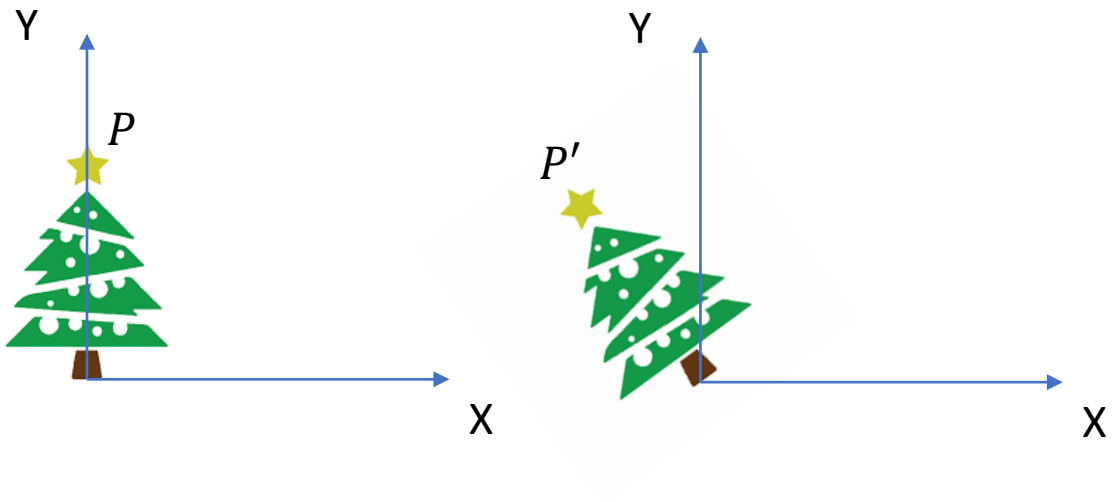
Translating & Scaling = Scaling and Translating?

- Translating & Scaling
- $P' = T \cdot P$
- $P'' = S \cdot P' = S \cdot T \cdot P$
- $P'' = \begin{bmatrix} s_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $P'' = \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & S_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $P'' = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$

$\neq$

- Scaling & Translating
- $P' = S \cdot P$
- $P'' = T \cdot P' = T \cdot S \cdot P$
- $P'' = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $P'' = \begin{bmatrix} s_x & 0 & t_x \\ 0 & S_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $P'' = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$

Rotation in HS



- Homogenous Coordinates
  - $P = (x, y) \rightarrow (x, y, 1)$
- $R = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta x - sin\theta y \\ sin\theta x + cos\theta y \\ 1 \end{bmatrix}$

Composite Affine Transformation

- One can integrate Translation, rotation and scaling together, say H

$$\mathrm{H} = \mathrm{RST} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- The composite transformation can be represented by an affine transformation

$$\mathrm{H} = \begin{bmatrix} s_x cos\theta & -s_y sin\theta & s_x t_x cos\theta - s_y t_y sin\theta \\ s_x sin\theta & s_y cos\theta & s_y t_y cos\theta + s_x t_x sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

- The ***inverse*** of the transformation can be calculated as

$$\mathrm{H}^{-1} = (\mathrm{RST})^{-1} = \mathrm{T}^{-1}\mathrm{S}^{-1}\mathrm{R}^{-1}$$

Affine Transformations

- Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 2D in-plane rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Shear

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In summary - Affine Transformations

- Combination of
  - Linear transformation
  - Translation

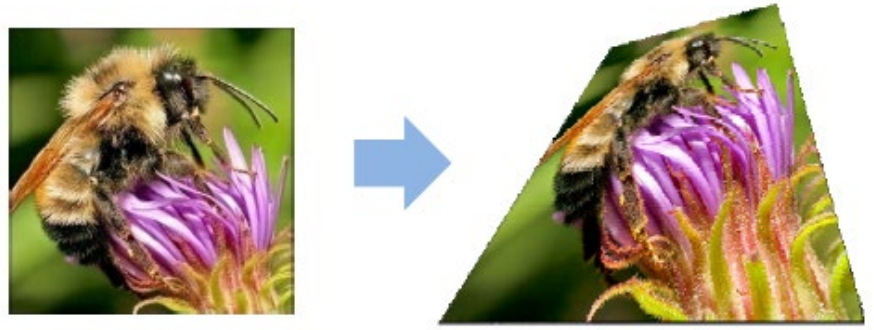
Affine transform $\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$

- Properties
  - Origin ***does not necessarily*** map to origin
  - Lines ***remain*** lines
  - Parallel lines ***remain*** parallel
  - Ratio ***preserved***

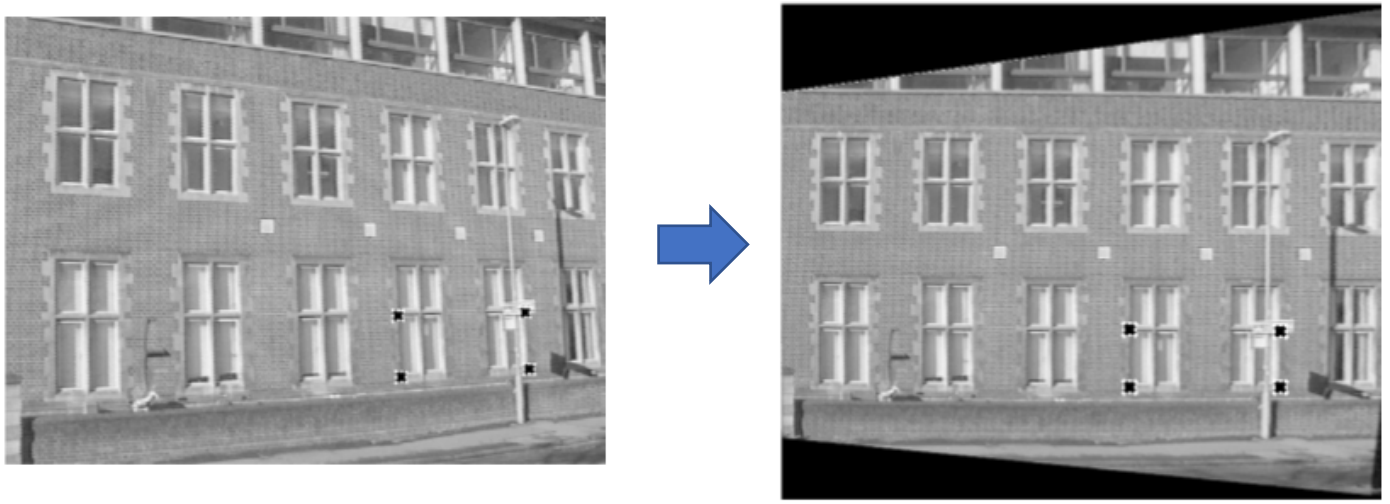
Homography

- Affine Transformation
  - $\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$

Effect of this row?



- Homography/planar perspective map
  - $H = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$

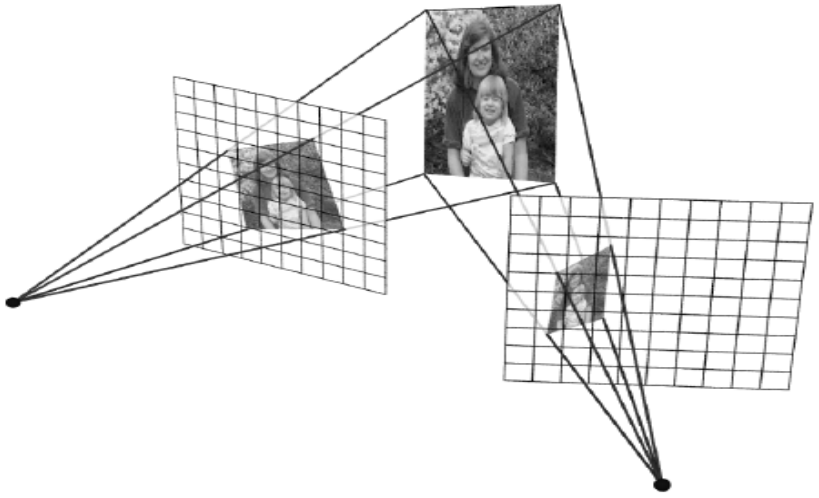


Projective Transformations

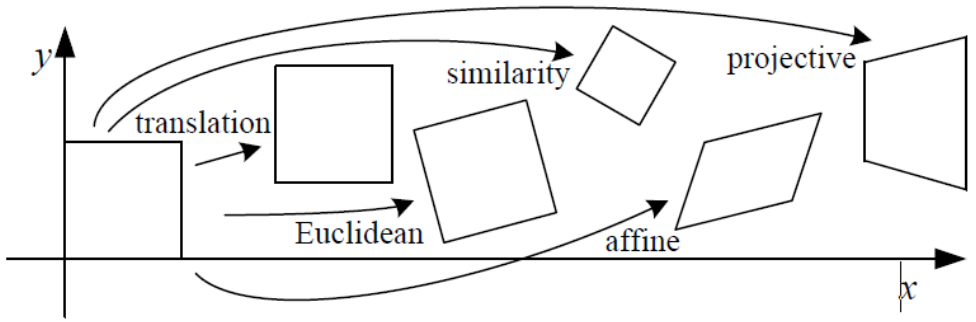
- Projective Transformations
  - Affine Transformation
  - Projective warps

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Properties of projective transformations
  - Origin ***does not necessarily*** map to origin
  - Lines ***map*** to lines
  - Parallel lines ***do not necessarily remain*** parallel
  - Ratio are ***not*** preserved



In Summary - 2D Transformations



Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times3}$	8	straight lines	

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

$$M = \begin{bmatrix} cos\theta & -sin\theta & t_x \\ sin\theta & cos\theta & t_y \end{bmatrix}$$

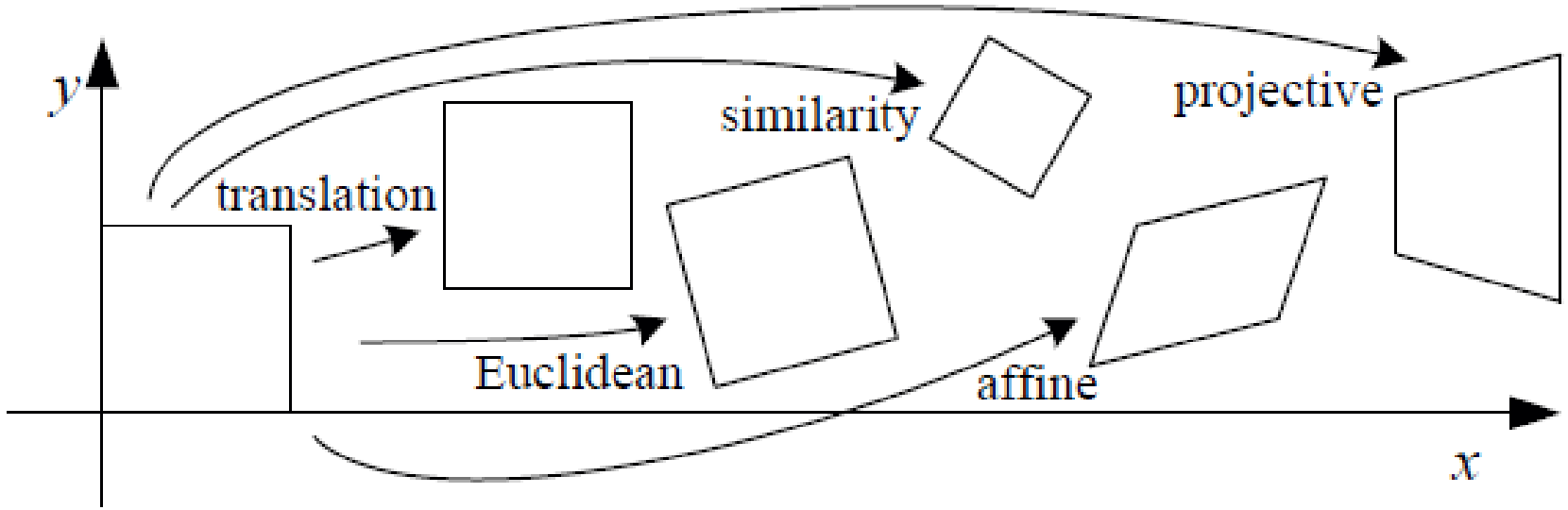
$$M = \begin{bmatrix} scos\theta & -ssin\theta & t_x \\ ssin\theta & scos\theta & t_y \end{bmatrix}$$

$$M = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$$

$$M = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Basic Set of 2-D Transformation

- Using 2D transformation, we can perform image warping



Source: R. Szeliski

Geometric Transformation (Warping) Between Images

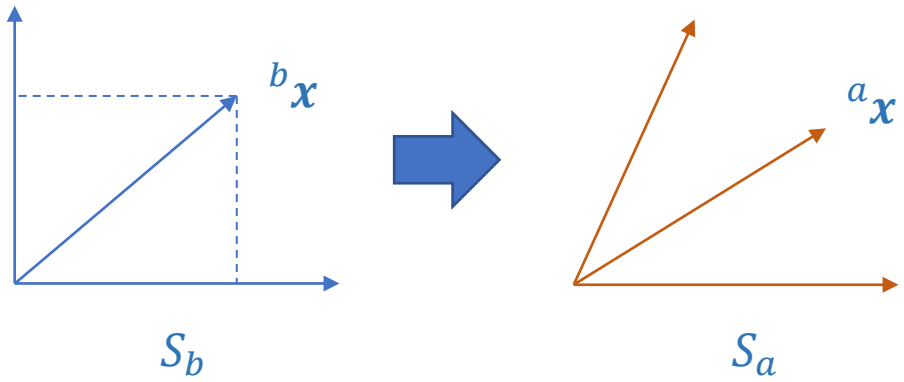
- Every image has its coordinate system
  - Image $a(x, y)$ has coordinate system S_a
  - Image $b(x, y)$ has coordinate system S_b

- Coordinates:
  - ${}^{a}\boldsymbol{x} = \begin{bmatrix} {}^{a}x \\ {}^{b}y \end{bmatrix}$ and ${}^{b}\boldsymbol{x} = \begin{bmatrix} {}^{b}x \\ {}^{b}y \end{bmatrix}$

Geometric Transformation (Warping) Between Images

- Goal: Transform from S_b to S_a

- This transformation from $[{}^{b}x,\ {}^{b}y] \rightarrow [{}^{a}x,\ {}^{a}y]$
- This transformation can be written as
  - ${}^{a}\boldsymbol{x} = {}^{a}W_b\ {}^{b}\boldsymbol{x},$
- Also
  - ${}^{b}\boldsymbol{x} = {}^{b}W_a\ {}^{a}\boldsymbol{x},$



Geometric Transformations (Warping) Between Images

- The expression

$$ {}^{a}\boldsymbol{x} = {}^{a}W_{b} \; {}^{b}\boldsymbol{x} $$

- We can write

$$ \begin{bmatrix} {}^{a}\boldsymbol{x} \\ {}^{a}\boldsymbol{y} \end{bmatrix} = \begin{bmatrix} W_x({}^{b}x, {}^{b}y) \\ W_y({}^{b}x, {}^{b}y) \end{bmatrix} $$

Simple Warping

- ***Translations***: (shift by a vector (d_x , d_y)

$$W_x: {}^a\boldsymbol{x} = {}^b x + d_x$$

$$W_y: {}^a\boldsymbol{y} = {}^b y + d_y$$

- ***Scaling***: (Strength or contract along x or y axis by a factor s_x, s_y

$$W_x: {}^a\boldsymbol{x} = s_x \, {}^b x$$

$$W_y: {}^a\boldsymbol{y} = s_x \, {}^b y$$

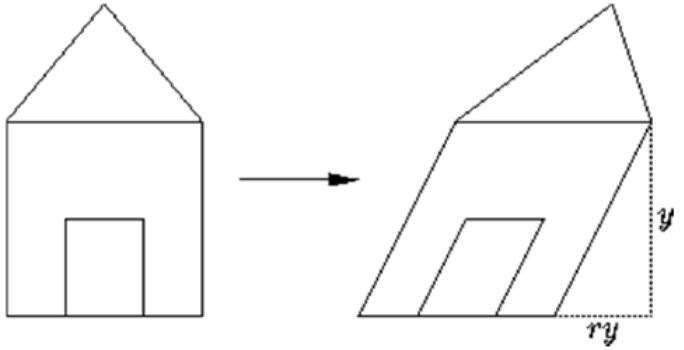
Photo Credit: Emmanuel Agu, WPI

Simple Warping (Cont'd)

- ***Shearing*** : along x and y axis by factor b_x, b_y

$$W_x: \ {}^{a}\boldsymbol{x} = {}^{b}x + b_x \, {}^{b}y$$

$$W_y: \ {}^{a}\boldsymbol{y} = {}^{b}y + b_y \, {}^{b}x$$



- ***Rotation*** : rotate image by angle Θ

$$W_x: \ {}^{a}\boldsymbol{x} = {}^{b}x \cdot \cos\Theta - {}^{b}y \cdot \sin\Theta$$

$$W_y: \ {}^{a}\boldsymbol{y} = {}^{b}x \cdot \sin\Theta + {}^{b}y \cdot \cos\Theta$$

Photo Credit: Emmanuel Agu, WPI

Issue in image warping!

- *For example, the pixel in [1 1] is transformed as follow*

$$\begin{bmatrix} {}^{a}\boldsymbol{x} \\ {}^{a}\boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$= \begin{bmatrix} 5.5 \\ 5.5 \end{bmatrix}$$

- After transformation, the pixel is *not integer* anymore

- Problem!

Resampling

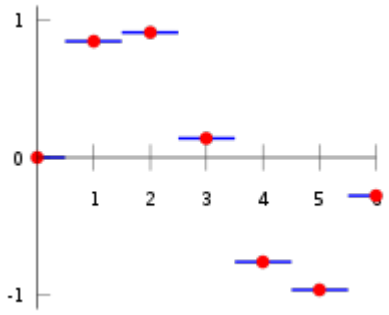
- The transformation gives ***non-integer coordinates***
- To assign intensity value after transformation, we need to do ***interpolation***
- This process for discretization and quantization is called ***resampling***.

Methods of Interpolation

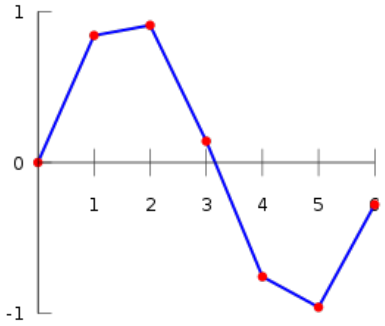
Three common method of interpolation:

- ***Nearest neighbour*** – simple and fastest
- ***Bilinear*** – Uses point from bounding rectangle – Better quality
- ***Bicubic*** - Use the 16 neighbours and cubic curve is fitted for the interpolation – Best Quality

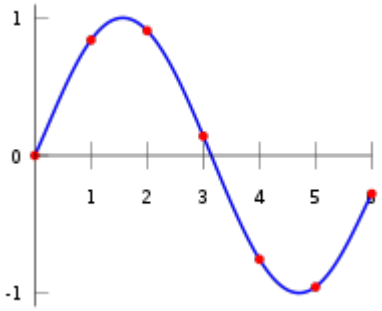
Interpolations in 1-D



Nearest Neighbour



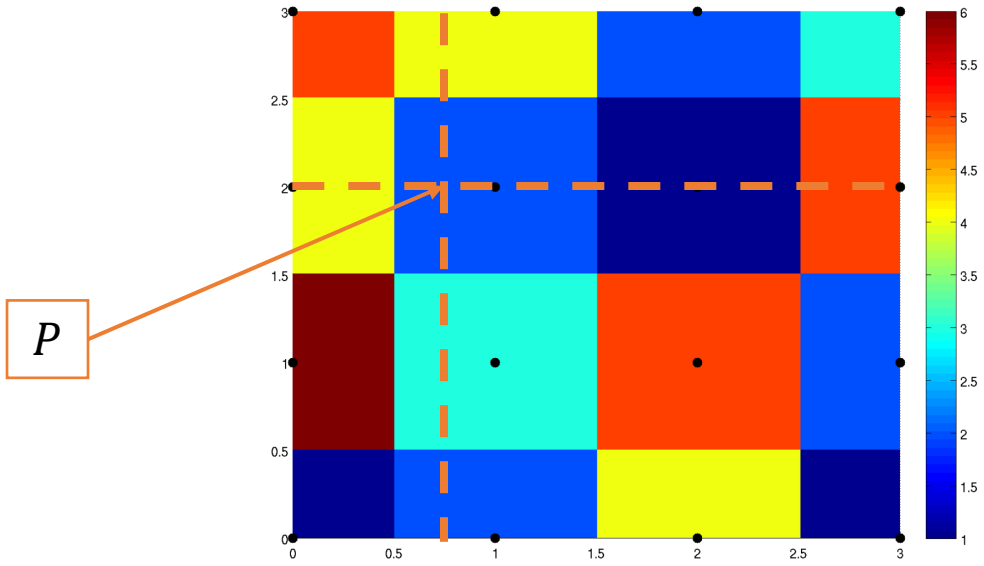
Linear interpolation



Cubic interpolation

Nearest Neighbor Interpolation

- Choose the same value as the closet pixel

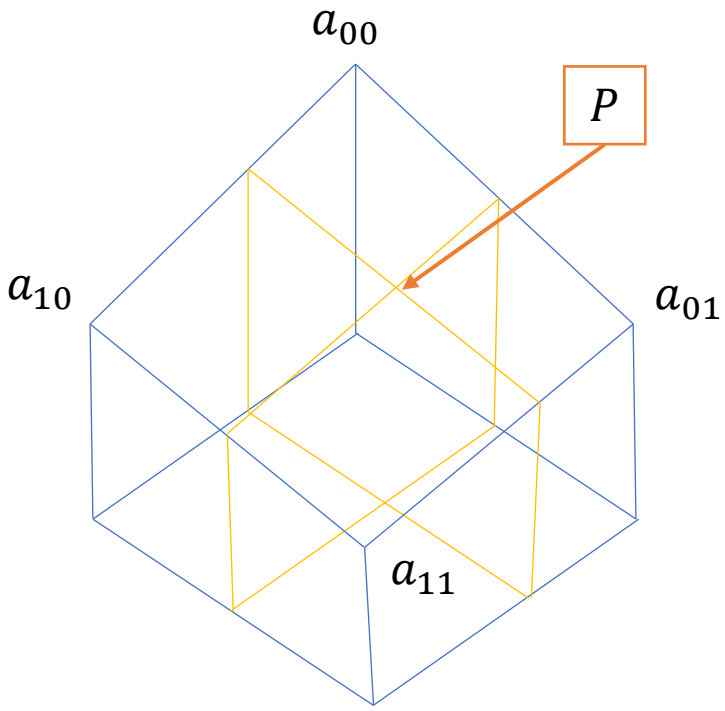


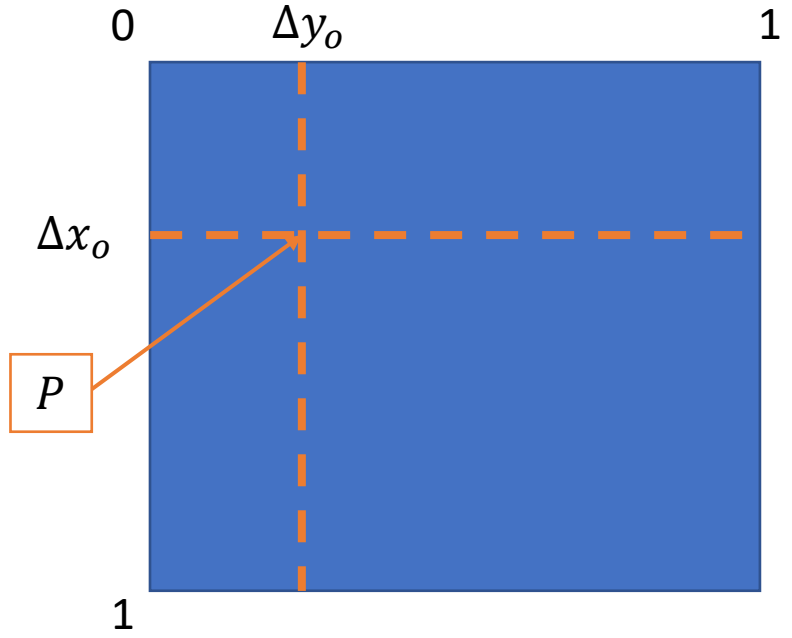
- Can we do better?

Image Courtesy: Wikipedia.com

Bilinear Interpolation

- Linear interpolation in X-direction and Y-Direction





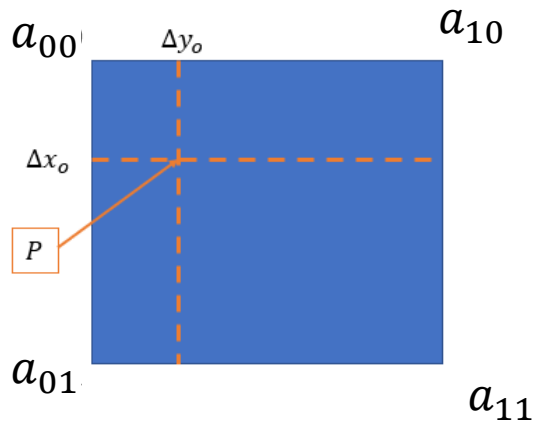
Bilinear Interpolation

- Linear interpolation in X-direction and Y-Direction

$$P = a_{00}(1-\Delta x)(1-\Delta y) + a_{01}(1-\Delta x)(\Delta y) + a_{10}(\Delta x)(1-\Delta y) + a_{11}(\Delta x)(\Delta y)$$

$$P = a_{00} + \Delta x\underbrace{(a_{10}-a_{00})}_{C_{10}} + \Delta y\underbrace{(a_{01}-a_{00})}_{C_{01}} + \Delta x\Delta y\underbrace{(a_{00}-a_{01}-}_{C_{11}}$$

$$P = \sum_{i\leq 1}\sum_{j\leq 1} C_{ij}\Delta x^i \Delta y^j$$



For more information about deviation of bilinear interpolation, you may refer to https://en.wikipedia.org/wiki/Bilinear_interpolation

Bilinear Interpolation

- Weighted average of the neighboring intensity value

$$P = \sum_{i \le 1} \sum_{j \le 1} C_{ij} \Delta x^i \Delta y^j$$

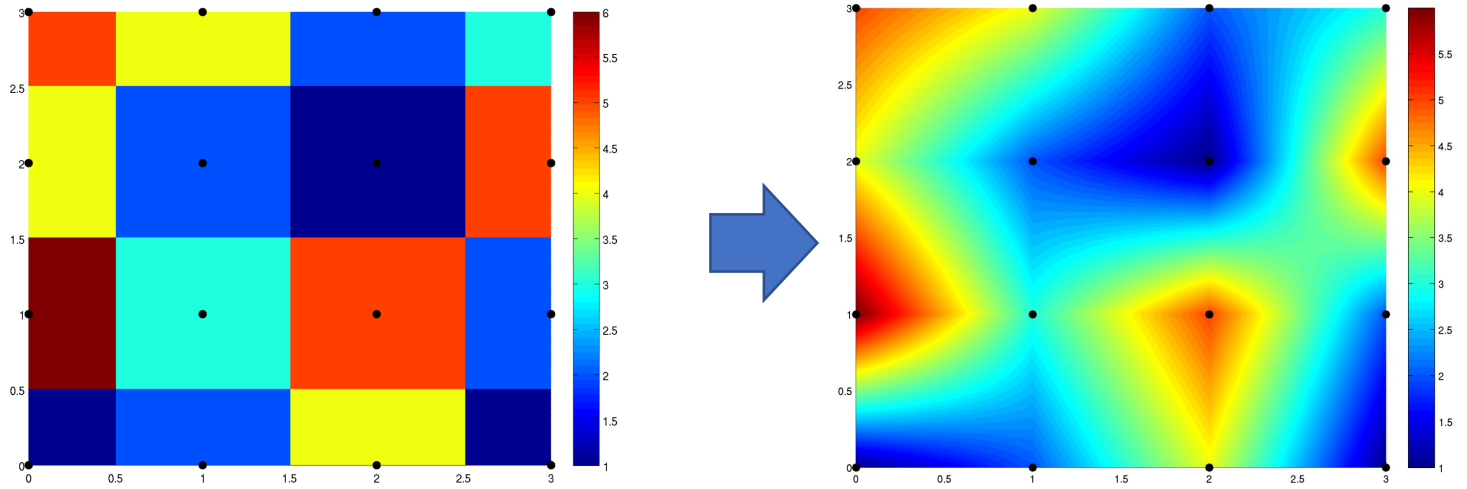
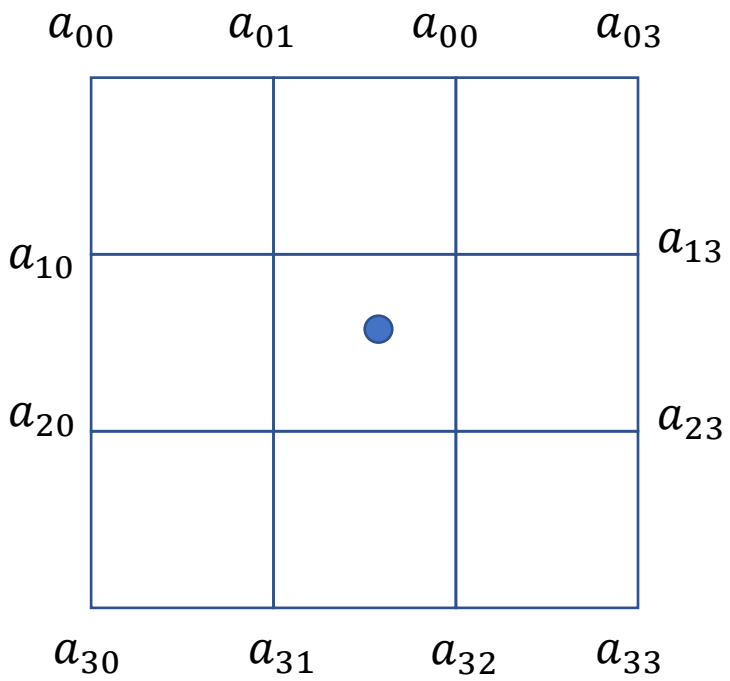


Image Courtesy: Wikipedia.com

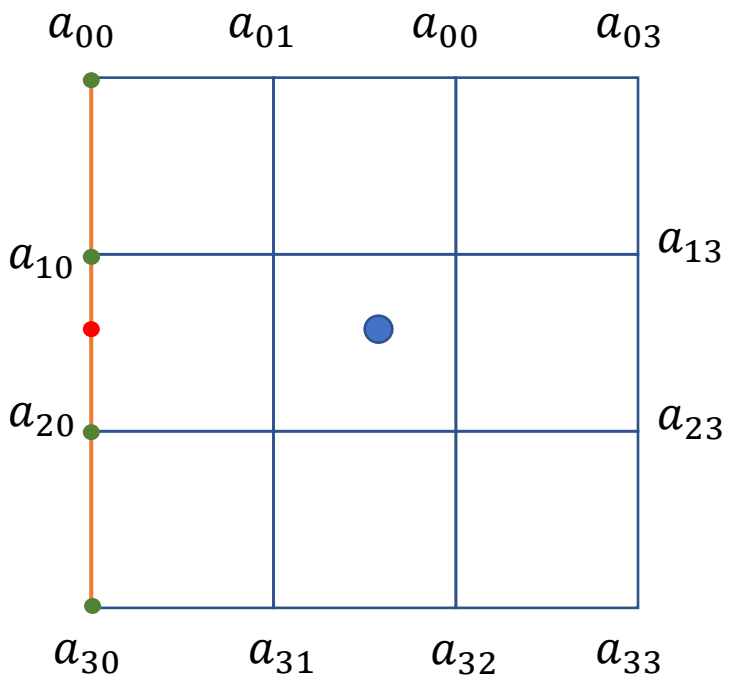
Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward



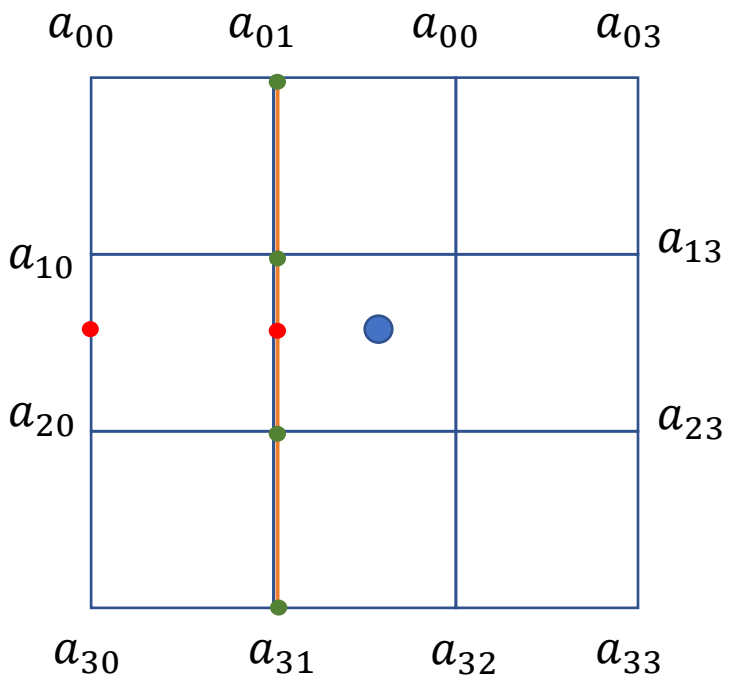
Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward



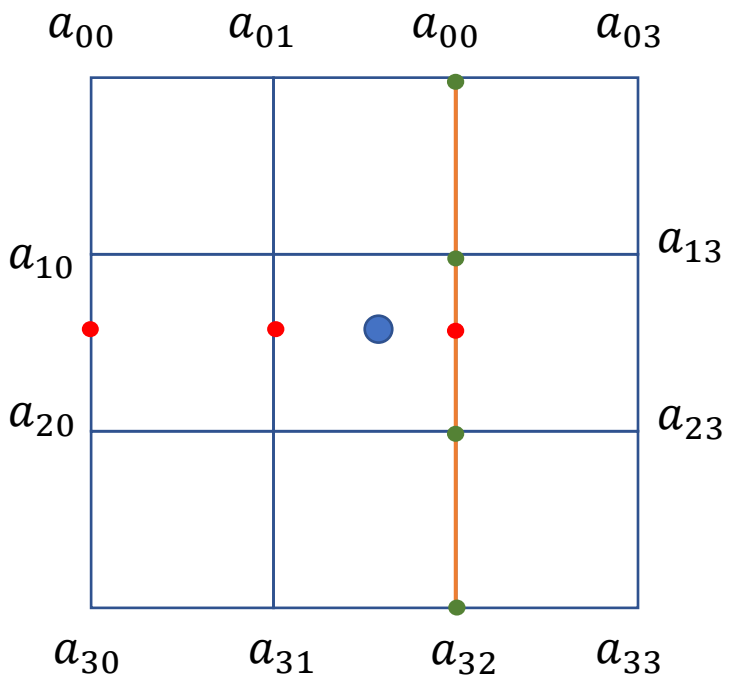
Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward



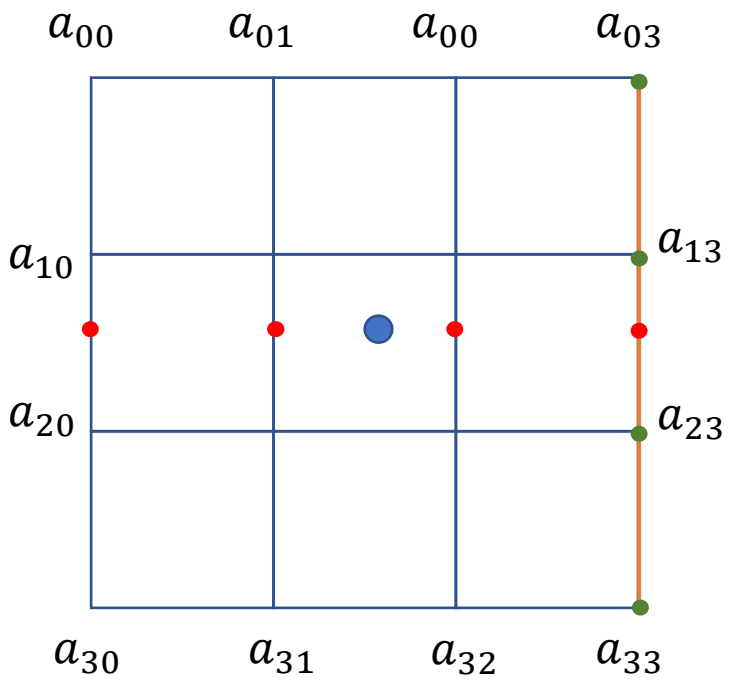
Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward



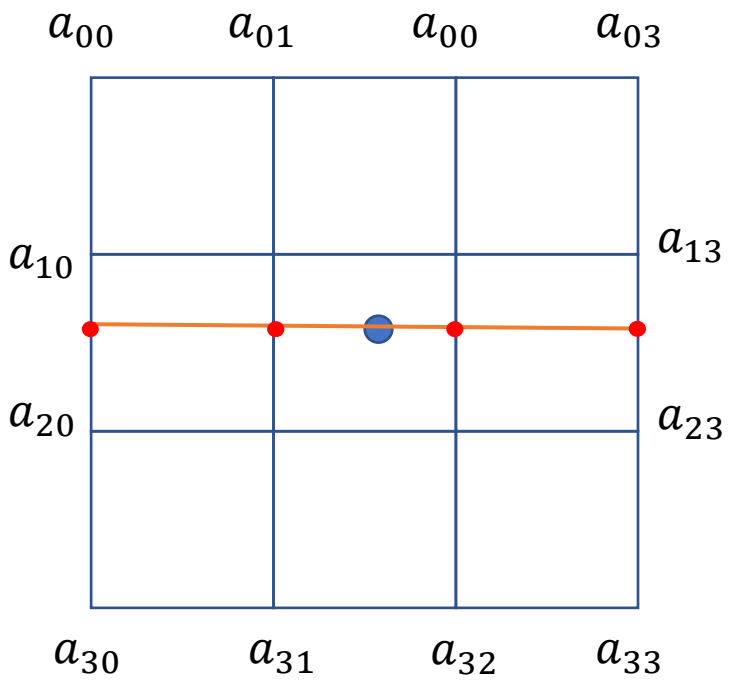
Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward



Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward



Bicubic Interpolation

- Cubic Interpolation in x-direction and y-direction afterward

$$P = \sum_{i=0}^{3} \sum_{j=0}^{3} C_{ij} \Delta x^i \Delta y^j$$

- Cubic spline is fitted
- The result depends on 16 neighboring intensity
- Please refer to https://en.wikipedia.org/wiki/Bicubic_interpolation for computation detail.

Bicubic Interpolation

$$P = \sum_{i \leq 1} \sum_{j \leq 1} C_{ij} \Delta x^i \Delta y^j$$

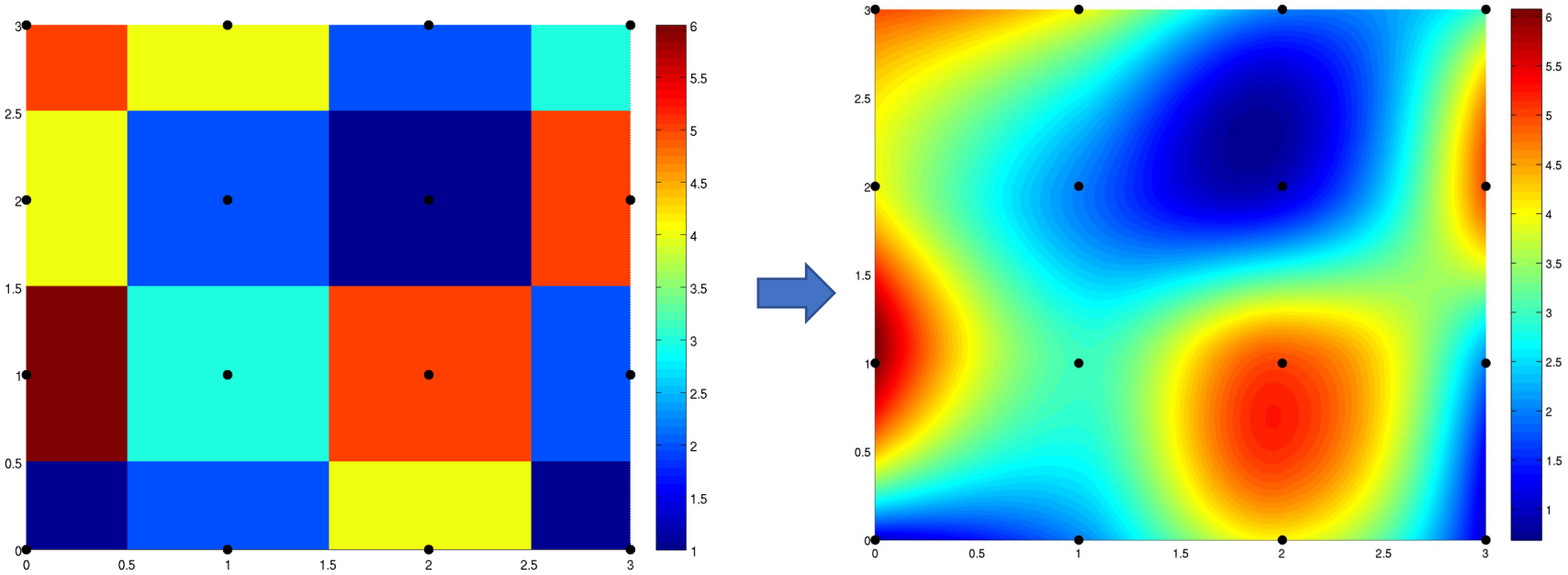
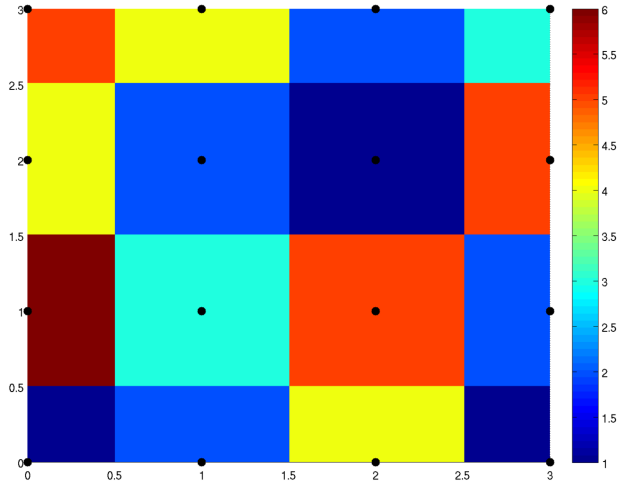


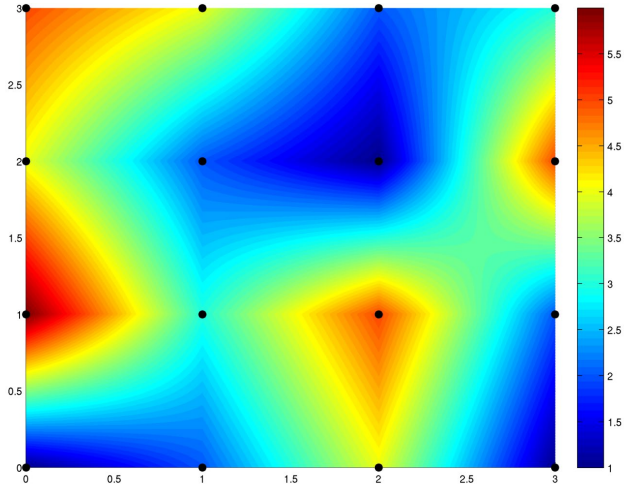
Image Courtesy: Wikipedia.com

Summary



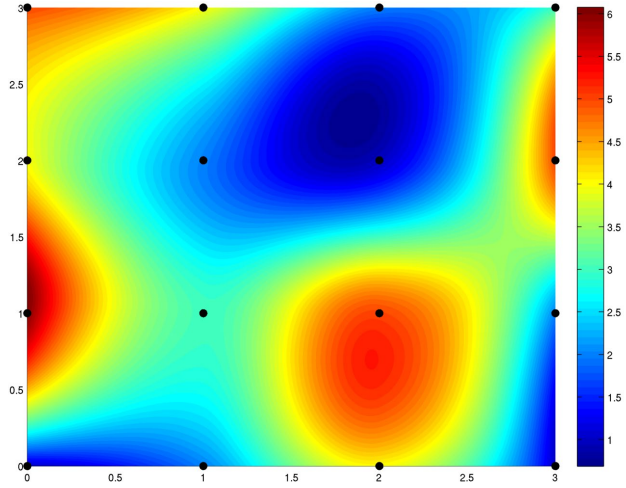
Nearest neighbor

Fast ++
Quality -



Bilinear

Fast +
Quality +



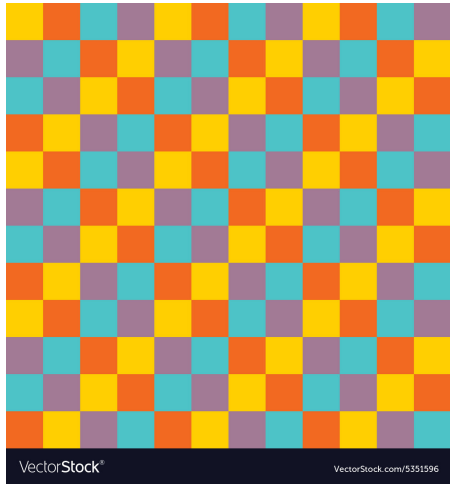
Bicubic

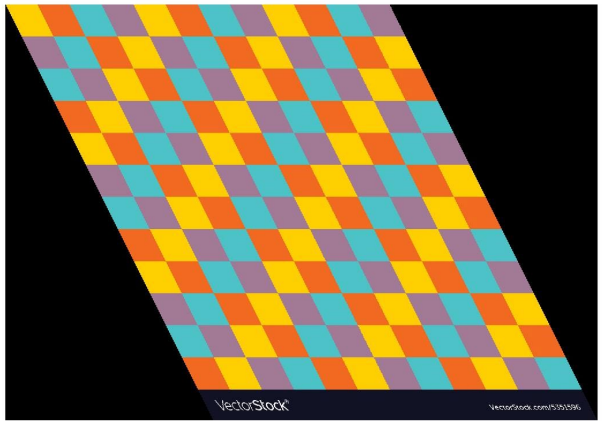
Fast -
Quality ++

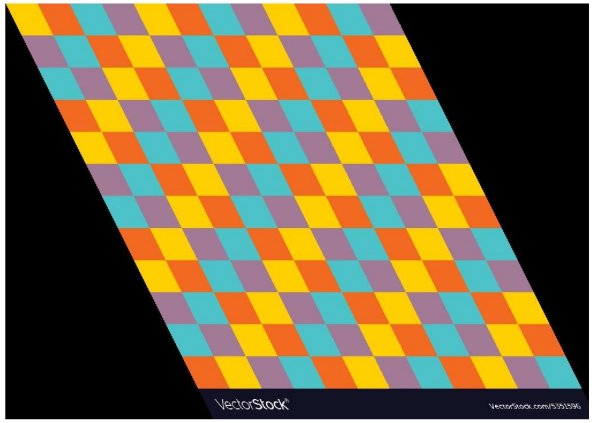
Image Courtesy: Wikipedia.com

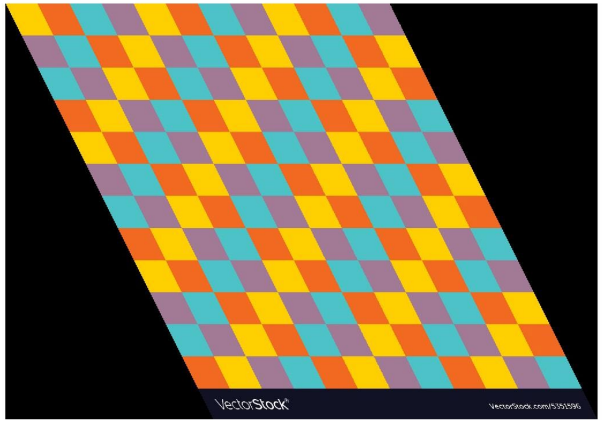
MATLAB Example

```
clear;
I=imread('chess.jpg');
imshow(I);

%define transformation matrix
tform = affine2d([1 0 0; .5 1 0; 0 0 1])

%use imwarp function to warp with nearest neighbour interpolation
A = imwarp(I,tform, 'nearest');
figure
imshow(A)

%use imwarp function to warp with bilinear interpolation
B= imwarp(I,tform,'bilinear');
figure
imshow(B)

%use imwarp function to warp with bicubic interpolation
C = imwarp(I,tform, 'cubic');
figure
imshow(C)
```

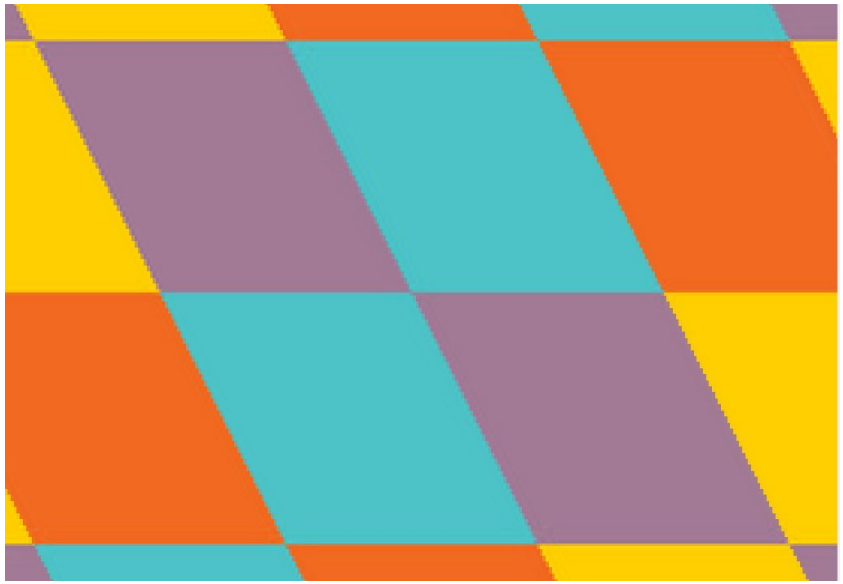








A closer view



Nearest neighbor

bilinear

bicubic

Another Example

Bilinear Interpolation

Bicubic Interpolation

Image Courtesy : Szeliski

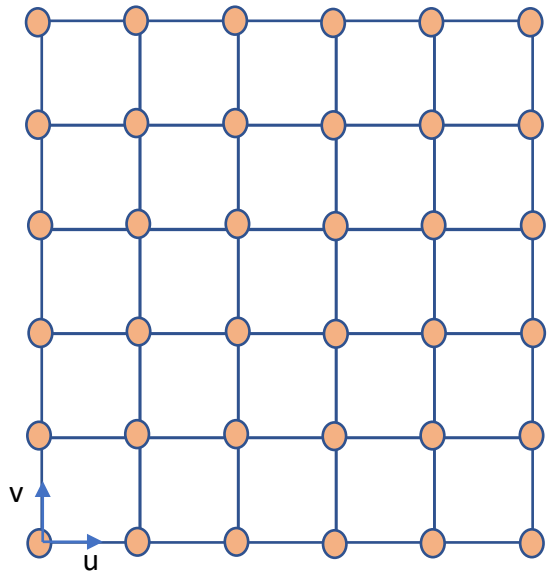
Resampling

Two Methods of warping

- Forward Warping
- Reverse Warping

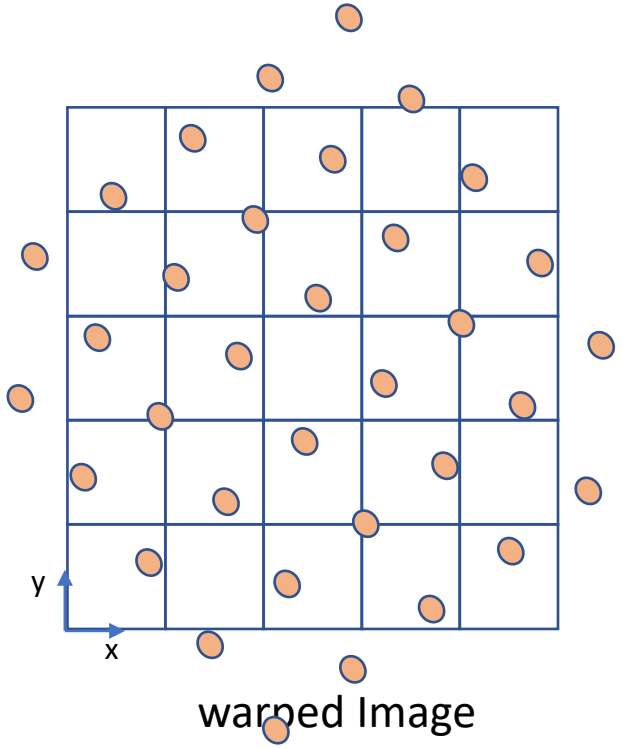
Resampling – Forward warping

- For every pixel in the input image *b*, compute the output value in the output image *a*



Original Image

$^{a}W_{b}$



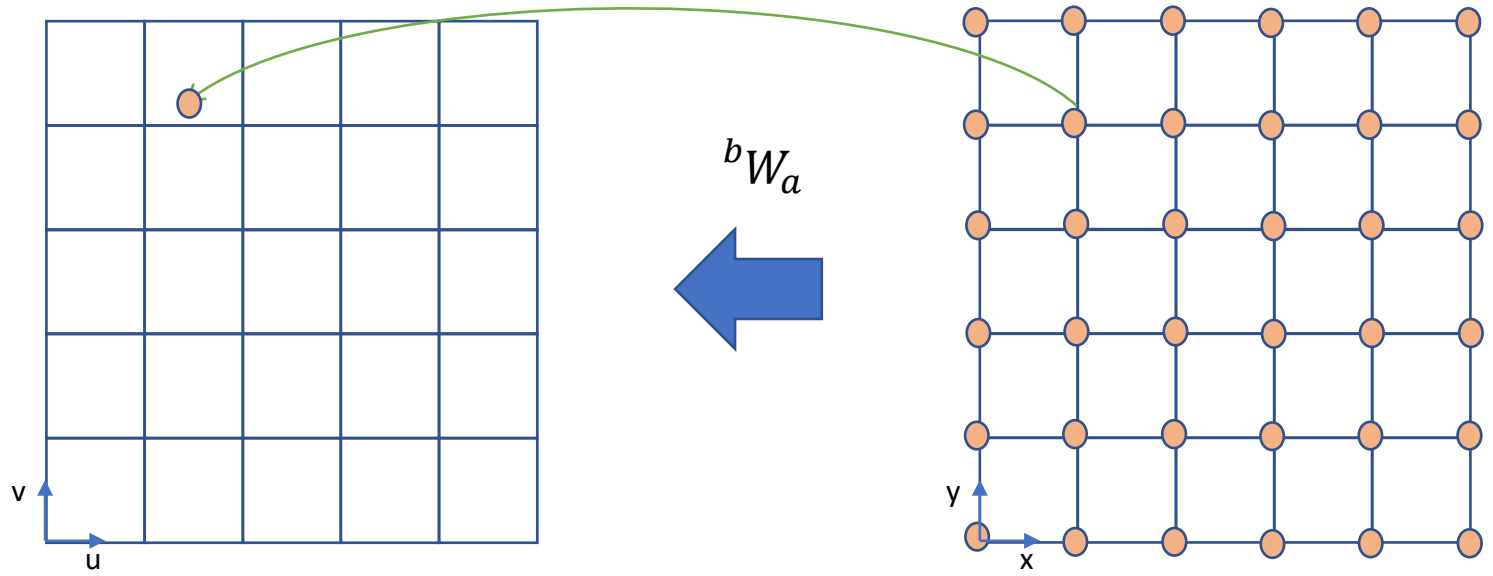
warped Image

Forward Warping

```
for (int u=0, u<umax;u++) {
  for (int v=0, v<umax;v++) {
    float x = W_x(u,v);
    float y = W_y(u,v);
    b(x,y)=a(u,v);
  }
}
```

Resampling – Inverse Warping

- For every pixel in the output, compute the inverse transform to input.
- Interpolation in input image to get the intensity value



Original Image

warped Image

Inverse Warping

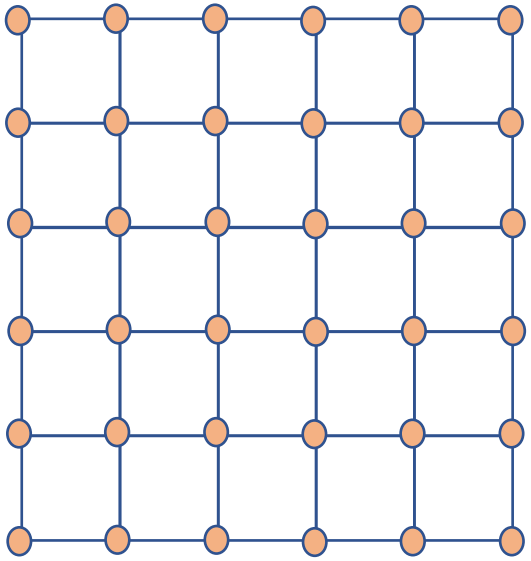
```
for (int x=0, x<umax;x++) {
  for (int y=0, y<umax;y++) {
    float u = W_x^-1 (x,y);
    float v = W_y^-1(x,y);
    b(x,y)=a(u,v);
  }
}
```

Which one is better?

Why?

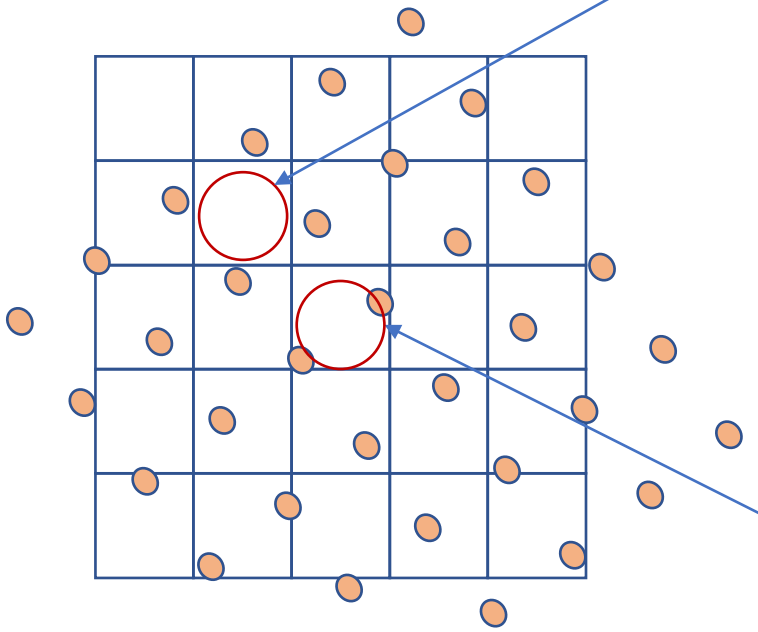
Resampling – Forward warping

- What is the problem?



Original Image

$^{a}W_{b}$



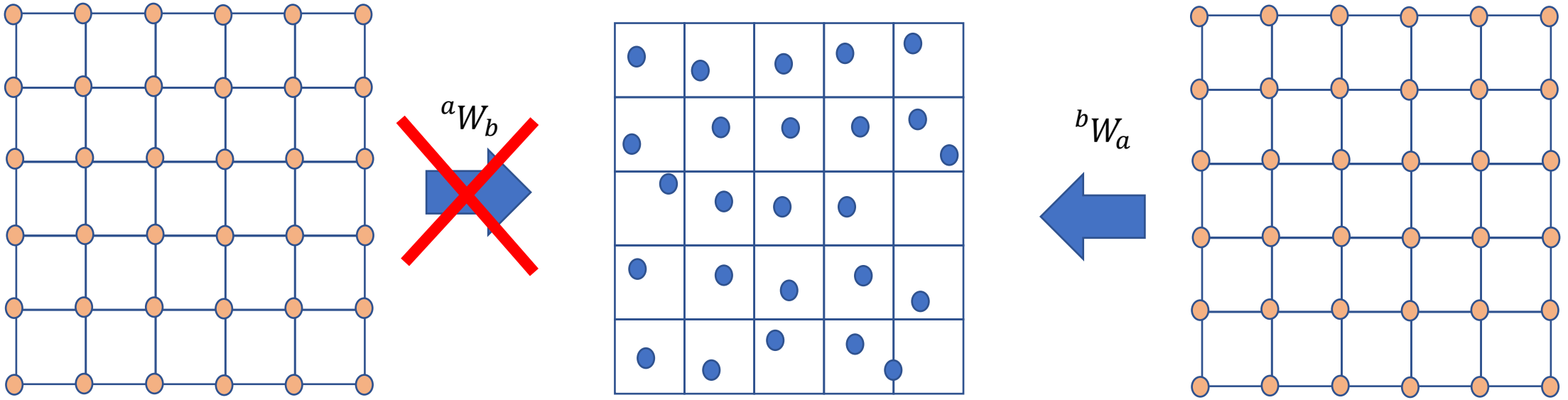
warped Image

Some destinations have no pixel

More than one pixels are mapped to the same destination

Summary

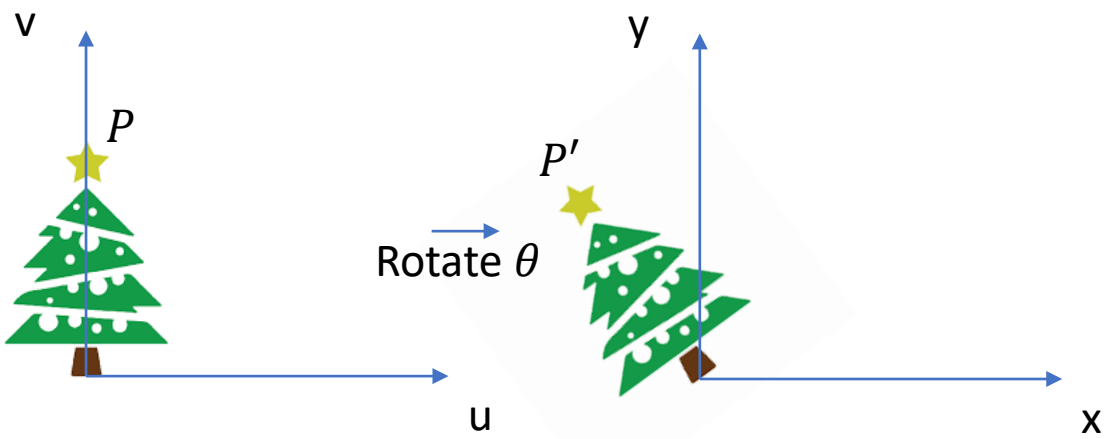
- Forward warping can lead to missing pixels in the output image
- Inverse warping method is adopted and followed by bilinear/bicubic interpolation method.
- Also use inverse warping



Example: Rotation

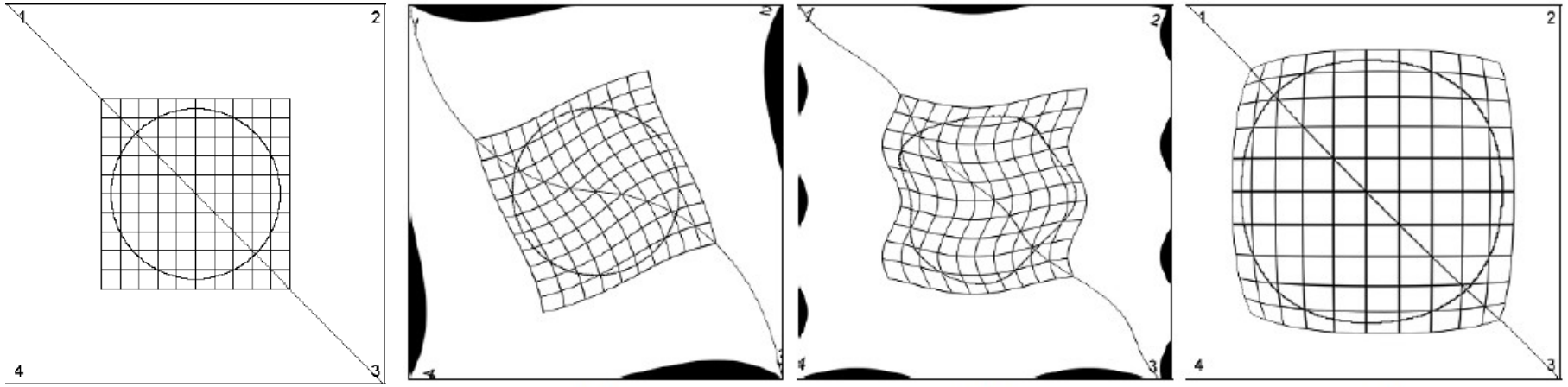
- Inverse Warping

- for (int x=0, x<umax;x++) {
- for (int y=0, y<umax;y++) {
- float u = $xcos(-\theta) - ysin(-\theta)$;
- float v = $xsin(-\theta) + ycos(-\theta)$;
- b(x,y)=resample(u,v);
- }
- }



$$x = ucos\theta - vsin\theta$$
$$y = usin\theta + vcos\theta$$

Non-Linear Image Warping



Original

Twirl

Ripple

Spherical

Slide from E. Agu. WPI

Twirl

- **Notation:** Instead using texture colors at (x',y'), use texture colors at twirled (x,y) location
- Twirl?
  - Rotate image by angle ***α*** at center or anchor point (x_c,y_c)
  - Increasingly rotate image as radial distance ***r*** from center increases (up to r_{max})
  - Image unchanged outside radial distance r_{max}

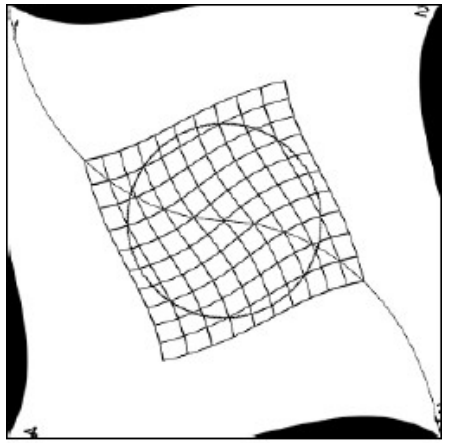
$$T_x^{-1}: \quad x = \begin{cases} x_c + r \cdot \cos(\beta) & \text{for } r \le r_{\max} \\ x' & \text{for } r > r_{\max}, \end{cases}$$

$$T_y^{-1}: \quad y = \begin{cases} y_c + r \cdot \sin(\beta) & \text{for } r \le r_{\max} \\ y' & \text{for } r > r_{\max}, \end{cases}$$

with

$$d_x = x' - x_c, \qquad r = \sqrt{d_x^2 + d_y^2},$$

$$d_y = y' - y_c, \qquad \beta = \operatorname{Arctan}(d_y, d_x) + \alpha \cdot \left(\tfrac{r_{\max} - r}{r_{\max}}\right).$$



(a)

(d)

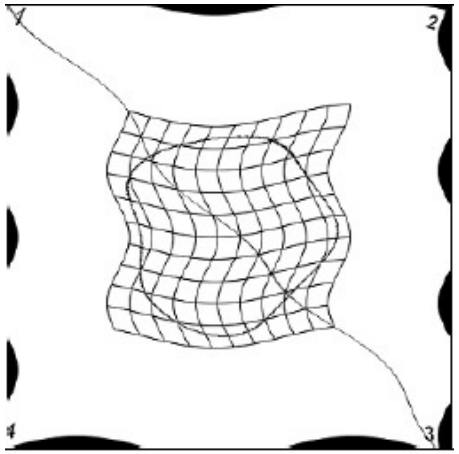
Slide from E. Agu. WPI

Ripple

- Ripple causes wavelike displacement of image along both the x and y directions

$$T_x^{-1}: \quad x = x' + a_x \cdot \sin\left(\frac{2\pi \cdot y'}{\tau_x}\right),$$
$$T_y^{-1}: \quad y = y' + a_y \cdot \sin\left(\frac{2\pi \cdot x'}{\tau_y}\right).$$

- Sample values for parameters (in pixels) are
  - τ_x= 120
  - τ_y= 250
  - a_x= 10
  - a_y= 15



(b)

(e)

Slide from E. Agu. WPI

Spherical Transformation

- Imitates viewing image through a lens placed over image
- Lens parameters: center (x_c, y_c), lens radius r_{max} and refraction index ρ
- Sample values ρ = 1.8 and r_{max} = half image width

$$T_x^{-1}: \quad x = x' - \begin{cases} z \cdot \tan(\beta_x) & \text{for } r \le r_{\max} \\ 0 & \text{for } r > r_{\max}, \end{cases}$$

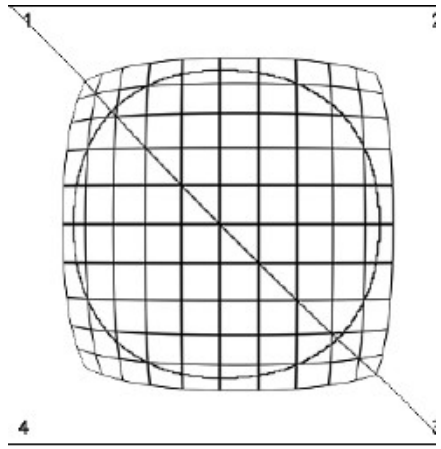
$$T_y^{-1}: \quad y = y' - \begin{cases} z \cdot \tan(\beta_y) & \text{for } r \le r_{\max} \\ 0 & \text{for } r > r_{\max}, \end{cases}$$

$$d_x = x' - x_c, \qquad r = \sqrt{d_x^2 + d_y^2},$$

$$d_y = y' - y_c, \qquad z = \sqrt{r_{\max}^2 - r^{\cdot 2}},$$

$$\beta_x = \left(1 - \tfrac{1}{\rho}\right) \cdot \sin^{-1}\left(\frac{d_x}{\sqrt{(d_x^2 + z^2)}}\right),$$

$$\beta_y = \left(1 - \tfrac{1}{\rho}\right) \cdot \sin^{-1}\left(\frac{d_y}{\sqrt{(d_y^2 + z^2)}}\right).$$



(c)

(f)

Slide from E. Agu. WPI

Other Applications of Image Warping

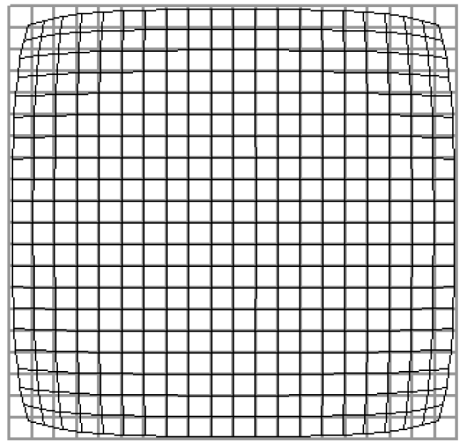
- The image warping and interpolation technique can be adopted in lens distortion correction

$$r^2 = (x - x_p)^2 + (y - y_p)^2$$

$$\delta_r = ((K_3 r^2 + K_2) r^2 + K_1) r^2$$

$$x_1 = x(1 + \delta_r)$$

$$y_1 = y(1 + \delta_r)$$



Barrel Distortion

- Where (x_p, y_p) are the coordinates of the lens axis and K_1, K_2, K_3 are distortion parameters.

Other Applications : Image Warping

Source: http://medium .com

Other Application: Image Scaling

- What if we want display a small image onto a big screen?
- We need to do image resizing?
- What is image resizing?
  - Enlarge or reduce the image size(number of pixels representing the images)
  - i.e. resample the image at different sampling rate

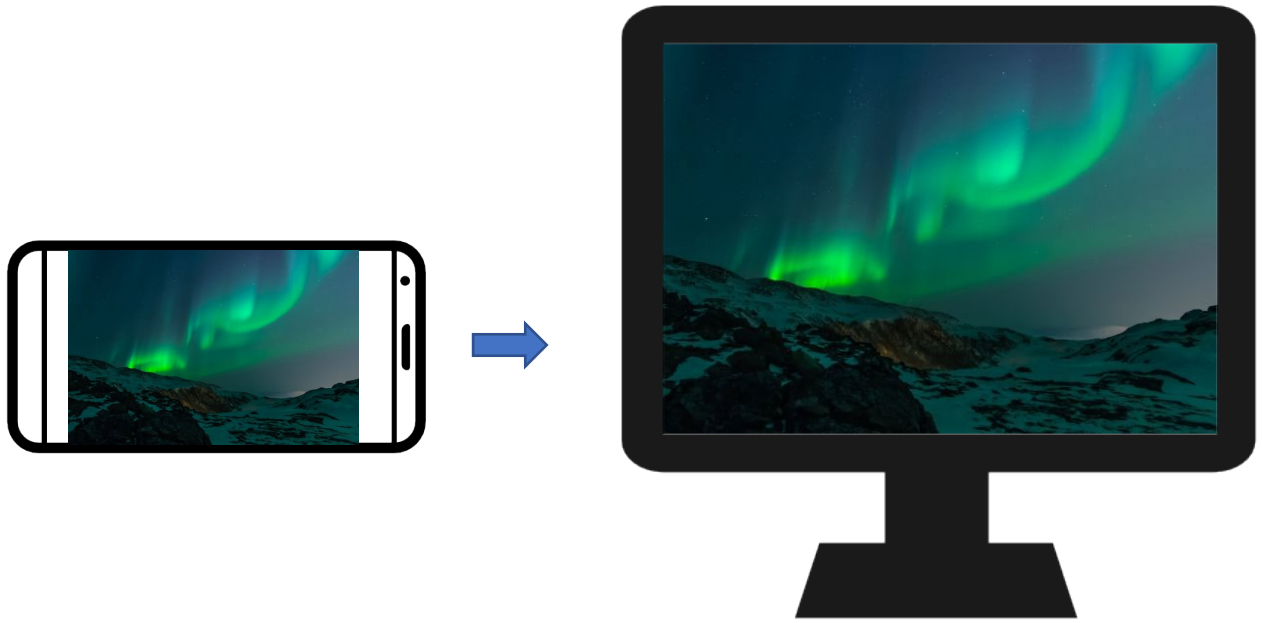


Photo source: https://www.w3schools.com/

Example (without smoothing filter)

Original

Nearest Neighbour

Bilinear

Bicubic

Example (with smoothing filter)

Original

Nearest Neighbour

Bilinear

Bicubic

MATLAB Code for Reference

```
img=imread('lena.jpg');

%downsampling without filtering
img1=imresize(img,0.5,'nearest');
%upsampling with different filters:
img2rep=imresize(img1,2,'nearest');
img2lin=imresize(img1,2,'bilinear');
img2cubic=imresize(img1,2,'bicubic’);
```

MATLAB Code without filtering

```
img=imread('lena.jpg');
%down sampling with filtering
img1=imresize(img,0.5,'bilinear',11);

%upsampling with different filters
img2rep=imresize(img1,2,'nearest');
img2lin=imresize(img1,2,'bilinear');
img2cubic=imresize(img1,2,'bicubic');
```

MATLAB Code with filtering

Summary of Today's lecture

- What is a camera?
- Image is a 2D rectilinear array of pixels
- Pinhole Camera
- Size of aperture: large vs small
- Lens:
  - depth of field
  - Field of view
- Affine transformation
- Image Warping
- Image Resizing