Elaborazione delle Immagini
Informazione Multimediale - Immagini

Raffaella Lanzarotti

General information

- Lanzarotti Raffaella:
  - *web page*: lanzarotti.di.unimi.it
  - e-mail: lanzarotti@di.unimi.it
  - reception: on appointment
- Course web page:
  - http://lanzarotti.di.unimi.it/DIP1.html
- Timetable:
  - monday (room DELTA)
    - 2:00 – 4:15 pm
  - wednesday (room DELTA)
    - 2:00 - 4:15 pm

General information

- Books
  - Rafael C. Gonzalez & Richard E. Woods, "**Digital Image Processing**", Pearson; 3rd ed.2008
  - R. C. Gonzalez, R. E. Woods, S.L. Eddins: **"Digital Image Processing using MATLAB"**
  - Richard Szeliski: Computer Vision: Algorithms and Applications, Springer 2010
  - Forsyth and Ponce: Computer Vision: A Modern Approach, Pearson 2011
  - Ze-Nian Li, Mark S. Drew, Jiangchuan Liu **"Fundamental of Multimedia"**, Springer, 2nd ed., 2014

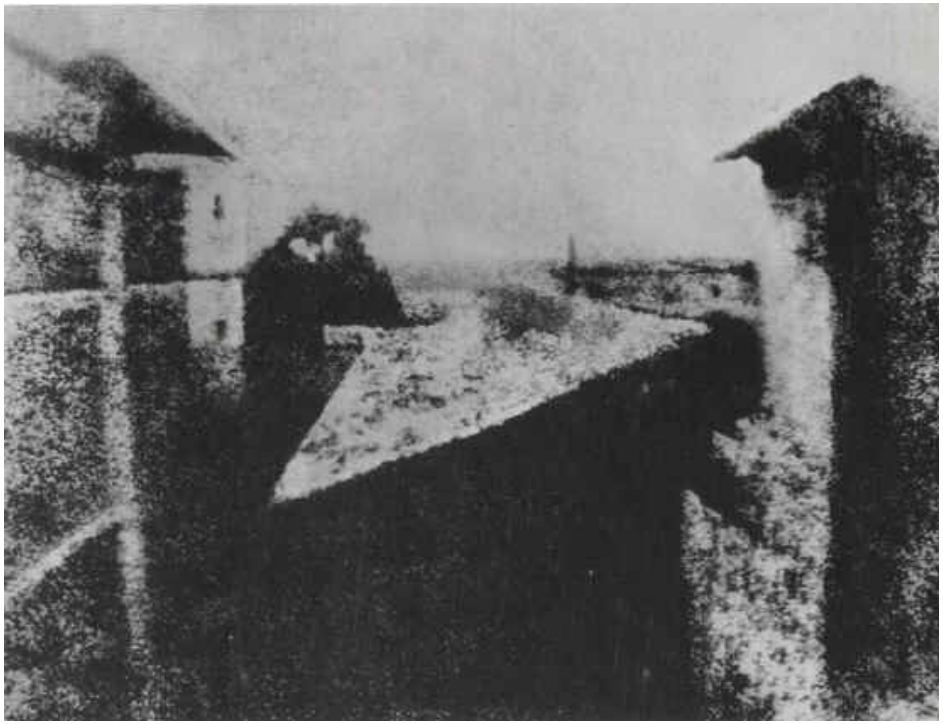
A picture is worth a thousand words.

--- Confucius

or *Printers' Ink* Ad (1921)

It will our motto!

How old is the first picture?



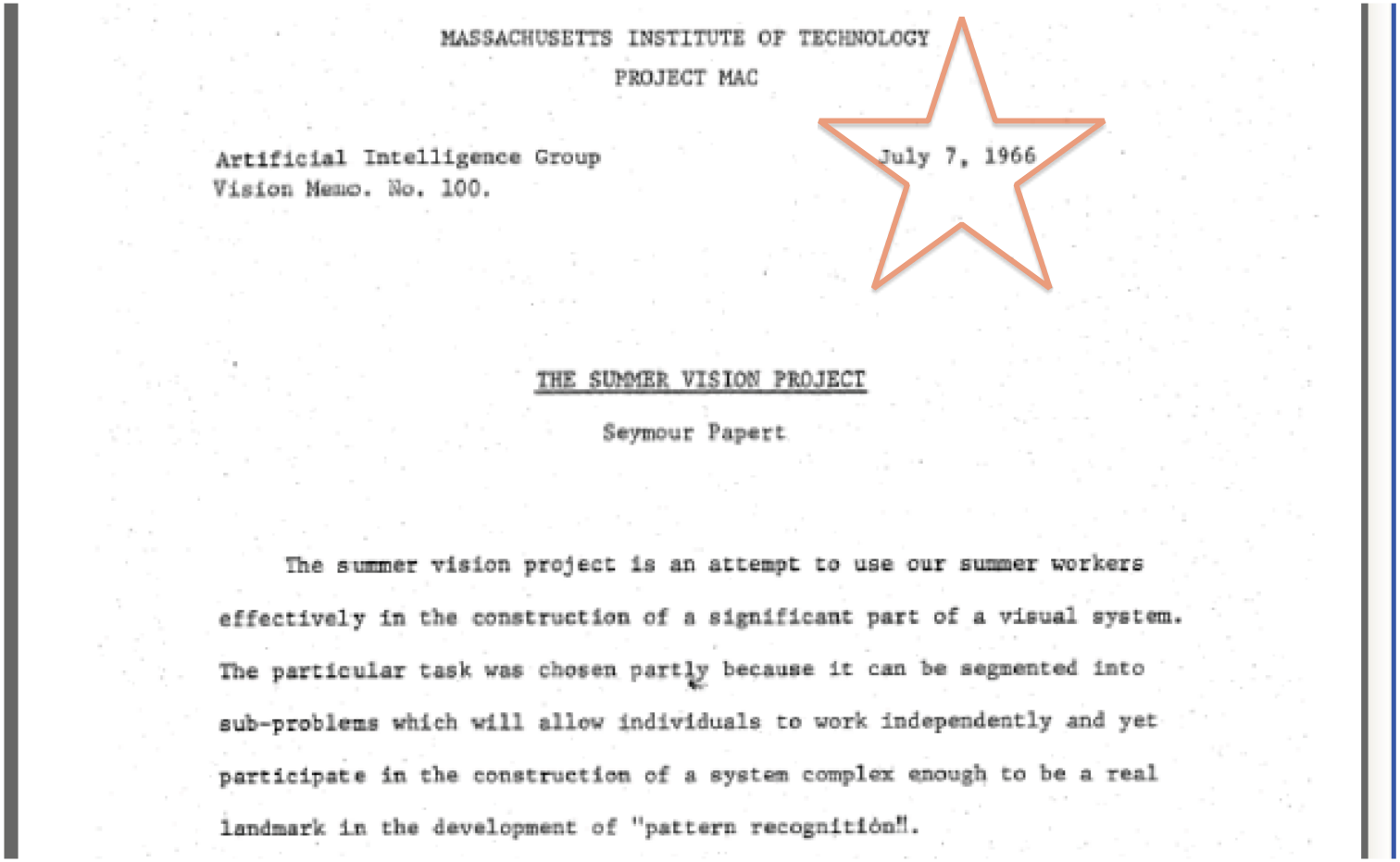
- “View from the Window at Le Gras”, France, 1826,
- taken by Nicéphore Niepce
- Obtained covering of bitumen a sheet
- Time of exposure: 8 hours!!!

What about digital picture?

FIGURE 1.1 A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces. (McFarlane.[†])

FIGURE 1.2 A digital picture made in 1922 from a tape punched after the signals had crossed the Atlantic twice. (McFarlane.)

Origins of computer vision: an MIT undergraduate summer project



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

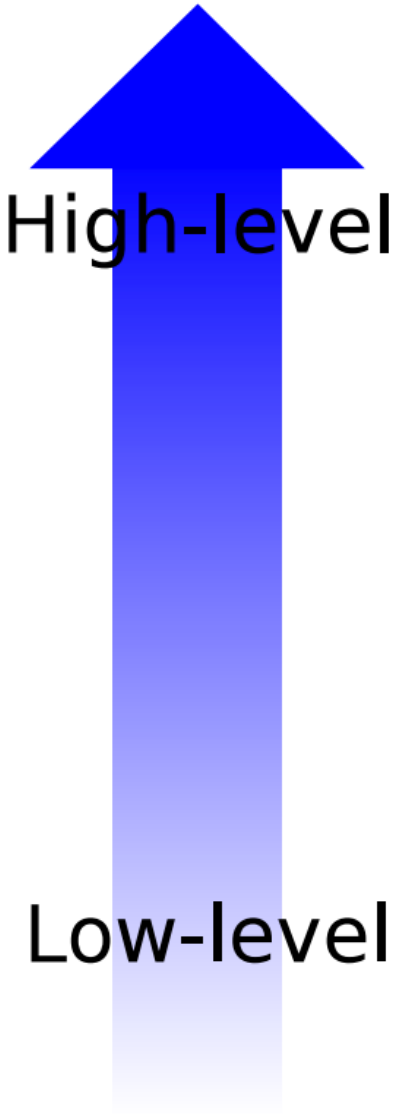
The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Image Processing, definition(s)

Strictly speaking, Image processing is the study of any algorithm that takes an image as input and returns an image as output.

Broadly speaking, Image processing it also concerns with the extraction of meaningful information (attributes) from an images: Image Analysis.

Framed in the Computer Vision research field



Computer Vision

Object detection, recognition, shape analysis, tracking
Use of Artificial Intelligence and Machine Learning

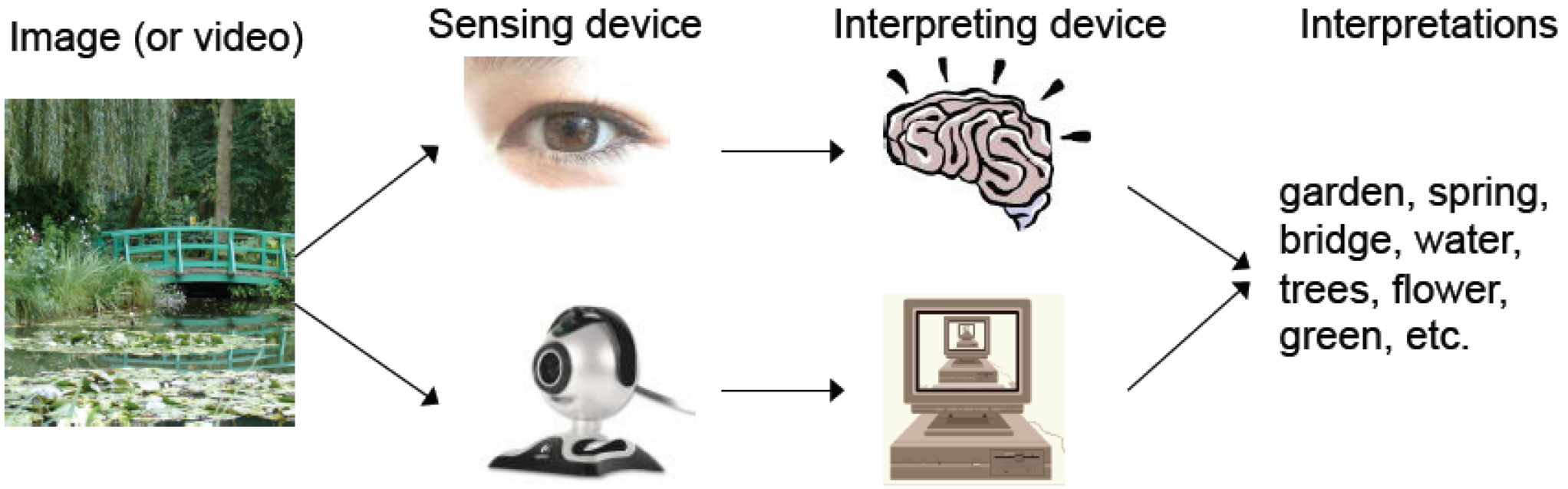
Image Analysis

Segmentation, image registration, matching

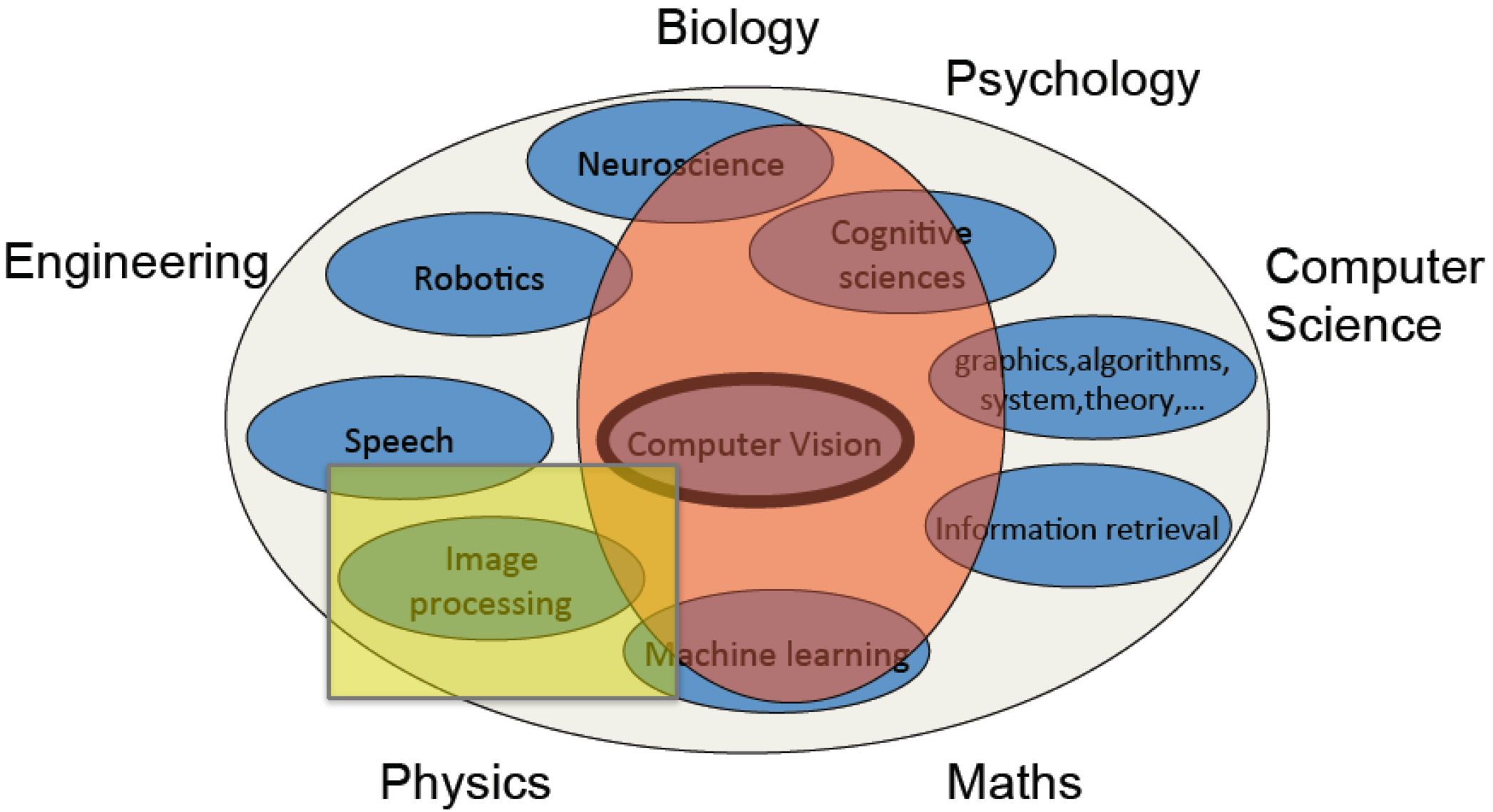
Image Processing

Image enhancement, noise removal, restoration,
feature detection, compression

What is (computer) vision?



What is it related to?



The goal of computer vision

- To bridge the gap between pixels and “meaning”

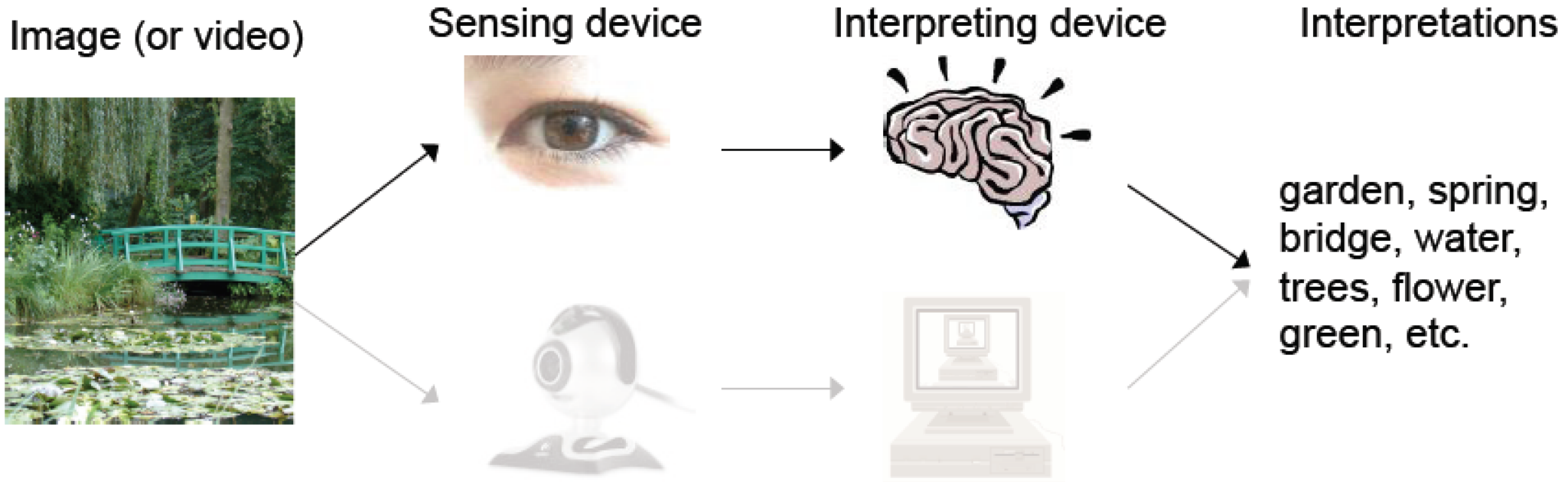
What we see

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

What a computer sees

Source: S. Narasimhan

What is (computer) vision?

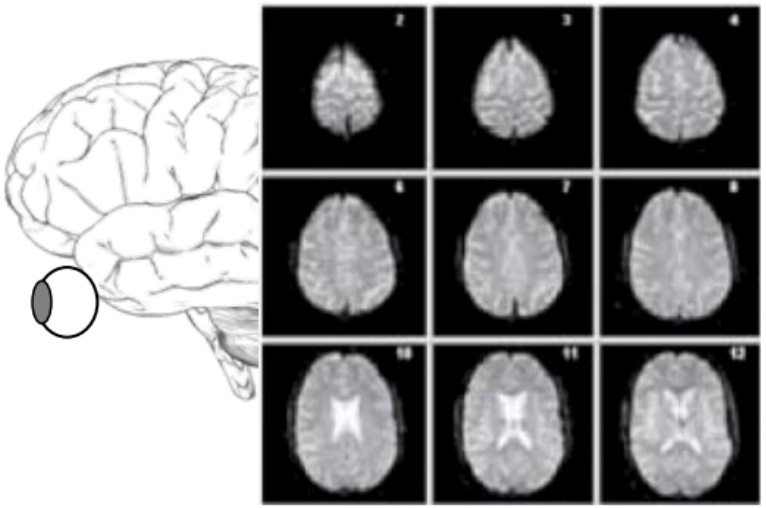


Pattern Recognition

Training Images

Testing Image

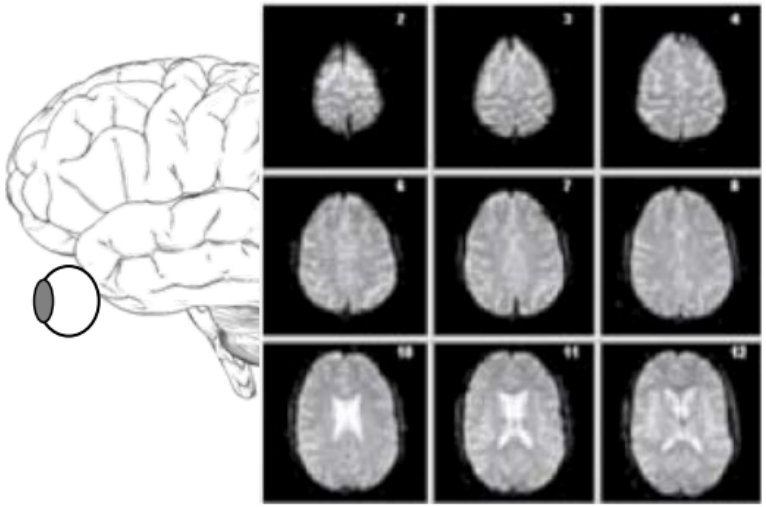
Select voxels



Train

Statistical Pattern Recognition Algorithm

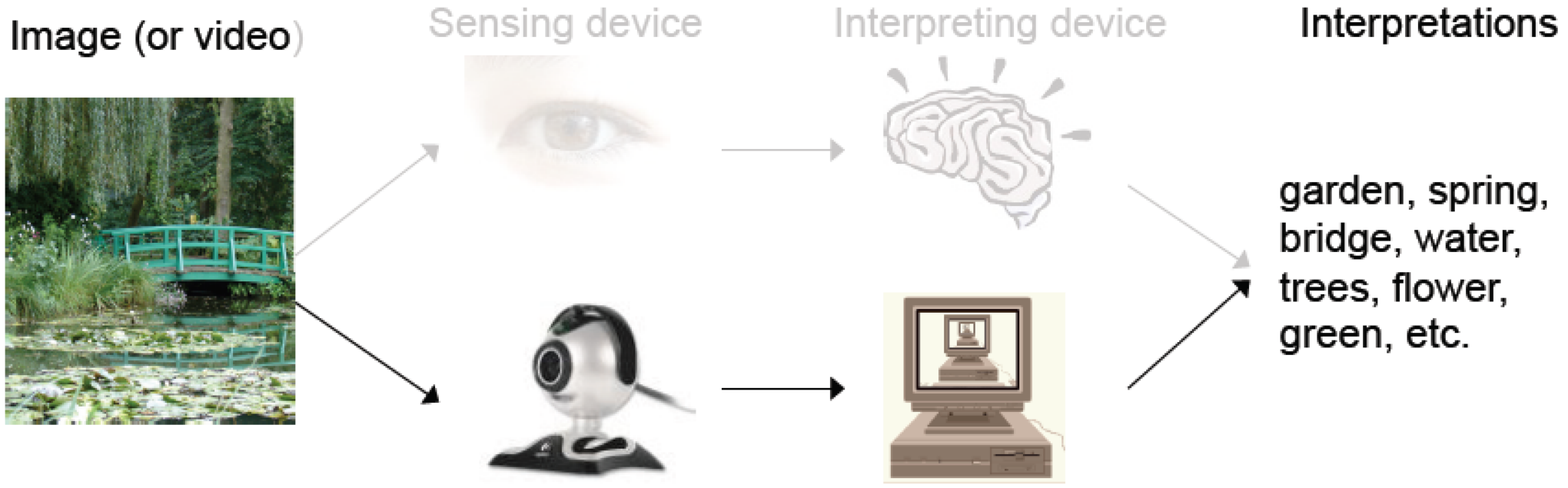
Select voxels



Test

Guess "beach"

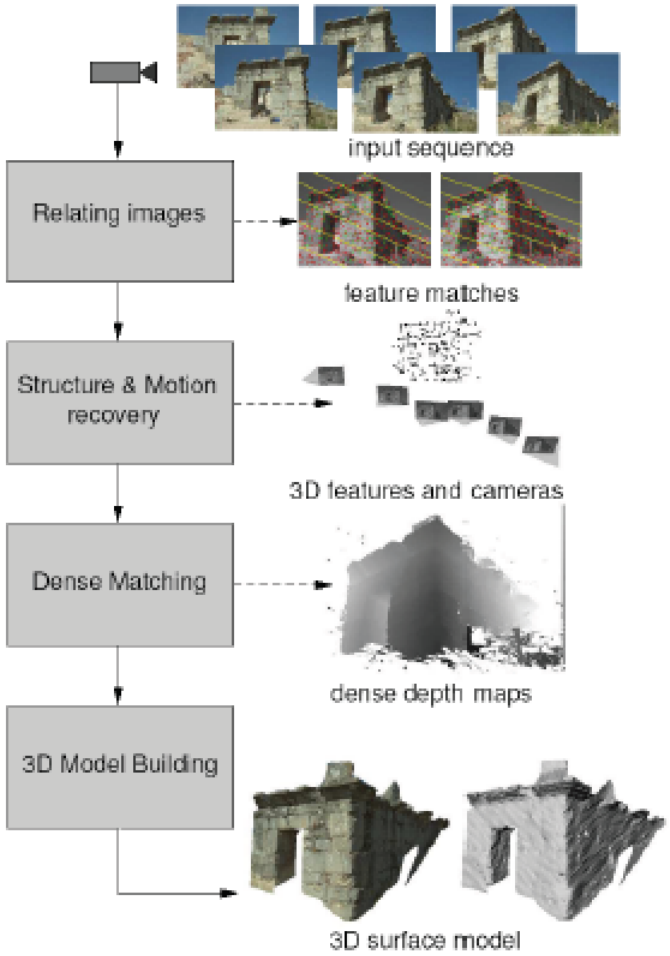
What is (computer) vision?



What kind of information can we extract from an image?

- Metric 3D information
- Semantic information

Vision as measurement device



Pollefeys et al.

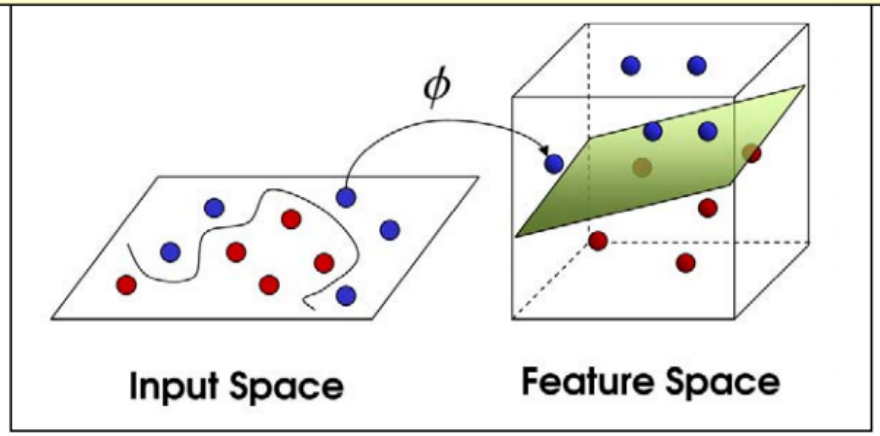
Goesele et al.

Vision as a source of semantic information

Slide credit: Kristen Grauman

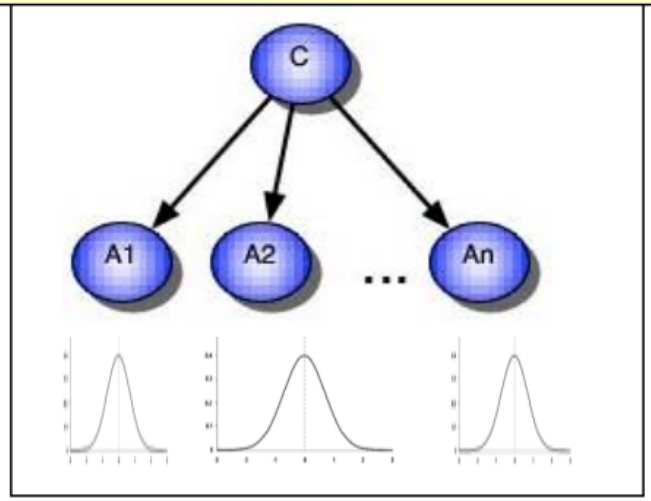
Pattern Recognition

Support Vector Machine (SVM)

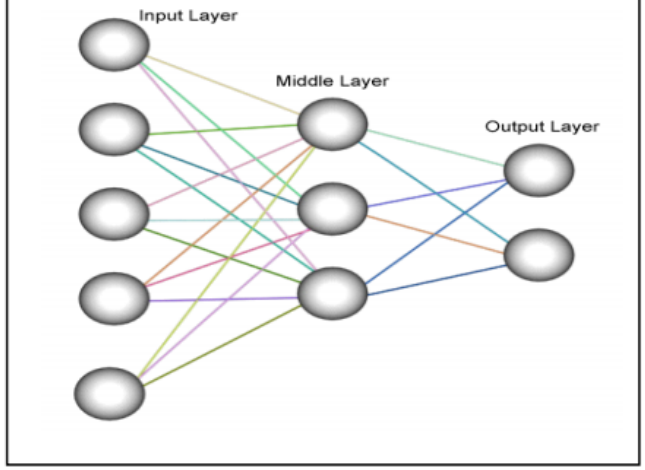


Statistical Pattern Recognition Algorithm

Gaussian Naïve Bayes (GNB)

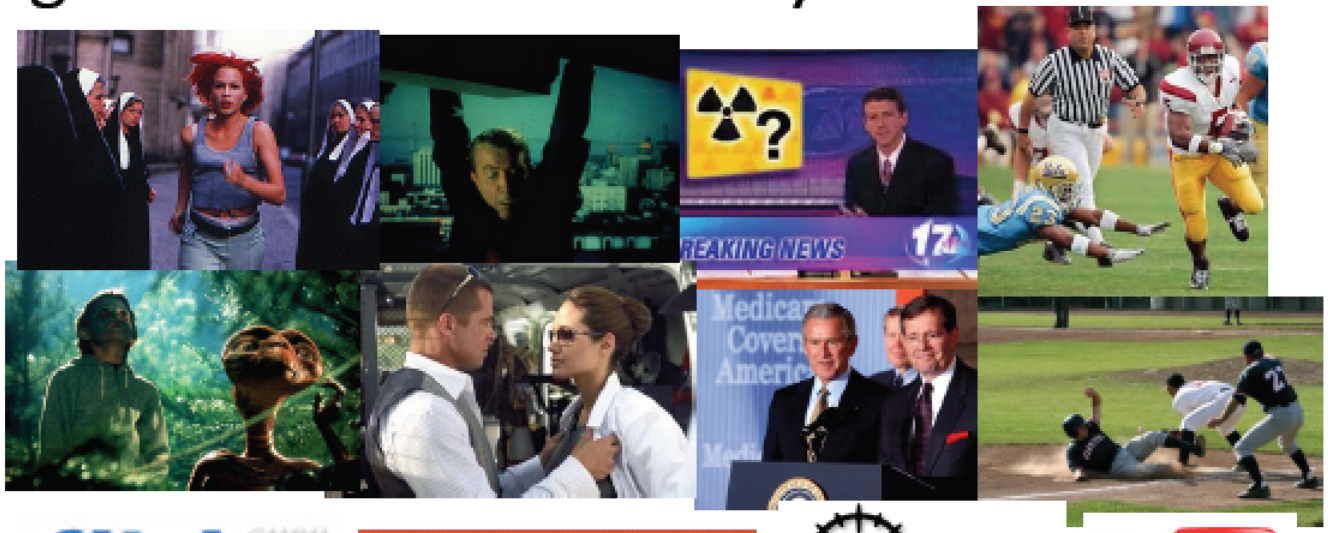


Neural Networks



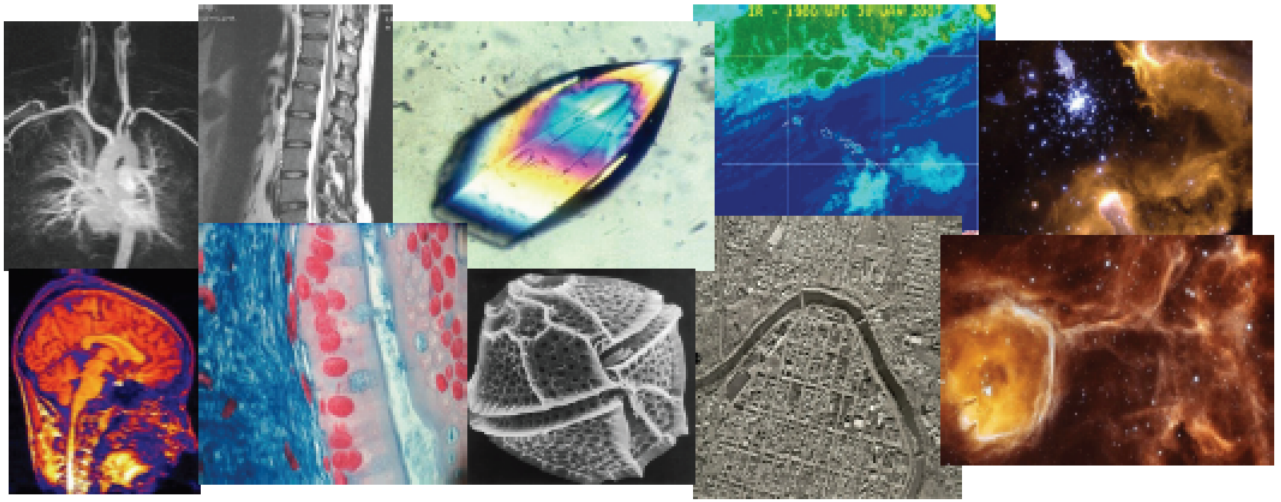
Why study computer vision?

- Vision is useful: Images and video are everywhere!



Google Image Search Picasa™ flickr™ websh ts picsearch™ YouTube Broadcast Yourself™

Surveillance and security

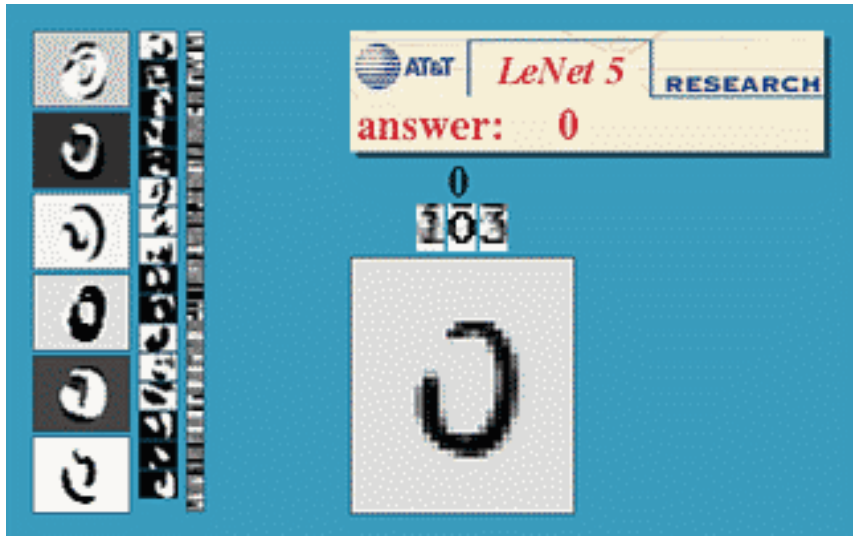


Medical and scientific images

Optical character recognition (OCR)

Technology to convert scanned documents to text

- Scanner are often equipped of OCR



Digit recognition, AT&T labs
http://www.research.att.com/

License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

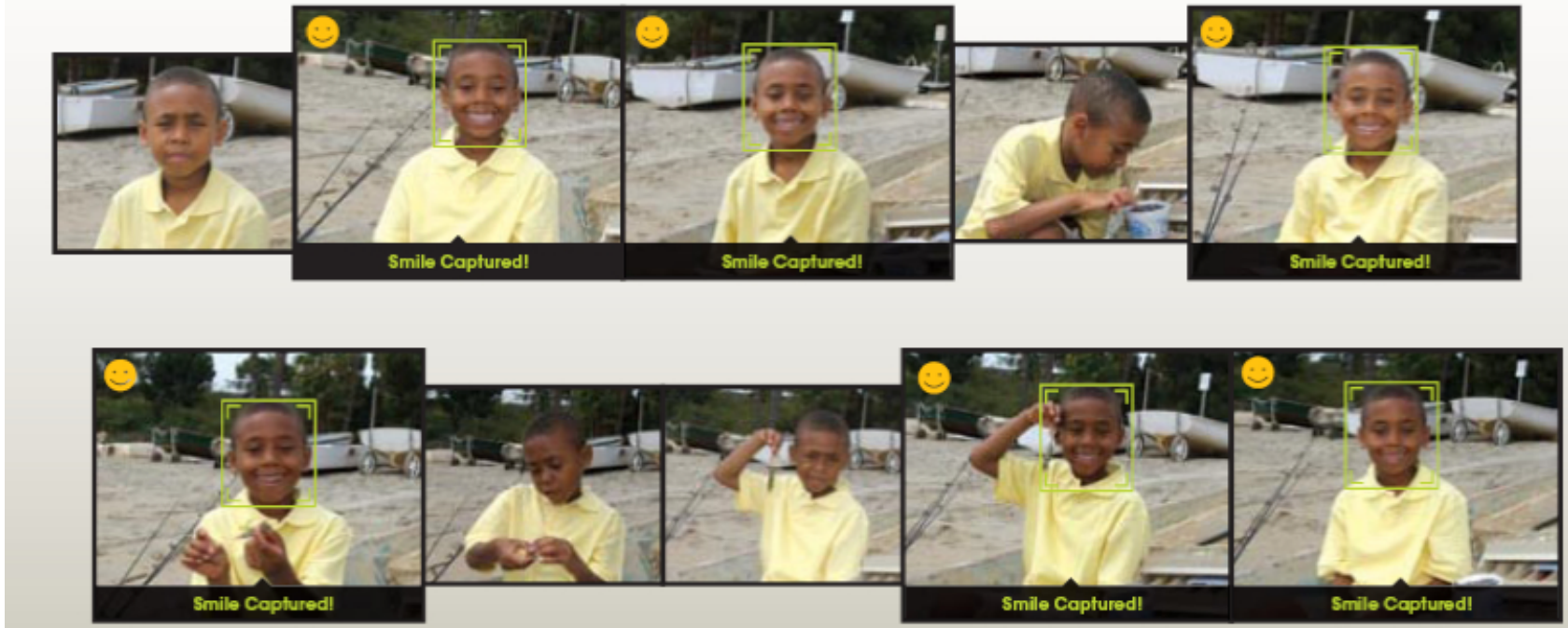
Face detection

- Digital fotocamera automatically detect faces
  - Canon, Sony, Fuji, …

Smile detection

The Smile Shutter flow

Imagine a camera smart enough to catch every smile! In Smile Shutter Mode, your Cyber-shot® camera can automatically trip the shutter at just the right instant to catch the perfect expression.



Sony Cyber-shot® T70 Digital Still Camera

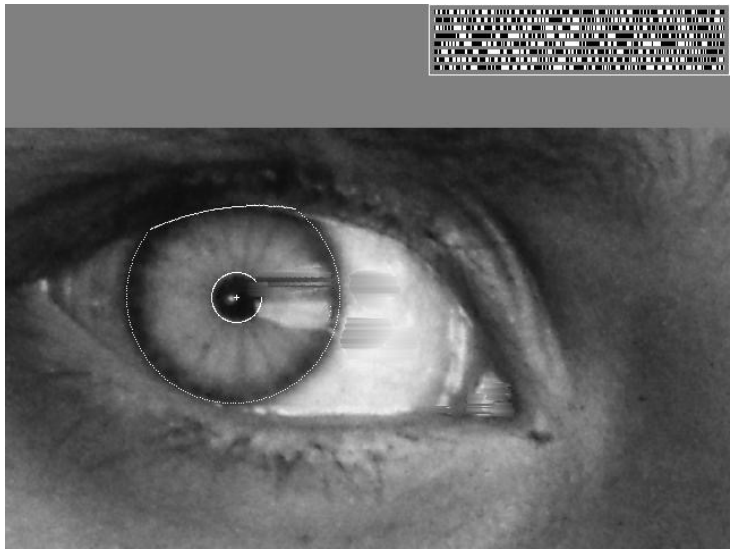
Object recognition (in supermarkets)

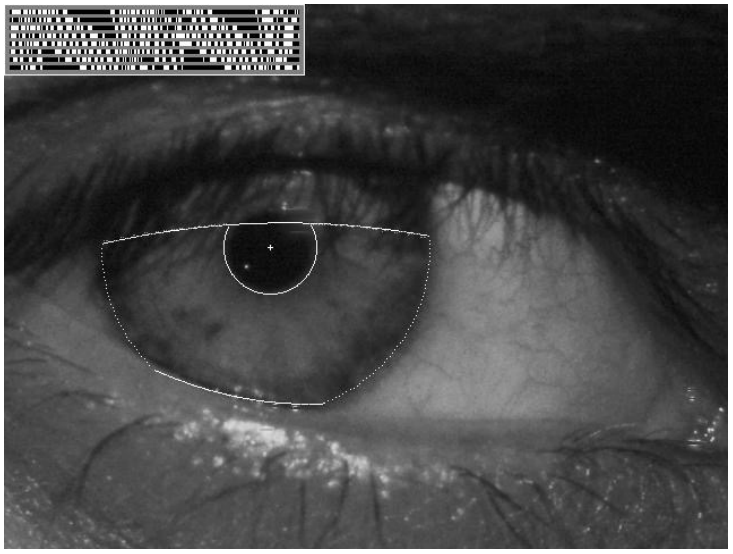
LaneHawk by EvolutionRobotics

“A smart camera is flush-mounted in the checkout lane, continuously watching for items. When an item is detected and recognized, the cashier verifies the quantity of items that were found under the basket, and continues to close the transaction. The item can remain under the basket, and with LaneHawk,you are assured to get paid for it… “

Vision-based biometrics

"*How the Afghan Girl was Identified by Her Iris Patterns*" Read the story wikipedia





Login without a password...

Fingerprint scanners on many new laptops, other devices

Face recognition systems now beginning to appear more widely
http://www.sensiblevision.com/

Object recognition (in mobile phones)

Point & Find, Nokia

Google Goggles

Special effects: shape capture

The Matrix movies, ESC Entertainment, XYZRGB, NRC

Smart cars

Slide content courtesy of Amnon Shashua

- Mobileye
  - Vision systems currently in high-end BMW, GM, Volvo models
  - By 2010: 70% of car manufacturers.

Google cars

http://www.nytimes.com/2010/10/10/science/10google.html?ref=artificialintelligence

Vision-based interaction (and games)

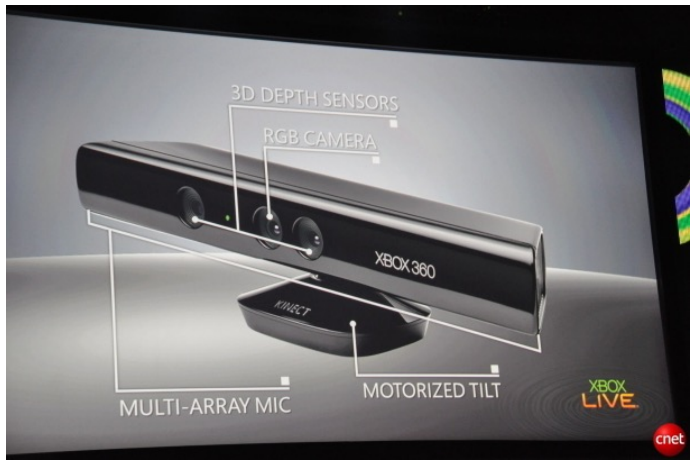
Nintendo Wii has camera-based IR tracking built in. See Lee's work at CMU on clever tricks on using it to create a multi-touch display!

Digimask: put your face on a 3D avatar.

"Game turns moviegoers into Human Joysticks", CNET
Camera tracking a crowd, based on this work.

Interactive Games: Kinect

- Object Recognition: http://www.youtube.com/watch?feature=iv&v=fQ59dXOo63o
- Mario: http://www.youtube.com/watch?v=8CTJL5lUjHg
- 3D: http://www.youtube.com/watch?v=7QrnwoO1-8A
- Robot: http://www.youtube.com/watch?v=w8BmgtMKFbY



Vision in space

NASA'S Mars Exploration Rover Spirit captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.

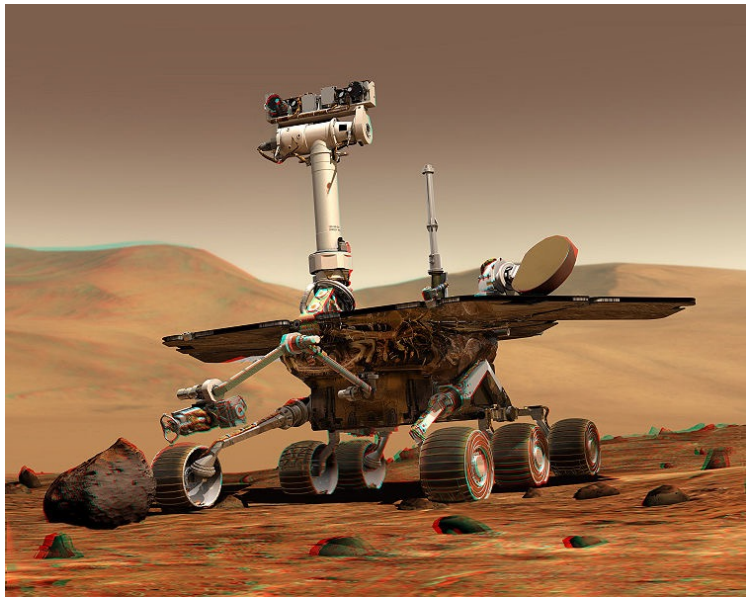
Vision systems (JPL) used for several tasks

- Panorama stitching
- 3D terrain modeling
- Obstacle detection, position tracking
- For more, read "Computer Vision on Mars" by Matthies et al.

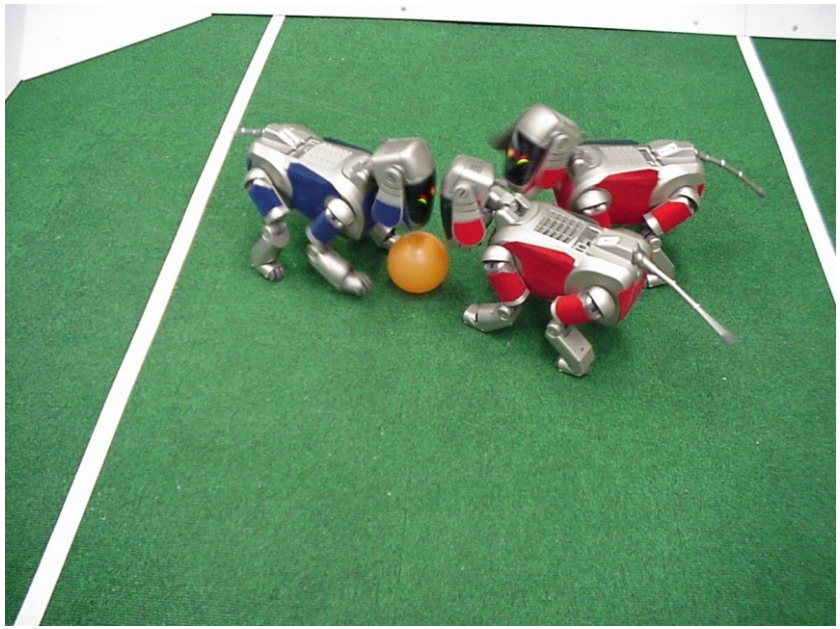
Industrial robots

Vision-guided robots position nut runners on wheels

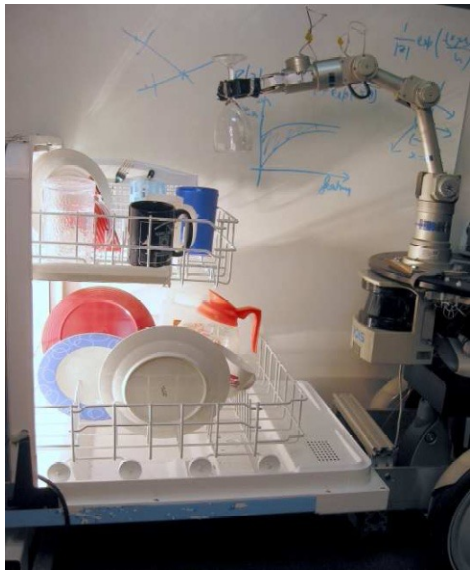
Mobile robots



NASA's Mars Spirit Rover
http://en.wikipedia.org/wiki/Spirit_rover

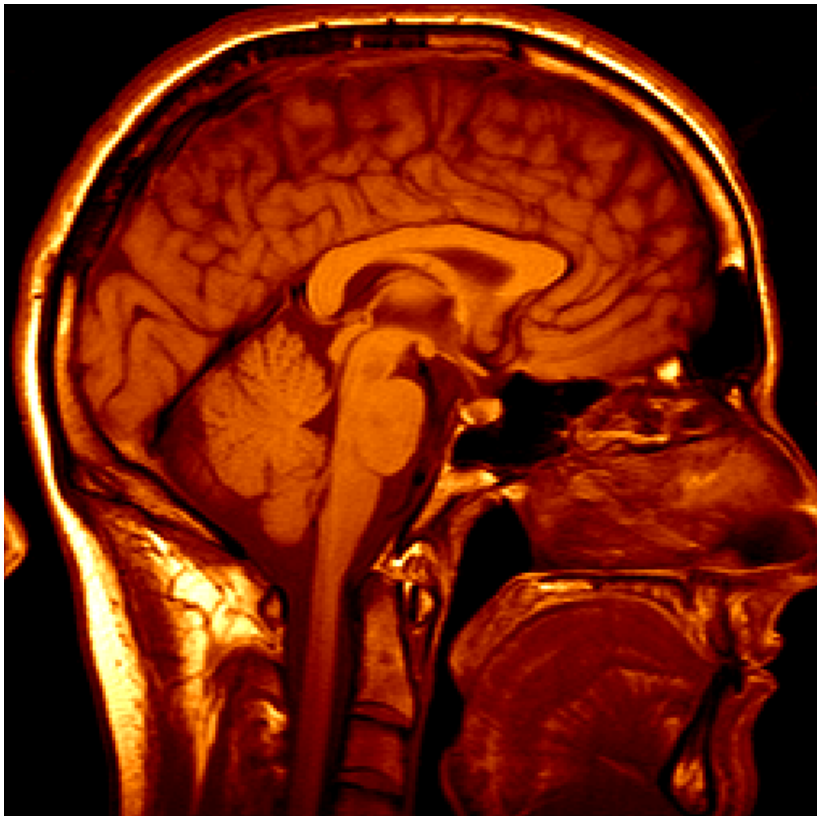


http://www.robocup.org/



Saxena et al. 2008
STAIR at Stanford

Medical imaging



3D imaging
MRI, CT

Image guided surgery
Grimson et al., MIT

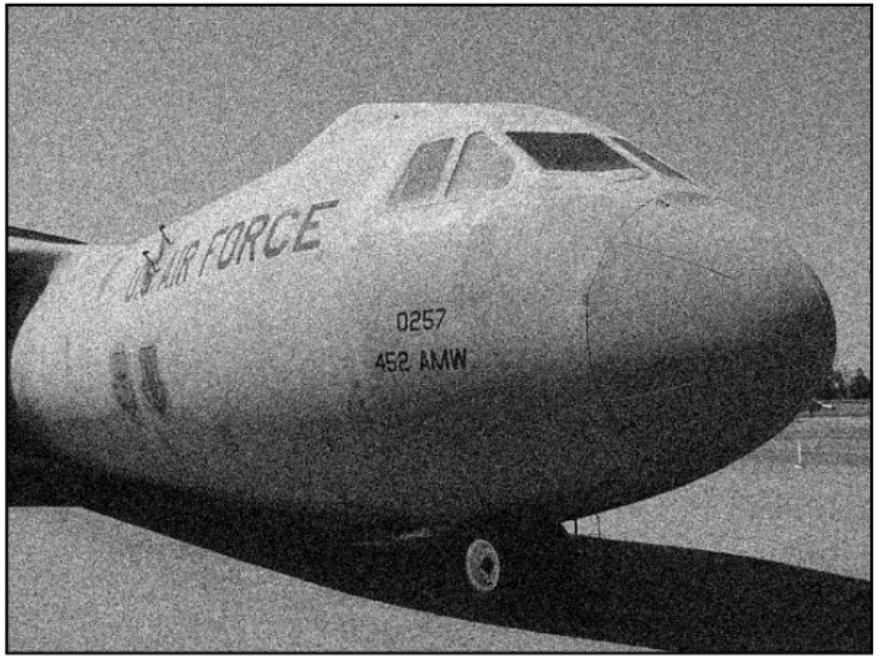
Summary: three levels of processing

- Low-level (img → img):
  - Noise removal
  - Contrast Adjustment
- Mid-level (img → attributes):
  - Edge detection (Estrazione dei bordi)
  - Segmentation (partitioning the img in regions)

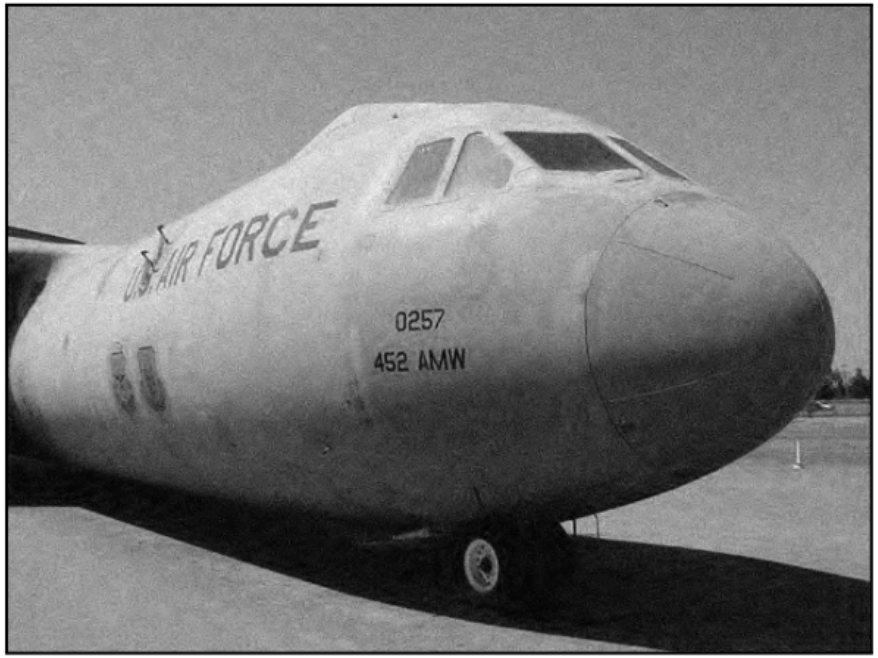
- High-level (img, attr. → classification)
  - Cognitive functions

Example: Noise Removal

Noisy Image



Denoised Image



Example: Contrast Adjustment

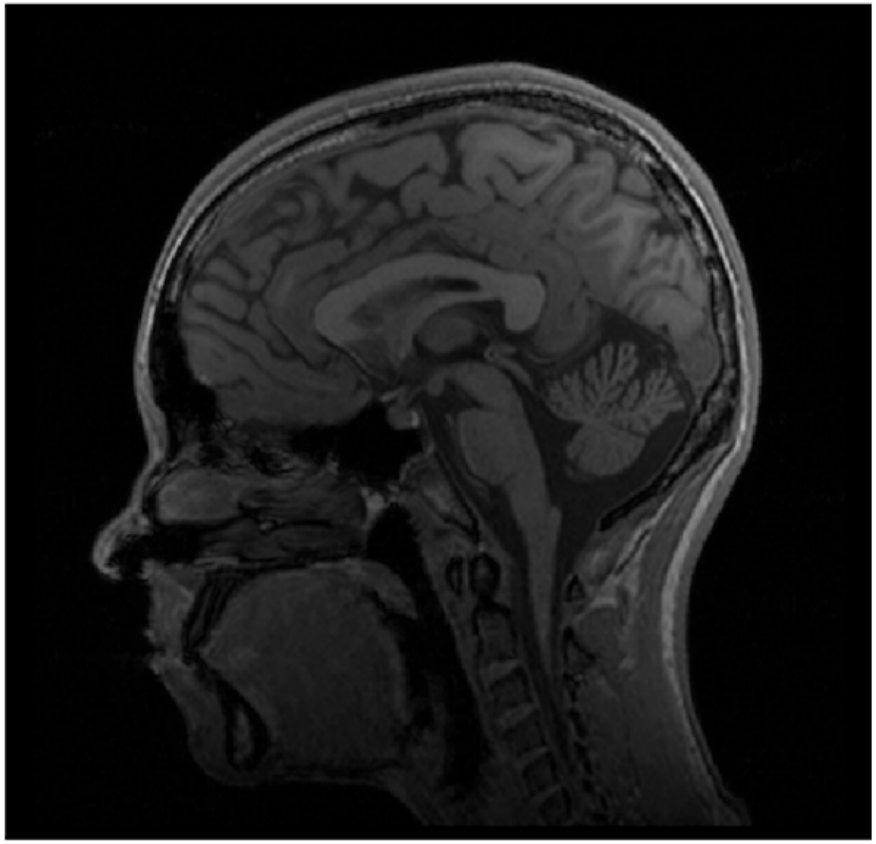
Low Contrast

Original Contrast

High Contrast

Example: Edge Detection

Example: Region Detection, Segmentation



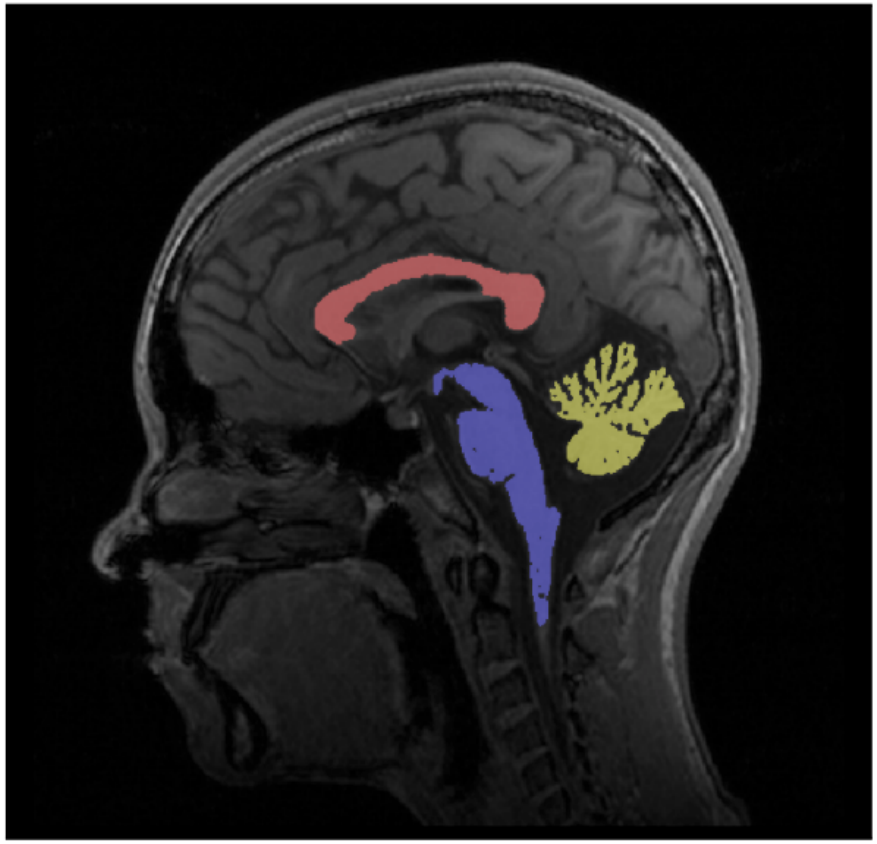


IMAGE ACQUISITION AND REPRESENTATION

On Digital Image Processing, and slides

Image acquisition

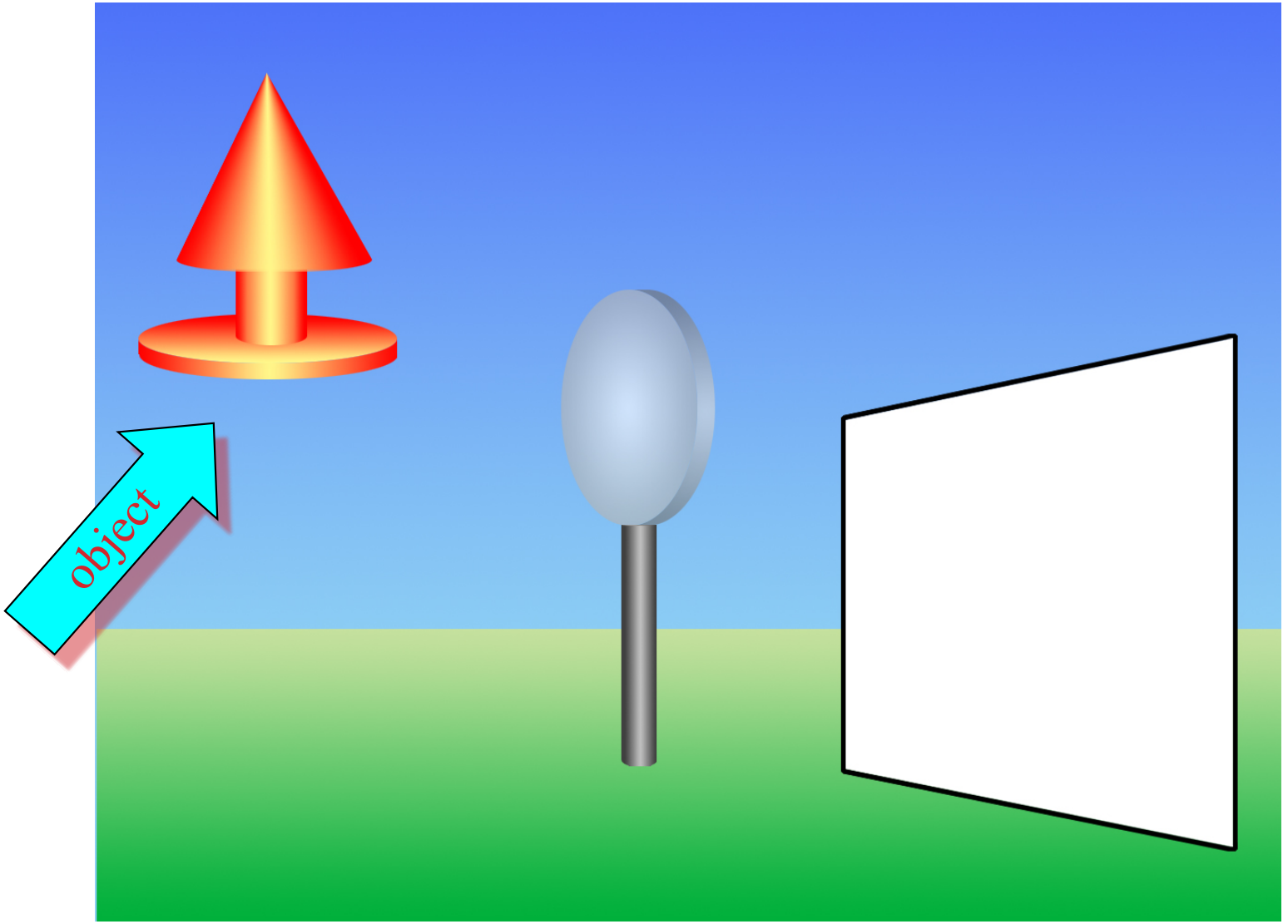


Image acquisition

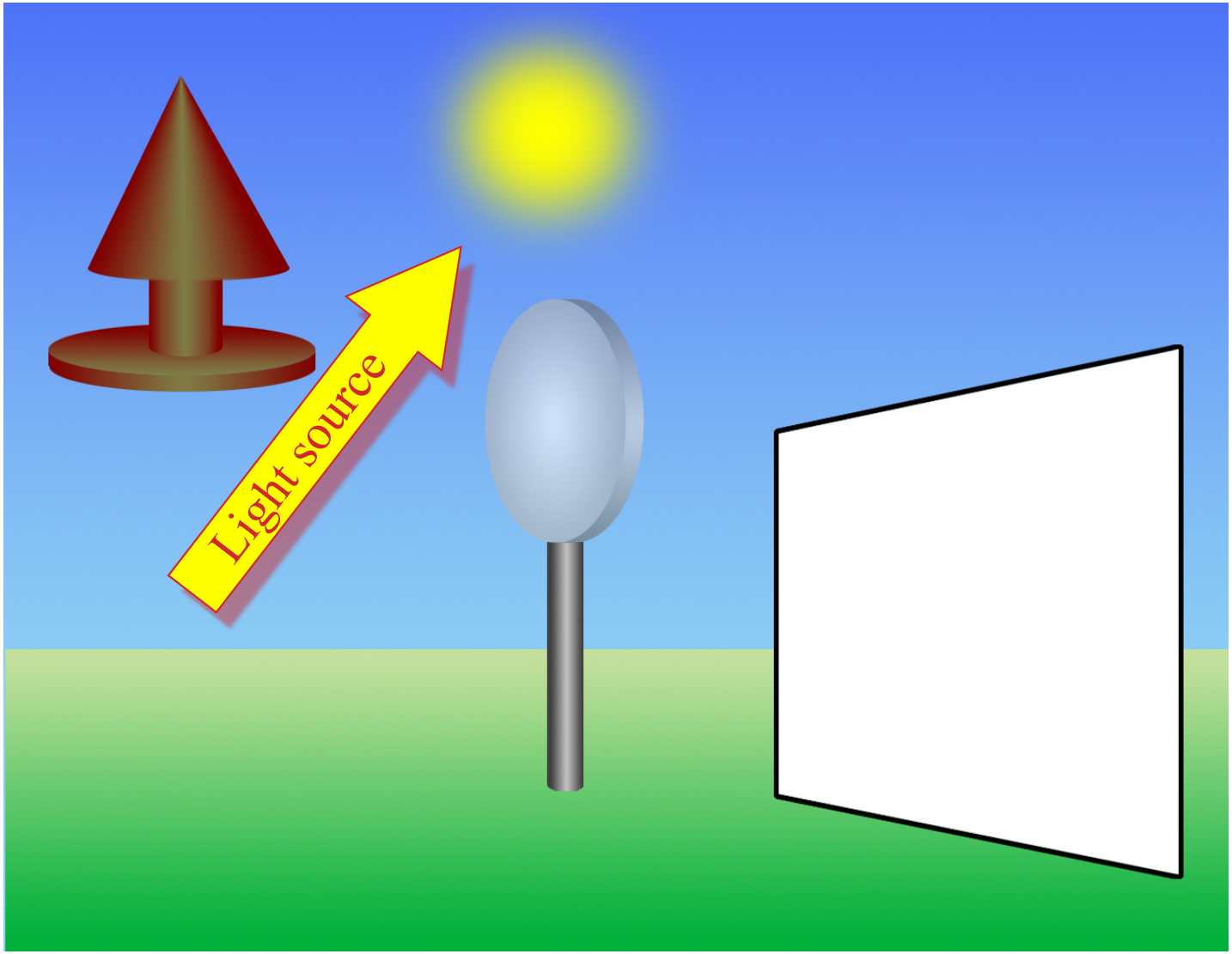
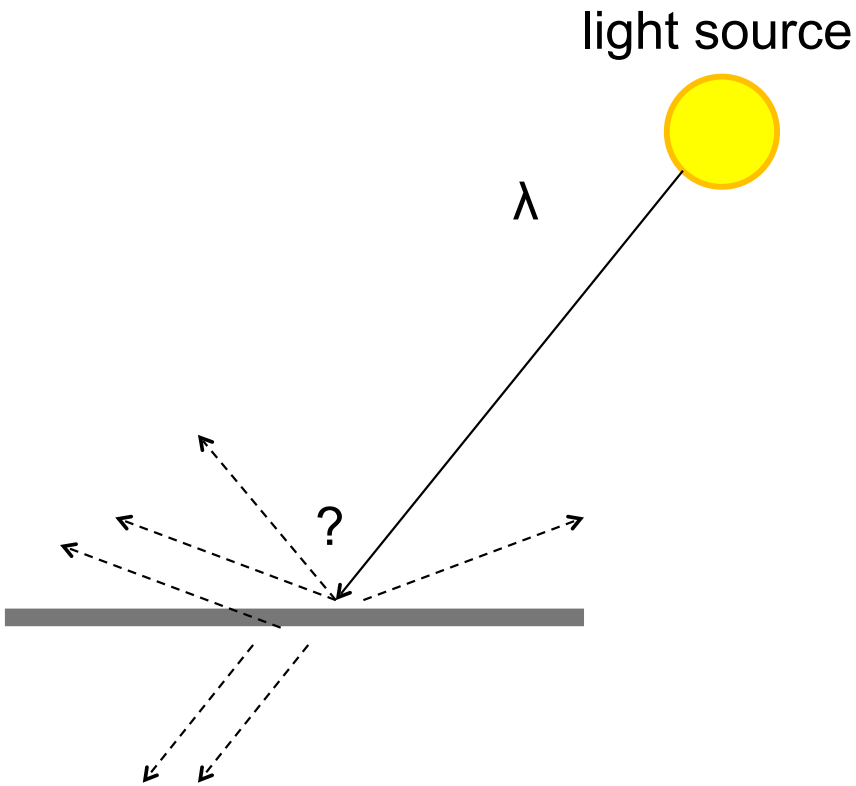


Image formation

Light Source	*i (candela)*
Solar Light	*9000*
Cloudy Sky	*1000*
Moon Light	*0.01*
Internal Light (Office)	*100*
Object	*r*
Black Velvet	*0.01*
White Wall	*0.80*
Silver and other metals	*0.90*
Snow	*0.93*

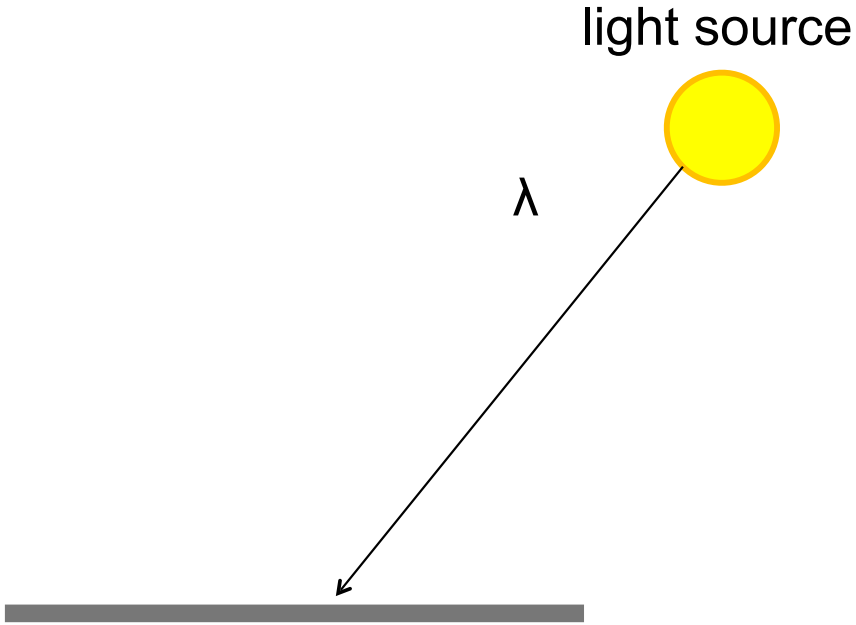
Surface Properties

- Absorption
- Diffusion
- Reflex
- Transparency
- Refraction



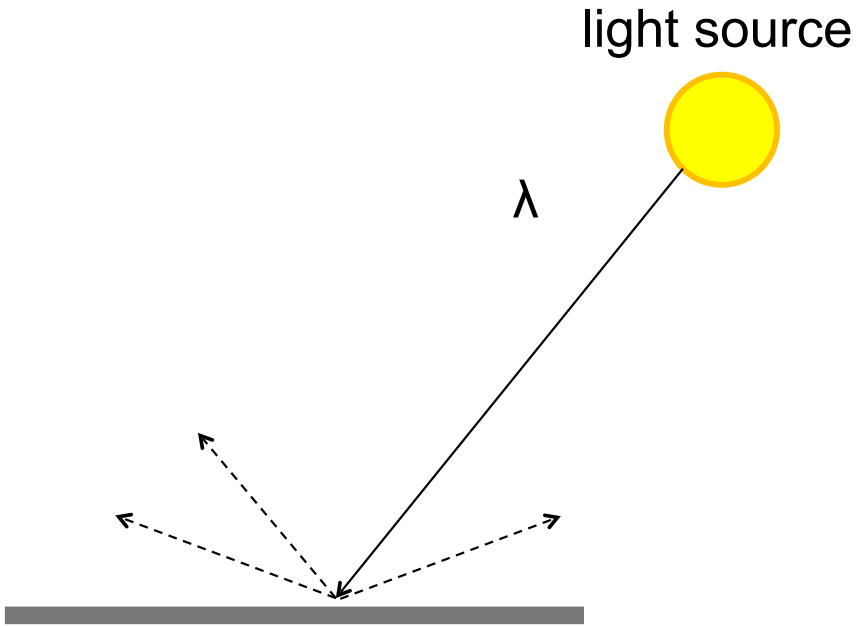
Surface Properties

- **Absorption**
- Diffusion
- Reflex
- Transparency
- Refraction



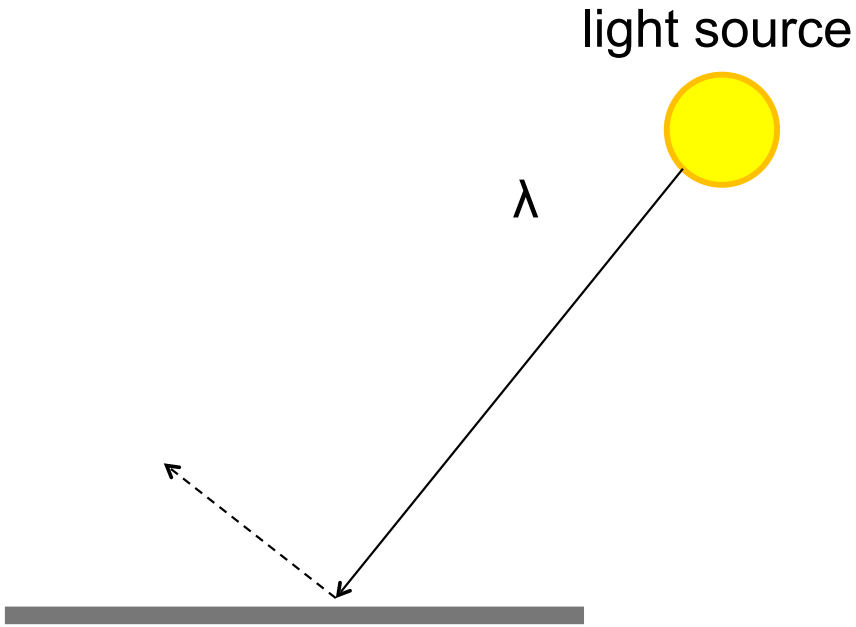
Surface Properties

- Absorption
- **Diffusion**
- Reflex
- Transparency
- Refraction



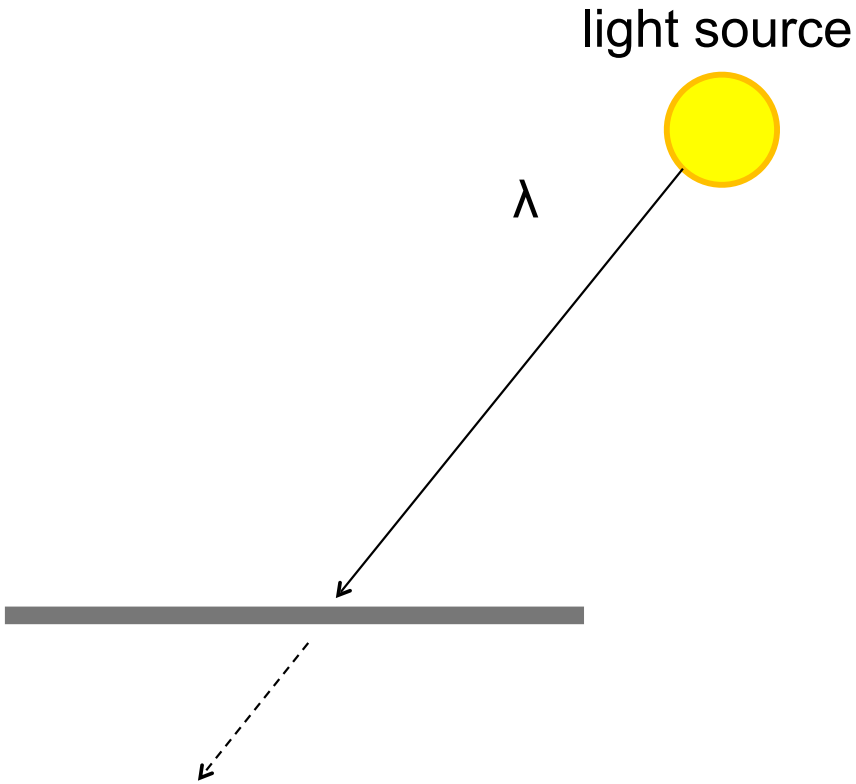
Surface Properties

- Absorption
- Diffusion
- **Reflex**
- Transparency
- Refraction



Surface Properties

- Absorption
- Diffusion
- Reflex
- **Transparency**
- Refraction



Surface Properties

- Absorption
- Diffusion
- Reflex
- Transparency
- **Refraction**

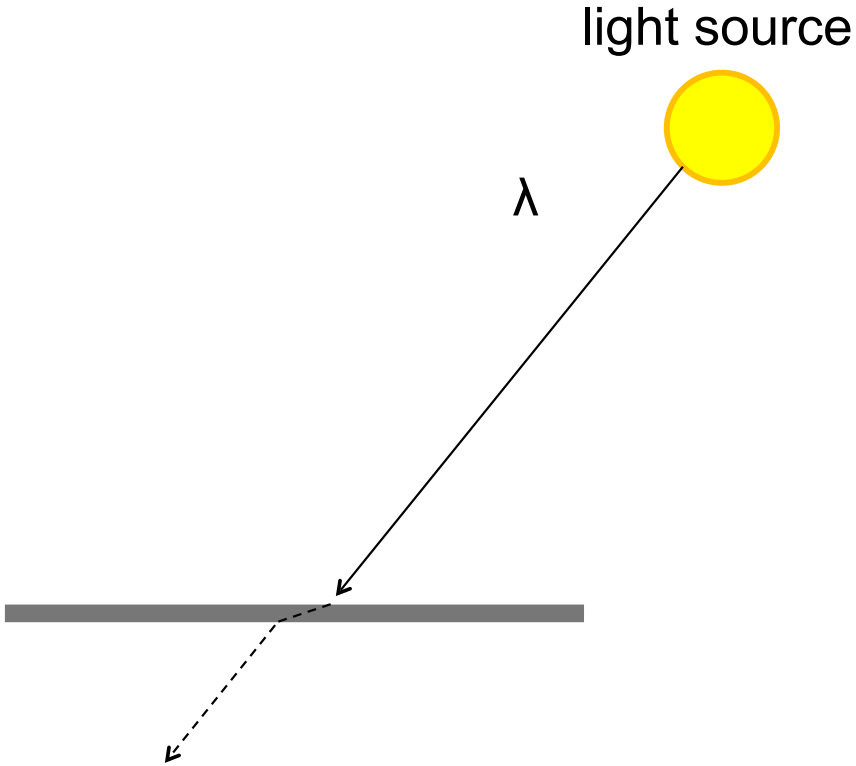


Image acquisition

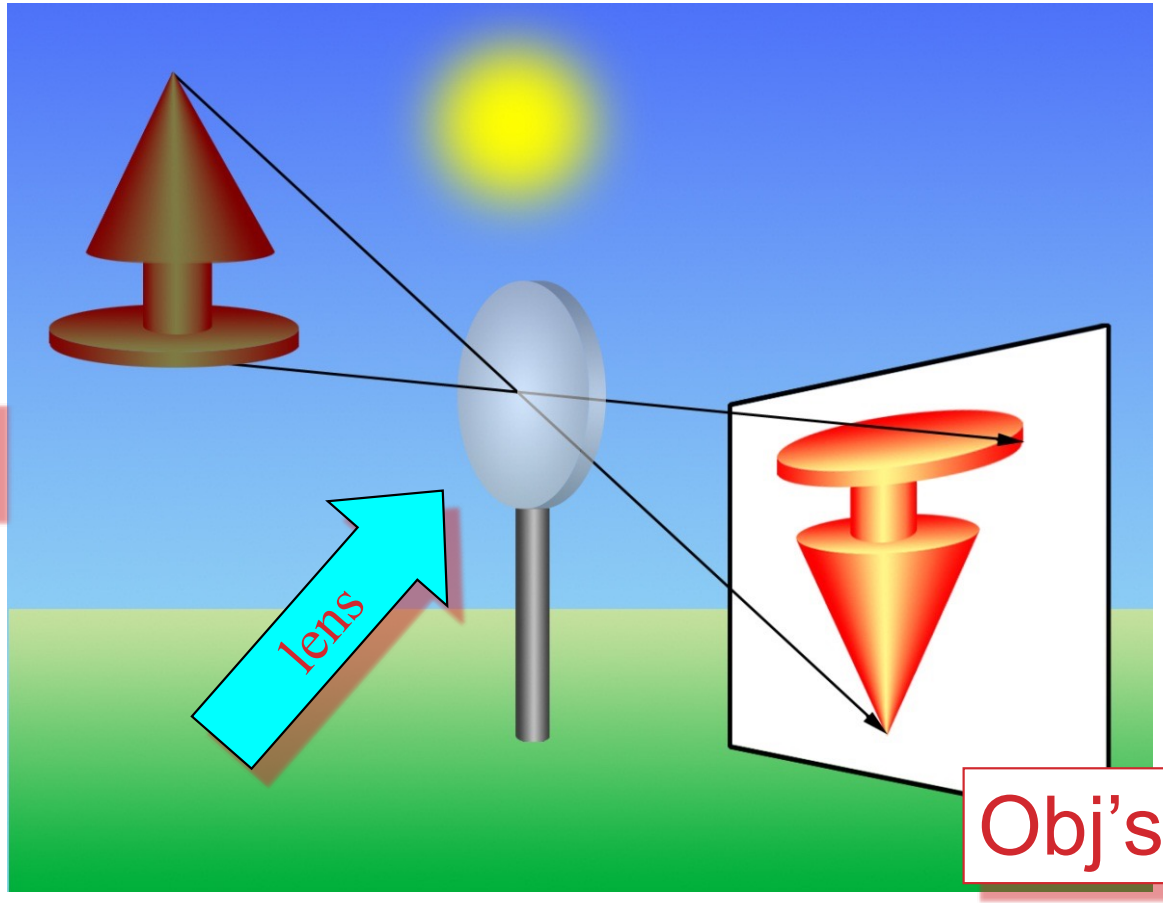
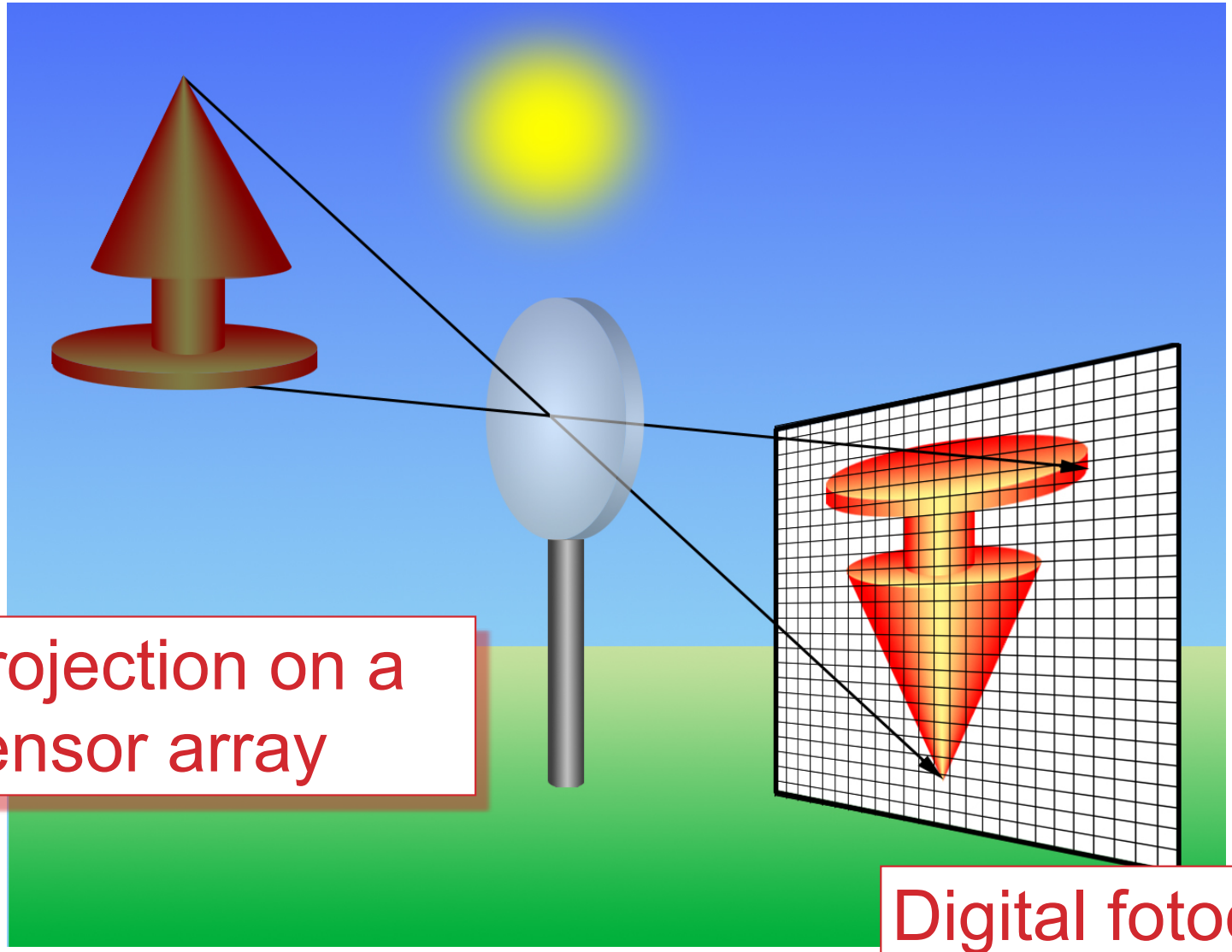
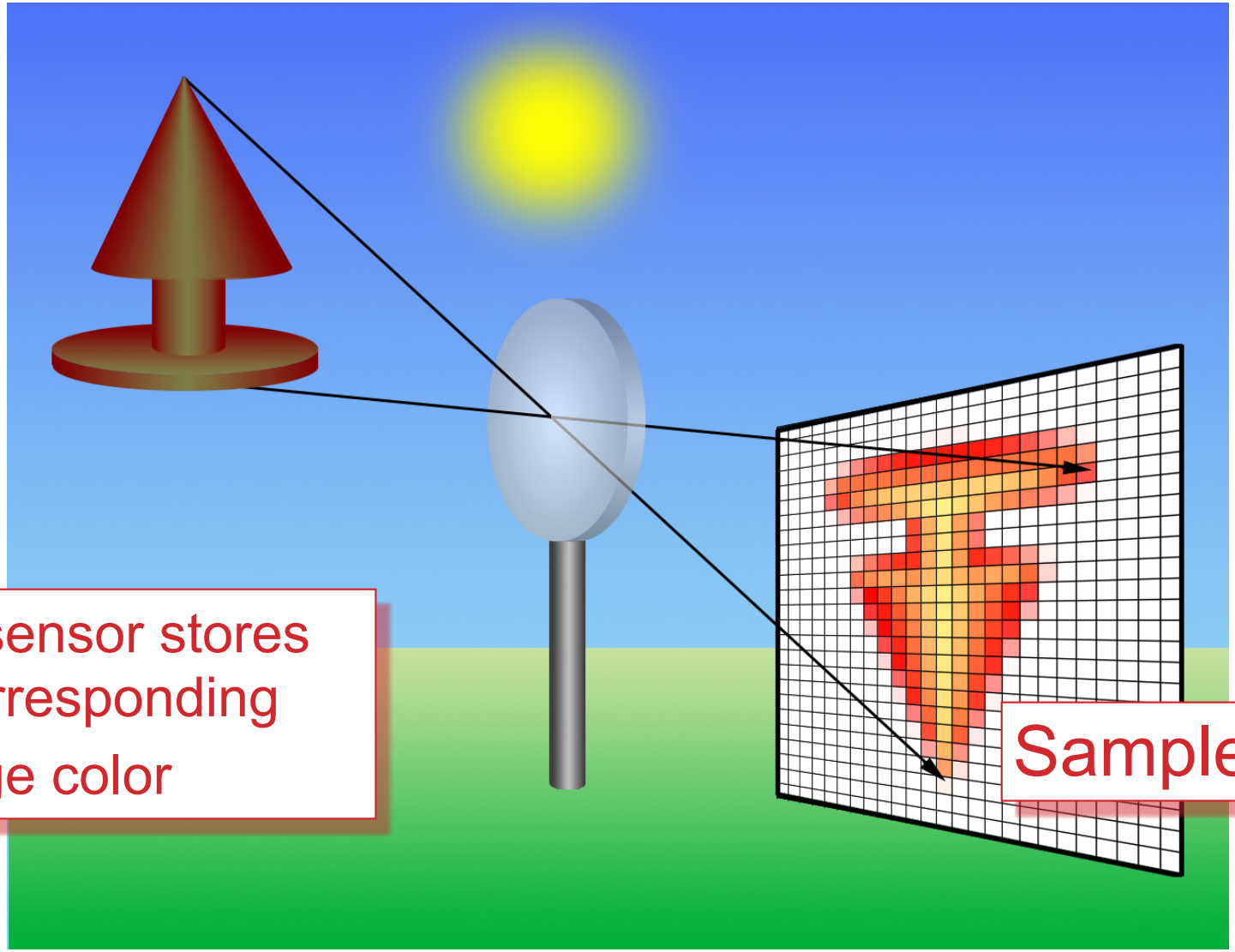


Image acquisition: sampling and quantization



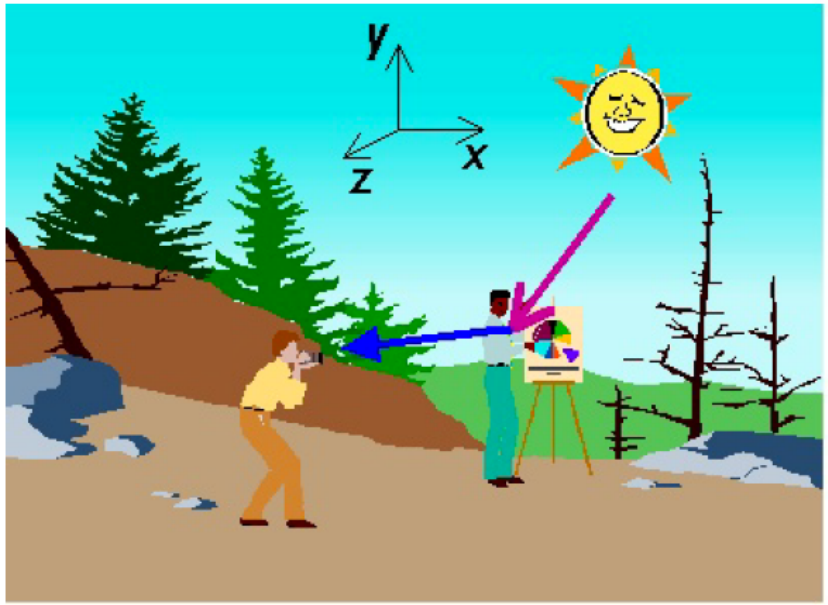
Sampling



MATHEMATICALLY…

Image Formation

- For natural images we need a light source *(λ: wavelength of the source)*
  - $E(x, y, z, \lambda)$: incident light on a point *(x, y, z world coordinates of the point)*
- Each point in the scene has a reflectivity function.
  - $r(x, y, z, \lambda)$: reflectivity function
- Light reflects from a point and the reflected light is captured by an imaging device.
  - $c(x, y, z, \lambda) = E(x, y, z, \lambda) \times r(x, y, z, \lambda)$: reflected light.



$$E(x, y, z, \lambda)$$

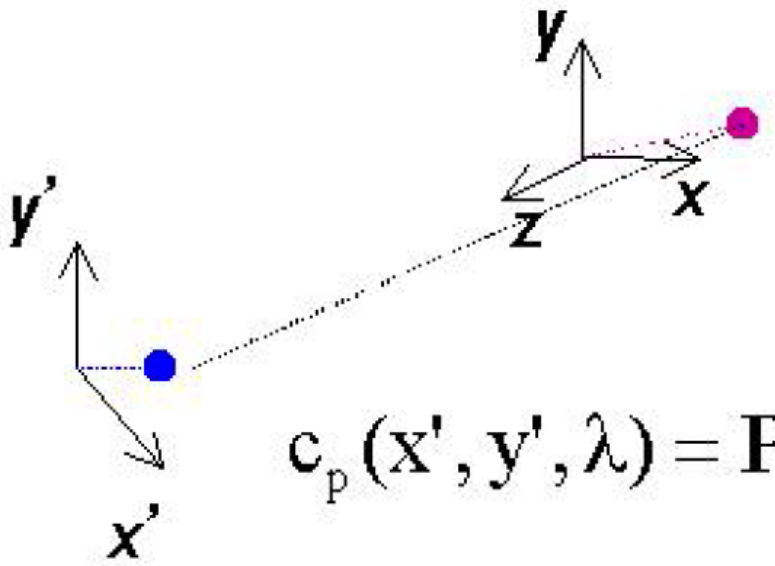
$$c(x, y, z, \lambda) = E(x, y, z, \lambda) \cdot r(x, y, z, \lambda)$$

$$\text{Camera}(c(x, y, z, \lambda)) =$$

Inside the Camera - Projection

Camera$(c(x, y, z, \lambda)) =$ 

- Projection $(\mathcal{P})$ from world coordinates (x, y, z) to camera or image coordinates (x', y') $[c_p(x', y', \lambda) = \mathcal{P}(c(x, y, z, \lambda))]$.

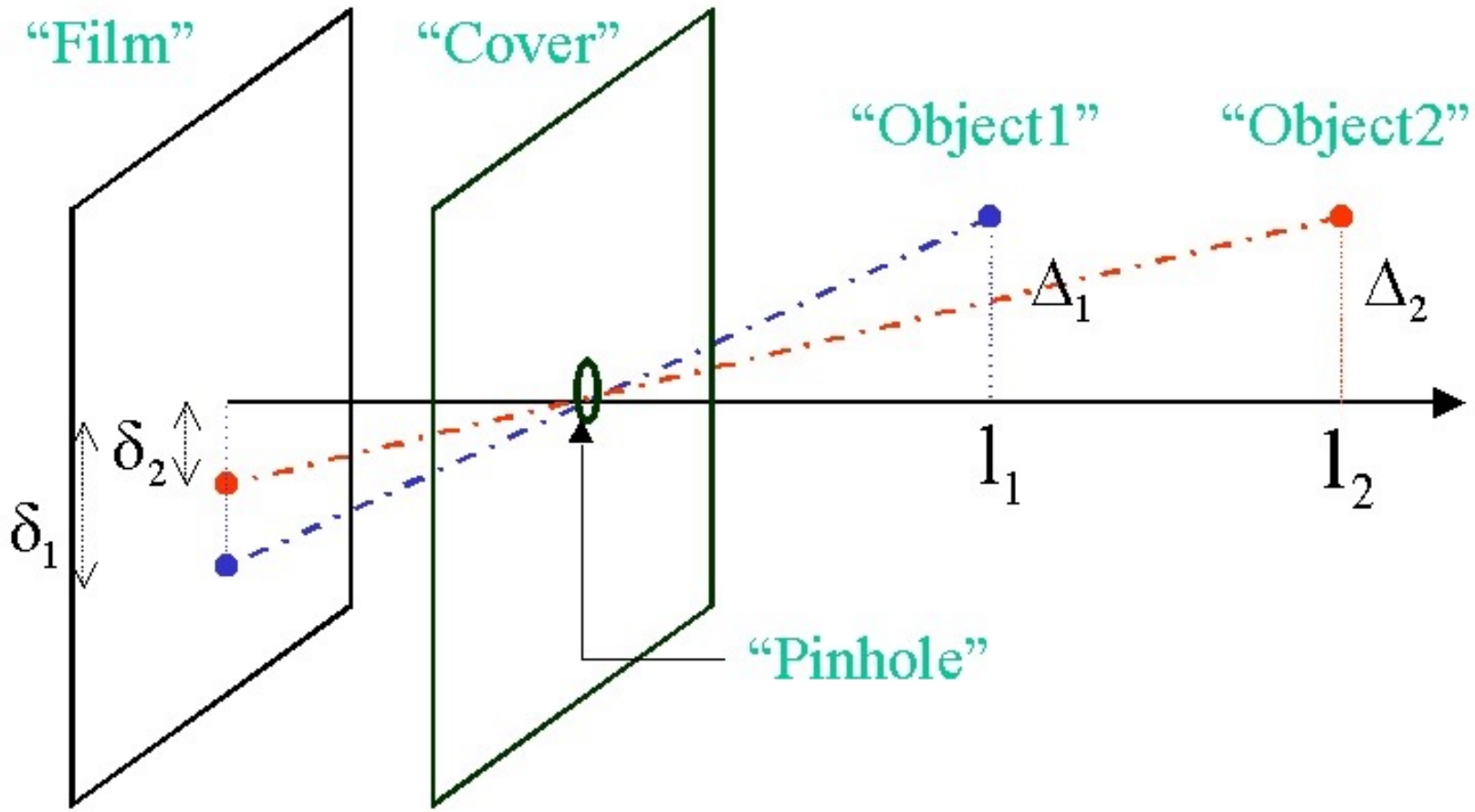


$$c_p(x', y', \lambda) = P(c(x, y, z, \lambda))$$

Projections

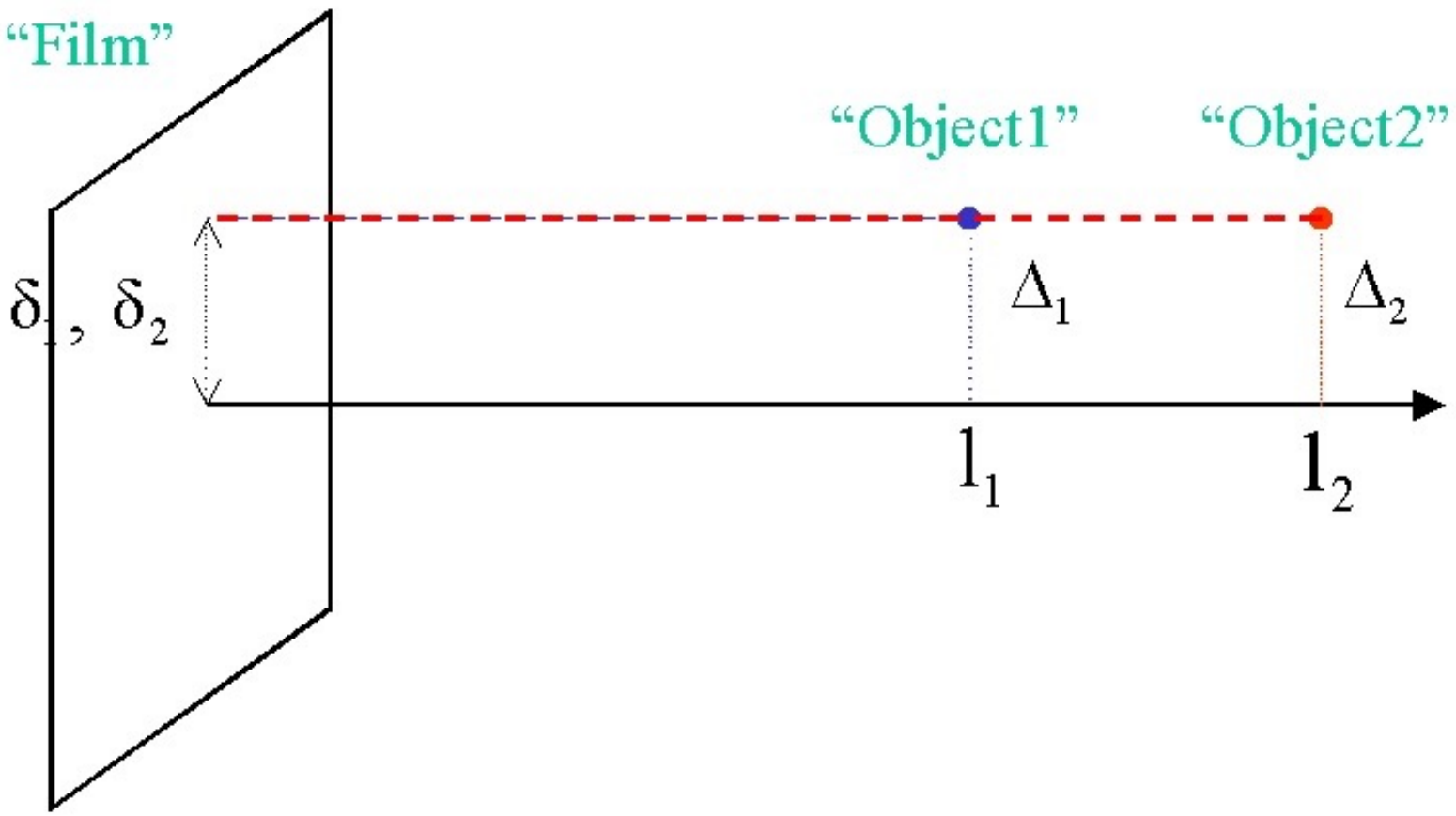
- Two types:
  - **Perspective projection**
    - Closer is bigger.
    - Human eye and camera under this model
    - natural model, but mathematically more complicate
  - **Ortographic projection**
    - Object size independent of the distance from the capturing device
    - unnatural, but mathematically easier

Example - Perspective



Perspective Projection: $\Delta_1 = \Delta_2, \quad l_1 < l_2 \rightarrow \delta_2 < \delta_1.$

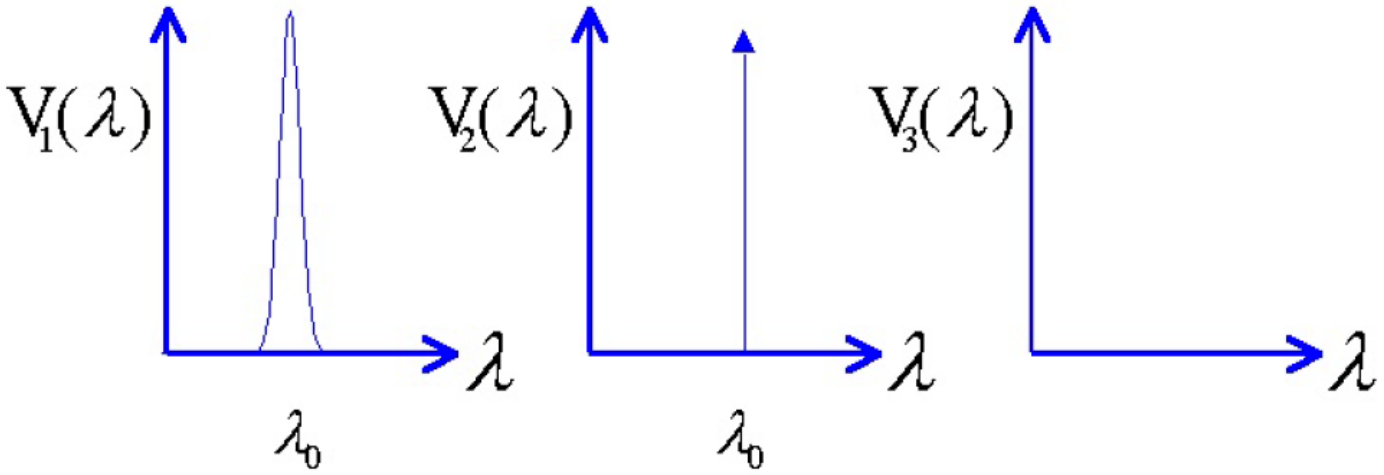
Example - Ortographic



Ortographic Projection: $\Delta_1 = \Delta_2, \quad l_1 < l_2 \rightarrow \delta_2 = \delta_1.$

Inside the Camera - Sensitivity

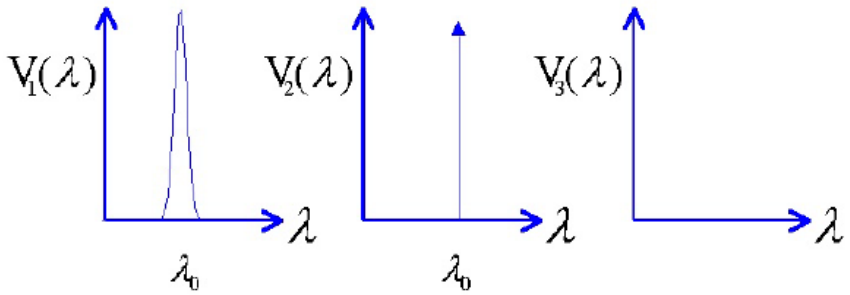
- Once we have $c_p(x', y', \lambda)$ the characteristics of the capture device take over.
- $V(\lambda)$ is the *sensitivity function* of a capture device. Each capture device has such a function which determines how sensitive it is in capturing the range of *wavelengths* (λ) present in $c_p(x', y', \lambda)$.



- The result is an "image function" which determines the amount of reflected light that is captured at the camera coordinates (x', y').

$$f(x', y') = \int c_p(x', y', \lambda) V(\lambda) d\lambda \tag{1}$$

Example



Let us determine the image functions for the above sensitivity functions imaging the same scene:

1. This is the most realistic of the three. Sensitivity is concentrated in a band around λ_0.

$$f_1(x', y') = \int c_p(x', y', \lambda) V_1(\lambda) d\lambda$$

2. This is an unrealistic capture device which has sensitivity only to a single wavelength λ_0 as determined by the delta function. However there are devices that get close to such "selective" behavior.

$$\begin{aligned} f_2(x', y') &= \int c_p(x', y', \lambda) V_2(\lambda) d\lambda = \int c_p(x', y', \lambda) \delta(\lambda - \lambda_0) d\lambda \\ &= c_p(x', y', \lambda_0) \end{aligned}$$

3. This is what happens if you take a picture without taking the cap off the lens of your camera.

$$\begin{aligned} f_3(x', y') &= \int c_p(x', y', \lambda) V_3(\lambda) d\lambda = \int c_p(x', y', \lambda)\ 0\ d\lambda \\ &= 0 \end{aligned}$$

Summary

$$f(x', y') = \int c_p(x', y', \lambda) V(\lambda) d\lambda$$

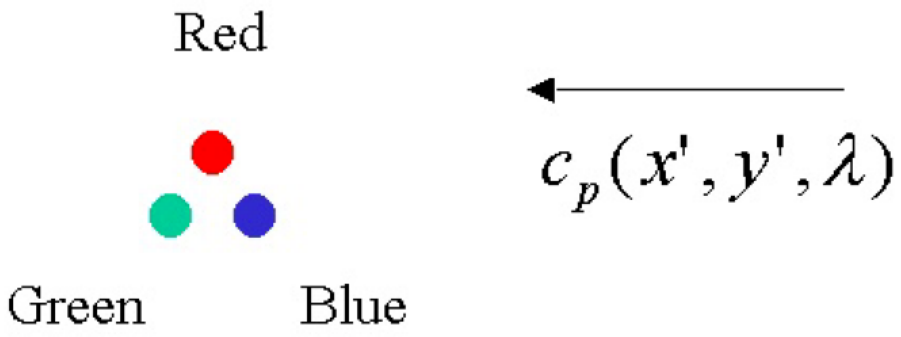
- It is the result of:
  1. Incident light $E(x, y, z, \lambda)$ at the point (x, y, z) in the scene,
  2. The reflectivity function $r(x, y, z, \lambda)$ of this point,
  3. The formation of the reflected light $c(x, y, z, \lambda) = E(x, y, z, \lambda) \times r(x, y, z, \lambda)$,
  4. The projection of the reflected light $c(x, y, z, \lambda)$ from the *three* dimensional world coordinates to *two* dimensional camera coordinates which forms $c_p(x', y', \lambda)$,
  5. The sensitivity function(s) of the camera $V(\lambda)$.

Ok, but…
something is still missing…

Colors !!!

Sensitivity and Color

Camera Sensors



- For a camera that captures color images, imagine that it has *three sensors* at each $(x',\ y')$ with sensitivity functions tuned to the colors or wavelengths red, green and blue, outputting *three* image functions:

$$\begin{aligned} f_{\mathrm{R}}(x', y') &= \int c_p(x', y', \lambda) V_{\mathrm{R}}(\lambda) d\lambda \\ f_{\mathrm{G}}(x', y') &= \int c_p(x', y', \lambda) V_{\mathrm{G}}(\lambda) d\lambda \\ f_{\mathrm{B}}(x', y') &= \int c_p(x', y', \lambda) V_{\mathrm{B}}(\lambda) d\lambda \end{aligned}$$

- These three image functions can be used by display devices (such as your monitor or your eye) to show a "color" image.

The end of this long story...

- The image function $f_C(x', y')$ $(C = R,\ G,\ B)$ is formed as:

$$f_C(x', y') = \int c_p(x', y', \lambda) V_C(\lambda) d\lambda \tag{2}$$

- It is the result of:
  1. Incident light $E(x, y, z, \lambda)$ at the point (x, y, z) in the scene,
  2. The reflectivity function $r(x, y, z, \lambda)$ of this point,
  3. The formation of the reflected light $c(x, y, z, \lambda) = E(x, y, z, \lambda) \times r(x, y, z, \lambda)$,
  4. The projection of the reflected light $c(x, y, z, \lambda)$ from the *three* dimensional world coordinates to *two* dimensional camera coordinates which forms $c_p(x', y', \lambda)$,
  5. The sensitivity function(s) of the camera $V_C(\lambda)$.

Uff, still not the end:

Digitalization !!!

Digital image formation

- The projected image is in a **continuum** domain and co-domain:

$$(x', y') \in \mathbb{R}^2 \text{ and } f(x', y') \in \mathbb{R}$$

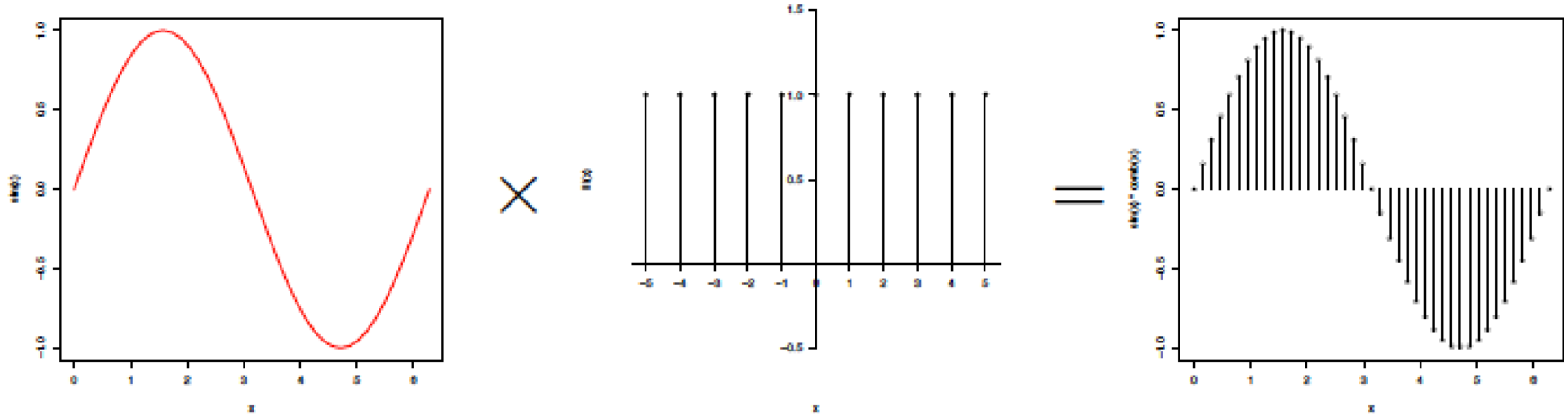
- Digital computers cannot process parameters/functions that vary in a continuum.
- We need to **discretize**:
- **Sampling:**

$$(x', y') \rightarrow (x'_i, y'_j) \subset \mathbb{N}^2 \quad s.t. \quad (i = 0, ..., N-1, \quad j = 0, ..., M-1)$$

- **Quantization:**

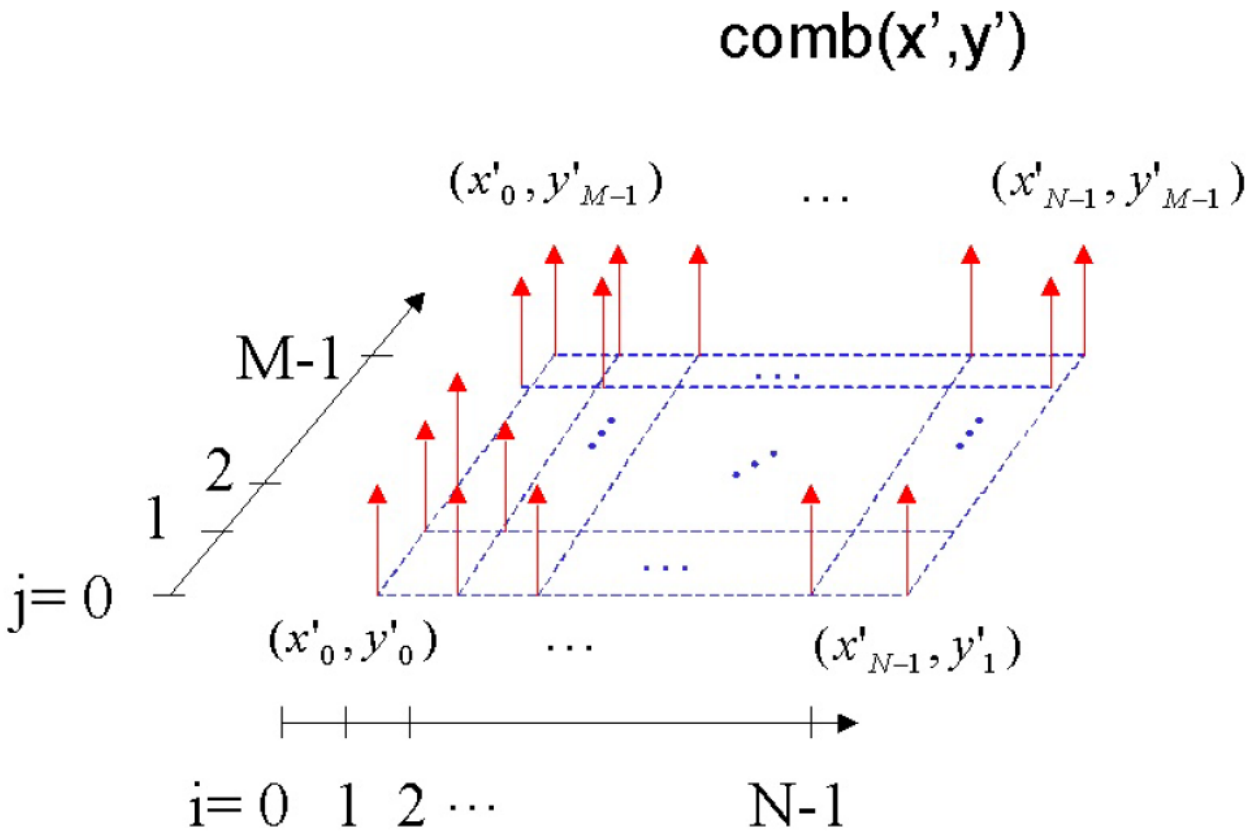
$$f(x'_i, y'_j) \rightarrow \hat{f}(x'_i, y'_j) \subset \mathbb{N}$$

Spatial Sampling 1D



We can think of spatial sampling as multiplication of a continuous signal with a comb function.

Sampling 2D



$$comb(x', y') = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \delta(x' - i\Delta_x, y' - j\Delta_y) \tag{3}$$

- Obtain sampling by utilizing $f_C(x', y') \times comb(x', y')$.

Sampling 2D

$$f(x'_i, y'_j) = f(x', y') \times comb(x', y')$$

for simplicity:

$$f(x'_i, y'_j) = f(i, j)$$

Quantization

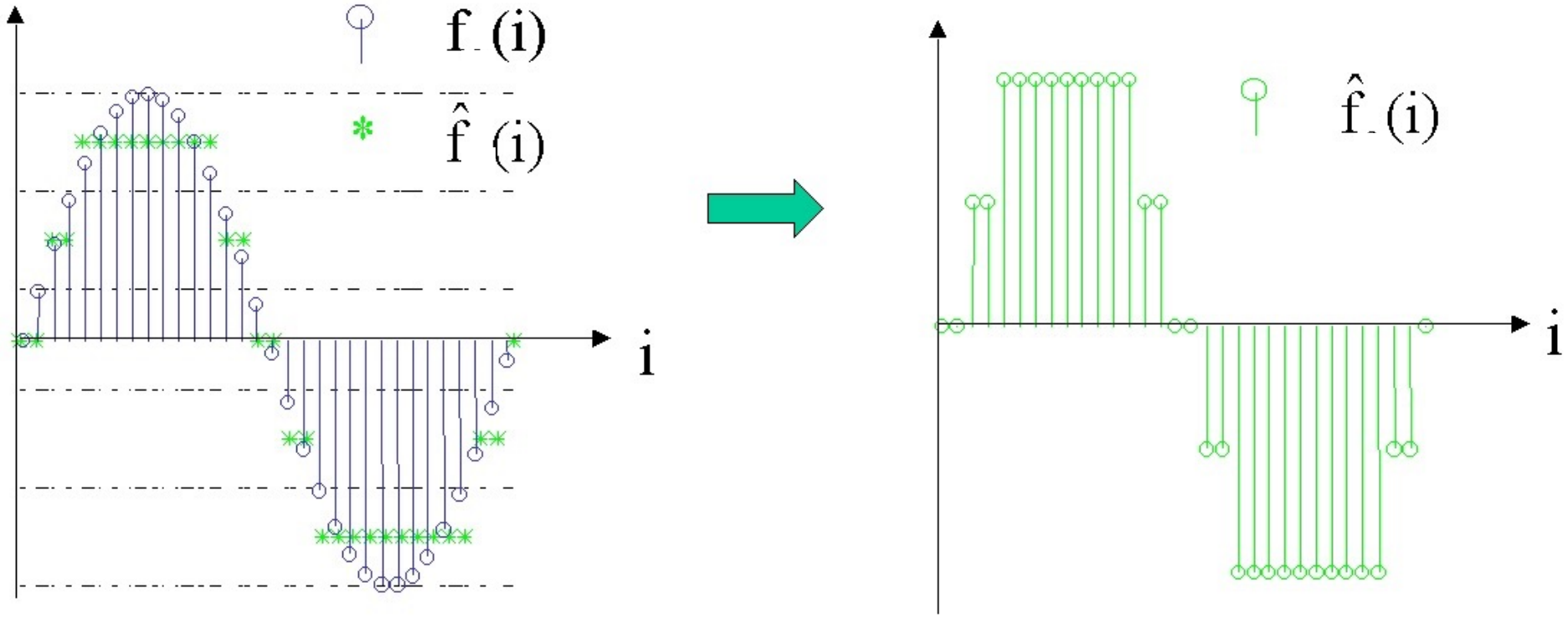
- discretize the values of f(i,j) to *P* levels:
  - Let $\Delta_Q = \frac{f_{max} - f_{min}}{P}$
  - Then $\hat{f}(i,j) = Q(f(i,j))$
  - Where:

$$Q(f(i,j)) = f_{min} + (k + 1/2)\Delta_Q$$

$$\iff \quad f_{min} + k\Delta_Q \leq f(i,j) < f_{min} + (k+1)\Delta_Q$$

$$\text{for} \quad k = 0, ..., P-1$$

Example



Images as Discrete Functions

After spatial sampling and quantization, an image is a discrete function. The image domain Ω is now discrete:

$$\Omega \subset \mathbb{N}^2,$$

and so is the image range:

$$I : \Omega \to \{1, \ldots, \mathsf{P}\},$$

where $\mathsf{P} \in \mathbb{N}$.

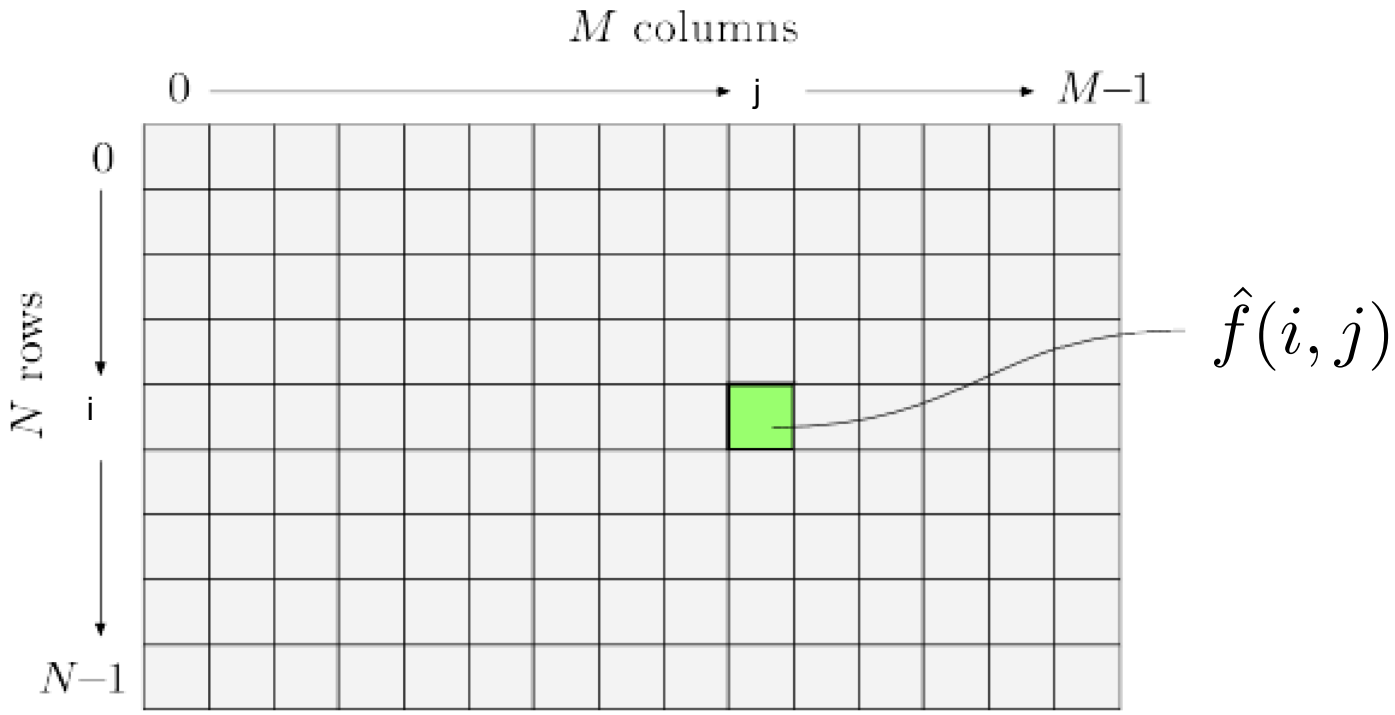
Typically: $P = 2^8 = 256$ and $log_2(P) = 8$ bit quantization.

Summary

- "Let there be light" $\rightarrow$ incident light $\rightarrow$ reflectivity $\rightarrow$ reflected light $\rightarrow$ projection $\rightarrow$ sensitivity $\rightarrow$ $f\ (x', y')$.
- Sampling: $f\ (x', y') \rightarrow f\ (i, j)$.
- Quantization: $f_{\cdot}\,(i, j) \rightarrow \hat{f}_{\cdot}(i, j) \in \{0, \ldots, 255\}$.

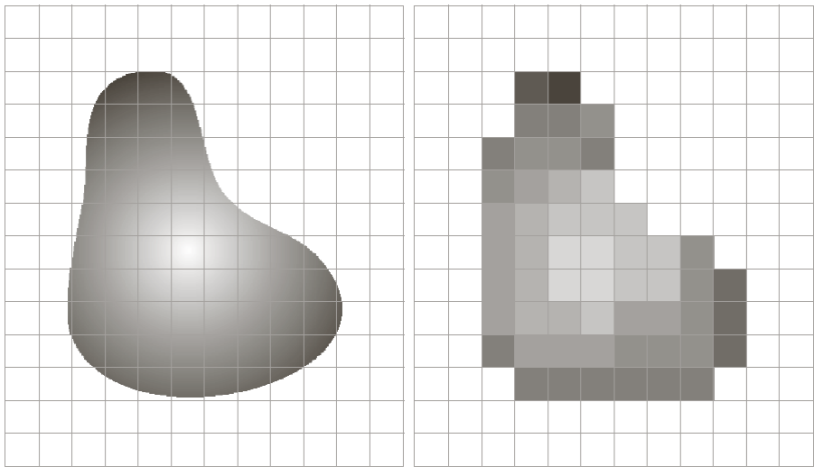
Representing an Image

The data structure for an image is simply a 2D array of values. The values in the array can be any datatype (bit, byte, int, float, double, etc.)



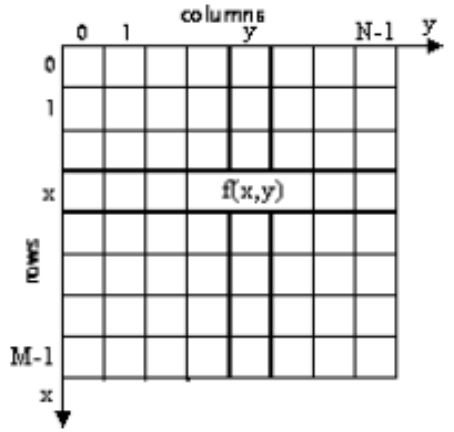
Pixel

- Pixel:
  - single value on the sampled grid
  - It does not correspond to a point but rather to an area, the smallest treatable



a b

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



Memory

- Pixel number: (# rows * # columns) = (M*N)
- k: image resolution : k = log(L), (L: intensity levels)
- N. bit per img: b = M * N * *k*

In the hypothesis of N=M ➔ $b=N^2 * k$

N/k	1 ($L = 2$)	2 ($L = 4$)	3 ($L = 8$)	4 ($L = 16$)	5 ($L = 32$)	6 ($L = 64$)	7 ($L = 128$)	8 ($L = 256$)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

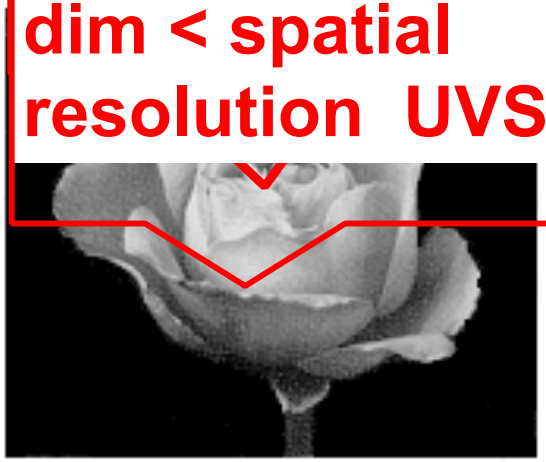
Spatial resolution

- **Spatial resolution**: the smallest distinguishable detail in an image
- Expressed as :
  - **ppi** : Pixel Per Inch
  - **dpi** : Dot Per Inch
  - **M*N** : does not express the relationship between the number of pixels and the detail level of the original scene

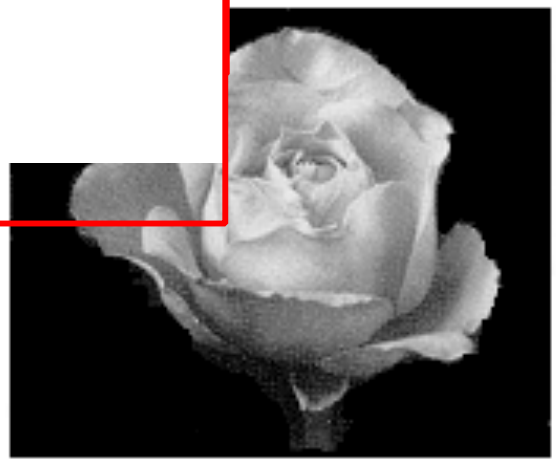
Effects of the spatial resolution

- nensions, varying the
  ≠ spatial resolution)

1. High resolution
2. Image perceived as continuum when pixel dim < spatial resolution UVS



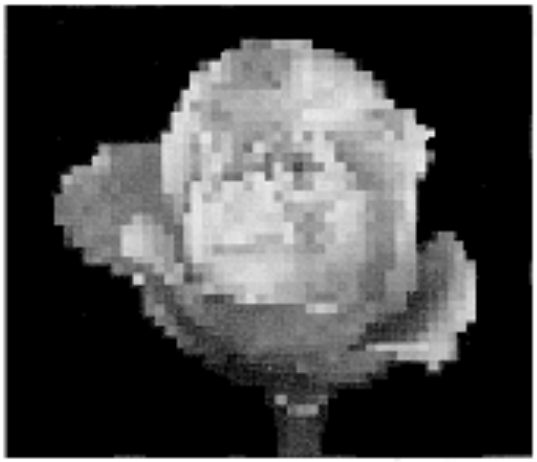
1024 x 1024



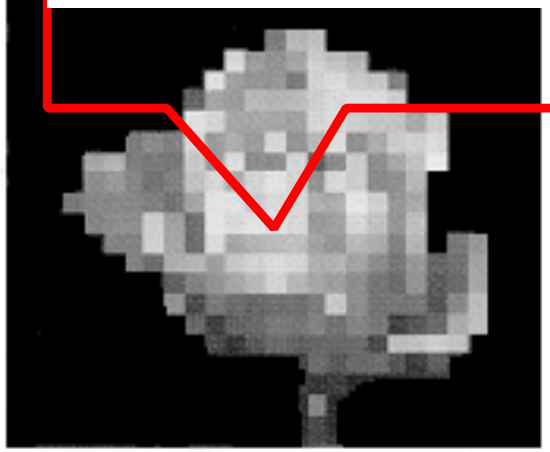
512 x 512

1. Low resolution
2. Visible pixel contours

128 x 128



64 x 64



32 x 32

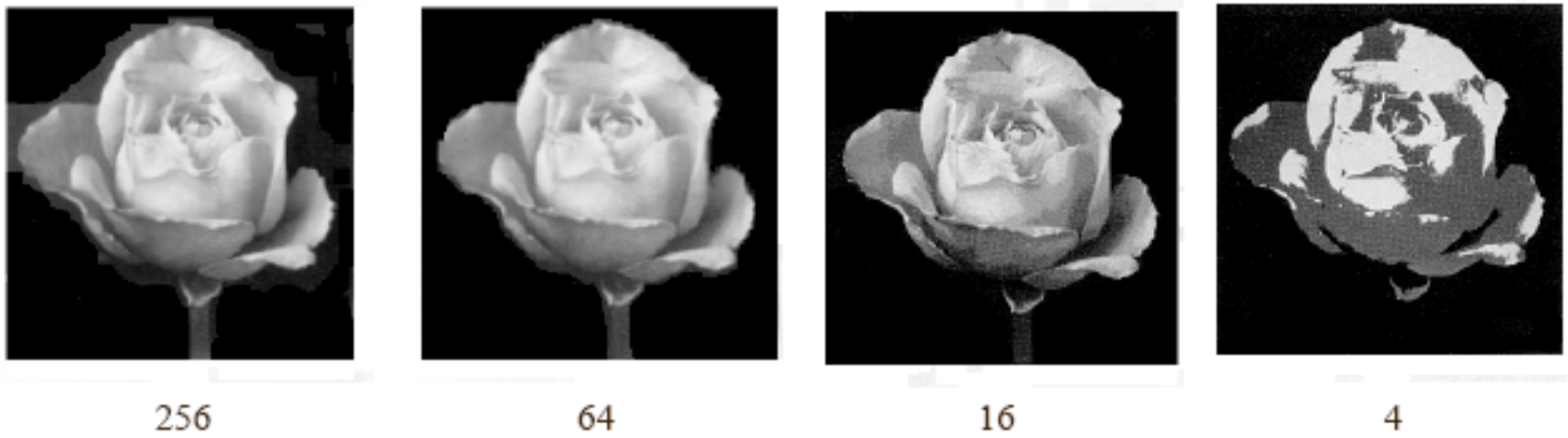
Effects of the spatial resolution

- If the pixel dimensions do not change, resolution variations cause the image size variation:

Quantization

Quantization: fin the number of gray levels

N.B: Better if greater than 50 gray levels



Resolution

- what the best?
- it depends on the application:
  - *visualization/web*: 72 dpi (dot per inch), that is the monitor resolution
  - *printing*: up to 600 dpi…

- If I need to pass from low resolution to high resolution? **Interpolation**, even if it is just an approximation…

Interpolation or Zooming

- Two phases:
  - grid creation
    - Es: zoom of 1.5 times
      original img (500 x 500) ➔ new img: (750 x 750)
  - gray level assignment to each new pixel

Zooming, Nearest-Neighbor interpolation

- The value of a pixel in the output image (D) is set equal to the value of the closest pixel in the input image (S):
  - (i_d, j_d) = position of the pixels in output (integer)
  - (x_s, y_s) = position of the pixels in input (real)

  $D(i_d, j_d) = S(round(x_s), round(y_s))$

  - n. of input pixel used for the interpolation: 1
  - Fastest method, but less precise

Zooming,
Nearest-Neighbor interpolation

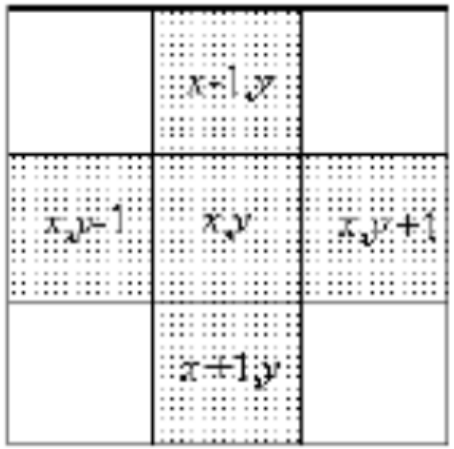
- Es monodimensional:
  - zooming of 1,5 times
  - S: mono-dimensional input image of 6 pixels:
    - S= [10 14 22 26 77 44]
  - D: output image of 9 pixels

i_d	1	2	3	4	5	6	7	8	9
X_s	0,7	1.3	2	2.7	3.3	4	4.7	5.3	6

- The value of D(i_d) = S(round(x_s))
- Es: D(4) = S(round(2,7)) = S(3) = 22

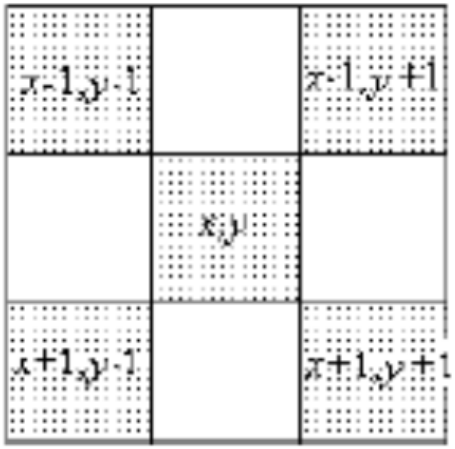
Relationship among pixels

- Given a pixel *p* of coordinates *(x,y)*, the four neighbors (in horizontal and vertical) have coordinates:
  (x+1,y), (x-1,y), (x, y+1),(x,y-1)
  constituting the set $N_4(p)$ of the 4 neighbors of *p.*



$N_4(p)$

- The four diagonal neighbors of *p* have coordinates
  (x-1, y-1), (x-1, y+1), (x+1,y-1), (x+1, y+1),
  constituting the set $N_D(p)$.



$N_D(p)$

- *The set of the 8 neighbors of p is given*

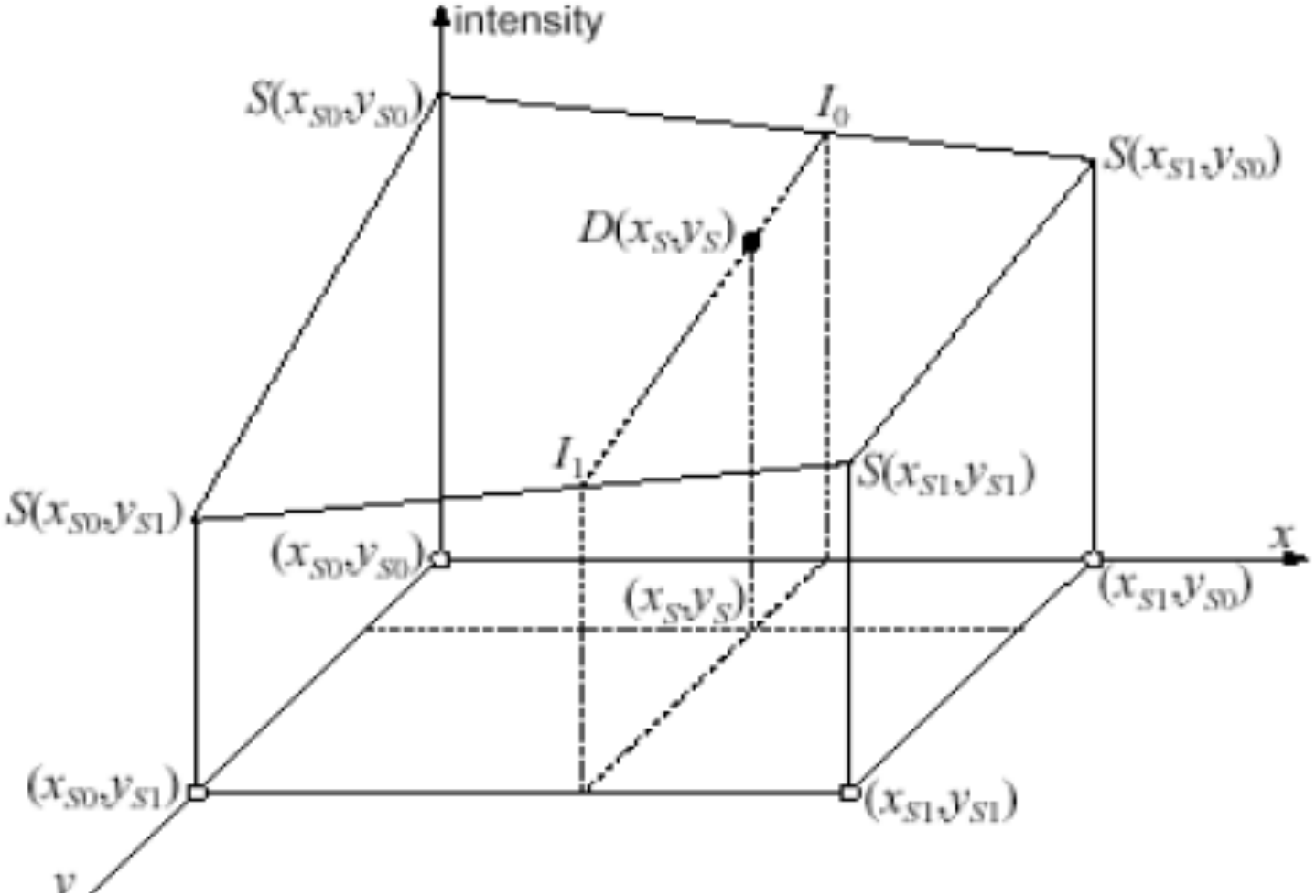
$$N_8(p) = N_4(p) \cup N_D(p)$$

Zooming, Bilinear Interpolation (1/3)

- The output values in D are obtained linearly interpolating before along the lines and then along the columns (or vice versa), considering the 4 closest pixels in the input image S:
  - (i_d, j_d) = position of the output pixels (integer)
  - (x_s, y_s)= position of the input pixels (real)
- computation of the 4 closest pixels in the input image:

$$
\begin{aligned}
x_{s0} &= \mathrm{int}(x_s) \\
x_{s1} &= x_{s0} + 1 \\
y_{s0} &= \mathrm{int}(y_s) \\
y_{s1} &= y_{s0} + 1
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
&(x_{s0}, y_{s0}) \\
&(x_{s1}, y_{s0}) \\
&(x_{s0}, y_{s1}) \\
&(x_{s1}, y_{s1})
\end{aligned}
$$

Zooming,
Bilinear Interpolation (2/3)



- n. of input pixel used for the interpolation: 4
- Computationally more expensive than NN, but more precise

Zooming, Bilinear Interpolation (3

Weighted average

- Linear interpolation along lines (X axis) producing 2 intermediate results:

$$I_0 = S(x_s, y_{s0}) = S(x_{s0}, y_{s0}) * (x_{s1} - x_s) + S(x_{s1}, y_{s0}) * (x_s - x_{s0})$$

$$I_1 = S(x_s, y_{s1}) = S(x_{s0}, y_{s1}) * (x_{s1} - x_s) + S(x_{s1}, y_{s1}) * (x_s - x_{s0})$$

- Then compute the value of the pixel in position (x_s,y_s) in the input image, corresponding to the integer coordinates (i_d,j_d) in the output image, linearly interpolating along the columns (y axis)

$$D(i_d, j_d) = S(x_s, y_s) = I_0 * (y_{s1} - y_s) + I_1 * (y_s - y_{s0})$$

Zooming,
Bicubic Interpolation

- Similar to the bilinear interpolation
- Differences:
  - Interpolation of **CUBIC polynomials**
  - Each output value is based on the value of **16 pixels** in the input image (S)

- The output value (D) is worked out interpolating the 16 closest pixels on the input images by cubic polynomials. As before the method is applied before along lines and then along columns.

- More computational expensive but more precise

Zooming,
Bicubic Interpolation