IMAGE PROCESSING

Dr. Cao Thị Luyen
Email: luyenct@utc.edu.vn
Tel: 0912403345

CONTENT

OBJECT

- General introduction to applied techniques in image processing: including theory and techniques on digital images to transform an original image into another image or another format:
  - Improving image information to increase human perception
  - Processing image data for storage, transmission and display of machines
- Computer Vision
- Image recognition

Reference

- Fundamentals of Digital Image Processing – A.K. Jain – Prentice Hall.
- Image Processing Toolbox – Matlab.
- Nhập môn xử lý ảnh số - Lương Mạnh Bá & Nguyễn Thanh Thuỷ - NXB Khoa học kỹ thuật.
- Xử lý ảnh bằng máy tính – Ngô Diên Tập – NXB Khoa học kỹ thuật.

Subject assessment methods

- Lesson preparation: 20%
- Homework: 20%
- Final test: 60%

IMAGE PROCESSING

Chapter 1: Overview of Image Processing

Dr. Cao Thi Luyen
Email: luyenct@utc.edu.vn
Tel: 0912403345

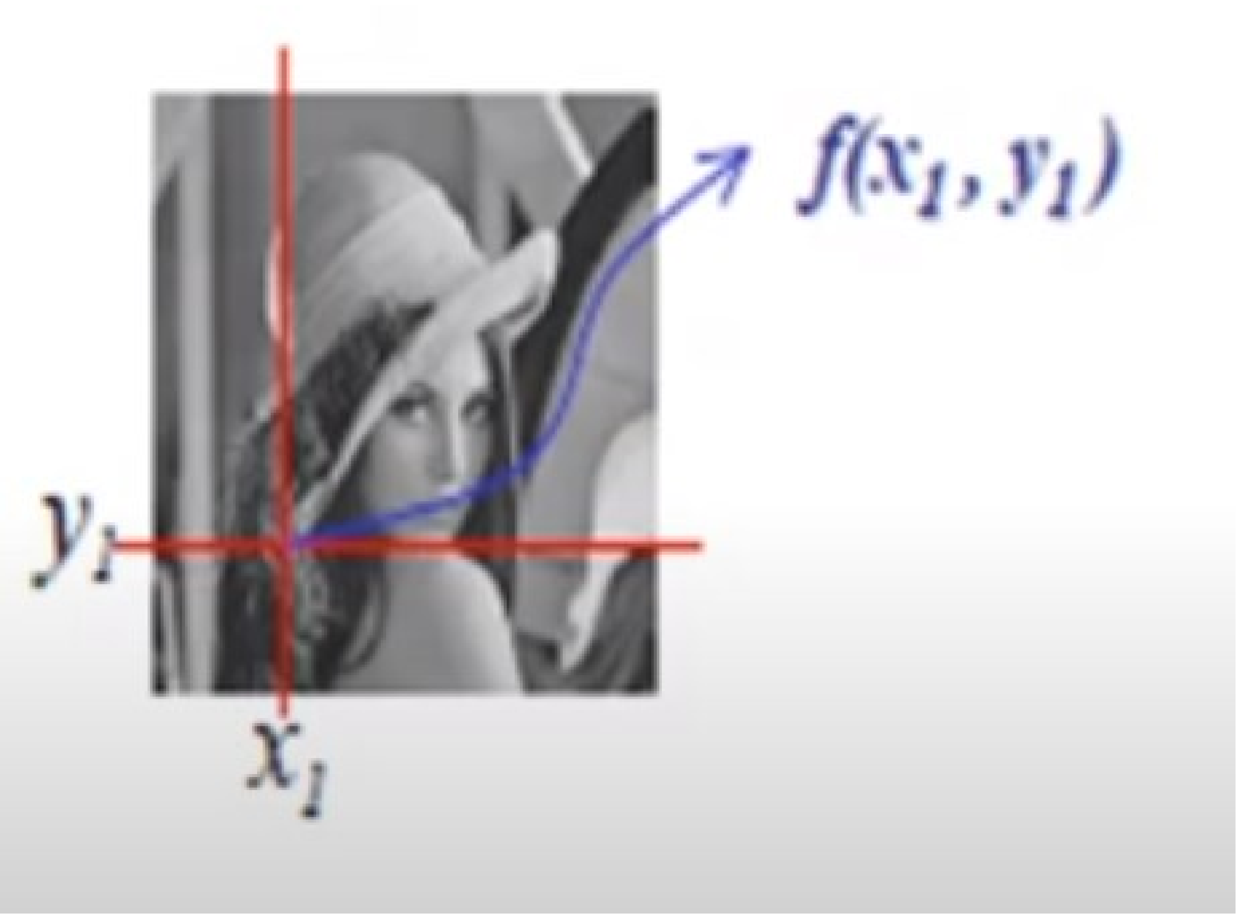
CONTENTS

1. What are images and digital images?





1. What are images and digital images?



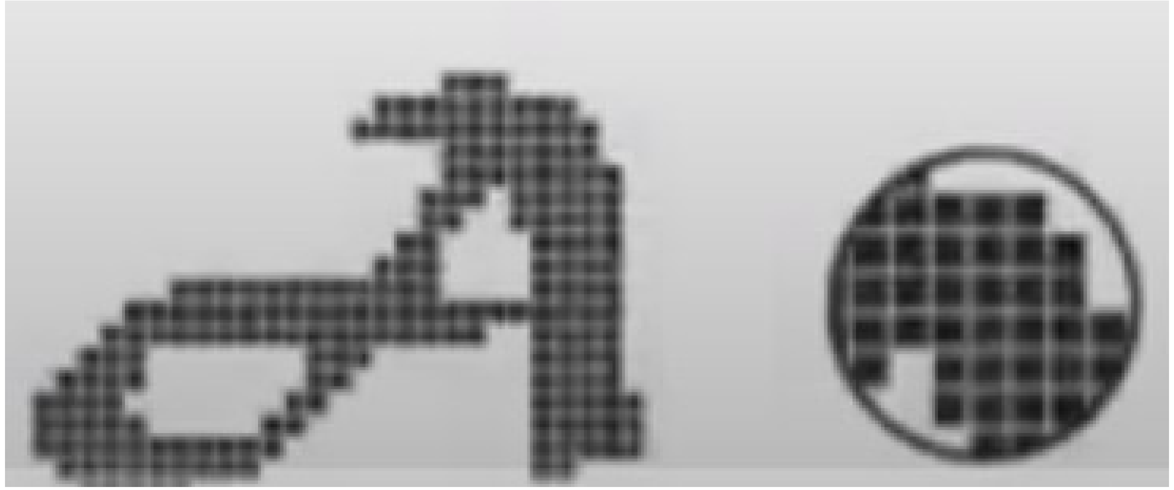
D*igital image* may be defined as a two-dimensional function, *f(x, y)*, where *x* and *y* are *spatial* (plane) coordinates, and the amplitude of *f* at any pair of coordinates (*x, y)* is called the *intensity* or *gray level* of the image at that point. When *x*, *y*, and the intensity values of *f* are all finite, discrete quantities

Core definitons

Pixel is an element of a digital image at coordinates **(x, y)** with a **certain gray level** or color.

The size and distance between those pixels are chosen appropriately so that the human eye perceives the spatial continuity and the gray level (or color) of the digital image is close to the real image.

Each element in the matrix is called a element of image



Core definitons

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

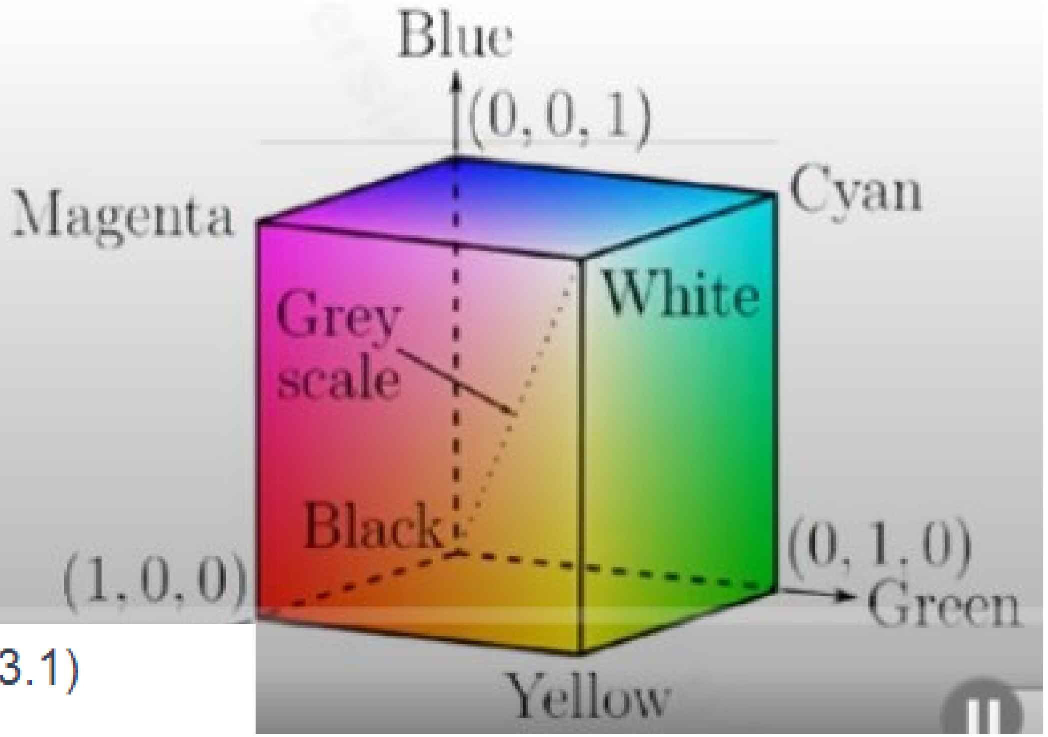
7	9	16	33	208	97	236	198	110	68
11	2	4	24	154	206	246	218	176	82
45	54	11	22	103	208	215	51	70	88
163	123	130	135	156	174	90	18	46	109
231	37	32	41	107	156	22	10	28	151
227	153	17	41	107	151	55	9	14	167
165	18	10	9	19	145	109	12	22	137
167	4	14	11	8	123	132	34	47	194
160	9	10	9	7	98	139	43	40	185
154	3	9	12	11	77	147	66	135	210

Grayscale

The result of mapping a brightness value of a pixel to a positive integer. Usually, it is defined within the range [0, 255], depending on the value represented by each pixel.

Grayscale value : 16, 32, 64, 128, 256 (256 levels is the most common.

Basic color system



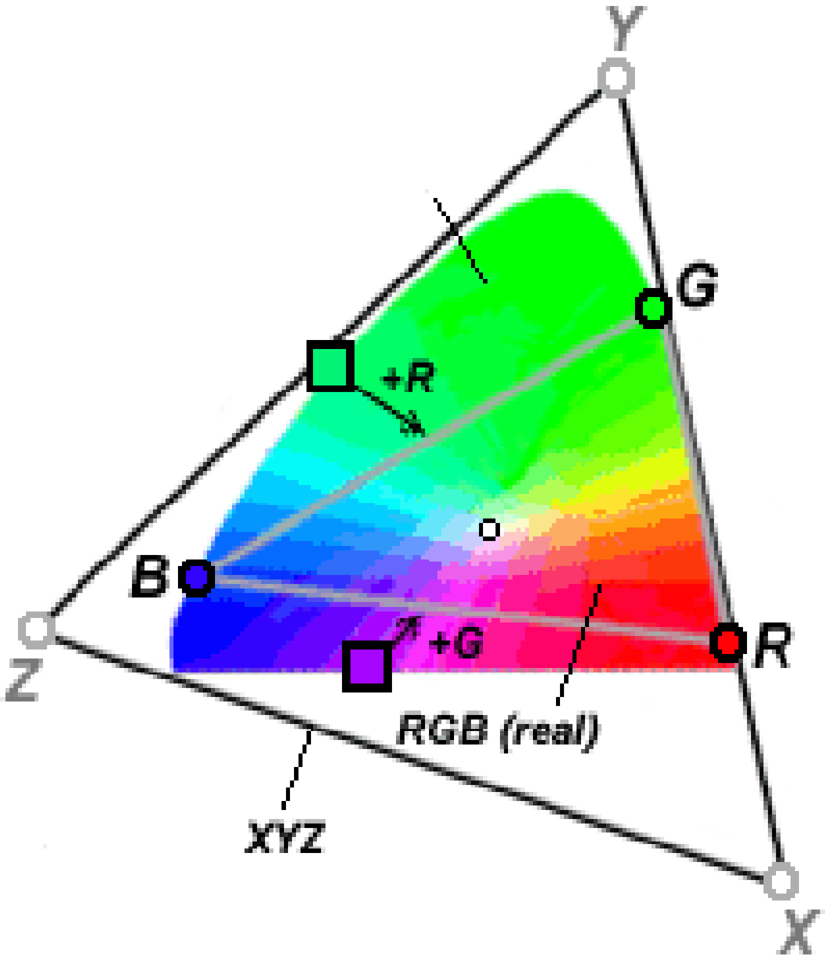
$$Y = 0.299R + 0.587G + 0.114B \quad (3.1)$$

$$Cr = 128 + 0.438R - 0.366G + 0.071B \quad (3.2)$$

$$Cb = 128 - 0.148R - 0.290G + 0.438B \quad (3.3)$$

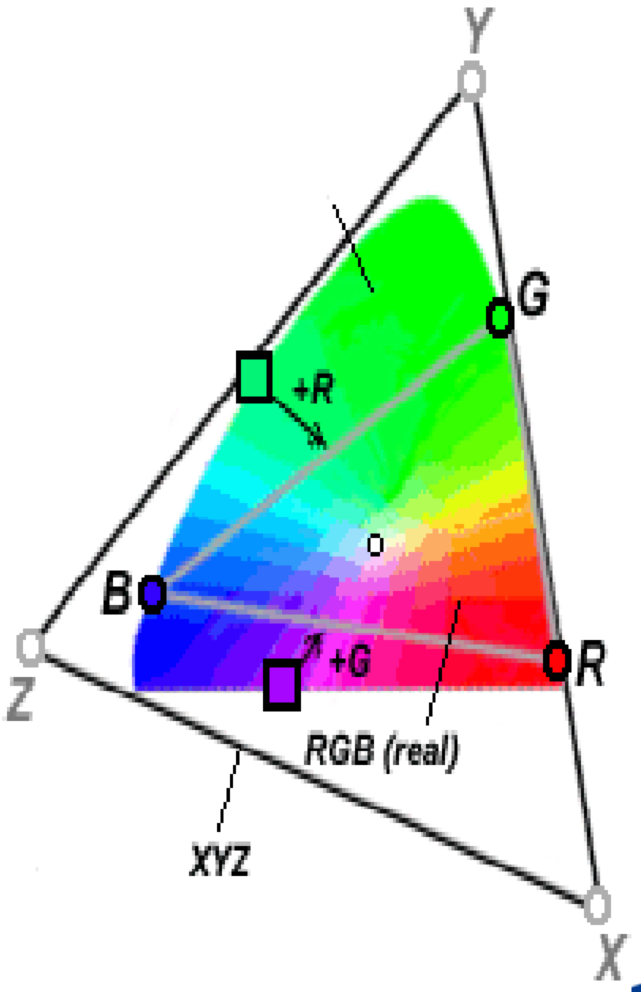
XYZ

- The XYZ color space is defined by a linear transformation of the RGB color space such that all visible spectrum colors lie within the XYZ triangle.
- Advantages and Disadvantages:
- Can represent colors outside the RGB triangle with 100% saturation.
- Points outside the curve are not real (outside the visible spectrum)



Convert RGB into XYZ

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2,769 & 1,7518 & 1,1300 \\ 1,0000 & 4,5907 & 0,0601 \\ 0,0000 & 0,0565 & 5,5943 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$



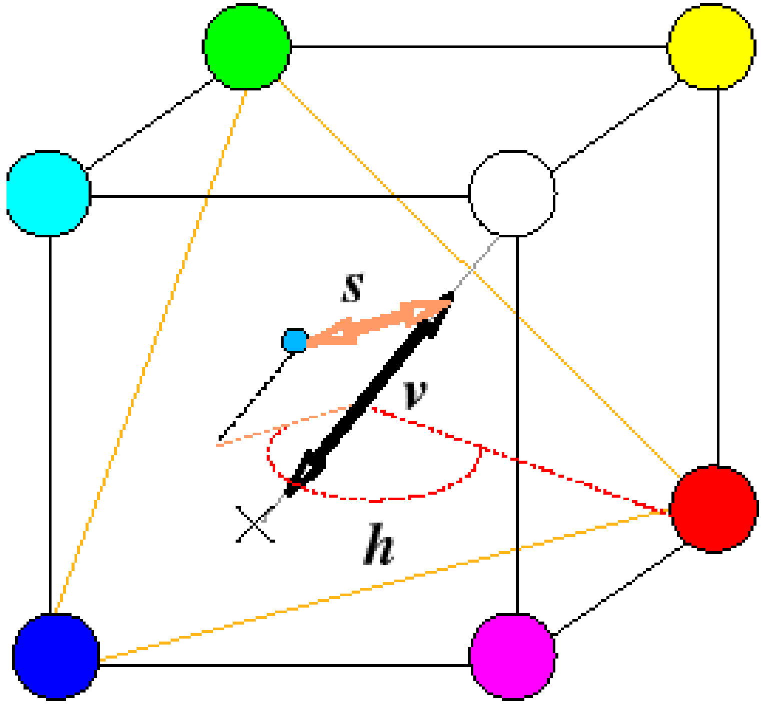
HSV

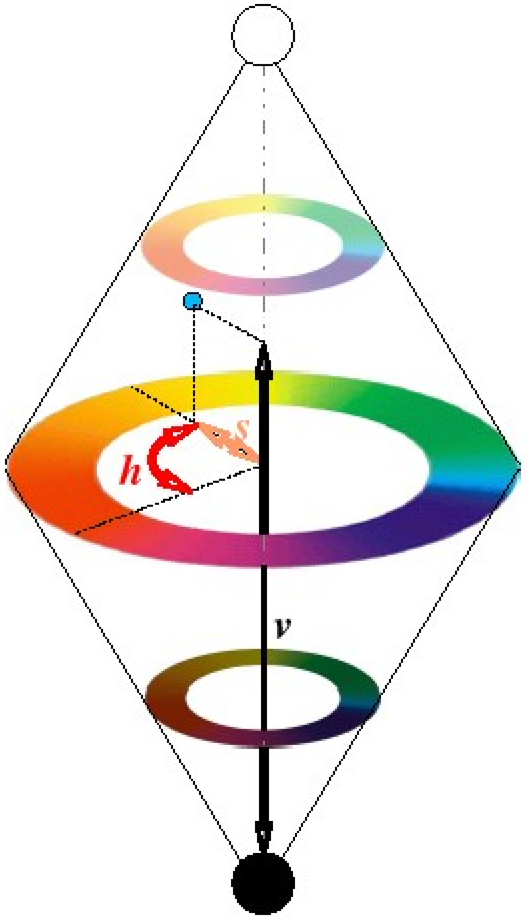
The color coordinate system is built based on three main quantities of light:

•**Hue:** Represents the shade of color (red, yellow, orange, etc.)

•**Saturation:** Represents the intensity of the color (deep red, pale red, etc.)

•**Value (Brightness):** Represents the power of the light source.



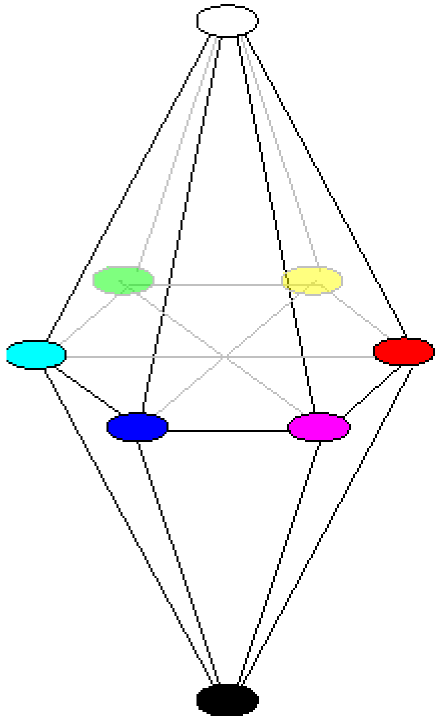


Conver RGB into HSV

$$v = \frac{r+g+b}{3}$$

$$s = 1 - \frac{3\,min(r,g,b)}{r+g+b}$$

$$h = \begin{cases} \theta & si \quad b \leqslant g \\ 2\pi - \theta & si \quad b > g \end{cases} \qquad \theta = \arccos\left(\frac{(r-g)+(r-b)}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right)$$



$$v = \frac{r+g+b}{3}$$

$$s = \begin{cases} \frac{3}{2}(M-v)\, si\ v \geqslant med \\ \frac{3}{2}(v-m)\, si\ v \leqslant med \end{cases} \qquad \begin{matrix} M = max(r,g,b) \\ m = min(r,g,b) \\ med = mediane(r,g,b) \end{matrix}$$

$$h = \frac{\pi}{3}\left(\lambda + \frac{1}{2} - (-1)^{\lambda}\frac{M+m-2\,med}{2v}\right);\ \lambda = \begin{cases} 0\ si\ r \geqslant g \geqslant b;\ 1\ si\ g \geqslant r \geqslant b \\ 2\ si\ g \geqslant b \geqslant r;\ 3\ si\ b \geqslant g \geqslant r \\ 4\ si\ b \geqslant r \geqslant g;\ 5\ si\ r \geqslant b \geqslant g \end{cases}$$

Biểu diễn ảnh trên HSV

Ảnh gốc và ảnh biểu diễn theo độ sáng

Biểu diễn ảnh trên HSV

Ảnh biểu diễn theo độ bão hòa và sắc thái

Color depth

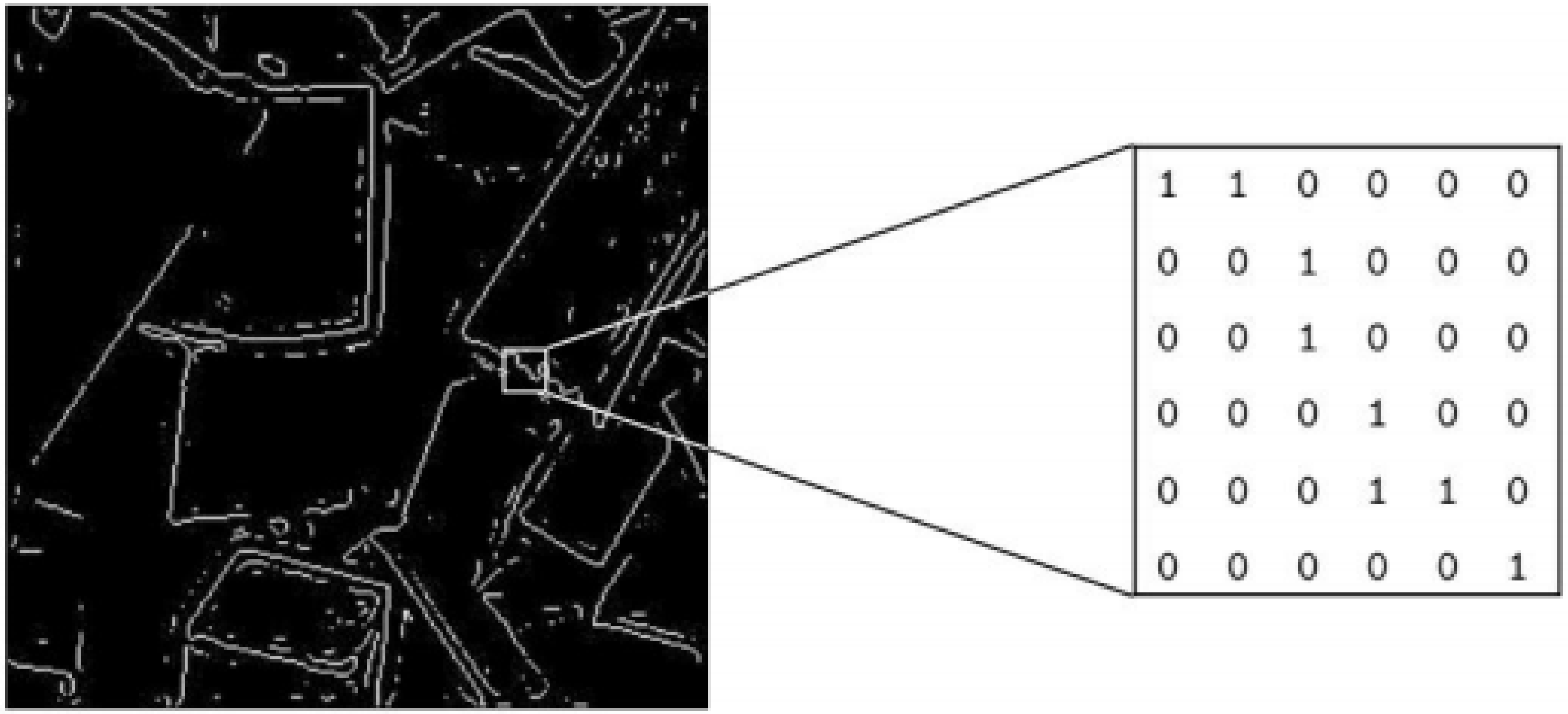
Color depth is an indicator that **measures the ability** to represent different colors of a pixel in a digital image.
Unit of measurement: The number of bits required to encode the color of a pixel.
Classification:
- Binary Image (1 bit/pixel): Only two colors, black and white.
- Grayscale Image (8 bit/pixel): 256 shades of gray, from black to white.
- Color Image (24 or 32 bit/pixel):
  - 24 bit/pixel: Each pixel is made up of 3 primary colors (Red, Green, Blue), with 8 bits per color, allowing for 16,777,216 different colors.
  - 32 bit/pixel: Adds an additional 8 bits to represent pixel transparency, often used in computer graphics."

Image resolution

- defined as the number of pixels in a given area of the image. In other words, it represents the level of detail in an image.
- **Unit of measurement:** number of pixels horizontally multiplied by the number of pixels vertically (e.g., 1920x1080 pixels).
- **Illustrative example**
- 4K resolution is usually 3840 x 2160 pixels. This means the image will have 3840 pixels horizontally and 2160 pixels vertically
- **Impact of resolution on image quality**
- **Sharpness:** The higher the resolution, the sharper the image, and the clearer the details.
- **Image size:** For the same resolution, an image displayed on a larger screen will appear less sharp than on a smaller screen.
- **File size:** High-resolution images typically have larger file sizes.

Image Classification

Binary image



Gray image

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Color Image

55	56	57	52	53
60	60	58	55	57
58	54	53	55	56
78	72	69	68	69
91	91	84	83	82

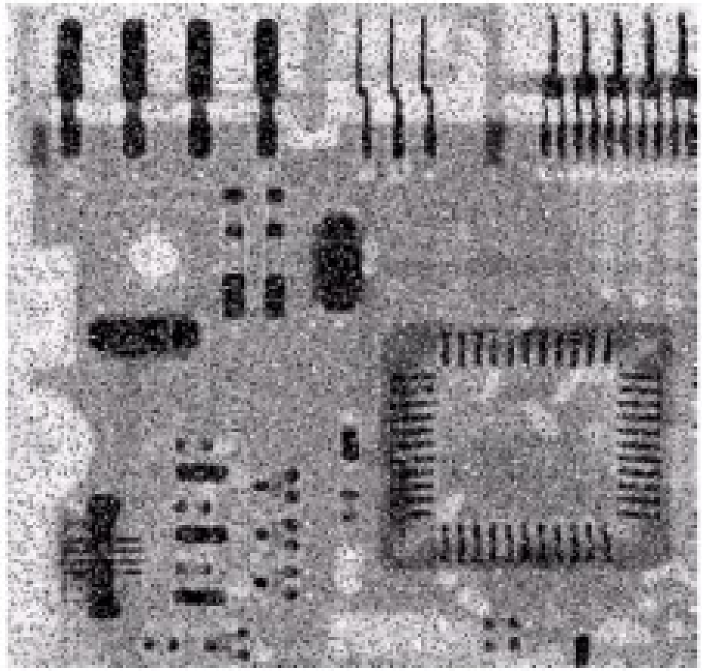
76	82	79	78	78
93	91	91	86	86
82	88	90	88	89
19	113	108	111	110
36	132	128	126	120

80	77	80	87	77
93	96	99	86	85
83	91	94	92	88
128	126	112	107	106
129	129	117	115	101

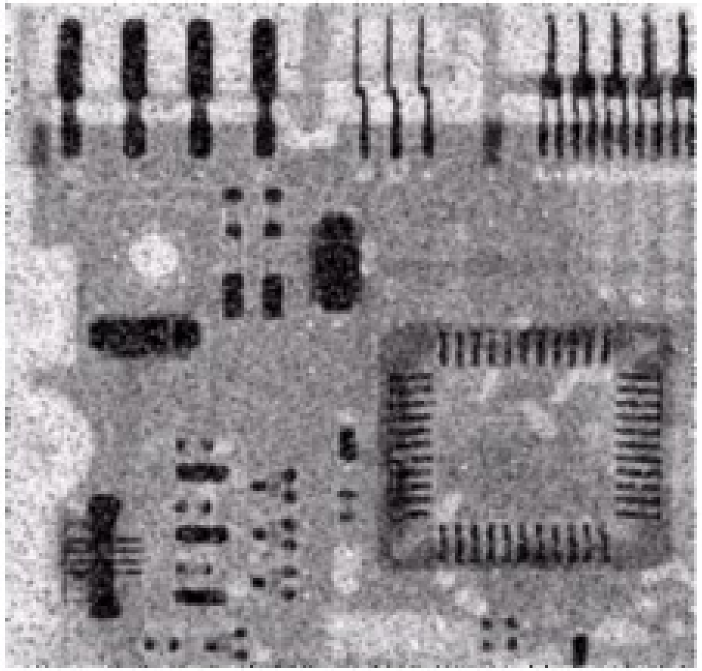
3. Image processing

- Image processing is a field of computer science and engineering that involves manipulating and analyzing digital images using algorithms and techniques to enhance or extract information from them.
- Digital image processing focuses on two objectives:
- 1. Enhancing image information to improve human perception.
- 2. Processing image data for storage, transmission, and display in machines

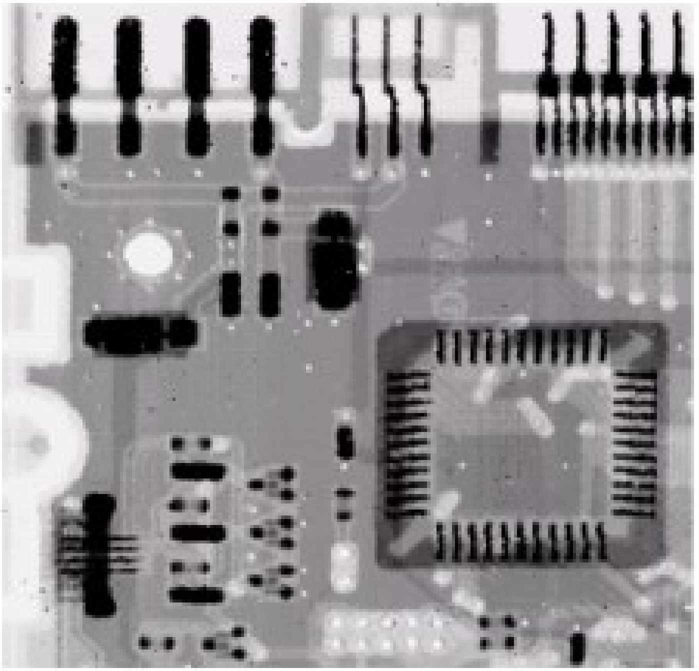
Examples and image processing.



Ảnh gốc có nhiễu



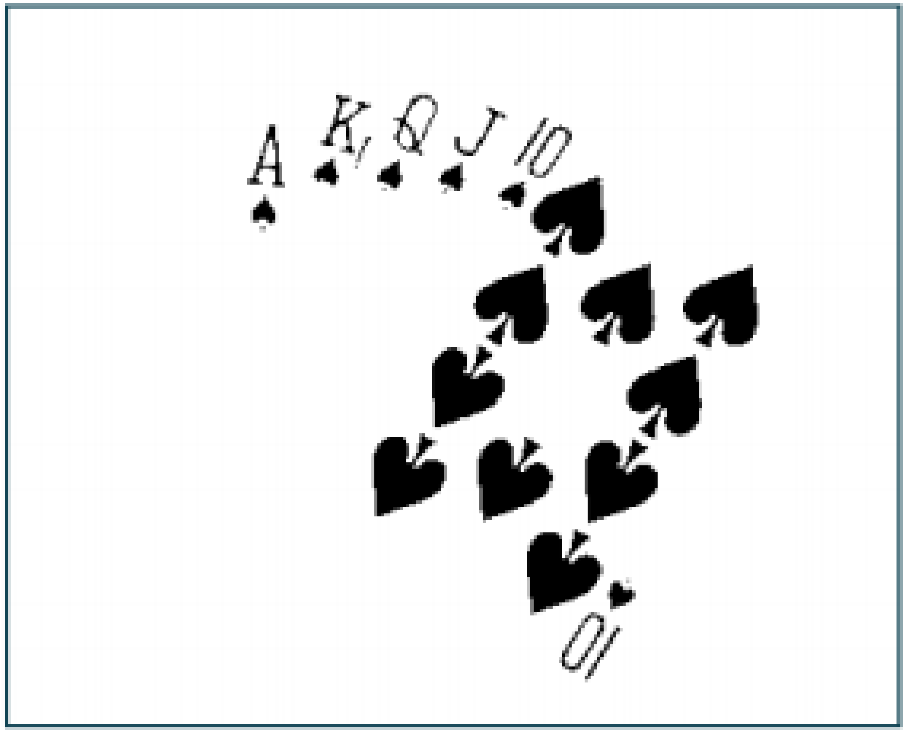
Ảnh sau khi lọc trung bình



Ảnh sau khi lọc trung vị

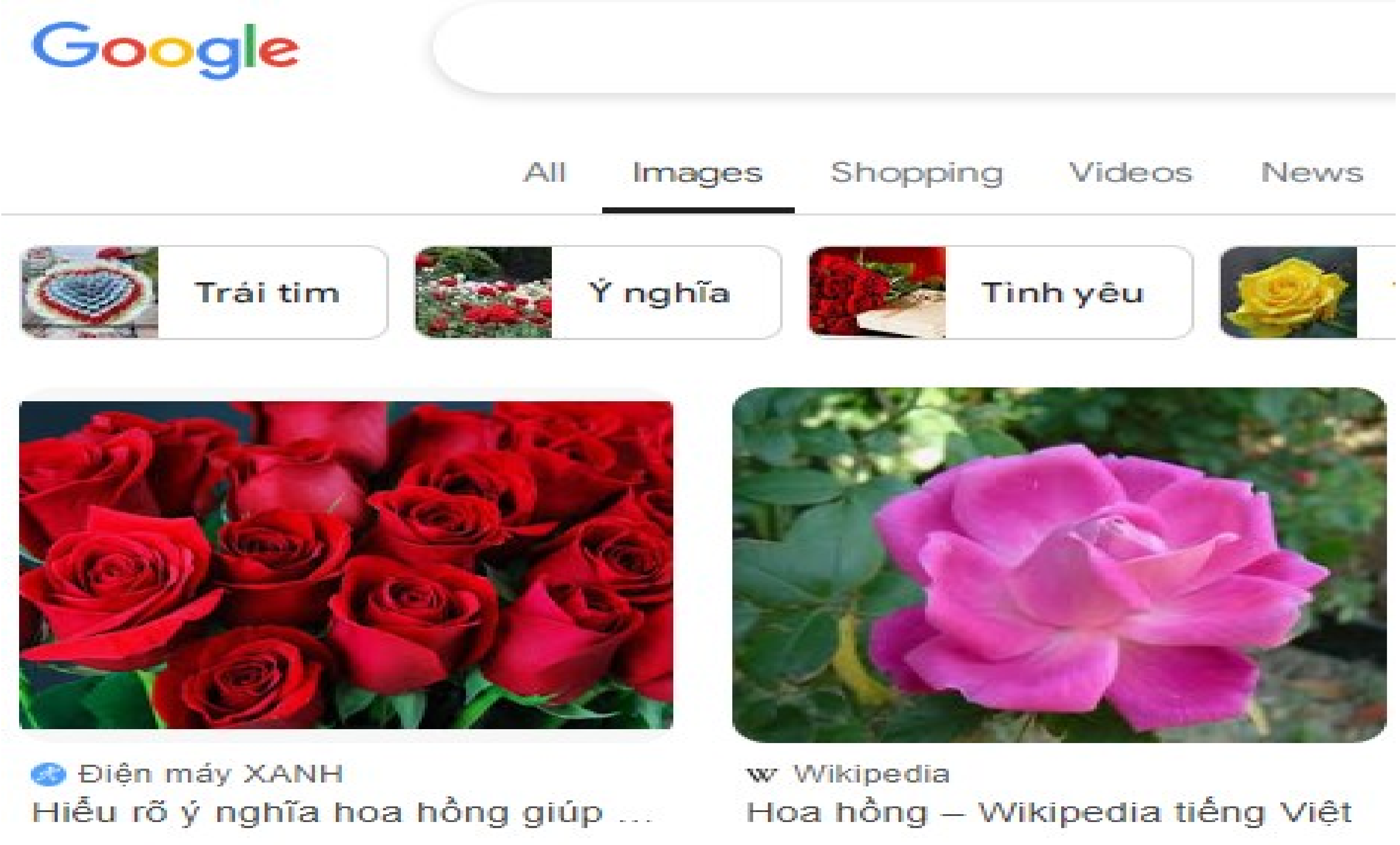
Examples and image processing.

Ảnh gốc



Ảnh sau phân ngưỡng

Examples and image processing.



3. Application

- **Medical field**:
- **Satellite and Aerial Imaging**:
- **Optical Character Recognition (OCR)**:
- **Surveillance and Security**:
- **Entertainment and Multimedia**:
- **Industrial Inspection**:
- **Forensic Science**:
- **Scientific Research**:
- **Transportation**:

image processing in matlab

- Read and show image:
  - I = imread('Lenna.jpg');
  - imshow(I);
  - figure, imhist(I);

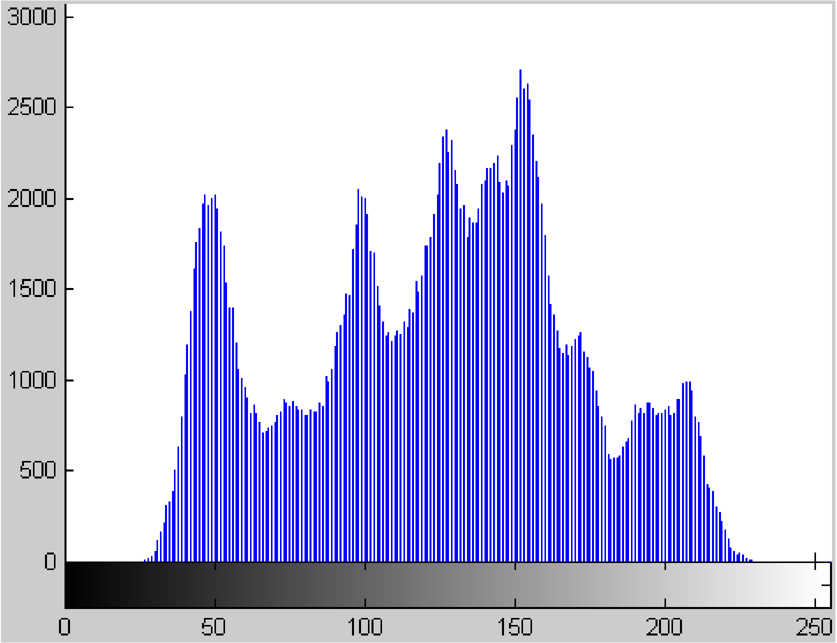


image processing in matlab

- Convert format:
  - I = imread('Lenna.jpg');
  - imwrite(I, 'Lenna.tiff');
- resize:
  - J = imresize(I, 1.5);
  - imshow(J);
- Fourrier transform:
  - F = fft2(I);
  - imshow(F);

image processing in matlab

- Eadge detection
  - I = imread('Lenna.jpg');
  - J = edge(I, 'sobel');
  - imshow (J);

image processing in matlab

- Add noise:
  - I = imread('Lenna.jpg');
  - N = imnoise(I,'salt & pepper',0.02);
  - imshow (N);

image processing in matlab

- Remove noising:
  - N = imnoise(I,'salt & pepper',0.02);
  - K = filter2(fspecial('average',3),N)/255;
  - imshow (K);

Basic steps of an image processing system

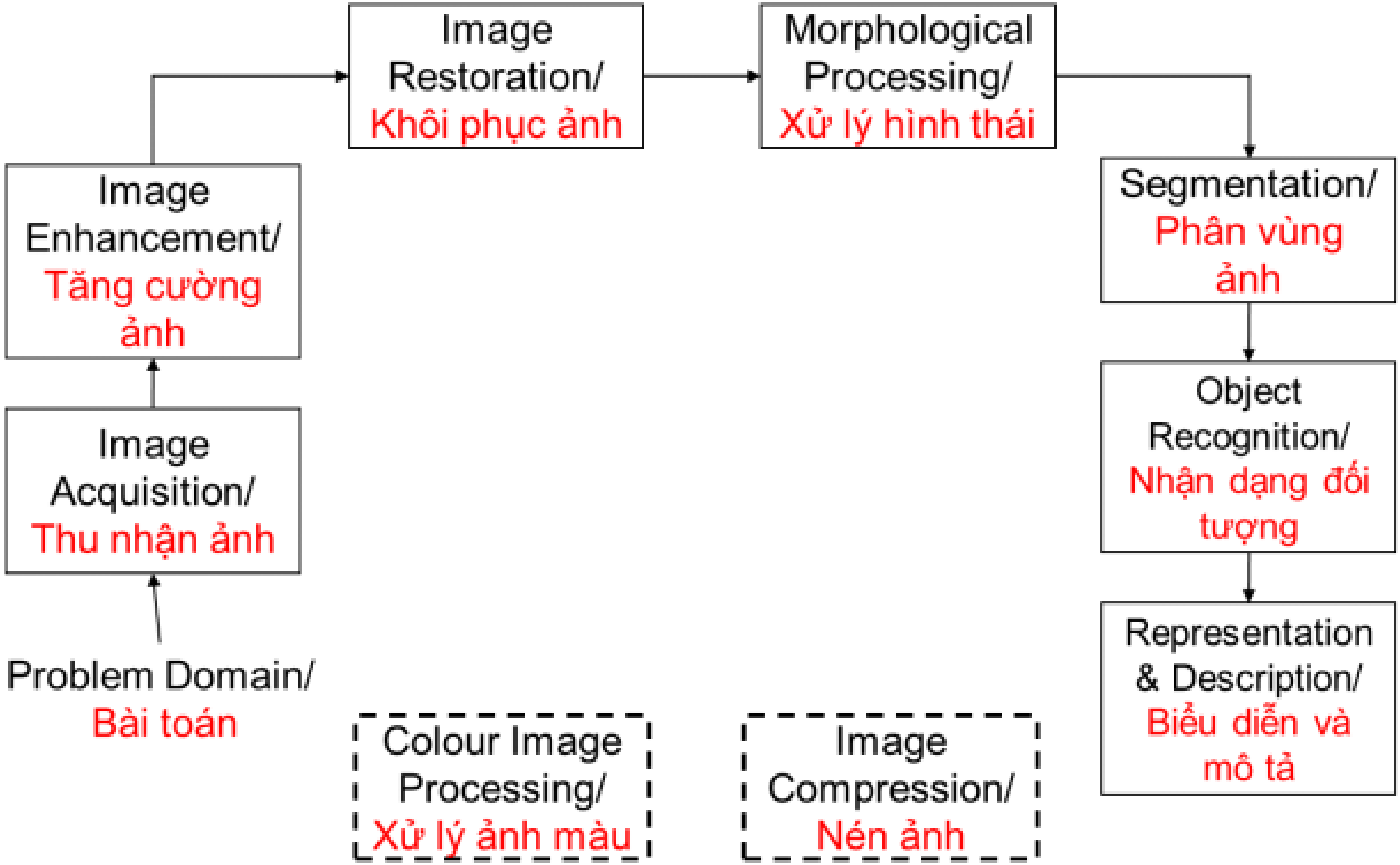
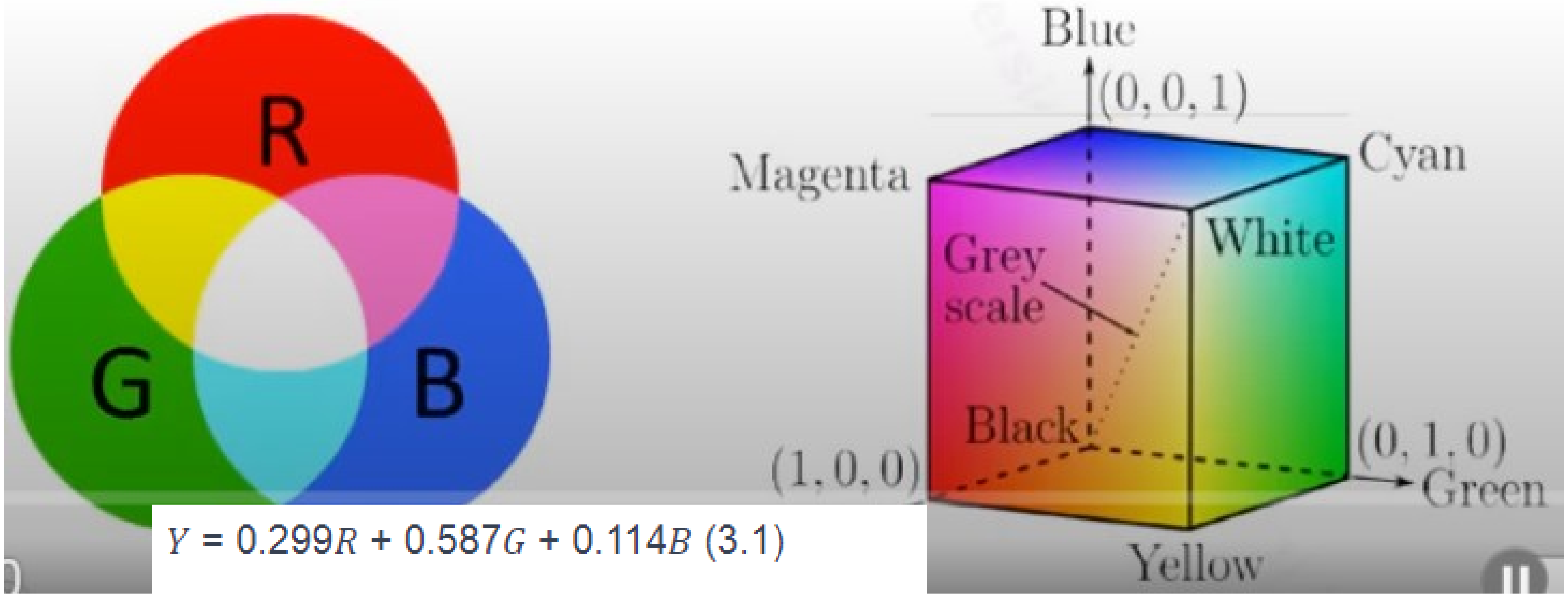


IMAGE PROCESSING

Chapter 2:Thu nhận và biểu diễn ảnh
Image Acquisition; image representation description

Dr. Cao Thi Luyen
Email: luyenct@utc.edu.vn
Tel: 0912403345

Basic color system



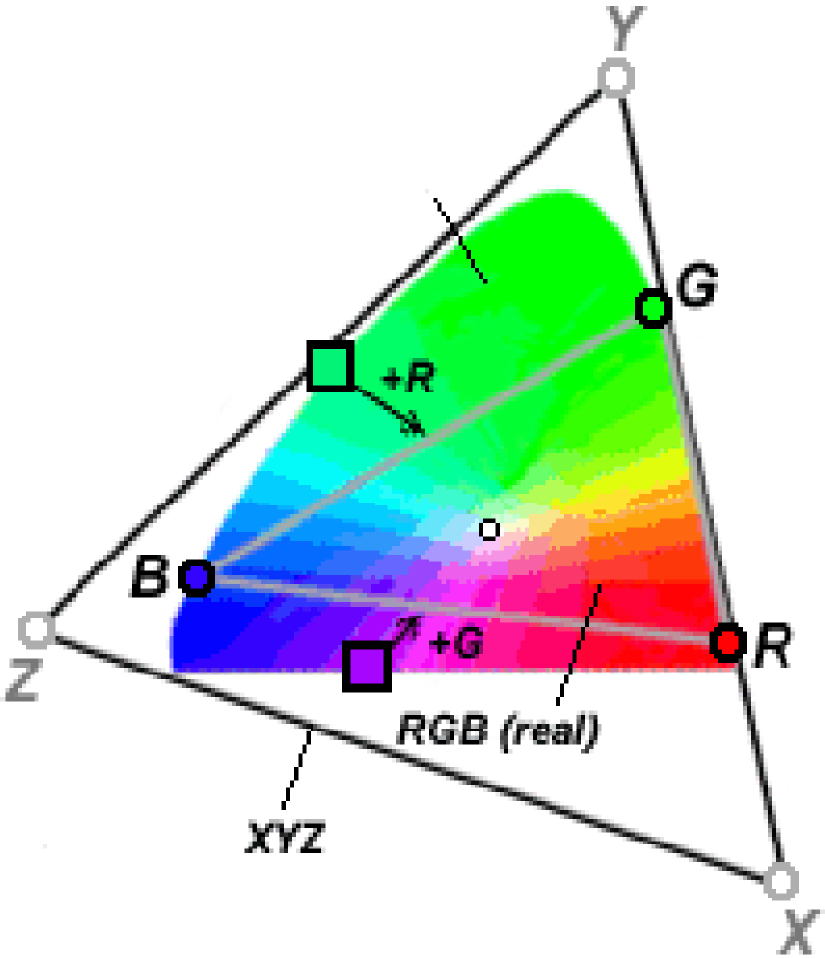
$$Y = 0.299R + 0.587G + 0.114B \quad (3.1)$$

$$Cr = 128 + 0.438R - 0.366G + 0.071B \quad (3.2)$$

$$Cb = 128 - 0.148R - 0.290G + 0.438B \quad (3.3)$$

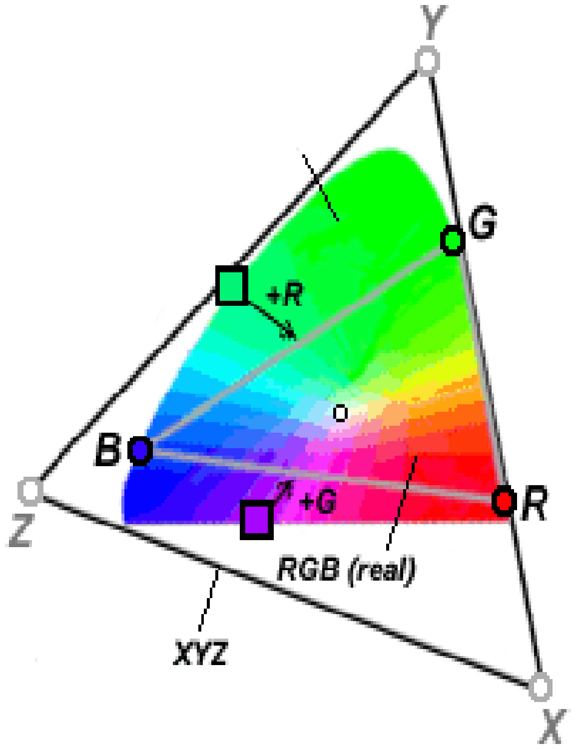
XYZ

- The XYZ color space is defined by a linear transformation of the RGB color space such that all visible spectrum colors lie within the XYZ triangle.
- Advantages and Disadvantages:
- Can represent colors outside the RGB triangle with 100% saturation.
- Points outside the curve are not real (outside the visible spectrum)



Convert RGB into XYZ hsv=rgb2xyz(image)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2,769 & 1,7518 & 1,1300 \\ 1,0000 & 4,5907 & 0,0601 \\ 0,0000 & 0,0565 & 5,5943 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$



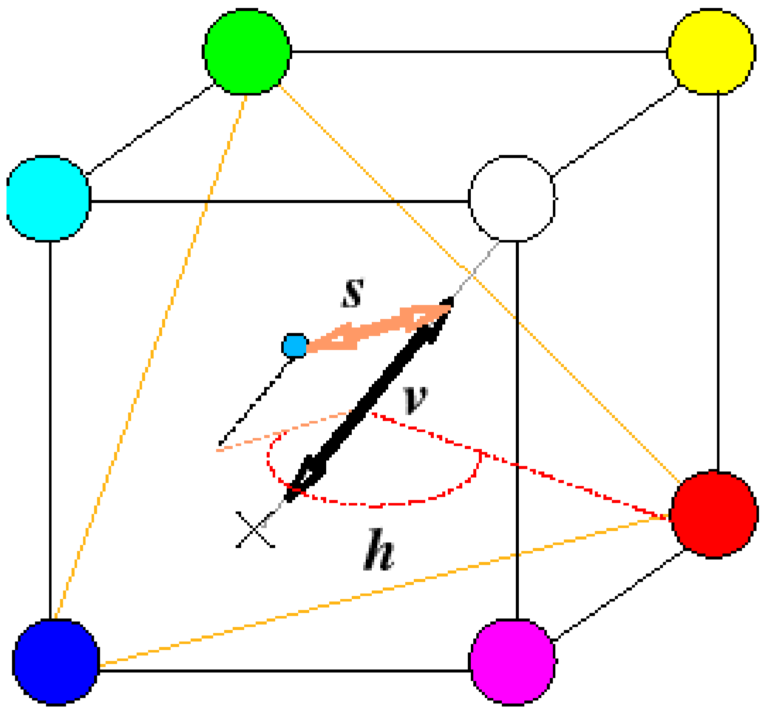
HSV

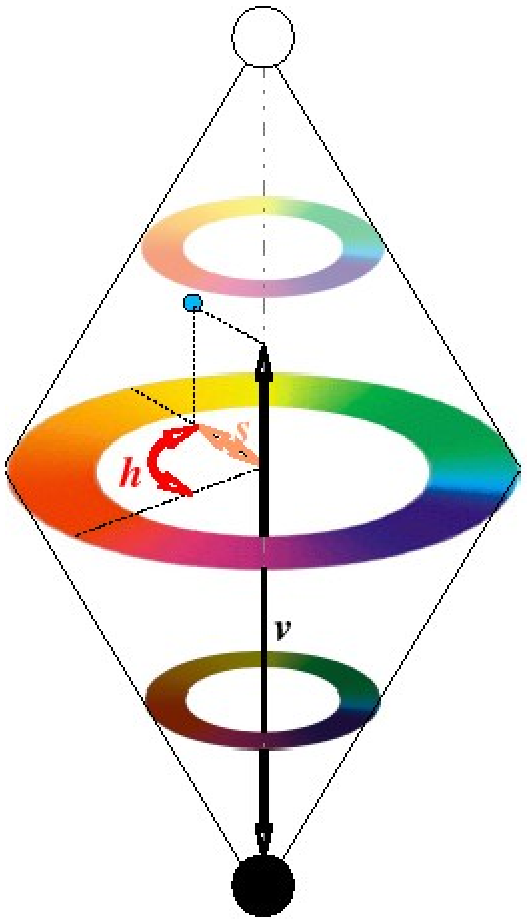
The color coordinate system is built based on three main quantities of light:

•**Hue:** Represents the shade of color (red, yellow, orange, etc.)

•**Saturation:** Represents the intensity of the color (deep red, pale red, etc.)

•**Value (Brightness):** Represents the power of the light source.



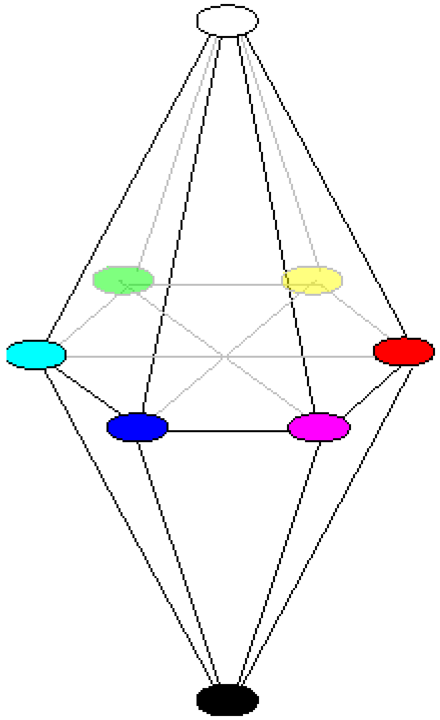


Conver RGB into HSV

$$v = \frac{r+g+b}{3}$$

$$s = 1 - \frac{3\,min(r,g,b)}{r+g+b}$$

$$h = \begin{cases} \theta & si \quad b \leqslant g \\ 2\pi - \theta & si \quad b > g \end{cases} \qquad \theta = \arccos\left(\frac{(r-g)+(r-b)}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right)$$



$$v = \frac{r+g+b}{3}$$

$$s = \begin{cases} \frac{3}{2}(M-v)\; si\; v \geqslant med \\ \frac{3}{2}(v-m)\; si\; v \leqslant med \end{cases} \qquad \begin{aligned} M &= max(r,g,b) \\ m &= min(r,g,b) \\ med &= mediane(r,g,b) \end{aligned}$$

$$h = \frac{\pi}{3}\left(\lambda + \frac{1}{2} - (-1)^{\lambda}\frac{M+m-2\,med}{2v}\right); \lambda = \begin{cases} 0\; si\; r \geqslant g \geqslant b;\; 1\; si\; g \geqslant r \geqslant b \\ 2\; si\; g \geqslant b \geqslant r;\; 3\; si\; b \geqslant g \geqslant r \\ 4\; si\; b \geqslant r \geqslant g;\; 5\; si\; r \geqslant b \geqslant g \end{cases}$$

HSV hsv=rgb2hsv(image)

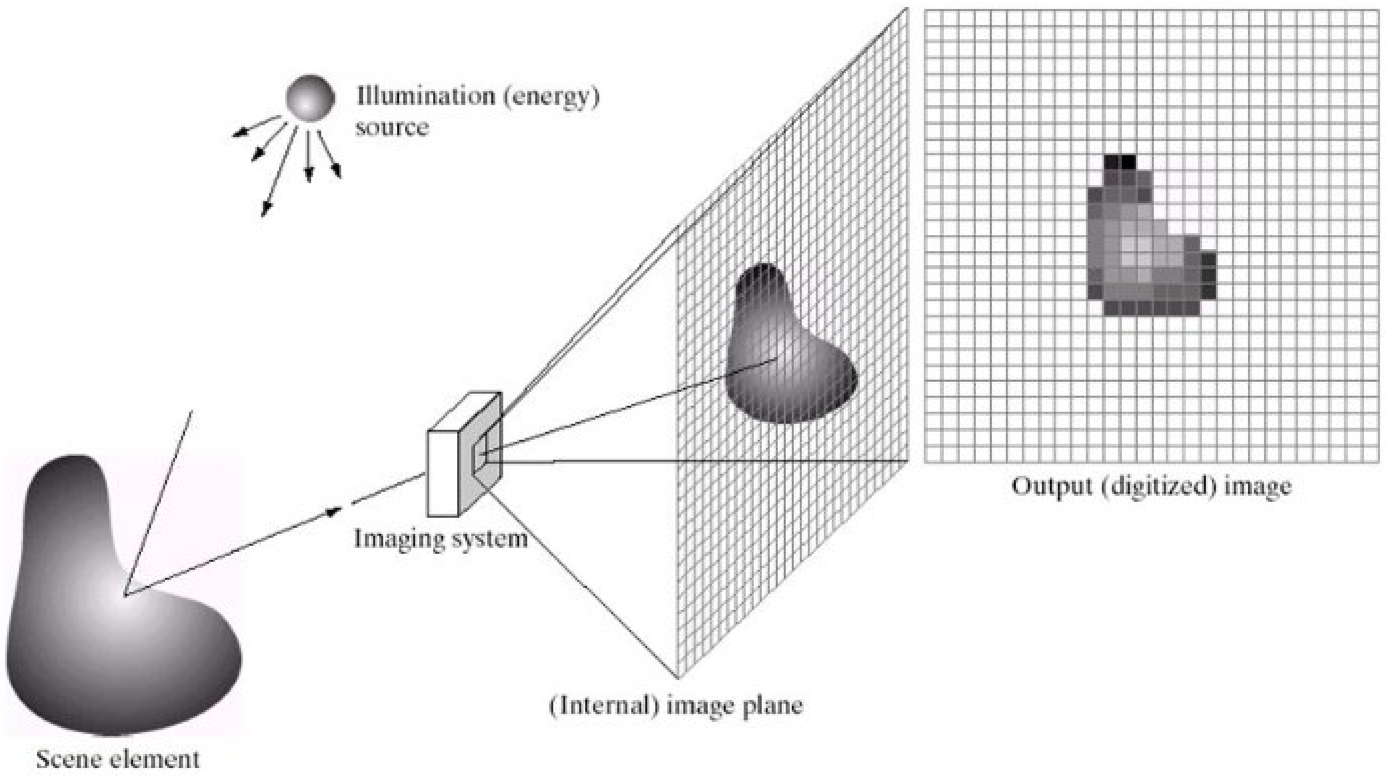
Ảnh gốc và ảnh biểu diễn theo độ sáng

HSV

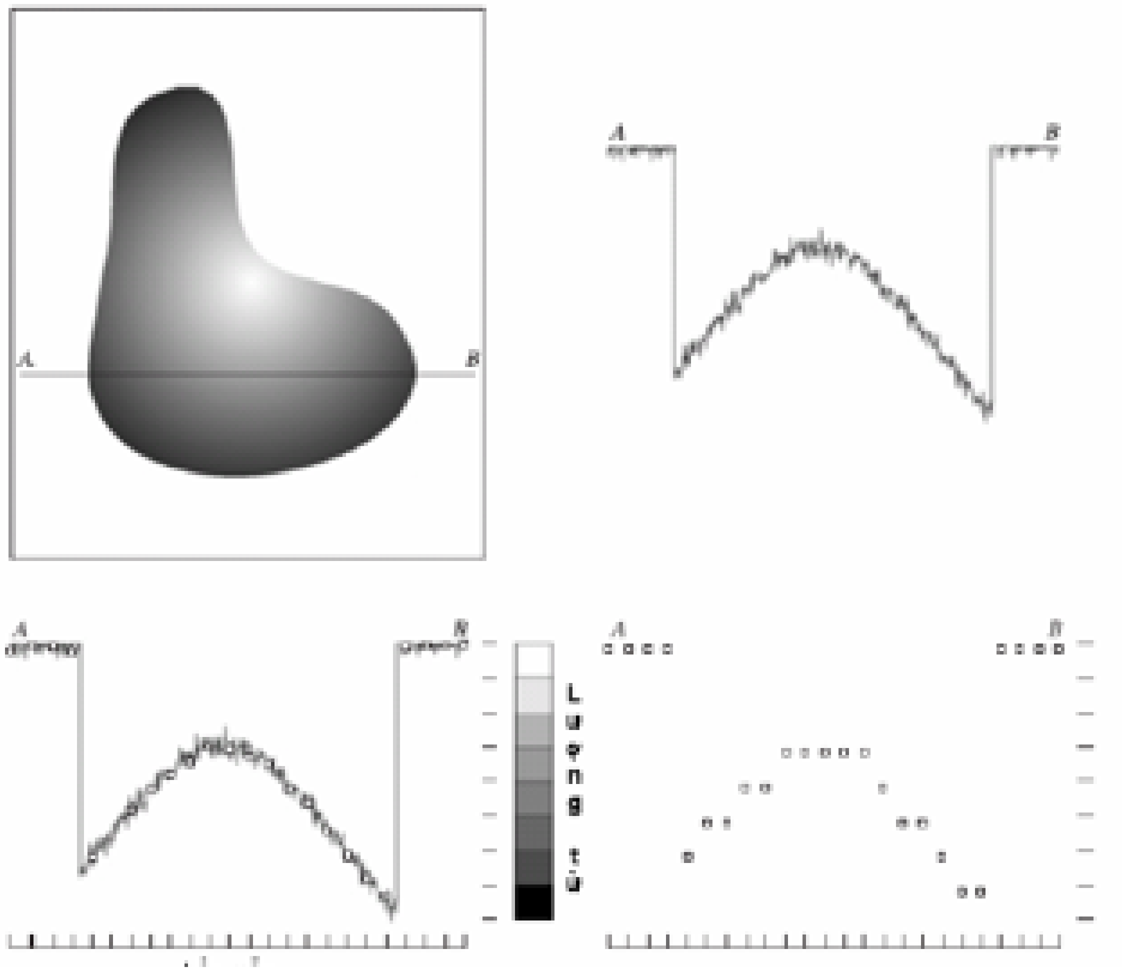
Ảnh biểu diễn theo độ bão hòa và sắc thái

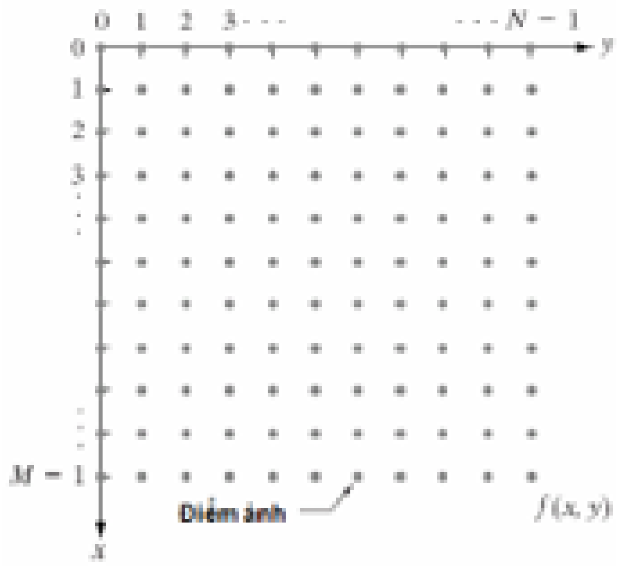
Image Acquisition

- **Data Sources:**
  - Digital cameras
  - Scanners
  - Remote sensing systems (e.g., satellites, drones)
- **Key Considerations:**
  - **Resolution:** Ensure the resolution is high enough to retain details during processing.
  - **Image Quality:** Eliminate images that are blurred, noisy, or have technical defects.



Sampling

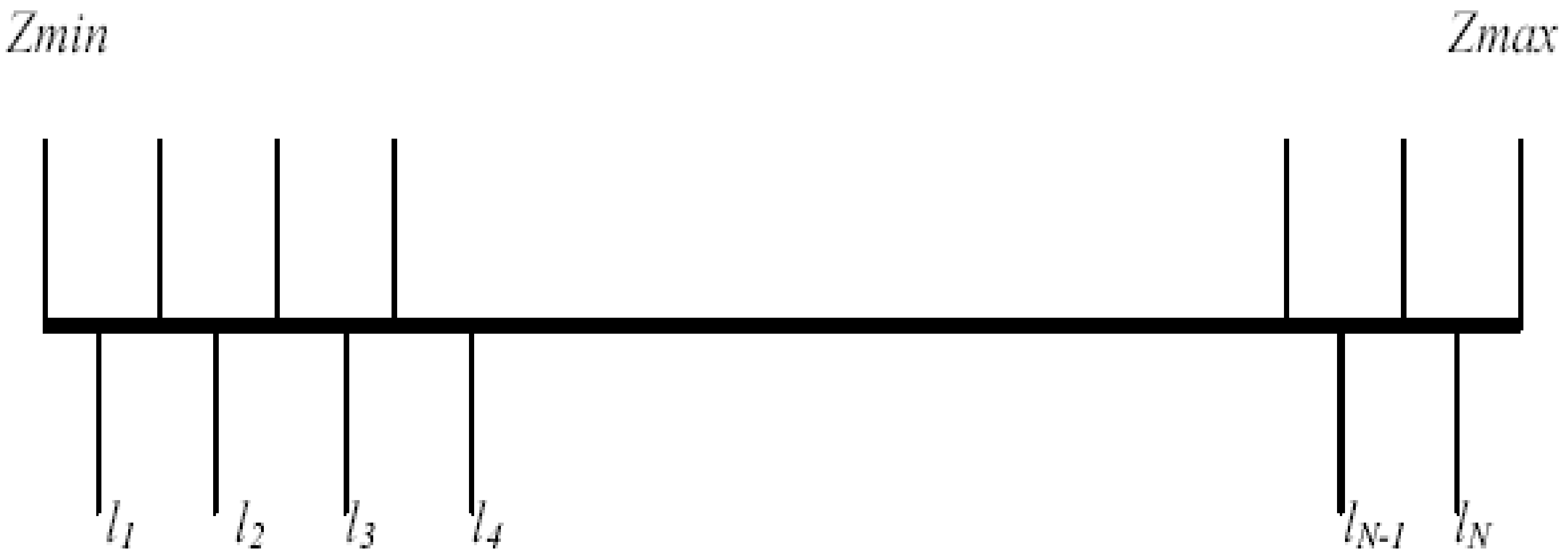




Quantization

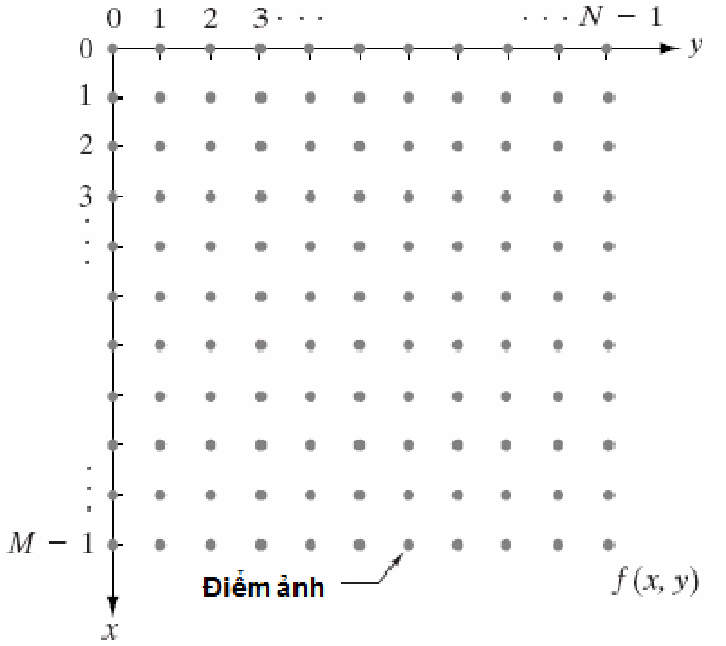
$$t = \frac{Z_{max} - Z_{min}}{N}$$

$$index_Z = \left\lfloor \frac{Z - Z_{min}}{t} \right\rfloor \Rightarrow l_{index_Z}$$



Digital Image Representation

- Matrix: Each pixel corresponds to an element of the matrix with a corresponding color value.



$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \vdots & \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Digital Image Representation

- **Run-Length Encoding Representation:** binary images
- Each run: start address of the run and the length of the run.

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

- Row 1: 1 1 (run of 2 ones), 0 0 0 0 (run of 4 zeros), 1 1 (run of 2 ones)Row 2: 1 (run of 1 one), 0 0 (run of 2 zeros), 1 1 (run of 2 ones), 0 0 (run of 2 zeros), 1 (run of 1 one)
- The RLE for the above matrix can be represented as:Row 1: (1,2), (0,4), (1,2)Row 2: (1,1), (0,2), (1,2), (0,2), (1,1)

Digital Image Representation

- **Run-Length Encoding Representation:** binary images
- Each run: start address of the run and the length of the run.

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

- Row 1: 1 1 (run of 2 ones), 0 0 0 0 (run of 4 zeros), 1 1 (run of 2 ones)Row 2: 1 (run of 1 one), 0 0 (run of 2 zeros), 1 1 (run of 2 ones), 0 0 (run of 2 zeros), 1 (run of 1 one)
- The RLE for the above matrix can be represented as:Row 1: (1,2), (0,4), (1,2)Row 2: (1,1), (0,2), (1,2), (0,2), (1,1)

```
function [bincode] = RLC(image)
i=imread(image);%binary image
bi=im2bw(i,0.3);imshow(i);
figure; imshow(bi);
[m,n]=size(bi);
X=m*n;
data=reshape(bi,1,X);
bincode='';
count=1;j=1;i=2;
while i<=X
    while data(i)==data(j)  end
        count=count+1;
```

```
        i=i+1;
        if i>X
            break;
        end
    end
    bincode=[bincode,num2str(data(j)),num2str(count)];
    count=0;j=i;
```

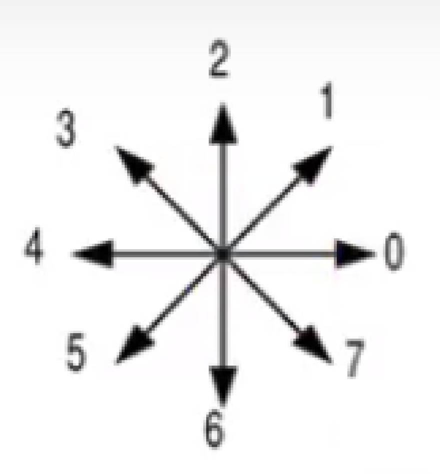
Digital Image Representation: chain code

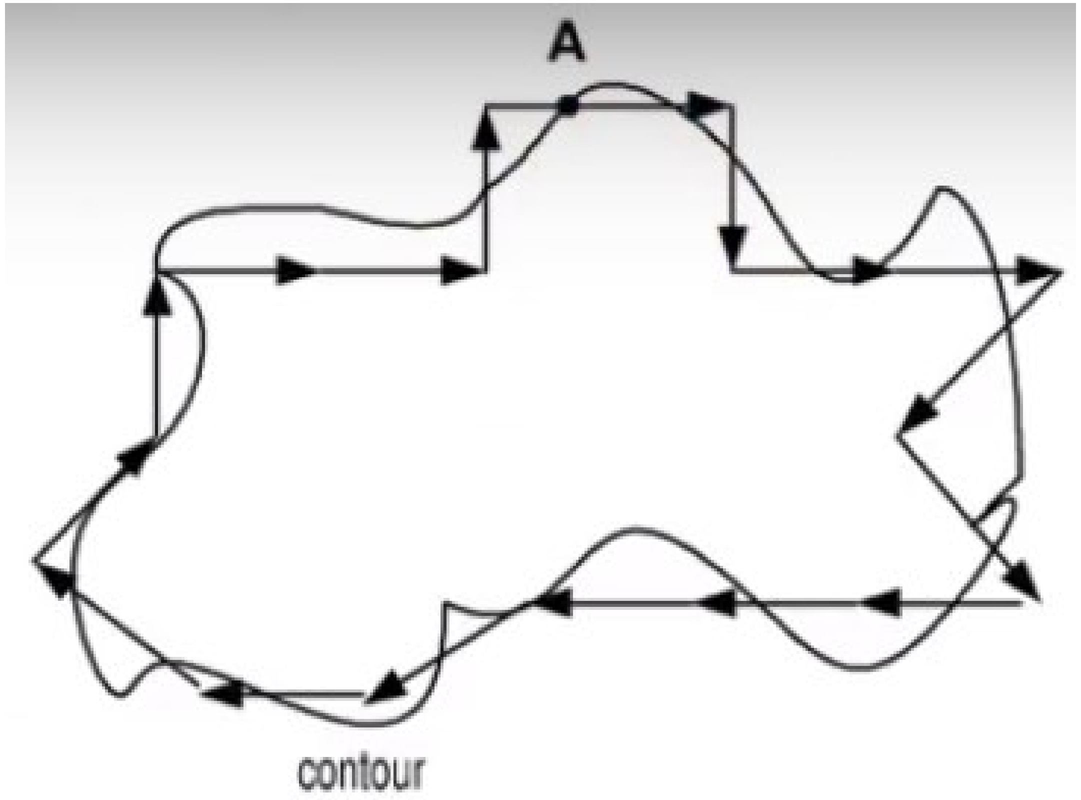
Algorithm:

1. Start at any boundary pixel A,

2. Find the nearest edge pixel and code its orientation. in case of a tie chose the one with largest (or smallest) code value.

3. Continue until there are no more boundary pixels.

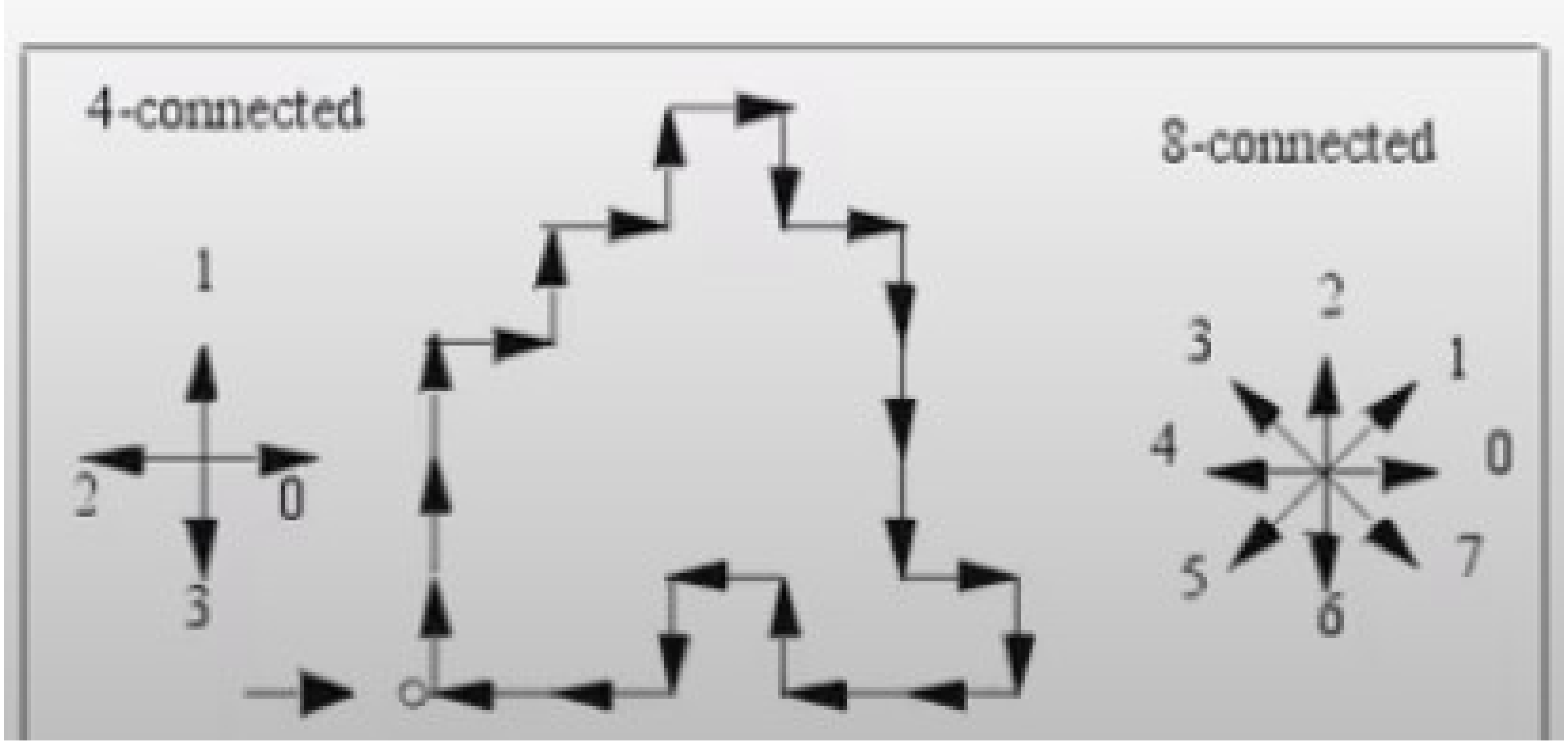




Boundary pixel orientations: (A), 060057444543120020

Chain code: A 000 110 000 000 101 111 100 100 100 101 100 011 001 010 000 000 010 000

Finding the chain code

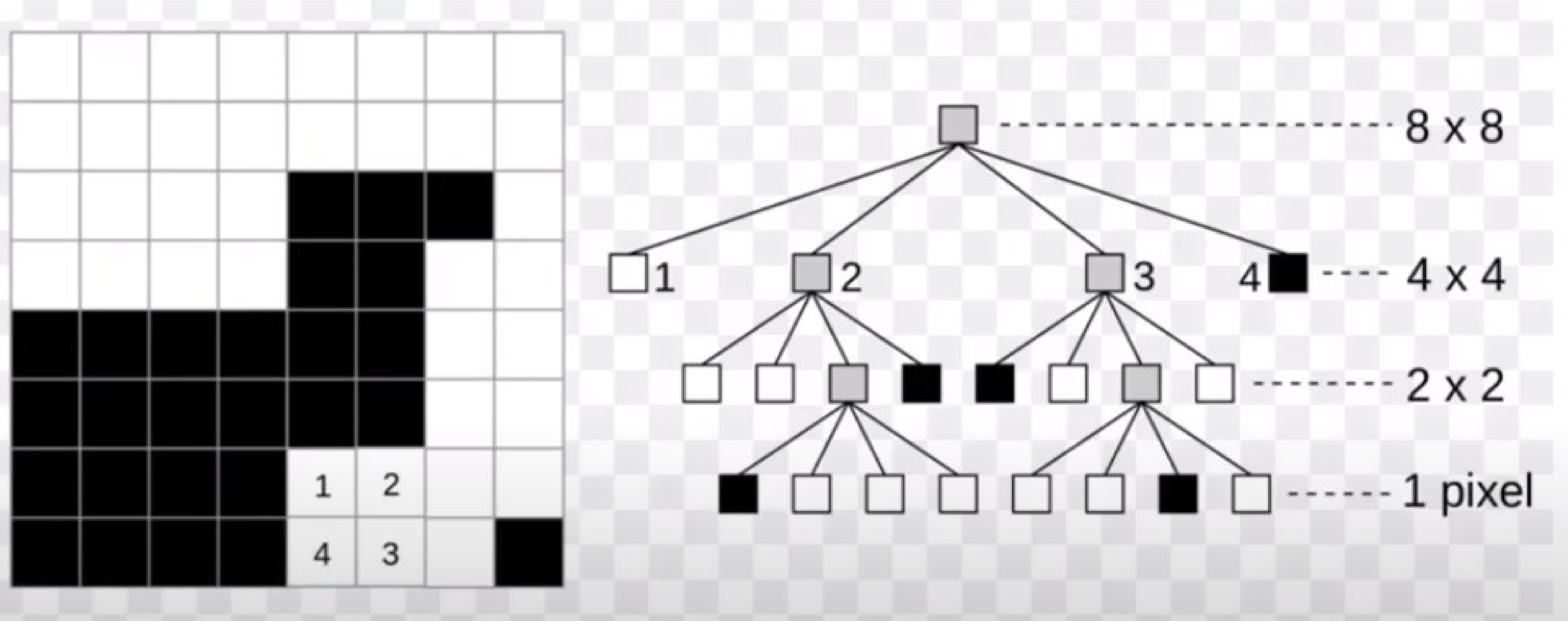


Digital Image Representation

- **Quad-tree code**
- Represent binary image blocks.
- Quadtree Representation Method: Starting from the initial image (m*n), the image is divided into 4 regions in same size. If the region is entirely black (or white), it does not need to be further divided. If not, continue dividing into sub-regions.

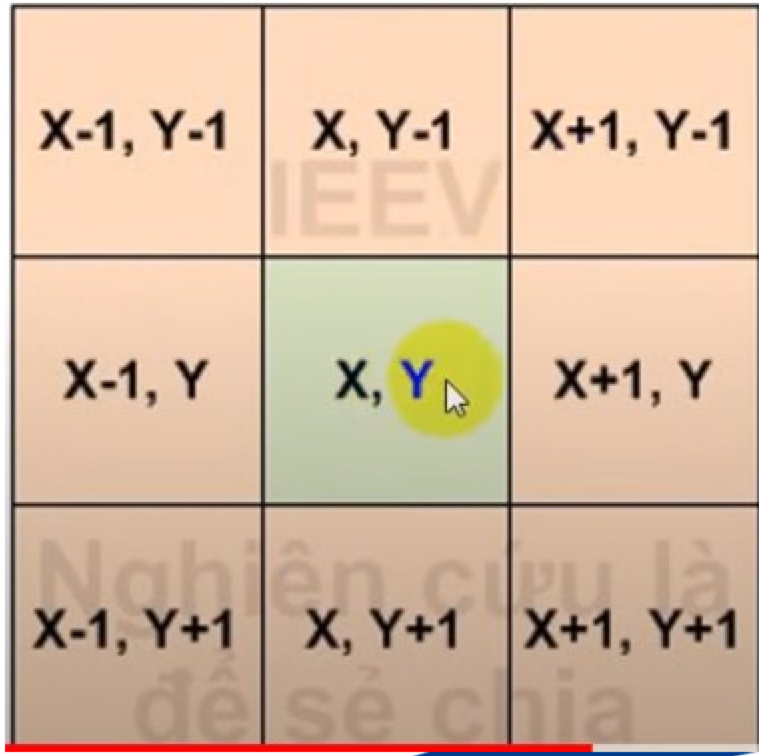
Digital Image Representation

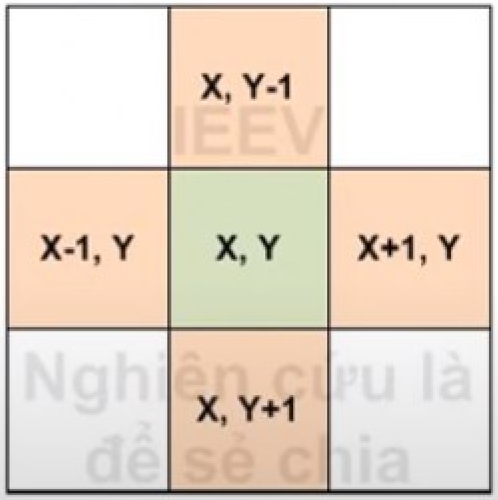
- Quad-tree code example

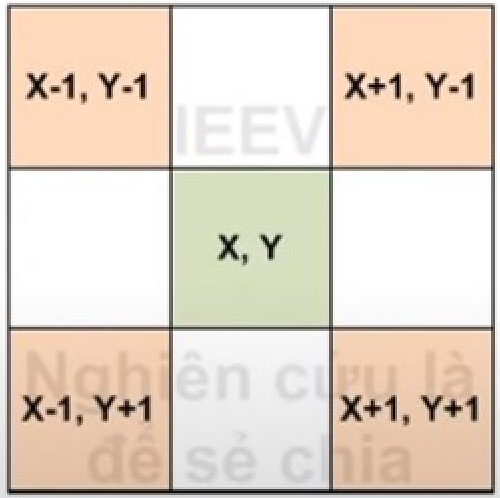


Some basic relationships between pixels

- Điểm láng giềng của p(x,y) neighbors of a pixel p(x,y)
- *4-neighbors* of *p N*4(p).
  8-neighbors of *p N*8(p)







Adjacency, connectivity, regions, and boundaries

4-adjacency: p and q are 4-adjacent if q is one of the four neighbors of p that are directly connected to p
8-adjacency: p and q are 8-adjacent if q is one of the eight neighbors of p that are directly connected
m-adjacency: Two pixels p and q are m-adjacent if they are either 4-adjacent or 8-adjacent
Path (or curve) as a sequence of distinct pixels that connect two pixels, p (x0, y0) and q (xn, yn). The sequence of pixels that make up the path is represented by the following coordinates:
(x0, y0), (x1, y1), ..., (xn-1, yn-1), (xn, yn)

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

Distance metric

The *Euclidean distance* between p and q is defined as

$$D_e(p,q) = \left[(x-u)^2 + (y-v)^2\right]^{\frac{1}{2}}$$

The D_4 *distance*, (called the *city-block distance*) between p and q

$$D_4(p,q) = |x-u| + |y-v|$$

The D_8 *distance* (called the *chessboard distance*) between p and q

$$D_8(p,q) = \max(|x-u|, |y-v|)$$