Digital Image Fundamentals and Programming

CHUI Chee Kong, PhD
Control & Mechatronics Group
Mechanical Engineering, NUS

- A/Prof Chui Chee Kong

  Room: E1-05-22

  Phone: 6516 1336

  Email: mpecck@nus.edu.sg

What you need to know

- Pixel, brightness/gray levels, resolution
- Coordinates of reference in images
- Sampling and quantization
- Simple image processing operations and implementation on MATLAB
- Distance measurement, connectivity, histogram
- Visual perception

What you need to know (continue)

- Image arithmetic and logic operations
- Point processing of monochrome image
- Gray level transformation
- Histogram equalization

Contents

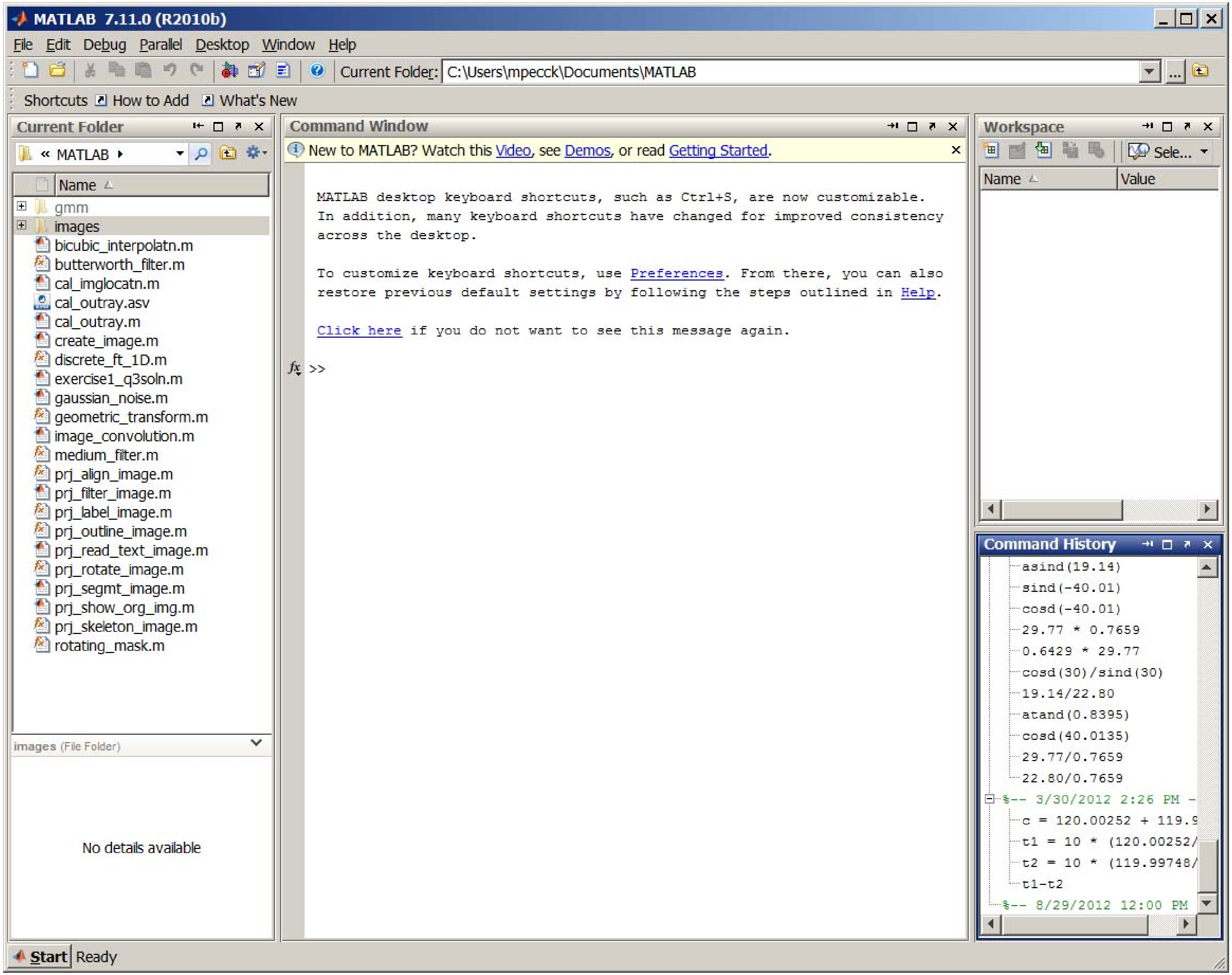
1. Introduction to MATLAB
2. Image Representation
3. Image Sampling and Quantization
4. Digital Image Properties
5. Basic Operations
6. Image Enhancement by Point Processing

1. Introduction to MATLAB

- MATLAB (short for Matrix Laboratory)
  - An interpreter:
    - Program remains in computer's memory after it is launched.
    - Command window for interpreting commands
    - If commands are considered correct, MATLAB executes it.

Implementation language for digital image processing

- MATLAB
  - Allows quick prototyping
  - Many implementations of image analysis algorithms available in MATLAB.
  - Can be easily rewritten in C, C++ or Java
  - Widely popular Image Processing Toolbox which covers basic image processing capabilities well.
  - Public domain/open source clones of MATLAB are widely available, for examples, Octava and SciLab.


MATLAB 7.11.0 (R2010b)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\mpecck\Documents\MATLAB
Shortcuts How to Add What's New
Current Folder
« MATLAB ›
Name
gmm
images
bicubic_interpolatn.m
butterworth_filter.m
cal_imglocatn.m
cal_outray.asv
cal_outray.m
create_image.m
discrete_ft_1D.m
exercise1_q3soln.m
gaussian_noise.m
geometric_transform.m
image_convolution.m
medium_filter.m
prj_align_image.m
prj_filter_image.m
prj_label_image.m
prj_outline_image.m
prj_read_text_image.m
prj_rotate_image.m
prj_segmt_image.m
prj_show_org_img.m
prj_skeleton_image.m
rotating_mask.m
images (File Folder)
No details available
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
MATLAB desktop keyboard shortcuts, such as Ctrl+S, are now customizable.
In addition, many keyboard shortcuts have changed for improved consistency
across the desktop.
To customize keyboard shortcuts, use Preferences. From there, you can also
restore previous default settings by following the steps outlined in Help.
Click here if you do not want to see this message again.
fx >>
Workspace
Sele...
Name
Value
Command History
asind(19.14)
sind(-40.01)
cosd(-40.01)
29.77 * 0.7659
0.6429 * 29.77
cosd(30)/sind(30)
19.14/22.80
atand(0.8395)
cosd(40.0135)
29.77/0.7659
22.80/0.7659
%-- 3/30/2012 2:26 PM -
c = 120.00252 + 119.9
t1 = 10 * (120.00252/
t2 = 10 * (119.99748/
t1-t2
%-- 8/29/2012 12:00 PM
Start Ready

Vectors and matrices:

```
>> % assignment of a real matrix
>> a = [1 2 3; 4 5 6]

a =

     1     2     3
     4     5     6

>> % implicit enumeration
>> a = (0:10) % or a = (0:1:10)

a =

     0     1     2     3     4     5     6     7     8     9    10

>> % incremental implicit enumeration
>> a = (0:2:9)

a =

     0     2     4     6     8
```

An *m* x *n* matrix is a rectangular array of entries or elements enclosed typically by square bracket, where *m* is the number of rows and *n* the number of columns.

A vector of dimension *n* is an ordered collection of *n* entries or elements.

An element can be a number or symbol representing number.

Vectors and matrices (Continue):

```
>> % extension of matrix
>> a = [1 2 3; 4 5 6];
>> a = [a a]

a =

     1     2     3     1     2     3
     4     5     6     4     5     6

>> a = [a; a]

a =

     1     2     3     1     2     3
     4     5     6     4     5     6
     1     2     3     1     2     3
     4     5     6     4     5     6
```

Arrays:

```
>> clear all
>> A = [1:3; 4:6]

A =

     1     2     3
     4     5     6

>> A

A =

     1     2     3
     4     5     6

>> A(:, :, 2)=zeros(2, 3), % or A(:, :, 2) =0

A(:,:,1) =

     1     2     3
     4     5     6


A(:,:,2) =

     0     0     0
     0     0     0
```

Array is a data structure; matrix is a mathematical concept.

All MATLAB variables are multidimensional arrays. A matrix is a two-dimensional (2D) array often used for linear algebra. A vector is a one-dimensional (1D) array.

Simple matrix and array operations:

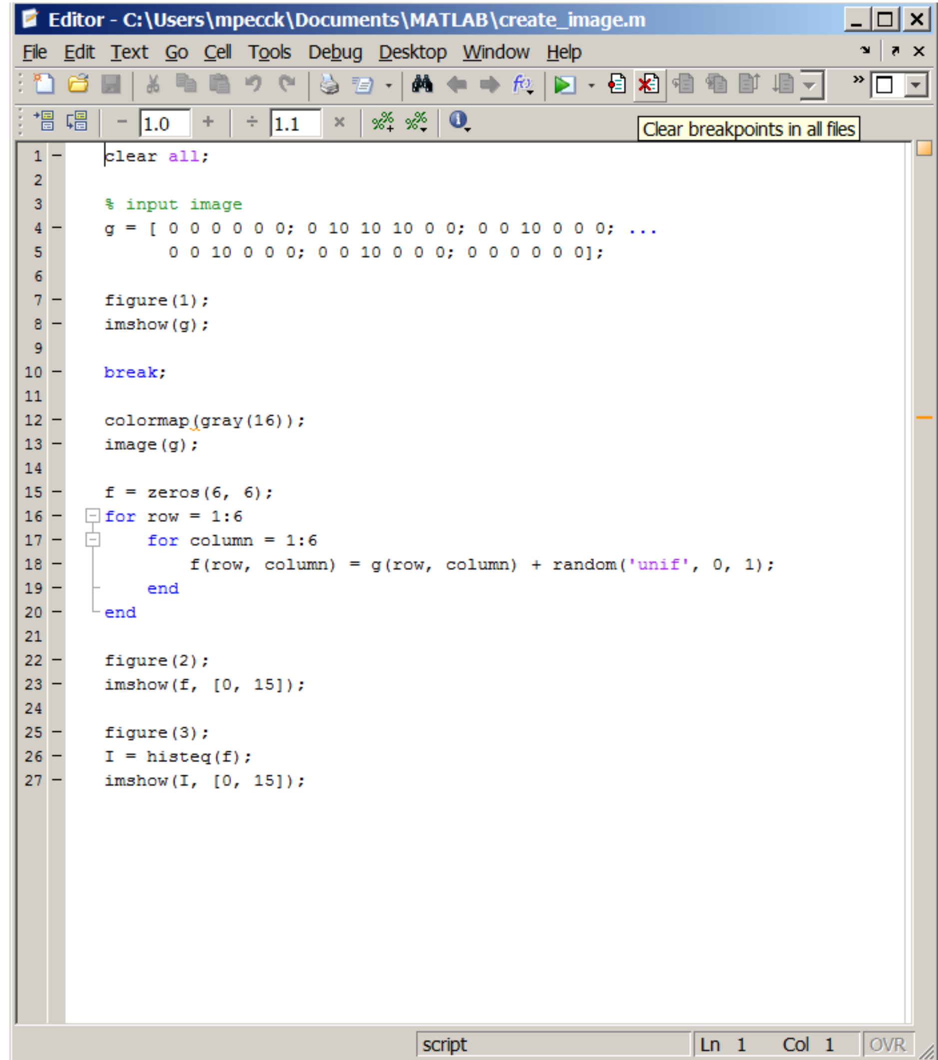
```
>> a = [ 1 2; 3 4] * [5; 6]

a =

    17
    39

>> size(a)

ans =

     2     1

>> A = [ 1 2 ; 2 1]; b = [1; 1]

b =

     1
     1
```

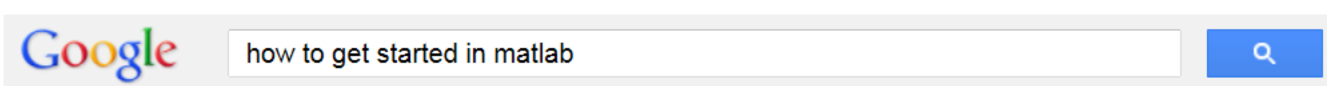
```
>> x = inv(A) * b

x =

    0.3333
    0.3333

>> a^2
??? Error using ==> mpower
Inputs must be a scalar and a square matrix.

>> A^2

ans =

     5     4
     4     5
```

```
>> A'

ans =

     1     2
     2     1

>> A' .* 2

ans =

     2     4
     4     2
```

“.*”: element-wise multiplication, is an example of array operations

```
>> help edit
 EDIT Edit or create a file
    EDIT FUN opens the file FUN.M in a text editor.  FUN must be the
    name of a file with a .m extension or a MATLABPATH relative
    partial pathname (see PARTIALPATH).

    EDIT FILE.EXT opens the specified file.  MAT and MDL files will
    only be opened if the extension is specified.  P and MEX files
    are binary and cannot be directly edited.

    EDIT X Y Z ... will attempt to open all specified files in an
    editor.  Each argument is treated independently.

    EDIT, by itself, opens up a new editor window.

    By default, the MATLAB built-in editor is used.  The user may
    specify a different editor by modifying the Editor/Debugger
    Preferences.

    If the specified file does not exist and the user is using the
    MATLAB built-in editor, an empty file may be opened depending on
    the Editor/Debugger Preferences.  If the user has specified a
    different editor, the name of the non-existent file will always
    be passed to the other editor.

    Overloaded methods:
       axischild/edit
       vrworld/edit

    Reference page in Help browser
       doc edit

>> edit create_image
>>
```





http://www.mathworks.com/videos/matlab/getting-started-with-matlab.html

2. Image Representation

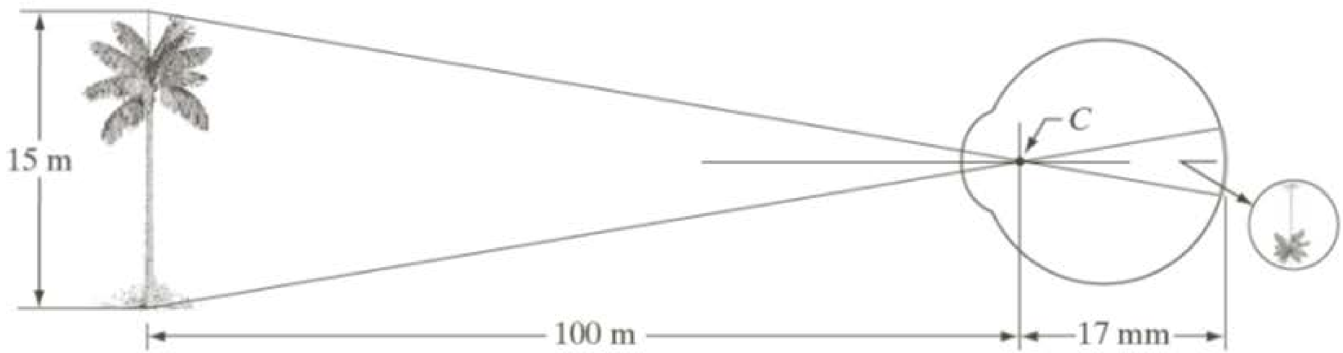


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

Image on human retina or captured by a TV camera or senor.

- A projection of 3D scene into 2D.

Continuous Image Function

- The image is modeled by a continuous function of two variables $f(x, y)$ or three variables $f(x, y, t)$.
  - (x, y) gives the coordinate of a plane. t is the time domain.

- The image function value f correspond to brightness at an image point.
  - An intensity image is a 2D image bearing information about brightness points.

Image can be considered as a 2D light intensity function, $f(x,y)$, where x and y are spatial coordinates, and f at (x, y) is related to the brightness or color of the image at that point.

Continuous Image Function (Continue)

- The brightness or intensity of the image is determined by
  - Illumination component: the amount of light incident on the viewing scene.
  - Reflection component: the amount of light reflected by the object in the scene.
    - Example values of typical surfaces: black silk (0.01), stainless steel (0.65), silver plate (0.90).
- The function value *f* can also express other physical quantities such as temperature etc.

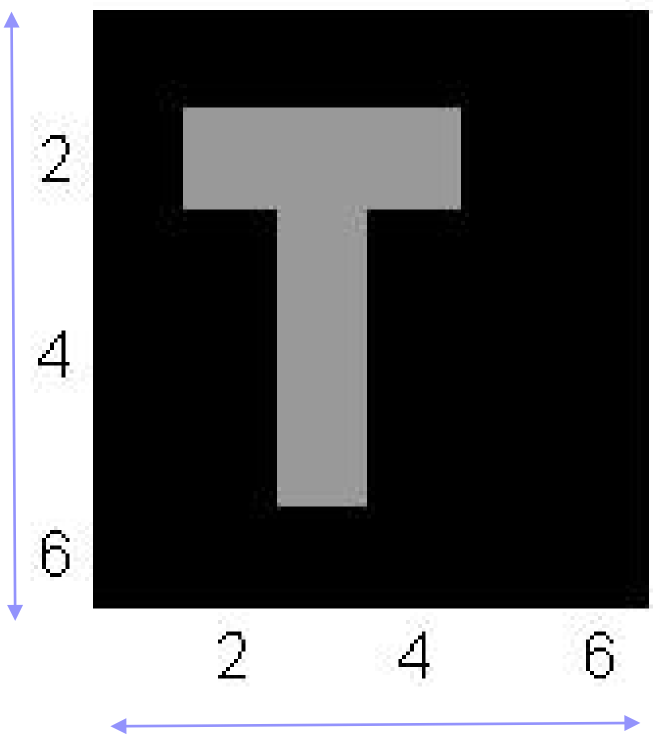
Image Digitalization

- An image to be processed by computer must be represented using appropriate discrete data structure.
  - Matrix or 2D array for 2D image
- Image digitalization is the process of converting an image into a numerical representation through sampling and quantization.
- Sampling turns the continuous function $f(x,y)$ into a matrix of N rows and M columns.
- Quantization assigns an integer value to each continuous sample.

Digital Image

- A digital image (== raster or bitmapped image) is the representation of a continuous image $f(x,y)$ by a matrix of discrete sample.
- The function value at each discrete sample is quantized to be represented by a finite number of bits.
- Each element of the matrix of samples is called pixel (for PICture ELement).

6
pixel



6 pixels

A digital image is
represented by numbers.

0	0	0	0	0	0
0	10	10	10	0	0
0	0	10	0	0	0
0	0	10	0	0	0
0	0	10	0	0	0
0	0	0	0	0	0

A pixel represents
brightness at a point.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A digital image
can be
represented as
a matrix.

Pixel

- Pixel is a point sample of the original image.
- It is the smallest piece of information in an image - the elemental part of an image.
- It has a position relative to other pixels in the image.
- It has a color capability measured in bits.
- Pixel is often represented using dot or square.

3. Image Sampling and Quantization

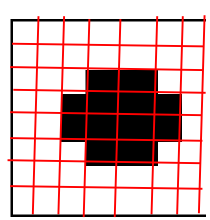
- Digital image processing is the use of computer algorithms to perform image processing on digital images.
- It is a sub field of digital signal processing.
- Image sampling and image quantization are examples of digital image processing.

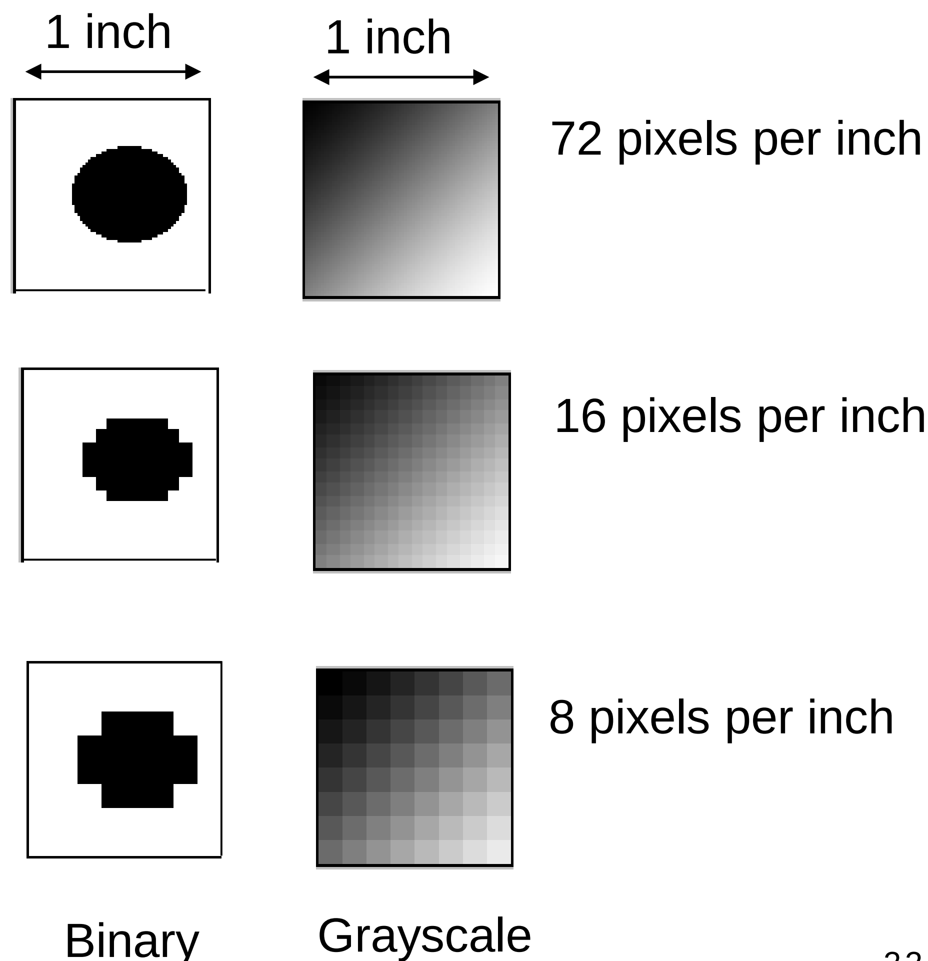
3.1 Sampling

Sampling turns the continuous function f(x,y) into a matrix of N rows and M columns.

- Image sampling – discretion in space
  - SAMPLE does not equal to the original shape

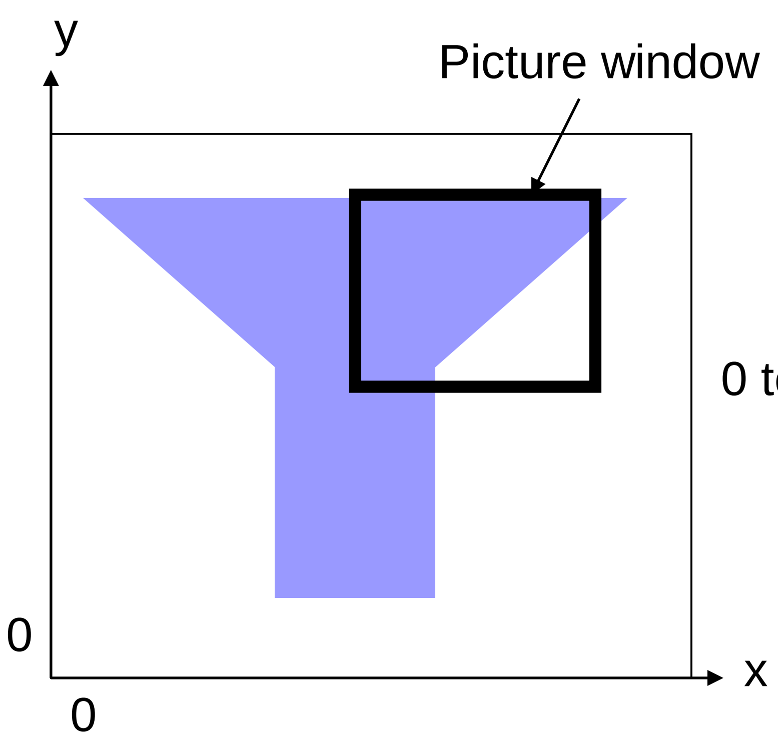
Fewer pixels mean lower spatial resolution. For example, we may reduce the size of a 640 x 480 image to 160 by 120. We will observe blurring when the 160 x 120 image is interpolated back to 640 x 480.





Picture Space and Image Space

IMAGE: *M* x *N* matrix of pixels *p*(*i*, *j*)



Picture space

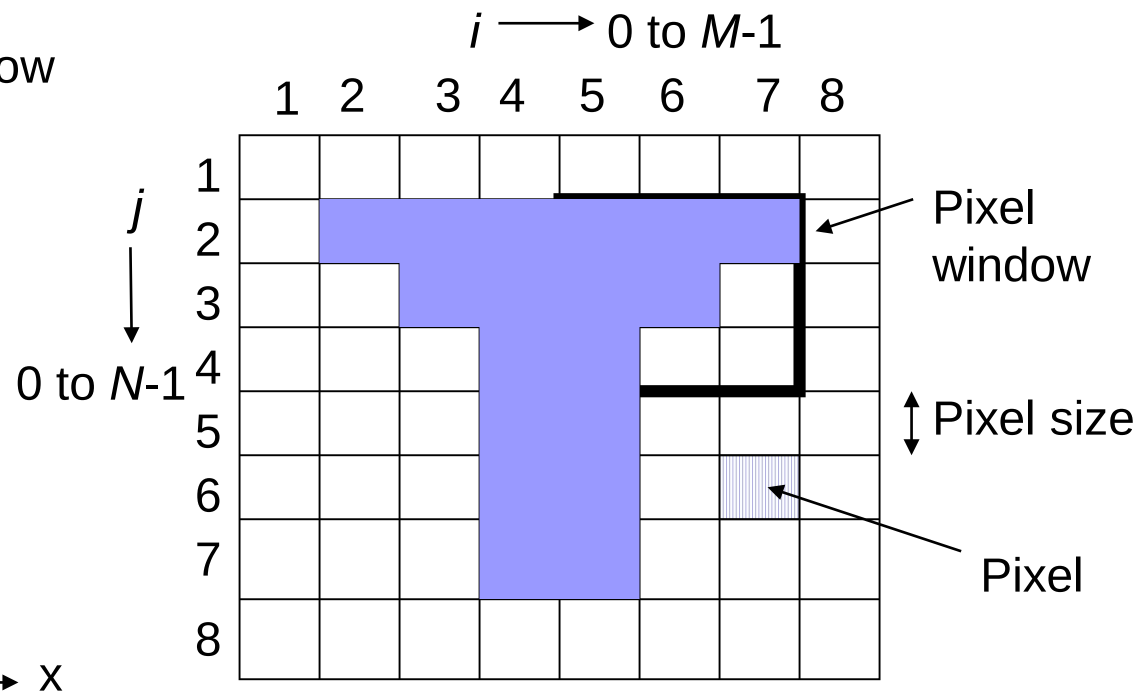
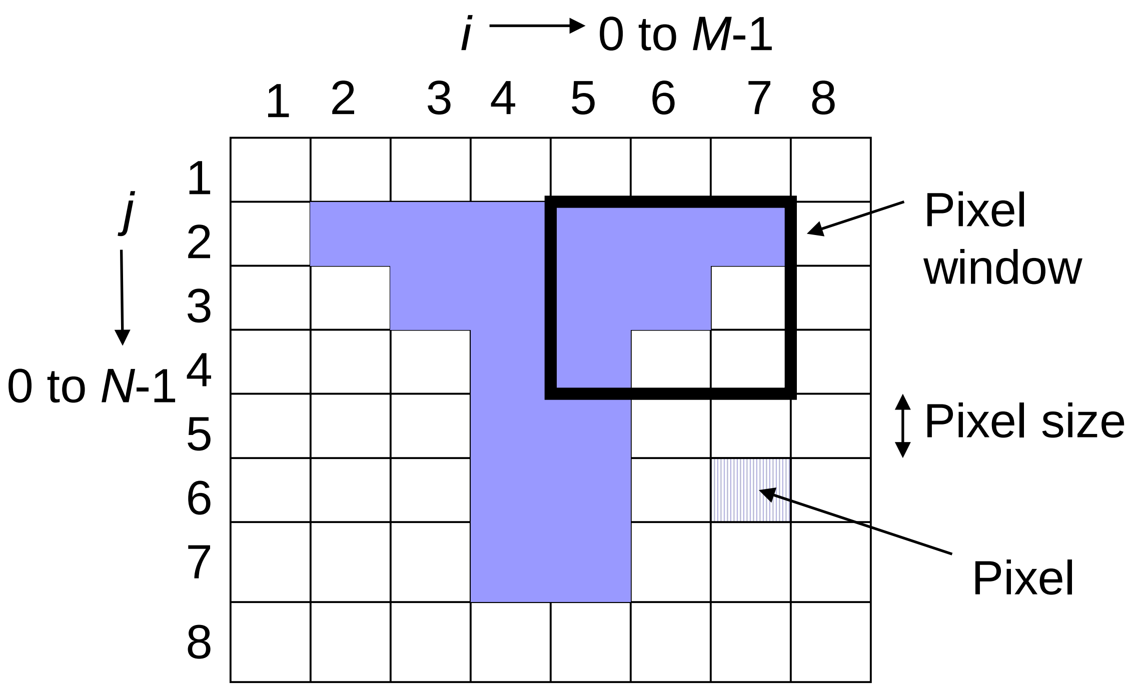


Image space – Pixel matrix

Raster Dimension

Raster dimension (= width x height) is the number of horizontal and vertical samples in the pixel grid.

IMAGE: *M* x *N* matrix of pixels *p*(*i*, *j*)



The pixel values *p*(*i*, *j*) are sorted into the matrix in “natural” order, with *i* (or *x*) corresponding to the column and *j* (or *y*) to the row index. Matlab uses this convention.

This results in $p(i, j) = p_{ji}$ where p_{ji} denotes an individual element in common matrix notation.

Raster Dimension (continue)

- Number of pixels in an image (== raster dimension)
  - Video Graphics Array (VGA) display = 640 by 480 display; 4:3 aspect ratio
    - M = 640 pixels, N = 480 pixels. 640 x 480 = 307,200 pixels or 0.3 megapixels
  - SVGA: 800 x 600 == 0.4 megapixels
  - XGA: 1024 x 768 == 0.8 megapixels
  - 1080i HDTV: 1920 x 1080 == 2.1 megapixels; 16:9 aspect ratio
    - High resolution digital TV format
  - 2K: 2048 x 1536 == 3.1 megapixels; 4:3 aspect ratio
    - Used for digital effects in feature films
  - Common CCD camera: at least 512 x 256 pixels = 131,072 pixels
  - Digital SLR camera: at least 4 million pixels

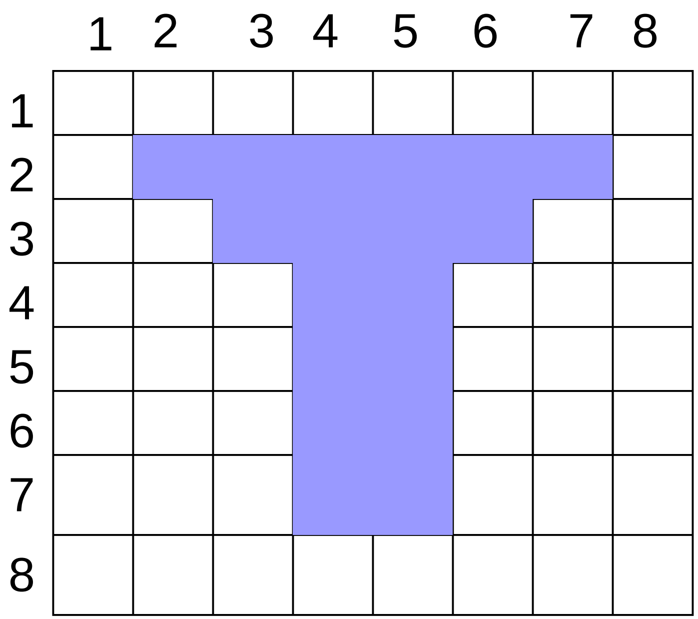
Raster Dimension (continue)

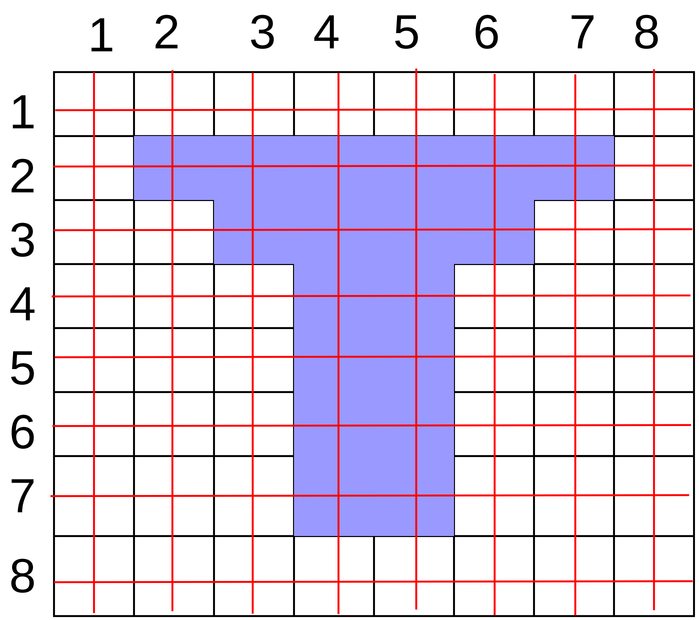
- Scaling (or resampling): the process to create an image with different dimensions from that of the source image.
- Scaling image down (or decimation): the process of reducing the raster dimension.
  - Averaging the values of source pixels contributing to each output pixel
- Scaling image up: the process of increasing the image size to create sample points between the original sample samples in the source raster.
  - Interpolation – using the values in the sample grid to guess the values of the unknown pixels

Image Resolution

- Image resolution is a measure of sampling density.
  - Provides a relationship between pixel dimensions and physical dimensions
- Pixels per inch (ppi)
  - If the dimension of an image is 1 inch x 1 inch, and *M*= *N* = 8, there are 8/1 = 8 pixels per inch.
  - Pixel size = 1/8 = 0.125 inch
  - Pixels per cm

Sampling Grid





3.2 Qualization

- Image quantization - discretion in light intensity
- Brightness level = gray level in monochrome image

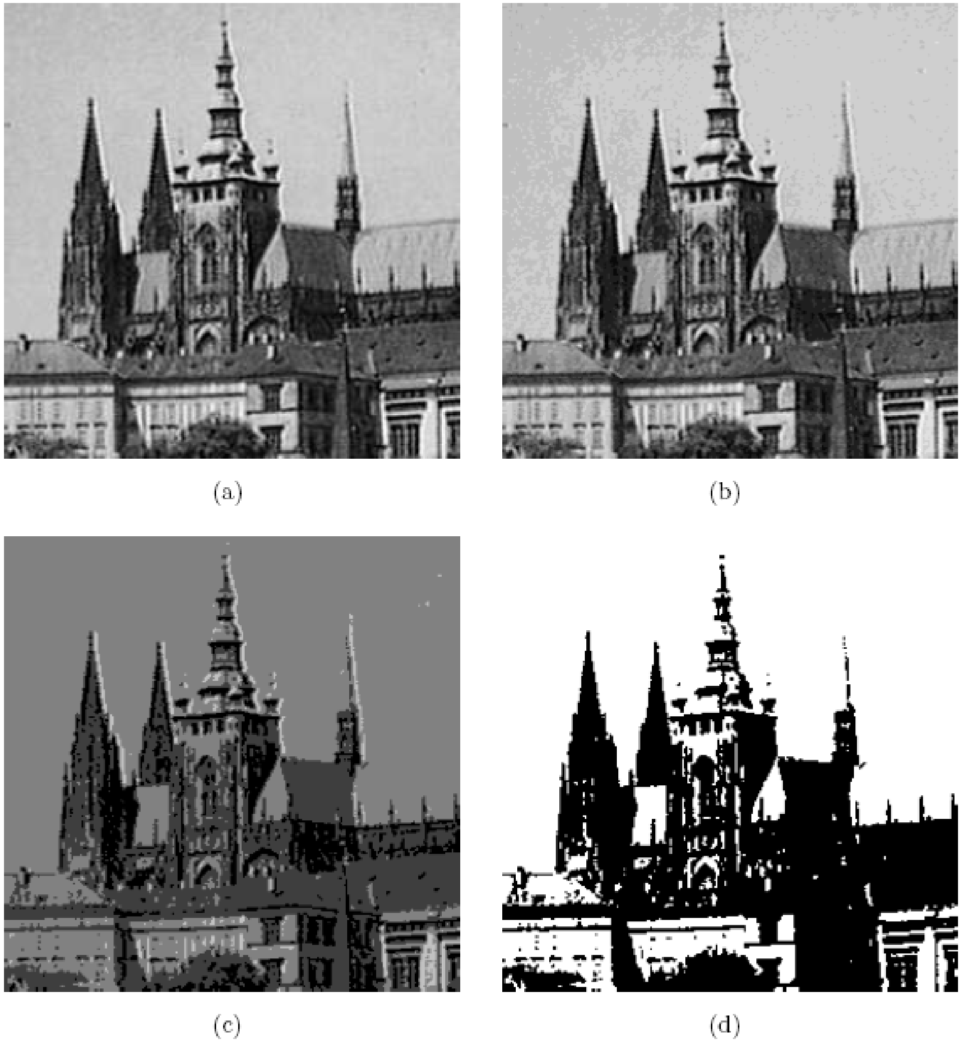
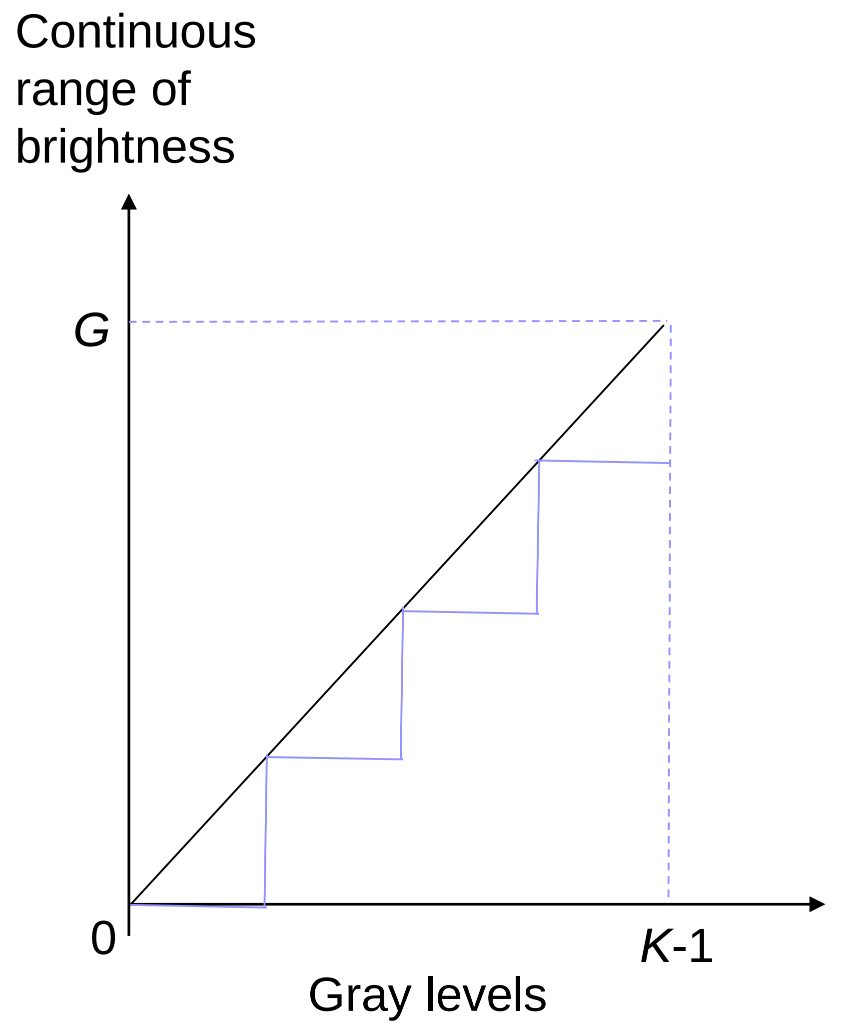


Figure 2.3: Brightness levels. (a) 64. (b) 16. (c) 4. (d) 2.

Quantization

- Quantizes the continuous range of brightness or intensity to *K* gray levels.
- $K = 2^n$ where *n* is known as the color capability or color depth or bit depth.
- *n* is the number of bits used to indicate the color of a single pixel.

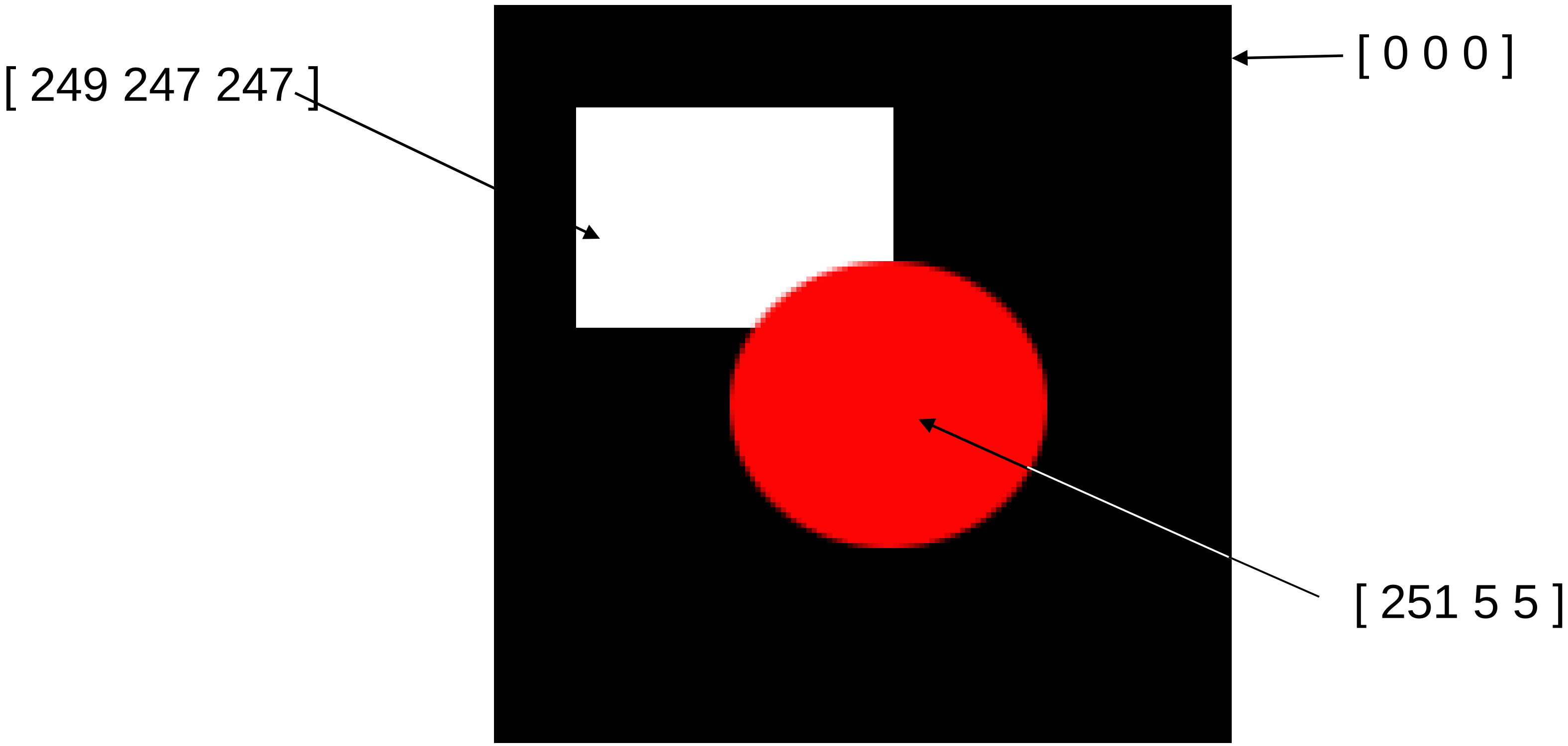


Color Depth

- Maximum number of data a pixel can store
- Monochrome: $p(i,j) = 0, 1, 2, \ldots, K\text{-}1$
- Binary: $p(i,j) = 0, 1$
  - 0: black (current off – no light), 1: white (current on – maximum illumination).
  - Color capability = 1 bit of data
- Grayscale: $p(i,j) = 0, 1, \ldots, 255$
  - 0: black, 1..254: gray, 255: white
  - Color capability = 8 bits of data

Color Depth (continue)

- Color: p(i,j) = [R(i,j) G(i,j) B(i,j)]
  - RGB/8: R(i,j) = 0..255, G(i,j) = 0..255, B(i,j) = 0..255. 8 bit of data per color channel.



Color Depth (continue)

- Indexed Color
  - Color capability = 8 bits per pixel, maximum number of values = 256. Each value is an index number corresponds to explicit color value in the file’s look-up table.
  - The look-up table (color look-up table or colormap) is stored at the header of the image file. Color look-up table can be a hardware device built into an imaging system.

Color Depth (continue)

- Indexed Color
  - A colormap is defined as a list of colors, each indexed by an integer pixel value. Each entry in a colormap is called a color cell. A color cell represents a color usually defined by a set of three numeric values, representing intensities of red, green and blue respectively.
  - Advantages: save memory/storage space and/or transmission time.

- The finer the sampling (i.e., the larger M and N) and quantization (i.e., the larger K), the better the approximation of the continuous image function $f(x,y)$ achieved.
- The number of quantization levels (brightness or gray levels) should be high enough for human perception of fine shading details in the image.

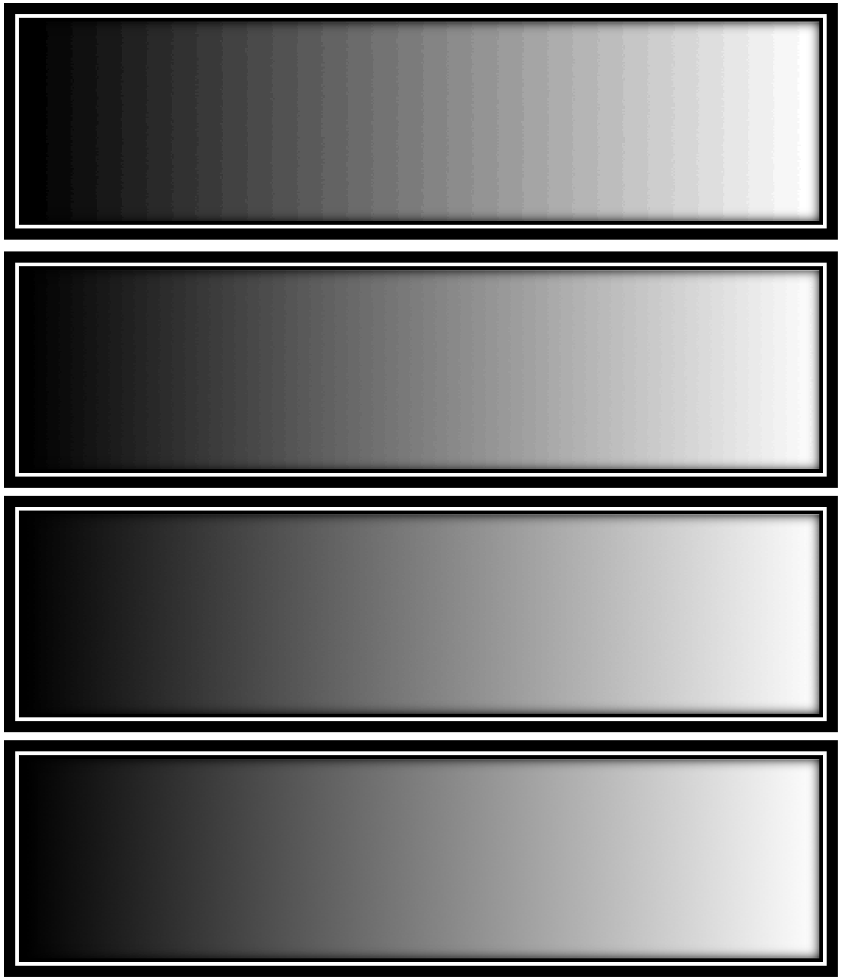
How many gray levels are required for human visual perception?

K = 32

K = 64

K = 128

K = 256

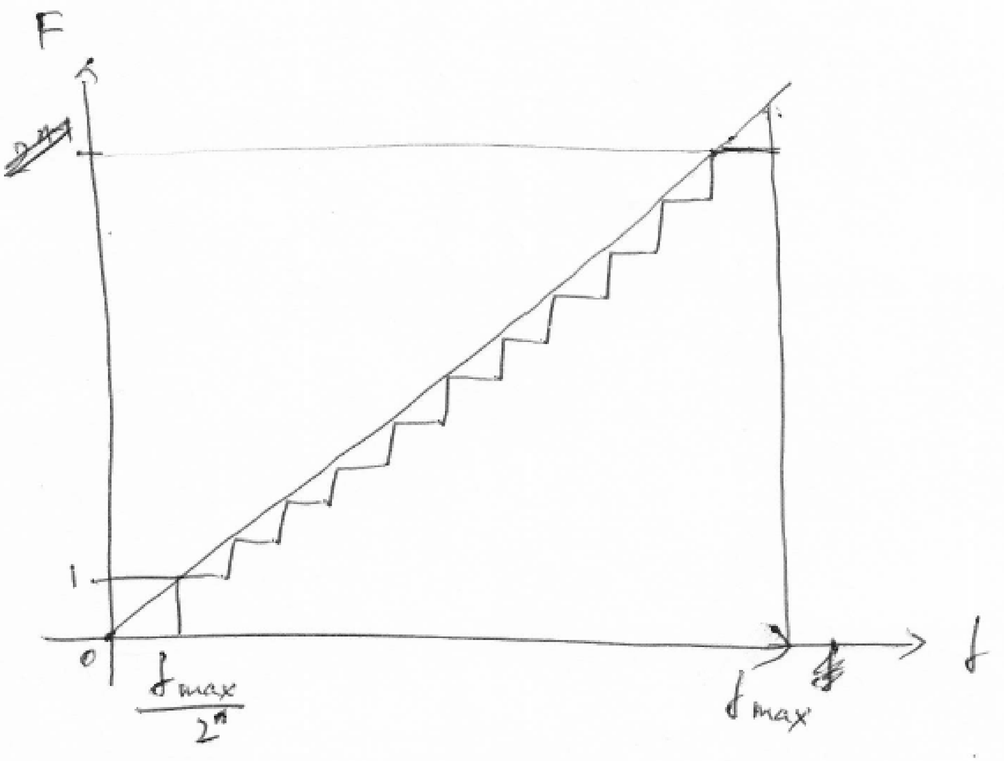


A human is able to recognize about 60 gray levels at most.

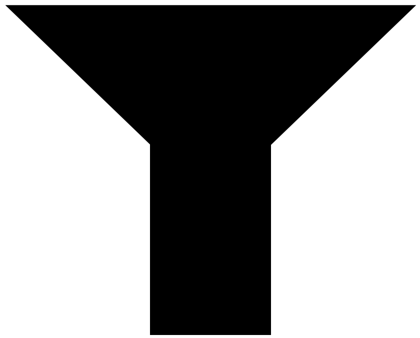
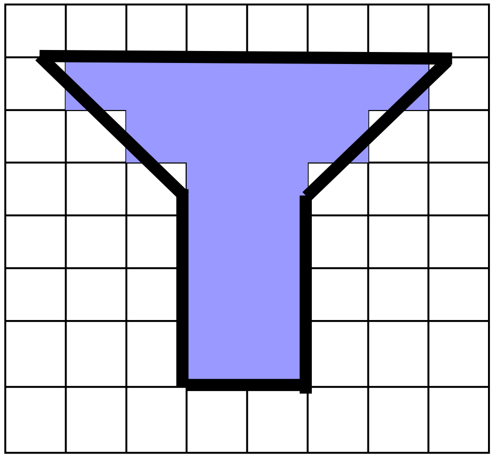
A digital image is usually quantized to 256 gray levels.

- Quantization error is defined as the difference between actual analog value and quantized digital value.
- It is due to rounding or truncation.

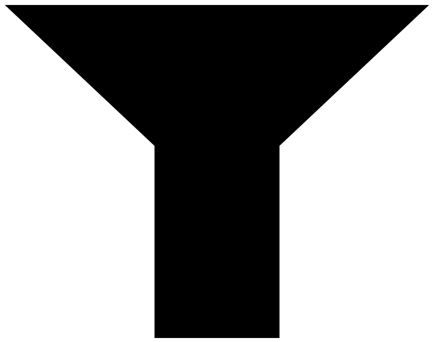
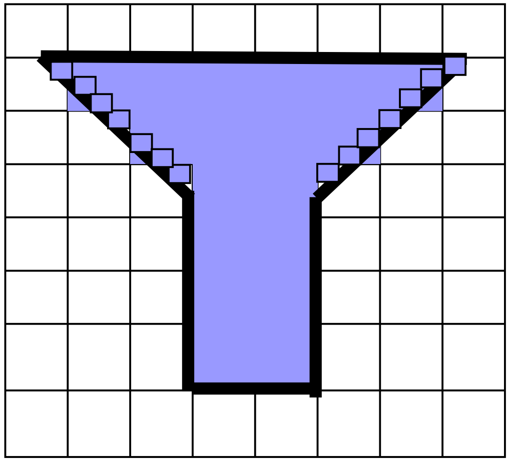
$K = 2^n$



- Pixel size determines the resolution

- Sub-pixel accuracy
  - Many pixels + computational model

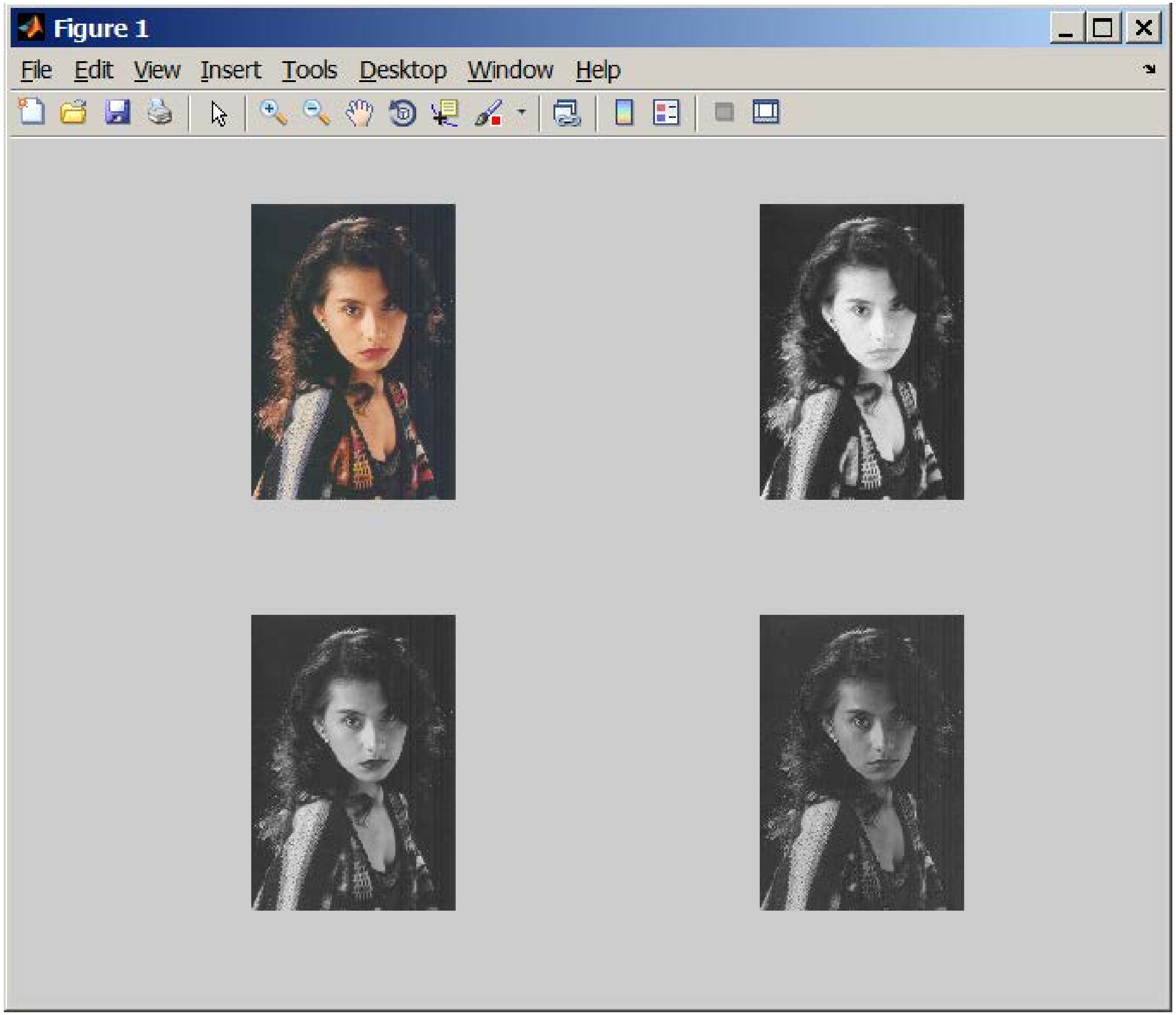
3.3 Matlab: Viewing an image

```
% read the input image
model = imread('class_me5405/model.jpg');
% display the image in a figure
image (model)
axis image
% add title
title('input image of a model');
```

Matlab: Viewing an image

```
% information on sizes, types, class and
   attributes
whos model;
% extract individual color components
modelR = model(:, :, 1);
modelG = model(:, :, 2);
modelB = model(:, :, 3);
```

```
>> figure;
>> subplot(2, 2, 1), subimage(model), axis off;
>> subplot(2, 2, 2), subimage(modelR), axis off;
>> subplot(2, 2, 3), subimage(modelG), axis off;
>> subplot(2, 2, 4), subimage(modelB), axis off;
```

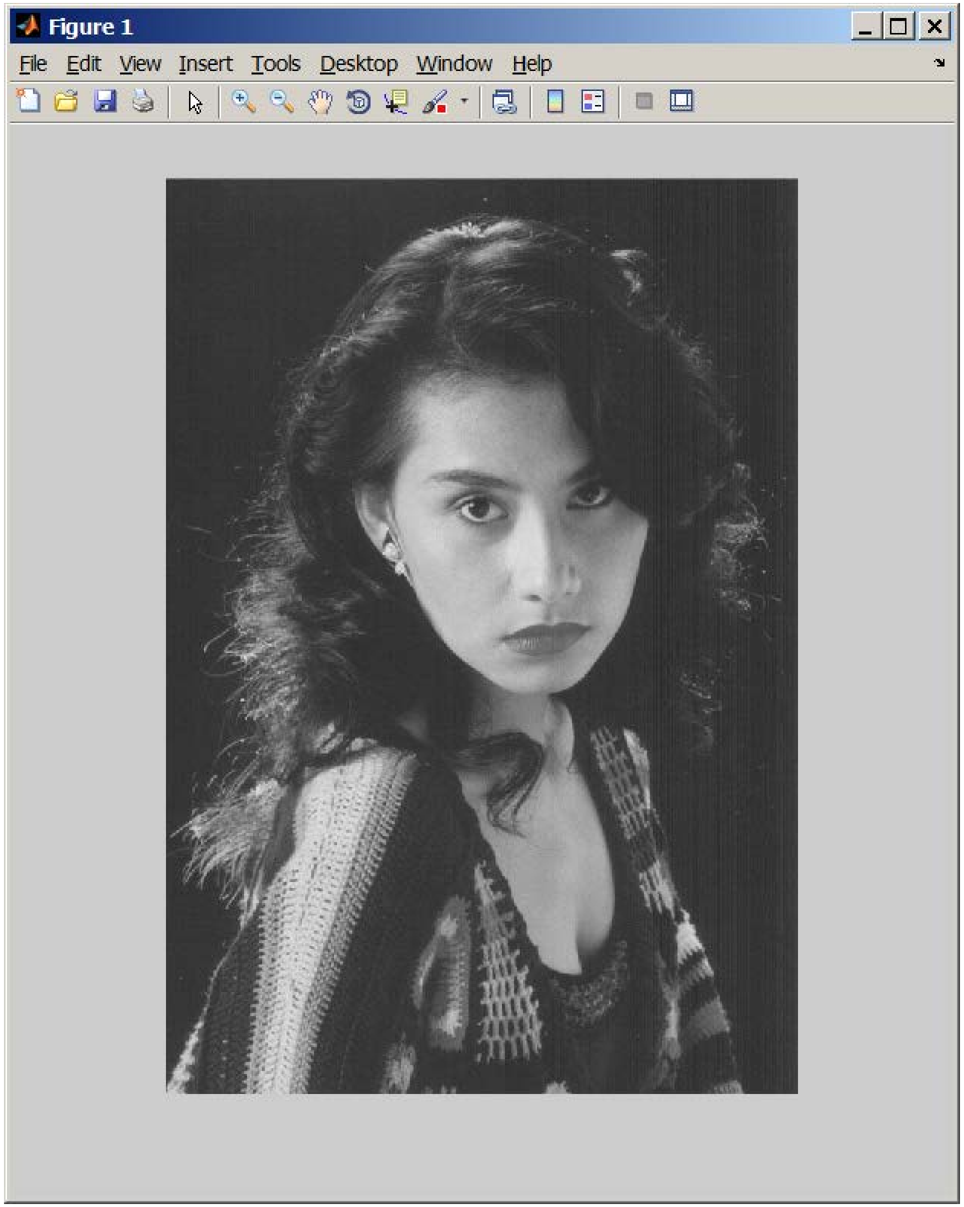


Matlab: Viewing an image

% convert the color image into a grayscale one

modelGray = rgb2gray(model);

```
>> imshow(modelGray);
Warning: Image is too big to fit on
screen; displaying at 67%
> In imuitools\private\initSize at 71
  In imshow at 282

>> size(modelGray)

ans =

   900   620


>> whos modelGray
  Name           Size              Bytes  Class    Attributes

  modelGray      900x620           558000  uint8


>> 900 * 620

ans =

     558000


>> whos model
  Name         Size                Bytes  Class    Attributes

  model      900x620x3           1674000  uint8
```



Summary

- A 2D image gray-scale image is represented by a scalar function $f(x, y)$ of two variables which give coordinates in a plane.
- In many cases, a 2D image is formed as the result of a projection of a 3D scene into 2D.
- The domain of the digitized image is a limited discrete grid the coordinates of which are natural numbers. The range of the digitized image is a limited discrete set of gray values (brightnesses).
- A pixel represents the elemental part of an image.

Summary

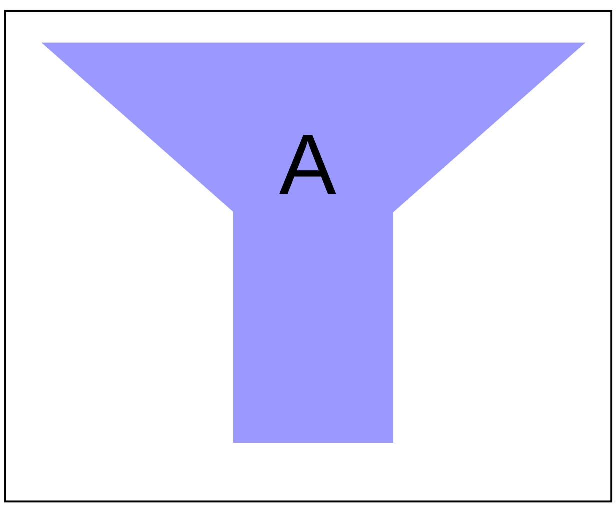
- Sampling considers the image only at a finite number of points and quantization refers to the representation of the brightness level (in grayscale image) or color value (in RGB format) at each sampled point (pixel) using a finite number of bits.
- Sampling distance is the distance between the sampling points. The smaller sampling distance the higher the resolution of the image.
- Gray level quantization governs the appearance of shading and false contour. A human is able to recognize about 60 gray levels at most.
- Images containing only black and white pixels are called the binary.

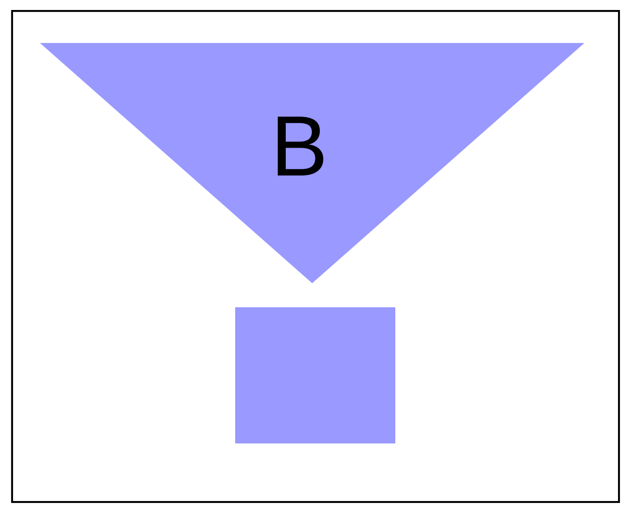
4. Digital Image Properties

- <u>Metric and topological properties</u>
  - A set is a collection of distinct objects.
  - A metric space is a set where a notion of metric (or distance) between elements of the set is defined.
  - A topological space is a mathematical structure that allows the formal definition of connectivity, convergence and continuity.

Connectivity

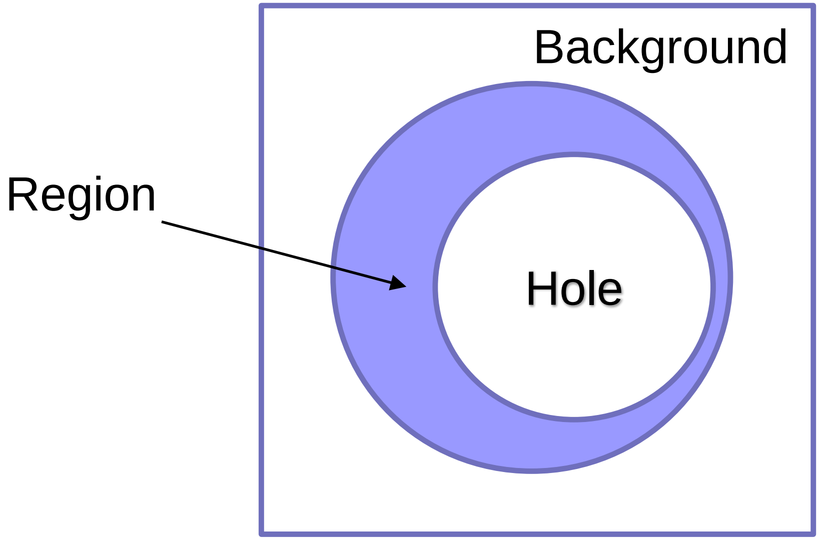
- A connected space is a topological space which cannot be represented as the disjoint union of two or more nonempty open subsets.





Connectivity

- To provide shape information
- To establish objects' components and boundaries

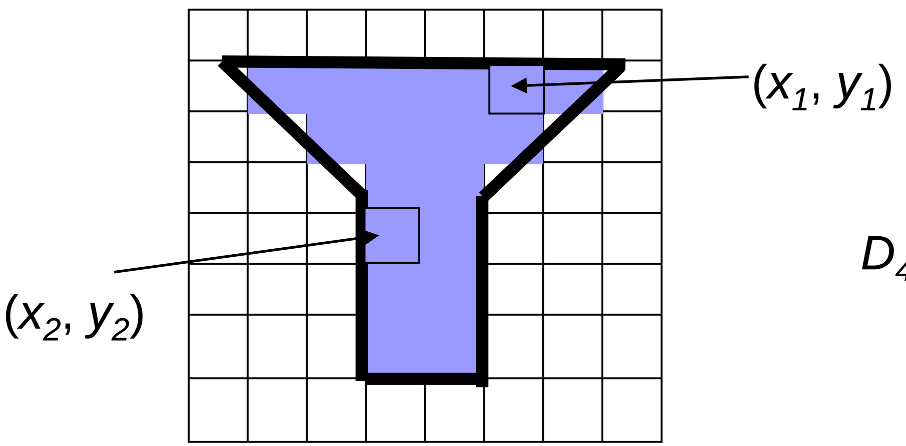


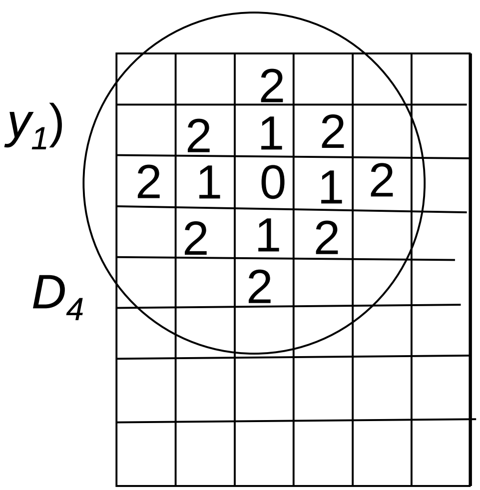
Distance to examine connectivity

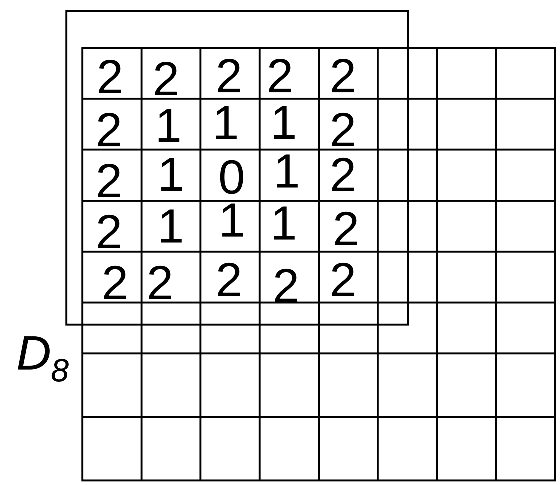
$$\text{Euclidean}: D_E = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\text{City block}: D_4 = |x_2 - x_1| + |y_2 - y_1|$$

$$\text{Chess board}: D_8 = \max(|x_2 - x_1|, |y_2 - y_1|)$$







Distance to examine connectivity

- What is the distance between (4,0) and (0,2)?

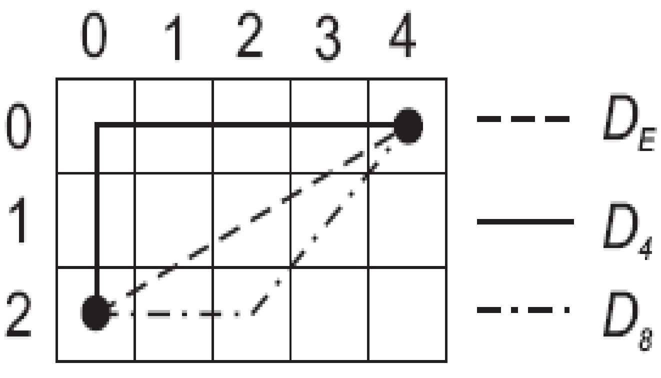
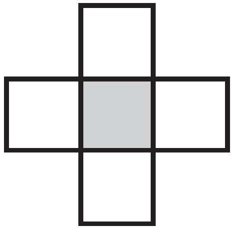


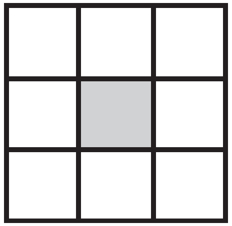
Figure 2.4: Distance metrics D_e, D_4, and D_8.

Typical Connectivity

- 4-neighborhood (or 4-connectivity)
- 8-neighborhood (or 8-connectivity)



(a) 4-neighborhood



(b) 8-neighborhood

Figure 2.5: Neighborhood of the representative pixel (gray filled pixel in the middle).

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

Figure 2.8: Input binary image. Gray pixels correspond to objects and white pixels to background.

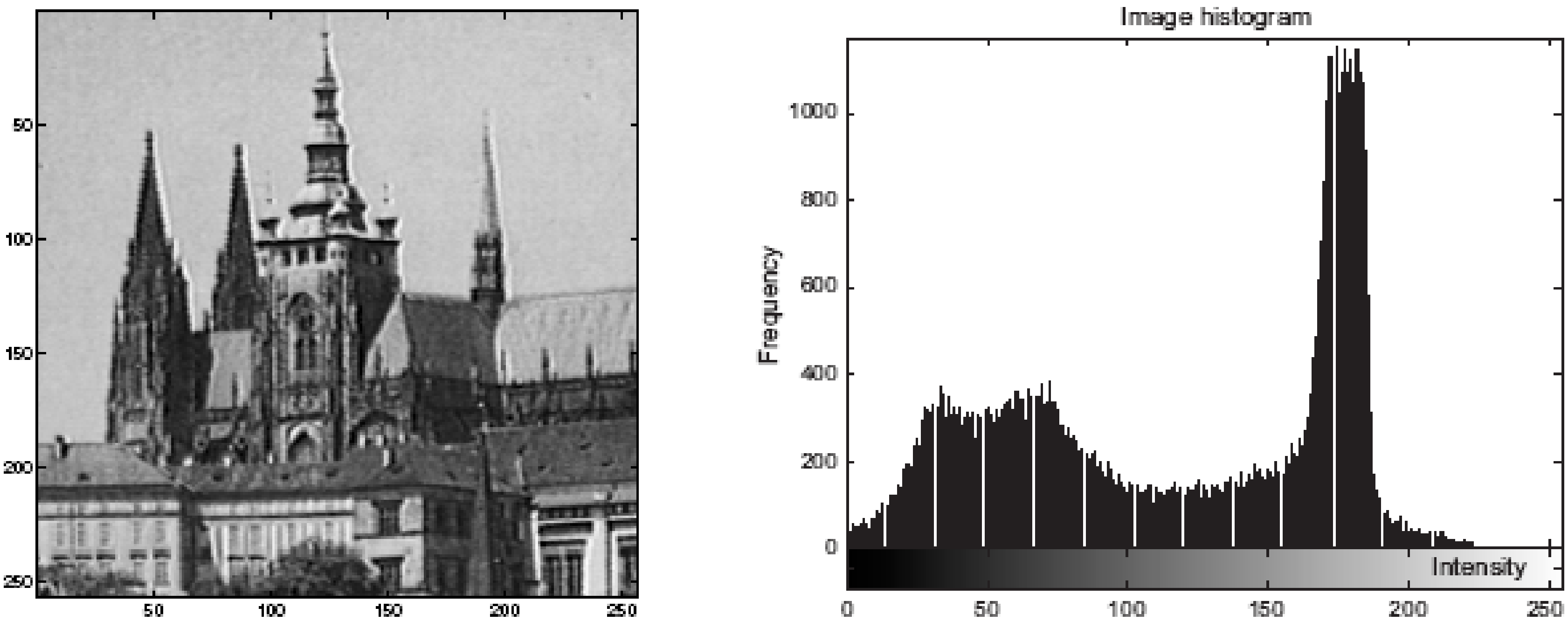
5	4	4	3	2	1	0	1
4	3	3	2	1	0	1	2
3	2	2	2	1	0	1	2
2	1	1	2	1	0	1	2
1	0	0	1	2	1	0	1
1	0	1	2	3	2	1	0
1	0	1	2	3	3	2	1
1	0	1	2	3	4	3	2

Figure 2.9: Result of the distance transform when the distance D_4 is considered in calculations.

Histogram

- The brightness histogram $H(z)$ of an image provides the frequency of the brightness value z in the image.
- The histogram of an image with L gray-level is represent by a 1D array with L elements.

Histogram



(a) (b)

Figure 2.16: Original image (a) and its brightness histogram (b).

Histogram

- Can be achieved by simply counting the number of pixels for each gray level.

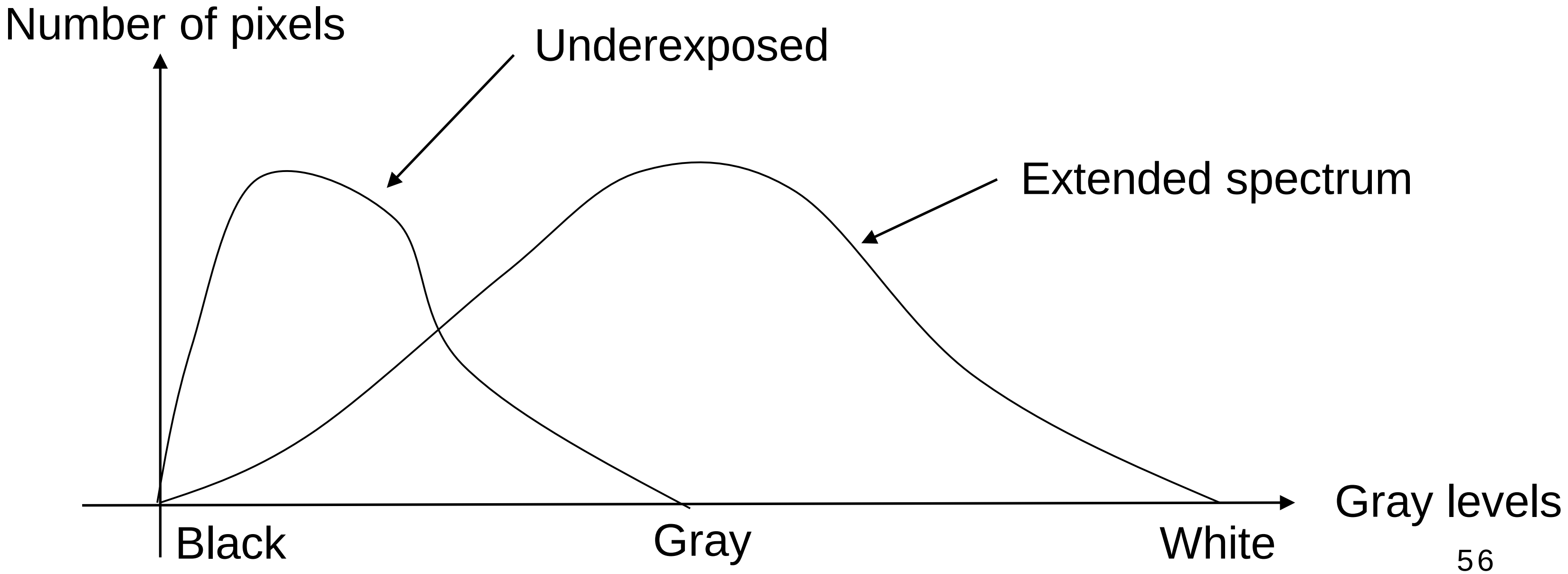
Algorithm 2.2: Computing the brightness histogram

1. Assign zero values to all elements of the array h_f.
2. For all pixels (x, y) of the image f, increment $h_f\big(f(x, y)\big)$ by 1.

```
% assumption: im is a grayscale image with intensity values between 0 and 255
% allocate memory and initialize H
H = zeros(1, levels);
% scan all pixels
for i=1:size(im,1)
        for j=1:size(im,2)
                %pixel intensity indexes the accumulator
                H(im(i,j) + 1)=H(im(i,j) + 1)+1;
        end
end
```

Histogram

- It can detect image with contrast problems, underexposed (too dark) and overexposed (too light).



Histogram

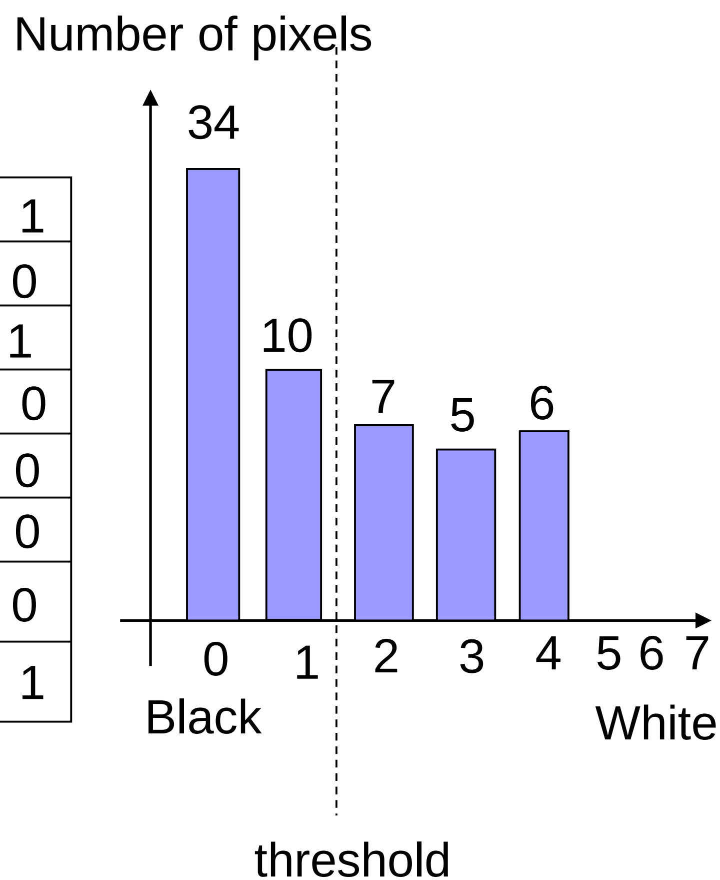
- More than one images can have the same histogram.
- It can remove background using thresholding.
- Invariant to typical image transformation such as rotation.
- It does not relate to object’s shape information.

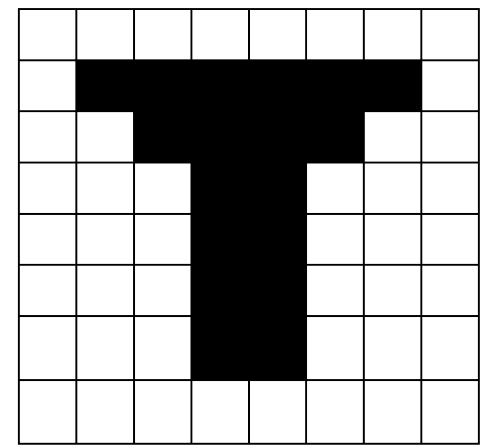
Thresholding

- Thresholding is the simplest method of image processing.
- A binary image is created from a grayscale image by marking individual pixels in an image as “object” pixels if their value is greater than some threshold value (assuming the object is lighter than background) and “background” pixel otherwise.
- Note that objects in an image are not necessary represented using darker shades.

Thresholding

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

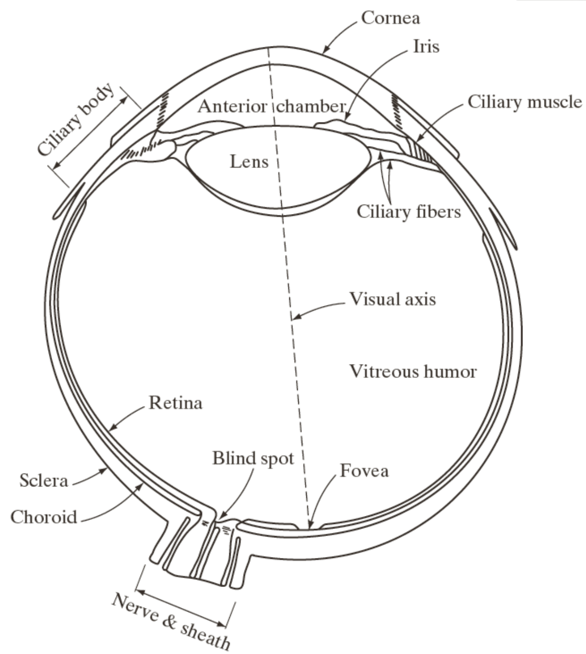




Original image → Histogram → Binary image

Visual Perception

FIGURE 2.1 Simplified diagram of a cross section of the human eye.



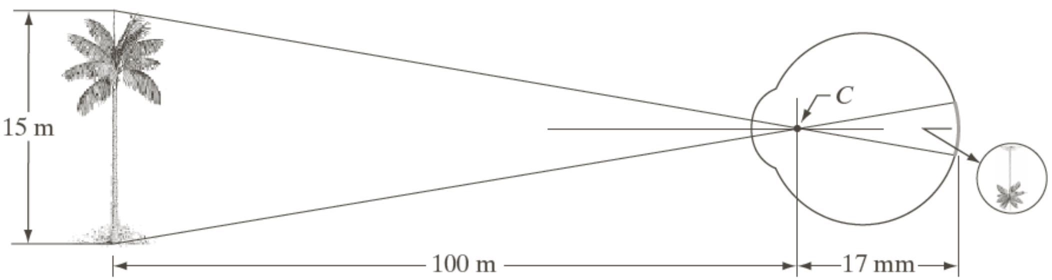
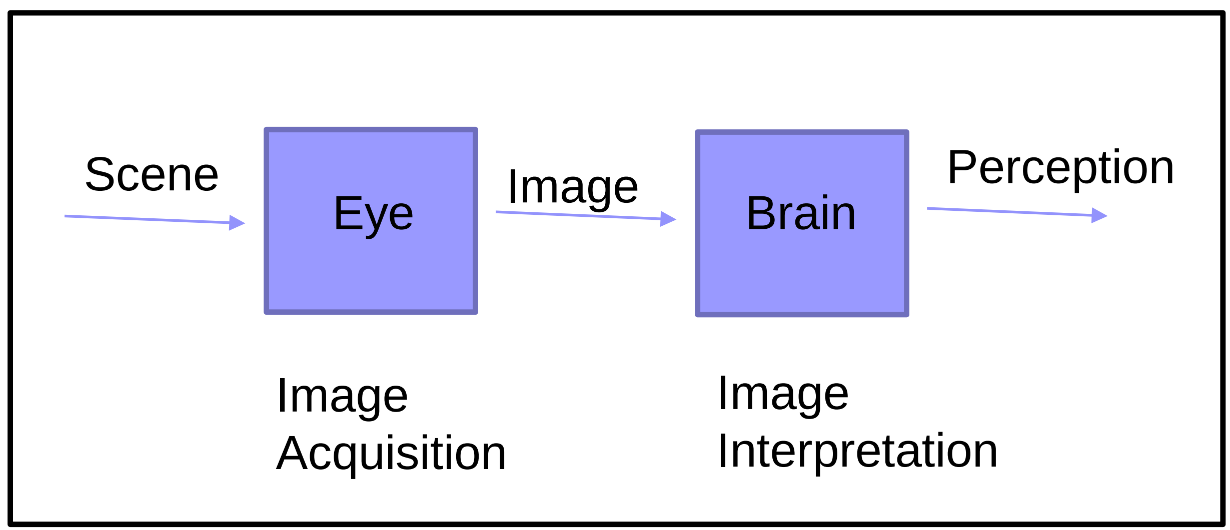


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

Does our eye see the tree?



Visual Perception of the Image

- Contrast
  - Contrast is the local change in brightness.
  - The ratio between average brightness of an object and the background.

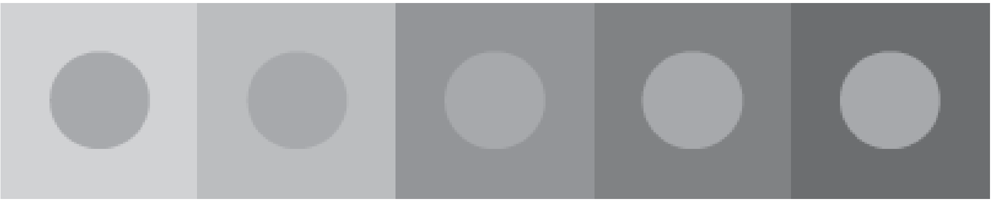


Figure 2.17: Conditional contrast effect. Circles inside squares have the same brightness and are perceived as having different brightness values.

Visual Perception of the Image

- Acuity
  - The ability to detect details in an image.
  - Acuity defines the resolution ability of human eye (sharpness).

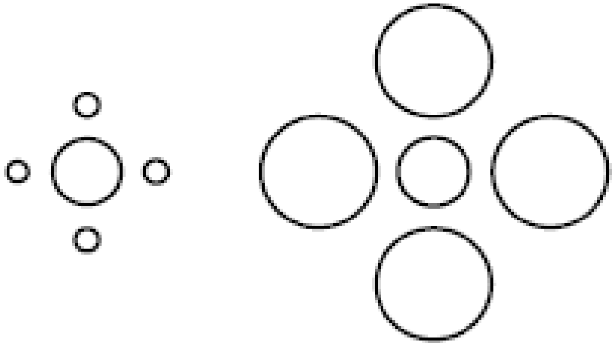


Figure 2.18: The Ebbinghaus illusion.

Visual Perception of the Image

- Some visual illusions

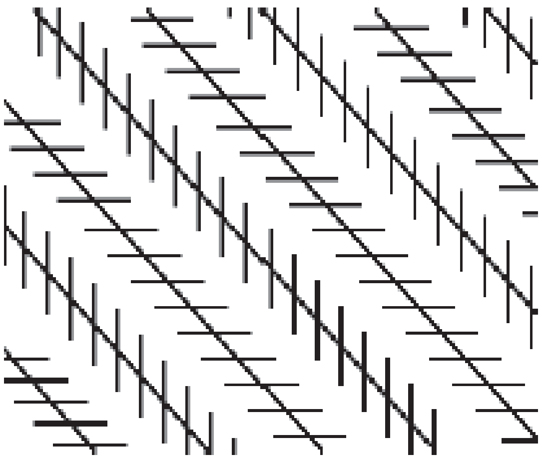


Figure 2.19: Disrupted parallel diagonal lines.

Parallel diagonal line segments are not perceived as parallel.

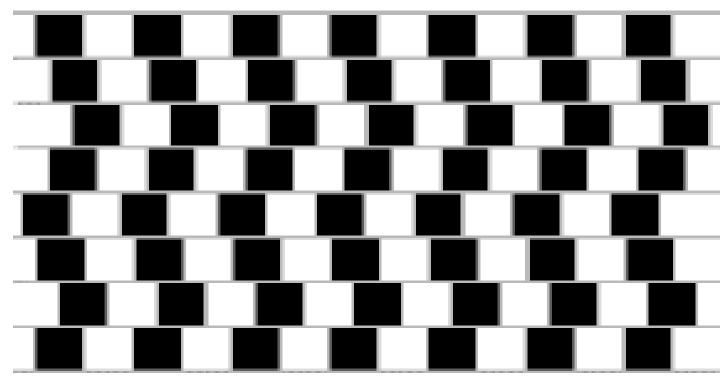


Figure 2.20: Horizontal lines are parallel, although not perceived as such.

Rows of black and white squares are all parallel.

Visual Perception of the Image

- Perceptual grouping
  - The human visual ability to extract significant image relations from lower-level image features without any knowledge of the image content, and group them to obtain meaningful higher level structure.

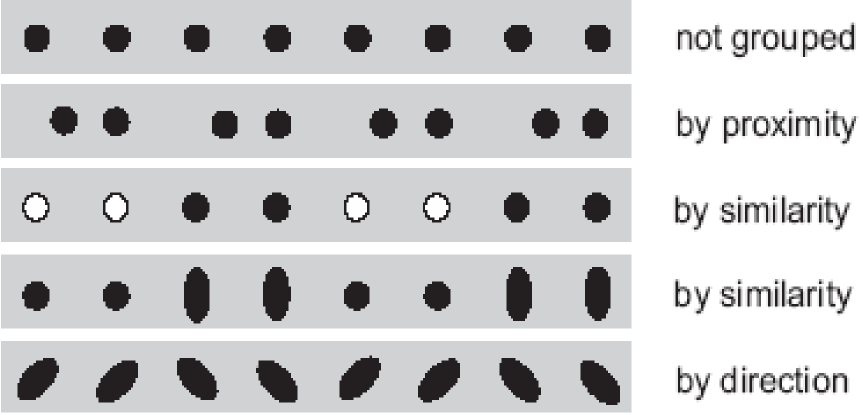


Figure 2.21: Grouping according to properties of elements.

Summary

- The neighborhood relation of a pixel has to be defined to be able to represent discrete geometry.
- A function providing distance between two pixels has to be established—there are several definitions used. The most commonly used is ‘city block’, ‘chessboard’, and the Euclidean distance used in everyday life.

Summary

- The brightness histogram is a global descriptor of the image giving the estimate of the probability density that a pixel has a given brightness.
- Human visual perception is vulnerable to various illusions. Some of the properties of human perception of images as perceptual grouping are inspirations for computer vision methods.

5. Basic Operations

- Image most frequently represented as matrix (or 2D array)
  - Matrices hold image data explicitly. Spatial characteristics are implicitly available.
  - A binary image (an image with 2 brightness levels only) is represented by a matrix containing only zeros and ones.

Basic Operations (Continue)

- A multi-spectral/multi-band image (a collection of several monochrome images of the same scene) is represented by several matrices, each containing information about one multi-spectral image or band.
  - Example: RGB color image consisting of a red, a green and a blue image.

Basic Operations (Continue)

- Arithmetic and logic operations applicable to matrix can be applied to image
  - The arithmetic and logic operations between pixels have been used extensively in image processing.
  - Linearity: relates to vector (linear) spaces where commonly matrix algebra is used.
    - Linear combination permits the expression of a new element of a vector space as a sum of known elements multiplied by coefficients: $ax + by$.

5.1 Arithmetic Operations

- Between two pixels $p = F(x_1,y_1)$ or $f(x_1,y_1)$, and $q = F(x_2, y_2)$ or $f(x_2,y_2)$.
  - Addition: $p + q$
  - Subtraction: $p – q$
  - Multiplication: $p \times q$ (or pq)
  - Division: p/q

Arithmetic Operations (Continue)

- Arithmetic operations on entire images are carried out pixel by pixel.
  - Image addition
    - Image averaging to reduce noise
  - Image subtraction
    - To remove static background information in medical imaging
  - Image multiplication and division
    - To correct gray level shading due to non-uniformilities in illumination or in the image sensor
  - Image multiplication
    - Mask computation for noise reduction

Image Addition

- An acquired image *g*(*x*, *y*) is the superposition of the original image *f*(*x*, *y*) and *n*(*x*, *y*). (Equation (1))
- Assuming that the noise is uncorrelated and has zero average value.
- The noise can be eliminated by averaging a sequence of the acquired images. (Equations (2) and (3))
- As *M* increases, the influence of the noise level at each pixel decreases (Equations (4) and (5))

$$g(x,y) = f(x,y) + n(x,y) \quad (1)$$

$$E\{\bar{g}(x,y)\} = \frac{1}{M}\sum_{i=1}^{M} g_i(x,y) \quad (2)$$

$$E\{\bar{g}(x,y)\} = f(x,y) \quad (3)$$

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{M}\sigma^2_{n(x,y)} \quad (4)$$

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}}\sigma_{n(x,y)} \quad (5)$$

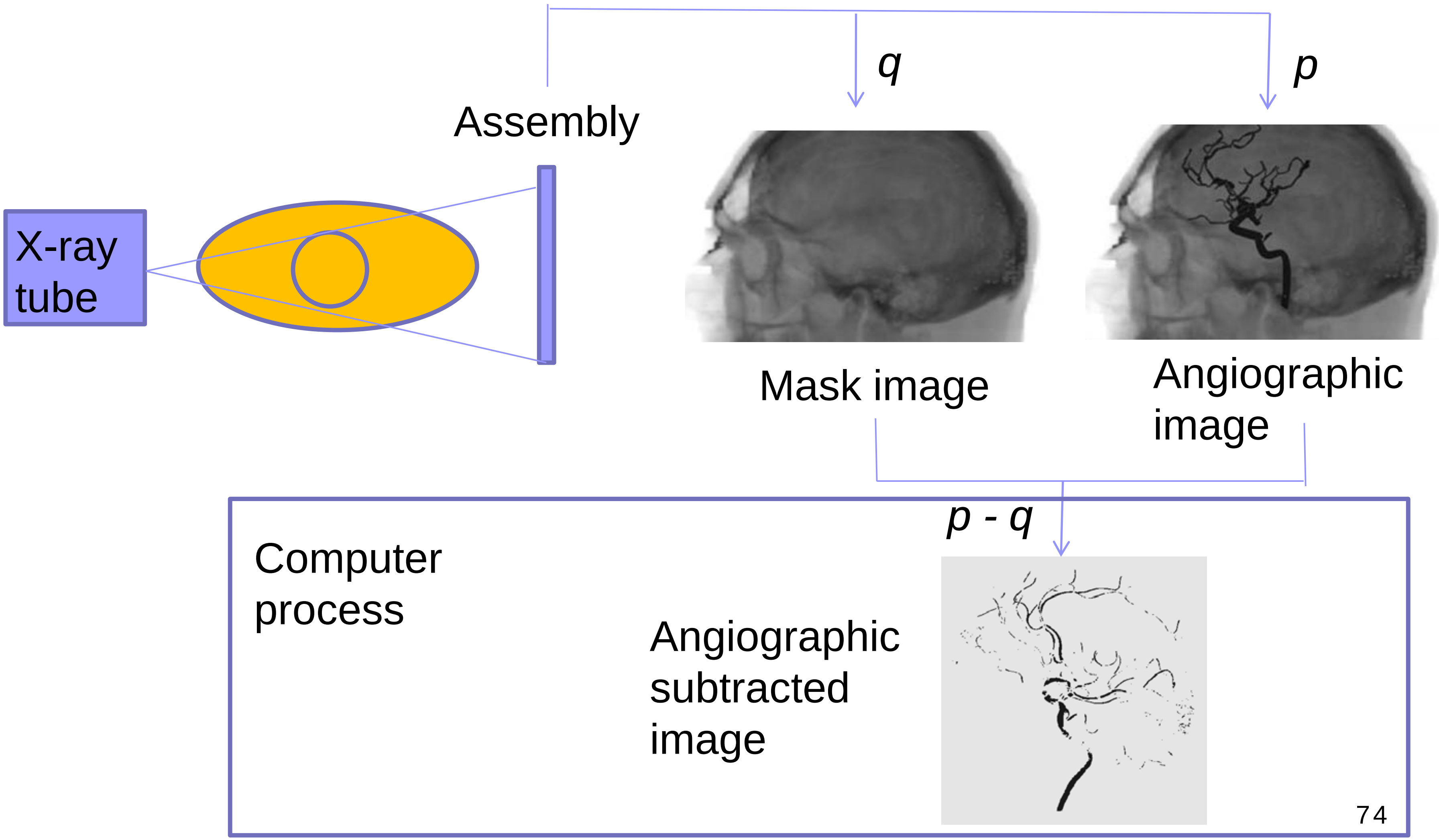
Image Subtraction

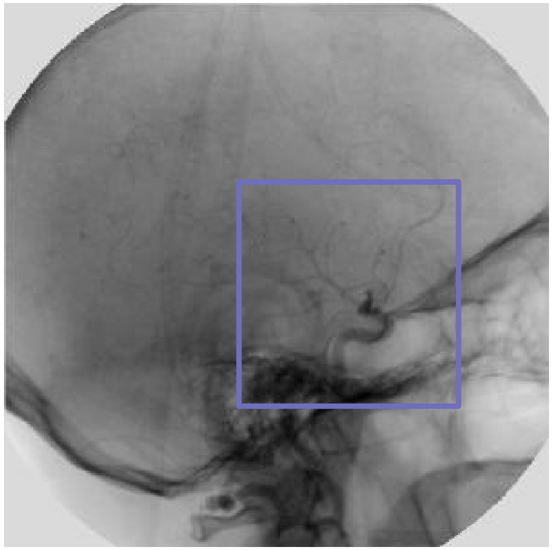
- The difference between 2 images $f(x, y)$ and $h(x, y)$ can be obtained by computing the difference between all pairs of corresponding pixels from f and h.

$$g(x, y) = f(x, y) - h(x, y)$$

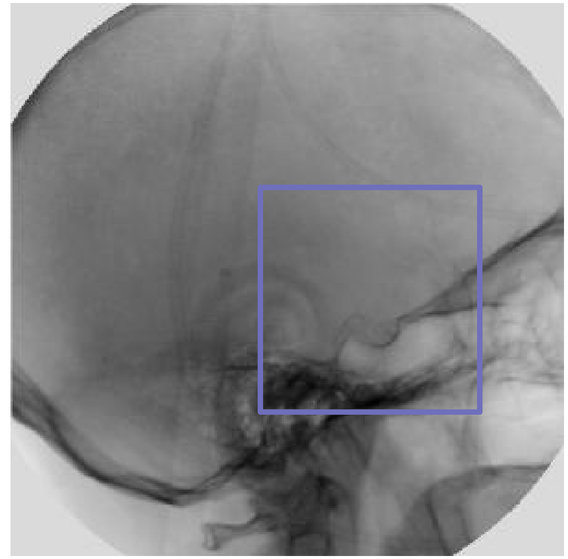
- Example: Digital subtraction angiography (DSA)

Digital Subtraction Angiography

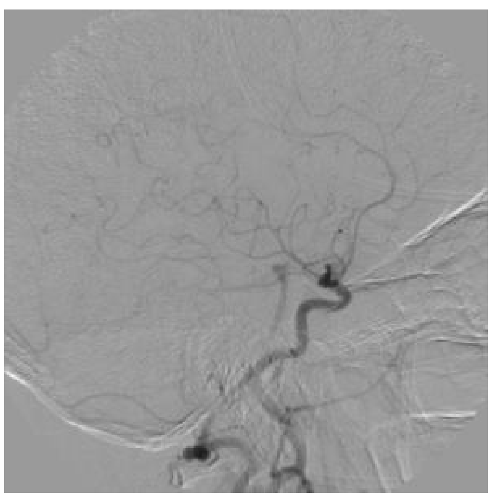




-



=



Image/Contrast Enhancement

→

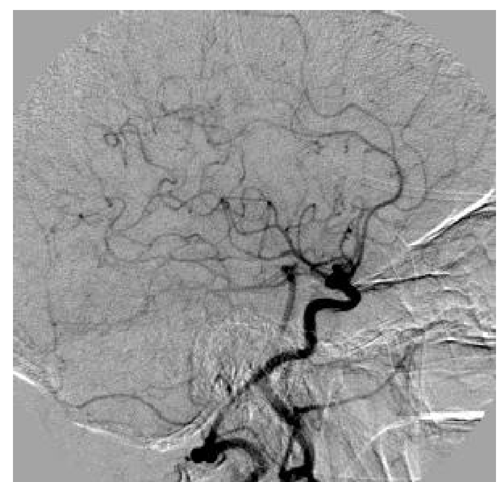


Image Multiplication

- Mask computation/operation
  - The value assigned to a pixel is a function of its gray level and the gray level of its neighbors.
  - We can replace the value of z_5 with the average value of the pixels in a 3 x 3 region centered at the pixel z_5 in Figure 1 (Equation (1)).
  - The same operation can be performed by centering a mask in Figure 2 at z_5 multiplying each pixel under the mask by the corresponding coefficient, and adding the results. (Equation (2)).

$$z = \frac{1}{9}(z_1 + z_2 + \ldots + z_9) = \frac{1}{9}\sum_{i=1}^{9} z_i \qquad (1)$$

$$z = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i \qquad (2)$$

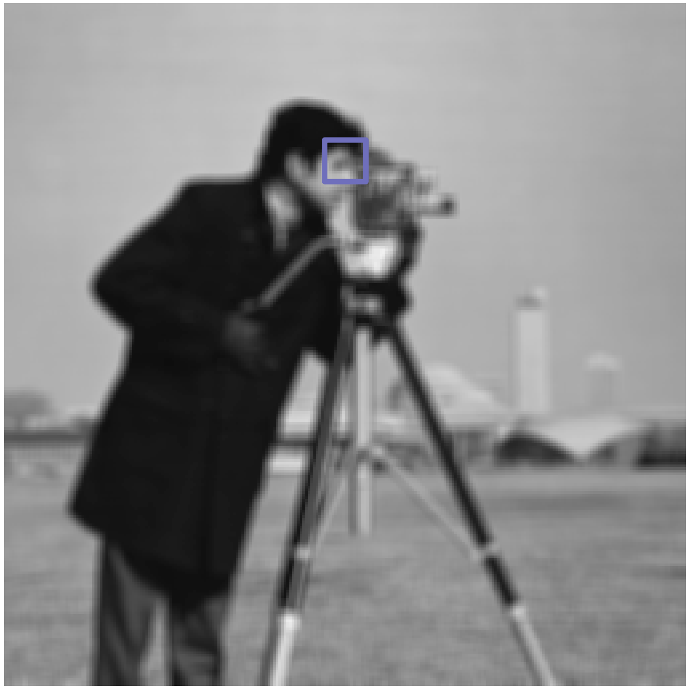
*z*1	*z*2	*z*3
*z*4	*z*5	*z*6
*z*7	*z*8	*z*9

Figure 1. Sub-image

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

Figure 2. Mask

$$.* \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



$$.*\frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Matlab Implementation of Image Arithmetic Operations

```
% read the first image
Flow_gray = imread('flow_web_gray.jpg');
model_gray = imread('model_web_gray.jpg');
whos flow_gray model_gray
```

Name	Size	Bytes	Class	Attributes
flow_gray	900x620	558000	uint8	
model_gray	900x620	558000	uint8	

Matlab Implementation of Image Arithmetic Operations

```
% scale an image
flow_duller = 0.5 * flow_gray;
flow_brighter = 1.5 * flow_gray;
% compare the images
subplot(1,3,1);
image(flow_gray);
axis off equal tight;
subplot(1,3,2);
image(flow_duller);
axis off equal tight;
subplot(1,3,3);
image(flow_brighter);
axis off equal tight;
```

Matlab Implementation of Image Arithmetic Operations

```
% add two images
combined = model + flow_duller;
% display the image
subplot(1,1,1);
cla;
image(combined);
axis equal; axis off;
```

5.2 Logical Operations

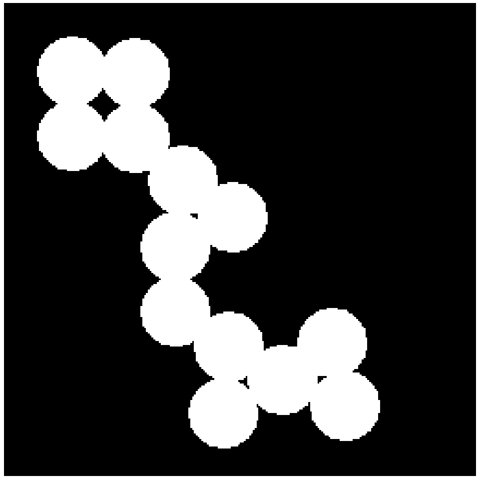
- Most applicable to binary image. A binary image contains only 0’s and 1’s (logical).
  - 0 = black; 1 = white
- AND : intersection (*p* AND *q*)
- OR : union (*p* OR *q*)
- C : complement (NOT *p*)
- XOR: *p* XOR *q*

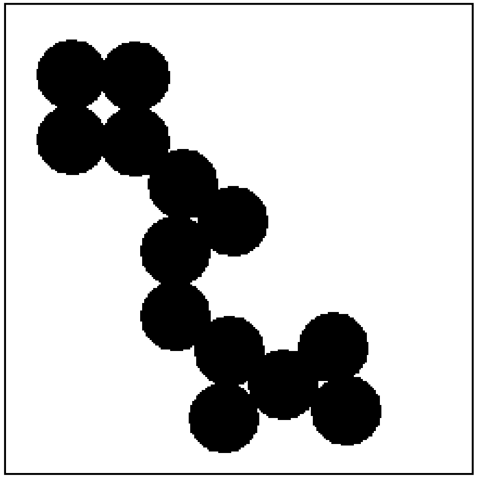
Combined Logic Operations

- Combining basic logic operations can produce other operations.
  - (A) AND (NOT(B))
  - (NOT(A)) AND (NOT(B))
  - (NOT(A)) XOR (B)
  - …

Matlab Implementation of Image Logical Operations

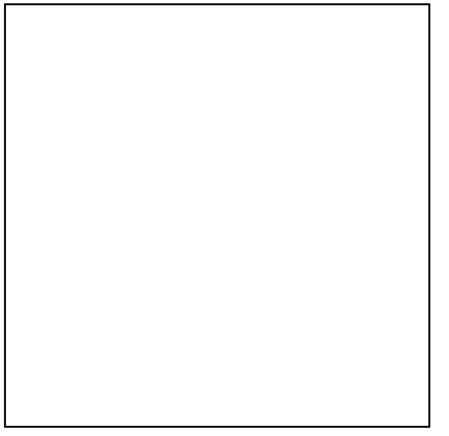
```
% read image
BW = imread('circles.jpg');
% display the image
imshow(BW);
% to invert a binary image
invert_BW = ~BW;
imshow(invert_BW);
```





Matlab Implementation of Image Logical Operations

```
% and operation
imshow(BW & invert_BW);
% or operation
imshow(BW | invert_BW);
```



Computational Requirement

- A typical medical image has a resolution of 512x512 16-bit pixels. This will require 0.5 Mb of storage.
- For one frame, there will be 512 x 512 = 262,144 pixel operations.
- With a 100 MHz or 100 x 10^6 operations per second computer, it will take

  $$262144 / 100 \times 10^6 = 0.00262144 \text{ s} = 2.62144 \text{ ms}$$

  to perform the image operation.
- The rate for real time system is 60-80 frames per second. Each frame need to be processed in 12-16 ms.
- With complex image processing operations that require more than one operation per pixel, hardware acceleration and/or parallel processing will be required.

6. Image Enhancement Using Point Processing

- Some definitions and classification
  - Image pre-processing refers to operations with images at the lowest level of abstraction—both input and output are intensity images.
  - The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing.
  - Hence, image pre-processing can be classified into image enhancement and image restoration.
  - Point processing is a type of image enhancement, also known as pixel brightness transformation. The size of pixel neighbor is 0. Point processing operates only one one pixel.

Point Processing

- In point processing, transformation is defined pixel by pixel.
- The transformation *T* has been referred to as gray-scale transformation or gray level mapping function.
- It changes the brightness of a pixel with no consideration of the pixel position
- The mapping function can be specified in different ways, such as piecewise linear function, or based on the histogram of the input image.

$$\boxed{P'(i,j) = T(P(i,j))}$$

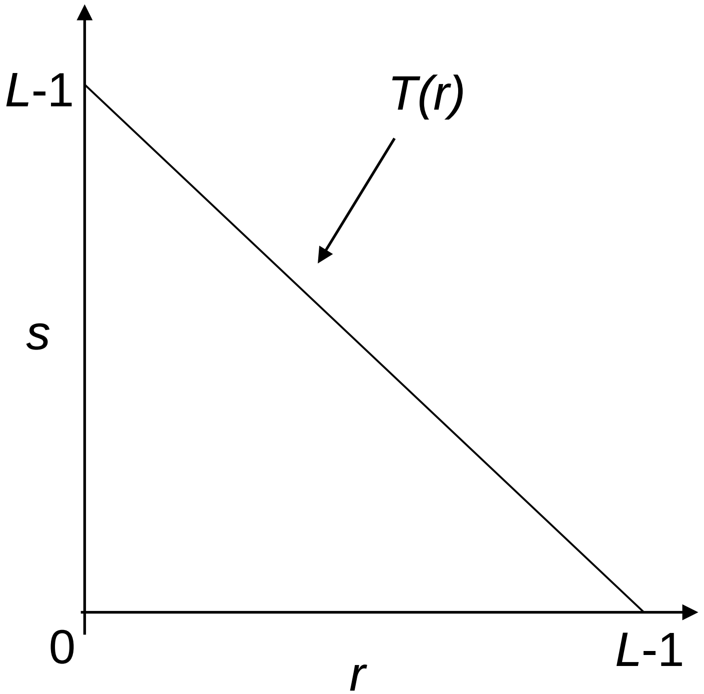
6.1 Gray-Scale Transformation

- The most common gray scale transformation are
  - Negative transformation
  - Brightness thresholding that results in a black and white image
  - Piecewise linear function that enhances the image contrast between two specific brightness values.

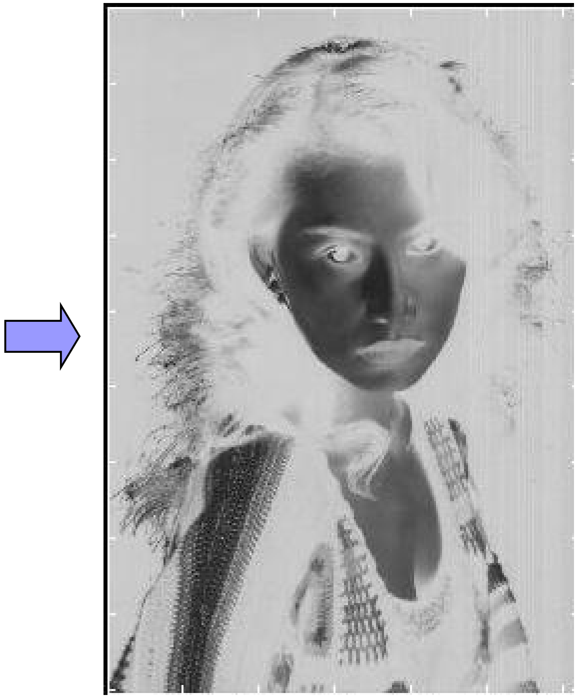
Negative Transformation

- Also known as negative image.
  It inverts the "color" of an image.

$$s = T(r) = L - 1 - r$$



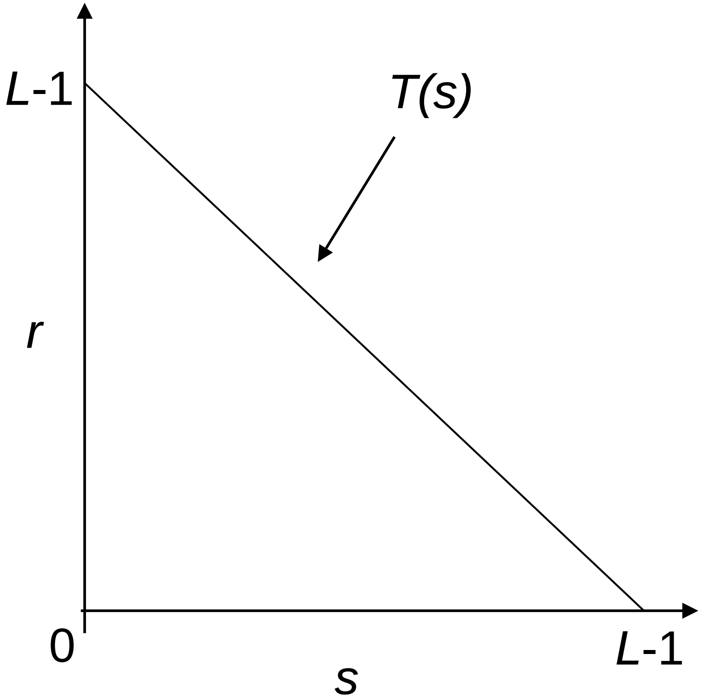
p(i,j)

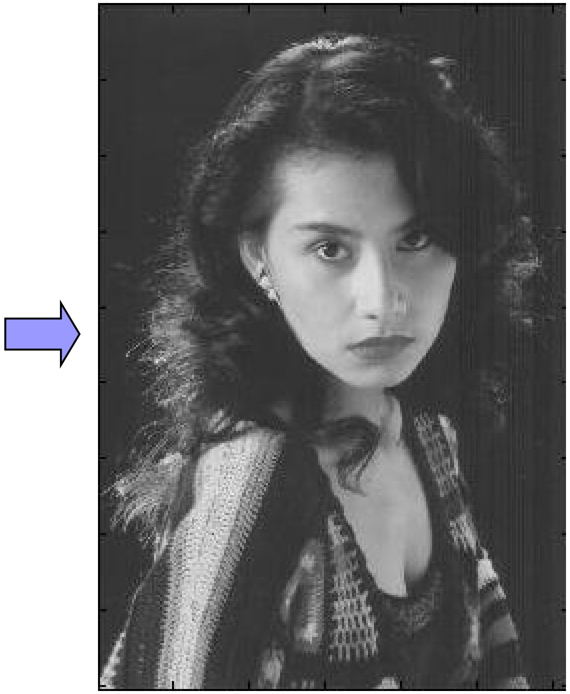


T(p(i,j))

Negative Transformation (Continue)

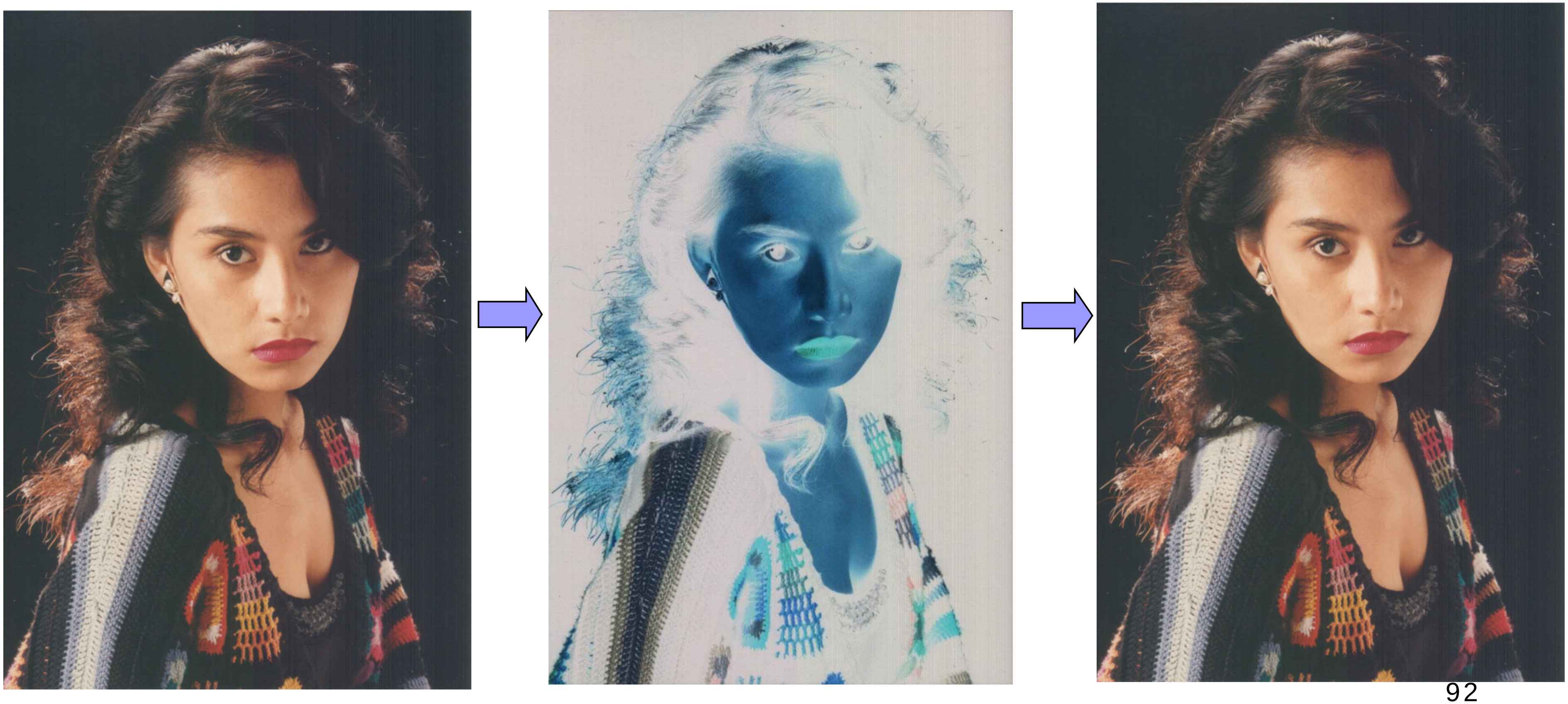
- Can we recover the original image by inverting the gray scale of a negative image?

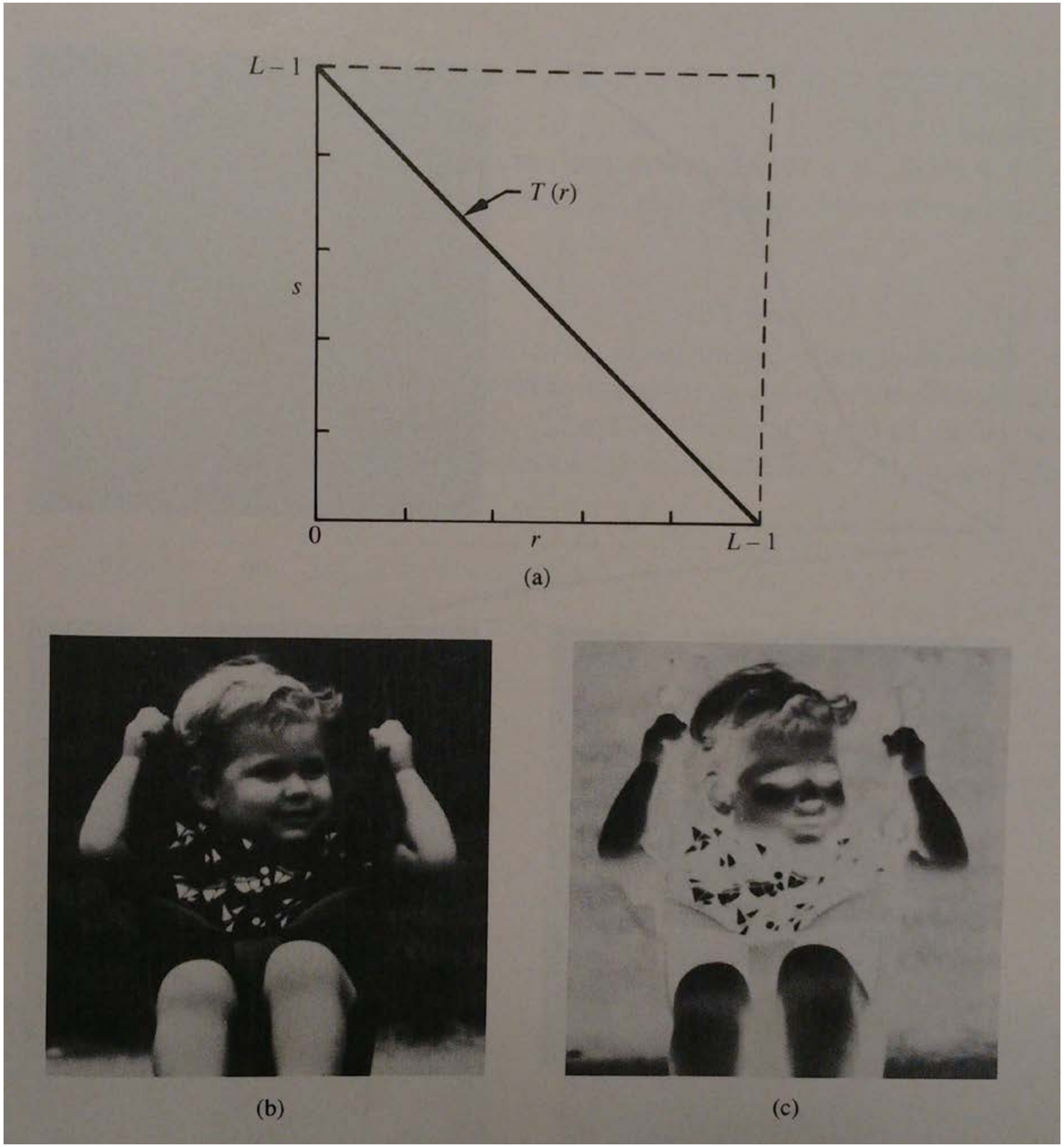




Negative Transformation (Continue)

- Example – invert colors of a color image





(a)

(b) (c)

Obtaining the negative of image:

(a) gray-level transformation function, *r* and *s* denote the input and output gray levels respectively;

(b) an image;

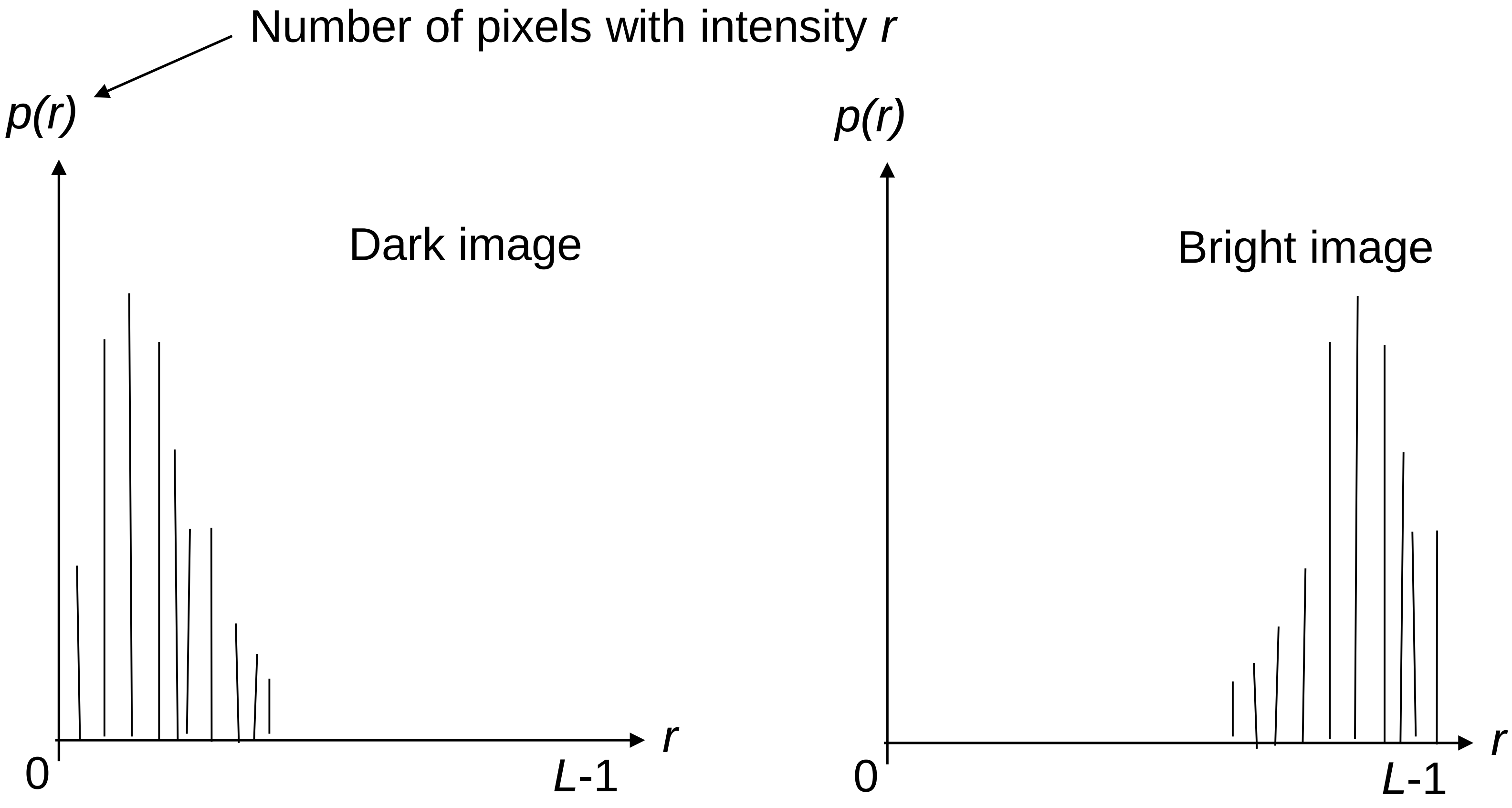
and (c) negative of the image.

Histogram

- The histogram of an image shows the distribution of the pixel values in the image over the intensity (or dynamic) range.
- For a 8-bit gray scale image, the pixel values typically from 0 to 255.
- The *i*th item of the histogram is $p(i) = n_i / N$, $i = 0..255$.
- It represents the probability of the a randomly chosen pixel has the gray level *i*, where n_i is the number of pixels of gray level *i*, and *N* is the total number of pixels in the image. (probability density function).

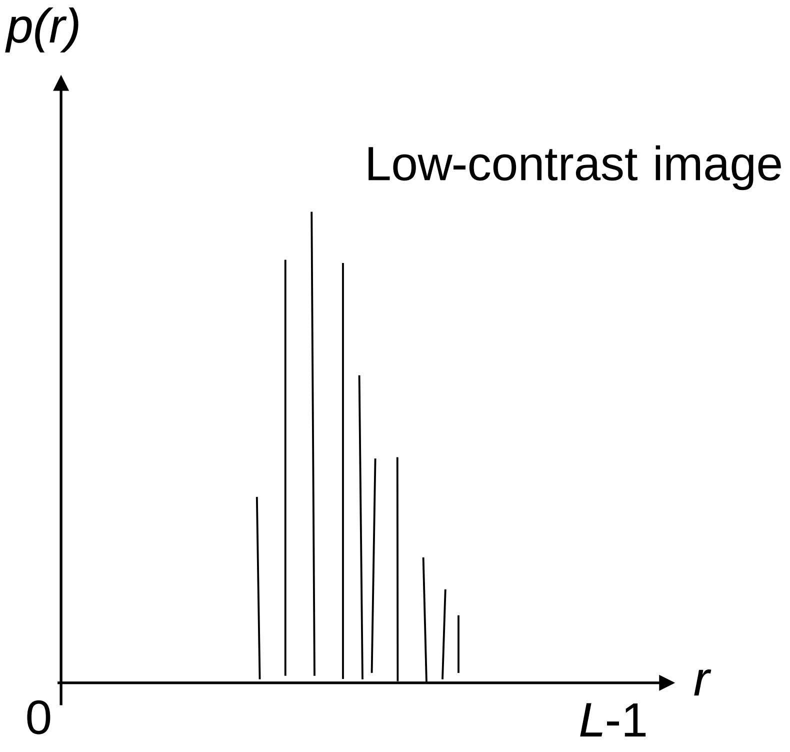
Histogram: Dark and Bright Images

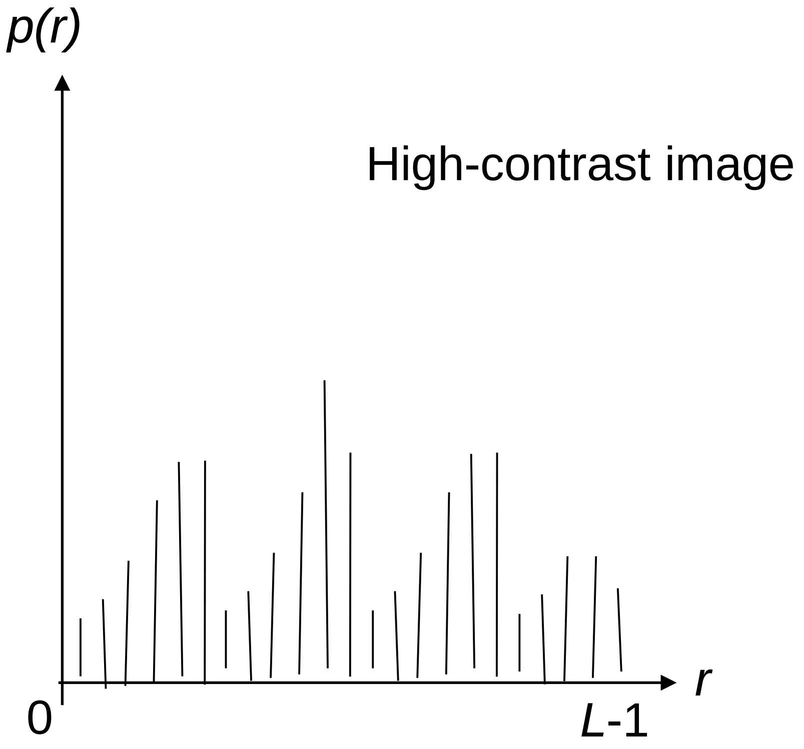
- Brightness refers to the overall lightness or darkness of an image.



Histogram: Low-contrast and High-contrast Images

- When there are no sharp differences between black and white in an image, the image lacks contrast or does not have sufficient contrast.





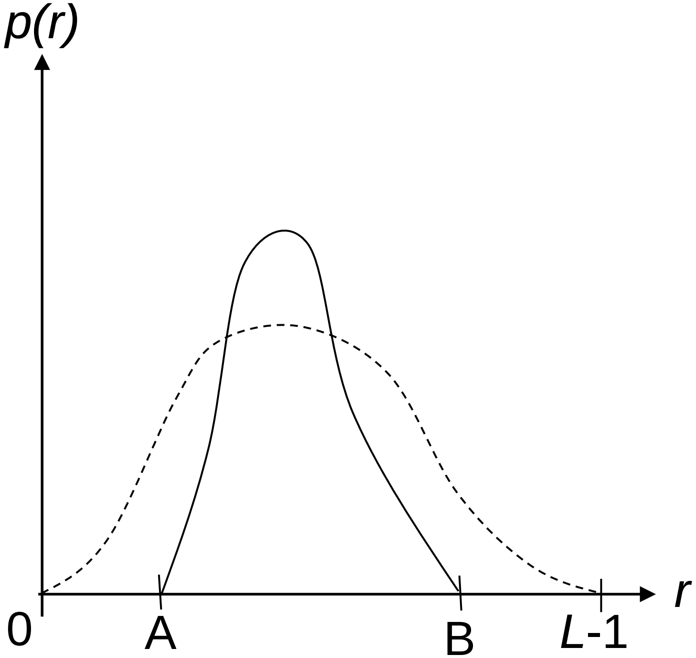
Histogram Processing

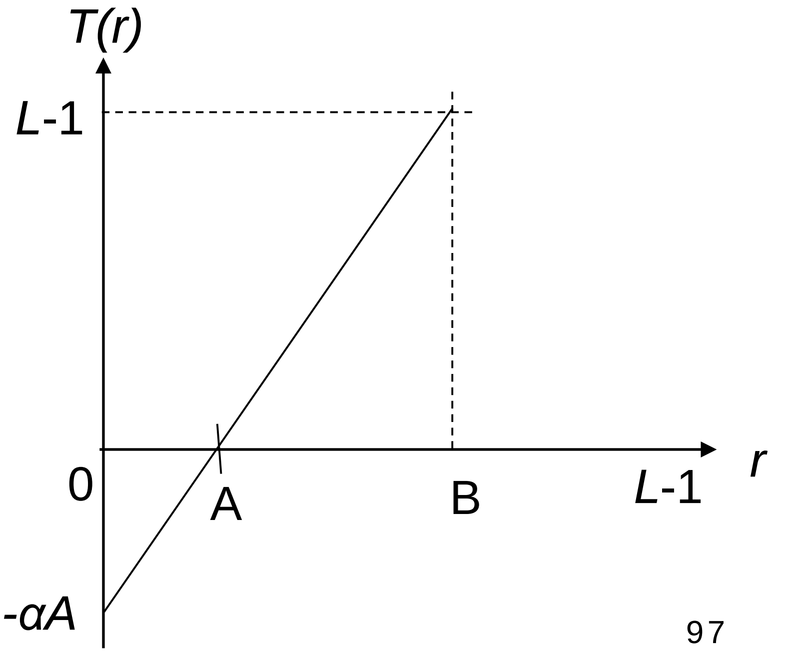
- Applying a linear function to transform the histogram.

$$f(r) = \alpha r + \beta$$

$$0 \equiv f(\mathrm{A}) = \alpha \mathrm{A} + \beta \Rightarrow \beta = -\alpha \mathrm{A}$$

$$L-1 \equiv f(\mathrm{B}) = \alpha \mathrm{B} + \beta \Rightarrow \alpha = \frac{L-1}{\mathrm{B}-\mathrm{A}}$$

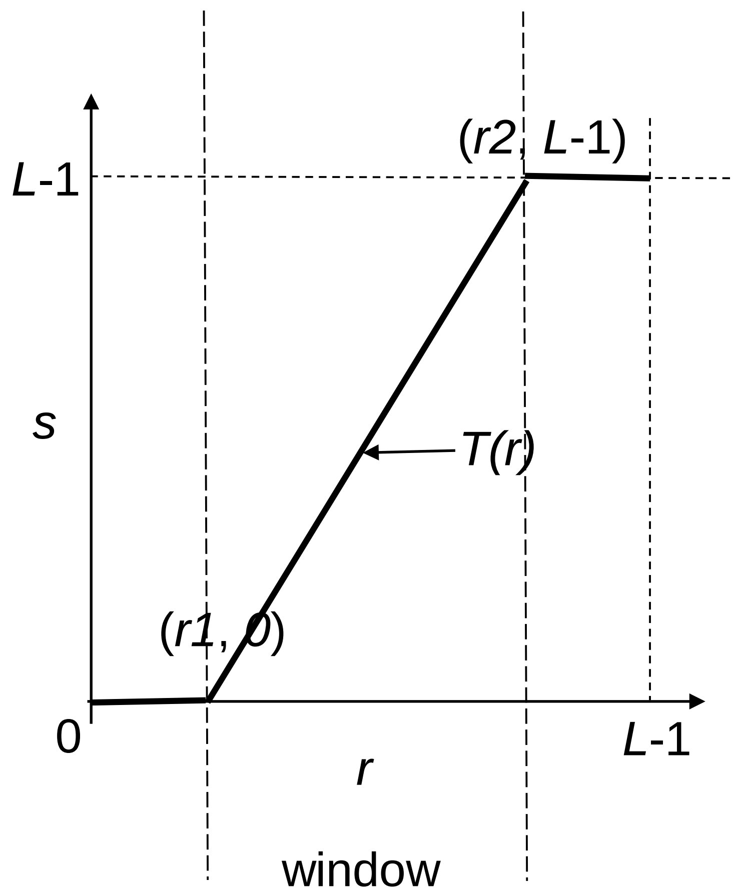


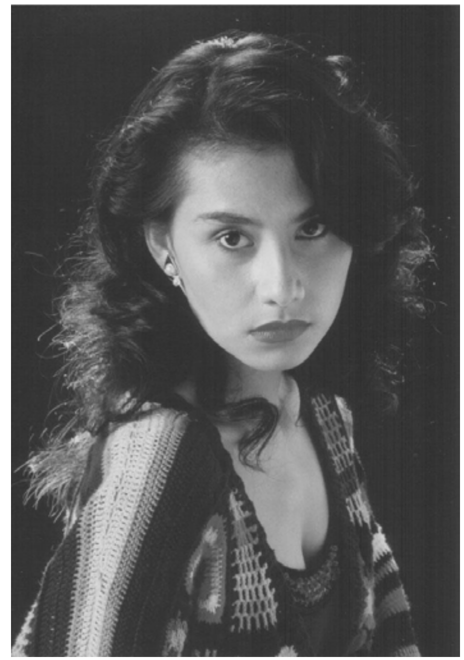


Contrast Stretching

- Contrast stretching is used to change the contrast or brightness of an image.
- In contrast stretching,
  - pixel values below a specified value are considered as black (or, have a value 0),
  - pixel values above another specified value are considered as white, and
  - pixel values in between these two values are considered as shades of gray.
- This is a linear mapping of a subset of pixel values to the entire range of grays from black to white.
- This will produce an image of higher contrast, but some details are lost.

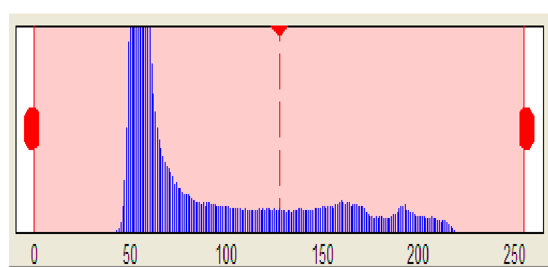
Contrast Stretching

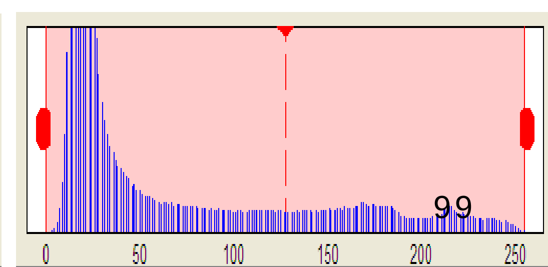




Original gray scale image

Image after contrast stretching





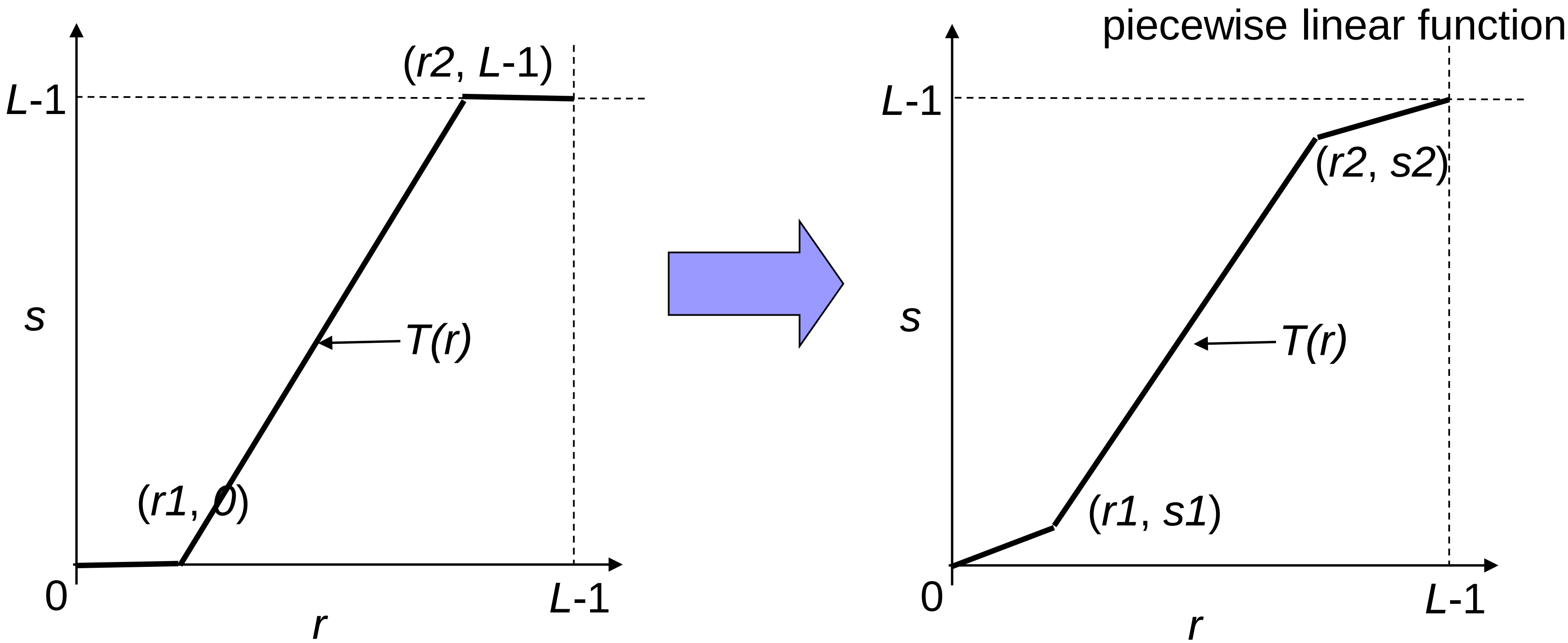
Contrast Stretching

- Using Matlab

```
% read the input image from disk
model = imread('model_web.jpg');

% convert the input color image to 256 gray level image
modelGray = rgb2gray(model);
colormap(gray(256));

% using image tool to adjust image contrast
imtool(modelGray);
```

Contrast Stretching

- To reduce the loss in details of image.

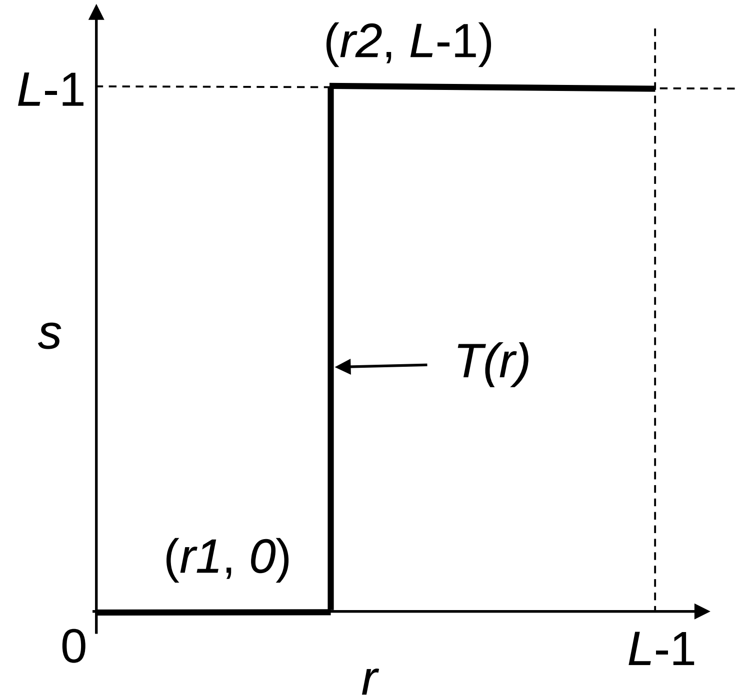


Piecewise Linear Function

- Piecewise linear function is a mapping function that is specified by a set of *n* break points (r_i, s_i), $i=1..n$, with neighboring points connected by straight lines.
- It is a generalization of typical contrast stretching.

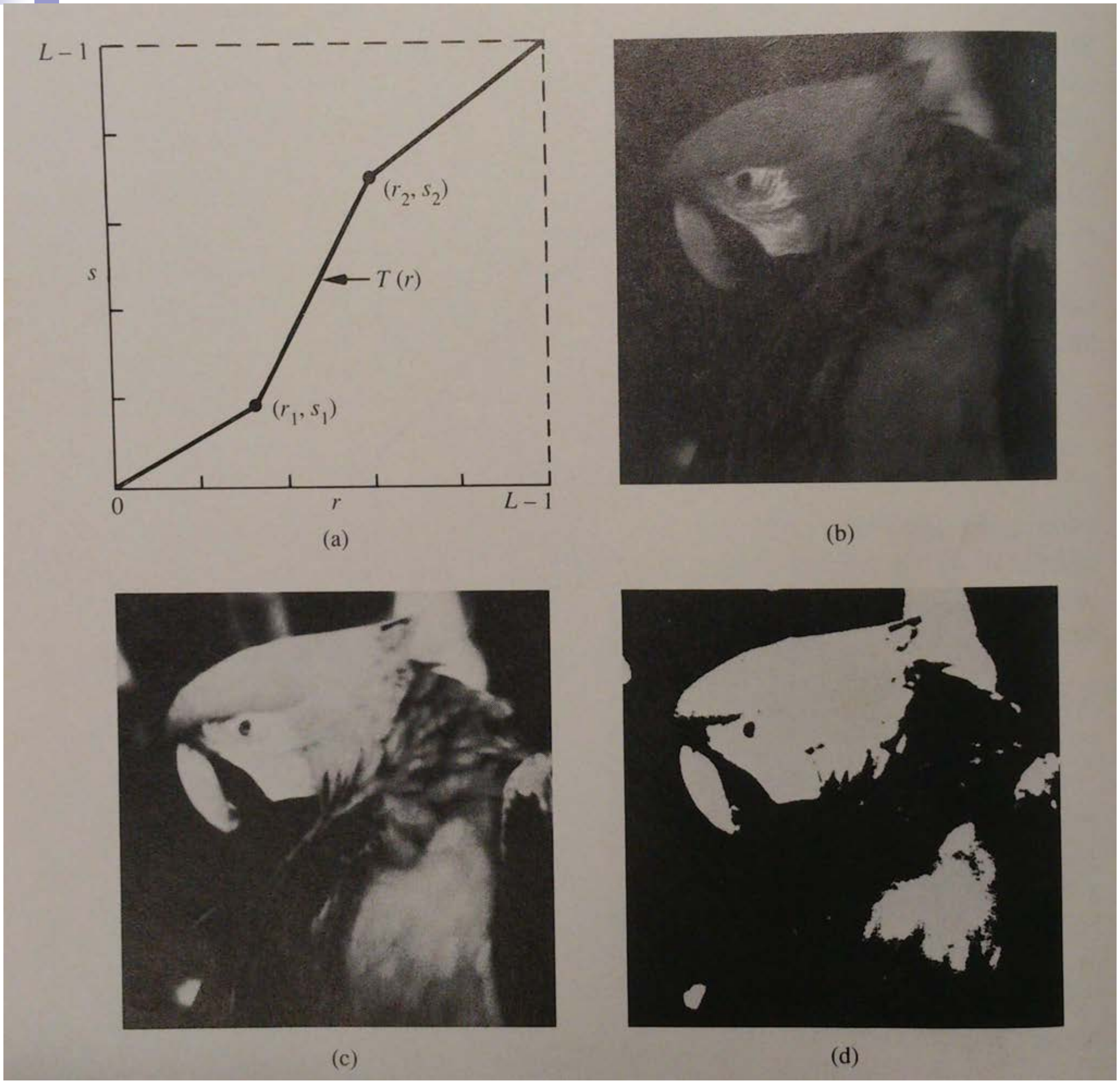
Brightness Thresholding

- It is a special case of piecewise linear function, and a simple way for image segmentation.
- A thresholding function will map all pixel values below a specified threshold to zero and all above to 255 for a 8-bit gray scale image.
- Brightness thresholding will result in a black-and-white image.



$r_1 = r_2$

$$s = T(r) = \begin{cases} 0 & r < r_1 \\ L-1 & r \geq r_2 \end{cases} \text{ or } s = T(r) = \begin{cases} 0 & r \leq r_1 \\ L-1 & r > r_2 \end{cases}$$



Contrast stretching:

(a) gray-level transformation function;

(b) a low-contrast image;

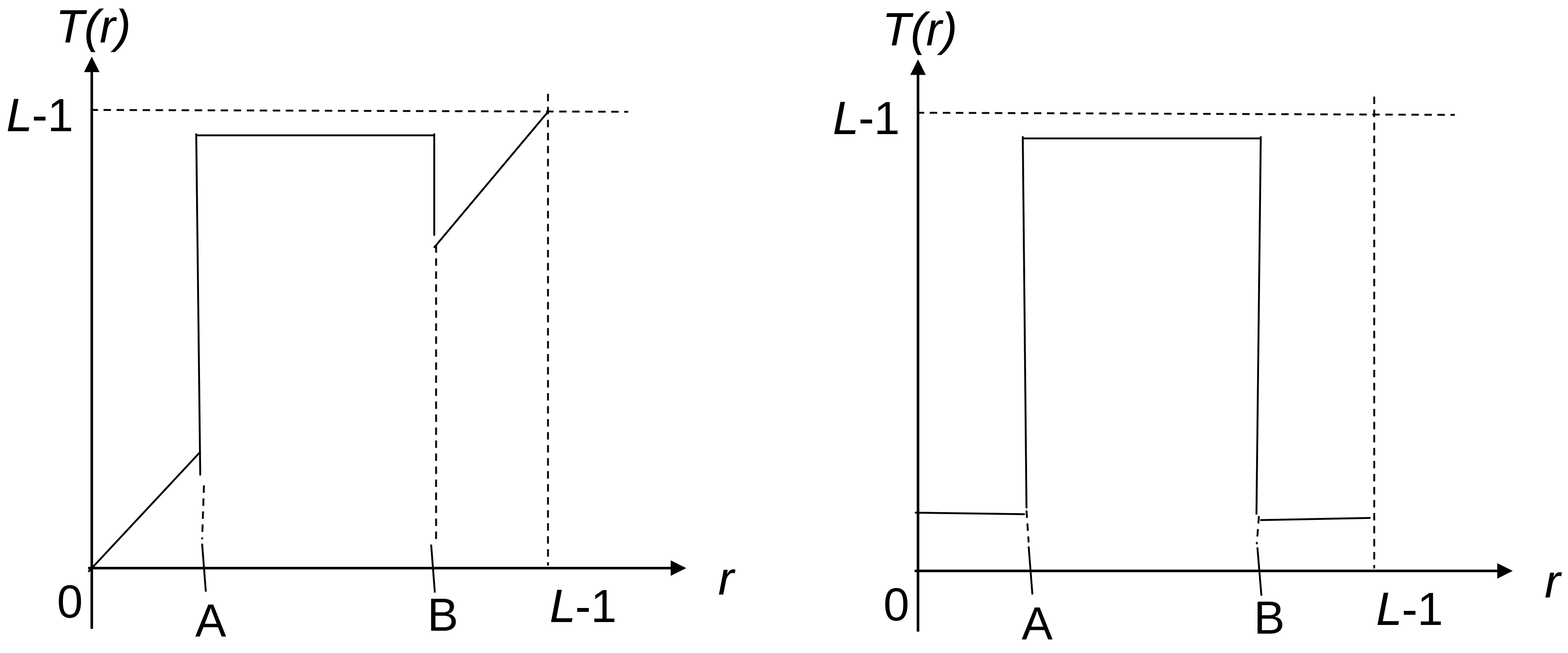
(c) result of contrast stretching;

(d) result of thresholding.

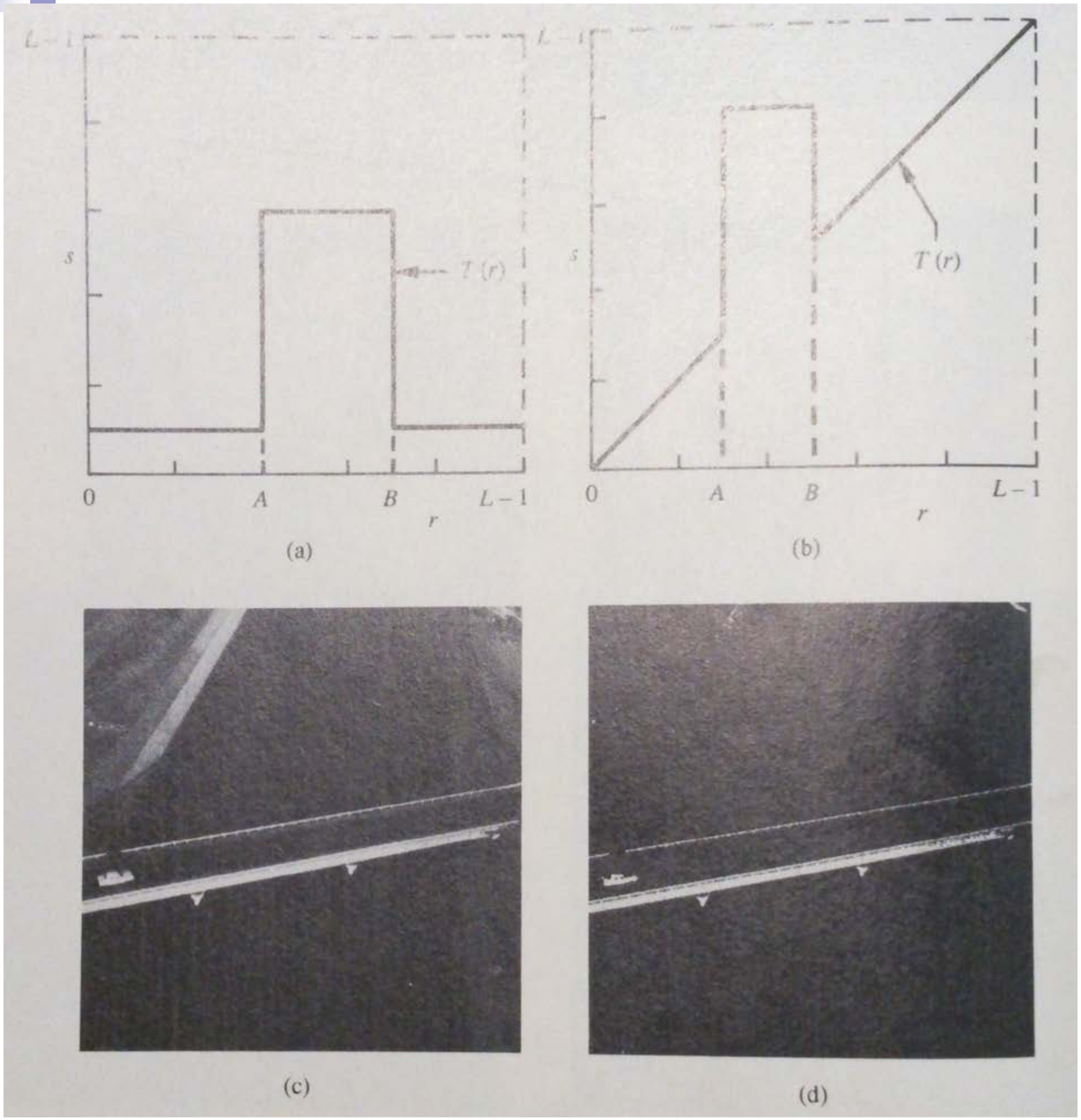
The threshold is set at $r = L/2$ with output set at L-1 (white) for any gray level in the input image of $L/2$ or higher and at 0 (black) for all other values.

Gray Level Slicing

- To highlight the gray scales between [A,B].



Note that gray scales outside [A,B] are diminished



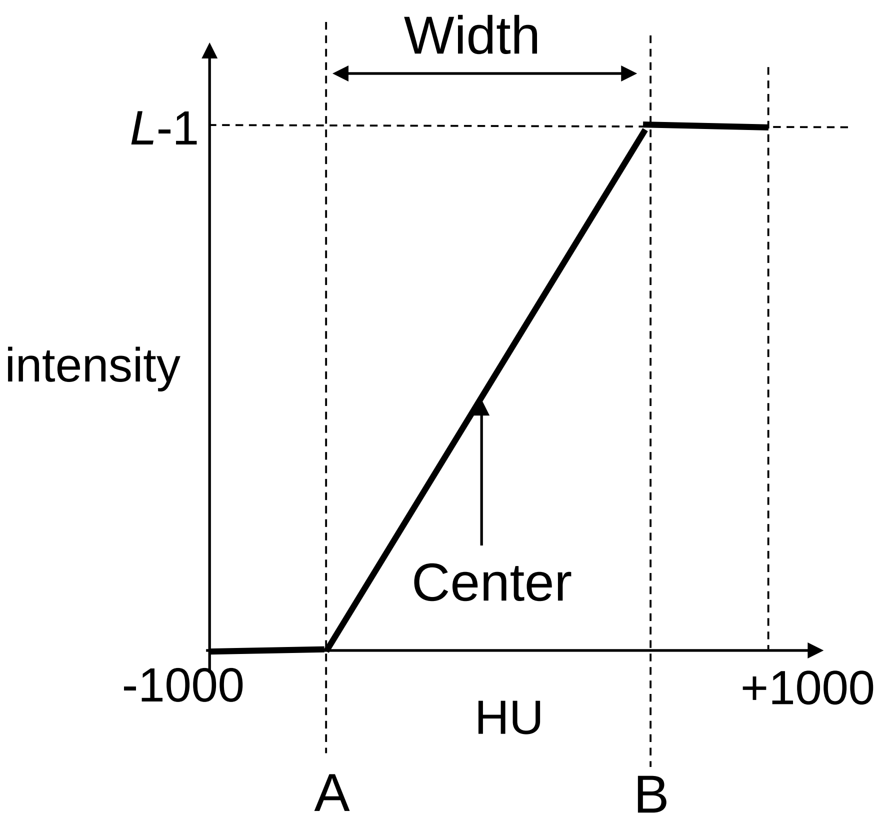
Gray level slicing (or intensity-level slicing):
(a) transformation function that highlights a range [A,B] of intensities while diminishing all others to a constant, low level; (b) transformation function that highlights a range [A,B] of intensities but preserves all others; (c) an image; (d) result of using the transformation in (b) with the intensities between A and B darkened.

Application – Computer Tomography

- Computer tomography (CT) scans are widely used in modern healthcare to produce accurate digital image of a cross-section of a patient's body.
- CT image data are stored as Hounsfield Unit (HU), which is a measure of tissue density relative to the density of distilled water. (not pixel intensities in typical images)
- Standard CT scan range from -1000 (black) to +1000 HU (white).
- "Windowing" is the process of transforming the HU into regular pixel intensity values for proper viewing.

Application – Computer Tomography

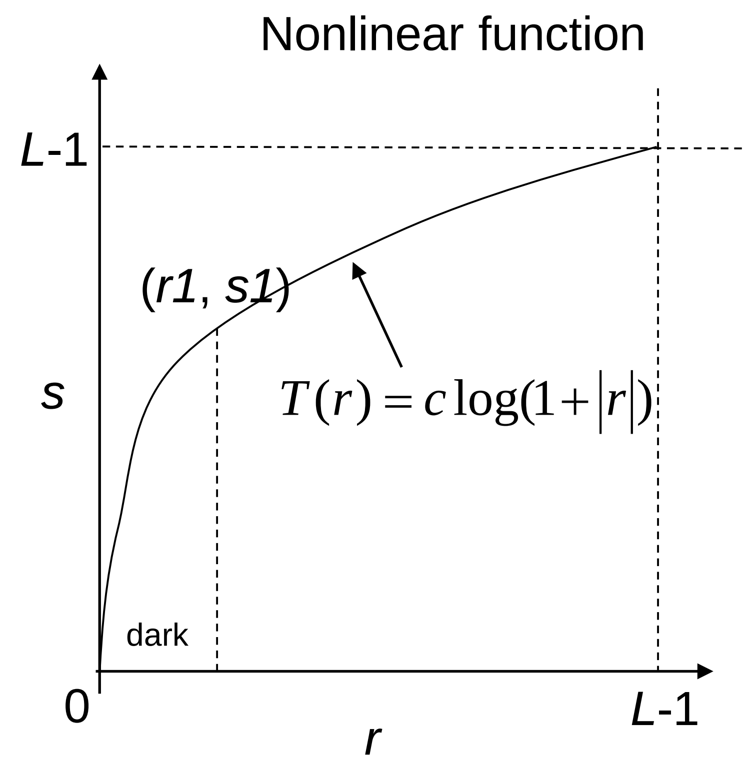
- Windowing process

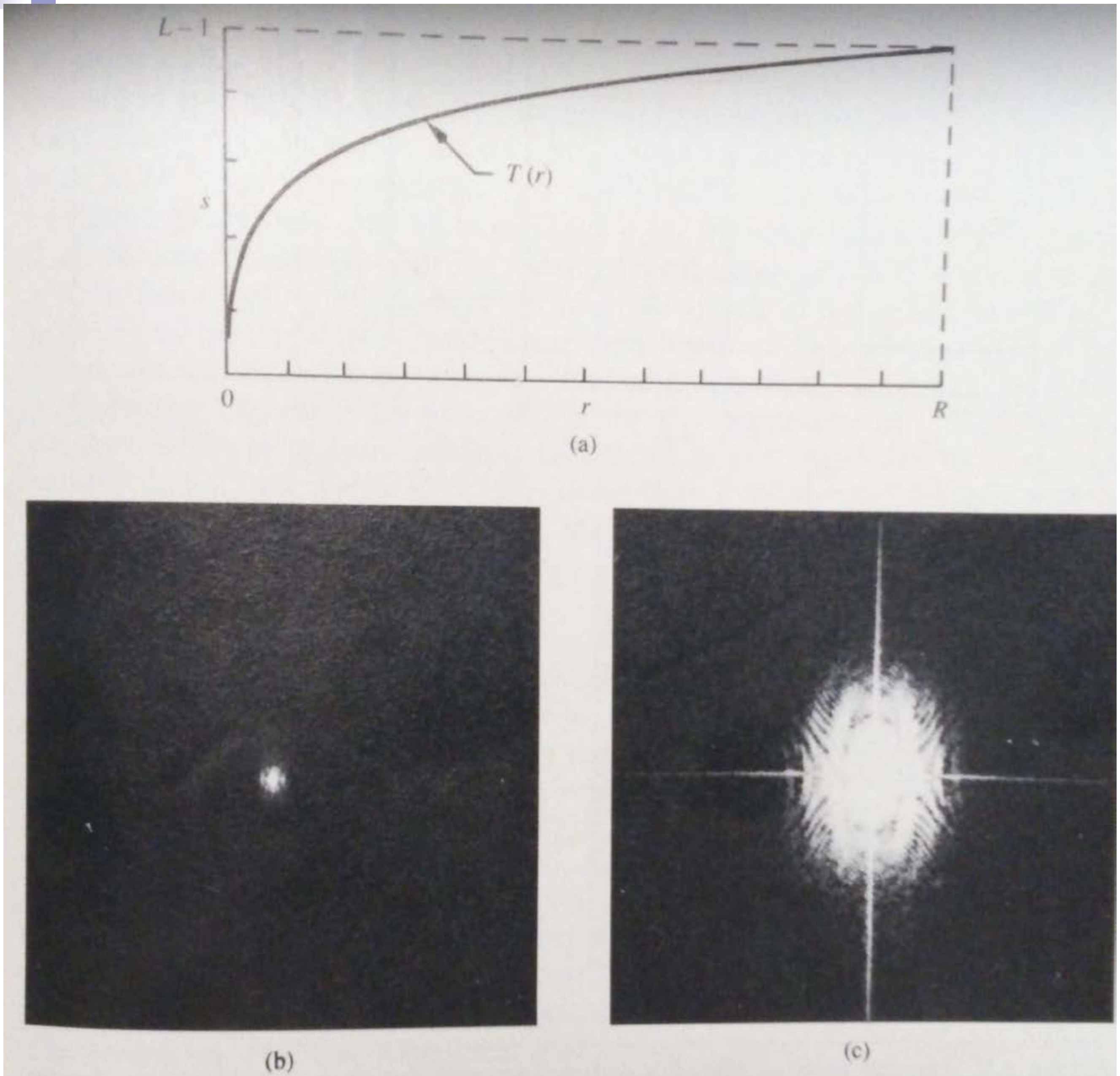


There have been much research into determining an accurate “Window” (or values of A and B) for robust segmentation of tissue types. Artificial intelligent (AI) methods such as Fuzzy Logics have been used.
Similar AI methods have also been used in determining the threshold value for brightness thresholding method.

Dynamic Range Compression

- Application of nonlinear functions
  - Compression of dynamic range
    - The dynamic range of a processed image may exceed the capability of display device – only the brightness parts of image are visible.
    - *c*: scaling constant; logarithm function performs desired compression.





Compression of dynamic range:

(a) logarithm gray-level transformation function;

(b) image with large dynamic range (pixel values ranging from 0 to 2.5×10^6;

(c) result after transformation.

6.2 Histogram Equalization

- Histogram equalization is a widely used image technique in image enhancement for increasing contrast.
- It is a method of contrast adjustment using the image’s histogram.
- It involves finding a gray level transformation function that creates an output image with a uniform or nearly uniform histogram.
- The transformation replaces each intensity in the input image by a new one.
- Transformation matrix is applied to the whole image at once.

Histogram Equalization

- The aim is to create an image with equally distributed brightness levels over the whole brightness scale
- An ideal equalized image has an equal number of pixels at all brightness levels, resulting in a straight horizontal line on the histogram graph.

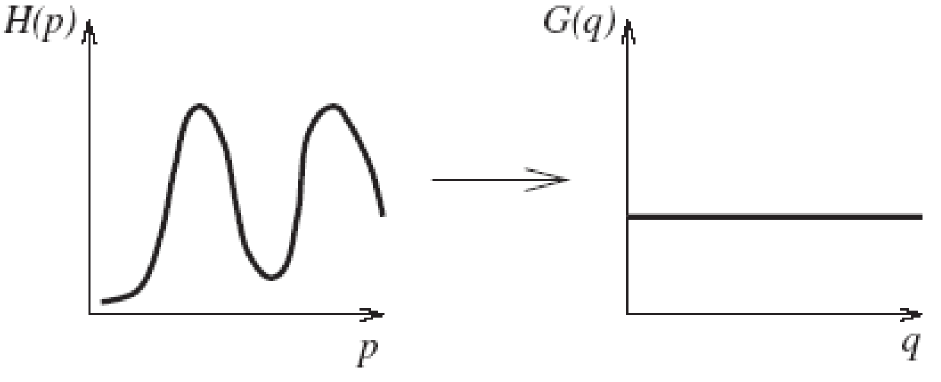


Figure 5.2: Histogram equalization.

Histogram Equalization

- $H(p)$ = input histogram, input gray-scale range = $[p_0, p_k]$
- To find a monotonic transformation $q=T(p)$ such that $G(q)$ is uniform over the output range of $[q_0, q_k]$.
- The histograms can be treated as a discrete probability density function. (Equation 1)
- Suppose that the image has N rows and columns, the equalized histogram $G(q)$ corresponds to the uniform probability density function f. (Equation 2)
- The equalized histogram can be obtained for the continuous probability density function (Equation 3).
- The desired pixel brightness transformation T is the cumulative histogram (Equation 4).
- The discrete approximation to the continuous $T(p)$ is given in Equation 5.

$$\sum_{i=0}^{k} G(q_i) = \sum_{i=0}^{k} H(p_i) \qquad (1)$$

$$f = \frac{N^2}{(q_k - q_0)} \qquad (2)$$

$$N^2 \int_{q_0}^{q} \frac{1}{q_k - q_0} ds = \frac{N^2 (q - q_0)}{q_k - q_0} = \int_{p_0}^{p} H(s) ds \qquad (3)$$

$$q = T(p) = \frac{q_k - q_0}{N^2} \int_{p_0}^{p} H(s) ds + q_0 \qquad (4)$$

$$q = T(p) = \frac{q_k - q_0}{N^2} \sum_{i=p_0}^{p} H(i) + q_0 \qquad (5)$$

Algorithm 5.1: Histogram equalization

1. For an $N \times M$ image of G gray-levels (often 256), create an array H of length G initialized with 0 values.
2. Form the image histogram: Scan every pixel and increment the relevant member of H—if pixel p has intensity g_p, perform

$$H[g_p] = H[g_p] + 1 .$$

3. Form the cumulative image histogram H_c:

$$H_c[0] = H[0] ,$$
$$H_c[p] = H_c[p-1] + H[p] , \quad p = 1, 2, \ldots, G-1 .$$

4. Set

$$T[p] = \text{round}\left(\frac{G-1}{NM} H_c[p]\right) .$$

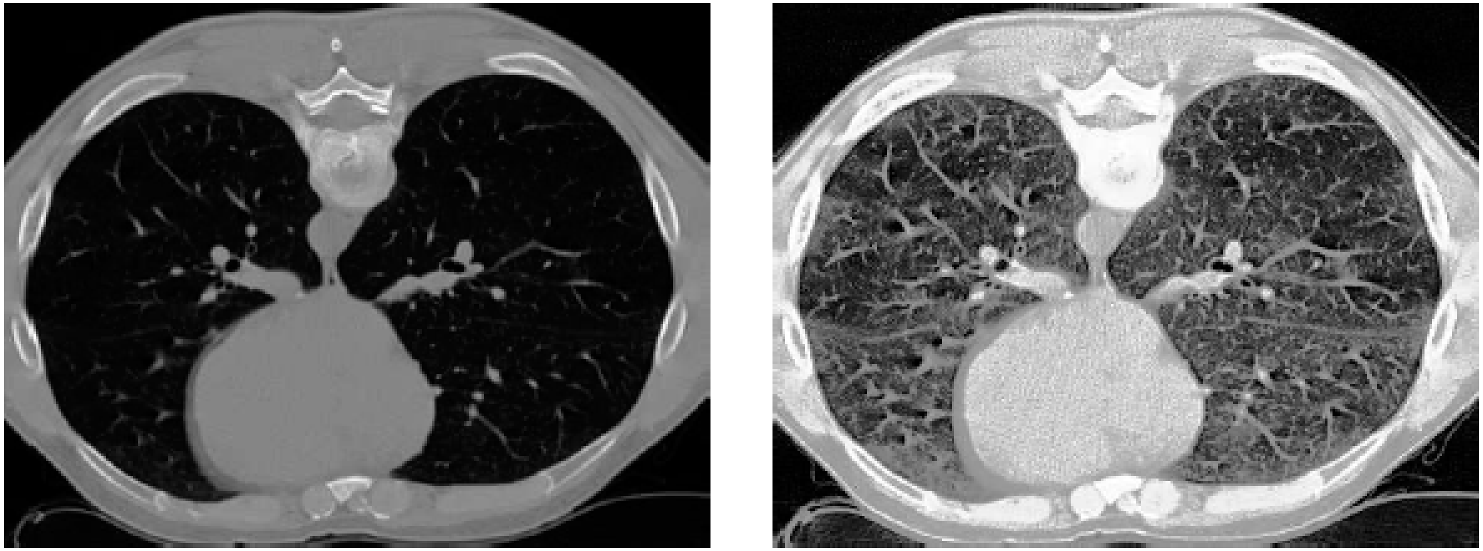
(This step obviously lends itself to more efficient implementation by constructing a look-up table of the multiples of $(G-1)/NM$, and making comparisons with the values in H_c, which are monotonically increasing.)

5. Rescan the image and write an output image with gray-levels g_q, setting

$$g_q = T[g_p] .$$

Histogram Equalization

- Example



(a) (b)

Figure 5.3: Histogram equalization. (a) Original image. (b) Equalized image.

Histogram Equalization

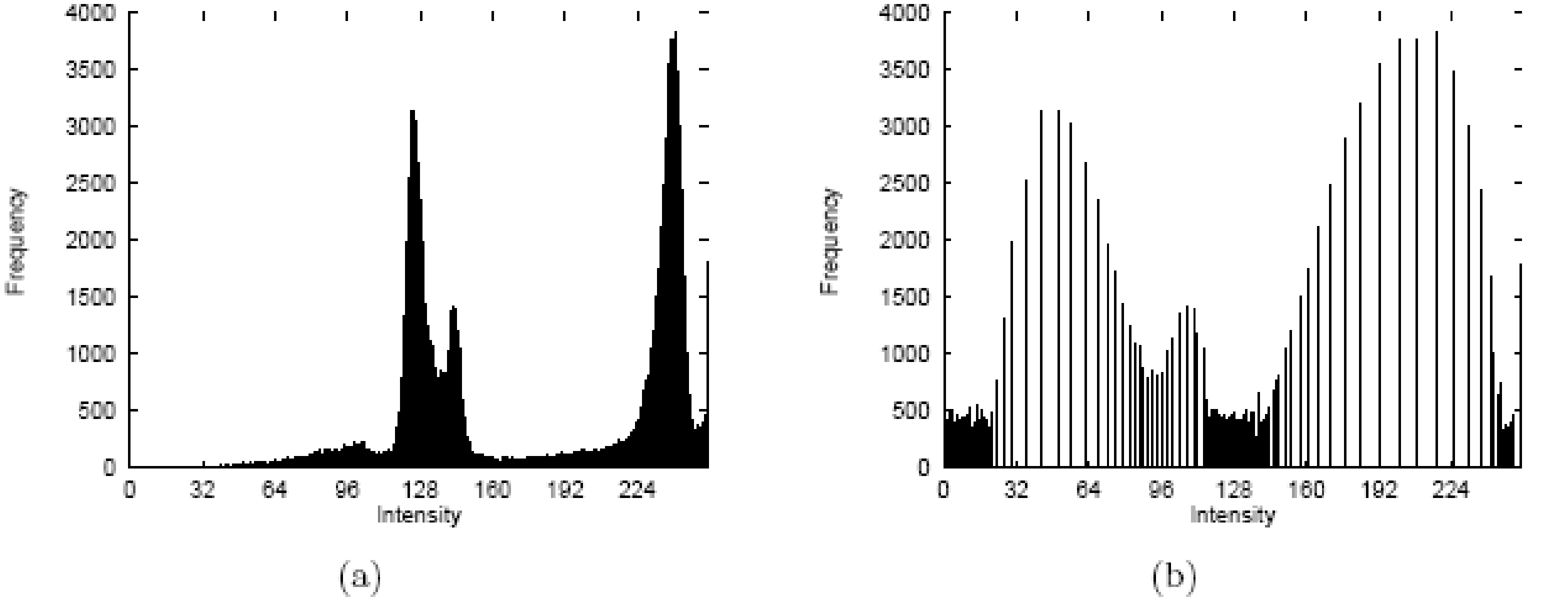


Figure 5.4: Histogram equalization: Original and equalized histograms corresponding to Figure 5.3a,b.

Matlab Implementation of Histogram Equalization

function [im_out, H, Hc, T] = hist_equal(im)

input:

im [m x n] input image

output:

im_out [m x n] equalized image

H [1x256] histogram of the input image

Hc [1x256] cumulative histogram of the input function

T [1x256] transformation function of the intensity

Matlab Implementation of Histogram Equalization

```
% note that matlab starts indexes at 1
imp = uint32(im)+1;

% allocate memory
H = zeros(1, levels);
% scan all pixels
for i=1:size(im,1)
    for j=1:size(im,2)
        %pixel intensity indexes the accumulator
        H(imp(i,j))=H(imp(i,j))+1;
    end
end
```

Matlab Implementation of Histogram Equalization

```
% form the cumulative image histogram Hc
Hc = zeros(size(H));
Hc(1) = H(1);
for i=2:size(Hc,2)
   Hc(i)=Hc(i-1)+H(i);
end
```

Matlab Implementation of Histogram Equalization

```
% create the look-up table normalizing
% the cumulative histogram to have integer
% values between 0-(levels-1)
T = round((levels-1)/(size(im,1)*size(im,2))*Hc)
% apply the look-up table to each level in
% the input image and write a new image
im_out = zeros(size(im));
im_out = T(imp);
```

Histogram Equalization

- Useful in images with both background and foreground are dark or bright.
  - It can reveal good detail in over or under-exposed photographs.
  - The method will provide good view of hard tissue (bone) in x-ray image.
- Disadvantage: global application leads to indiscriminate modification of image
  - The method may increase contrast of background noise and decrease usable signal.
  - It could not support the need to highligh certain gray level range.

Histogram Specification

- Histogram Specification is a generalized version of histogram equalization
- A “target” histogram that actually define the desired shape of the image histogram is specified.
- A nonlinear stretch operation is applied to force the image histogram to have that shape.

Summary

- Image most frequently represented as multidimensional array can be added, subtracted and made equal to another image.
- Logical operation can also be perform on image.
- Gray-scale transformations change brightness without regard to position in the image. The common gray-scale transformations are piecewise linear function, negative transformation and brightness thresholding.
- The goal of histogram equalization is to create an image with equally distributed brightness levels over the whole brightness scale.

ME5405 Machine Vision

LIM Kah Bin and CHUI Chee Kong
Control & Mechatronics Group
Mechanical Engineering, NUS

Course Mechanics

- Lecture: Wednesday (6:10 pm - 8:45 pm): LT 1
- Class info, lecture slides, notes and tutorials can be found in IVLE.
- IVLE Discussion Forum
  - Collaborative tool – comments and questions for discussion will be posted
  - Example Matlab codes for image processing
  - Peer-to-peer learning

Main Text

Digital Image Processing: International Version, 3/e

Author : Gonzalez

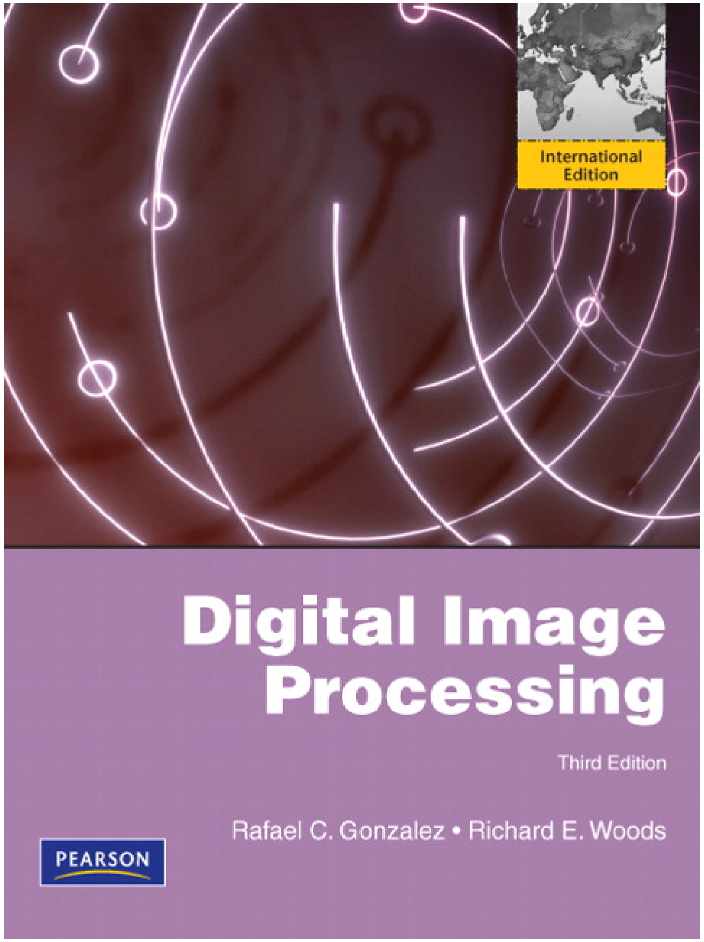
Woods

Publisher : Pearson

ISBN : 9780132345637

Available at NUS Co-op @ Forum !

http://www.imageprocessingplace.com/



Main Text (Continue)

- M Sonka, V Hlavac, R Boyle, “Image Processing, Analysis and Machine Vision”, Thomson Learning, 2008, ISBN 10:0-495-08252-X, ISBN 13:978-0-495-08252-1
  - Image Processing, Analysis, and Machine Vision – A MATLAB Companion (http://visionbook.felk.cvut.cz/)

- Other references will be announced in class or via IVLE Discussion Forum.

Chapter 1 - Introduction

LIM Kah Bin, Dr.-Ing.
Control & Mechatronics Group
Mechanical Engineering, NUS

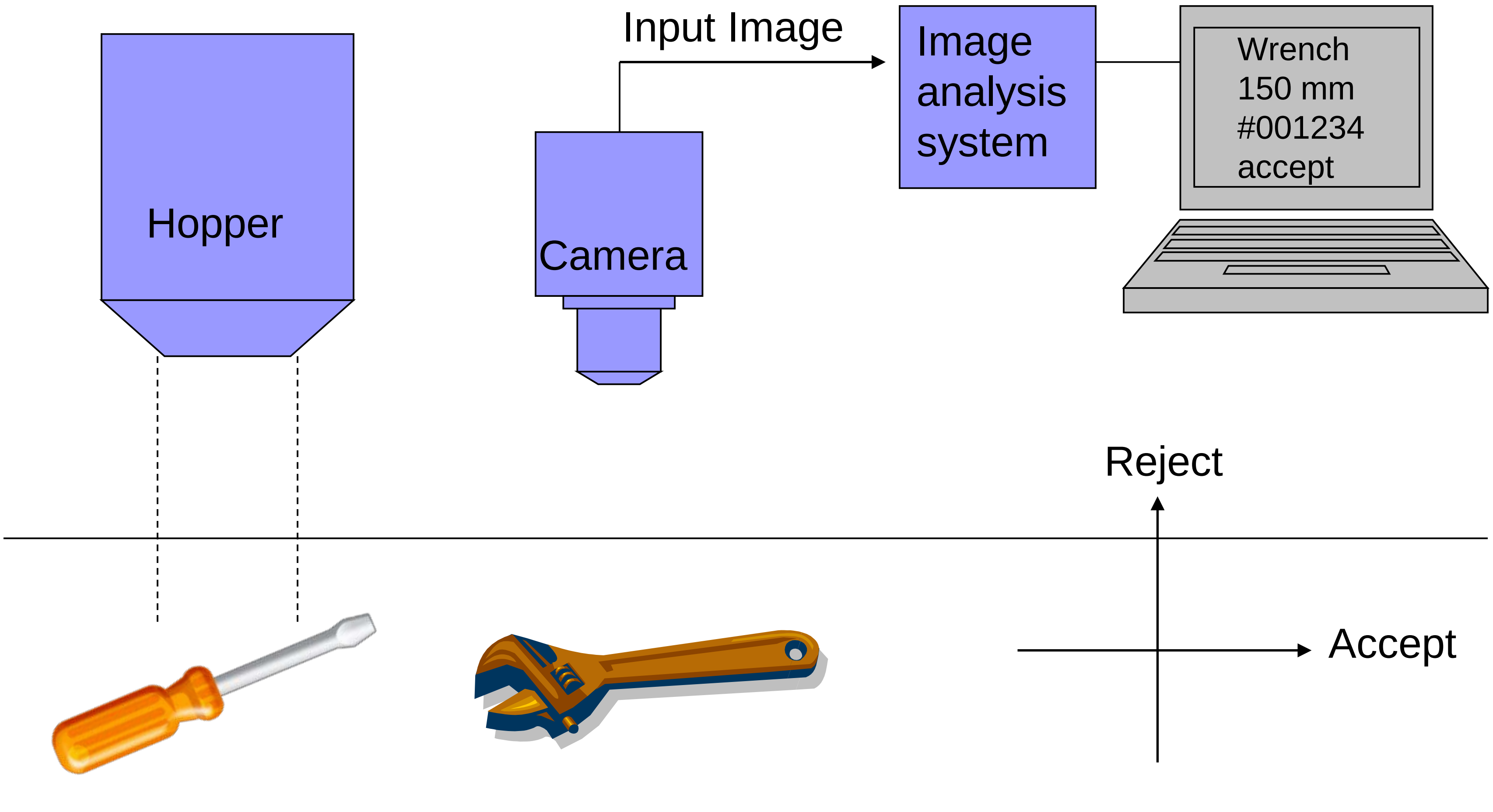
Human Vision, Computer Vision and Machine Vision

- Human Vision: allows humans to perceive and understand the world around him
- Computer Vision: aims to duplicate the effect of human vision by electronically perceiving and understanding an image
- Machine Vision: is the application of computer vision to industries including media and healthcare, and manufacturing.
  - A subfield of engineering that encompasses mechanical engineering, optics, computer science and automation.

Examples of Machine Vision

- Industrial vision – inspection of manufactured goods such as semiconductor chips, automobile components, tools, food and pharmaceuticals
  - Uses digital cameras, smart cameras and image processing software
  - Smart camera is an integrated machine vision system that comprises of image capture circuitry and a microprocessor to extract and process information from images. It normally has an interface device to make results available to other devices.

Examples of Machine Vision (Continue)

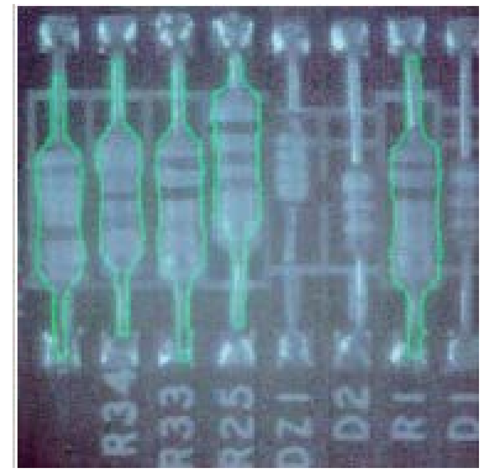


Examples of Machine Vision (Continue)

- X-ray for health screening
- Satellite maps for battle planning
- Photoshop to prepare photographs
- Face detection
- Capture an image from moon, and then transmit the image to Earth

Machine Vision for Quality Control

- Machine Vision allows manufacturing industry to
  - Detect defects
  - Calibrate and control the manufacturing process
  - Optimize the use of resources
- Leading to
  - Result repeatability
  - Product reliability
  - 100% high speed inspection
  - Consumer confidence and satisfaction



Source: British Machine Vision Association and Society for Pattern Recognition

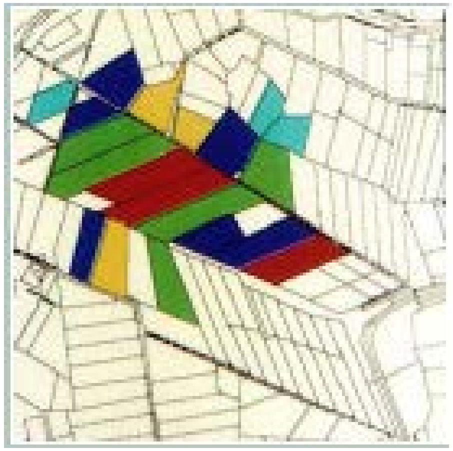
Machine Vision for Security and Surveillance

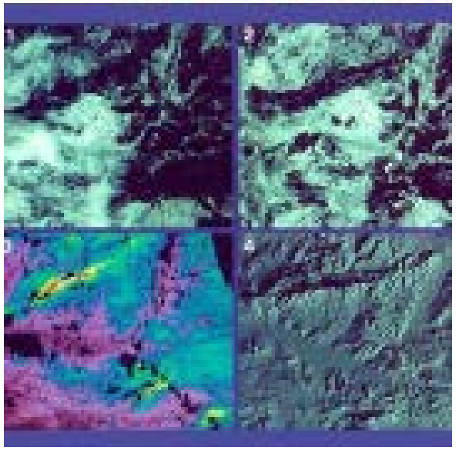
- Machine vision provides the abilities to
  - Track objects and people in 3D
  - Recognize and register specific and generic objects
  - Model and identify gestures, actions and behavior
  - Perform biometric measurements
- Leading to
  - Safer environment
  - Efficient non-obstructive monitoring
  - Reduce crime rates

Source: British Machine Vision Association and Society for Pattern Recognition

Machine Vision for Environment

- Machine vision can be used to
  - Monitor pollution from refuse sites
  - Map and monitor the condition of gas pipelines and railways
  - Police the pollution of the seas
  - Monitor the spread of disease in crops
- Leading to
  - Cleaner environment
  - Greater safety
  - Better planning and use of resources





Source: British Machine Vision Association and Society for Pattern Recognition

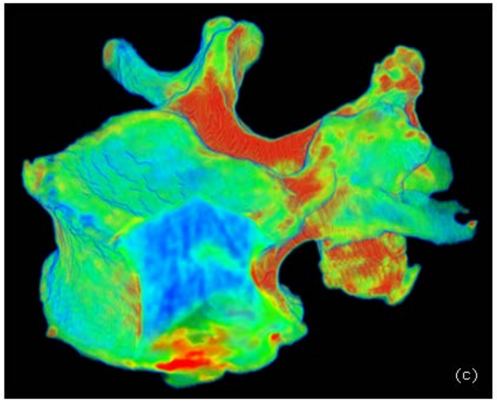
Machine Vision for Entertainment

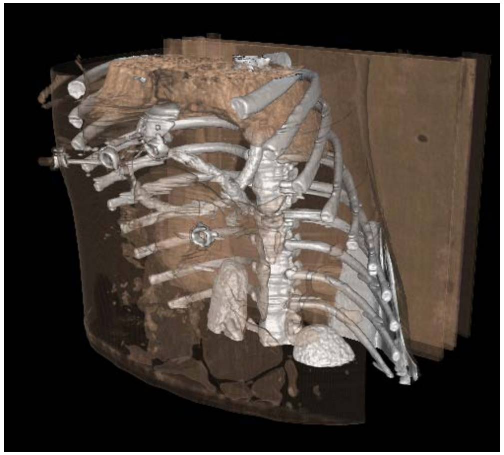
- Machine vision allows
  - Searching of image databases and video libraries by content
  - Efficient image compression
  - Multiview scenes creation
  - Realistic models of objects generation
- Leading to
  - Greater realism
  - Lower cost
  - Wider accessibility

Source: British Machine Vision Association and Society for Pattern Recognition

Machine Vision for Medicine

- Machine vision empowers the clinician with
  - 3D/4D visualization
  - 3D texture analysis
  - 3D and 2D image registration
  - Virtual object manipulation
- Leading to
  - Objectivity in measurement and estimation
  - Result repeatability
  - Decision consistency





Why is Computer/Machine Vision Difficult?

- Loss of information in 3D to 2D
  - Pinhole model and single available view - the projective transformation sees a small object close to the camera in the same way as a big object remote from the camera.

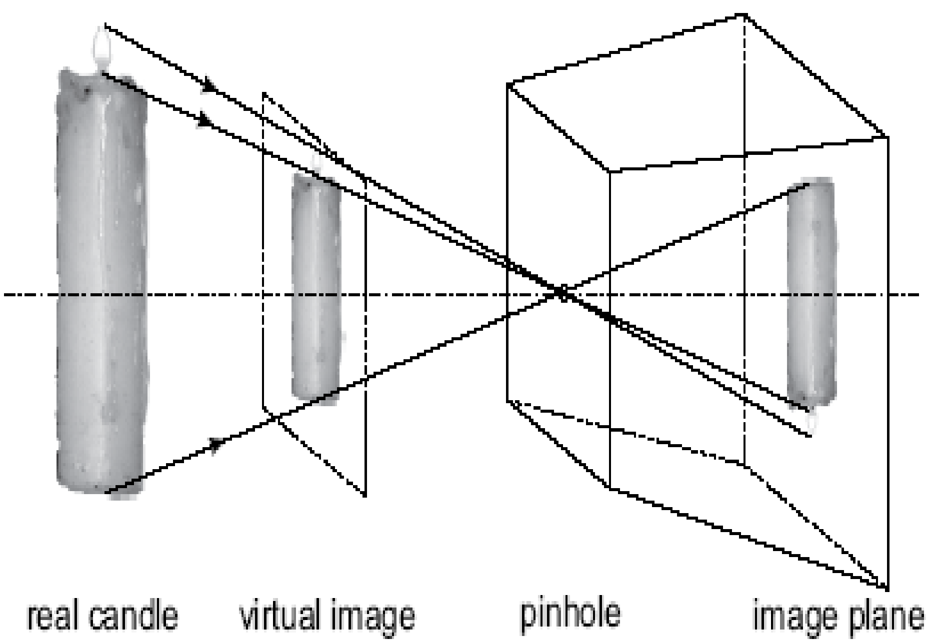


Figure 1.4: The pinhole model of imaging geometry does not distinguish size of objects.

Why is Computer/Machine Vision Difficult? (Continue)

- Interpretation
  - A human uses previous knowledge and experience to understand an image currently under observation.
- Noise
  - Inherently present in every measurement in the real world.

Why is Computer/Machine Vision Difficult? (Continue)

- Too much data
  - An A4 sheet of paper scanned at 300 dots per inch (dpi) at 8 bits per pixel (grayscale) = 8.5 MB
- Brightness measured
  - The brightness or radiance is dependent on the irradiance (light source type, position and intensity), the observer's position, the surface local geometry, and the surface reflectance properties.

Scope of ME5405

- Introductory graduate-level vision and image processing class
- Emphasis on general principles, systems and applications

Contents of ME5405

- Introduction (LKB)
- Digital Image Fundamentals and Programming (CCK)
- Image Acquisition (CCK)
- Binary Machine Vision (LKB)
- Image Enhancement (LKB)
- Image Segmentation (LKB)

Contents of ME5405 (Continue)

- Color Image Processing (LKB)
- Image Geometry (CCK)
- Noise and Filtering in Frequency Domain (CCK)
- Machine Vision Application and Design (CCK)

Assignment

Computing Project

- 30% of the final grade
- Computing project – group of 2 or 3
- Codes written in Matlab or clones (http://www.dspguru.com/dsp/links/matlab-clones)

Computing Project (Continue)

- **Image: Characters**
  - You are given two set of images of characters.
  - For each image,
    - Threshold the image and convert it into binary image
    - Display the original image on screen
    - Determine the outline(s)
    - Segment the image to separate and highlight the different characters
    - Rotate the characters about their own respective centroids by 90 degrees clockwise and then 30 degrees counterclockwise
    - Determine a one-pixel thin image of the characters
    - Scale and display the characters in each image according to specified patterns

Computing Project (Continue)

- Report
  - An introduction to the problem
  - A description of your algorithm and a flow chart
  - Screen dumps of every stage of the image processing
  - An explanation of the method and why you choose the method employed in your project
  - A conclusion including on how your codes can be improved in the future
- Softcopy of your program and data file (with readme.txt on how to execute the codes)
  - Upload onto IVLE – Workbin - Student Submission
- Due: about one week before ME5405 final exam

ME 5405 Machine Vision

Assignment

Computing Project

You are required to form a group of 2-3 students to work on the computing project. The software must be developed using MATLAB or its open source alternatives such as Octave, Scilab or FreeMat. Your report should include the followings:

1. an introduction to the problem,
2. a description of your algorithm and flow chart,
3. screen dumps of every stage of the image processing,
4. an explanation on why you choose the method employed in your project, and
5. a conclusion including comments on how processing the two images are similar and/or different.

Images 1 is a 64x64, 32 level images. The image is shown a coded array that contains an alphanumeric character for each pixel in the image. The range of these characters is 0-9 and A-V, which corresponds to 32 gray levels. Image 2 is a BMP image of a label on a microchip.

Image 1: Available on IVLE-ME5405-Files-Lecture Notes – charact1.txt
Image 2: Available on IVLE-ME5405-Files-Lecture Notes – charact2.bmp

For each image,

1. Display the original image on screen.
2. Threshold the image and convert it into a binary image.
3. Determine the outline(s).
4. Segment the image to separate and identify the different characters.
5. Rotation of the characters about their own respective centroids by 90 degrees clockwise.
6. Rotation of the characters about their own respective centroids by 30 degrees counterclockwise.
7. Determine a one-pixel thin image of the characters from Step 4.
8. Scale and display the characters of Image 1 in one line with the sequence: **A1B2C3**.
9. Scale and display the characters of Image 2 in one line with the sequence: **7M2HD44780A00**.

You should upload your report and software to IVLE-ME5405-Files-Student Submission by 21 November 2017 (Tuesday) which is about one week before ME5405 final examination.

Image 1

```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000004200000000
000000000BJH0000000000000000MURRRSLS000000000000000COJJKLL000000
000000004LLK5000000000000000KNNOPONKO0000000000000HJNHDEJKJ00000
00000000CMMLJ000000000000000MOD4579MMF00000000000BKL40004FJC0000
00000002LMILM400000000000000LMS0000EMJ00000000000JM7000003C90000
00000005MJ4OMO00000000000001MO40000CMF00000000005KK0000000000000
0000000BMB07NM00000000000002LNA00003NM3000000000DKH0000000000000
0000000JL301KN2000000000001MOMPROLM300000000000CL00000000000000
0000002JJ000FMB0000000000002NOMHHIMNP00000000000BK00000000000000
0000004MJ000OMO0000000000002MO40000CNM0000000000SLK0000000000000
000000DNOUUUMML0000000000003ON300000MN10000000001KM4000000LN0000
000001KLHOFFFMMD000000000003LO500001MN20000000000FLK00000BJH0000
000007LI00000FML000000000000MO70000MOL000000000002LLM500JMK50000
00000DMD000004JK000000000000NPPTVUONPA0000000000003KNJJJKMA00000
00000CE3000000A8000000000000JNNNLNMLO400000000000009OIOB00000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000004MJ000000000000000002STRRD000000000000000000NNLM9000000
000000004LMH000000000000000005OMPPOM0000000000000008JKHOJJD00000
000000BONMMI00000000000000000KOE00DNM80000000000000IJ6003CI10000
0000007FOKMH00000000000000000NL0000HMD0000000000004KE0000BJ50000
0000000009MI00000000000000000BA0000FME000000000000000000EJ300000
0000000008MH00000000000000000000000KNA000000000000000000FJ900000
0000000008MI0000000000000000000000JMK10000000000000000007JJH000000
0000000008MI000000000000000000000LNL40000000000000000004CIJ40000
0000000008MH00000000000000000000MOI30000000000000000000007OF0000
0000000006MH0000000000000000000LOD000000000000000005000001OH0000
0000000007MH000000000000000000JMB00000000000000000DJB00005HO0000
0000000007MH00000000000000000AMMJADFID0000000000003IK4003OID0000
00000000061D00000000000000000JPOOOOONK0000000000000AKJJIIJF10000
00000000001000000000000000000BAAAABA7400000000000000005EHOE7000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

Image 2