MODULE 1: INTRODUCTION TO VIDEO ANALYTICS AND COMPUTER VISION

Dr. Matthew CHUA, PhD

Institute of System Science, NUS

About Me…

- DR. Matthew CHUA
- Faculty, Institute of Systems Science & Joint Faculty, Saw Swee Hock School of Public Health, NUS
- Research Interests: Medical Systems, Robotics and Artificial Intelligence.
- Heading Research in ISS with over $1 million grant
- Professional Opera Singer

Some Photos of My Life…

Course Outline

Day 1	• **Introduction to computer vision systems** • **Foundations of computer vision system I: Modelling and processing**
Day 2	• **Foundations of computer vision system II & workshop on image manipulation** • **Foundations of computer vision system III & workshop on object identification**
Day 3	• **Vision system using machine learning I & workshop on image segmentation** • **Vision system using machine learning II & workshop on object detection**
Day 4	• **Vision system using machine learning III: Feature representation for scene understanding** • **Video analytics system**
Day 5	• **Vision system architecture and solution** • **Case studies of vision system and workshop** • **Final written assessment test**

Main References

- Sonka, M., Hlavac, V., & Boyle, R. (2014). *Image processing, analysis, and machine vision*. Cengage Learning.
- Rafael C.. Gonzalez, Richard E.. Woods, & Steven L.. Eddins. (2010). *Digital Image Processing Using MATLAB®*. McGraw Hill Education.
- Smith, S. W. (1997). The scientist and engineer's guide to digital signal processing.
- Orfanidis, S. J. (1995). *Introduction to signal processing*. Prentice-Hall, Inc.
- Nagrath, I. J. (2006). *Control systems engineering*. New Age International.
- Ogata, K., & Yang, Y. (1970). Modern control engineering.
- Chui *et. al.* ME5405 Machine Vision. NUS

Contents of this lecture

1. Introduction
2. Digital Imaging Fundamentals
3. Basic Image Processing
4. Binary Machine Vision

Introduction

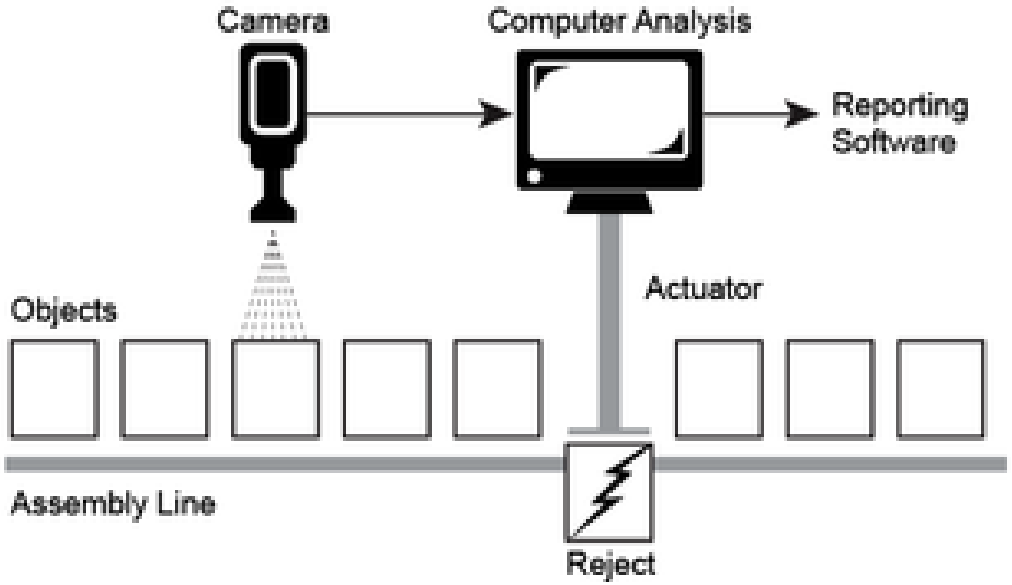
Human Vision & Computer Vision

- Human vision is the natural ability to convert incoming light on our corneal to brain signals
  - Allowing us to perceive our surrounding
- Computer vision follows human vision through conversion of light input into digital form
  - Allows for future processing and learning

http://news.mit.edu/2015/nasa-gives-mit-humanoid-robot-future-space-missions-1117

Examples of Machine Vision

- Process line vision
  - To inspect the quality of manufactured parts such as hard disk and other components
  - Integrates cameras, GPUs and image processing and learning software
  - Replaces blue collar workers in factories



https://www.designnews.com/content/streaming-video-versus-machine-vision-how-do-they-compare/182616920933123

Security and Surveillance

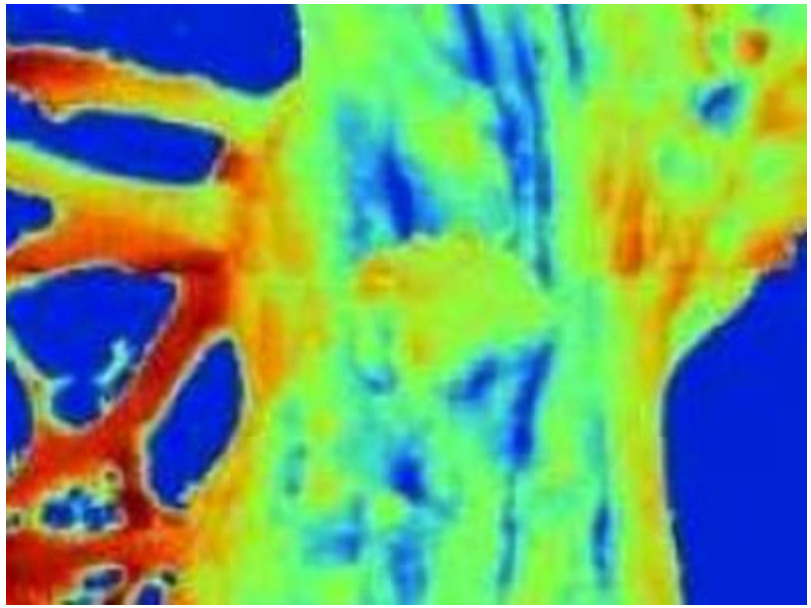
- Can be deployed in security by:
  - Tracking people in real time
  - Authentication
  - Human activity recognition
  - Biomechanical measurements
- Advantages:
  - Safety enhanced
  - Efficiency improvements

https://phys.org/news/2018-07-facial-recognition-privacy.html

Machine Vision for Environment

- Environmental aspects:
  - Monitor forest fires
  - Pipe leakages under ground
  - Trees health

- Benefits:
  - Cleaner environment
  - Increased safety
  - Less downtime



https://advanced-treecare.com/tree-services/commercial-tree-service/

Machine Vision for Entertainment

- Applications:
  - Immersive games in VR and AR
  - Realistic object generation
  - Social media

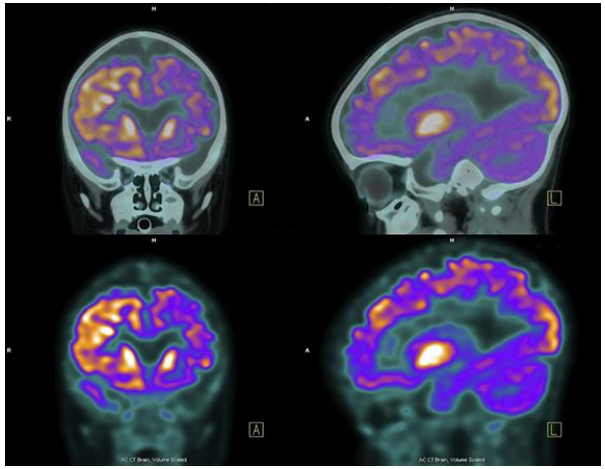
https://sandboxvr.com/

- Benefits:
  - Better connectivity
  - Realism
  - Lower costs

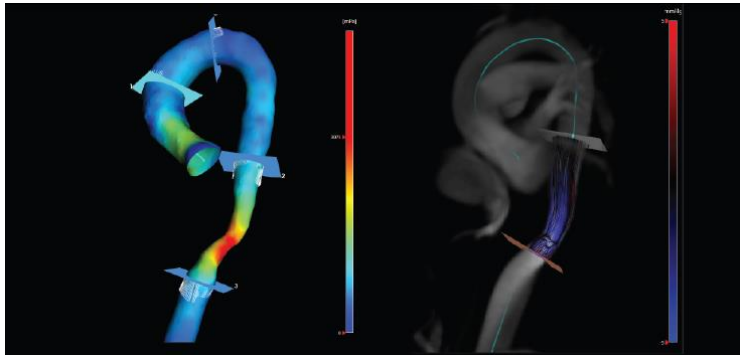
Machine Vision for Medicine

- Applications in Medicine:
  - 3D/4D visualization
  - Finite element analysis
  - Imaging for diagnosis and planning

- Benefits:
  - Better measurement and estimation
  - Accuracy
  - Repeatability
  - Additional insights



https://www.nibib.nih.gov/node/56256



https://www.piemedicalimaging.com/product/mr-solutions/caas-mr-4d-flow/

Limitations of Computer Vision

- Loss of information from 3D to 2D
  - Due to pinhole model
  - Spatial information is “lost” when transforming to an image

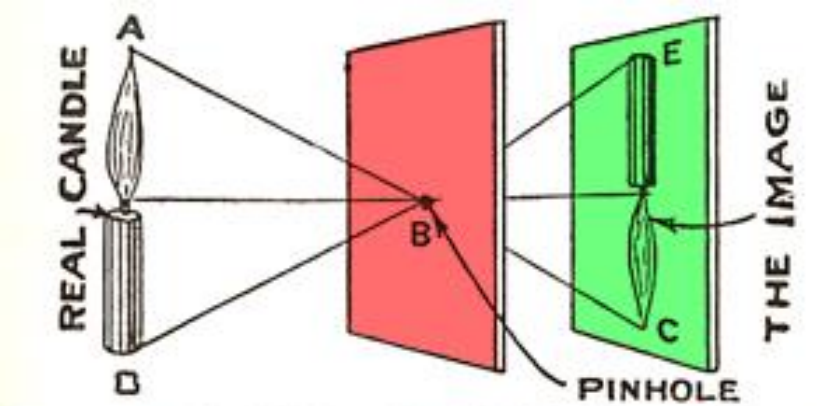
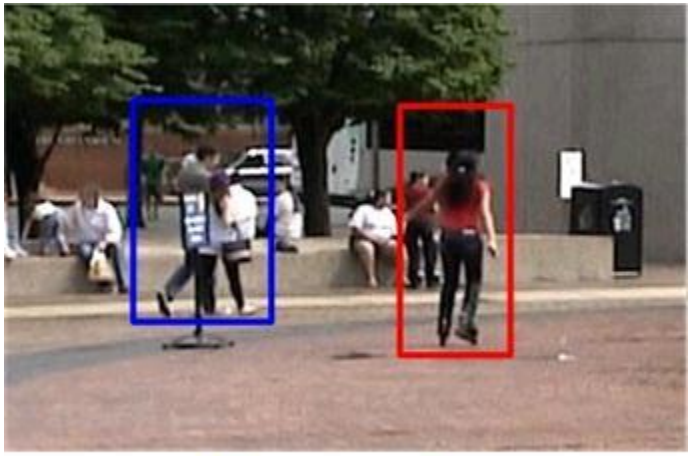


FIG. 131.—How Light and a Pinhole Form an Image.

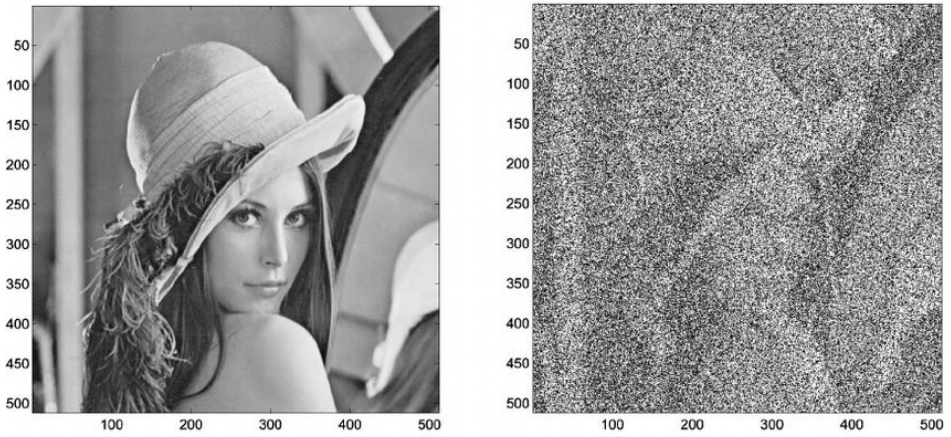
https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera

Limitations of Computer Vision

- Interpretation and understanding
  - Humans and some living things use previous knowledge and experience to interpret an image
- Noise
  - May distort the intended information of an image



http://www.bu.edu/ids/research-projects/action-recognition/



https://www.researchgate.net/publication/252225009_Combining_the_discrete_wavelet_transforms_and_rank-order_based_filters_for_image_restoration/figures?lo=1&utm_source=google&utm_medium=organic

Limitations of Computer Vision

- Data intensive
  - Large amount of space required to store images and videos
  - Impose a toll on the processing during machine learning

- Subjected to contrast issues
  - Images and videos are subjected to illumination, reflectance and ambient lighting issues

Dynamic Contrast : Off

Dynamic Contrast : High

https://www.forbes.com/sites/johnarcher/2017/08/10/lg-issues-bizarre-response-to-latest-oled-tv-gaming-problems/#4e4e58237774

Digital Image Fundamentals

Image Fundamentals

1. Image function *f(x, y)*

- ***f(x,y)*** consists of both spatial (x,y) and properties values f(x,y), g(x,y), etc..
- ***Image Sampling***: Digitization of the spatial coordinates ***(x, y)***.
- ***Gray-level quantisation***: Digitization of amplitude f(x,y).
- A continuous image ***f(x, y)*** can be represented by a *N* x *M* array shown below, where each element is a discrete quantity:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

https://www.geeksforgeeks.org/digital-image-processing-basics/

- The array represented by ***f(x,y)*** can form a digital image.

Image Fundamentals

- For Gray-scaled images
- f(x, y), x and y are integers.
- The value of f(x, y) is the gray level of the image at the coordinates x and y.
- N X M is the size of the image (eg. 512 by 512 image = 262,144 pixels)
- Gray levels is given by 2^G, where G is the number of bits in a byte which is used to represent the brightness of a pixel.
  - For example, for an 8-bit byte, the number of gray level available is $2^8 = 256$, or gray levels run from 0 to 255.

Image Fundamentals

2. Image Coordinates

The image will be described by an N X M matrix of pixel values (the element p(i, j) are positive scalars), that indicate the light intensity of the flux on the picture element at (x, y) represented by the pixel.

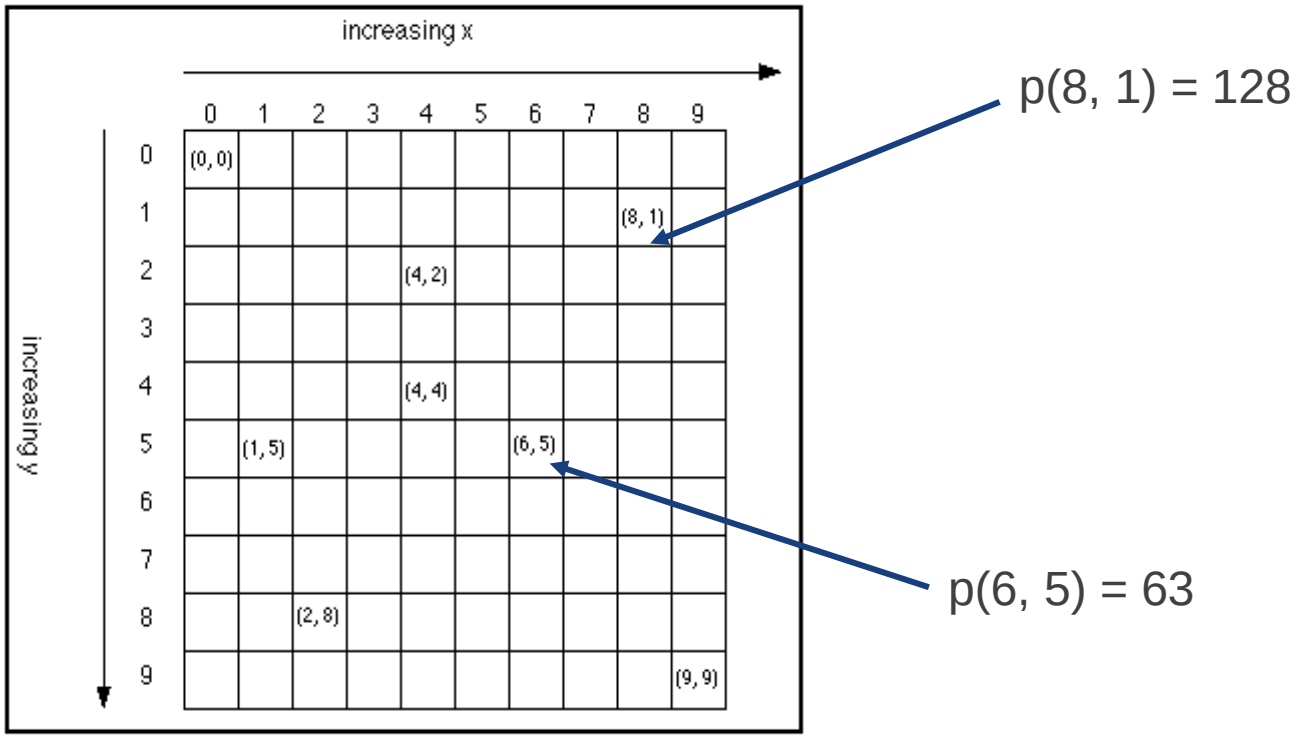
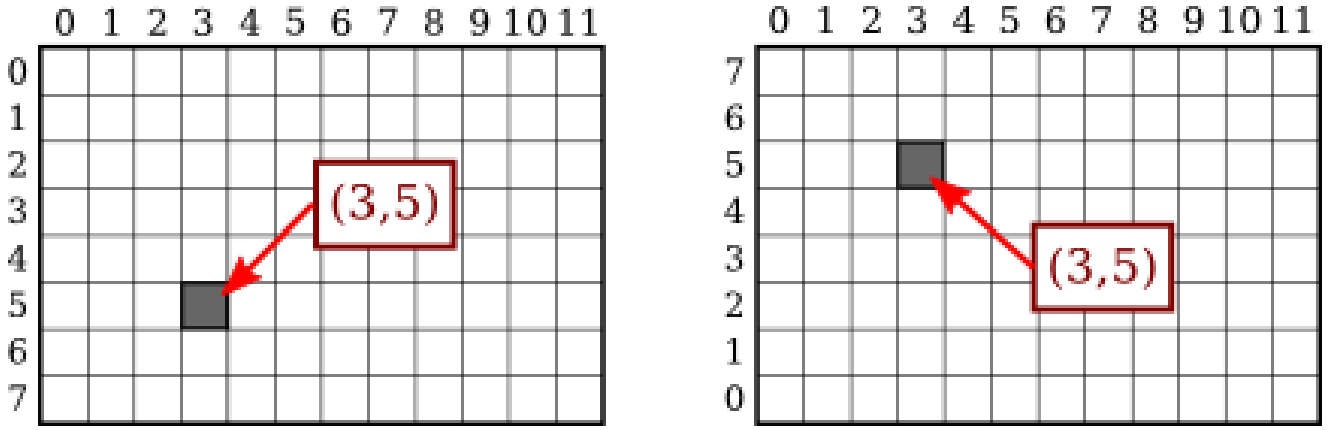


Image Fundamentals

2. Image Coordinates

The origin in the picture and matrix can be different; the x and y coordinates in the picture at the lower left corner, whereas the numbering of pixels starts at the upper left corner of the matrix.

Note that this is not a standard convention. Some people may use different conventions; eg. First location is given as (0, 0) instead of (1, 1).



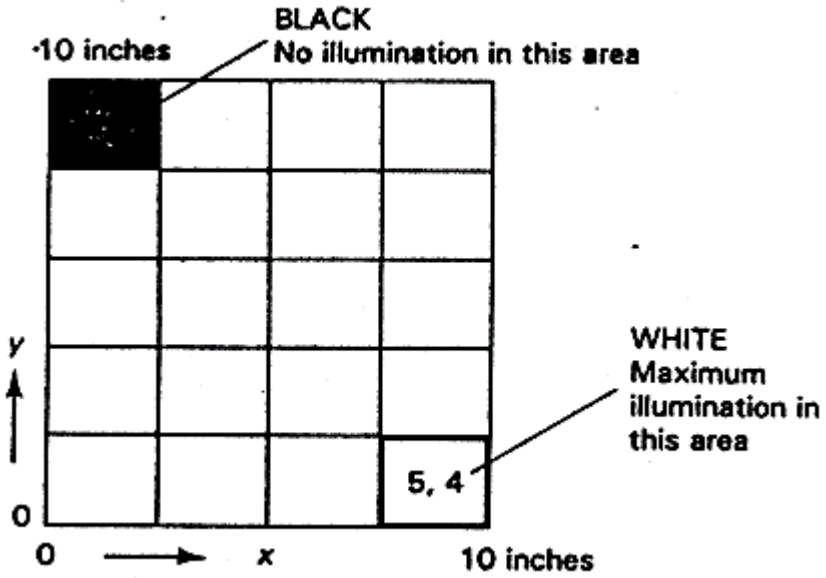
12-by-8 pixel grids, shown with row and column numbers.
On the left, rows are numbered from top to bottom,
on the right, they are numberd bottom to top.

http://math.hws.edu/graphicsbook/c2/s1.html

Image Fundamentals

2. Image Coordinates

- Pixel location
  - A pixel at location (n, m) has a numerical value which is the illumination (represented by gray level value on it.
  - For 3-bit gray image
  - 0 – black
  - 8 - white

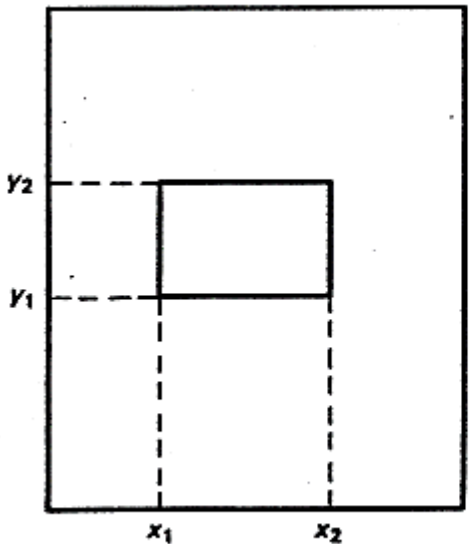


Chui *et. al.* ME5405 Machine Vision. NUS

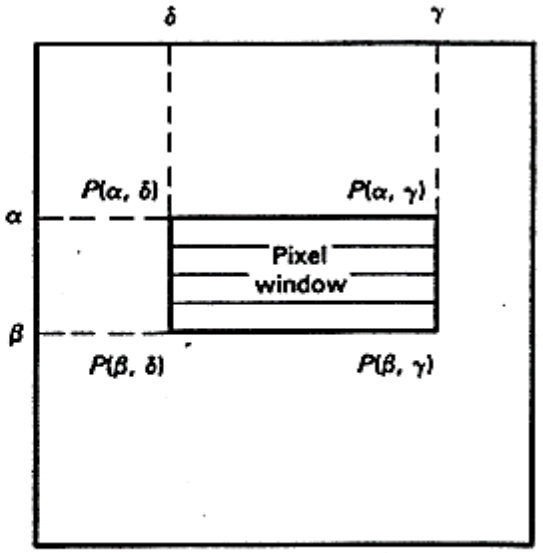
Image Fundamentals

2. Image Coordinates

- Window Area
  - If a certain region of an image is of interest, we can encapsulate it with a window area for processing
  - The region is known as Window of Region of Interest (ROI).



Picture window Area



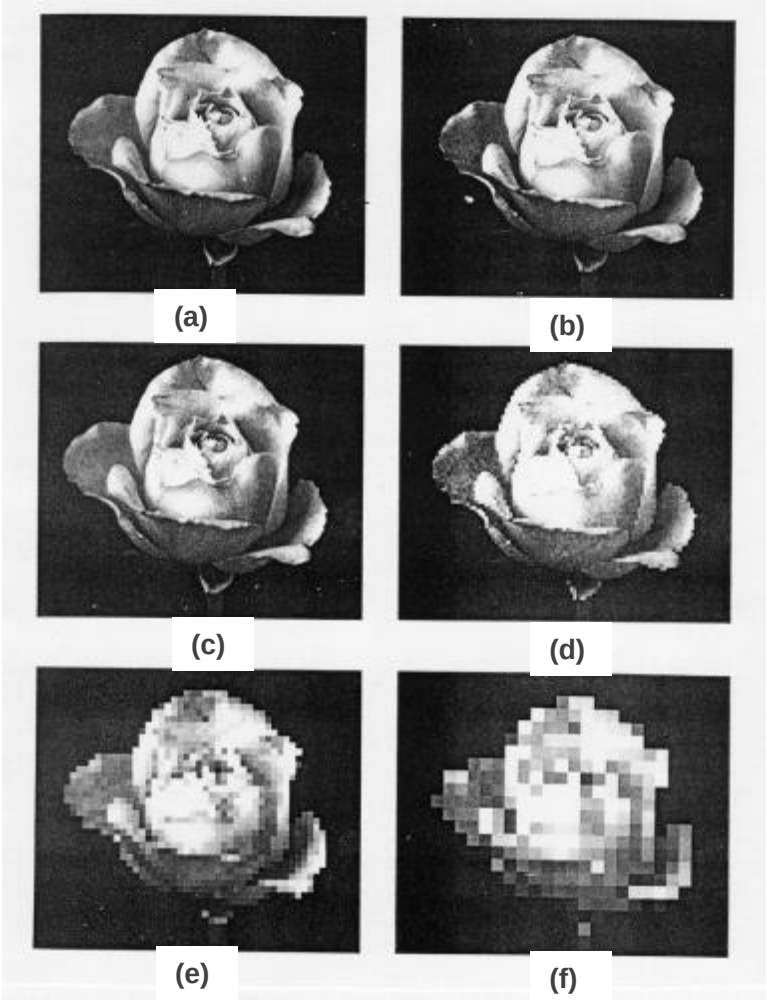
Pixel Matrix

Chui *et. al.* ME5405 Machine Vision. NUS

Image Fundamentals

3. Image Properties

- **Spatial resolution**
  - Figure shows a digital image of a rose.
  - (a) to (f) shows the results of reducing the spatial resolution from N = 1024, 512, 256, 128, 64 and 32, respectively.
  - Display area is kept constant at 1024 X 2014
  - Pixels in the lowest resolution images were duplicated to fill the entire display.
  - Hence, a checker board effect for lower resolution

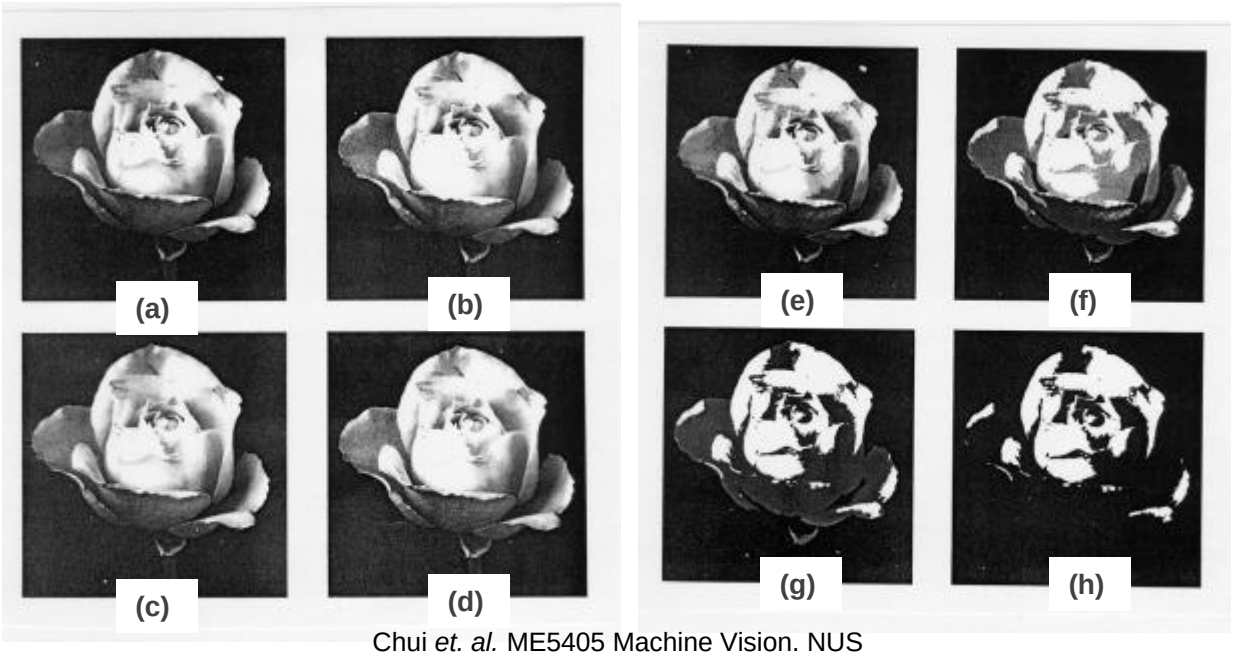


Chui *et. al.* ME5405 Machine Vision. NUS

Image Fundamentals

3. Image Properties

- **Effect of number of gray levels**
  - Figure shows a 1024 X 1024, 8-bit image. (b) to (h) were obtained by reducing the number of bits from m = 7 to 1, with the same spatial resolution constant
  - The 256, 128 and 64 gray level images are almost identical. (h) with m =1 is basically a binary image.



Chui *et. al.* ME5405 Machine Vision. NUS

Image Fundamentals

3. Image Properties

Gray Levels

For a range of values for illumination, we can increase the number of bits representing the pixel value.

Bit number	Gray Scale	Gray level value
1	2^1 = 2 values	0 to 1
3	2^3 = 8 values	0 to 7
4	2^4 = 16 values	0 to 15

The lowest value is assigned to black, and the maximum value is assigned to white. The values assigned to pixel are always integers.

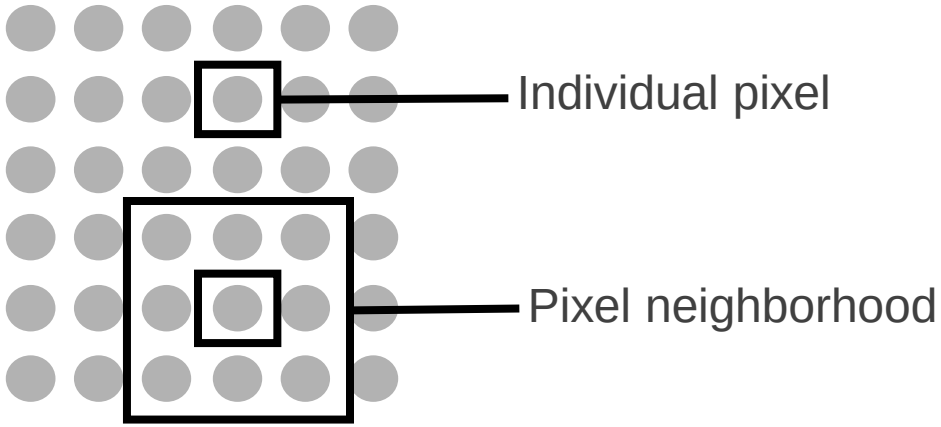
Basic Image Processing

Basic Image Processing

The processing of image can be categorized into:

- Point by point (monadic) alteration of data on a global scale, and
- Multiple points (dyadic) determination of elements of a new array of the image.

The generation of the new pixel image matrix will be a function of either individual pixel location values or values of pixels in the neighborhood of the reference cell, as shown below.



Basic Image Processing

Monadic Operations:

- Involve the generation of new array by modifying pixel value at every single location based on a global rule
- Process involves:
  - Obtaining pixel value of original array
  - Modifying it by a linear or non-linear operation
  - Placing the new pixel value in the corresponding location of the new array

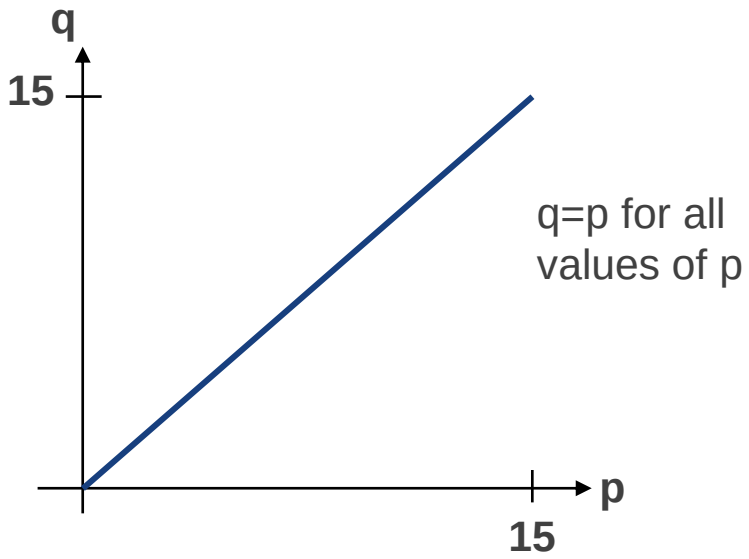
- Process is repeated and continued over entire array.

$$g(i,j) = f[p(i,j)]$$

Basic Image Processing

a. Identity Operator

This operation results in the creation of an output image which is identical to the input image. The input image is p and the output image is q. The function f is a straight line starting at the origin and extending to the maximum pixel value of the image system.



Input

Output

http://ict.udlap.mx/people/oleg/docencia/imagenes/chapter2/image_21_IS548.html

$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix}$$

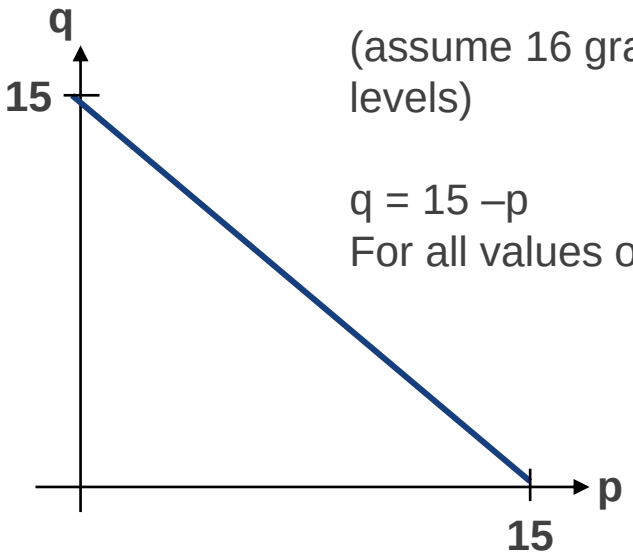
Input matrix (p)

Output matrix (q)

Basic Image Processing

b. Inverse Operator

This operation results in the creation of an output image which is the inverse of the input image. The function f is a straight line having a value of maximum at minimum gray level input value and equal to zero at the maximum gray input value.



(assume 16 gray levels)

q = 15 –p
For all values of p

Input

Output

http://ict.udlap.mx/people/oleg/docencia/imagenes/chapter2/image_21_IS548.html

$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 15 \\ 15 & 13 & 1 \\ 4 & 13 & 12 \end{bmatrix}$$

Input matrix (p)

Output matrix (q)

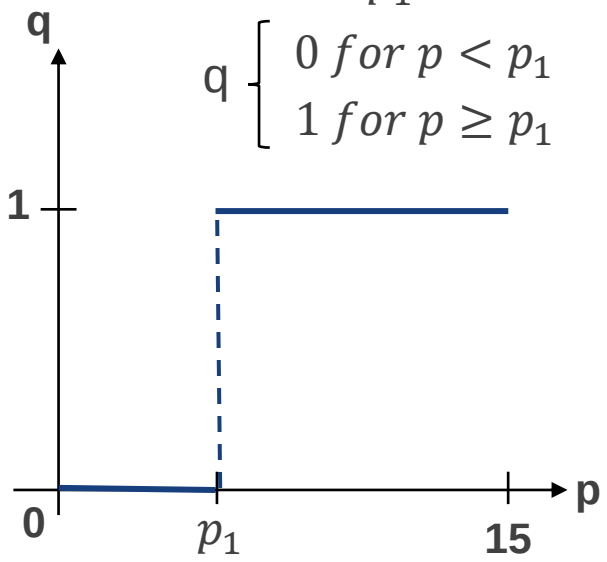
Basic Image Processing

c. Threshold Operator

This class operator results in a binary output image from a gray scale input image where the level of transition is given by the input parameter p_1 as the threshold. All pixels below p_1 are converted to zero and pixels equal to or greater than p_1 are converted to 1.

$$Let\ p_1 = 5$$

$$q \begin{cases} 0\ for\ p < p_1 \\ 1\ for\ p \geq p_1 \end{cases}$$



Input

Output

http://ict.udlap.mx/people/oleg/docencia/imagenes/chapter2/image_21_IS548.html

$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

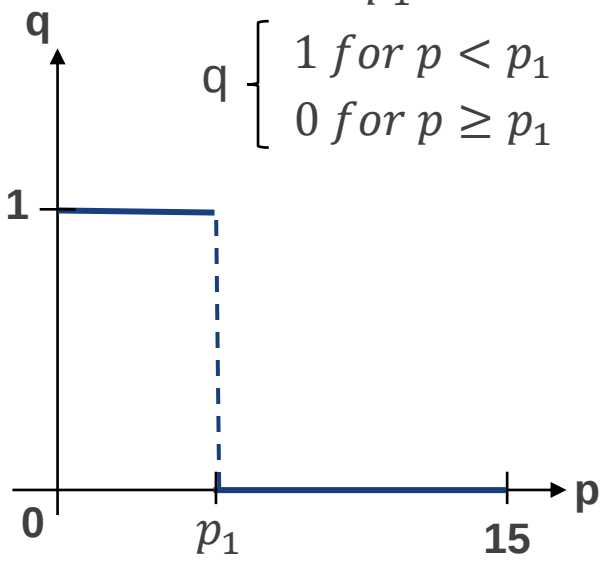
Basic Image Processing

d. Inverse Threshold Operator

This class operator is similar to the Threshold Operator. However, all pixels below p_1 are converted to 1 and pixels equal to or greater than p_1 are converted to 0.

$$Let\ p_1 = 5$$

$$q \begin{cases} 1\ for\ p < p_1 \\ 0\ for\ p \geq p_1 \end{cases}$$



Input Output

http://ict.udlap.mx/people/oleg/docencia/imagenes/chapter2/image_21_IS548.html

$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

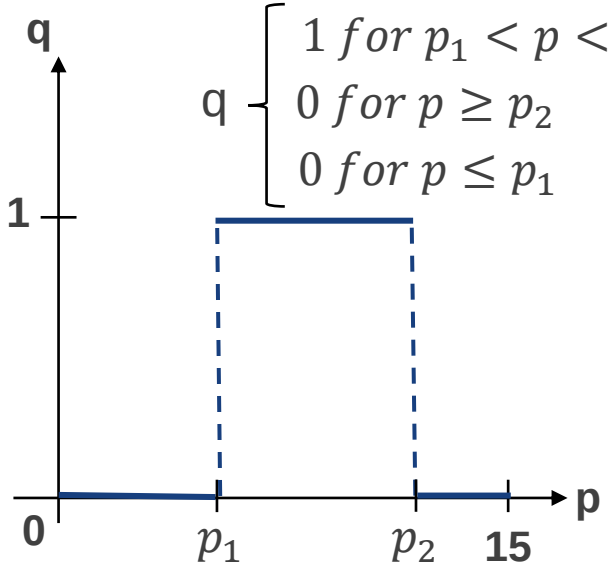
Basic Image Processing

e. Binary Threshold Interval Operator

This operator can be used to convert a multi-gray scale to a binary image, or it can be applied to a binary image for inversion purposes. All pixels with value between p_1 and p_2 are converted to one and the rest are converted to zero.

$$Let\ p_1 = 3$$
$$Let\ p_2 = 11$$

$$q = \begin{cases} 1\ for\ p_1 < p < p_2 \\ 0\ for\ p \geq p_2 \\ 0\ for\ p \leq p_1 \end{cases}$$



$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

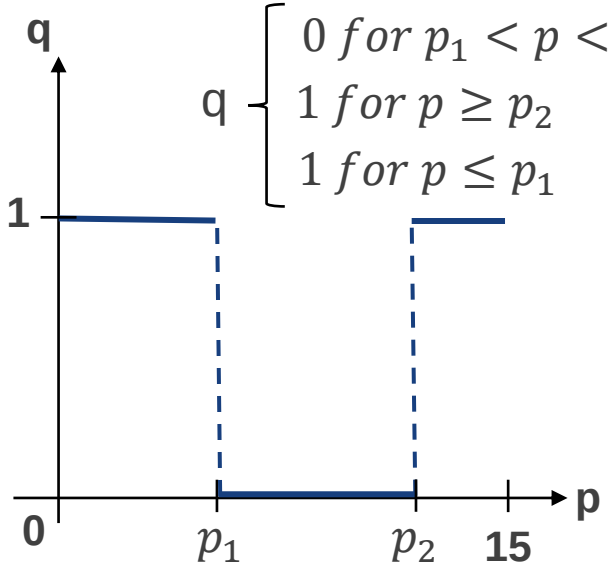
Basic Image Processing

f. Inverted Binary Threshold Interval Operator

This operator is similar to the previous in that it can be applied to a binary image for inversion purposes. All pixels with value between p_1 and p_2 are converted to zero and the rest are converted to one.

$$Let\ p_1 = 3$$
$$Let\ p_2 = 11$$

$$q \begin{cases} 0\ for\ p_1 < p < p_2 \\ 1\ for\ p \geq p_2 \\ 1\ for\ p \leq p_1 \end{cases}$$



$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

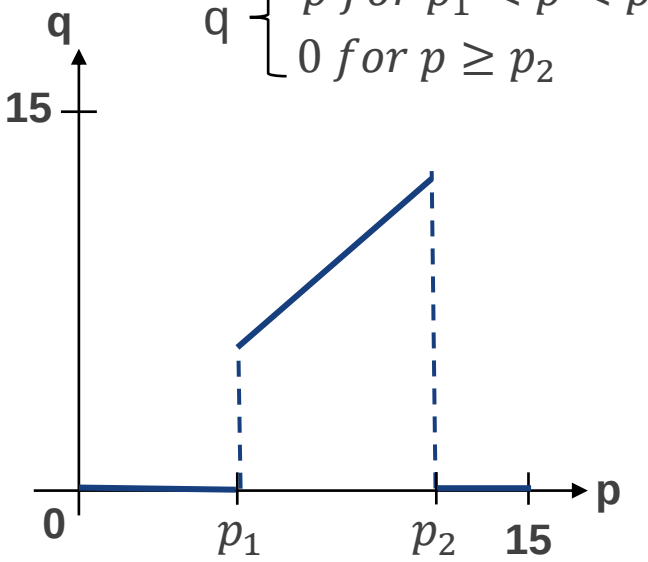
Input matrix (p) Output matrix (q)

Basic Image Processing

g. Gray Scale Threshold Operator

This class of operator results in a gray scale output image value between p_1 and p_2 and makes all input values outside of p_1 to p_2 zero. This operator can be used to identify image features having a specific value for pseudo-color application processing techniques.

$$Let\ p_1 = 3$$
$$Let\ p_2 = 11$$
$$q \begin{cases} p\ for\ p_1 < p < p_2 \\ 0\ for\ p \geq p_2 \end{cases}$$



$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 10 & 0 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

Basic Image Processing

h. Inverted Gray Scale Threshold Operator

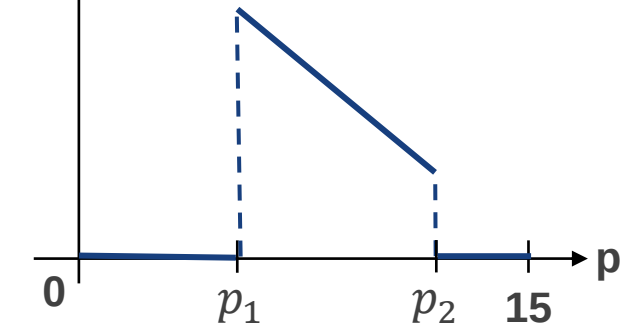
This class of operator is the inverted of the previous. It can highlight specific features such as roads or all the similar areas in the image

Assuming 16 gray levels

$Let\ p_1 = 3$

$Let\ p_2 = 11$

$$q \begin{cases} 15 - p\ for\ p_1 < p < p_2 \\ 0\ for\ p \geq p_2 \\ 0\ for\ p \leq p_1 \end{cases}$$



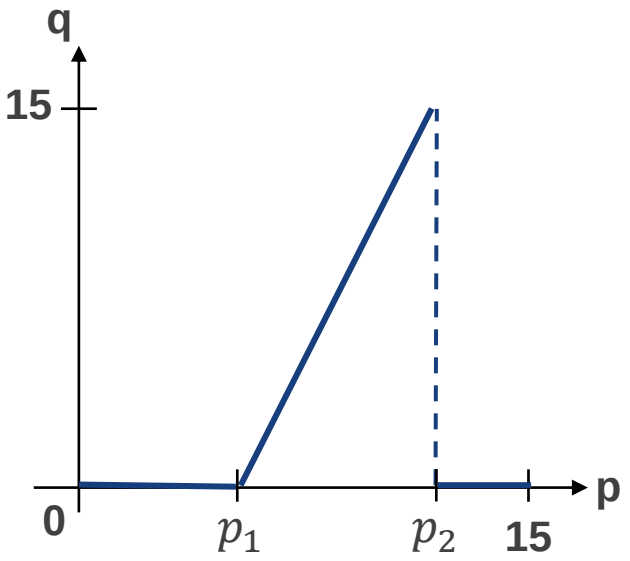
$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 5 & 0 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

Basic Image Processing

i. Stretch Operator

This class of operators results in a full gray scale output image corresponding to the input interval p_1 and p_2 and suppresses all values outside this range.



Assuming 16 gray levels

$Let\ p_1 = 3$

$Let\ p_2 = 11$

$$q \begin{cases} (p - p_1) \times \dfrac{15}{p_2 - p_1} for\ p_1 < p < p_2 \\ 0\ for\ p \leq p_1\ and\ \geq p_2 \end{cases}$$

$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 13 & 0 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

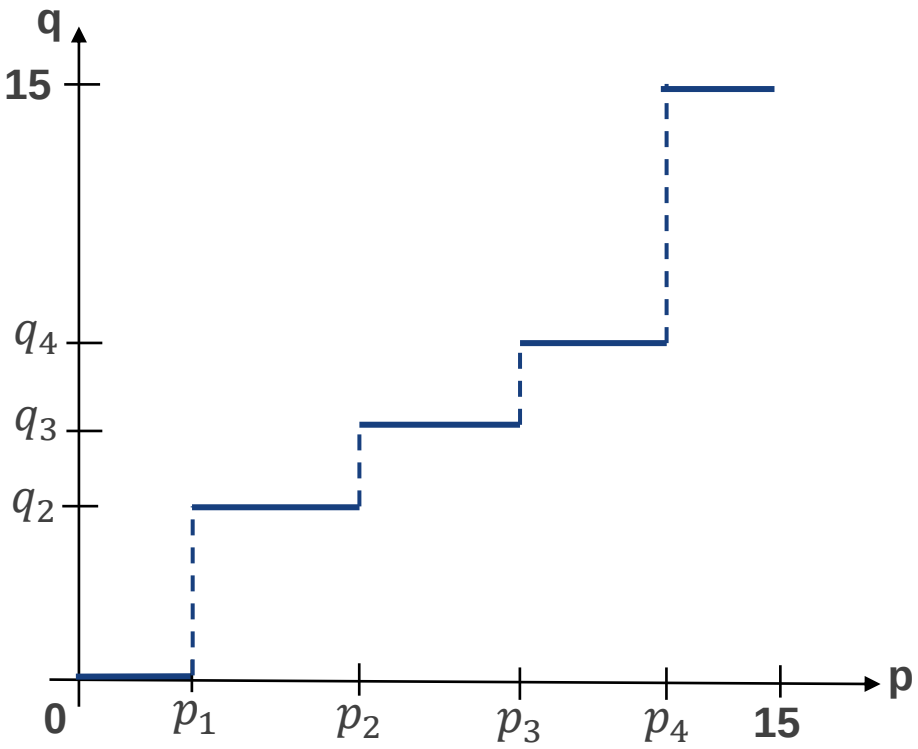
Basic Image Processing

j. Gray Level Reduction Operator

This class of operators results in an output image which has a smaller number of gray levels than the number of gray levels of the input image, where an input image with 15 levels is converted to an image having five levels: q_1, q_2 , q_3, q_4 and 15.

$$Let\ p_1 = 2\ and\ q_2 = 4$$
$$Let\ p_2 = 4$$
$$Let\ p_3 = 7\ and\ q_3 = 7$$
$$Let\ p_4 = 11\ and\ q_4 = 10$$

$$q \begin{cases} 0\ for\ p \le p_1 \\ q_2\ for\ p_1 \le p \le p_2 \\ q_3\ for\ p_2 \le p \le p_3 \\ q_4\ for\ p_3 \le p \le p_4 \\ 15\ for\ p_4 \le p \le 15 \end{cases}$$



$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 15 & 15 & 0 \\ 0 & 7 & 15 \\ 10 & 10 & 4 \end{bmatrix}$$

Input matrix (p) | Output matrix (q)

Basic Image Processing

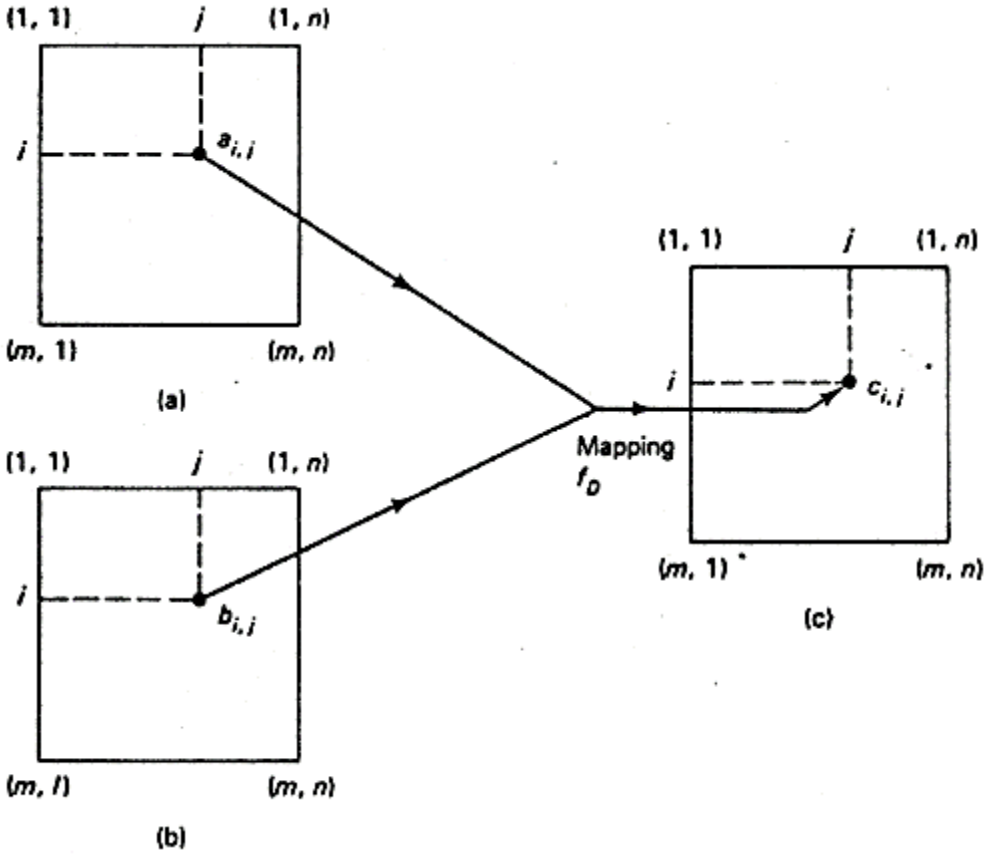
Dyadic Two Point Transformations

The dyadic point-by-point operator uses the information contained at the same location in two images. The size of image does not change and can be linear or non-linear.

Transformation involves two variables associated with pairs of corresponding pixels.

$$c(i, j) = f[a(i, j) , b(i, j)]$$

where a and b are input matrices, f is functional operator and c is output matrix.



Chui *et. al.* ME5405 Machine Vision. NUS

Basic Image Processing

a. Image Addition

It can be used to reduce the effect of noise in the data. The value of output c_{ij} is given by

$$c_{ij} = \frac{(a_{ij} + b_{ij})}{k}$$

over the range of values where k equals to number of samples.

Special rules, such as rounding up, must be applied to the function to produce meaningful results.

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix}$$

a_{ij}
Input 1

$$\begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix}$$

b_{ij}
Input 2

$$\begin{bmatrix} 7 & 12 & 76 & 254 \\ 2 & 6 & 40 & 254 \\ 18 & 1 & 20 & 255 \\ 23 & 2 & 11 & 248 \end{bmatrix}$$

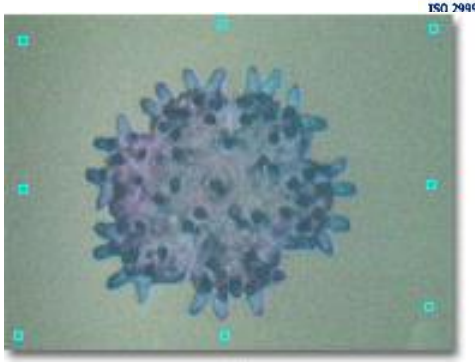
c_{ij}
Output

Basic Image Processing

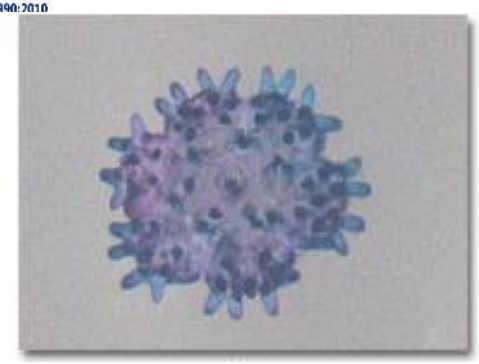
b. Image Subtraction

It can be used to remove background or detect changes in a scene. The operator is given by:

$$c_{ij} = k(a_{ij} - b_{ij})$$



Original



Background subtracted

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} \begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix} \begin{bmatrix} -14 & 1 & 133 & 2 \\ -2 & -19 & -60 & 254 \\ 15 & -1 & -60 & 254 \\ 0 & 0 & -100 & -15 \end{bmatrix}$$

Input 1 = a_{ij} ***Input 2 = b_{ij}*** ***Output = c_{ij}****

$$\begin{bmatrix} 14 & 1 & 133 & 2 \\ 2 & 19 & 60 & 254 \\ 15 & 1 & 60 & 254 \\ 0 & 0 & 100 & 15 \end{bmatrix}$$

Output = c_{ij}

Basic Image Processing

b. Image Subtraction

$c_{ij} = k(a_{ij} - b_{ij})$

Since image processing uses positive numbers, it is necessary to define the output as positive. This could involve a rescaling where the largest negative number is set equal to zero and the largest number is set to the maximum gray scale value. Another approach is to obtain the absolute of the output.

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix}$$

Input 1 = a_{ij}

$$\begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix}$$

Input 2 = b_{ij}

$$\begin{bmatrix} -14 & 1 & 133 & 2 \\ -2 & -19 & -60 & 254 \\ 15 & -1 & -60 & 254 \\ 0 & 0 & -100 & -15 \end{bmatrix}$$

Output = c_{ij}*

$$\begin{bmatrix} 14 & 1 & 133 & 2 \\ 2 & 19 & 60 & 254 \\ 15 & 1 & 60 & 254 \\ 0 & 0 & 100 & 15 \end{bmatrix}$$

Output = c_{ij}

Absolute of the output method

Basic Image Processing

b. Image Subtraction

Rescaling method:

$$\boldsymbol{R_{ij} = (c_{ij} + 100)X \frac{255}{354}} \qquad (7)$$

is used to convert the values. R_{ij} will be 0 when c_{ij} is -100 (min value) and 255 when c_{ij} is 254 (max value).

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} \quad \begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix} \quad \begin{bmatrix} -14 & 1 & 133 & 2 \\ -2 & -19 & -60 & 254 \\ 15 & -1 & -60 & 254 \\ 0 & 0 & -100 & -15 \end{bmatrix} \quad \begin{bmatrix} 82 & 74 & 158 & 74 \\ 71 & 59 & 29 & 255 \\ 83 & 72 & 29 & 255 \\ 72 & 72 & 0 & 62 \end{bmatrix}$$

Input 1 = a_{ij} ***Input 2 = b_{ij}*** ***Output = c_{ij}**** ***Output = c_{ij}***

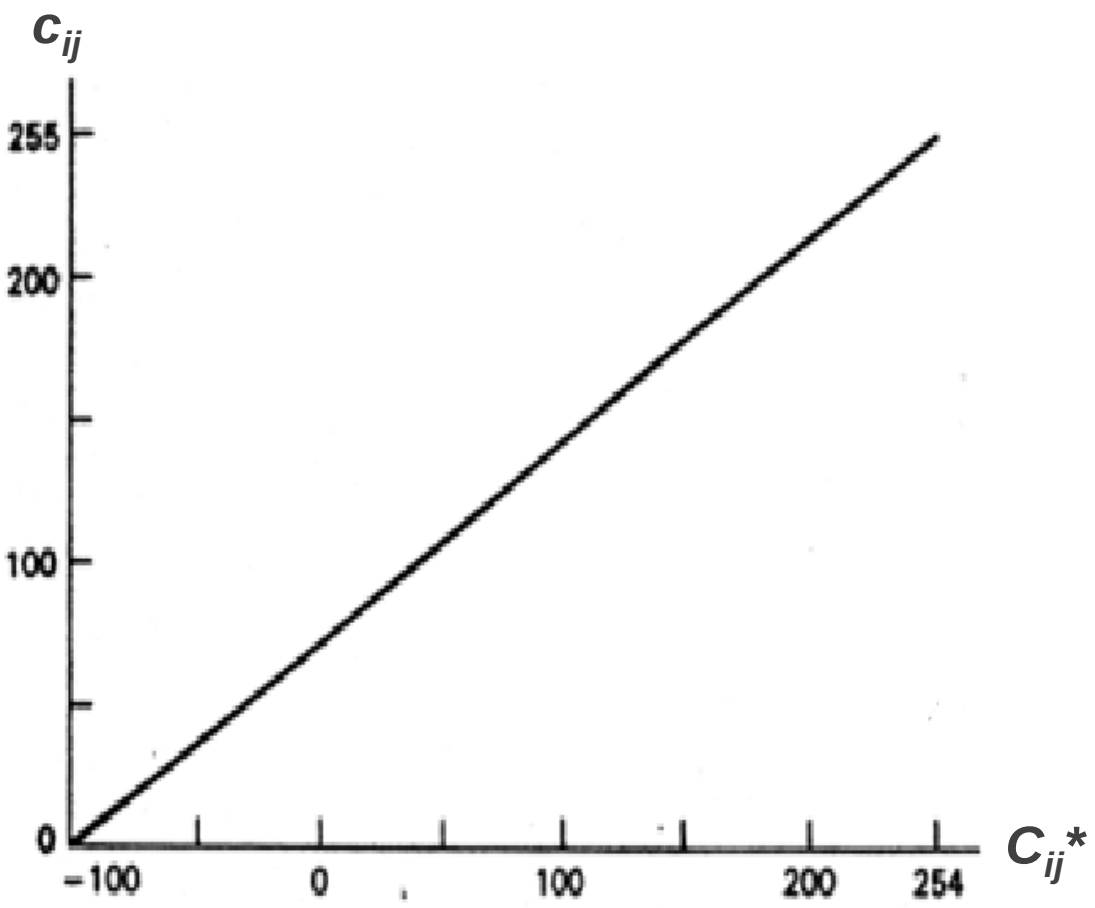
Rescaling of the output method for image subtraction

Basic Image Processing

b. Image Subtraction

$$c_{ij} = \left(c_{ij}^{*} + 100\right) X \frac{255}{354}$$

c_{ij} will be zero when c_{ij} is -100 (minimum value) and 255 when c_{ij}* is 254 (max value).*



Basic Image Processing

c. Convolution: Spatial Transformation

A new image can be generated where the pixel assigned to each location is a function of the pixel values of the adjacent locations. The mask can be 3x3 or 5x5 or any number.
In a convolution, the pixel value is computed based on its neighbors. The new value does not replace the old matrix but is instead placed in the corresponding position in a new matrix. The location is then shifted by 1 and the process is repeated.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} + & + & - \\ - & + & + \\ + & + & - \end{bmatrix} = \begin{bmatrix} & e^* & \end{bmatrix}$$

$$e^* = +a + b - c - d + e + f + g + h - i$$

Basic Image Processing

c. Convolution: Spatial Transformation – an example

Given a partial image shown in Fig. 20(a). It is to be operated on (or filtered) by the masks (or filter) given by Fig. 20(b(i) and (ii)) separately on the central pixel ’10’..

$$\begin{bmatrix} x & x & x & x & x \\ x & 254 & 251 & 255 & x \\ x & 250 & 10 & 249 & x \\ x & 255 & 252 & 248 & x \\ x & x & x & x & x \end{bmatrix}$$

Fig. 20b(i)

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & x & x & x & x \\ x & 254 & 251 & 255 & x \\ x & 250 & 225 & 249 & x \\ x & 255 & 252 & 248 & x \\ x & x & x & x & x \end{bmatrix}$$

The central pixel
=(254+251+255+250+10+249+255+252+248)
= 224.89 = 225

Fig. 20b(ii)

$$\begin{bmatrix} 2 & 1 & 2 \\ 0 & 0 & 0 \\ -2 & -1 & -2 \end{bmatrix}$$

$$\begin{bmatrix} x & x & x & x & x \\ x & 254 & 251 & 255 & x \\ x & 250 & 11 & 249 & x \\ x & 255 & 252 & 248 & x \\ x & x & x & x & x \end{bmatrix}$$

The central pixel =
(2×254+1×251+2×255+0×250+0×10+0×249+(-2)×255+(-1) ×252+(-2)×248)
= 11

3. Neighbors and Connectivity

Neighbors

A pixel **p** at coordinates (x, y) has four horizontal and vertical neighbors:

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

This set of pixels, called the 4-neighbours of **p**, is denoted by $\mathbf{N_4(p)}$. Each of these pixels is a unit distance from (x, y) and some of these neighbors of **p** will be outside the digital image if (x, y) is on the border of the image.

The four diagonal neighbors of **p** have coordinates:

$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$

and will be denoted by $\mathbf{N_D(p)}$.

The combined of $\mathbf{N_4(p)}$ and $\mathbf{N_D(p)}$, are called the 8-neighbours of **p**, denoted by $\mathbf{N_8(p)}$.

Basic Image Processing

3. Neighbors and Connectivity

Connectivity

Connectivity between pixels is an important concept used in establishing boundaries of objects and components of regions in an image.

To establish whether connectivity exists between two pixels, we must set certain criteria:

- whether they are adjacent in some sense (e.g., if they are 4-neighbours); and
- whether their gray levels satisfy a specified criterion of similarity (e.g., if they are equal).

Two pixels in an image might be $\mathbf{N_4}$ neighbors, but they are connected only when they have the same (or closely related) gray level value.

Let **V** be the set of gray-level values to determine connectivity; for example, if only connectivity of pixels with intensities of 59, 60, and 61 are important, then **V** = {59, 60, 61}.

Basic Image Processing

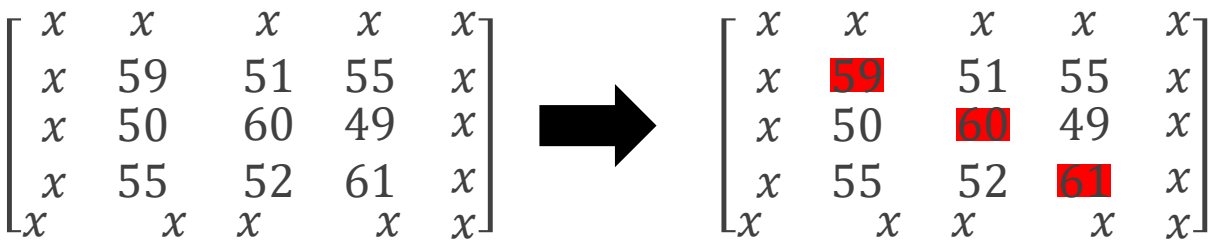
3. Neighbors and Connectivity

Connectivity

If **V** is a set of neighboring pixels which have some common characteristics, there are two main types of connectivity:

1. **4-connectivity**:Two pixels **p** and **q** with values from **V** are 4-connected if **q** is in the set $\mathbf{N_4(p)}$.

2. **8-connectivity**:Two pixels **p** and **q** with values from **V** are 8-connected if **q** is in the set $\mathbf{N_8(p)}$.



Binary Machine Vision

Introduction

For object recognition and detection by a robotic system, the input image may be assumed and simplified by generating an output whose pixels:

- Tend to have high values if they are part of an object of interest
- Low values if they are not

Introduction

To highlight an object from the image, the easiest way is to perform a **Thresholding** operation to produce a binary image.

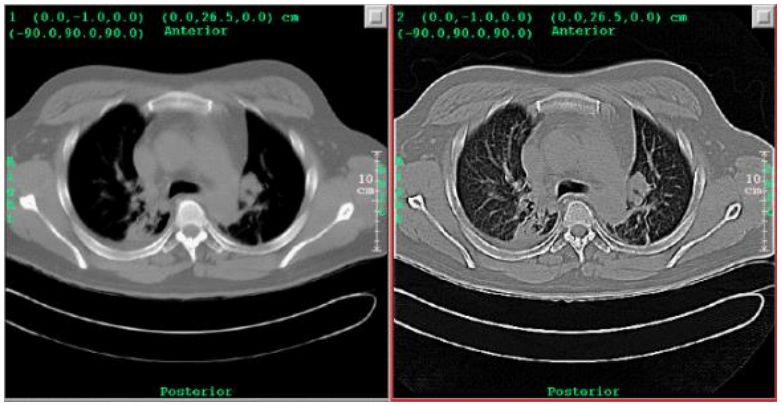
- Determine a threshold value
- Pixels having gray value higher than threshold are given 1
- Other lower than threshold are given 0

https://www.scipy-lectures.org/packages/scikit-image/index.html

Introduction

The conversion of gray-level images to binary representation is important for a number of reasons:

- To identify objects of interest in the image. For example, we may wish to separate a visual target from its background.
- For shape analysis, in which case the intensities of pixels are less significant than the shape of a region.
- Enhancement of edges in an image. It is necessary to distinguish strong edges that correspond to object outlines (and features) from weak edges due to illumination changes, shadows, etc.

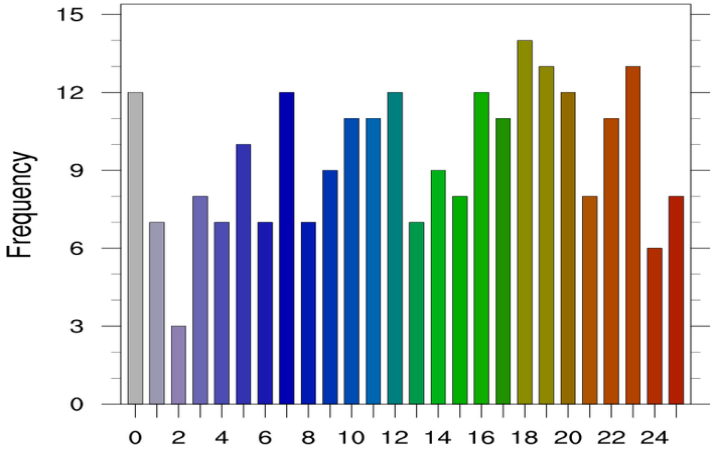


http://www.mathresolutions.com/edgeenh.htm

Thresholding

- Thresholding is a labelling operation
- Main difficulties:
  - The distribution of the bright and dark pixels are usually not known
  - The only clue can be obtained from image histogram, but not always accurate.

?



https://www.ncl.ucar.edu/Applications/histo.shtml

Thresholding

55	104	154	55	60
66	180	144	104	60
71	60	113	71	71
68	123	154	180	68
60	104	86	180	201

Threshold = 113

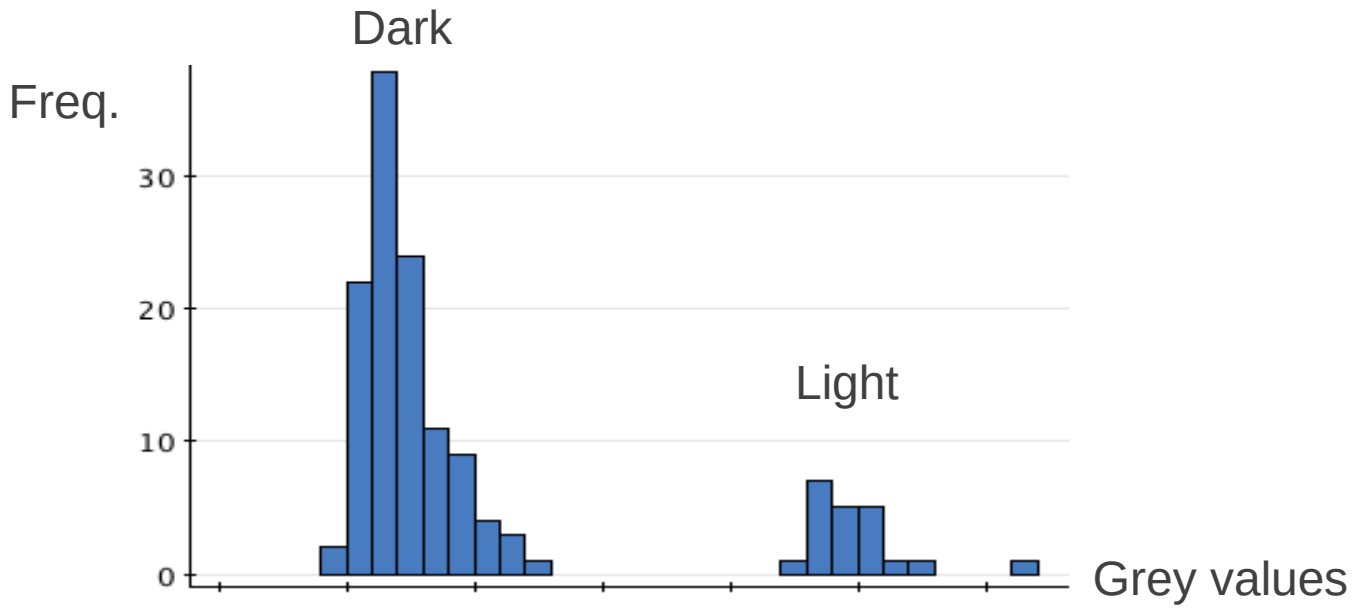
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	1	1	1	0
0	0	0	1	1

Threshold = 123

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

Thresholding

Determine Threshold Value from Histogram



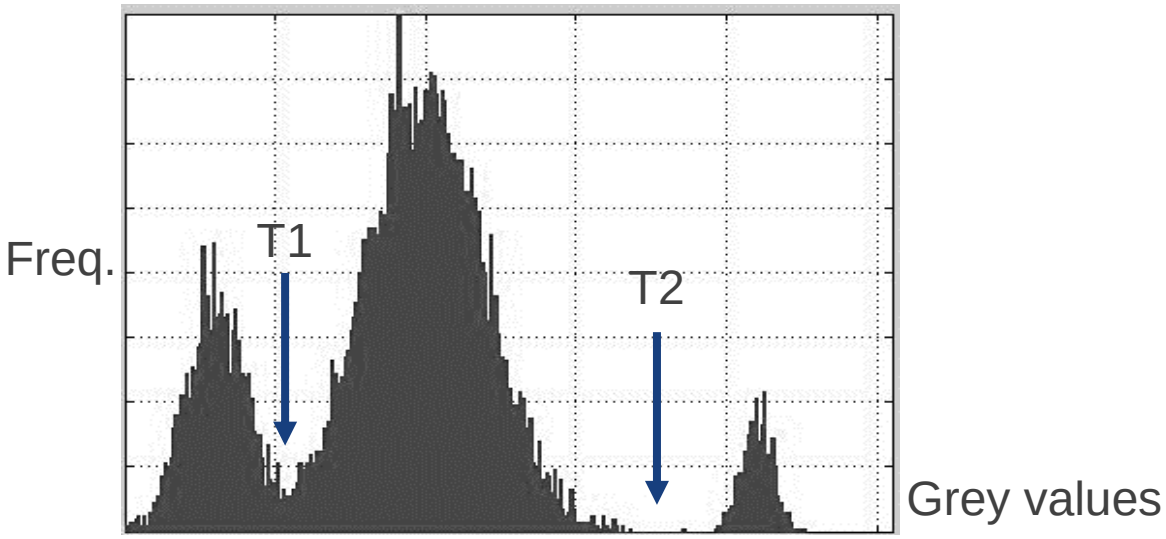
https://www.statcrunch.com/5.0/viewresult.php?resid=867679

- Bi-modal Histogram with little distribution overlaps between the dark and bright pixels. There are two dominant peaks. (If you are lucky!)

- The Threshold value will be at the valley between the two peaks, T.
- Any point (x, y) for which f(x,y) > T is called the object point; otherwise, it is called a background point.

Thresholding

Multiple Thresholds

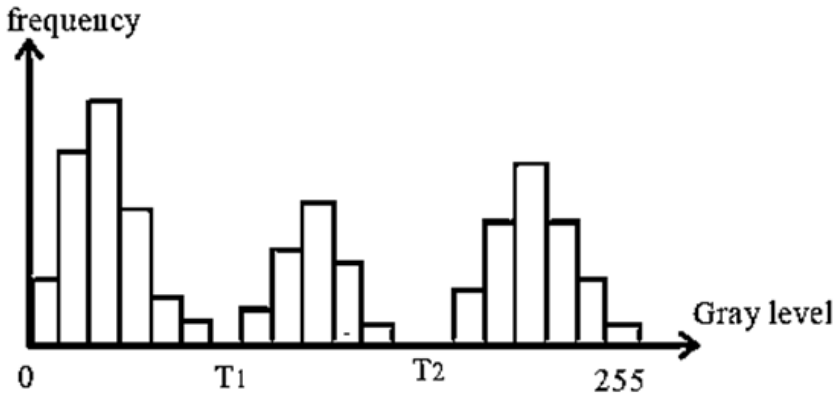


https://robotacademy.net.au/lesson/image-thresholding/

- Above presents three dominant modes in histogram.
- There two Threshold values T_1 and T_2.

Thresholding

Multiple Thresholds



The same classification approach as in the previous case classifies a point (*x,y*) as

belong to the object class if

$$T_1 < f(x,y) \leq T_2$$

belong to the other object class if

$$f(x,y) > T_2$$

to the background is

$$f(x,y) \leq T_1$$

- This type of multi-level thresholding is less reliable than its single-threshold counterpart.
- Difficult to establish the multiple threshold especially when the number of histogram modes is large

Thresholding

Mathematical Definition of Thresholding

In formal mathematical terms, thresholding may be viewed as an operation that involves tests against a function T of the form:

$$T = T[x, y, p(x, y), f(x, y)]$$

where f(x, y) is the gray level of point (x, y) and p(x, y) denotes some local property of the image (for example, the average gray level of a neighborhood centered on(x, y).

An image after having gone through thresholding is defined as:

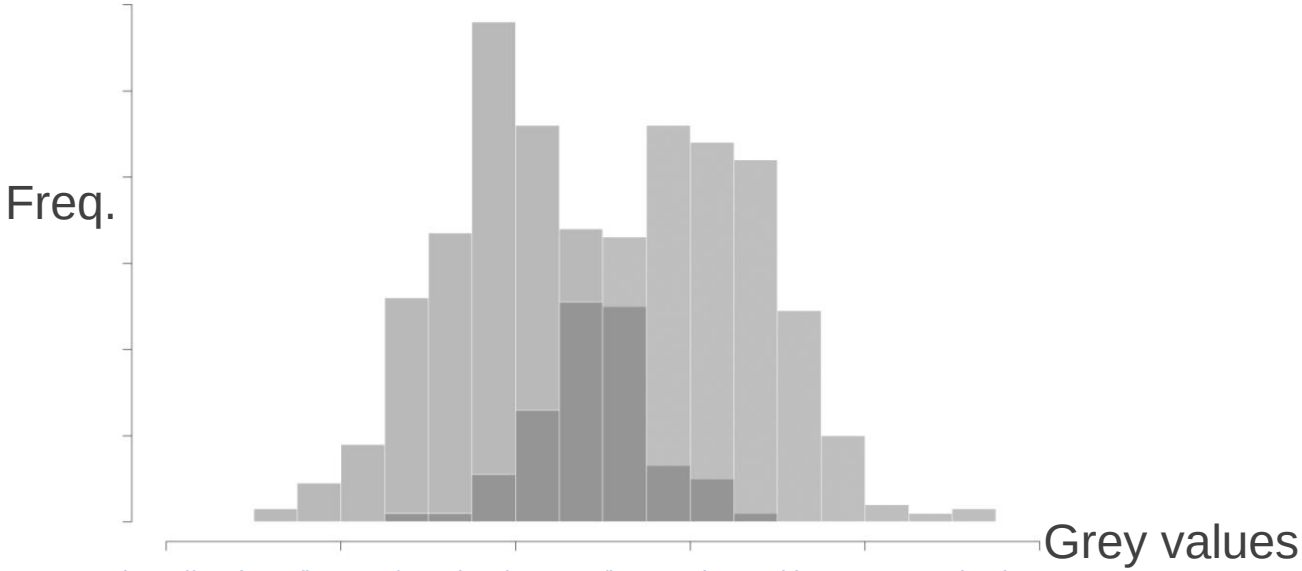
$$g(x, y) = \begin{cases} 1 & if\ f(x, y) > T \\ 0 & if\ f(x, y) \leq T \end{cases}$$

When T depends only on f(x, y), the threshold is called global.
If T depends on both f(x, y) and p(x, y), then the threshold is called local.
If T depends on the spatial coordinates x and y, the threshold is called dynamic

Difficulties in Thresholding

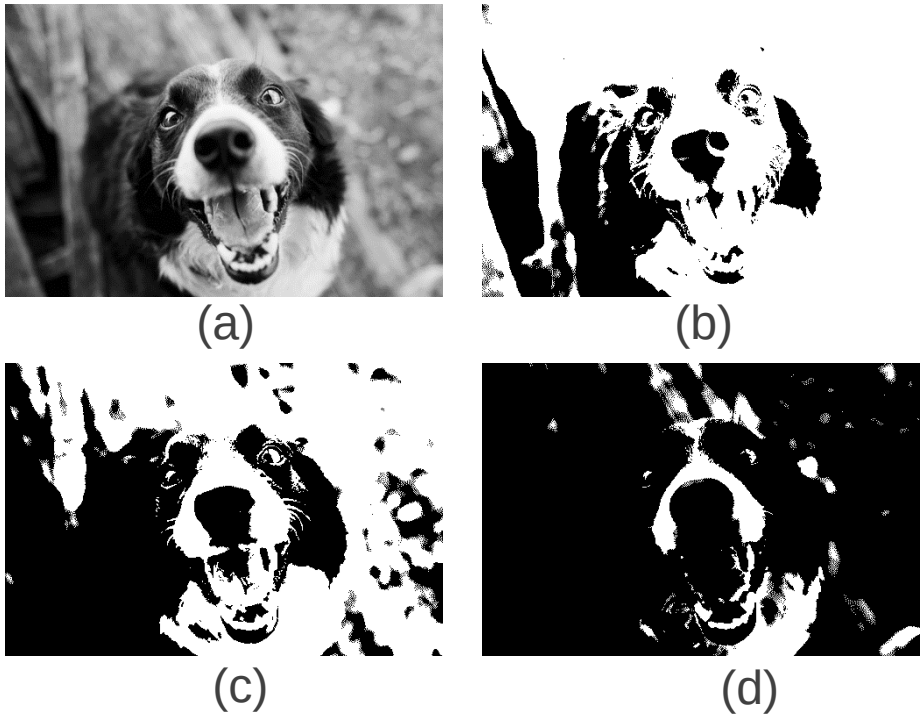
The determination of a threshold value becomes difficult:

- When the distribution of dark and bright pixels become more and more overlapped
- When there is substantial overlap, difficult to minimize classification error



https://stackoverflow.com/questions/3541713/how-to-plot-two-histograms-together-in-r

Difficulties in Thresholding



Chui *et. al.* ME5405 Machine Vision. NUS

Image of the dog under different threshold values.
(a) Original image, (b) under low
(c) under medium and (d) under high threshold values.

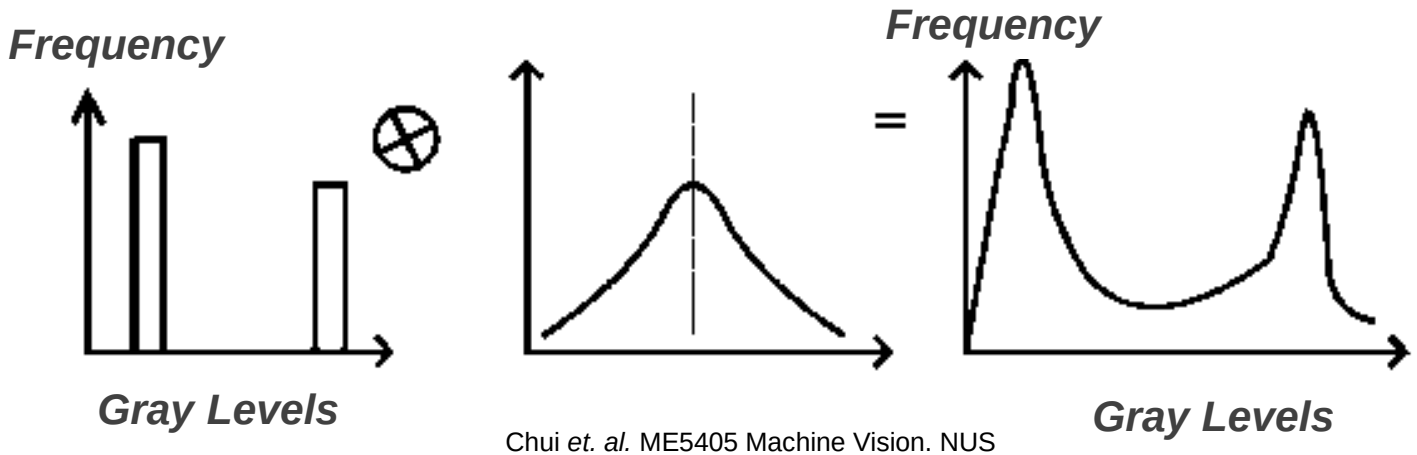
Low threshold results in labelling of more pixel as bright and features are lost.

High threshold results in labelling of more pixels as dark. The object of interest becomes smaller and loses features.

Difficulties in Thresholding

Influence of Noise

If we have a black object on a white background, the histogram will be bi-modal. With the influence of measurement noise, the histogram will be affected as shown below.

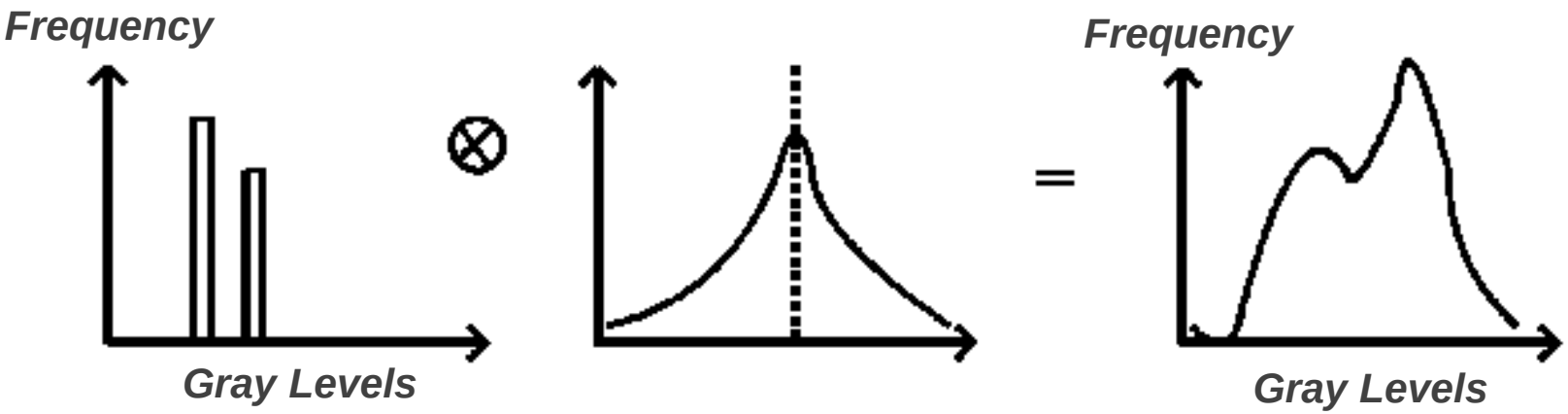


Chui *et. al.* ME5405 Machine Vision. NUS

Difficulties in Thresholding

Influence of Noise

If the gray levels of the object and the background are fairly close, the influence of noise may result in the object only appearing as a shoulder in the histogram. The threshold is also difficult to identify. This problem could be due to poor lighting conditions.

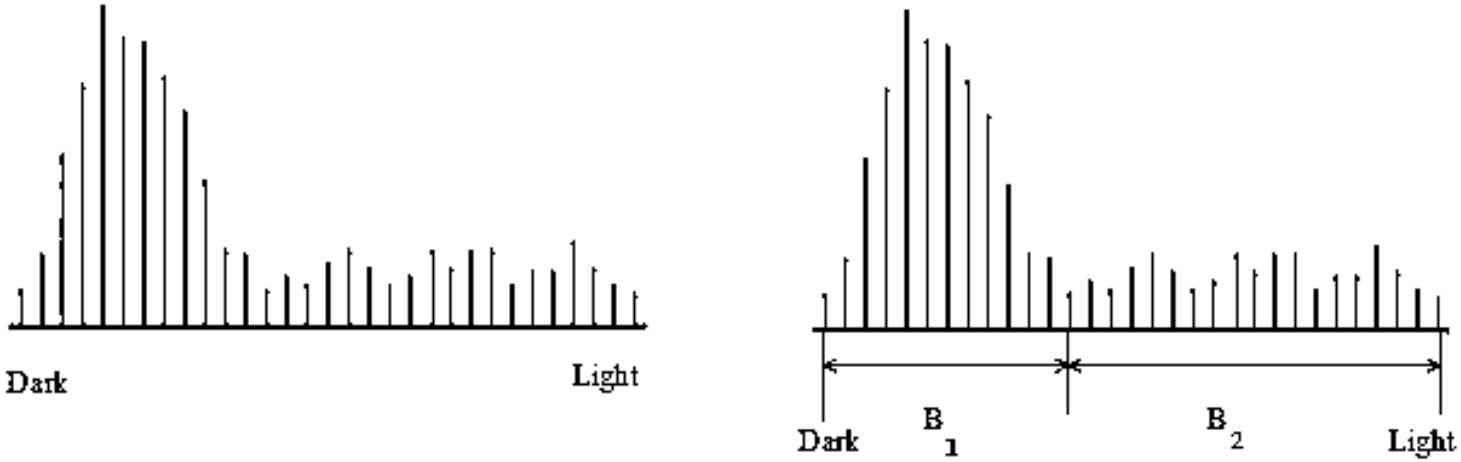


Chui *et. al.* ME5405 Machine Vision. NUS

Thresholding Techniques

Global Thresholding

This is a simple technique and is useful to segment an image by dividing the gray levels into bands and use thresholds to determine regions or to obtain object boundaries.

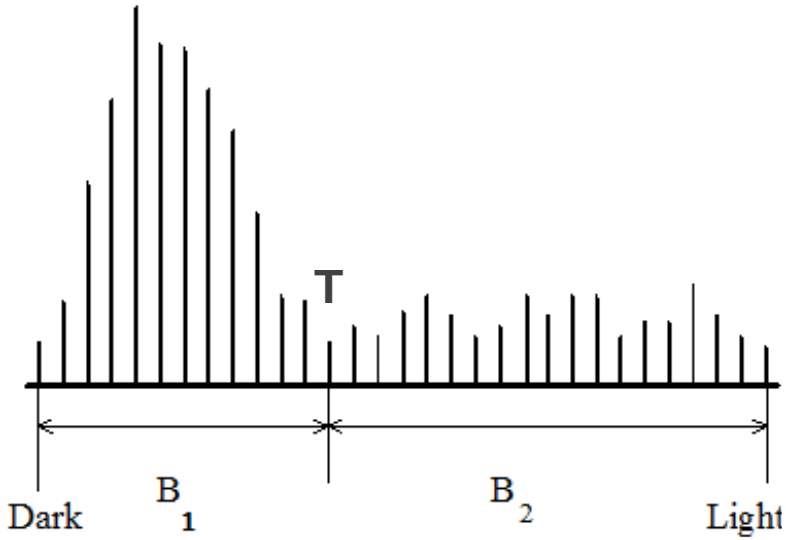


Chui *et. al.* ME5405 Machine Vision. NUS

Global Thresholding

Global Thresholding

This is a simple technique and is useful to segment an image by dividing the gray levels into bands and use thresholds to determine regions or to obtain object boundaries.



Chui *et. al.* ME5405 Machine Vision. NUS

We would like to outline the boundary between the objects and the background, by dividing the two bands separated by a threshold **T**.

We would like to select a **T** such that:
Band B_1: Contains as closely as possible, pixels associated with the background.
Band B_2: Contains as closely as possible, pixels associated with the objects.

Global Thresholding

Process

After fixing T, we scan the given image f(x,y) of size N by N. A change in gray level from one band to the other denotes the presence of a boundary.

Scanning is done is two passes:

Pass 1: for each row in f(x,y), i.e. x = 0, 1, …., N-1;

Pass 2: for each column in f(x,y), i.e., y = 0, 1, ….., N-1.

Pass 1 and 2 are to detect changes in the y and x directions, respectively. The combination of the results of the two passes will yield the boundary of the objects in the image.

Global Thresholding

Pass 1

For each row in f(x,y), i.e., x = 0, 1,…., N-1, create a corresponding row in an intermediate image g1(x,y) using the following relation for y = 0, 1,…,N-1:

$$g1(x,y) = \begin{cases} LE & \text{If the level of } \boldsymbol{f(x,y)} \text{ and } \boldsymbol{f(x,y-1)} \text{ are in different bands of the gray scale} \\ LB & \text{If otherwise} \end{cases}$$

where LE and LB are specified edge and background levels, respectively.

Global Thresholding

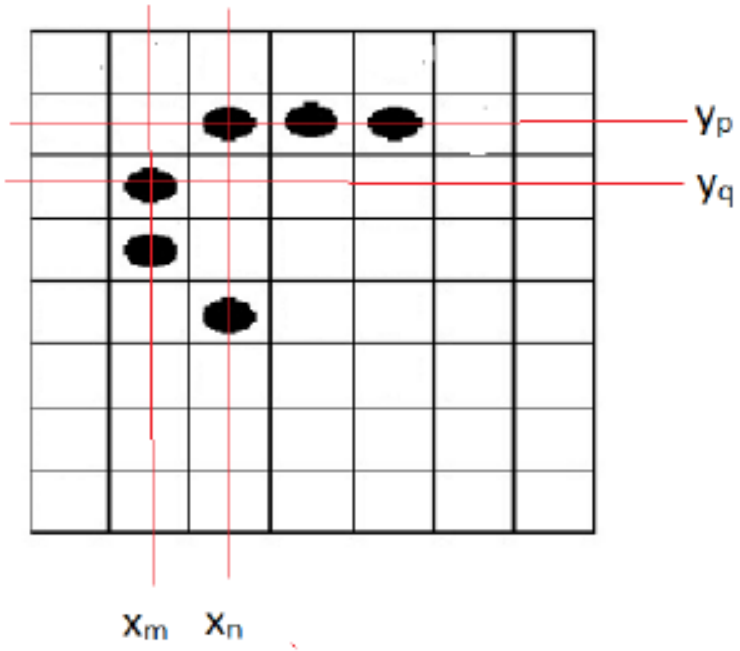
Pass 2

For each column in f(x,y), i.e., y = 0, 1,…., N-1, create a corresponding column in an intermediate image g2(x,y) using the following relation for y = 0, 1,…,N-1:

$$g2(x,y) = \begin{cases} LE & \text{If the level of } \boldsymbol{f(x,y)} \text{ and } \boldsymbol{f(x\text{-}1,y)} \text{ are in different bands of the gray scale} \\ LB & \text{If otherwise} \end{cases}$$

where LE and LB are specified edge and background levels, respectively.

Global Thresholding



$f(x_n, y_p)$ and $f(_{xn+1}, y_p) \Rightarrow g_2(x_n, y_p) = L_B$
$f(x_n, y_p)$ and $f(x_n, y_{p-1}) \Rightarrow g_1(x_n, y_p) = L_E$

$f(x_m, y_q)$ and $f(x_{m-1}, y_q) \Rightarrow g_2(x_m, y_q) = L_E$
$f(x_m, y_q)$ and $f(x_m, y_{q+1}) \Rightarrow g_1(x_m, y_q) = L_B$

Chui *et. al.* ME5405 Machine Vision. NUS

Global Thresholding

Putting them together

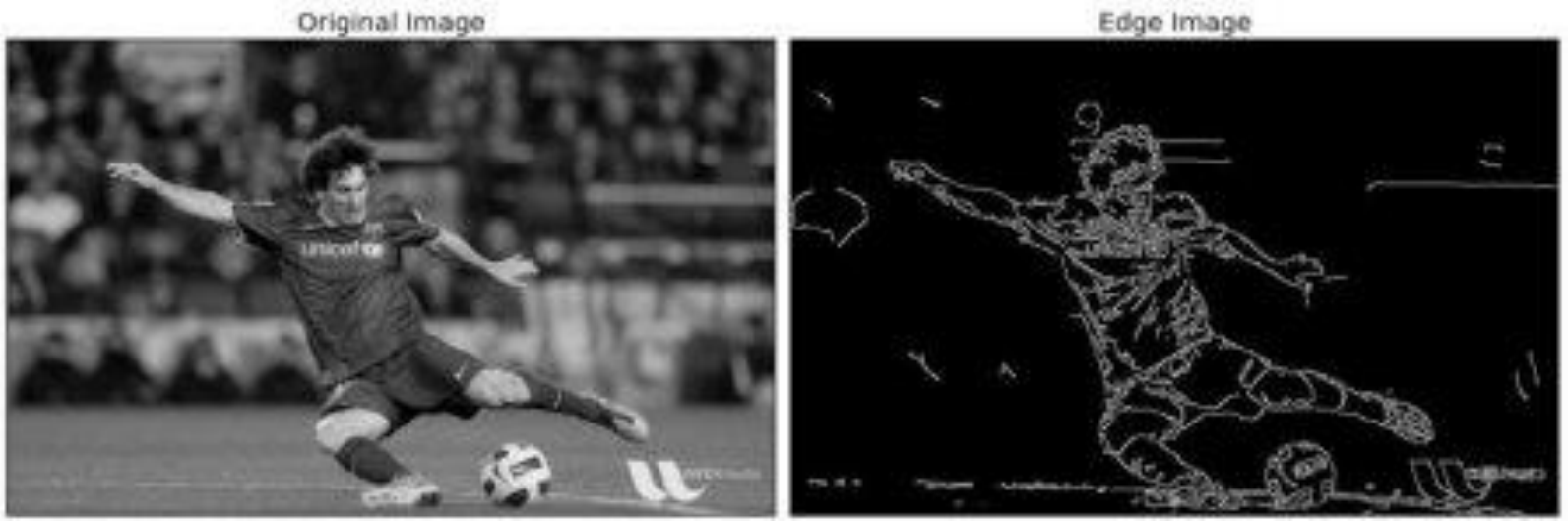
The desired image, consisting of the points on the boundary of objects different (as defined by ***T***) from the background, is obtained by combining the output of pass 1 and 2:

$$g(x,y) = \begin{cases} LE & \text{If } \textbf{\textit{g1(x,y) or g2(x,y)}} \text{ is equal to LE} \\ LB & \text{If otherwise} \end{cases}$$

- The problem in this multi-bands situation is where to place the thresholds.

Global Thresholding

Example of edge detection using thresholding



Ref: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html

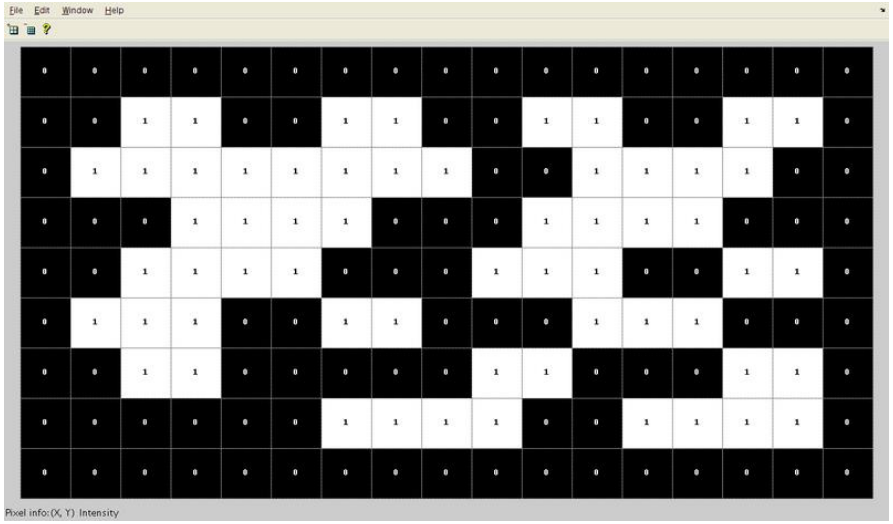
Component Labeling

- Connected components analysis of a binary image consists of the **connected components labeling** of the pixels followed by property measurement of the component regions and decision making.

- **The connected components labeling** operation performs the unit change from pixel to region of segment.

- All pixels that have value binary 1 are given identifying **labels** depends on some conditions.

- The label is a unique name or index of the **region to which the pixels belong**. The label is the identifier for a potential object region.



https://en.wikipedia.org/wiki/Connected-component_labeling

Component Labeling

- Connected components labeling is a grouping operation that can change from pixel to region, which is a more complex unit.

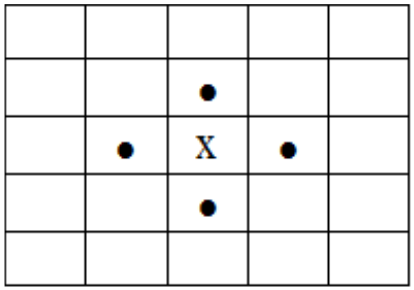
- A region has a richer set of properties.
  - It has shape and position properties as well as statistical properties of the gray levels of the pixels in the region.

- We will examine connected component labeling algorithm, which in essence **group together all pixels belonging to the same region and give them the same label**.

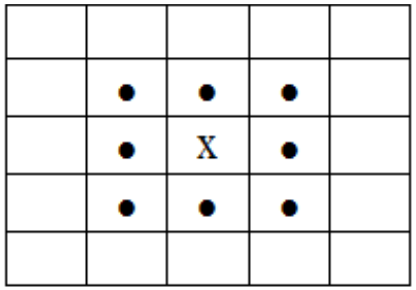
Connected Components Operators

Connectivity

For the connectivity, it is imperative to first define the type of connectivity adopted. In most cases, they are the 4-connectivity and 8-connectivity.



4- connected



8-connected

Connected Components Algorithms

General Principles:

- All the algorithms process a row of the image at a time
- Modifications to process a sub-image rectangular window at a time are straight forward.
- All the algorithms assign **new labels** to the **first pixel of each component** and attempt to propagate the label of a pixel to its neighbors.

Connected Components Algorithms

Simple illustration:

We first perform a 4-connectivity; left-right top-down scan.

1. In the first row, two 1s separated by three 0s are encountered. The first is assigned label 1; the second label 2.
2. In row 2, the first 1 pixel is assigned label 1 because it is a 4 neighbor of the already labelled 1 pixel above it.
3. The second 1-pixel of row 2 is also assigned label 1 because it is a 4-neighbour of the previous labelled 1.
4. The process continues until the pixel marked A. A connected region 1 and 2, hence we can say that they are **equivalent**.

Connected Components Algorithms

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

original image

0	0	1	0	0	0	2
0	0	1	1	0	0	2
0	0	1	1	1	0	2
0	0	1	1	1	1	A

partially processed image

Chui *et. al.* ME5405 Machine Vision. NUS

An Iterative Algorithm

The previous CCA is limited, hence an **iterative algorithm** needs to be adopted

The algorithm consists of the following steps:

1. An initialization step where which pixel is given a unique label

2. A top-down pass, and on each row, a left-right pass in which the value of each non zero pixel is replaced by the minimum value of its non-zero neighbors in a recursive manner

3. A bottom-up pass, and a right-left pass with the same procedure in step 2.

This algorithm selects the minimum label of its neighbors to assign to the pixel of interest.

An Iterative Algorithm

	1	1		1	1	
	1	1		1	1	
	1	1	1	1	1	

(a)

	1	2		3	4	
	5	6		7	8	
	9	10	11	12	13	

(b)

	1	1		3	3	
	1	1		3	3	
	1	1	1	1	1	

(c)

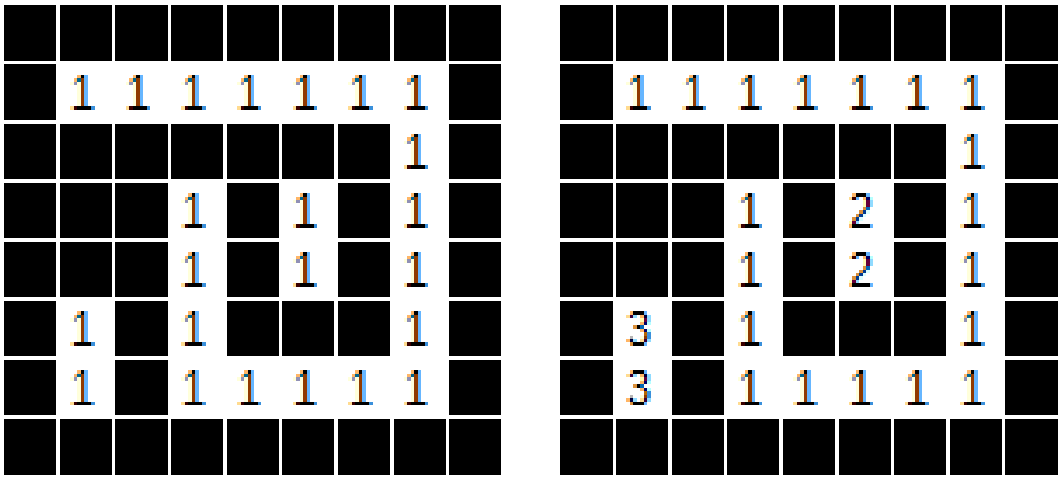
	1	1		1	1	
	1	1		1	1	
	1	1	1	1	1	

(d)

Chui *et. al.* ME5405 Machine Vision. NUS

Iterative algorithm for connected component labeling.
(a) Original image
(b) Result after initialization with unique labels
(c) Result after the first top-down, left-right pass where each non-zero pixel is replaced by the minimum value of its non-zero neighbour
(d) The results after the bottom-up pass

An Iterative Algorithm



https://www.codeproject.com/Articles/825200/An-Implementation-Of-The-Connected-Component-Label

End of Module 1