

Computer Vision 2024

Week – 1 lecture One

Miaomiao Liu



- What is Computer Vision?
- Why to study Computer Vision?
- What can Computer Vision do?
- What do you expect to learn from this course?

Synonym

Computer Vision

- Image Understanding
- Machine Vision
- Robotic Vision

What is Computer Vision?

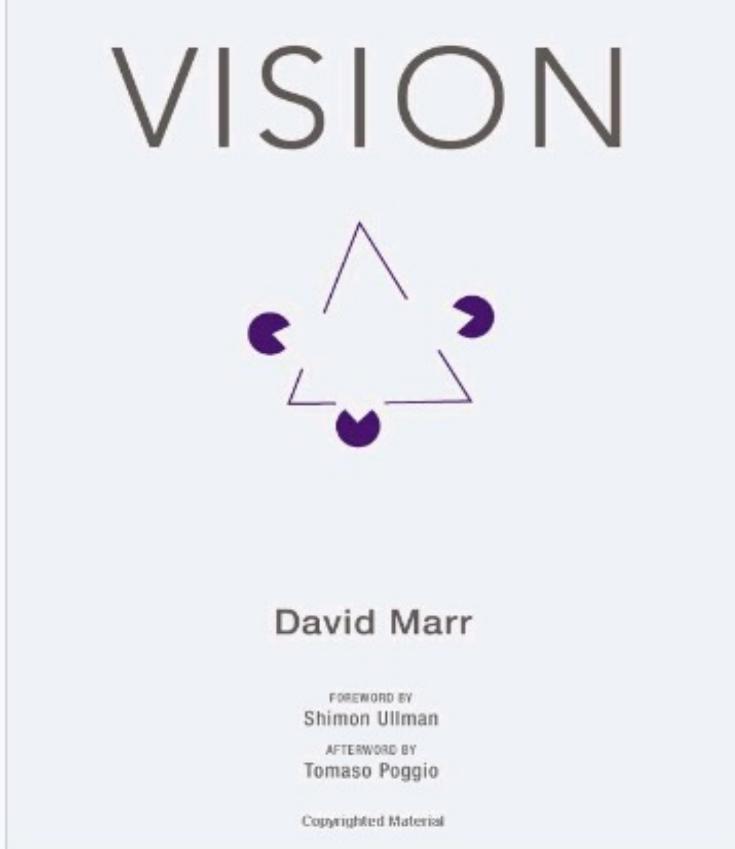
- (aka machine vision, robot vision, image understanding)
- and **applications of Computer Vision**

What is Computer Vision?

- Humans see things very easily.
- Can a computer do the same task?
 - Relieve human from tedious work
 - Perform more accurate measurement than what the human can do
- **How to teach a computer (or robot) to see (and understand what it sees)?**

What is Vision?

- Is [the art] *``to know what is where, by looking.''*



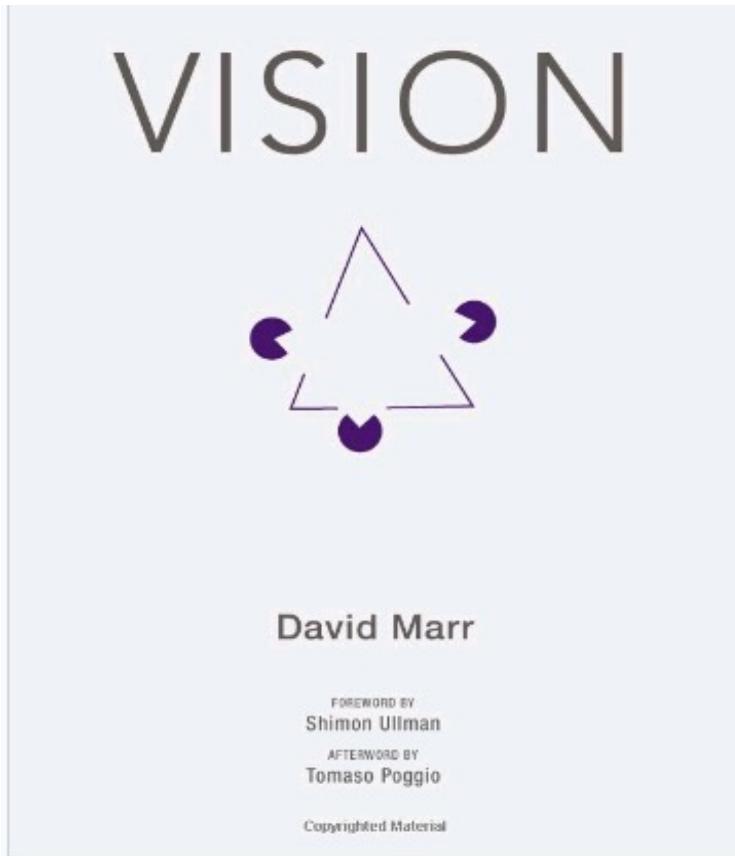
Published in 1982



David Marr (1945 --1980)

What is Vision?

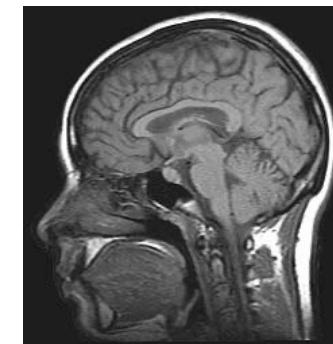
- The process of discovering from images what is present in the world and where it is.



David Marr (1945 --1980)

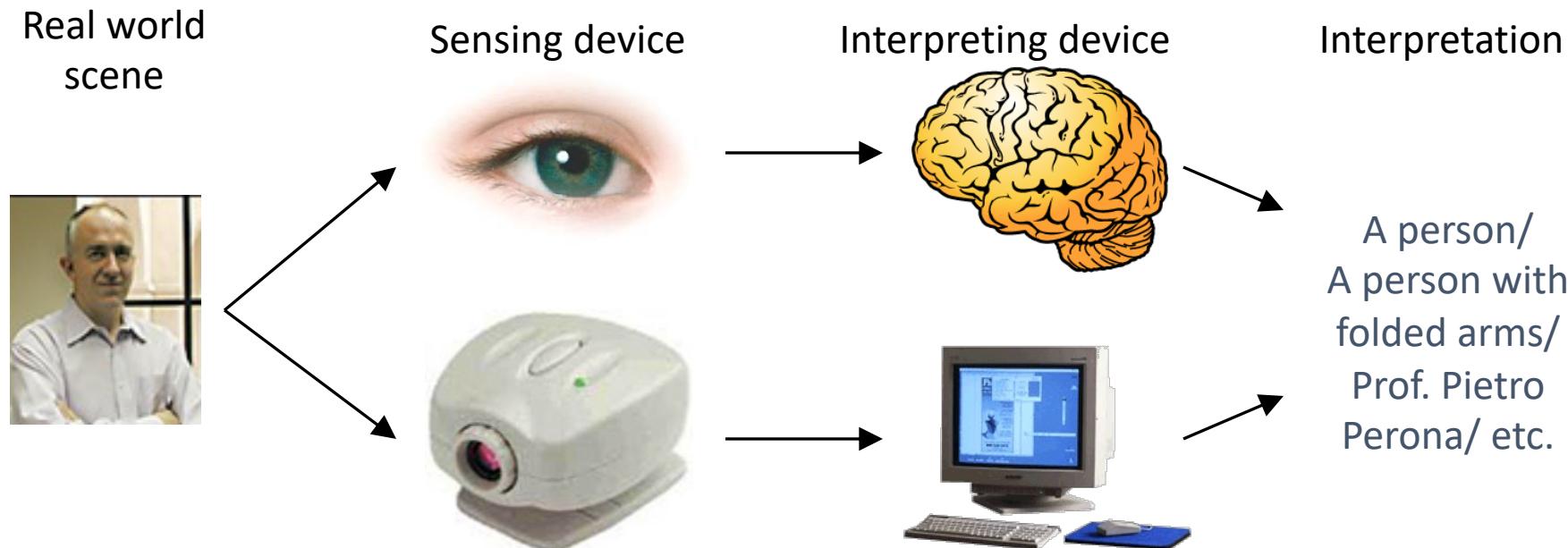
What is Computer Vision?

- Computer Vision is concerned with *the theory* of building artificial vision systems that obtain useful information from images.
- Image data taken by many forms, e.g. *video sequence, depth images, multi-dimensional data from a medical scanner etc.*



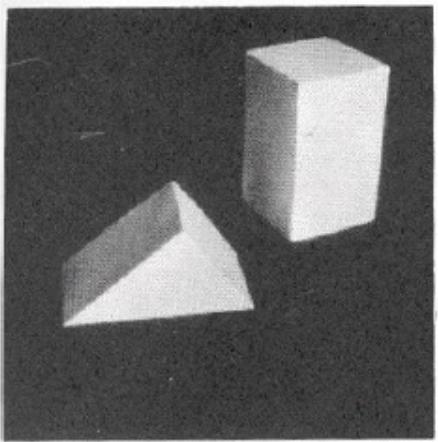
Computer Vision Problems

- Make a computer see and understand images.
- We know it is physically possible – we do it every day and effortlessly!

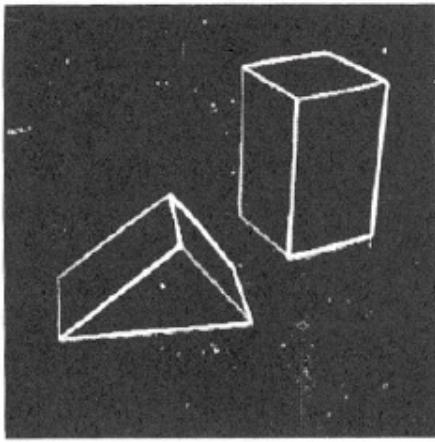


Brief history of computer vision research

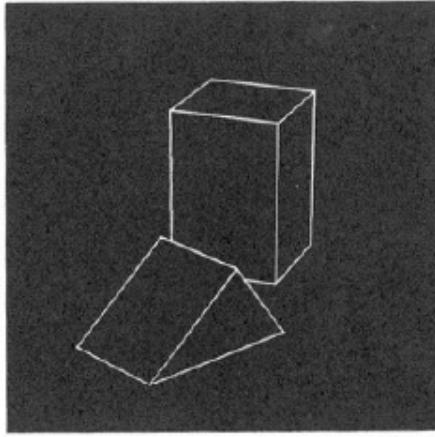
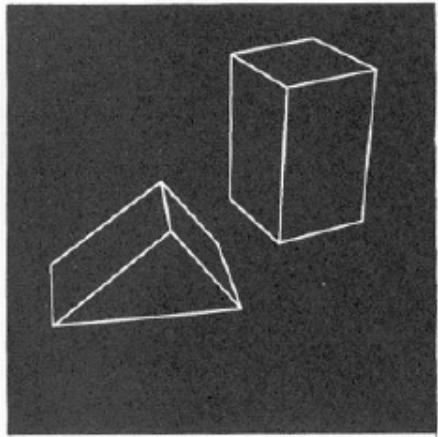
Origins of Computer Vision



(a) Original picture.



(b) Differentiated picture.



L. G. Roberts, *Machine Perception of Three Dimensional Solids*, Ph.D. thesis, MIT Department of Electrical Engineering, 1963.

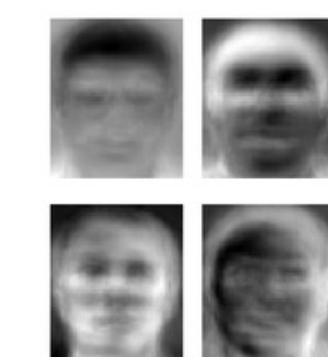
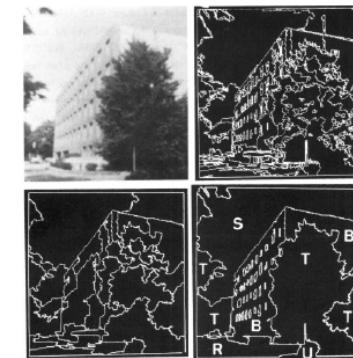
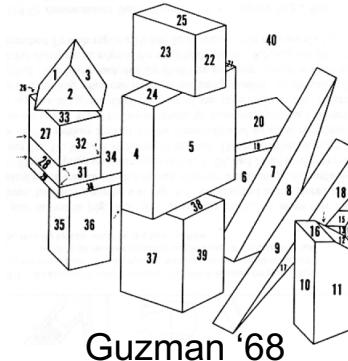
An (overly) optimistic start

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “*spend 3 months in this summer linking a camera to a computer and getting the computer to describe what it saw*”.

Now, more than fifty years, we know the problem is significantly more difficult than a 3-month student project.

Brief history of computer vision

- 1963: Robert's thesis
- 1966: Minsky assigns computer vision as an undergrad summer project
- 1960's: interpretation of synthetic worlds
- 1970's: some progress on interpreting selected images
- 1980's: ANNs come and go; shift towards geometry and increased mathematical rigor, Marr's book published (after he died).
- 1990's: face recognition; statistical analysis in vogue
- 2000's: broader recognition; large annotated datasets available; video processing starts; vision & graphics; vision for HCI; internet vision, etc.
- 2012: Kinect, big-data, Google Car, ... Deep Neural network, Deep learning

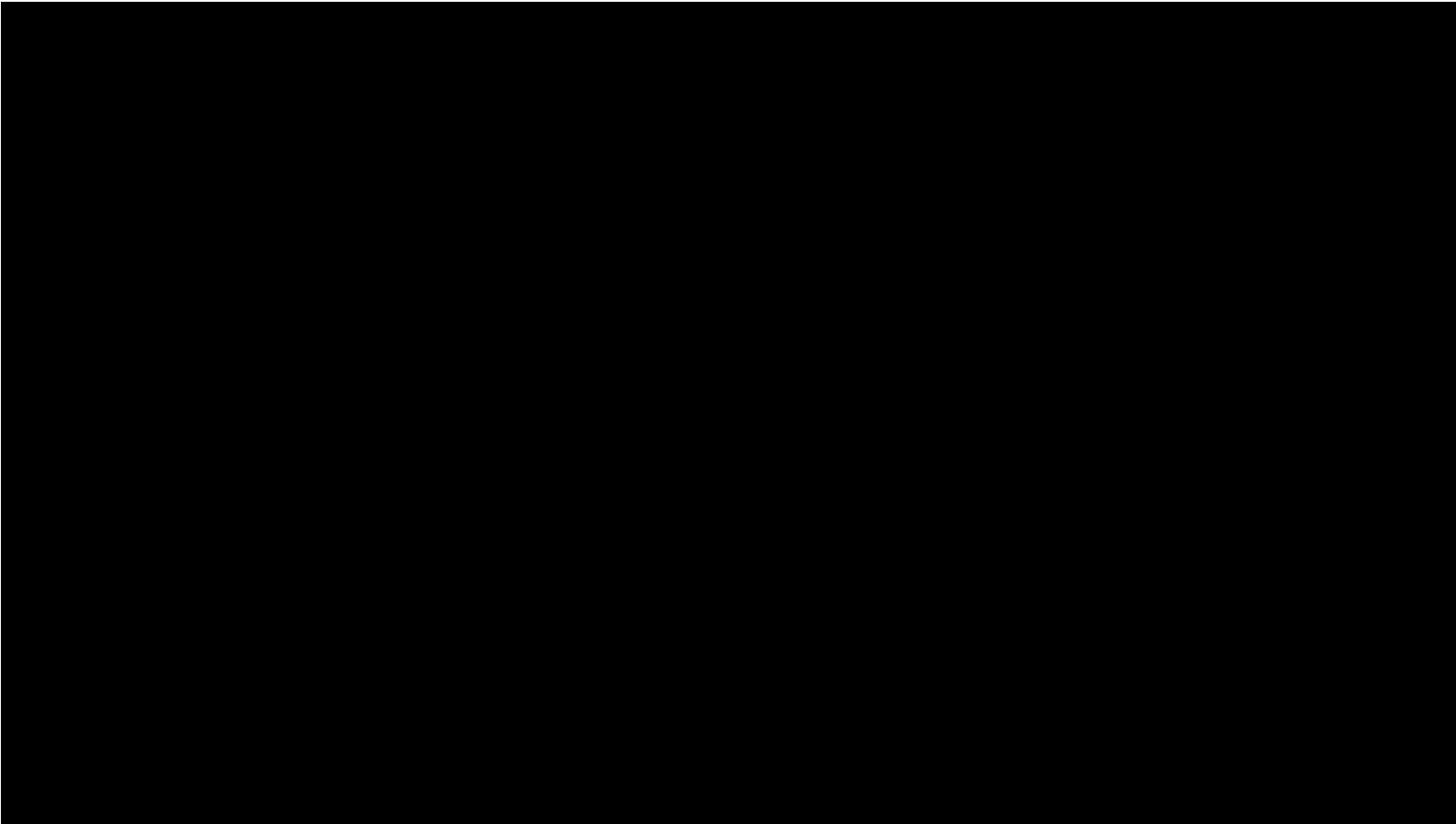


Turk and Pentland '91

Current State of the art of CV applications

- Apple Vision Pro
 - https://www.youtube.com/watch?v=IY4x85zqoJM&list=PPSV&ab_channel=Apple
- Google Car
 - <https://www.youtube.com/watch?v=B8R148hFxPw>
- Amazon Go
 - <https://www.youtube.com/watch?v=NrmMk1Myrxc>
- HoloLens- Microsoft
 - <https://www.youtube.com/watch?v=aYdB2xBNFek>

State of the art google self-driving car (GOOGLe)

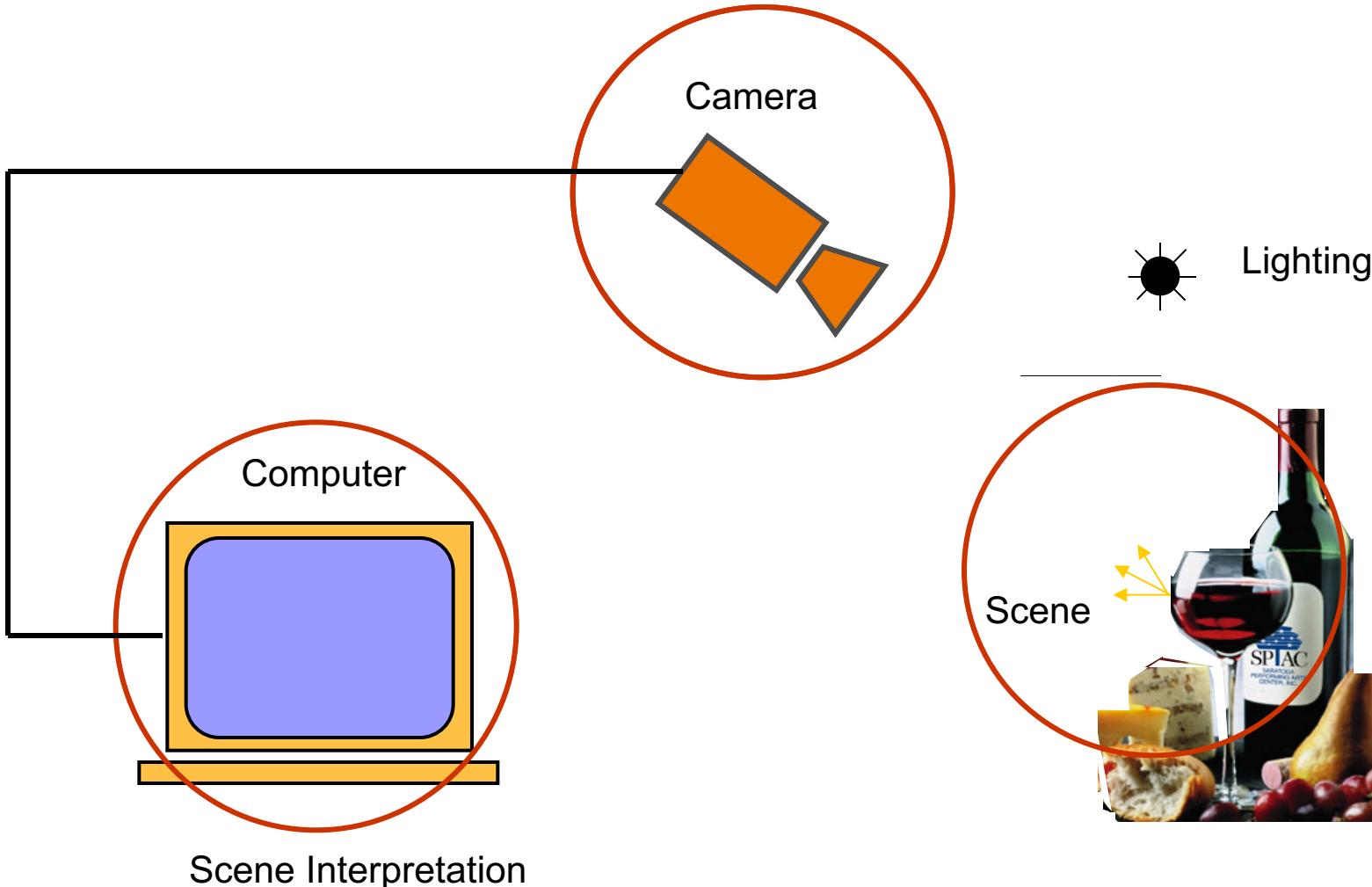


<https://www.youtube.com/watch?v=B8R148hFxPw>

What makes a **Computer Vision system** ?

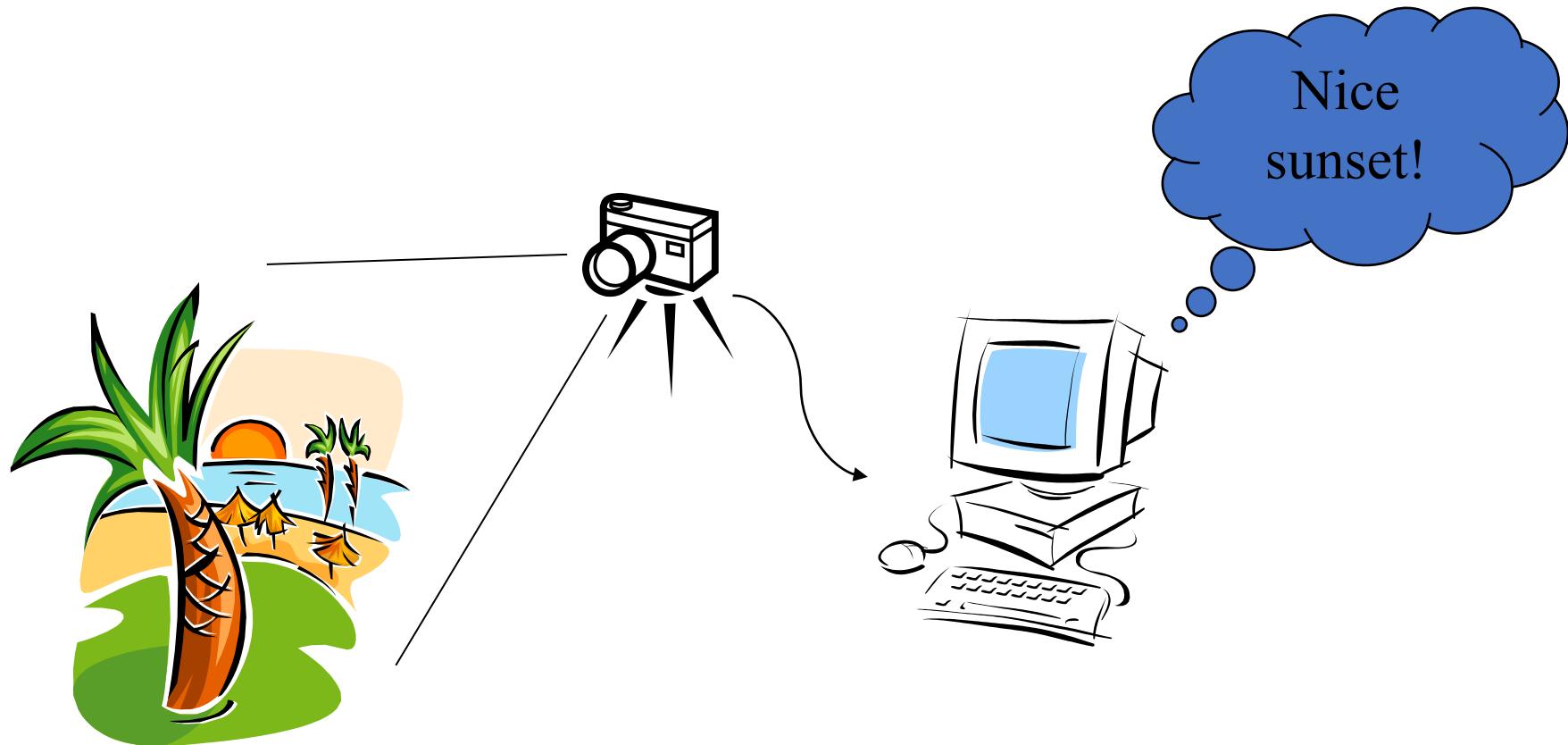
- In a **Computer Vision System**, a camera (or several cameras) is linked to a computer.
- The computer interprets images of a real world to obtain information useful for tasks such as navigation, manipulation and recognition.

Computer Vision System Example



Srinivasa Narasimhan's slide

- “*Making computers see*”



- *To ‘see’ is not only to take images, but also ‘to understand’ it.*

Understand the contents of images and videos.



What kind of scene?

Where are the buildings?

How far are the buildings from the camera?

...

What we could see



What computer sees

...and the output is

- A *harbor*...
- ... *with many dozens of boats*;
- ... *water is calm and glassy*;
- ... *vertical masts*;
- ... *mountains in background*,
- ... *blue sky with a touch of clouds*...

Interpret images.

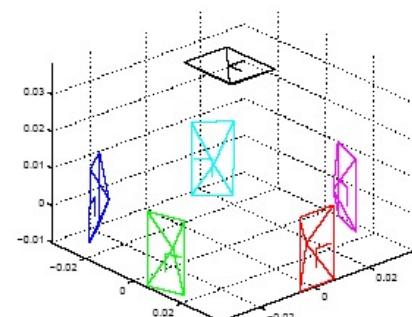
Image → Symbols, Semantics, Meanings,..

- Previous Computer Vision Projects @ ANU

Autonomous vehicles / self-driving cars



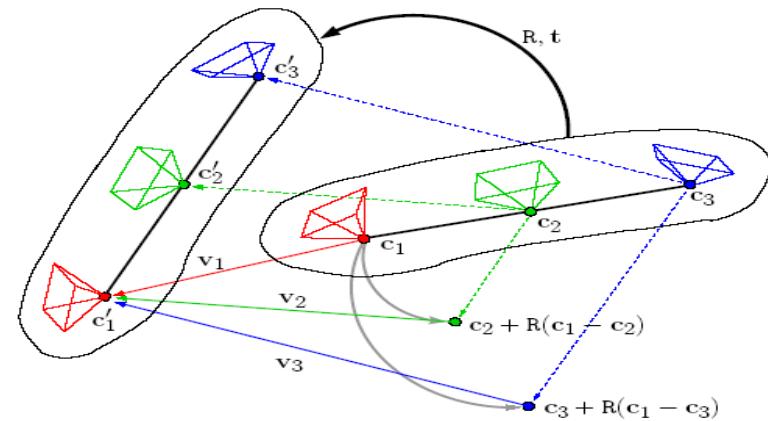
Multiple Camera City Modelling



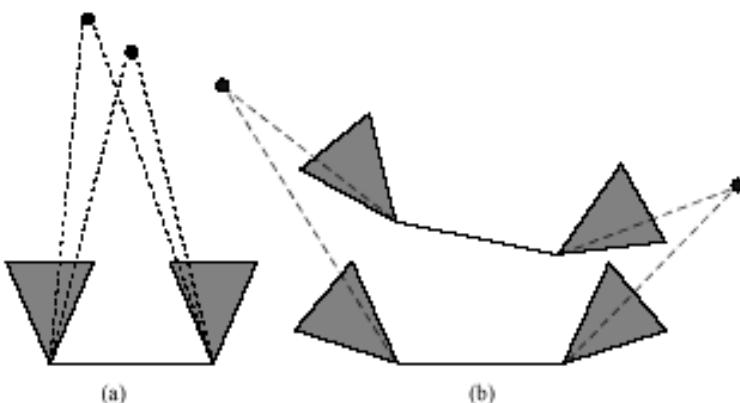
(a)



(b)



Multi-camera system



(a) Overlapping vs. (b) non-overlapping
Multi-camera systems

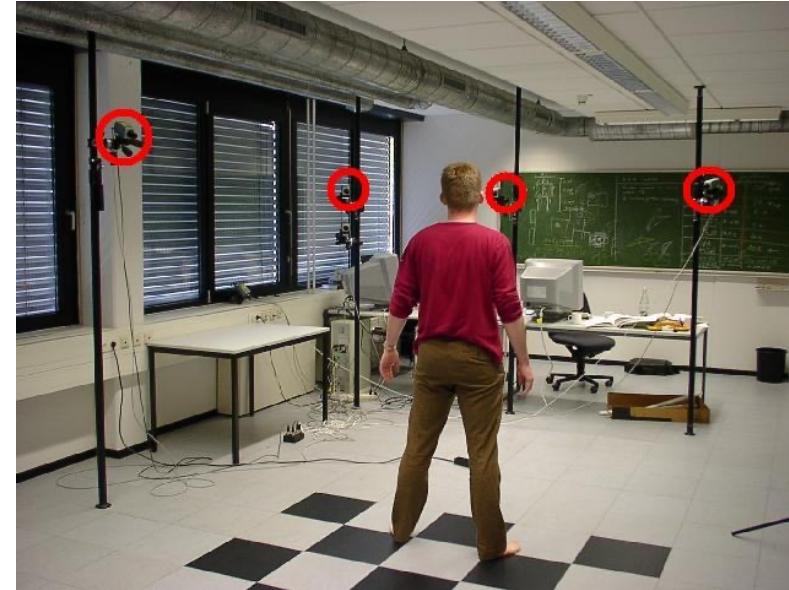


Images from one camera cluster

Kim and Li, Hartley, CVPR 08

Human Body Shape and Motion 3D Capture

- Marker-free
- No need to wear a special suit or gloves
- No dedicated hardware
- Standard video cameras
- Cost-effective

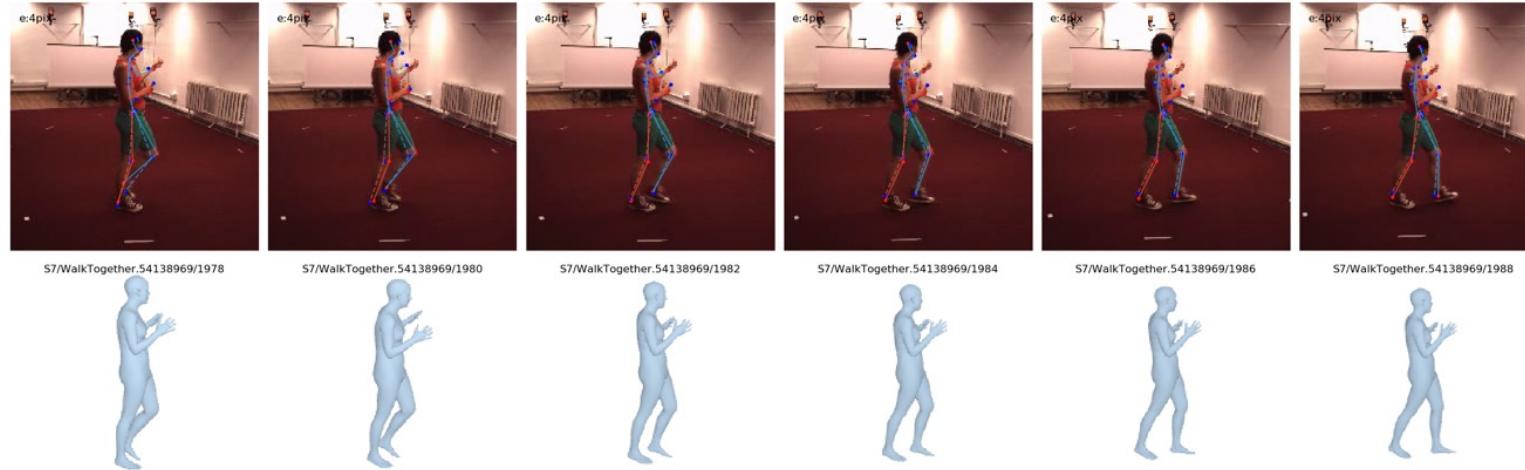


Motion Capture Environment (Image courtesy of Christian Theobalt)

Human Body Shape and Motion Prediction

- Predict Human Motion Dynamics from a Video sequence

Historical Data:

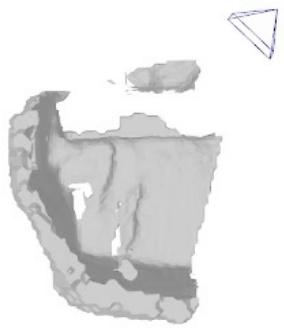


Predict Future Human Motion:



Mao, Liu, Salzmann, Li, ICCV19

3D Reconstruction

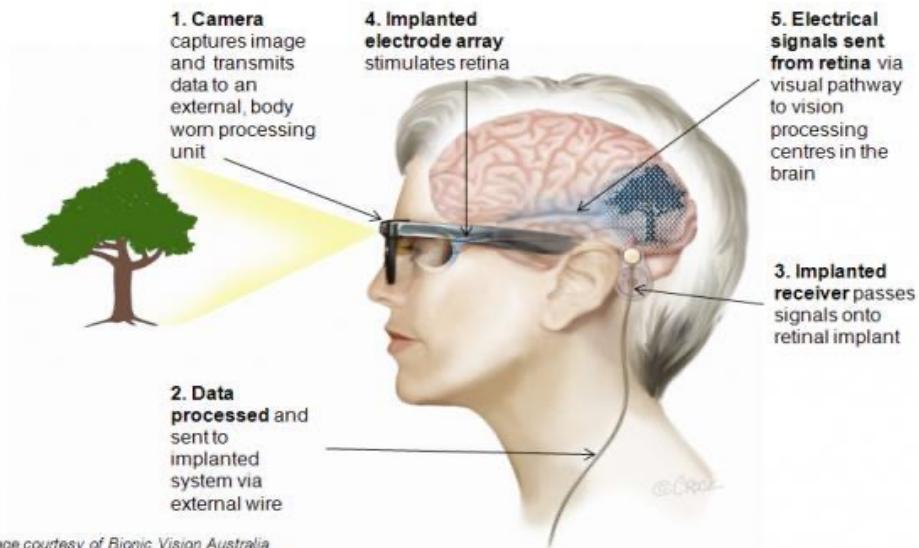


Australian Bionic Eye

- 2020 Summit Project
- Budget \$50M for first 4 years
- 5 BVA Members and 2 contributing universities
- Officially started July 2010

The bionic eye - how it works

First prototype: Wide-view neurostimulator



Bringing knowledge to life



Patient trial in Canberra 2014



- Ms Ashworth had her first 'unplugged' trial in *Canberra*, 2014.
- <https://www.abc.net.au/news/2014-04-30/bionic-eye-patients-start-first-navigation-tests/5422174>



Australian Centre for Robotic Vision

- \$25M government funding for 2014 - 2020
- 4 Centre Nodes (QUT, ANU, Adelaide, Monash)
- 13 Chief Scientists national-wide





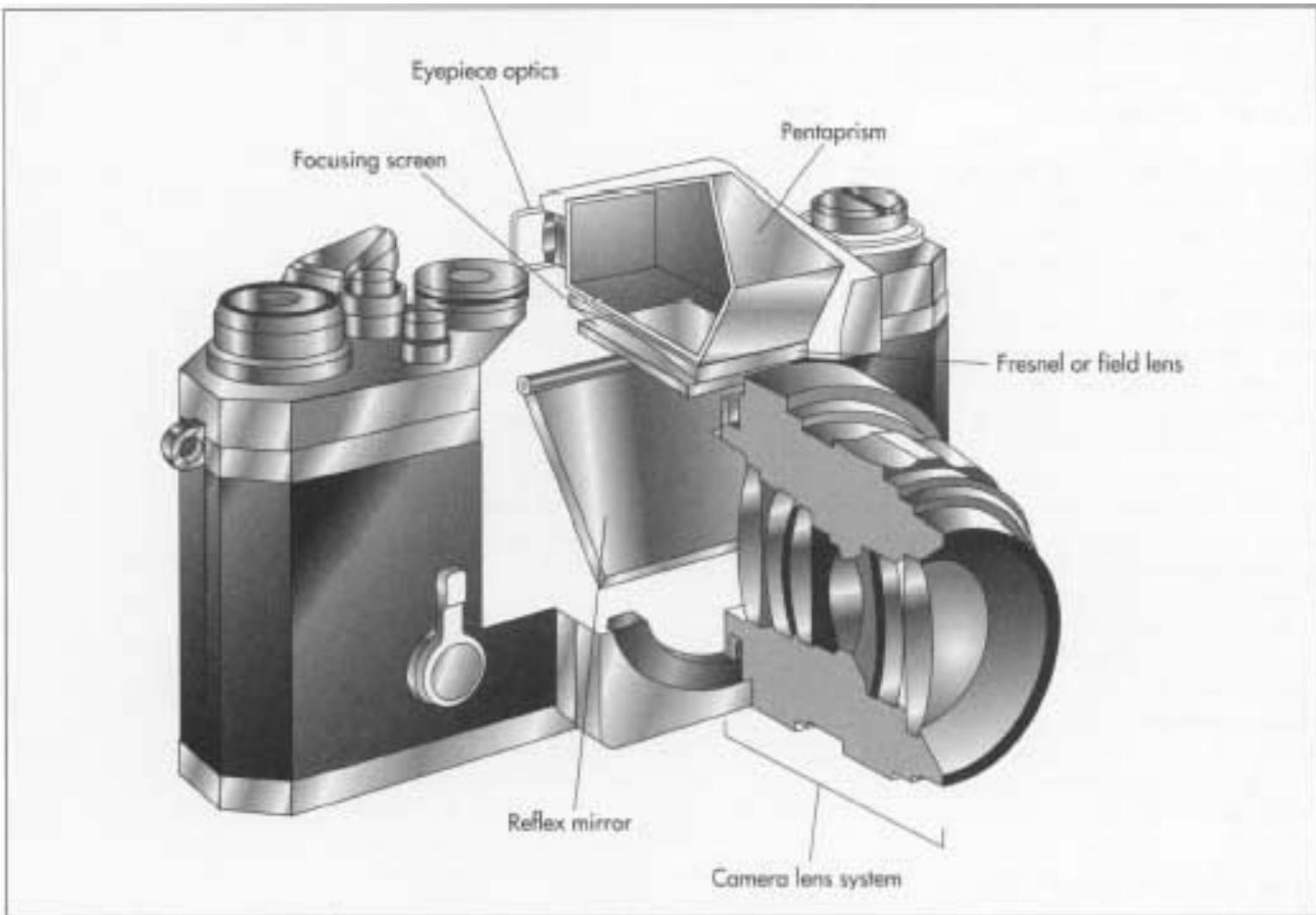
Outline

- What is a camera (sensor) ?
- Geometric image formation (details in 3D Vision Lectures)
- Photometric image formation
- Image representation
- Point Operation

Outline

- What is a camera (sensor) ?
- *Geometric image formation (details in 3D Vision Lectures)*
- Photometric image formation
- Image representation
- Point Operation

What is a camera ?

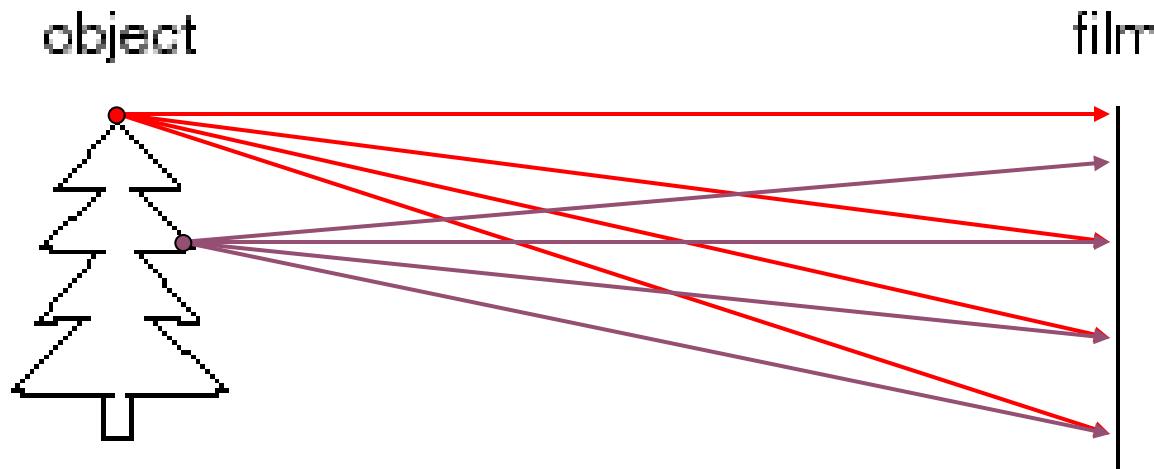


Slides from: F. Durand, S. Seitz, S. Lazebnik, S. Palmer

Let us design a camera

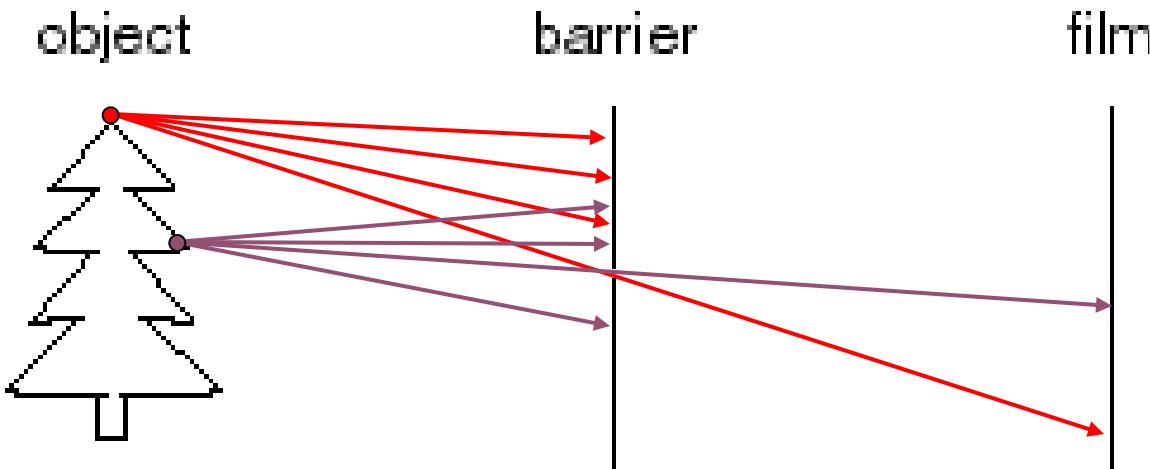


Let's design a camera



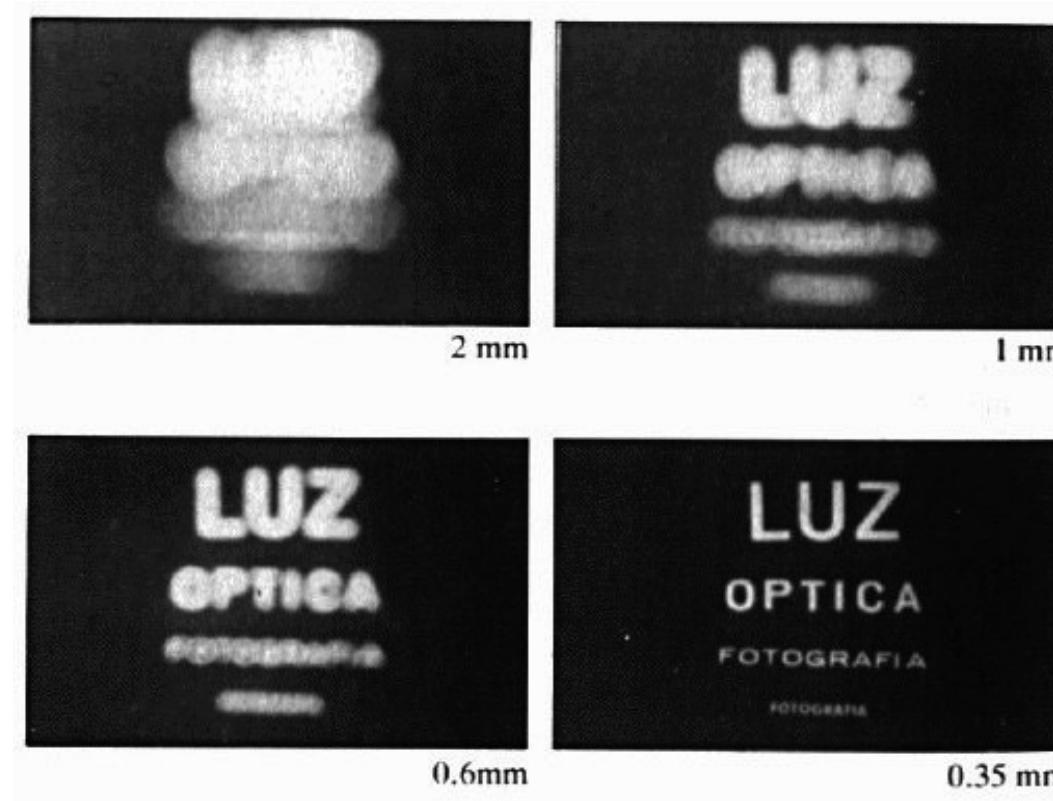
- Idea 1: put a piece of film in front of an object
- Will we get an image?

Pinhole camera

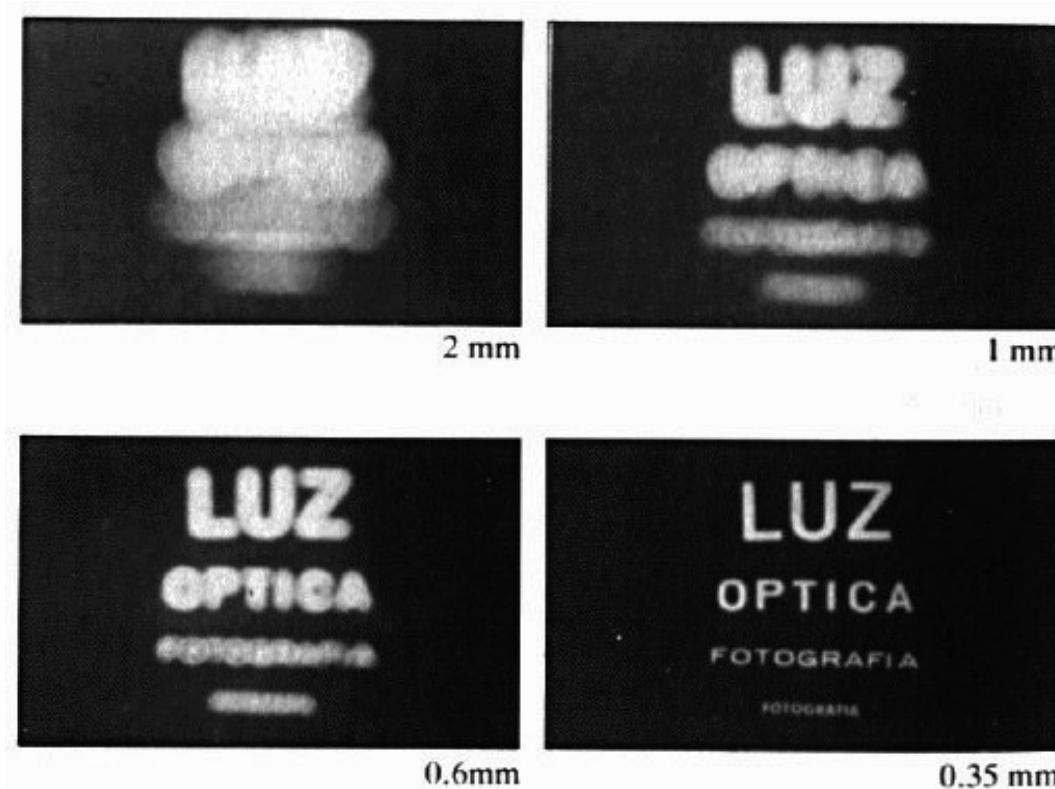


- Add a barrier to block off most of the rays
 - This reduces blurry effect
 - The opening is known as the *aperture*

Shrinking the aperture

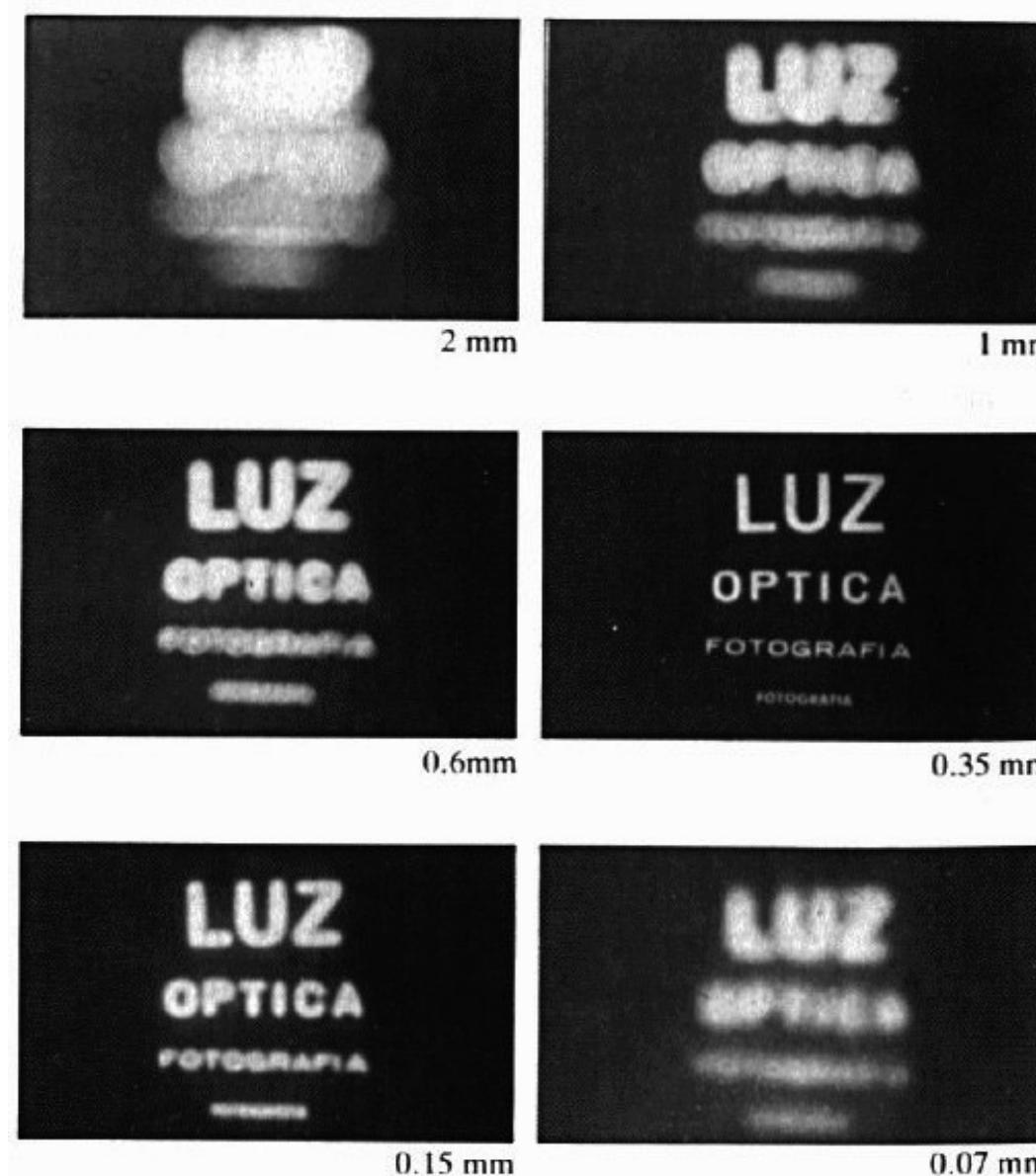


Shrinking the aperture

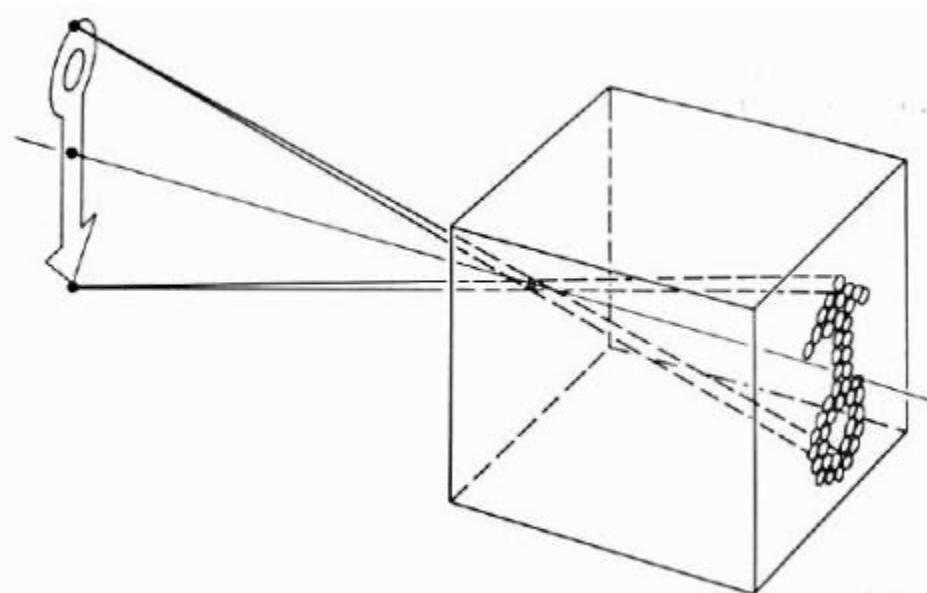


- Why not make the aperture as small as possible?
 - Less light gets through
 - Diffraction effects...

Shrinking the aperture

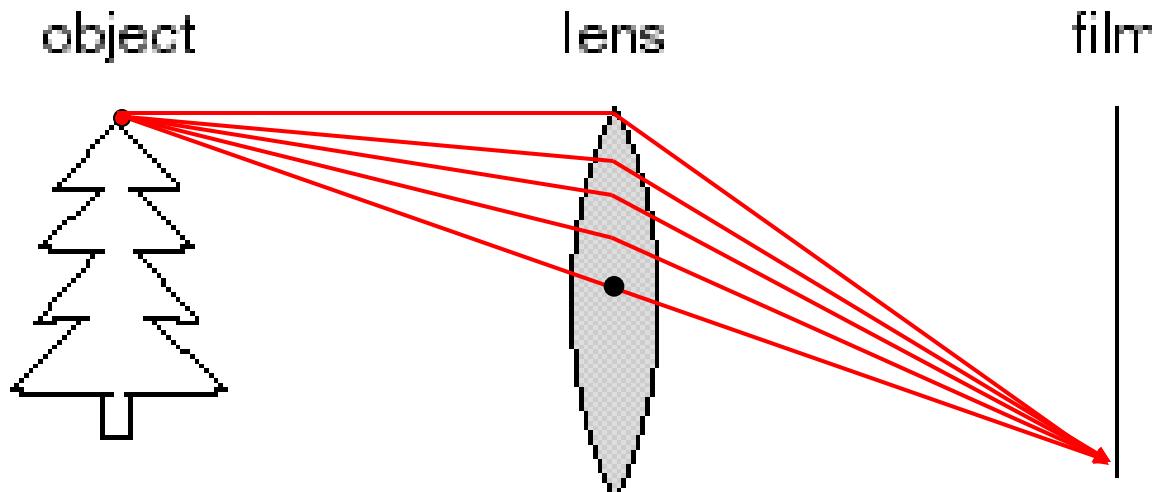


Pinhole camera model



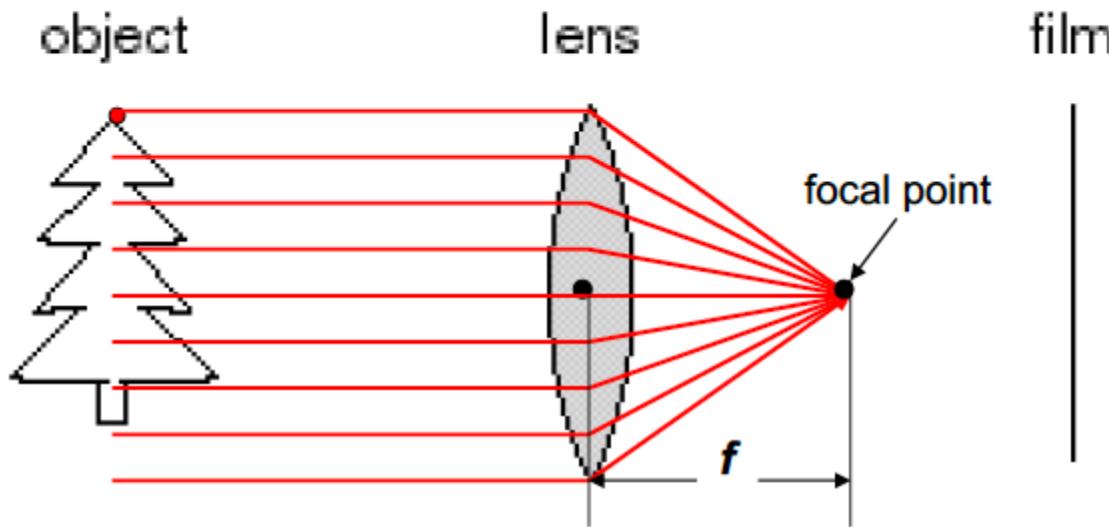
- Pinhole model:
 - Captures ***pencil of rays*** – all rays through a single point
 - This point is called ***Center of Projection (focal point)***
 - The image is formed on the ***Image Plane***

Adding a lens... to capture more light



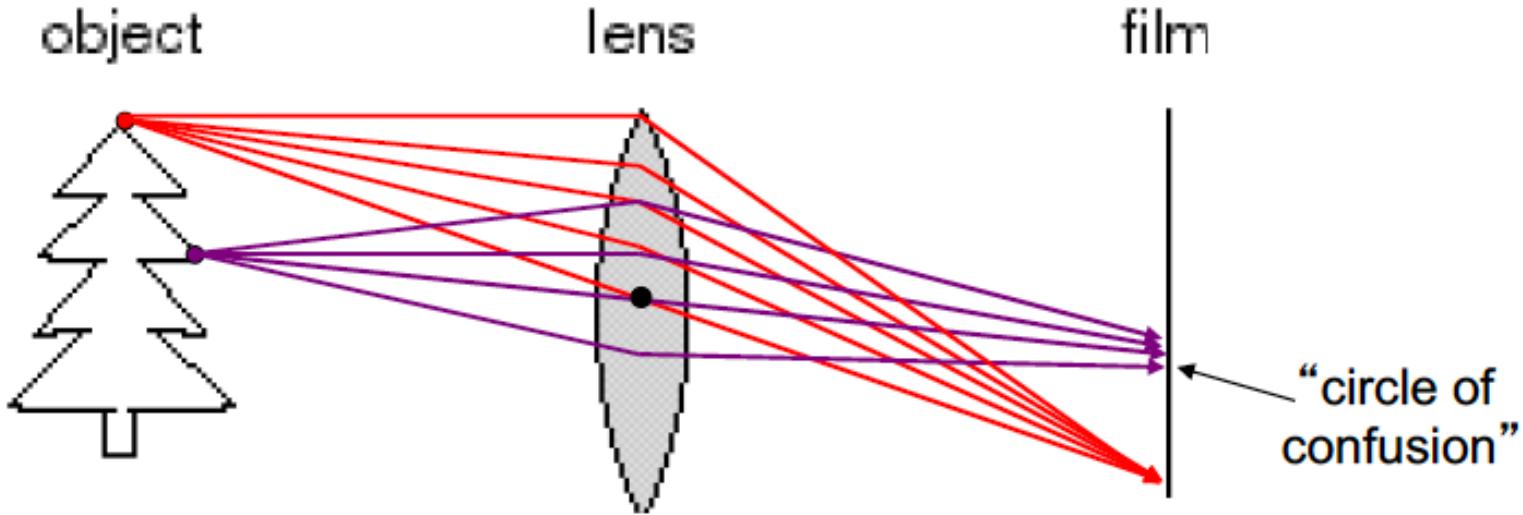
- A lens focuses light onto the film
 - Rays passing through the center are not deviated

Adding a lens... to capture more light



- A lens focuses light onto the film
 - Rays passing through the center are not deviated
 - All parallel rays converge to one point on a plane located at the focal length f

Adding a Lens

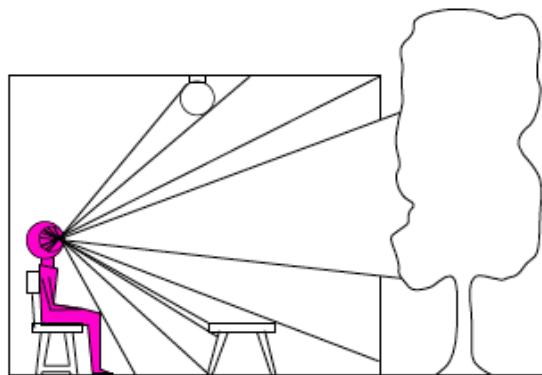


- A lens focuses light onto the film
 - There is a specific distance at which objects are “in focus”
 - other points project to a “circle of confusion” in the image

Geometric Image Formation: Pinhole Camera Model

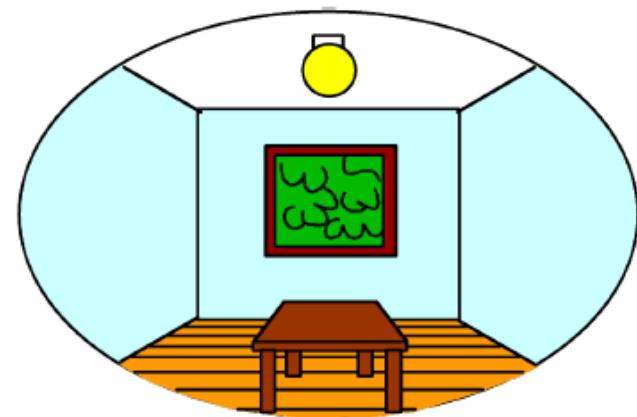
(brief introduction, details to be covered in Multiple-view geometry,
Week-7)

3D world



Point of observation

2D image



What have we lost?

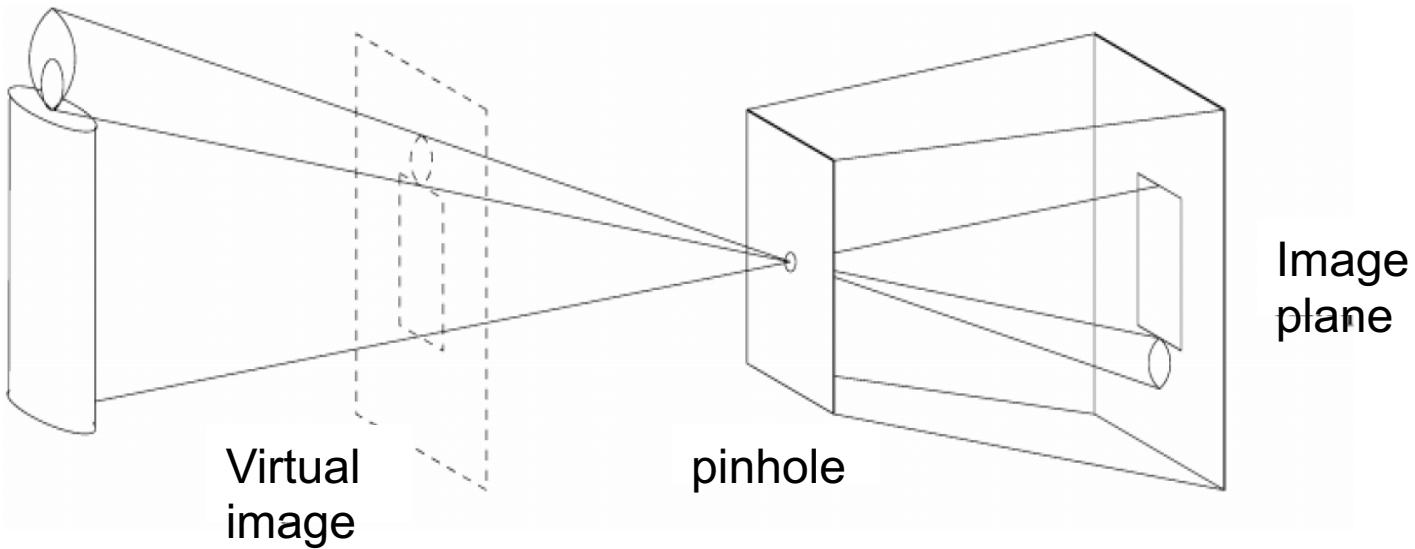
- Angles
- Distances (lengths)

Slide by A. Efros

Figures © Stephen E. Palmer, 2002

Pinhole camera

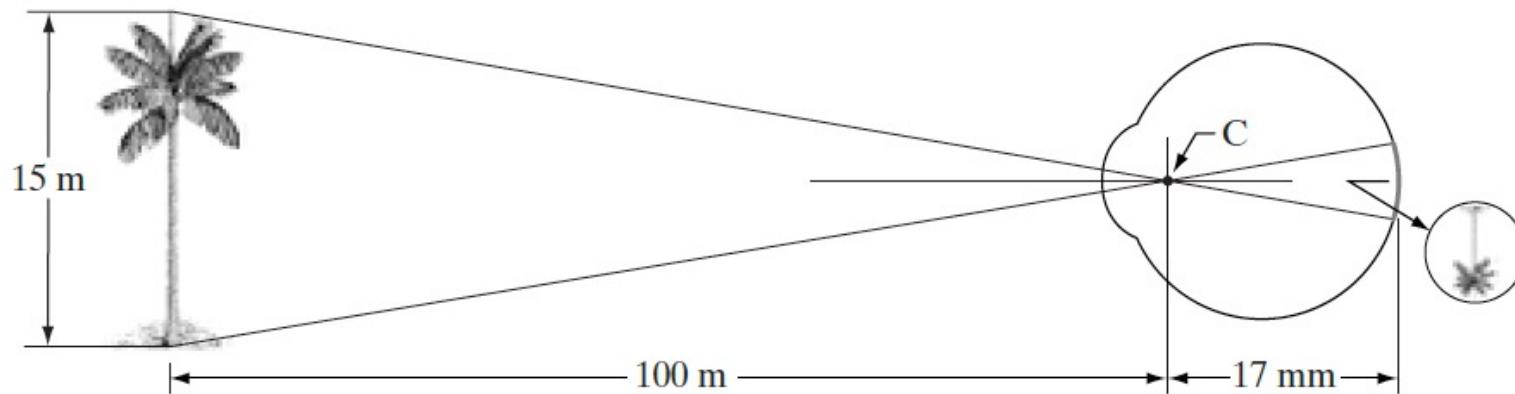
- Pinhole camera is a simple model to approximate imaging process: **perspective projection**.



- If we treat pinhole as a point, only one ray from any given point can enter the camera.

Image Formation in the Eye

FIGURE 2.3
Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.



- (reading pp37-38, Image formation in the eye)
- Book: Digital Image Processing, Second Edition

Linear Algebra

$$\textcircled{1} \quad P = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad P \in \mathbb{R}^3$$

$$\textcircled{2} \quad P = \begin{pmatrix} u \\ v \end{pmatrix} \quad P \in \mathbb{R}^2$$

\textcircled{3} P and p are vectors.

\textcircled{4} I : identity matrix

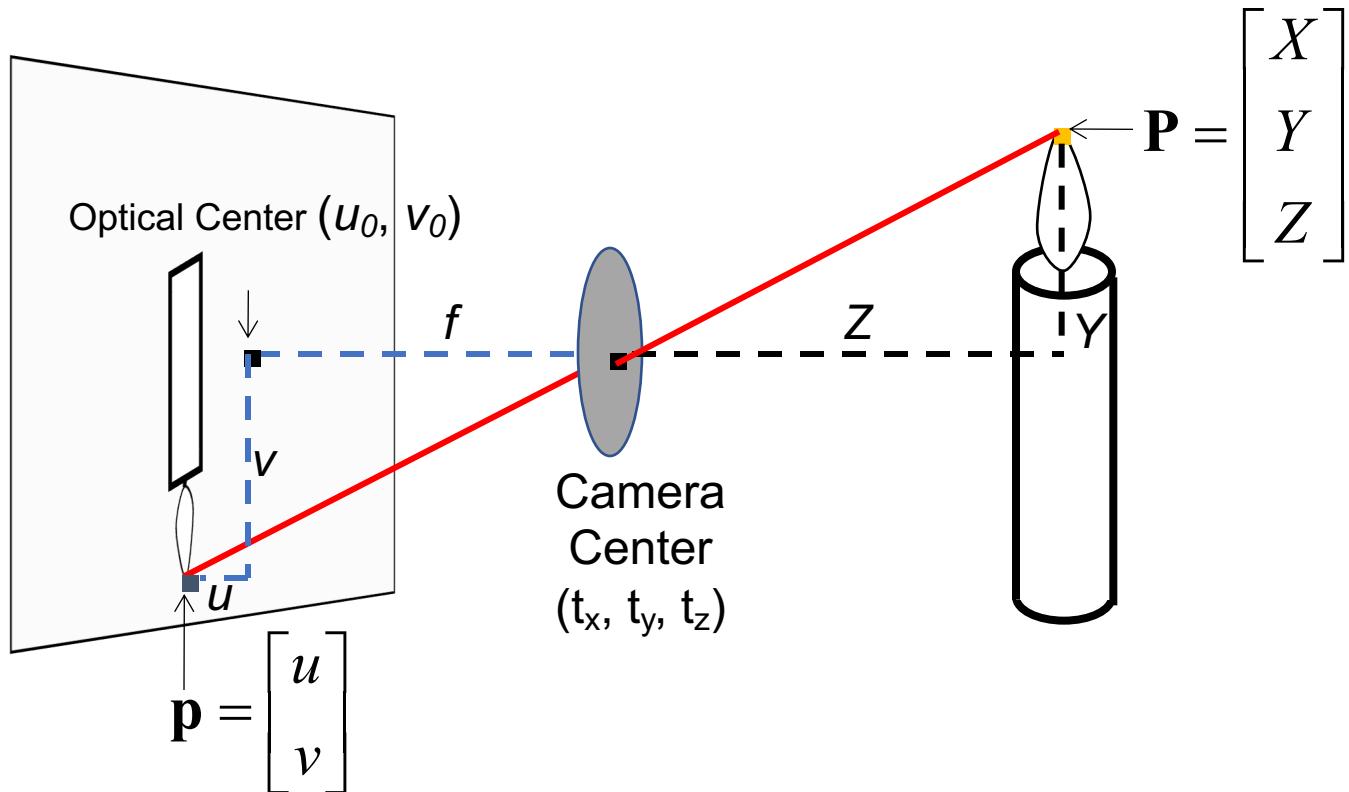
$$\begin{bmatrix} I & 0 \end{bmatrix}$$

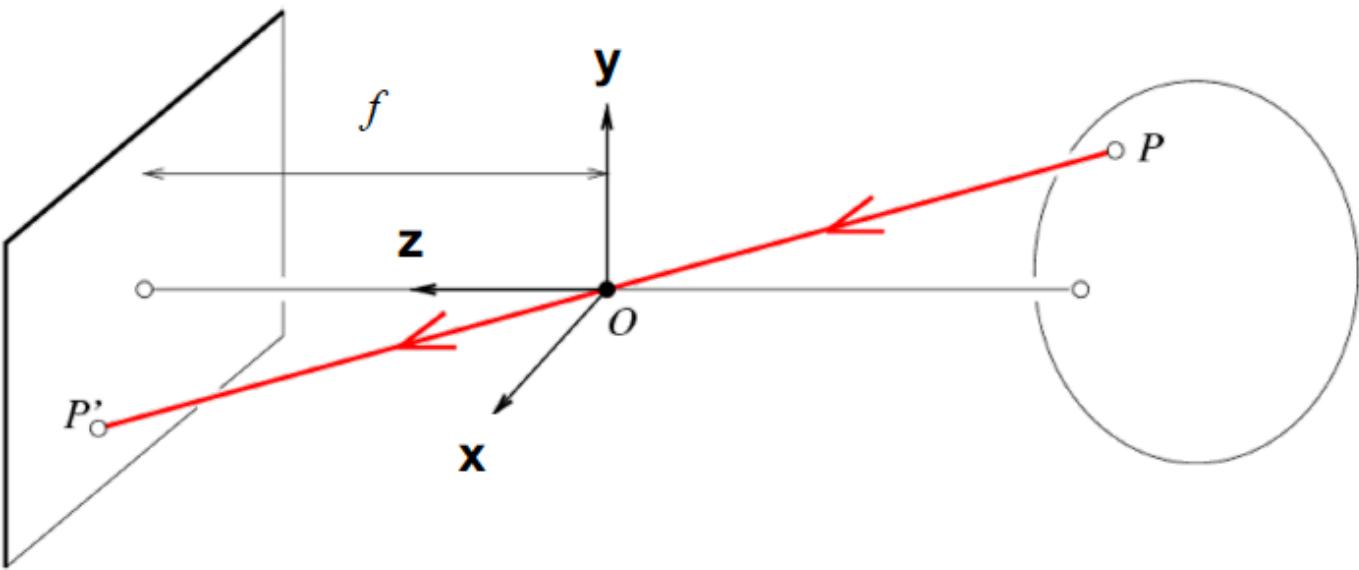
Assume $I \in \mathbb{R}^{3 \times 3}$

$$0 \in \mathbb{R}^{3 \times 1}$$

then: $\begin{bmatrix} I & 0 \end{bmatrix}_{3 \times 4}$

Projection: world coordinates → image coordinates





- Projection equations

- Compute intersection with image plane of ray from $P = (x, y, z)$ to O
 - Derived using similar triangles

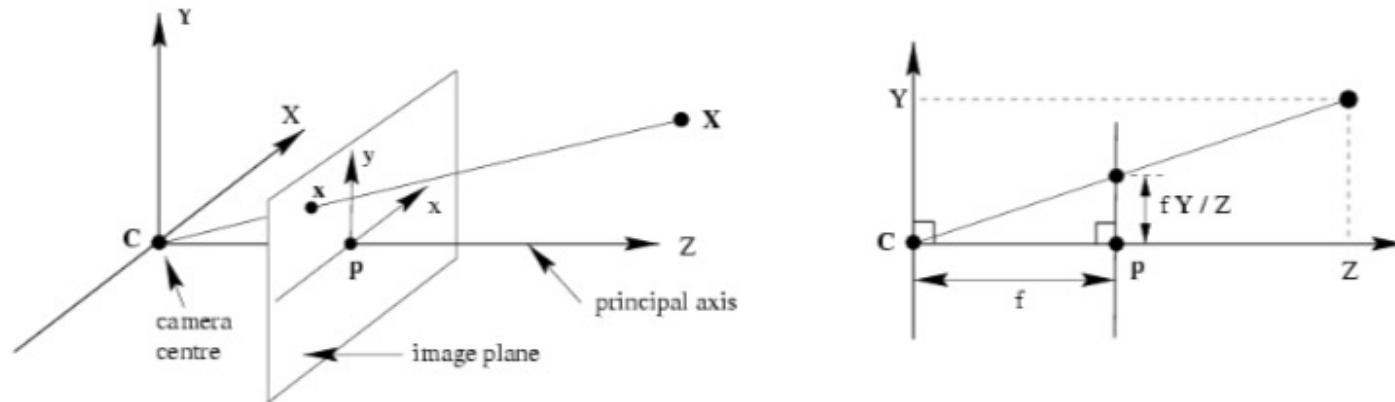
$$(x, y, z) \rightarrow \left(f \frac{x}{z}, f \frac{y}{z}, f \right)$$

- We get the projection by throwing out the last coordinate:

$$(x, y, z) \rightarrow \left(f \frac{x}{z}, f \frac{y}{z} \right)$$

Source: J. Ponce, S. Seitz

Pinhole camera model



- By similar triangles

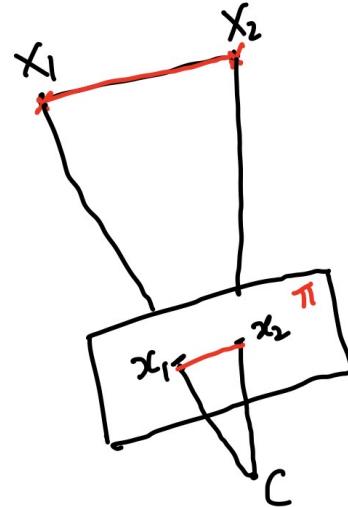
$$(X, Y, Z)^T \mapsto (fX/Z, fY/Z, f)^T$$

- Dropping third coordinate

$$(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T$$

Explanation of scale and size of the object in the image

- Object far away from the camera is projected as a smaller object in the image
- Object close to the camera is projected as larger size in the image



$$x_1 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \quad x_2 = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}$$

Assume: $z_1 = z_2 = z$, camera focal length f

$$\text{then: } x_1 = \begin{pmatrix} u_1 \\ v_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} f \frac{x_1}{z} \\ f \frac{y_1}{z} \\ z_1 \end{pmatrix}, \quad x_2 = \begin{pmatrix} u_2 \\ v_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} f \frac{x_2}{z} \\ f \frac{y_2}{z} \\ z_2 \end{pmatrix}$$

$$d_{12} = \begin{pmatrix} f \frac{x_2 - x_1}{z} \\ f \frac{y_2 - y_1}{z} \end{pmatrix}, \quad \|d_{12}\| = \frac{f}{z} \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Homogeneous coordinates Representation

Conversion

- Converting *from* Cartesian *to* homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous **image**
coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous **scene**
coordinates

- Converting *from* homogeneous coordinates *to* cartesian

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

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Homogeneous Coordinates

- Invariant to scaling

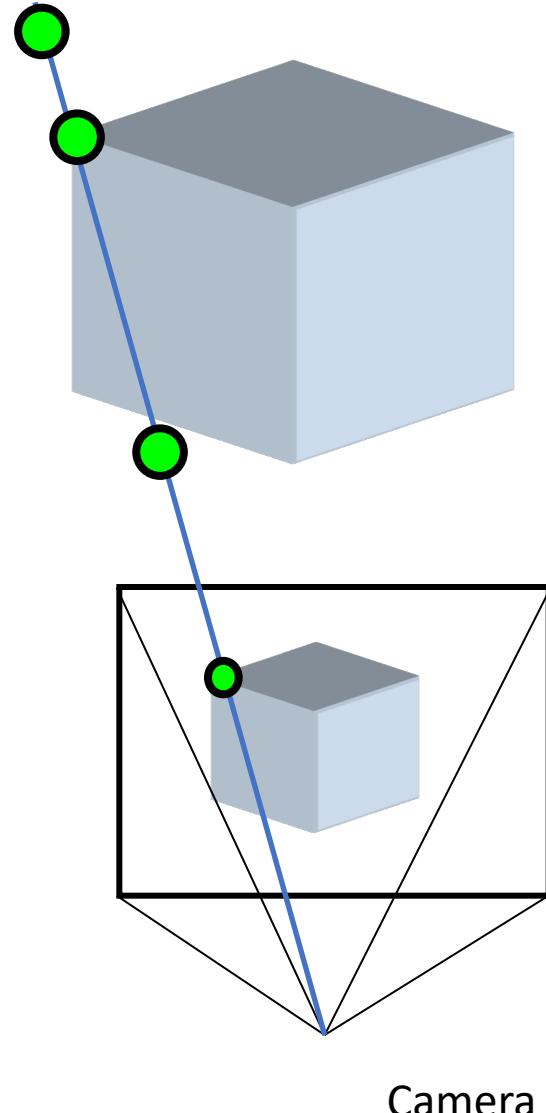
$$k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

Homogeneous
Coordinates

Cartesian
Coordinates

- Point in **cartesian** is a ray in **homogeneous**

Example



$$(X, Y, Z)^T \rightarrow (fX/Z, fY/Z)$$

- Cartesian coordinate represented by homogenous coordinate

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix}$$

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Perspective effects



3D computer vision aims to solve an inverse problem of computer graphics

Photometric (radiometric) image formation

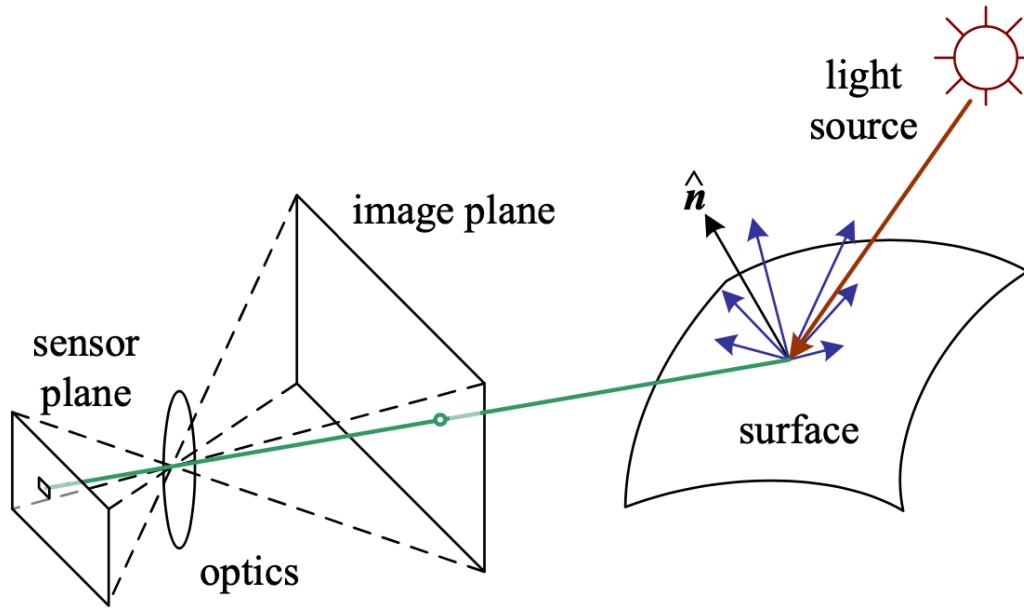
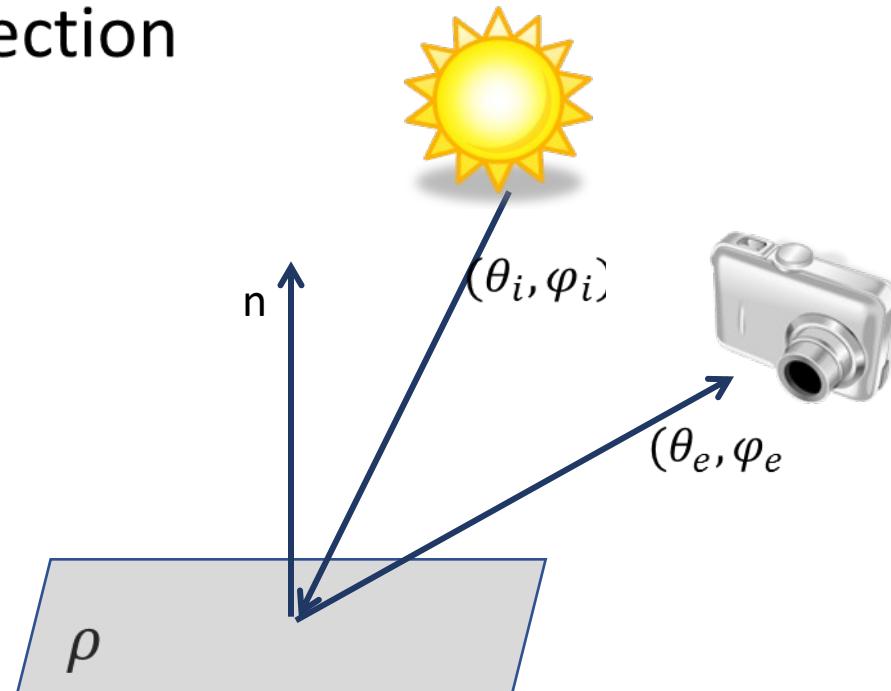


Figure 2.14 A simplified model of photometric image formation. Light is emitted by one or more light sources and is then reflected from an object's surface. A portion of this light is directed towards the camera. This simplified model ignores multiple reflections, which often occur in real-world scenes.

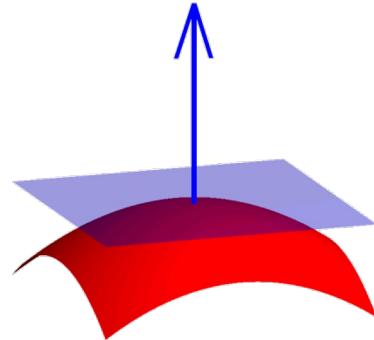
What determines the scene radiance?

- The amount of light that falls on the surface
- The fraction of light that is reflected (albedo)
- Geometry of light reflection
 - ▣ Shape of surface
 - ▣ Viewpoint

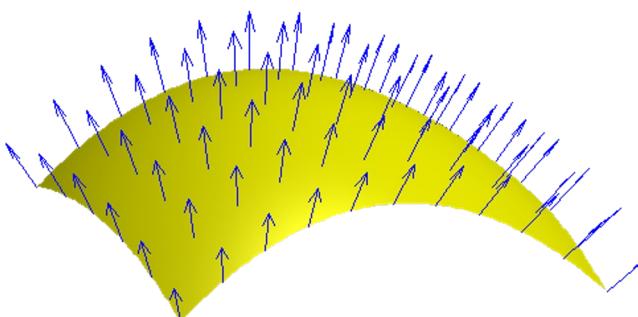


Surface Normal

- ❖ Convenient notation for surface orientation
- ❖ A smooth surface has a tangent plane at every point

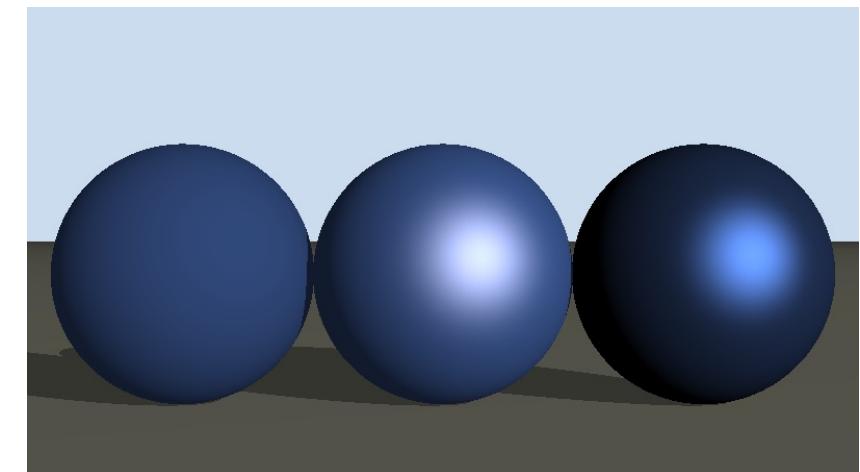
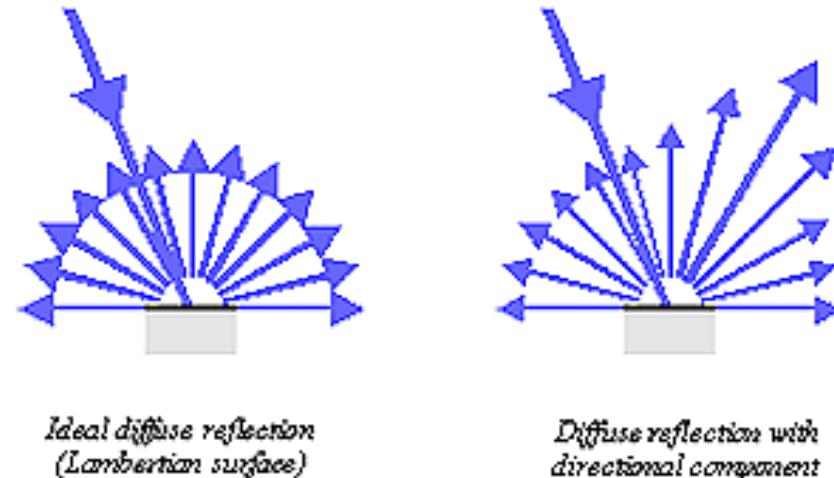


- ❖ We can model the surface using the normal at every point



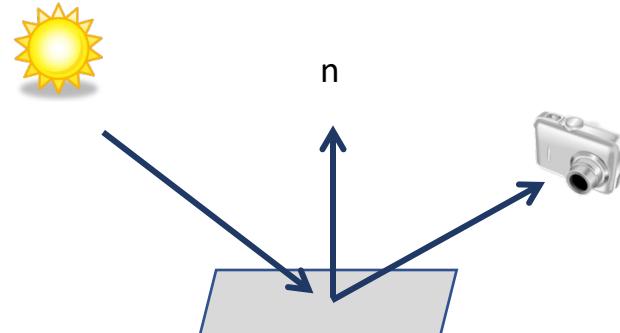
Lambertian surface

- Ideal diffuse reflectors.
- The apparent brightness of such a surface to an observer is the same regardless of the observer's angle of view.

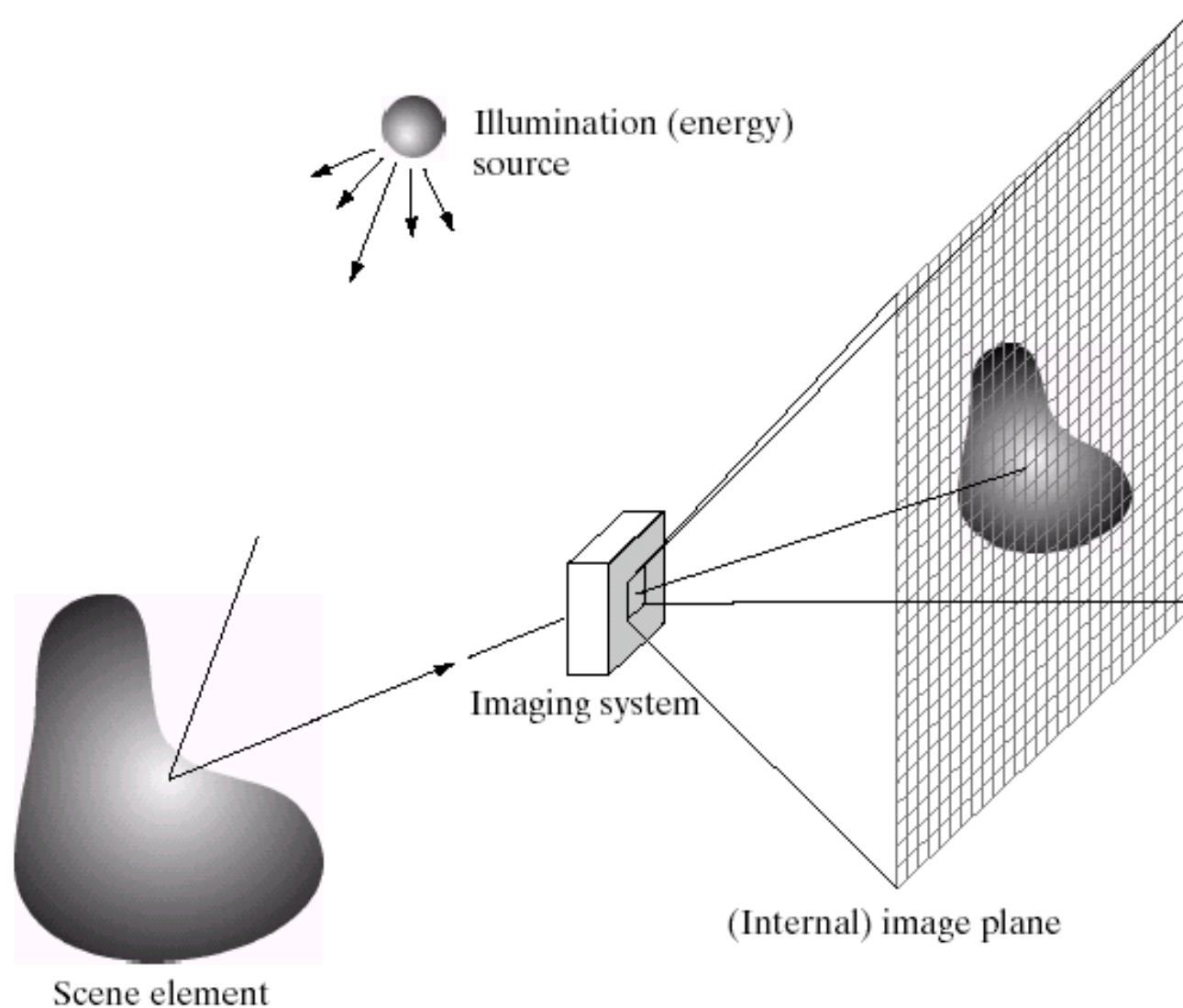


Lambertian Surface

- Appears equally bright from all viewing directions
- Reflects all light without absorbing
- Matte surface, no “shiny” spots
- Brightness of the surface as seen from camera is linearly correlated to the amount of light falling on the surface



Photometric Image Formation



Sensor Array

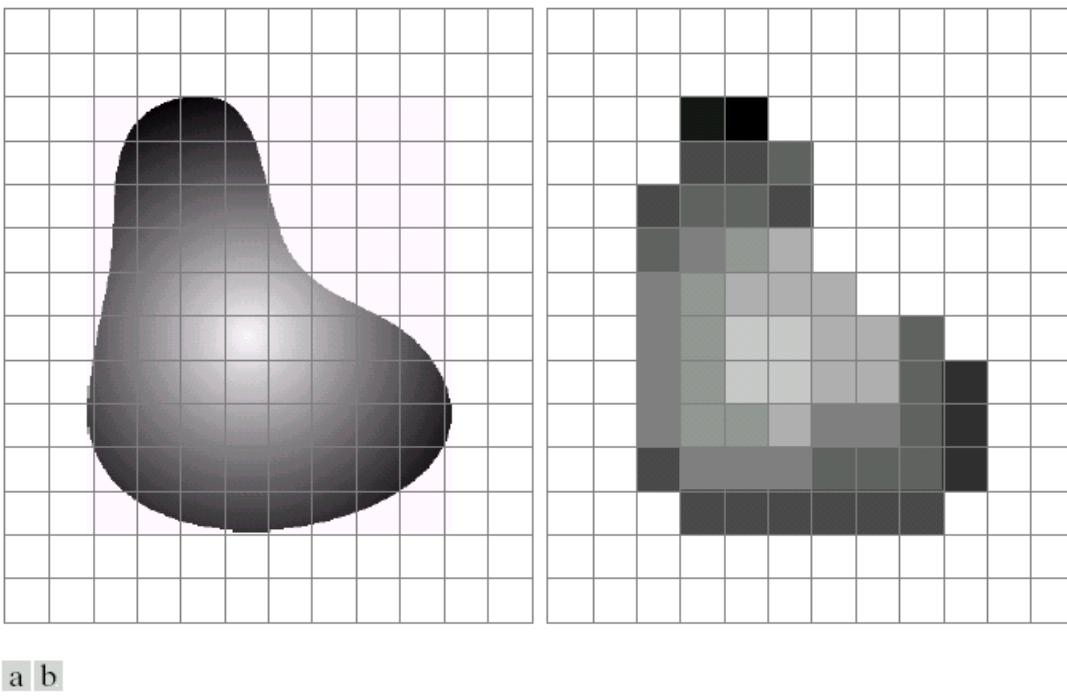
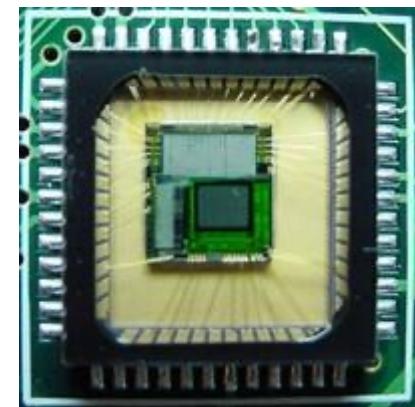
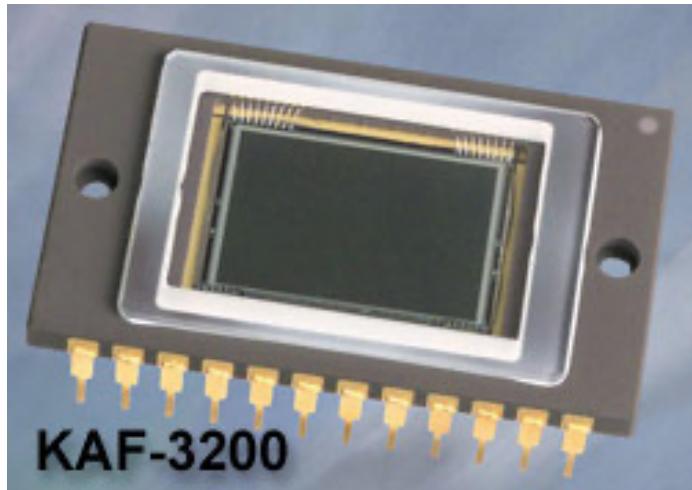


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.



CCD/CMOS sensor

Image Sensors : Array Sensor

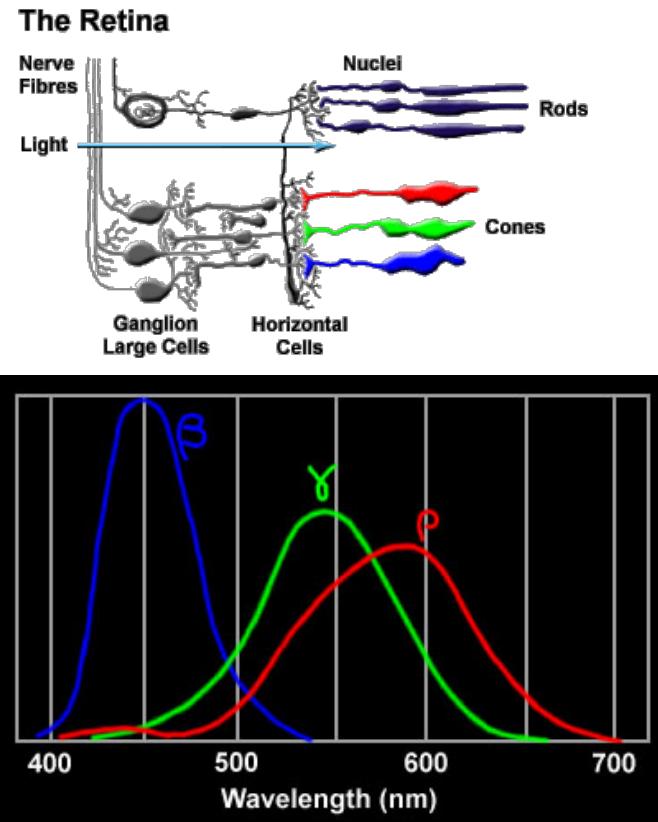
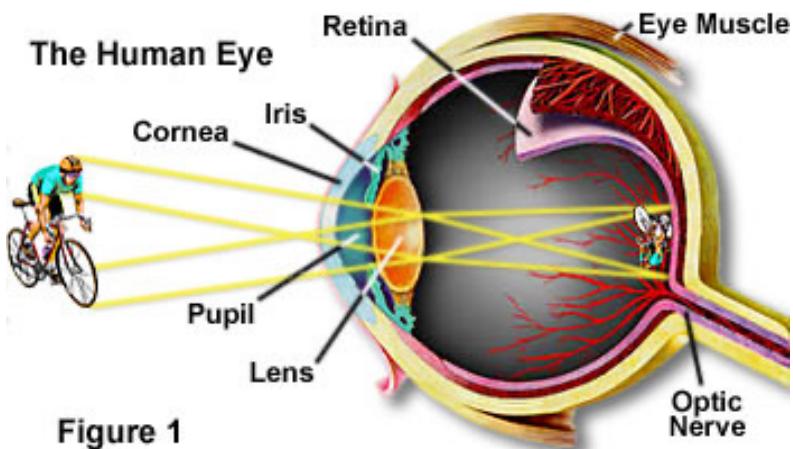


Charge-Coupled Device (CCD)

- ◆ Used for convert a continuous image into a digital image
- ◆ Contains an array of light sensors
- ◆ Converts photon into electric charges accumulated in each sensor unit

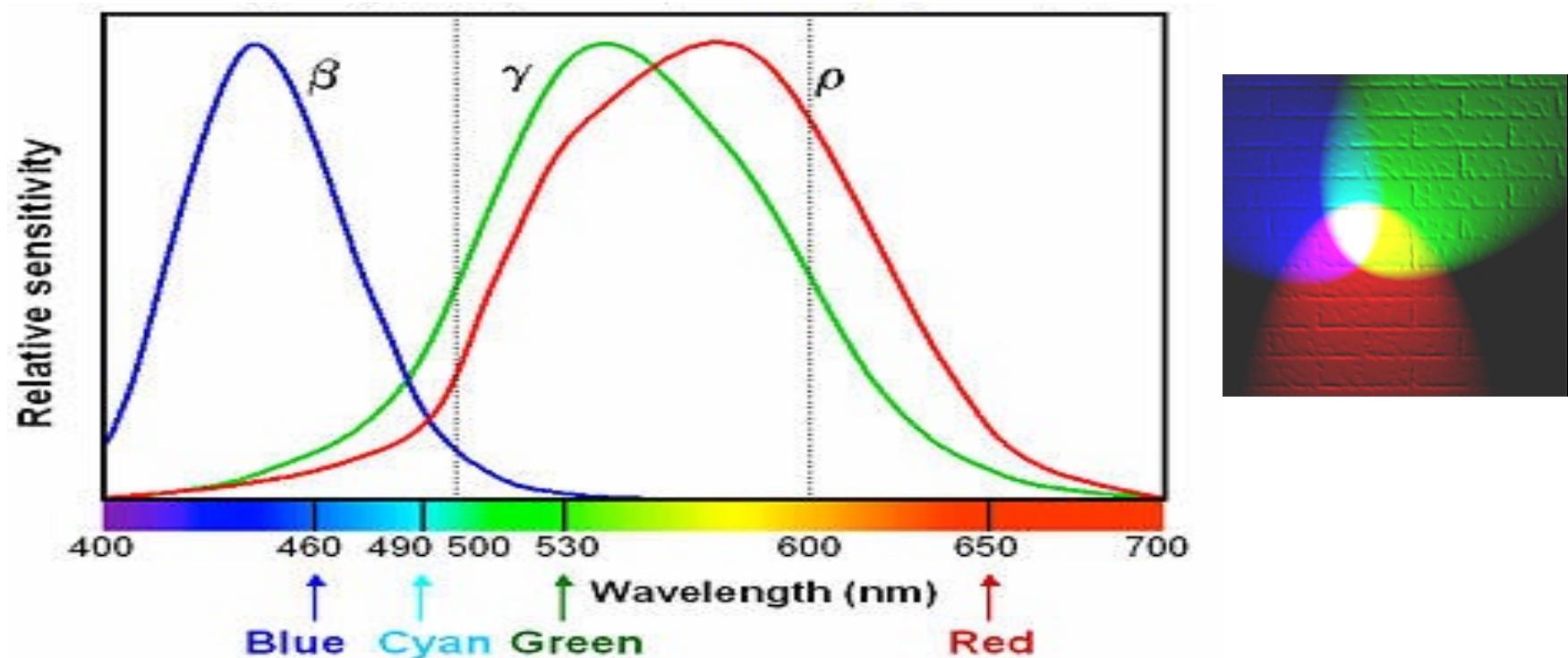
CCD KAF-3200E from Kodak.
(2184 x 1472 pixels, Pixel size
6.8 microns²)

Human Color Perception



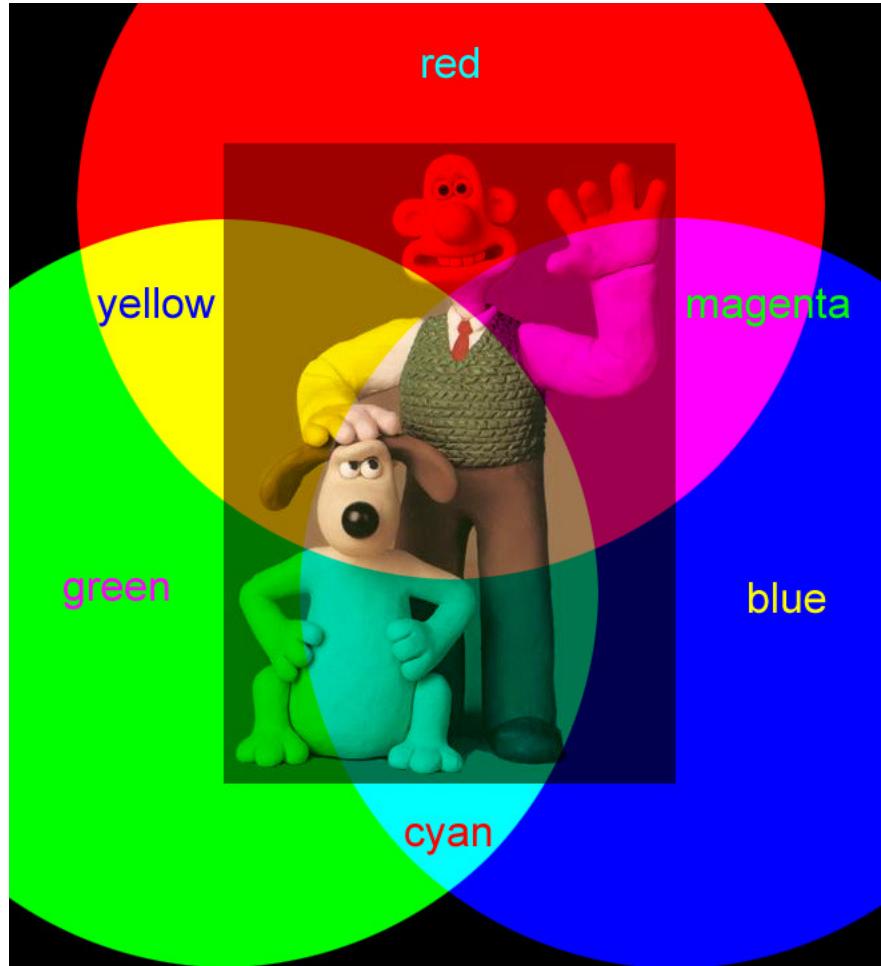
https://en.wikipedia.org/wiki/Color_vision

- Wavelength of the light



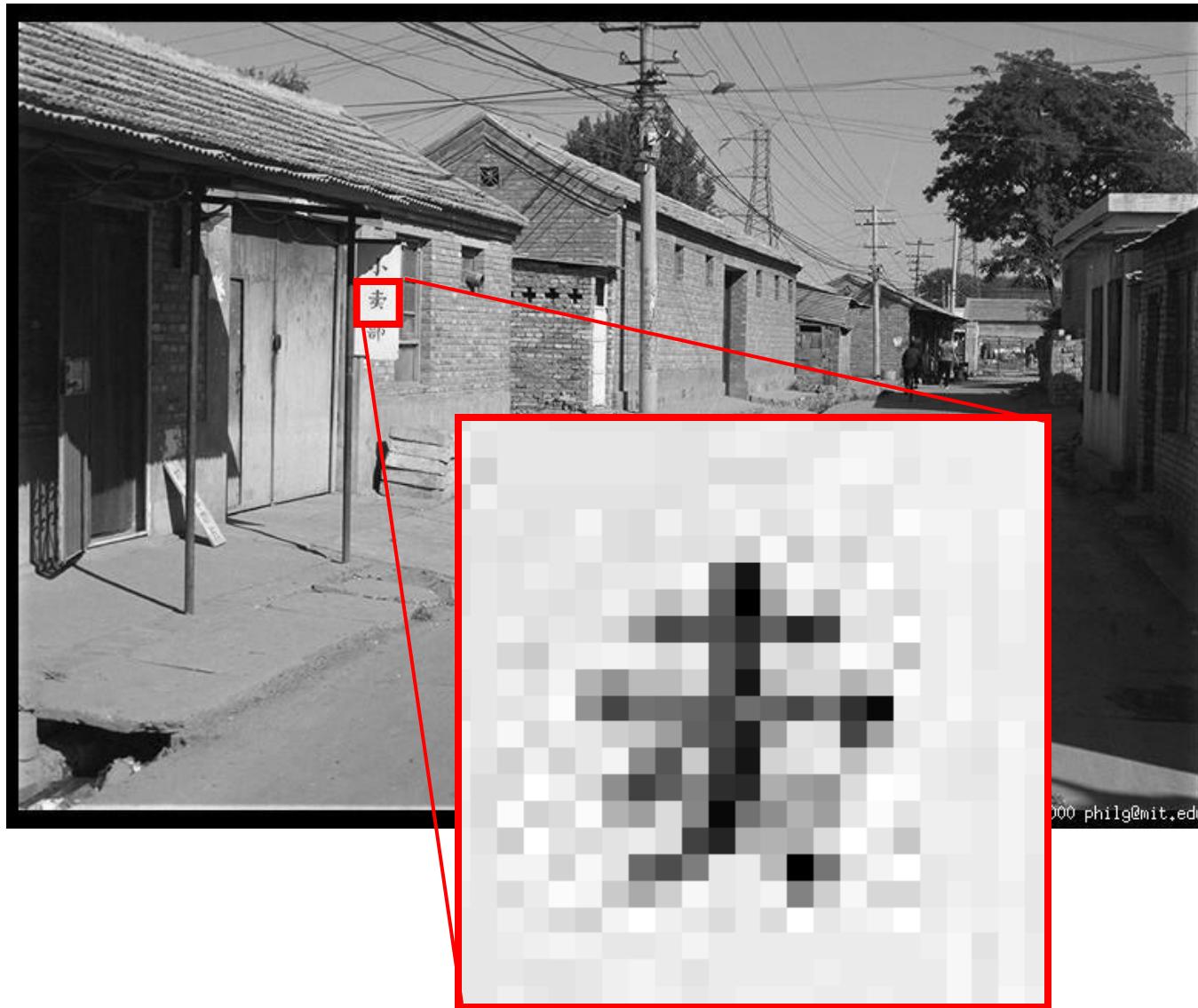
Colour Images

- are constructed from three intensity maps.
- Each intensity map is projected through a colour filter (e.g., red, green, or blue, or cyan, magenta, or yellow) to create a monochrome image.
- The intensity maps are overlaid to create a color image.
- Each pixel in a color image is a three-dimensional vector.

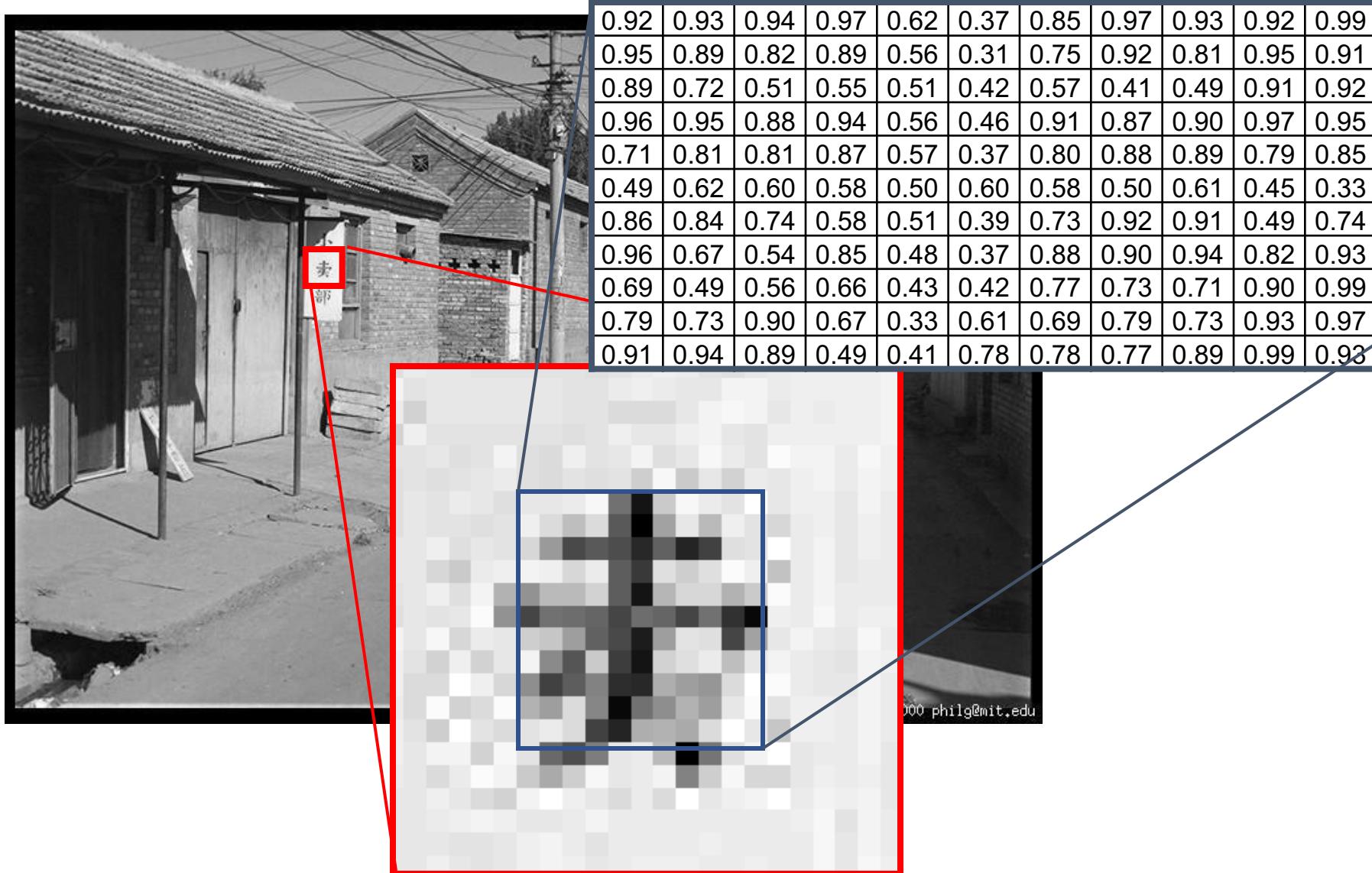


Digital image representation

The digital image (pixel matrix)



The digital image (pixel matrix)

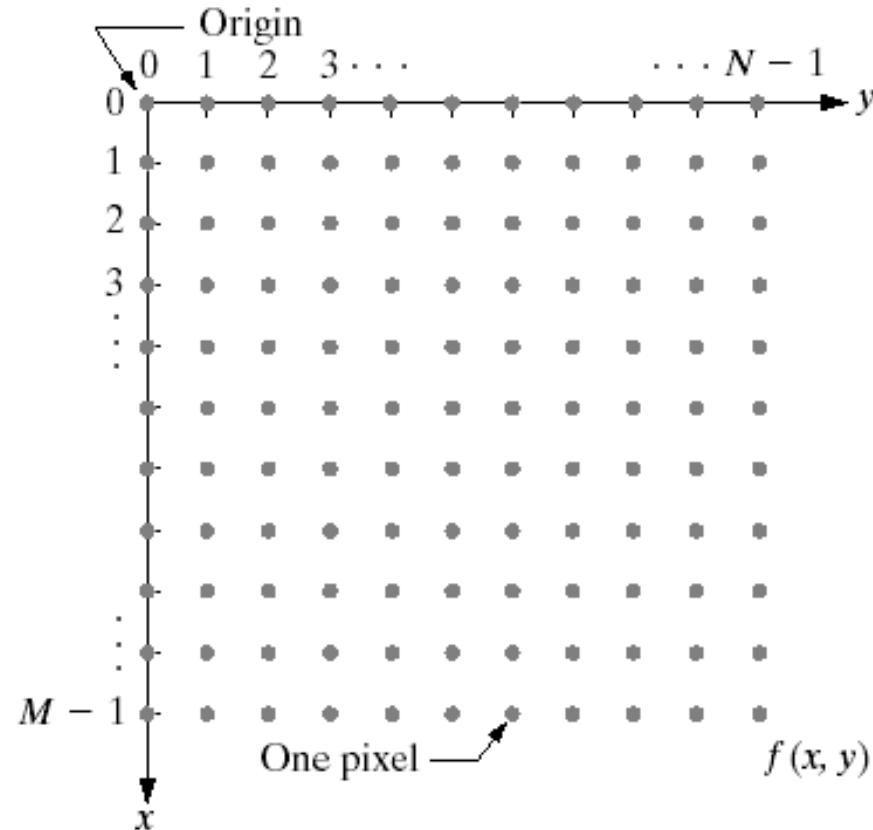


Fundamentals of Digital Images



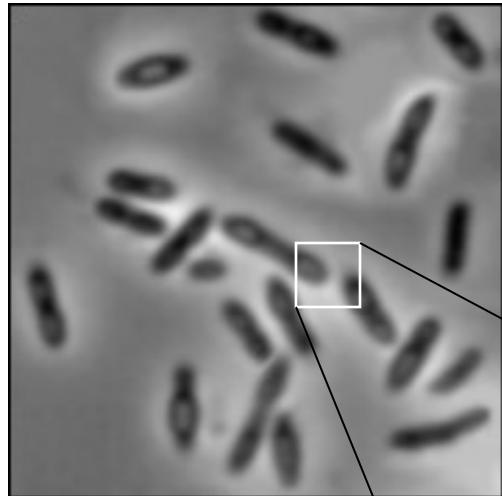
- ◆ An image = a function of spatial coordinates.
- ◆ Spatial coordinate: (x,y) for 2D case such as photograph,
 (x,y,z) for 3D case such as CT scan images
 (x,y,t) for video.
- ◆ The function f may represent the intensity (for greyscale images)
or color (for color images) or other associated values.

Conventional Coordinate for Image Representation



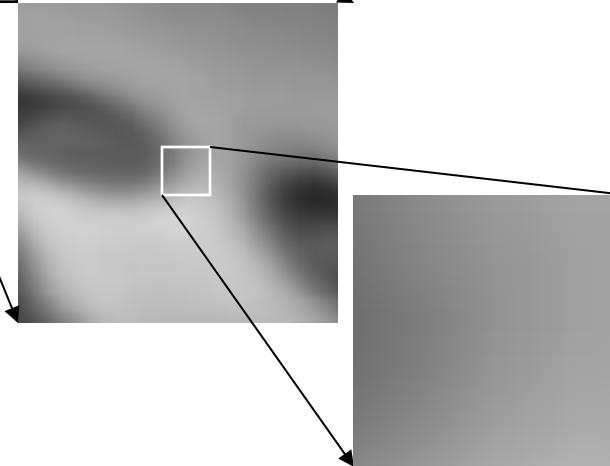
(Images from Rafael C. Gonzalez and Richard E.
Wood, Digital Image Processing, 2nd Edition.

Grey-scale Image



Intensity image or grey-scale image

- Each pixel corresponds to light intensity normally represented in gray scale (gray level).



Gray scale values

10	10	16	28
9	6	26	37
15	25	13	22
32	15	87	39

Effect of Spatial Resolution



FIGURE 2.19 A 1024×1024 , 8-bit image subsampled down to size 32×32 pixels. The number of allowable gray levels was kept at 256.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Effect of Spatial Resolution

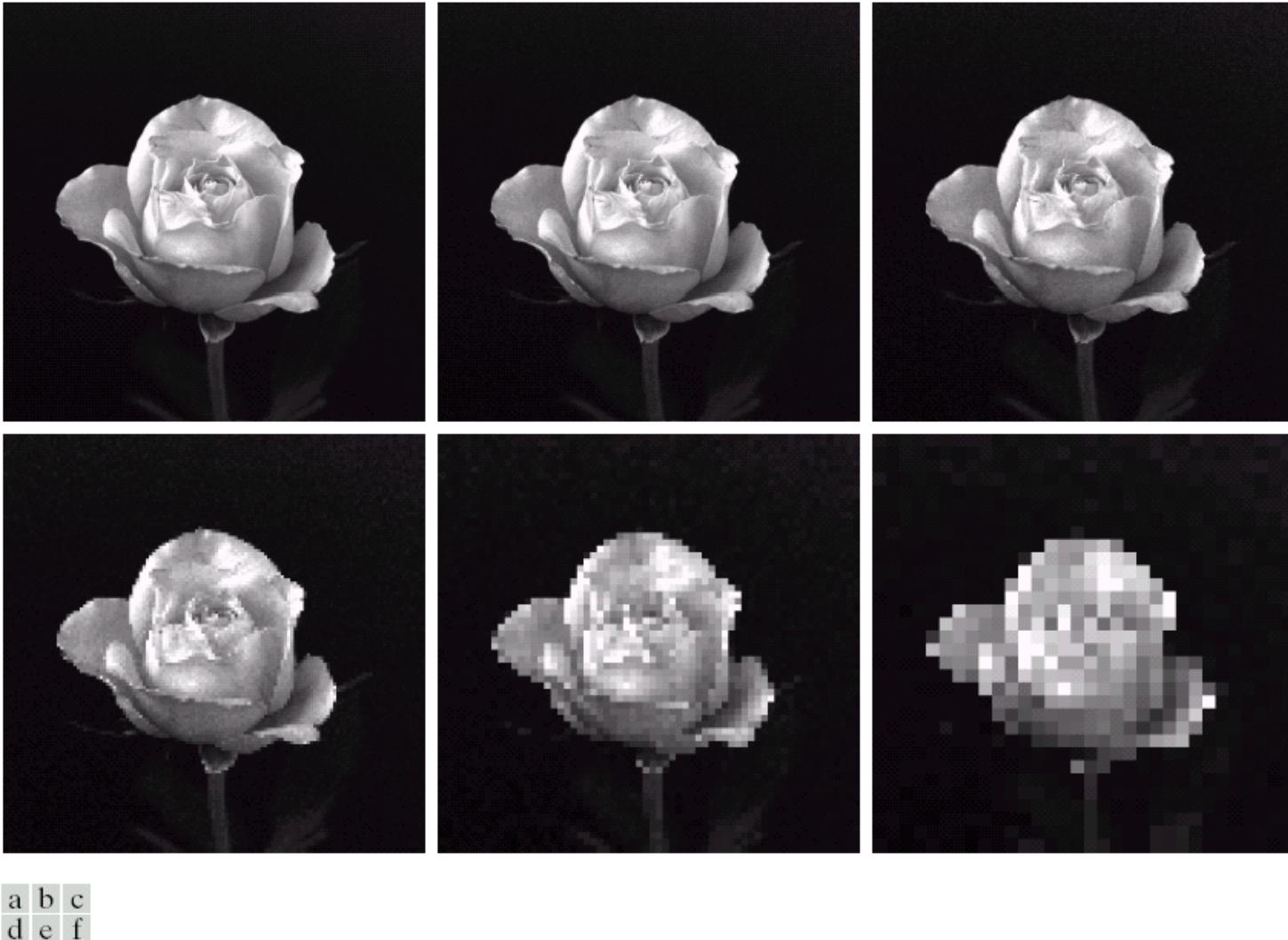
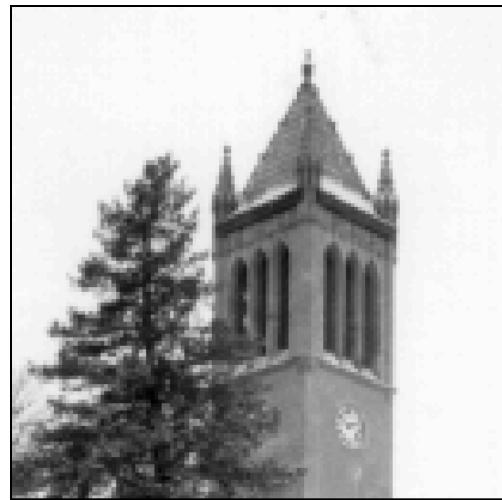


FIGURE 2.20 (a) 1024×1024 , 8-bit image. (b) 512×512 image resampled into 1024×1024 pixels by row and column duplication. (c) through (f) 256×256 , 128×128 , 64×64 , and 32×32 images resampled into 1024×1024 pixels.

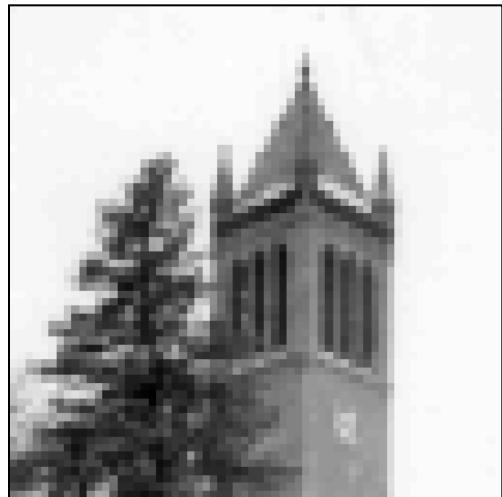
Effect of Spatial Resolution



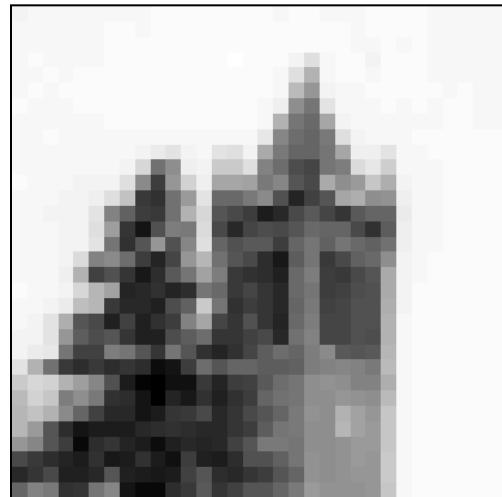
256x256 pixels



128x128 pixels



64x64 pixels

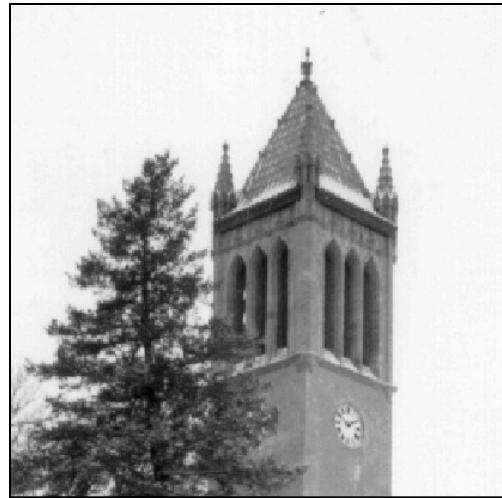


32x32 pixels

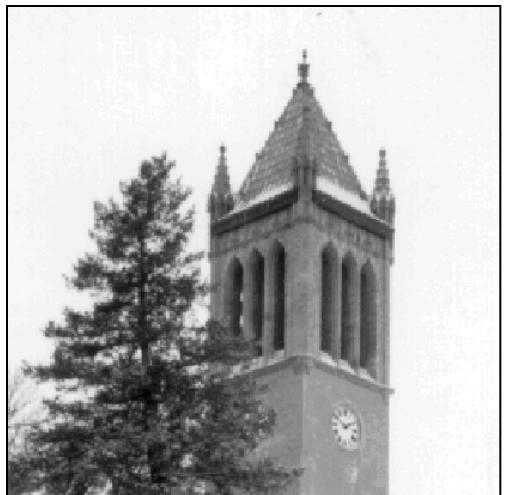
Effect of Quantization Levels



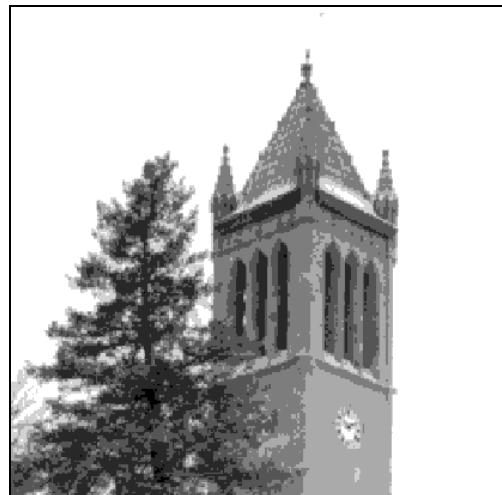
256 levels



128 levels

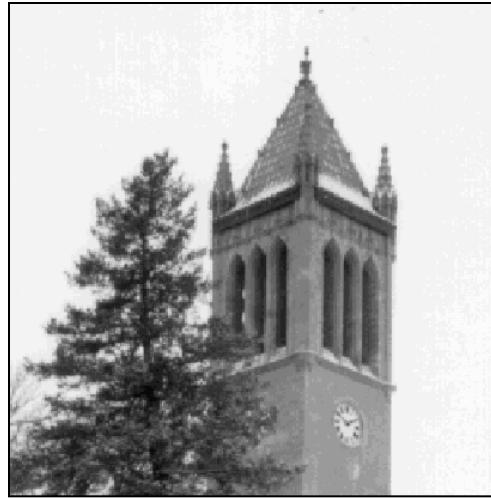


64 levels

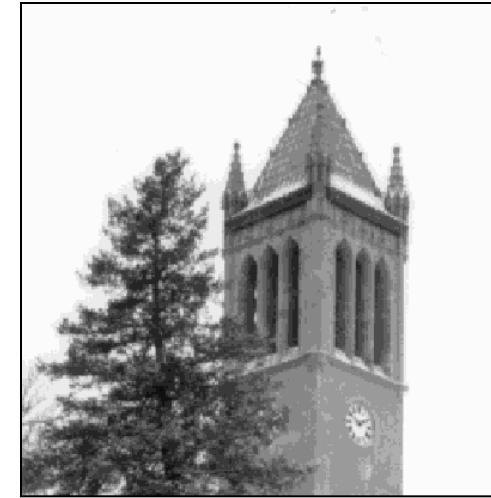


32 levels

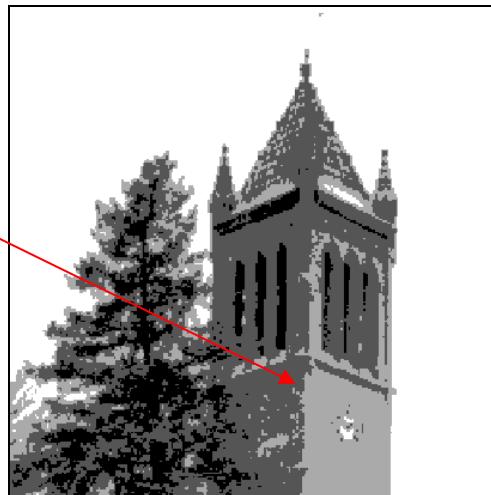
Effect of Quantization Levels (cont.)



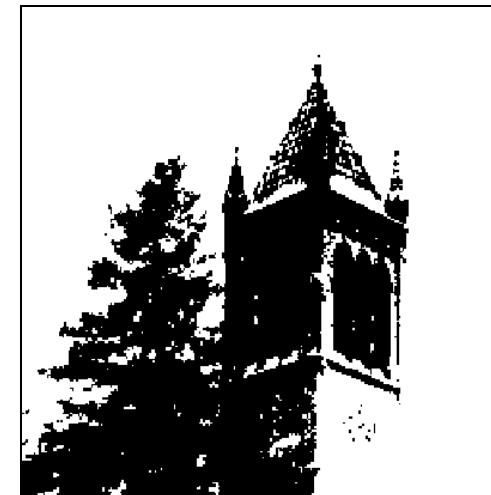
16 levels



8 levels



4 levels



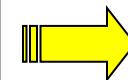
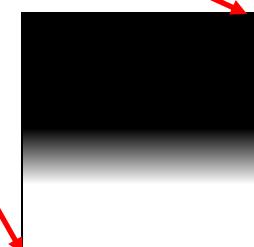
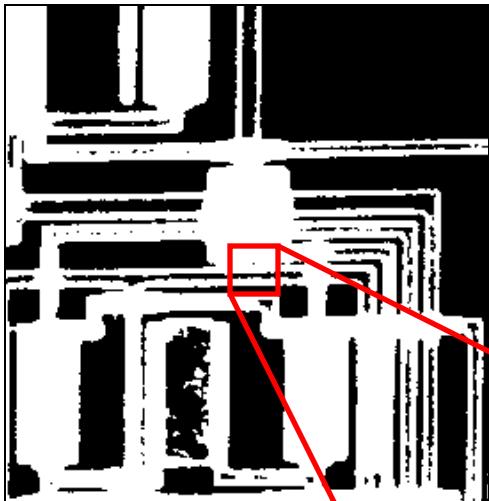
2 levels

In this image,
it is easy to see
false contour.

Two-level image: binary image

Binary image or black and white image

- Each pixel contains one bit :
 - 1 represent white
 - 0 represents black



Binary data

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Colour Images

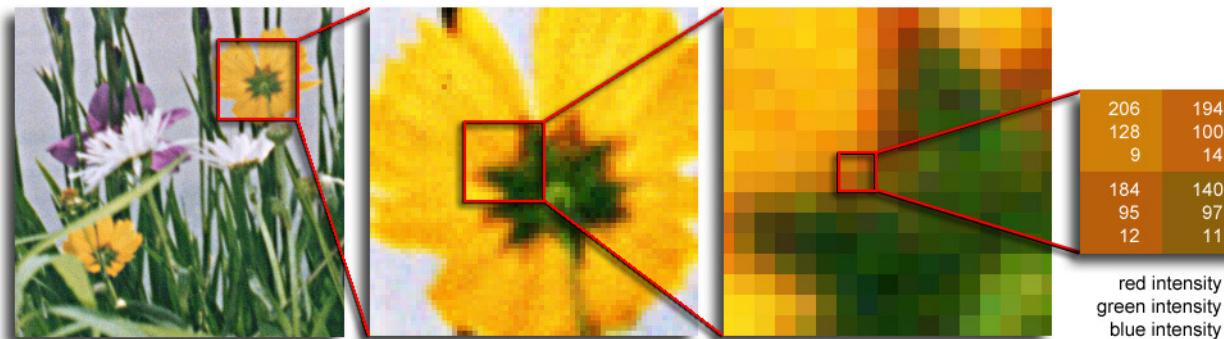
A colourful image



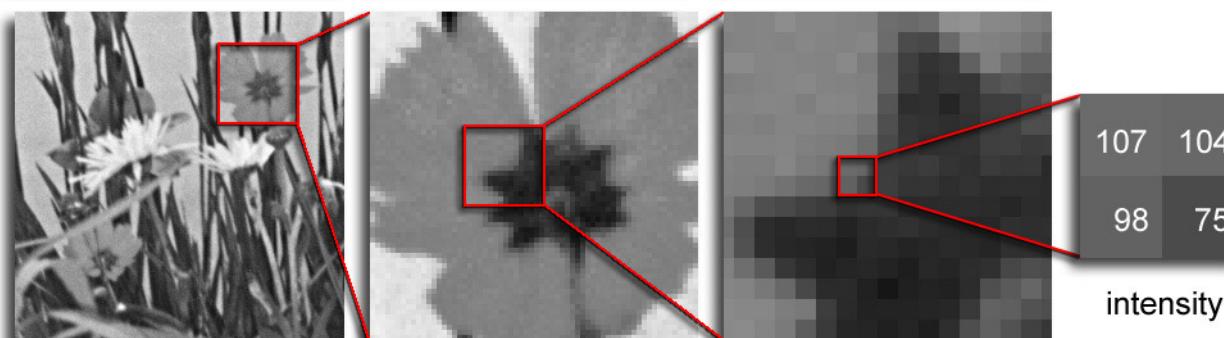
Colour Image

- Colour images have 3 values per pixel;
- Greyscale images have 1 value per pixel.

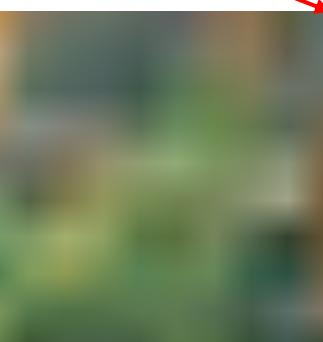
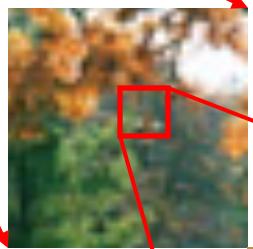
a grid of squares, each of which contains a single colour



each square is called a pixel (for *picture element*)



Colour Image



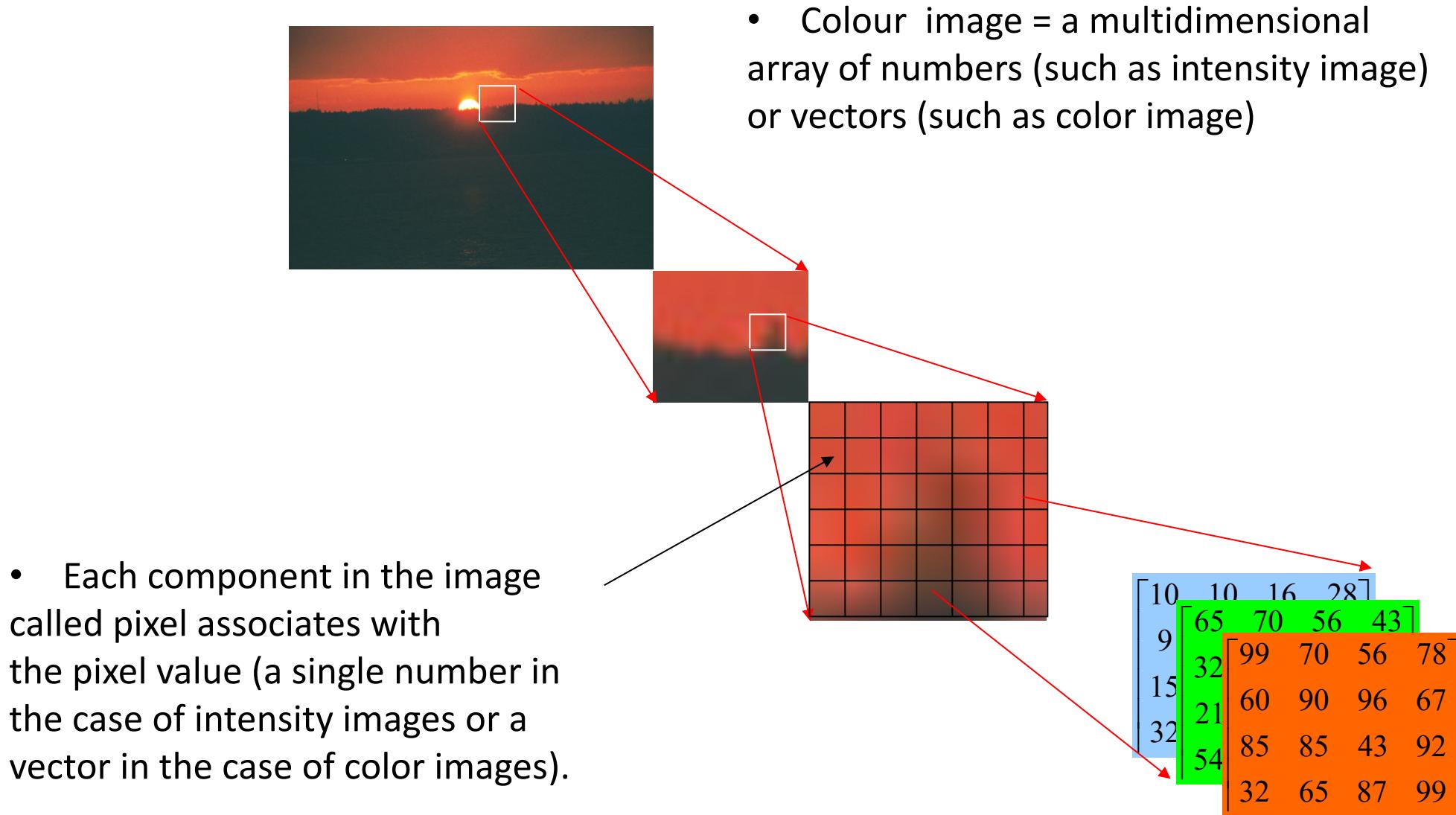
Color (RGB) image:

- each pixel contains a vector representing red, green and blue components.

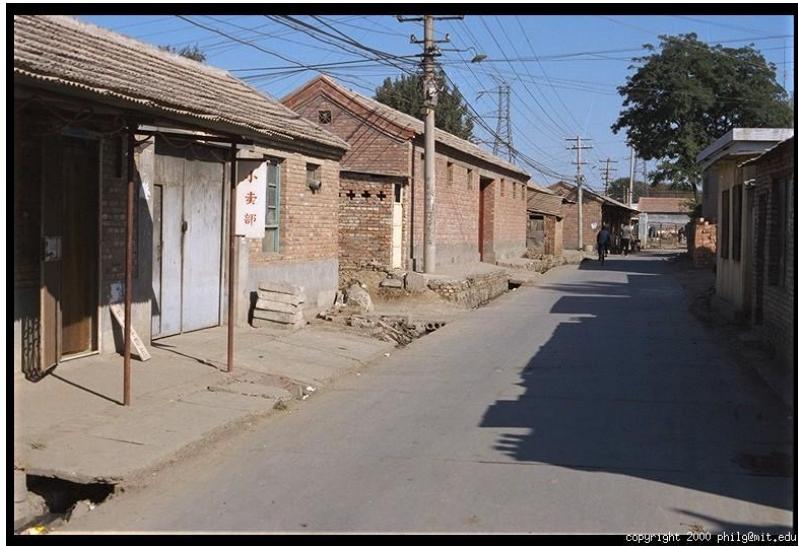
RGB components

$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 65 & 70 & 56 & 43 \\ 32 & 99 & 70 & 56 & 78 \\ 15 & 60 & 90 & 96 & 67 \\ 32 & 21 & 85 & 85 & 43 & 92 \\ 54 & 85 & 85 & 43 & 92 \\ 32 & 32 & 65 & 87 & 99 \end{bmatrix}$$

Colour Image



Colour Image

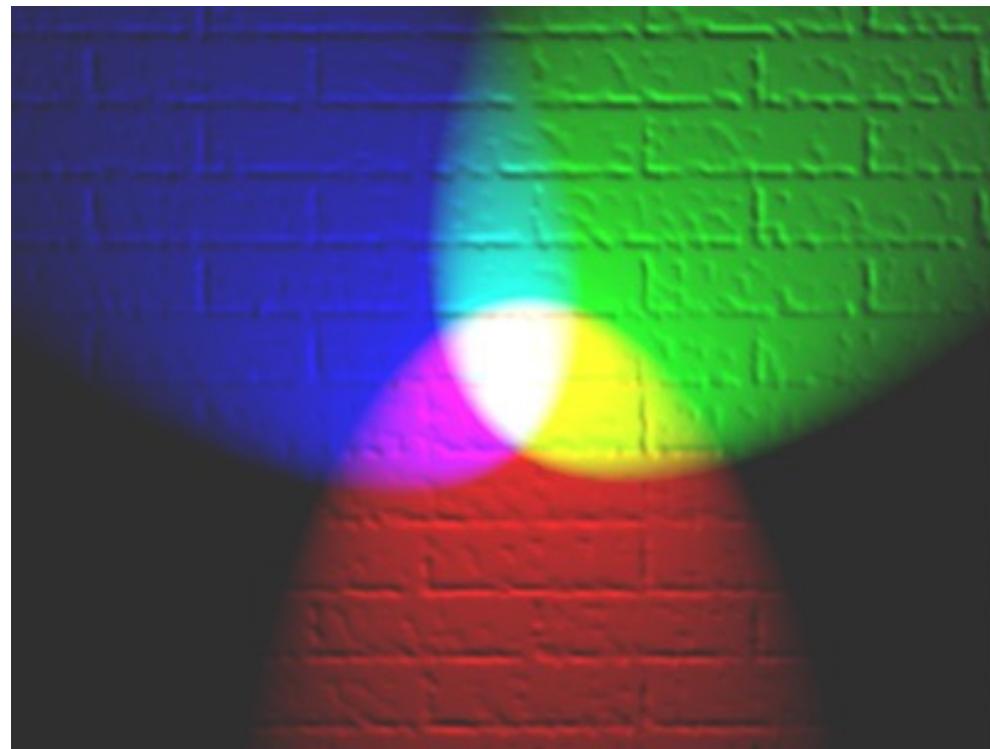


Colour Images in Matlab

- Images represented as a matrix
- Suppose we have a NxM RGB image called “im”
 - $im(0,0,0)$ = top-left pixel value in R-channel
 - $im(y, x, b)$ = y pixels down, x pixels to right in the bth channel
- `cv2.imread(filename)` returns a uint8 image (values 0 to 255)
 - **Convert to double format (values 0 to 1) (important !)**

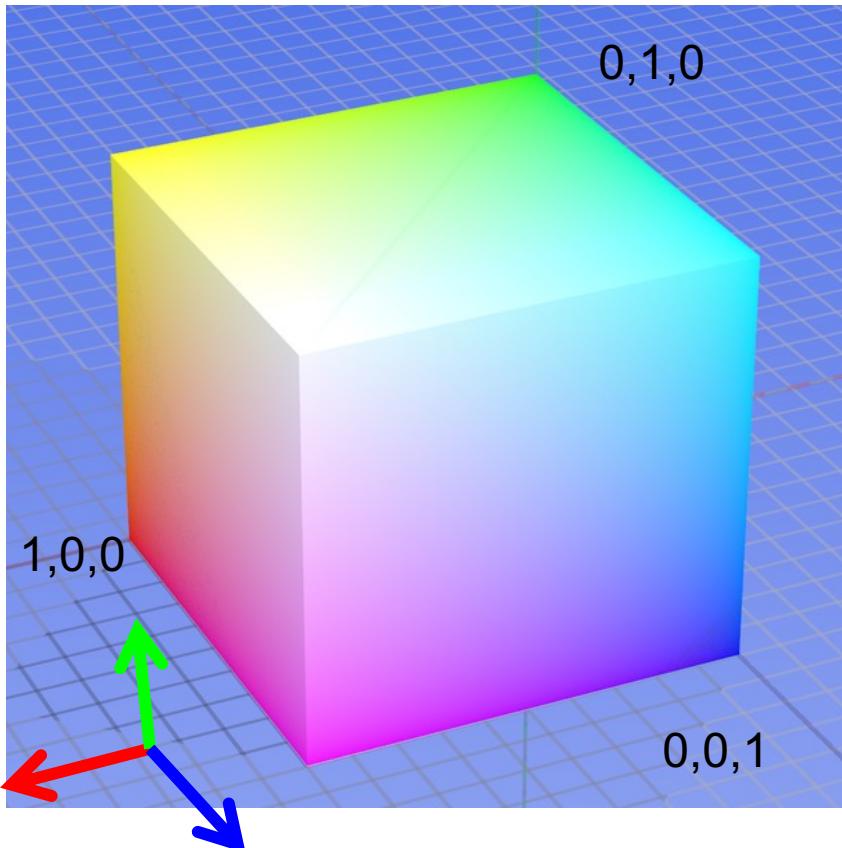
row	column	R	G	B
0	0	0.92	0.99	0.92
0	1	0.93	0.99	0.94
0	2	0.94	0.99	0.95
0	3	0.97	0.99	0.96
0	4	0.62	0.99	0.97
0	5	0.37	0.99	0.98
0	6	0.85	0.99	0.99
0	7	0.97	0.99	0.99
0	8	0.93	0.99	0.99
0	9	0.92	0.99	0.99
0	10	0.99	0.99	0.99
1	0	0.95	0.99	0.95
1	1	0.89	0.99	0.96
1	2	0.82	0.99	0.97
1	3	0.89	0.99	0.98
1	4	0.56	0.99	0.99
1	5	0.31	0.99	0.99
1	6	0.75	0.99	0.99
1	7	0.92	0.99	0.99
1	8	0.81	0.99	0.99
1	9	0.95	0.99	0.99
1	10	0.91	0.99	0.99
1	11	0.91	0.99	0.99
2	0	0.89	0.99	0.92
2	1	0.72	0.99	0.93
2	2	0.51	0.99	0.94
2	3	0.55	0.99	0.95
2	4	0.51	0.99	0.96
2	5	0.42	0.99	0.97
2	6	0.57	0.99	0.98
2	7	0.41	0.99	0.99
2	8	0.49	0.99	0.99
2	9	0.91	0.99	0.99
2	10	0.92	0.99	0.99
2	11	0.91	0.99	0.99
3	0	0.96	0.99	0.95
3	1	0.95	0.99	0.96
3	2	0.88	0.99	0.97
3	3	0.94	0.99	0.98
3	4	0.56	0.99	0.99
3	5	0.46	0.99	0.99
3	6	0.91	0.99	0.99
3	7	0.87	0.99	0.99
3	8	0.90	0.99	0.99
3	9	0.97	0.99	0.99
3	10	0.99	0.99	0.99
3	11	0.95	0.99	0.99
4	0	0.71	0.99	0.93
4	1	0.81	0.99	0.94
4	2	0.81	0.99	0.95
4	3	0.87	0.99	0.96
4	4	0.57	0.99	0.97
4	5	0.37	0.99	0.98
4	6	0.80	0.99	0.99
4	7	0.88	0.99	0.99
4	8	0.89	0.99	0.99
4	9	0.79	0.99	0.99
4	10	0.79	0.99	0.99
4	11	0.85	0.99	0.99
5	0	0.49	0.99	0.91
5	1	0.62	0.99	0.92
5	2	0.60	0.99	0.93
5	3	0.58	0.99	0.94
5	4	0.50	0.99	0.95
5	5	0.60	0.99	0.96
5	6	0.58	0.99	0.97
5	7	0.50	0.99	0.98
5	8	0.61	0.99	0.99
5	9	0.45	0.99	0.99
5	10	0.33	0.99	0.99
5	11	0.33	0.99	0.99
6	0	0.86	0.99	0.91
6	1	0.84	0.99	0.92
6	2	0.74	0.99	0.93
6	3	0.58	0.99	0.94
6	4	0.51	0.99	0.95
6	5	0.39	0.99	0.96
6	6	0.73	0.99	0.97
6	7	0.92	0.99	0.98
6	8	0.91	0.99	0.98
6	9	0.49	0.99	0.99
6	10	0.74	0.99	0.99
6	11	0.74	0.99	0.99
7	0	0.96	0.99	0.93
7	1	0.67	0.99	0.94
7	2	0.54	0.99	0.95
7	3	0.85	0.99	0.96
7	4	0.48	0.99	0.97
7	5	0.37	0.99	0.98
7	6	0.88	0.99	0.99
7	7	0.90	0.99	0.99
7	8	0.94	0.99	0.99
7	9	0.82	0.99	0.99
7	10	0.94	0.99	0.99
7	11	0.92	0.99	0.99
8	0	0.69	0.99	0.95
8	1	0.49	0.99	0.96
8	2	0.56	0.99	0.97
8	3	0.66	0.99	0.98
8	4	0.43	0.99	0.99
8	5	0.42	0.99	0.99
8	6	0.77	0.99	0.99
8	7	0.73	0.99	0.99
8	8	0.71	0.99	0.99
8	9	0.90	0.99	0.99
8	10	0.99	0.99	0.99
8	11	0.99	0.99	0.99
9	0	0.79	0.99	0.97
9	1	0.73	0.99	0.98
9	2	0.90	0.99	0.99
9	3	0.67	0.99	0.99
9	4	0.33	0.99	0.99
9	5	0.61	0.99	0.99
9	6	0.69	0.99	0.99
9	7	0.79	0.99	0.99
9	8	0.73	0.99	0.99
9	9	0.93	0.99	0.99
9	10	0.93	0.99	0.99
9	11	0.49	0.99	0.99
10	0	0.79	0.99	0.97
10	1	0.73	0.99	0.98
10	2	0.90	0.99	0.99
10	3	0.67	0.99	0.99
10	4	0.33	0.99	0.99
10	5	0.61	0.99	0.99
10	6	0.69	0.99	0.99
10	7	0.79	0.99	0.99
10	8	0.73	0.99	0.99
10	9	0.99	0.99	0.99
10	10	0.99	0.99	0.99
10	11	0.97	0.99	0.99
11	0	0.91	0.99	0.93
11	1	0.94	0.99	0.94
11	2	0.89	0.99	0.95
11	3	0.49	0.99	0.96
11	4	0.41	0.99	0.97
11	5	0.78	0.99	0.98
11	6	0.78	0.99	0.98
11	7	0.77	0.99	0.98
11	8	0.89	0.99	0.98
11	9	0.99	0.99	0.98
11	10	0.93	0.99	0.98
11	11	0.90	0.99	0.98
12	0	0.79	0.99	0.97
12	1	0.73	0.99	0.98
12	2	0.90	0.99	0.99
12	3	0.67	0.99	0.99
12	4	0.33	0.99	0.99
12	5	0.61	0.99	0.99
12	6	0.69	0.99	0.99
12	7	0.79	0.99	0.99
12	8	0.73	0.99	0.99
12	9	0.99	0.99	0.99
12	10	0.99	0.99	0.99
12	11	0.97	0.99	0.99
13	0	0.91	0.99	0.93
13	1	0.94	0.99	0.94
13	2	0.89	0.99	0.95
13	3	0.49	0.99	0.96
13	4	0.41	0.99	0.97
13	5	0.78	0.99	0.98
13	6	0.78	0.99	0.98
13	7	0.77	0.99	0.98
13	8	0.89	0.99	0.98
13	9	0.99	0.99	0.98
13	10	0.93	0.99	0.98
13	11	0.90	0.99	0.98

Color spaces



RGB space

Default color space



Some drawbacks

- Strongly correlated channels
- Not perceptually meaningful.



R
(G=0,B=0)

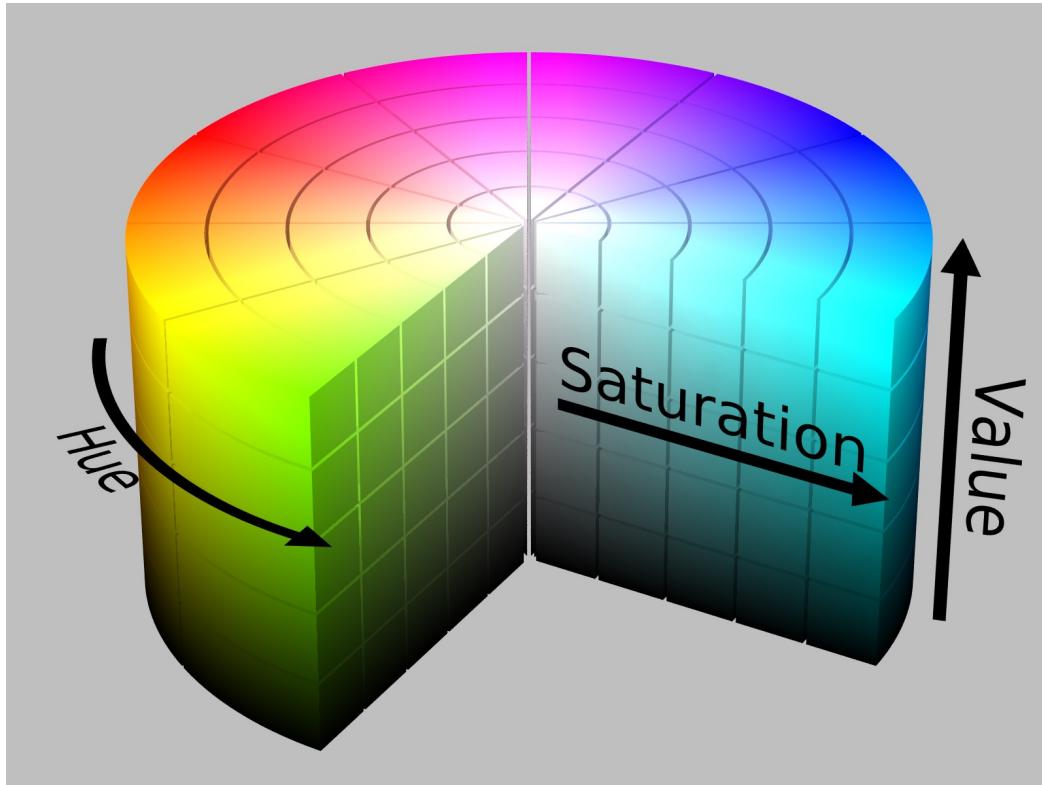


G
(R=0,B=0)

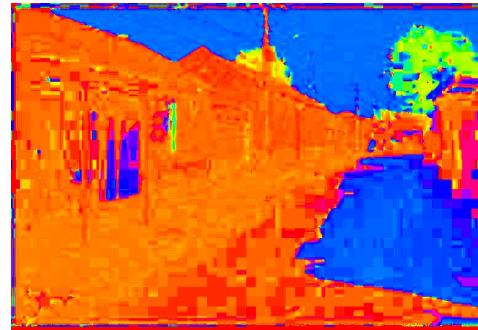


B
(R=0,G=0)

HSV space



Intuitive color space



H
(S=1, V=1)



S
(H=1, V=1)



V
(H=1, S=0)

Subjective terms to describe color

Hue

Name of the color
(yellow, red, blue, green, ...)

Value/Lightness/Brightness

How light or dark a color is.

Saturation/Chroma/Color Purity

How “strong” or “pure” a color is.

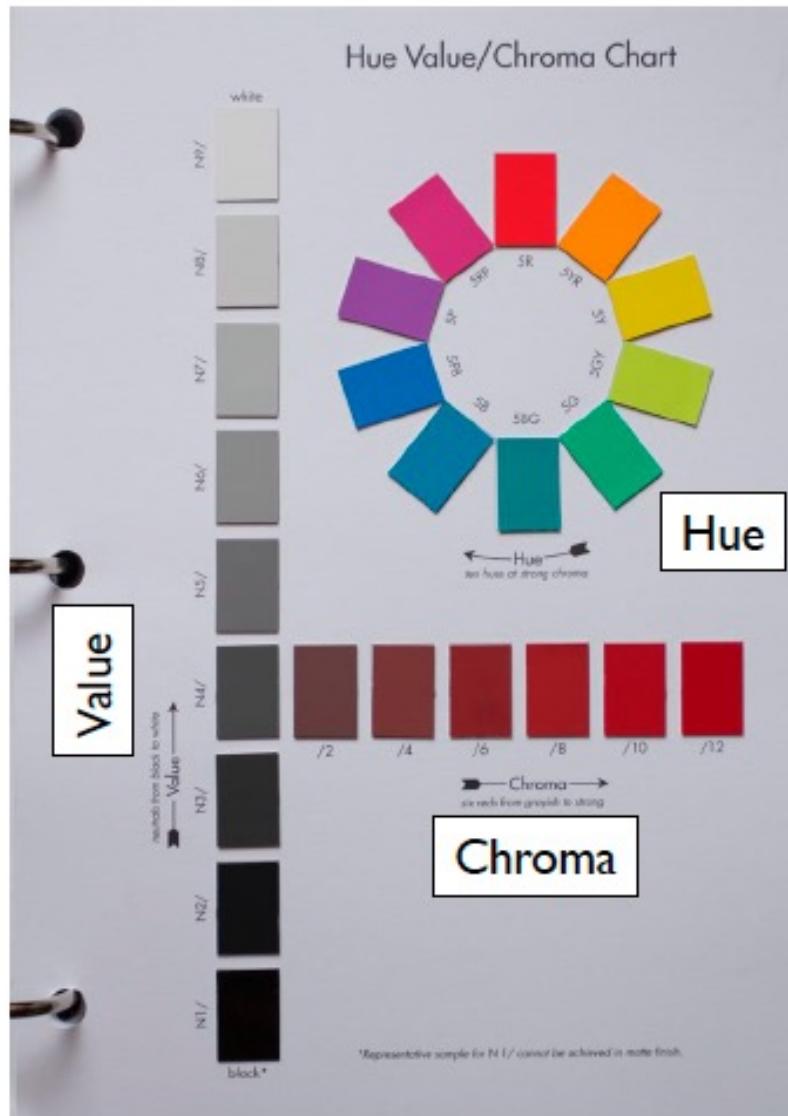


Image from Benjamin Salley
A page from a Munsell Student Color Set

HSV space

$$X_{max} := \max(R, G, B) =: V$$

$$X_{min} := \min(R, G, B)$$

$$C := X_{max} - X_{min}$$

$$H := \begin{cases} 0, & \text{if } C = 0 \\ 60^\circ \cdot \left(0 + \frac{G-B}{C}\right), & \text{if } V = R \\ 60^\circ \cdot \left(2 + \frac{B-R}{C}\right), & \text{if } V = G \\ 60^\circ \cdot \left(4 + \frac{R-G}{C}\right), & \text{if } V = B \end{cases}$$

$$S_V := \begin{cases} 0, & \text{if } V = 0 \\ \frac{C}{V}, & \text{otherwise} \end{cases}$$

From wikipedia. : https://en.wikipedia.org/wiki/HSL_and_HSV

Most semantic information is contained in the intensity band



Original image

Most semantic information is contained in intensity channel



copyright 2000 philg@mit.edu

Summary

- Geometric image formation describes **where** the 3D objects are projected in the image. (location)
- Photometric image formation describes **the appearance** of the objects. (intensity, color, appearance.)

Readings

- Computer Vision: Algorithms and Applications
 - 2nd Edition, Chapter-2 (2.1.4- perspective, camera intrinsics, camera matrix, 2.3.2 color)
 - Digital image processing, Chapter 2.4.2, 2.4.3
-
- Take home message:
 - Have a general idea of geometric and photometric image formation process.
 - Understand the color image representation.