OUTLINE

- 1D digital signal processing
  - Convolution, Fourier Transform, Nyquist rate
- Image formation
  - Pinhole camera
  - Visual sensors
  - Digital image
- Images
  - Binary Images
  - Indexed Images (Grayscale Images)
  - Truecolor Images
- Digital image operations

1D SIGNAL PROCESSING

- The convolution operator can be thought as a general moving average.
  - Continuous domain:

$$(f * g)(t) = \int f(\tau)g(t-\tau)d\tau$$

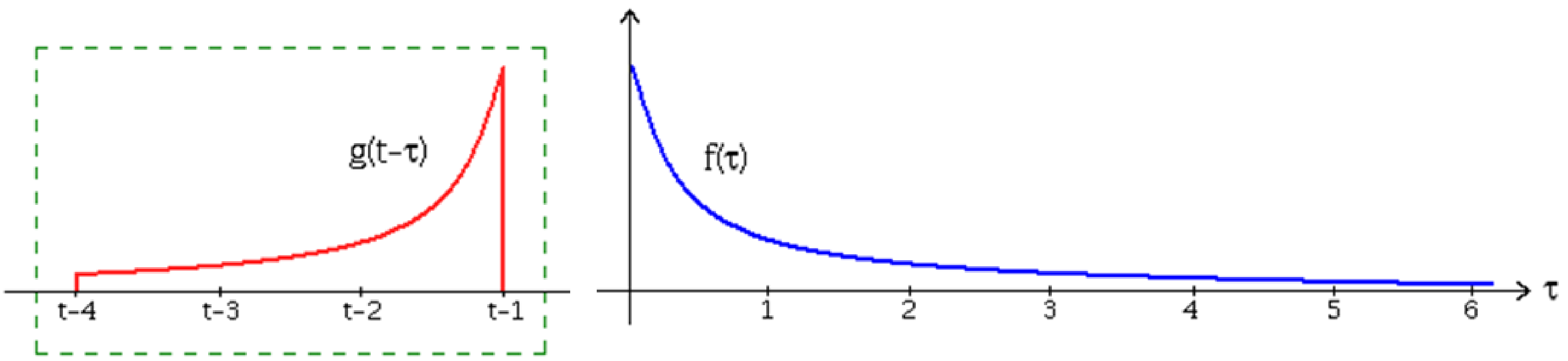
  - Discrete domain:

$$(f * g)(m) = \sum_{n} f(n)g(m-n)$$

- Convolution can describe the effect of an LTI (Linear Time-Invariant) system on a signal.

SIGNAL PROCESSING

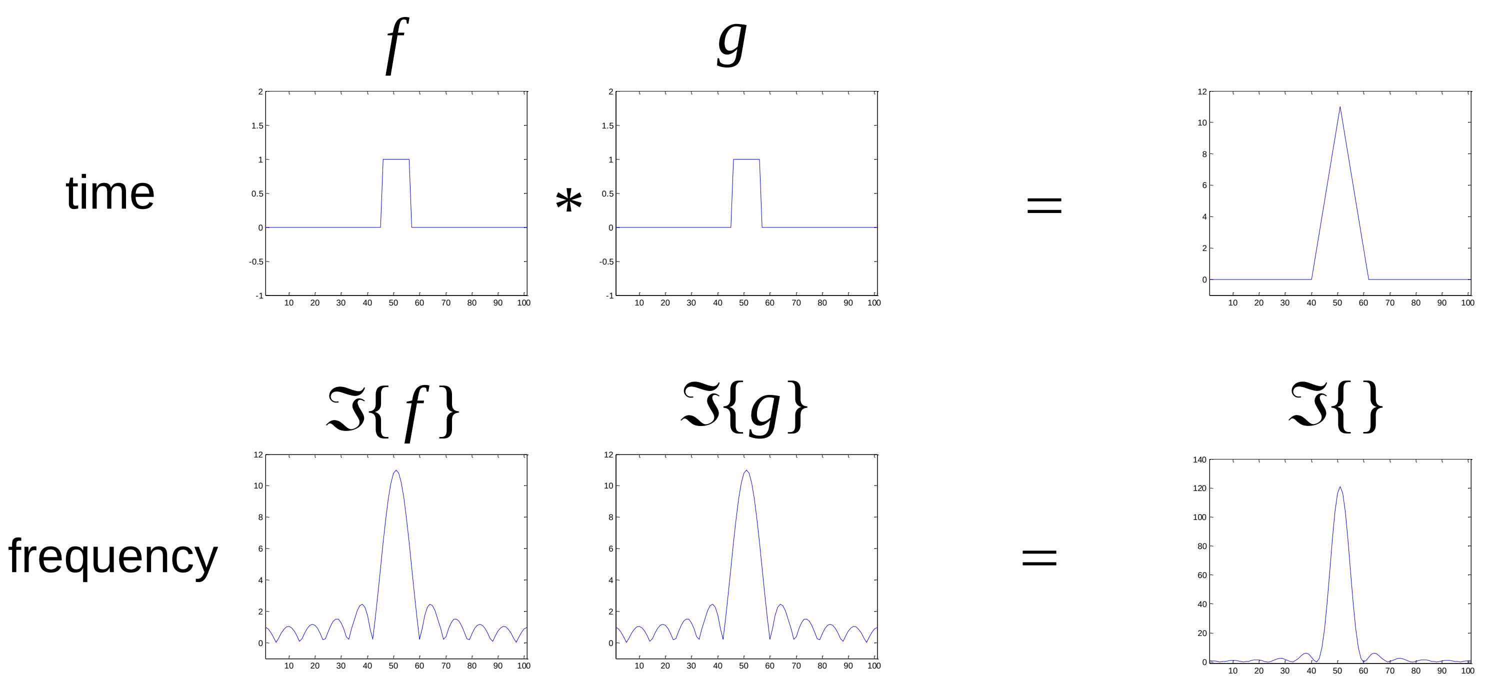
- Basic steps of the convolution:
  - Flip (reverse) one of the digital functions.
  - Shift it along the time axis by one sample.
  - Multiply the corresponding values of the two digital functions.
  - Summate the products from step 3 to get one point of the digital convolution.
  - Repeat steps 1-4 to obtain the digital convolution at all times that the functions overlap.



SIGNAL PROCESSING

- Frequency Domain (we use the FFT):
  - Note that convolution in the time domain is equivalent to multiplication in the frequency domain (and vice versa):

$$\Im\{(f * g)(t)\} = \Im\{f\}\Im\{g\} = F(\omega)G(\omega)$$

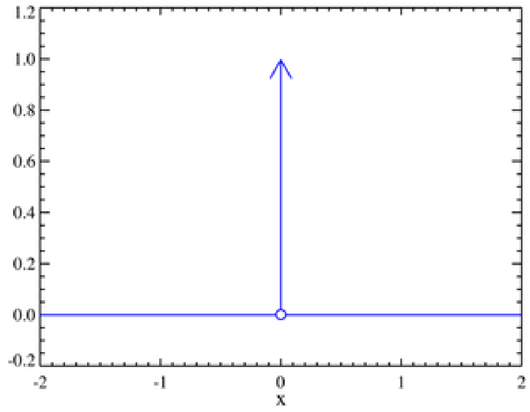


See conv_fft.m

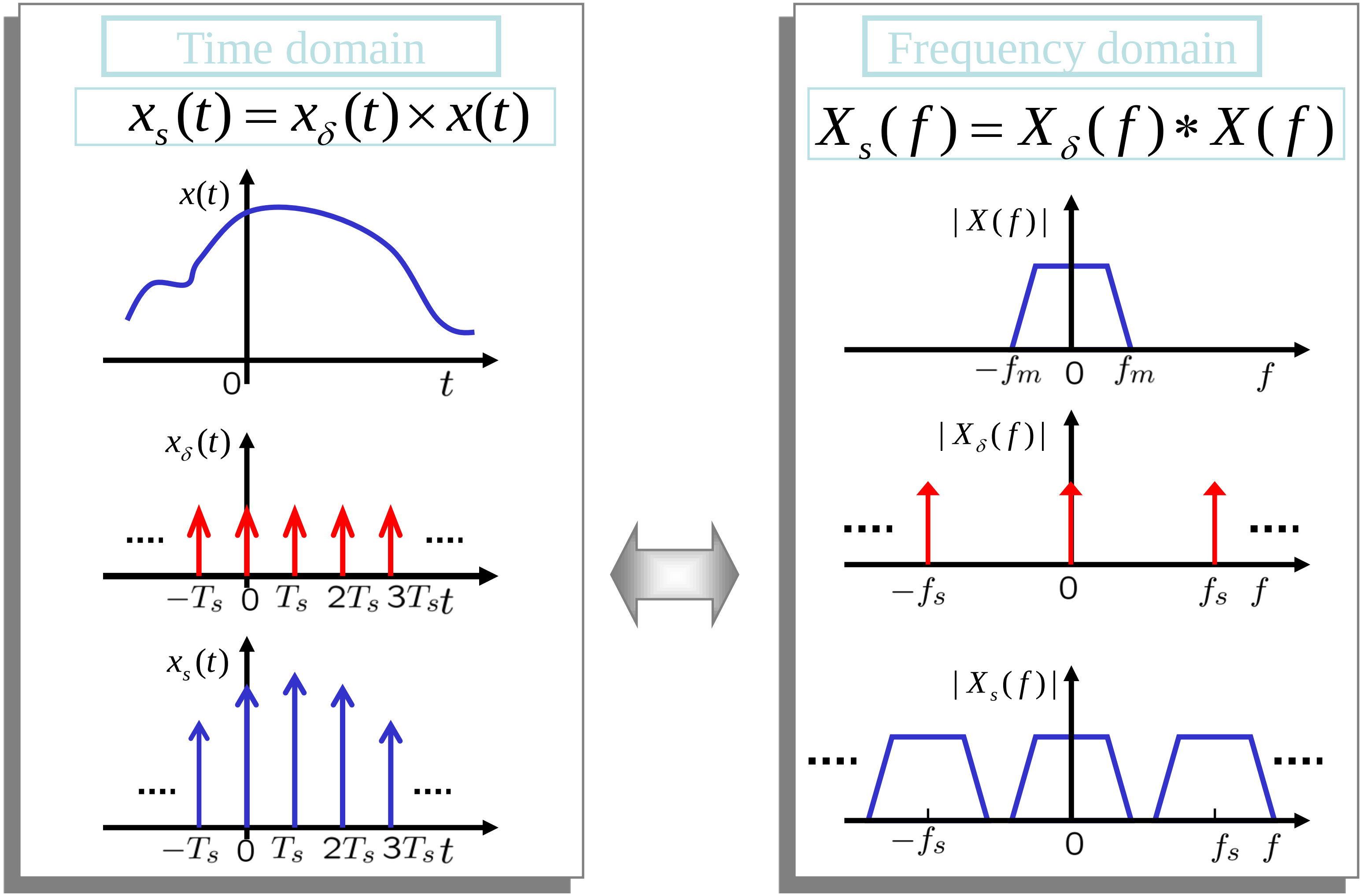
SIGNAL PROCESSING

- Often, the result of the Fourier Transform needs to be expressed in terms of the *delta function*

$$\delta(x) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad \int_{-\infty}^{\infty} \delta(x)\, dx = 1. \quad \int_{-\infty}^{\infty} f(x)\, \delta(x)\, dx = f(0)$$



SIGNAL PROCESSING

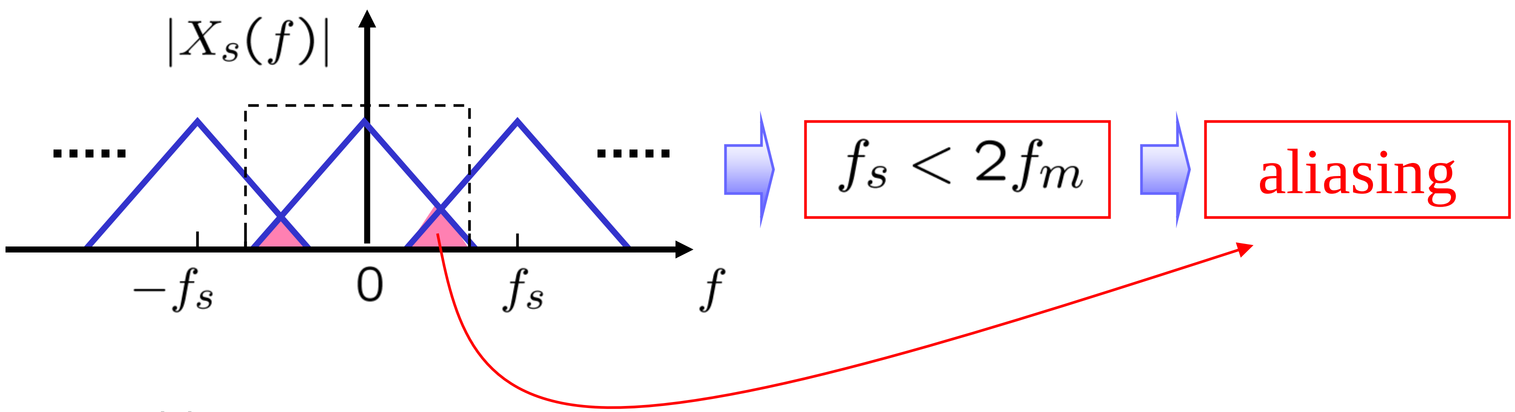


SIGNAL PROCESSING

- Sampling theorem: A bandlimited signal with no spectral components beyond f_m, can be uniquely determined by values sampled at uniform intervals of

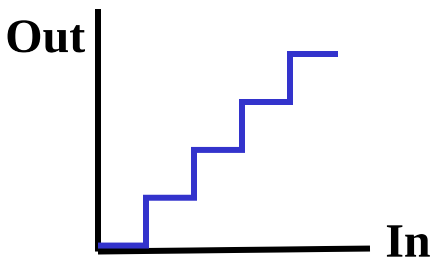
$$f_s = \frac{1}{T_s} = 2f_m$$

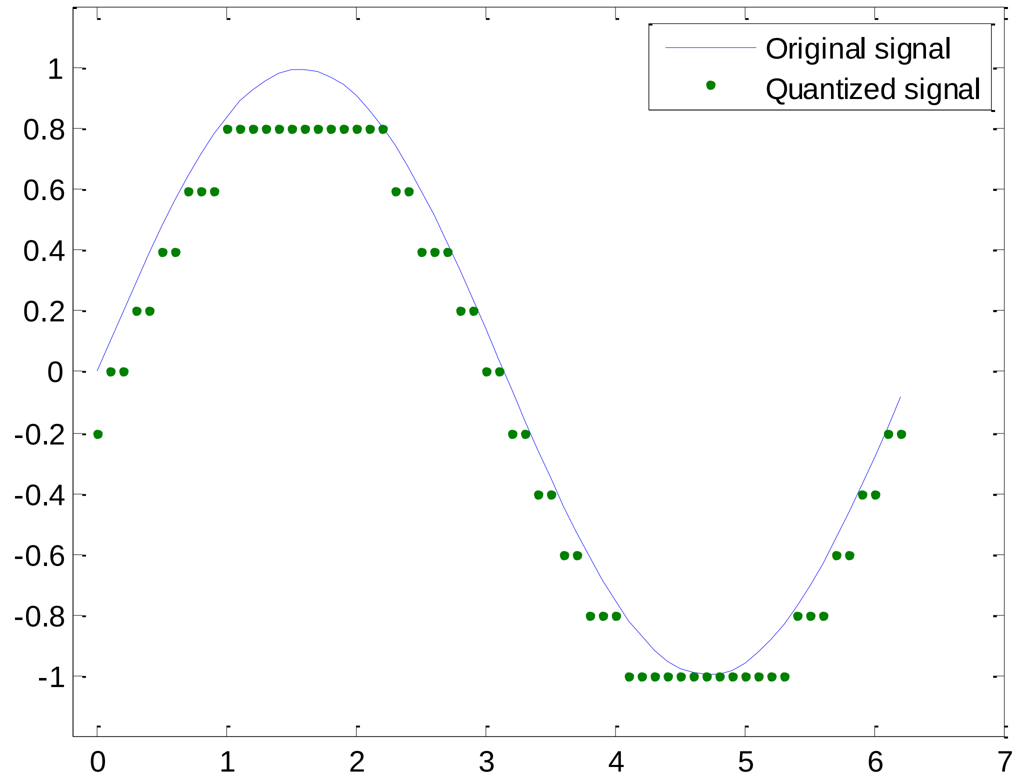
- The sampling rate, $T_s \leq \frac{1}{2f_m}$ is called Nyquist rate.



SIGNAL PROCESSING

- Amplitude quantizing: Mapping samples of a continuous amplitude waveform to a finite set of amplitudes.





See quantizer.m

IMAGE PROCESSING AND ANALYSIS

- Digital Image processing concerns the transformation of an image to a digital format and its processing by digital computers.
  - Both the input and output of digital image processing system are digital images.

- Digital image analysis is related to the description and recognition of the digital image content.
  - Its <u>input</u> is a digital <u>image</u> and its <u>output</u> is a <u>symbolic</u> image <u>description</u>.

IMAGE FORMATION

Photometry overview

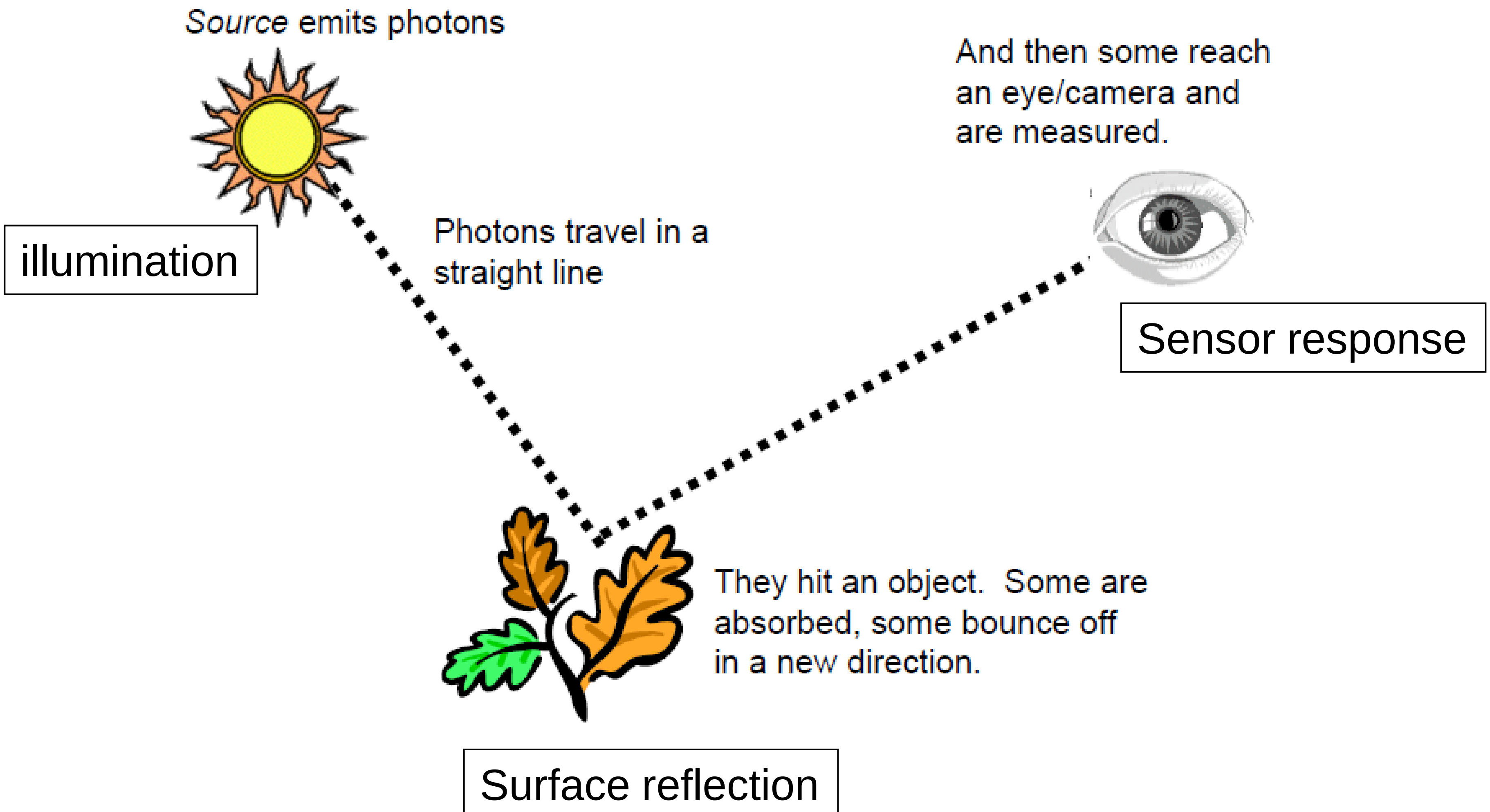


IMAGE FORMATION

Photometry overview

$s(x,y)=il(x,y)\ r(x,y)$

$s(x,y)$: intensity at the point (x,y)

$il(x,y)$: illumination at the point (x,y) (the amount of source illumination incident on the objects)

$r(x,y)$: reflectance at the point (x,y) (the amount of illumination reflected/transmitted by the object)

Where $0< il(x,y,)<inf$ and $0<r(x,y)<1$

s is a two-dimensional function and x and y are spatial coordinates. The amplitude of s is called intensity or gray level at the point (x, y).

IMAGE FORMATION

Photometry overview

- Illumination

  Lumen A unit of light flow or luminous flux

  Lumen per square meter (lm/m^2) — The metric unit of measure for illuminance of a surface

  – On a clear day, the sun may produce in excess of 90,000 lm/m^2 of illumination on the surface of the Earth

  – On a cloudy day, the sun may produce less than 10,000 lm/m^2 of illumination on the surface of the Earth

  – On a clear evening, the moon yields about 0.1 lm/m^2 of illumination

  – The typical illumination level in a commercial office is about 1000 lm/m^2

IMAGE FORMATION

Photometry overview

- Reflectance
  - 0.01 for black velvet
  - 0.65 for stainless steel
  - 0.80 for flat-white wall paint
  - 0.90 for silver-plated metal
  - 0.93 for snow

IMAGE FORMATION

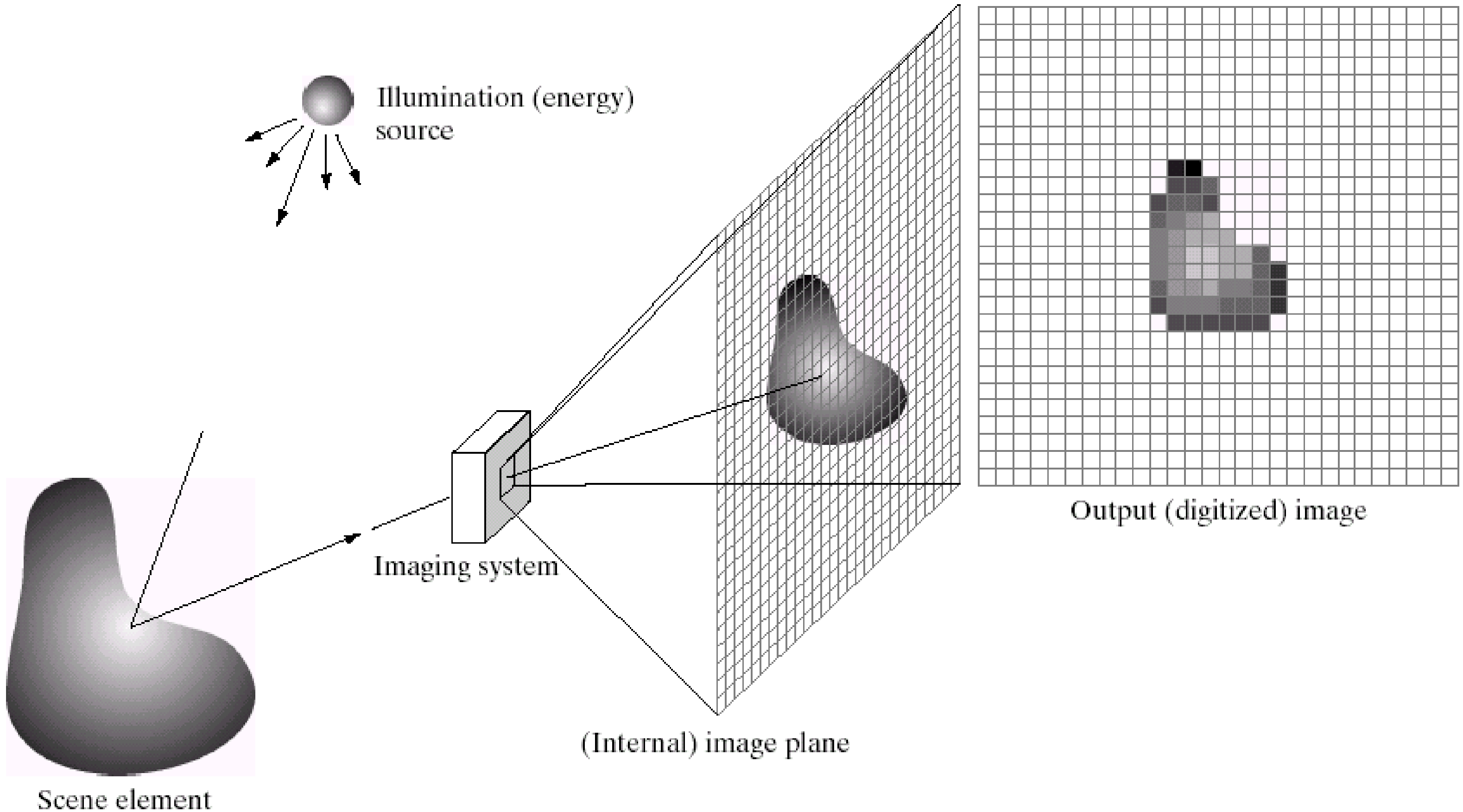
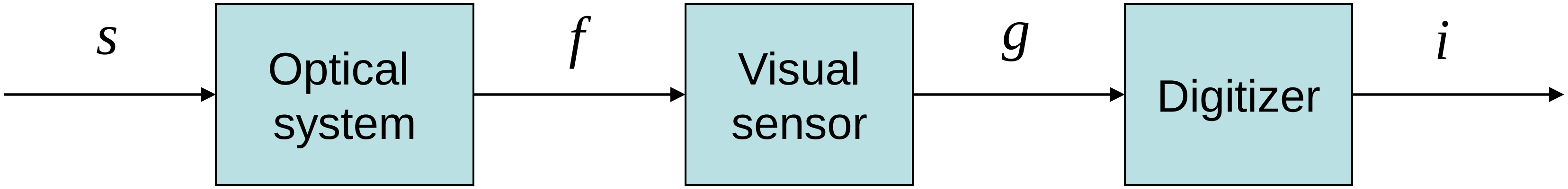


IMAGE FORMATION

- Digital image formation is the first step in any digital image processing application.
- The digital *image formation system* consists basically of the optical system, the sensor and the digitizer.



The effect of the recording process is the addition of a noise contribution. The recorded image *i* is called noisy image.

OPTICAL SYSTEM

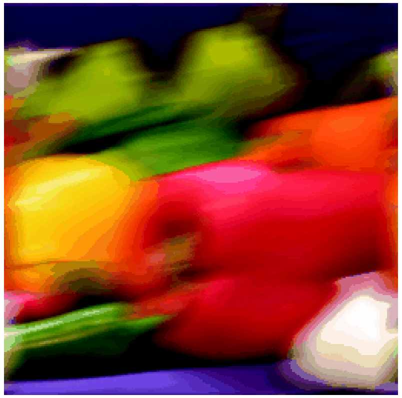
- The optical system can be modeled as a linear shift invariant system having a two-dimensional impulse response $h(x,y)$.
- The input-output relation of the optical system is described by a 2D convolution (both signals *s* and *f* represent optical intensities):

$$f(x,y) = \iint s(\xi,\eta) h(x-\xi, y-\eta) d\xi d\eta$$

OPTICAL SYSTEM

- The two-dimensional impulse response $h(x,y)$ is also called PSF (*point spread function*) and its Fourier transform is called transfer function.
  - Linear motion blur: it is due to the relative motion, during exposure, between the camera and the object being photographed, $H(\mathbf{w}) \sim exp(jTv\mathbf{w})sinc(Tv\mathbf{w})$.
  - Out-of-focus blur: $H(\mathbf{w}) \sim (1/D|\mathbf{w}|)^{3/2} \cos(D|\mathbf{w}|)$.
  - Atmospheric turbulence blur: $H(\mathbf{w}) \sim exp(-s^2 |\mathbf{w}|^2)$.
  - No blur: $h(x, y) = \delta(x, y)$.

Motion blur →



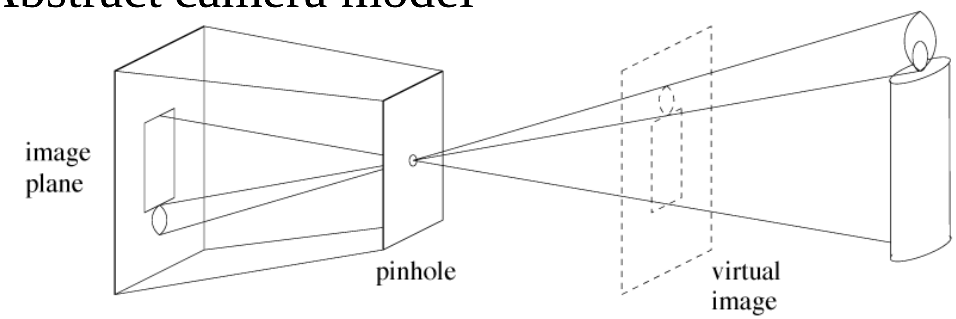
OPTICAL SYSTEM

- Images are two-dimensional patterns of brightness values. They are formed by the projection of 3D objects.

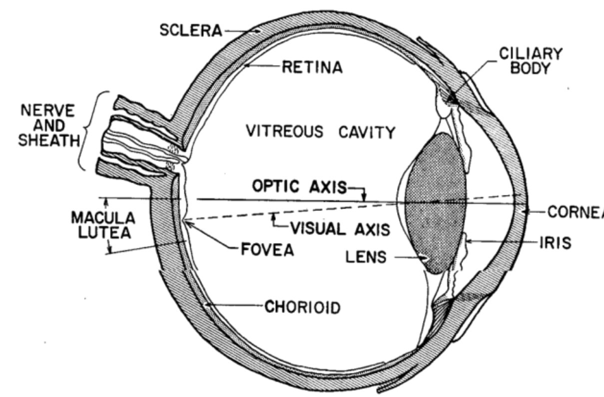
- Basic abstraction is the pinhole camera:
  - Lenses required to ensure image is not too dark.
  - Pinhole cameras work in practice.

PINHOLE CAMERA

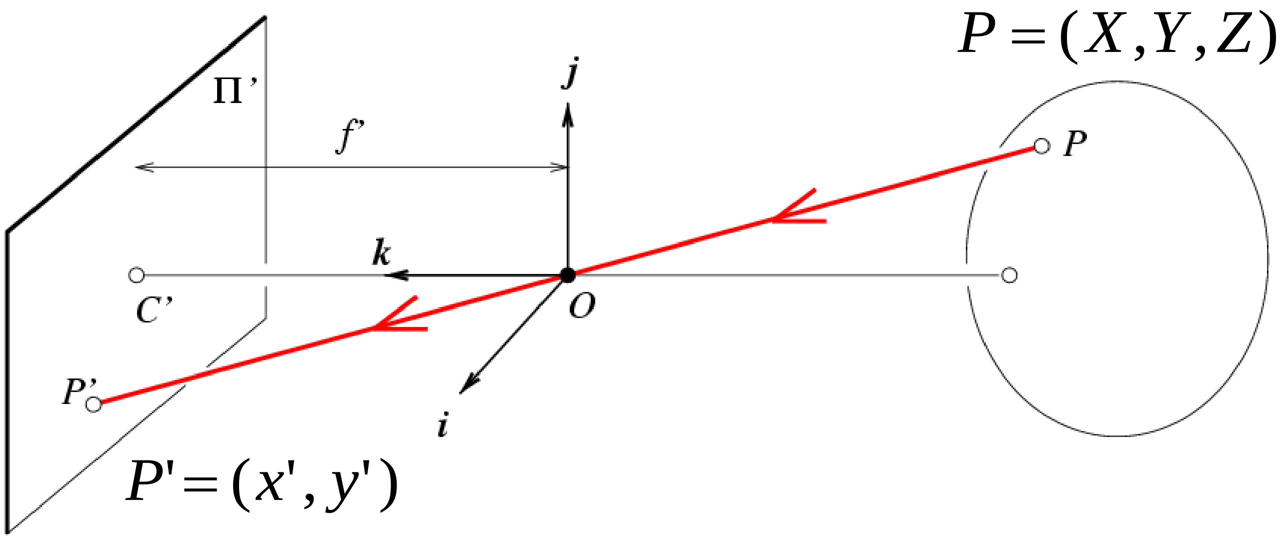
Abstract camera model



Animal eye



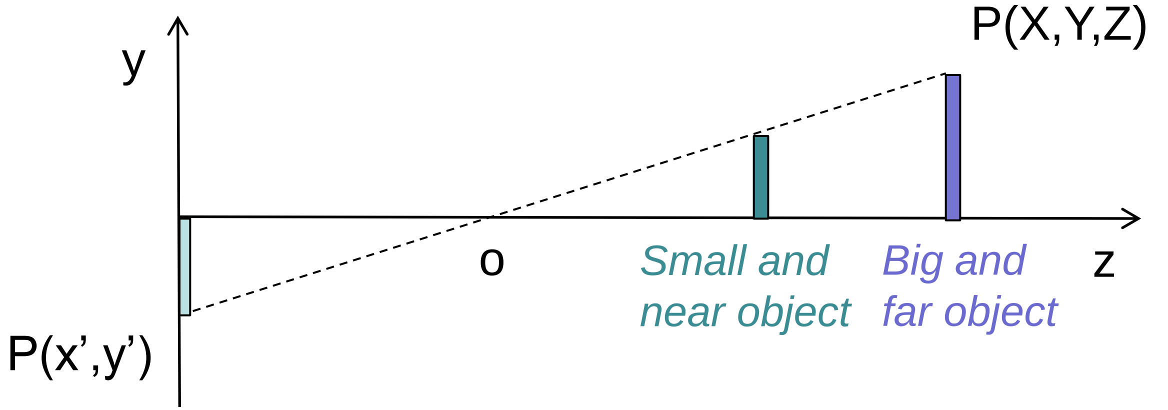
Pinhole Perspective Equation



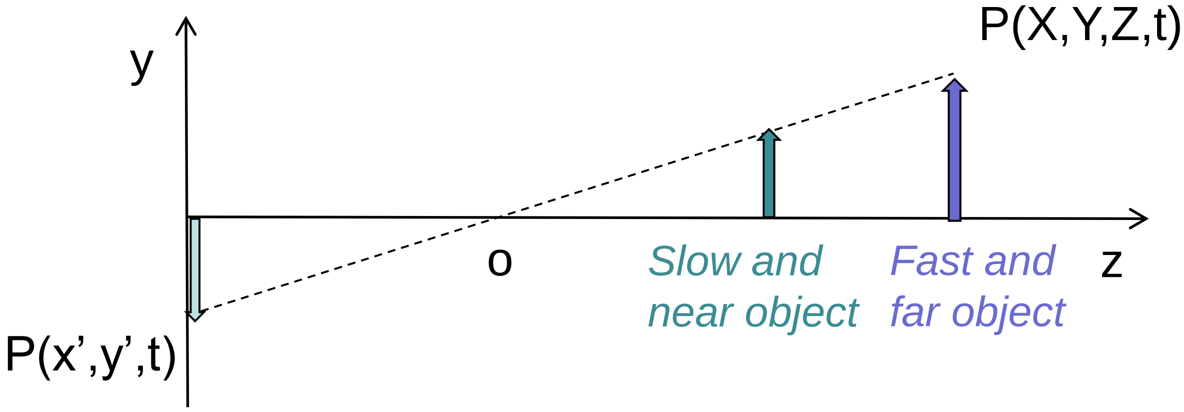
$$
\begin{cases}
x' = f' \dfrac{X}{Z} \\
y' = f' \dfrac{Y}{Z}
\end{cases}
$$

PINHOLE CAMERA

inverse problems, which are prone to be ill-conditioned

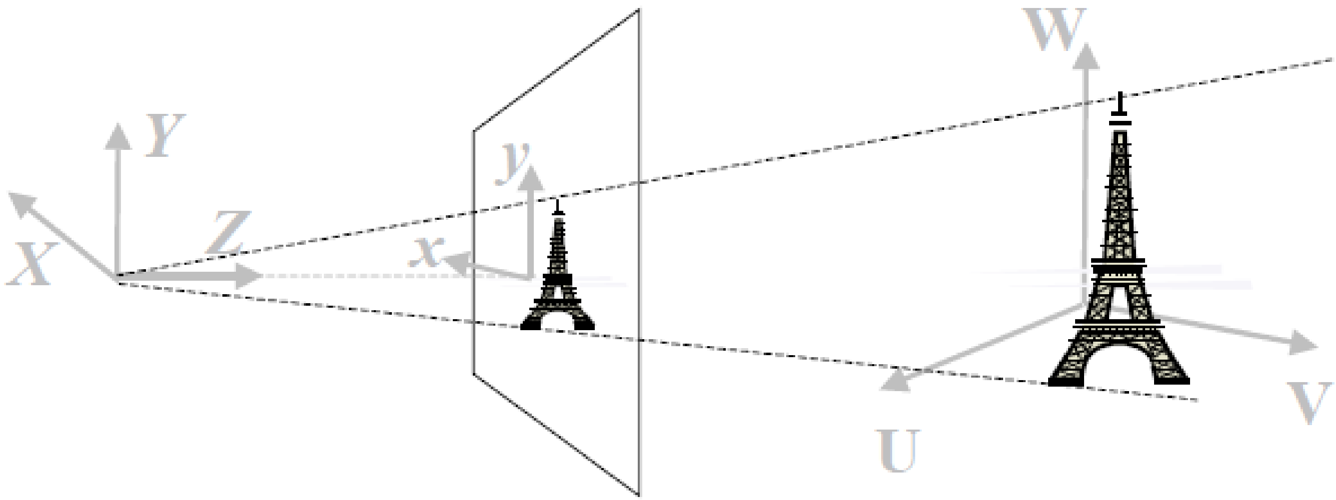


$$
\begin{cases}
x' = f' \dfrac{X}{Z} \\[2ex]
y' = f' \dfrac{Y}{Z}
\end{cases}
$$



PINHOLE CAMERA

Camera projection model



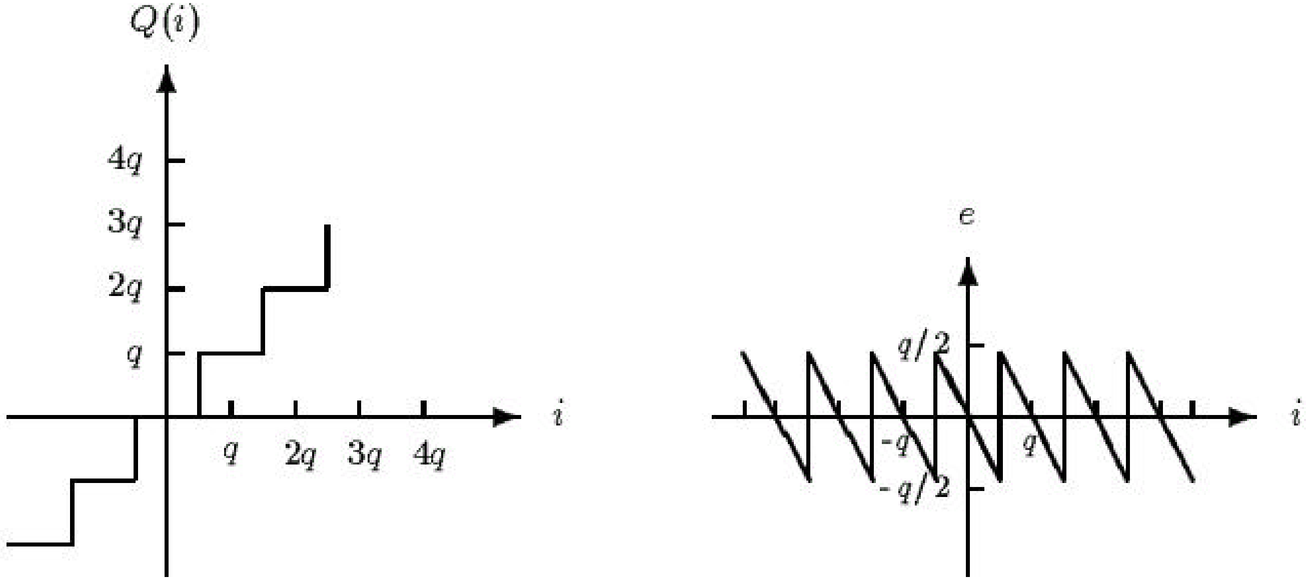
Projection model
Intrinsic (lens) parameters
Extrinsic (pose) parameters
Camera calibration

VISUAL SENSOR AND DIGITIZER

- The visual sensor *transduces* the intensity signal into an electric signal.
- In the case of a analog device, the two-dimensional signal is analog yet: <u>sampling and digitization</u> are performed by an A/D converter (in a frame grabber).
- It transforms the analog image $g(x,y)$ to a digital image $i(n1,n2)$, $n1=1,\ldots N$, $n2=1,\ldots,M$.
- If q is the quantization step, the quantized image is allowed to have illumination at the levels kq, $k=0,1,2,\ldots$

VISUAL SENSOR AND DIGITIZER

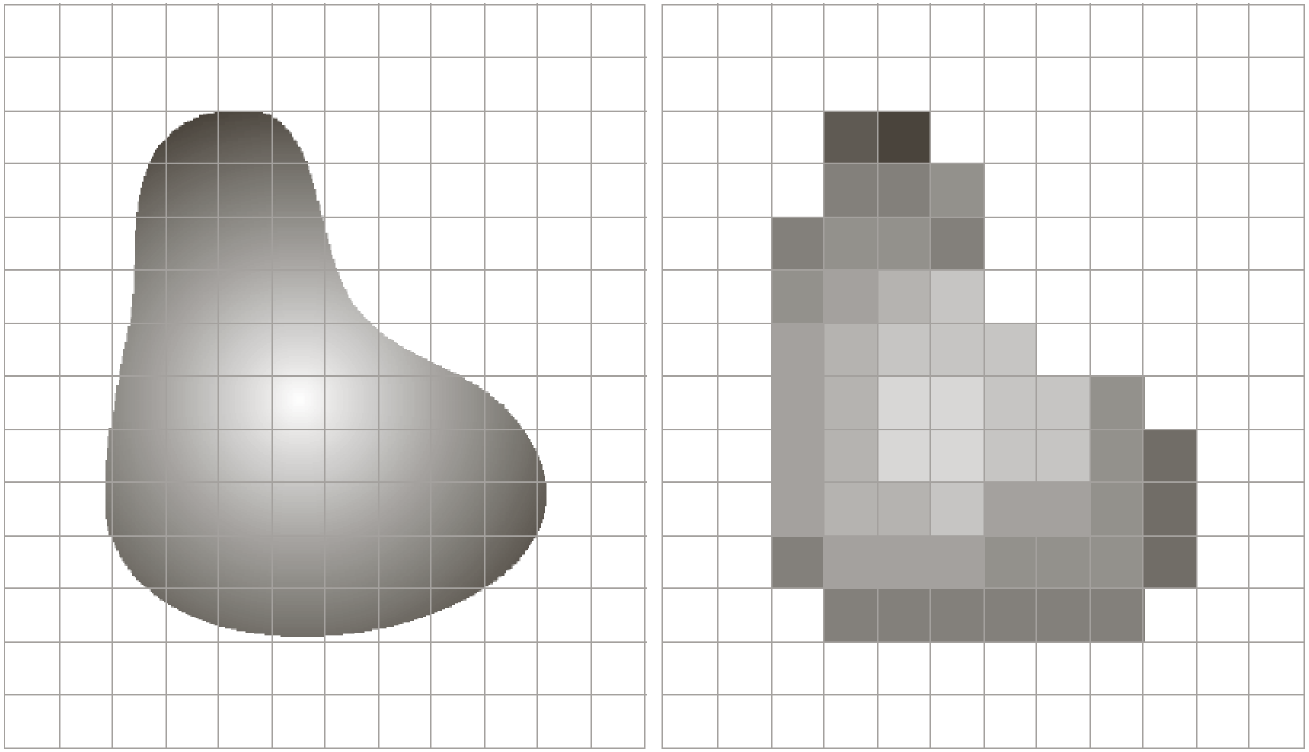
- Image quantization introduces an error.



- Grayscale images are quantized at 256 levels and require 1 byte (8 bits) for the representation of each pixel.
- Binary images have only two quantization levels:0,1. They are represented with 1 bit per pixel.

VISUAL SENSOR AND DIGITIZER

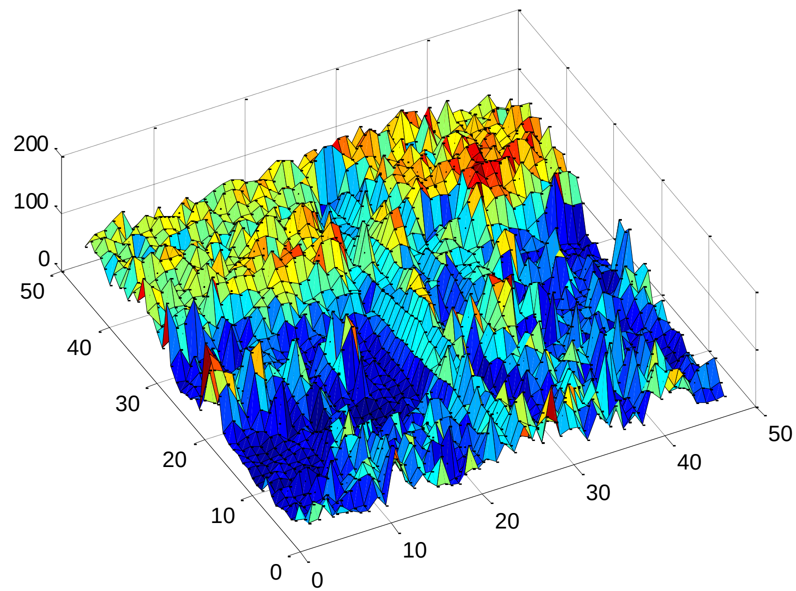
Image sampling and quantization (errors are introduced)



DIGITAL IMAGE REPRESENTATION

42	40	22	84	64	100
40	49	54	67	38	14
18	30	13	76	19	33
19	39	34	90	70	112
81	103	128	57	75	91
79	78	89	55	101	100

The image represented as a two-dimensional matrix *N*x*M* of integers (subsampled).



The image represented as a two-dimensional surface (subsampled).

The image represented as a “picture”.

See image_example.m

DIGITAL IMAGE

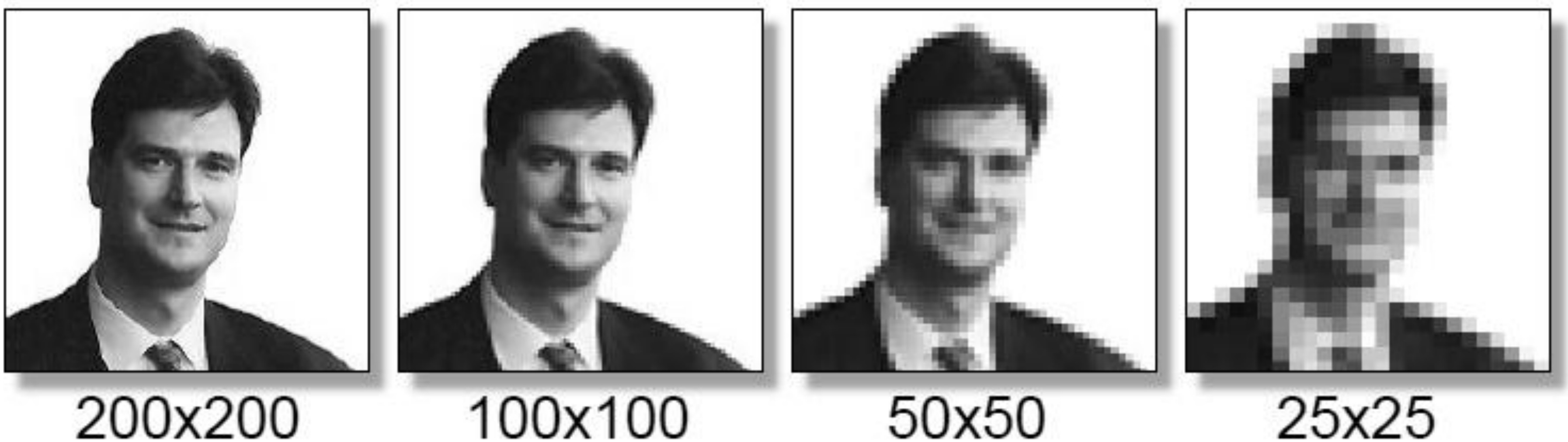
- We can think of an image as a function, *g*, from R^2 to R: $g(x,y)$ gives the intensity at position (x,y). Realistically, $g: [a,b]x[c,d] \rightarrow [0,1]$.

- The digital (discrete) image is obtained by:
  - Sampling the 2D space (domain) on a regular grid.
  - Quantizing the amplitude of each sample (round to nearest integer).
- If our samples are *step* apart, we can write this as:

$$i(\mathrm{h},\mathrm{k}) = \mathrm{Quantize}\{\, g(h\ step,\ k\ step) \,\}$$

DIGITAL IMAGE

- Spatial resolution
  — A measure of the smallest discernible detail in an image
  — stated with *dots (pixels) [dots per inch (dpi)]*

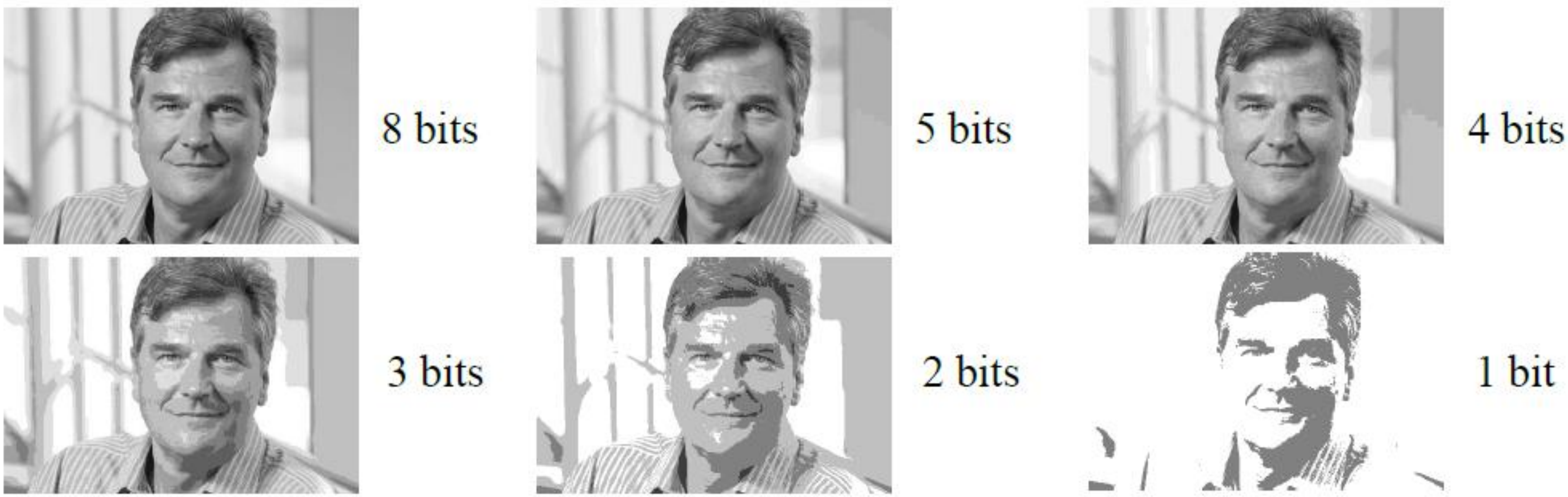


pixels

See image_example.m

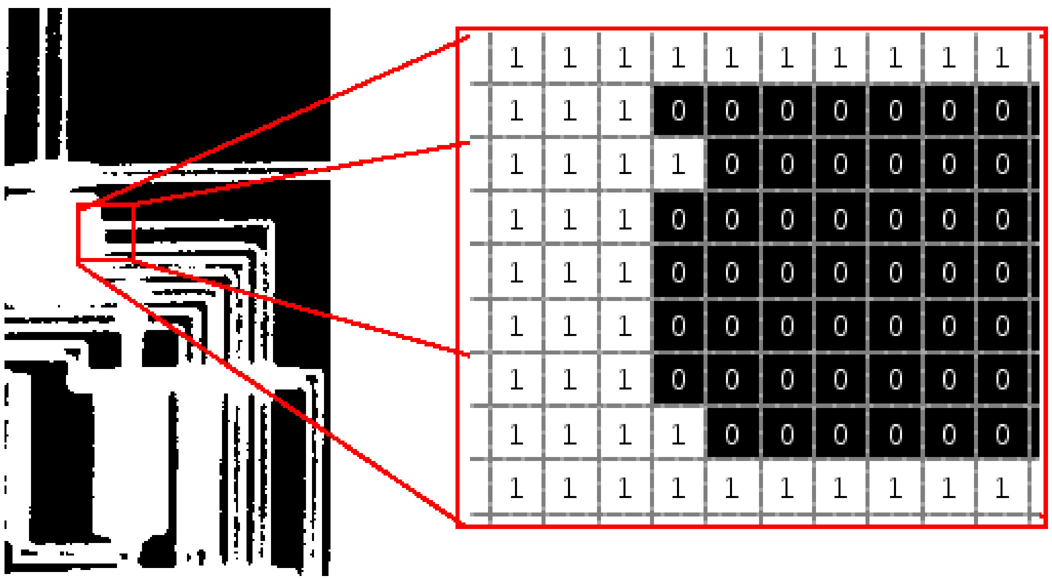
DIGITAL IMAGE

- Intensity resolution
  — The smallest discernible change in intensity level
  — stated with *8 bits, 12 bits, 16 bits, ...*



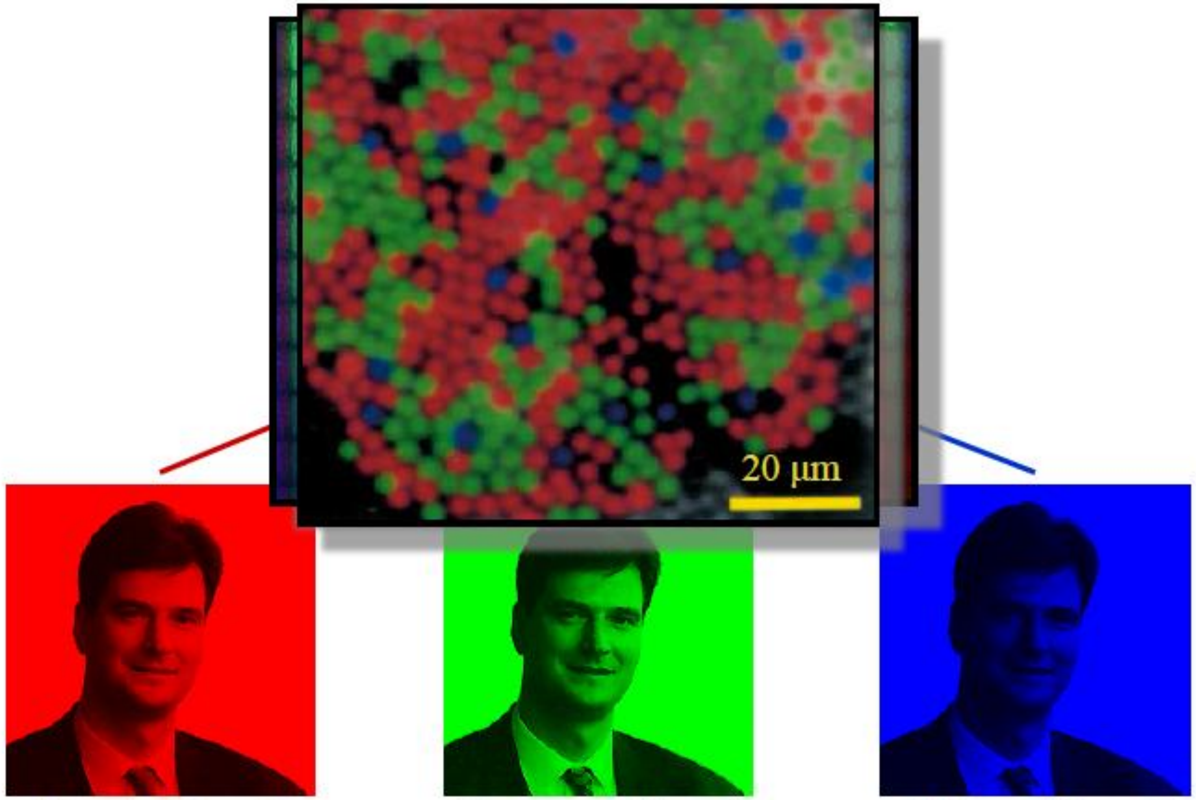
BINARY IMAGES

- In a binary image, each pixel assumes one of only two discrete values: 1 or 0



Color components: trichromacy

Human retina: three different receptors



Red $R[x,y]$ Green $G[x,y]$ Blue $B[x,y]$

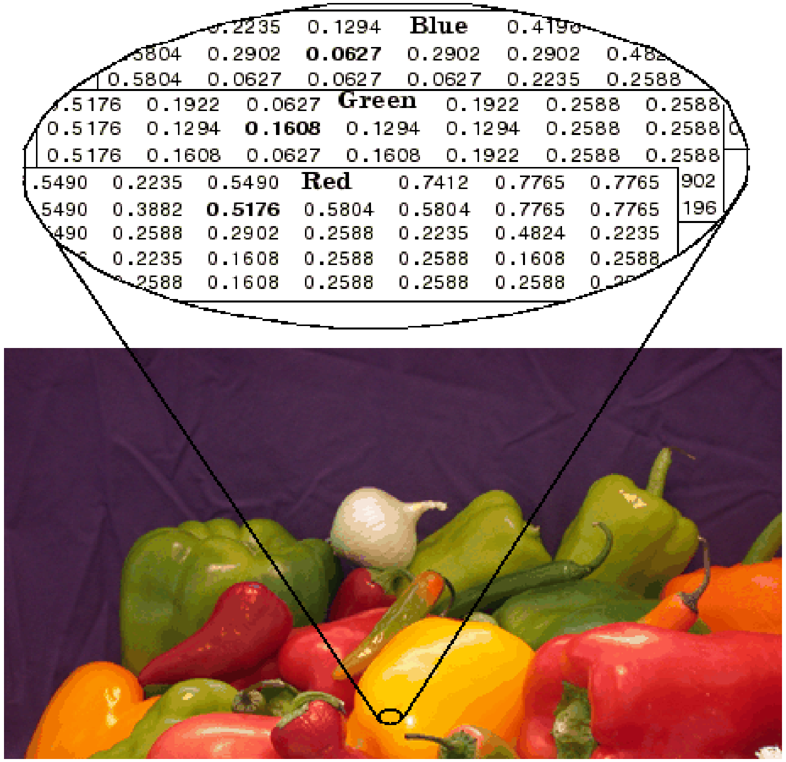
We perceive

Monochrome image

$R[x,y] = G[x,y] = B[x,y]$

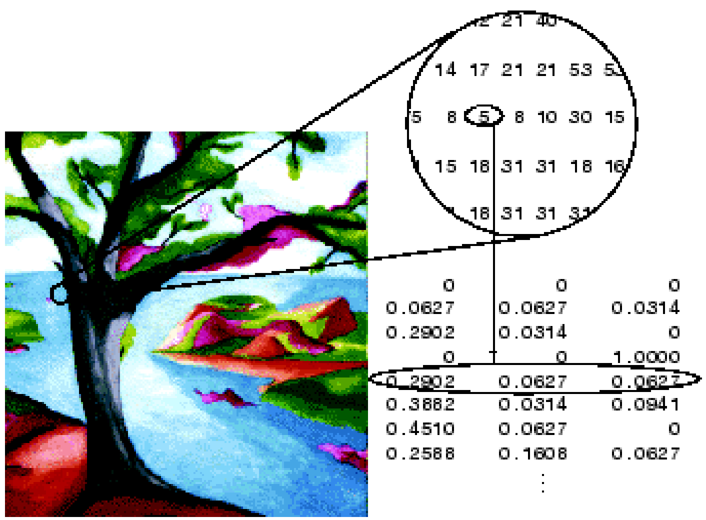
TRUECOLOR IMAGES

- A truecolor image is an image in which each pixel is specified by <u>three values</u> (RGB), one each for the red, blue, and green components of the pixel's color.



INDEXED IMAGES

- An indexed image consists of an array and a colormap matrix. The pixel values in the array are direct indices into a colormap. The colormap matrix is an m-by-3 array of values in the range [0,1]. Each row of map specifies the red, green, and blue components of a single color.



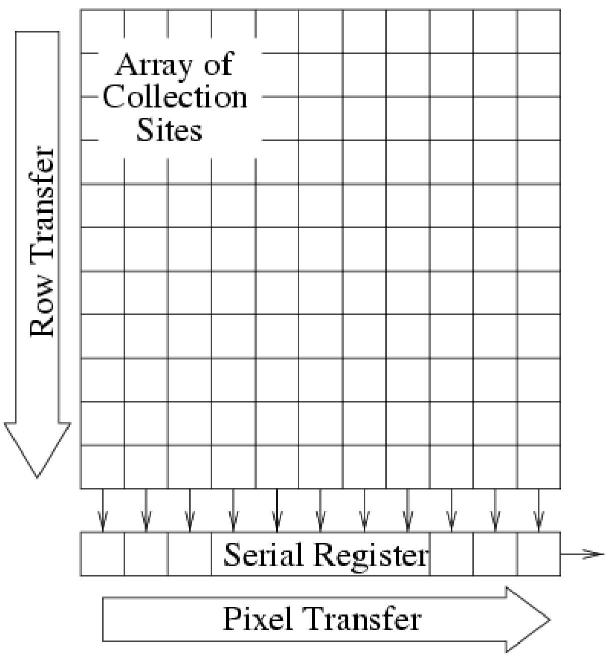
See image_example.m

VISUAL SENSOR AND DIGITIZER

- In the case of a digital acquisition device, the two-dimensional signal is a digital image.
- We consider two types of cameras:
  - CCD (charge-coupled devices).
  - CMOS (complementary metal oxide semiconductor).
- There are separate photo-sensors at regular positions and no scanning.

VISUAL SENSOR AND DIGITIZER

- A possible architecture of a CCD:

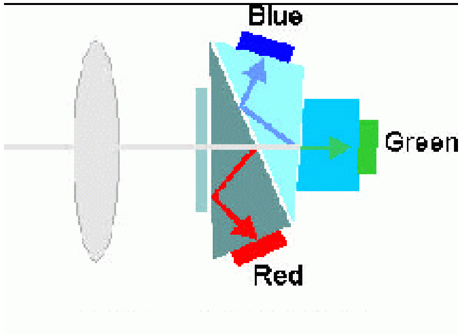


- CMOS:
  - Same sensor elements as CCD.
  - Each photo sensor has its own amplifier.
  - More noise (reduced by subtracting 'black' image)
  - Lower sensitivity (lower fill rate).
  - Allows to put other components on chip "Smart' pixels".

VISUAL SENSOR AND DIGITIZER

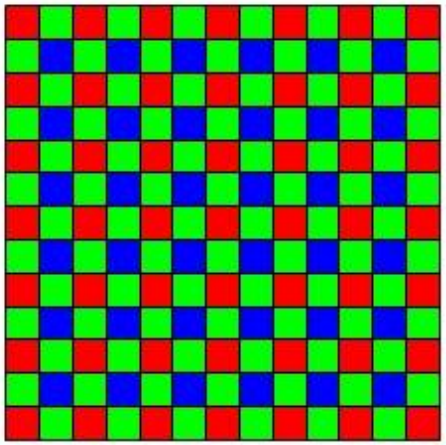
- If we consider a *color image*, its RGB (red,green,blue) values are represented by three matrices.
- Color camera:
  - Prism (with 3 sensors).
  - Filter mosaic (mostly used).

- The prism color camera separates the source light in 3 beams using a prism.

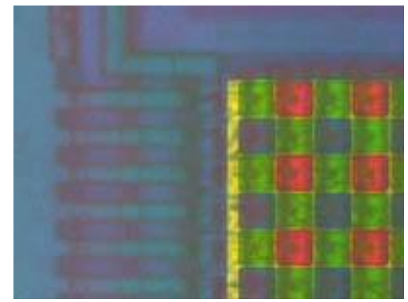


VISUAL SENSOR AND DIGITIZER

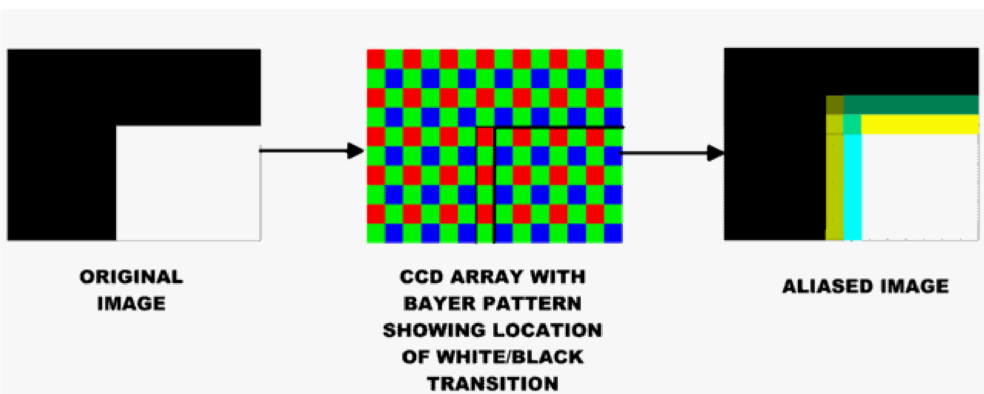
Filter mosaic is directly into sensor.

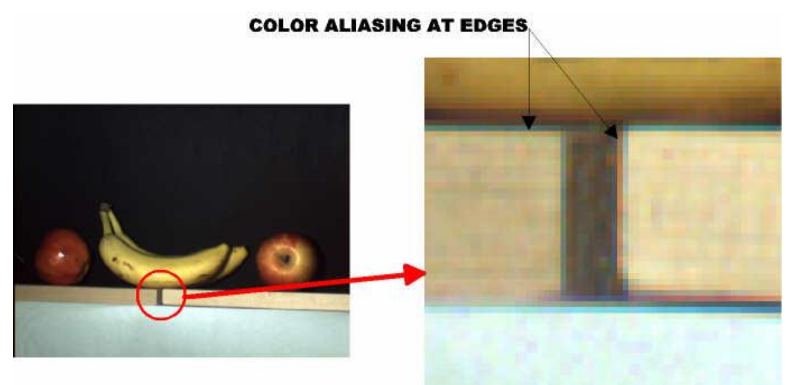


Bayer filter



Demosaicing (to obtain full color & “full” resolution image)

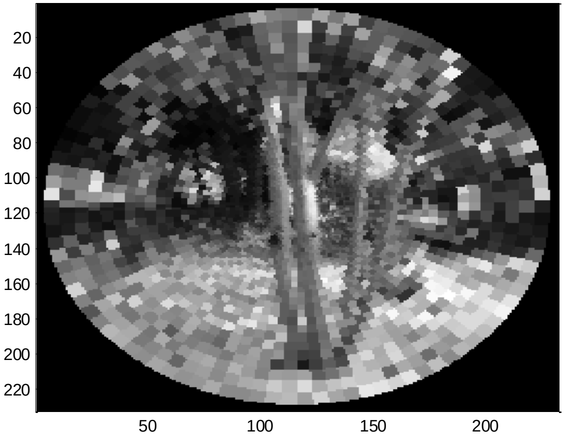




See demosaicing.m

LOG-POLAR VISION

- The log-polar image geometry, first introduced to model the space-variant topology of the human retina receptors in relation to the data compression it achieves, has become popular in the active vision community for the important algorithmic benefits it provides.



F. Solari, M. Chessa, S.P. Sabatini. Design strategies for direct multi-scale and multi-orientation visual processing in the log-polar domain.
Pattern Recognition Letters, 33: pp. 41-51, 2012

BASIC RELATIONSHIPS BETWEEN PIXELS

- **Neighbors** of a pixel p at coordinates (x,y)

➢ **4-neighbors of p**, denoted by $\mathbf{N_4(p)}$:
(x-1, y), (x+1, y), (x,y-1), and (x, y+1).

➢ **4 diagonal neighbors of p**, denoted by $\mathbf{N_D(p)}$:
(x-1, y-1), (x+1, y+1), (x+1,y-1), and (x-1, y+1).

➢ **8 neighbors of p**, denoted $\mathbf{N_8(p)}$
$N_8(p) = N_4(p) \cup N_D(p)$

BASIC RELATIONSHIPS BETWEEN PIXELS

- **Adjacency**

  Let V be the set of intensity values

- **4-adjacency**: Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.

- **8-adjacency**: Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.

BASIC RELATIONSHIPS BETWEEN PIXELS

- **Adjacency**

  Let V be the set of intensity values

- **m-adjacency**: Two pixels p and q with values from V are m-adjacent if

  (i) q is in the set $N_4(p)$, or

  (ii) q is in the set $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

BASIC RELATIONSHIPS BETWEEN PIXELS

- **Path**

➢ A (digital) path (or curve) from pixel p with coordinates (x_0, y_0) to pixel q with coordinates (x_n, y_n) is a sequence of distinct pixels with coordinates

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

Where (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \le i \le n$.

➢ Here *n* is the *length* of the path.

➢ If $(x_0, y_0) = (x_n, y_n)$, the path is ***closed*** path.

➢ We can define 4-, 8-, and m-paths based on the type of adjacency used.

BASIC RELATIONSHIPS BETWEEN PIXELS

V = {1, 2}

$$\begin{matrix} 0_{1,1} & 1_{1,2} & 1_{1,3} \\ 0_{2,1} & 2_{2,2} & 0_{2,3} \\ 0_{3,1} & 0_{3,2} & 1_{3,3} \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{matrix}$$

8-adjacent **m-adjacent**

The 8-path from (1,3) to (3,3):

(i) (1,3), (1,2), (2,2), (3,3)

(ii) (1,3), (2,2), (3,3)

The m-path from (1,3) to (3,3):

(1,3), (1,2), (2,2), (3,3)

BASIC RELATIONSHIPS BETWEEN PIXELS

- **Connected in S**

  Let S represent a subset of pixels in an image. Two pixels p with coordinates (x_0, y_0) and q with coordinates (x_n, y_n) are said to be **connected in S** if there exists a path

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

Where $\forall i, 0 \le i \le n, (x_i, y_i) \in S$

BASIC RELATIONSHIPS BETWEEN PIXELS

Let S represent a subset of pixels in an image

- For every pixel *p* in S, the set of pixels in S that are connected to *p* is called a ***connected component*** of S.

- If S has only one connected component, then S is called ***Connected Set*****.**

- We call R a **region** of the image if R is a connected set

- Two regions, R_i and R_j are said to be ***adjacent*** if their union forms a connected set.
- Regions that are not to be adjacent are said to be ***disjoint.***

BASIC RELATIONSHIPS BETWEEN PIXELS

- **Boundary (or border)**

➢ The ***boundary*** of the region R is the set of pixels in the region that have one or more neighbors that are not in R.

➢ If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns of the image.

- **Foreground and background**

➢ An image contains K disjoint regions, R_k, $k = 1, 2, \ldots, K$. Let R_u denote the union of all the K regions, and let $(R_u)^c$ denote its complement.

All the points in R_u is called **foreground**;

All the points in $(R_u)^c$ is called **background**.

DISTANCE MEASURES

- Given pixels *p, q* and *z* with coordinates (x, y), (s, t), (u, v) respectively, the distance function D has following properties:

1. $D(p, q) \geq 0$ [$D(p, q) = 0$, iff $p = q$]

2. $D(p, q) = D(q, p)$

3. $D(p, z) \leq D(p, q) + D(q, z)$

DISTANCE MEASURES

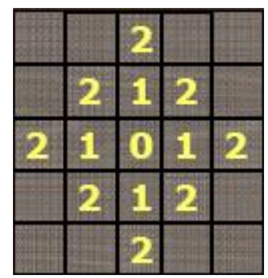
The following are the different Distance measures:

a. Euclidean Distance :

$D_e(p, q) = [(x-s)^2 + (y-t)^2]^{1/2}$

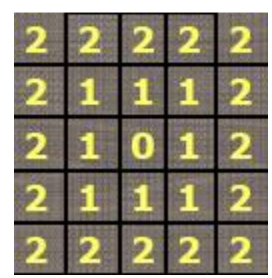
b. City Block Distance:

$D_4(p, q) = |x-s| + |y-t|$



c. Chess Board Distance:

$D_8(p, q) = max(|x-s|, |y-t|)$



MATHEMATICAL OPERATIONS

- **Array vs. Matrix Operation**

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Array product operator

$$A\ .*\ B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Array product

Matrix product operator

$$A * B = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Matrix product

MATHEMATICAL OPERATIONS

- **Linear vs. Nonlinear Operation**

$$H\left[f(x,y)\right]=g(x,y)$$

$$\begin{aligned}
&H\left[a_i f_i(x,y)+a_j f_j(x,y)\right] \\
&=H\left[a_i f_i(x,y)\right]+H\left[a_j f_j(x,y)\right] \\
&=a_i H\left[f_i(x,y)\right]+a_j H\left[f_j(x,y)\right] \\
&=a_i g_i(x,y)+a_j g_j(x,y)
\end{aligned}$$

Additivity

Homogeneity

H is said to be a **linear operator;**

H is said to be a **nonlinear operator** if it does not meet the above qualification.

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- We consider the images $a(i,j)$, $b(i,j)$ and $c(i,j)$ with $i=1,\ldots,N$ and $j=1,\ldots,M$.

- Image addition/subtraction: $c(i\,,j) = a(i\,,j) \pm b(i\,,j)$
- Multiplication of an image by a constant value: $b(i\,,j) = const\ a(i\,,j)$

- Point nonlinear transformations of the form: $b(i\,,j) = h(\,a(i\,,j)\,)$

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- <u>Clipping</u>:

$$b(i,j) = \begin{cases} cmax & \text{if} \quad a(i,j) > cmax \\ a(i,j) & \text{if} \quad cmin <= a(i,j) <= cmax \\ cmin & \text{if} \quad a(i,j) < cmin \end{cases}$$

- <u>Thresholding</u>:

$$b(i,j) = \begin{cases} a1 & \text{if} \quad a(i,j) < T \\ a2 & \text{if} \quad a(i,j) >= T \end{cases}$$

- <u>Contrast stretching</u>:

$$b(i,j) = \underbrace{((a(i,j) - a\_min)/(a\_max - a\_min))}\ (omax - omin) + omin$$

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- Another set of elementary digital image processing operations is related to *geometric transforms*.

- Image translation: *b(i ,j) = a(i + k ,j + h)*

- Image rotation: if the image point *a(x,y)* is rotated by θ degrees, its new coordinates *(x',y')* are given by:

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

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- <u>Scaling</u> is another basic geometrical transformation. It is described by the equation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Image scaling or <u>zooming</u> is given by:

$$b(x',y') = b(\alpha x, \beta y) = a(x,y)$$

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- Geometric transformation (rubber-sheet transformation)

— A spatial transformation of coordinates

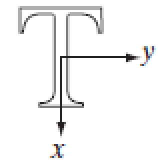
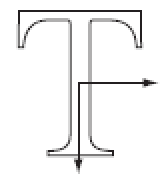
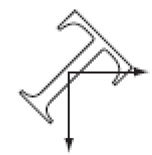
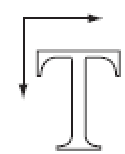
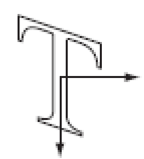
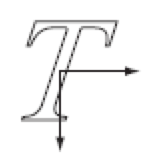
$$(x, y) = T\{(v, w)\}$$

— intensity interpolation that assigns intensity values to the spatially transformed pixels.

- Affine transform

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- The rotated version *b(x',y')* of the image *a(x,y)* is given by:

$$b(x',y') = b(x\ cos - y\ sin\ ,x\ sin + y\ cos)= a(x,y)$$

- It must be notated that the digital coordinates are integers.
- It is worth noting that the actual algorithm maps pixels of the destination image to the source image. The opposite is possible and it is easier to implement, but it creates pattern spikes in the rotated image.
- This operation can requires image interpolation.

ELEMENTARY DIGITAL IMAGE PROCESSING OPERATIONS

- Forward Mapping

$$(x, y) = T\{(v, w)\}$$

It's possible that two or more pixels can be transformed to the same location in the output image.

- Inverse Mapping

$$(v, w) = T^{-1}\{(x, y)\}$$

The nearest input pixels to determine the intensity of the output pixel value.

Inverse mappings are more efficient to implement than forward mappings.

See zooming.m

IMAGE INTERPOLATION

- **Interpolation** — Process of using known data to estimate unknown values

  e.g., zooming, shrinking, rotating, and geometric correction

- **Interpolation** (sometimes called ***resampling***) — an imaging method to increase (or decrease) the number of pixels in a digital image.

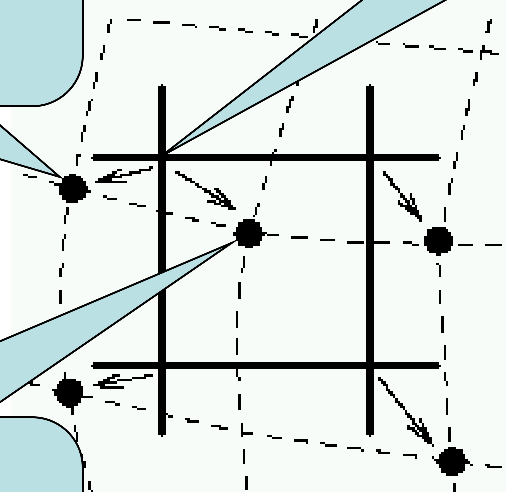
  Some digital cameras use interpolation to produce a larger image than the sensor captured or to create digital zoom

IMAGE INTERPOLATION

Nearest Neighbor Interpolation

$f_1(x_2,y_2) =$
$f(\mathrm{round}(x_2), \mathrm{round}(y_2))$
$=f(x_1,y_1)$

$f(x_1,y_1)$



$f_1(x_3,y_3) =$
$f(\mathrm{round}(x_3), \mathrm{round}(y_3))$
$=f(x_1,y_1)$

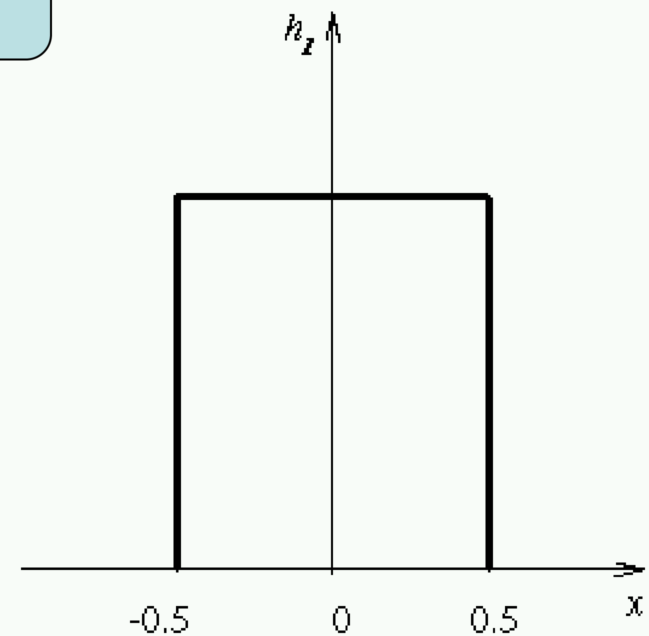
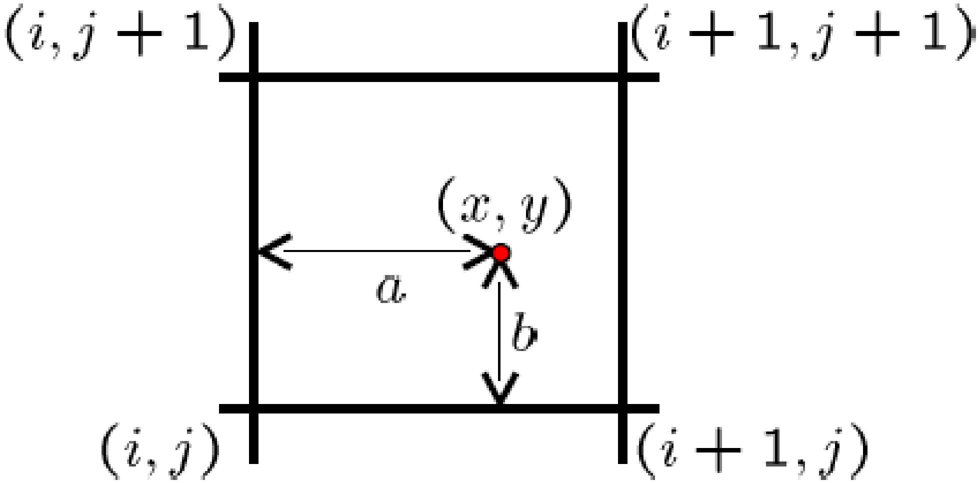


IMAGE INTERPOLATION

Bilinear Interpolation



a= *x* – *floor(x)*
b= *y* – *floor(y)*

$$\begin{aligned} f(x,y) = \;& (1-a)(1-b) & f[i,j] \\ & +a(1-b) & f[i+1,j] \\ & +ab & f[i+1,j+1] \\ & +(1-a)b & f[i,j+1] \end{aligned}$$

HOMEWORK

- Reading Assignments:
  - Gonzalez and Woods book: chapters 1 and 2.
  - MATLAB “getting started” documentation at http://www.mathworks.com.
- Programming assignments:
  - To implement some examples in MATLAB.