

# **Lecture 8**

## **Color image processing & Morphological image processing**

**Dr. Xiran Cai**

**Email: [caixr@shanghaitech.edu.cn](mailto:caixr@shanghaitech.edu.cn)**

**Office: 3-438 SIST**

**Tel: 20684431**

**ShanghaiTech University**



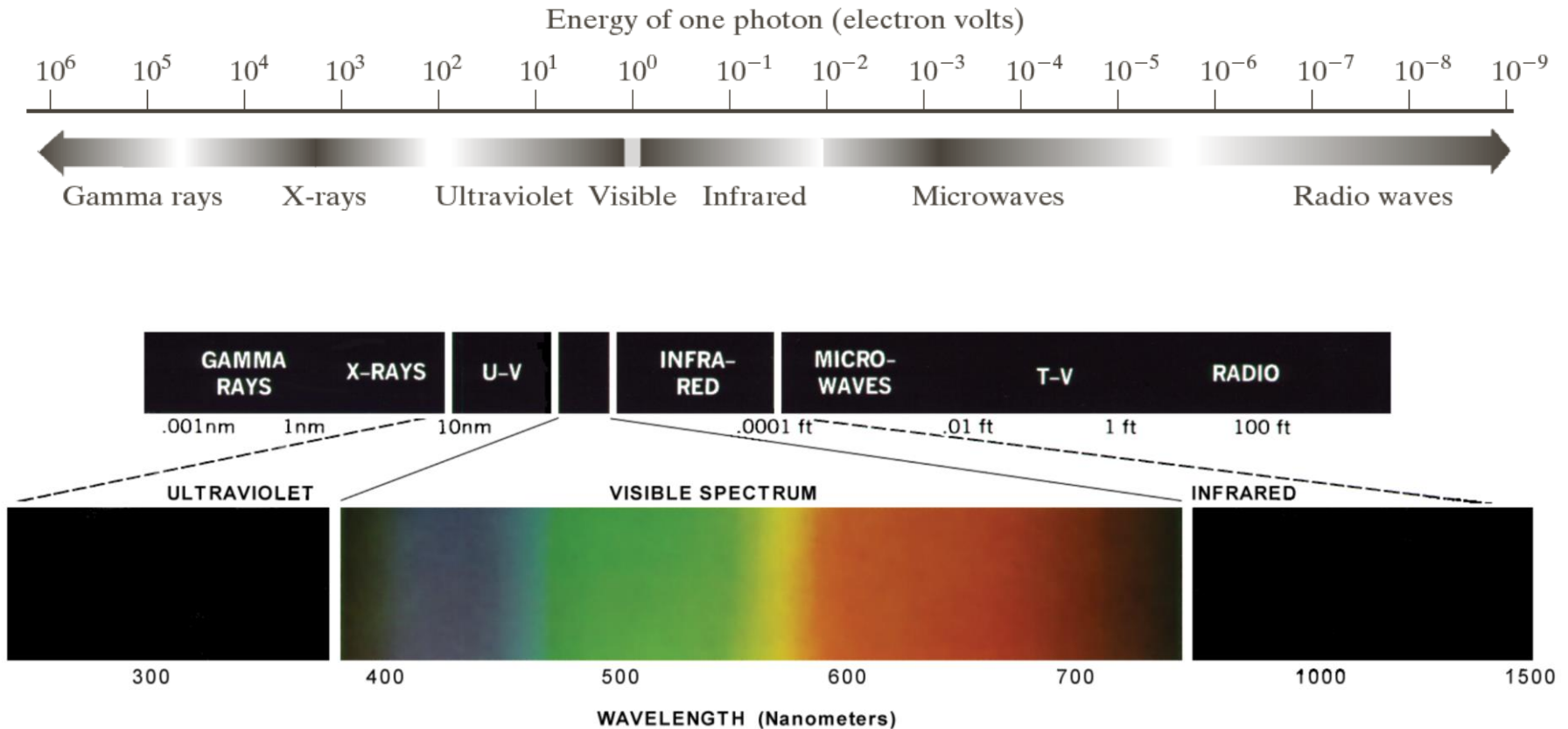
# Color image processing: Outline

---

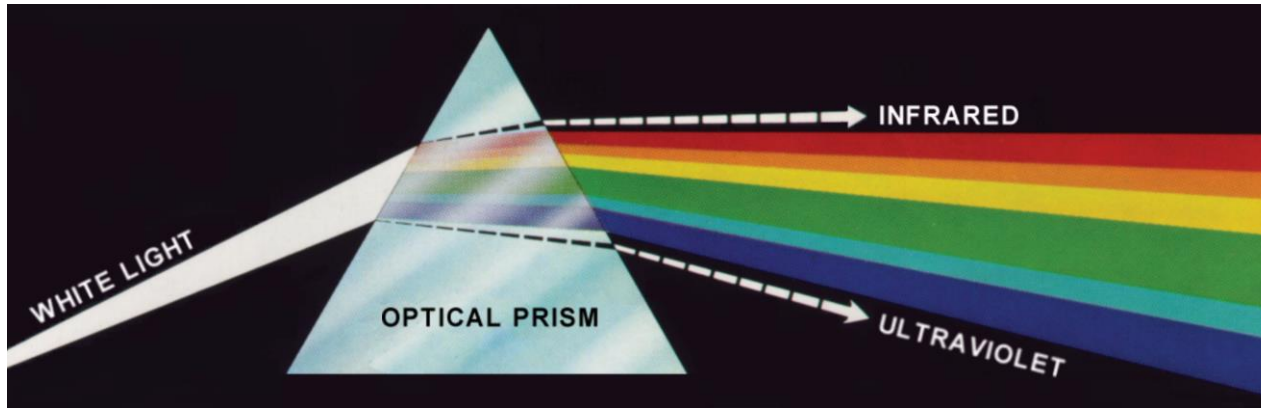
- ❑ Standard color spaces
  - RGB
  - CMYK
  - HSI/HSV
- ❑ Transform between color spaces
  - RGB to gray scale
  - RGB to HSI
- ❑ Color balance



# Electromagnetic spectrum

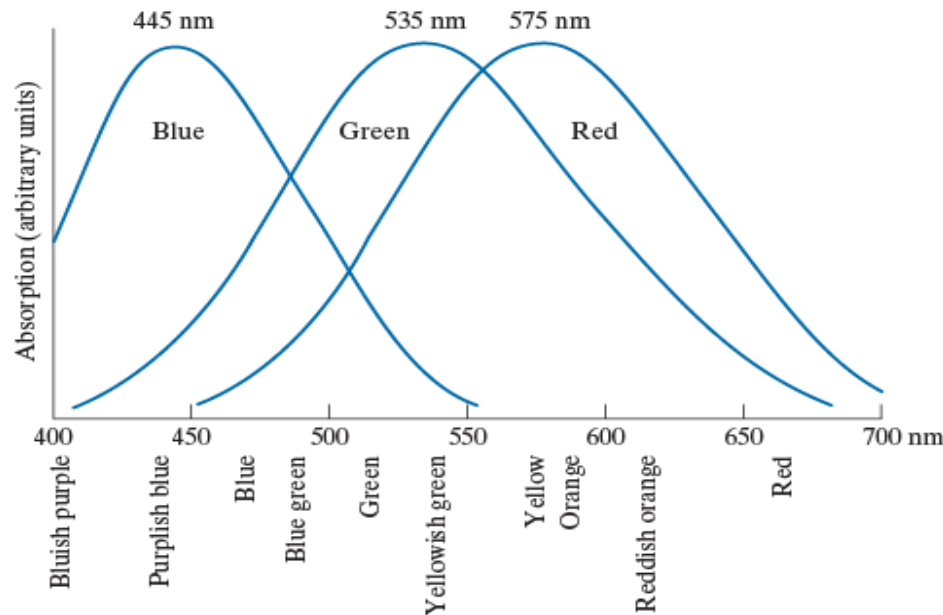


# Dispersion of light



**FIGURE 7.2**

Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lighting Division.)



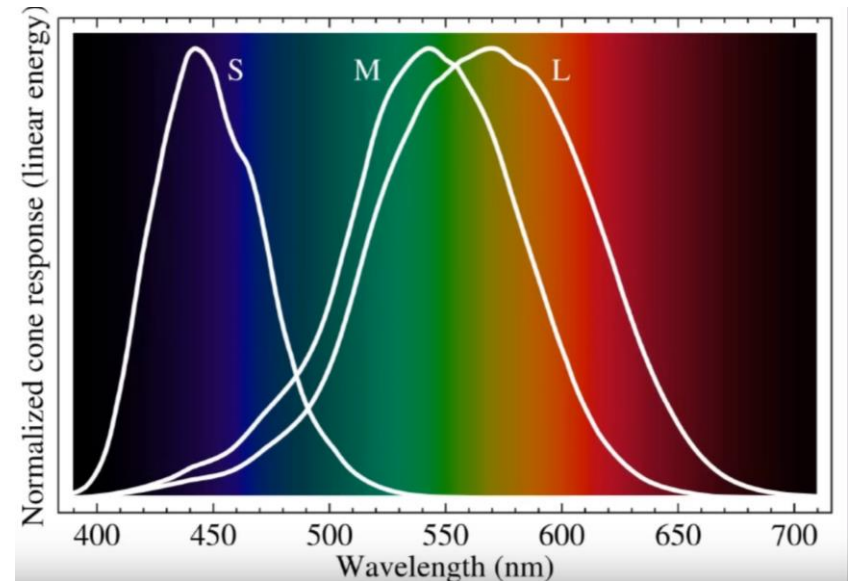
**FIGURE 7.3**

Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.



# Human visual system color Space

- ❑ **The LMS color space:** Cones enable color perceptions;
- ❑ **3 Types of cones:**
  - **Long:** sensitive to  
“RED” (more yellow to blue) 65%
  - **Middle :** sensitive to  
“GREEN” (more green to blue) 33%
  - **Short :** sensitive to  
“BLUE” (more blue to purple) 2%  
(But most sensitive)



# Primary colors

## ❑ CIE RGB Standard (International Commission on Illumination)

French: Commission Internationale d'éclairage

- Blue = 435.8 nm
- Green = 546.1 nm
- Red = 700 nm

## ❑ The white light is achieved with a mixture of RGB light with:

- 1.0000 : 4.5907 : 0.0601 of intensity (Luminous intensity) .

## ❑ CMY and CMYK color

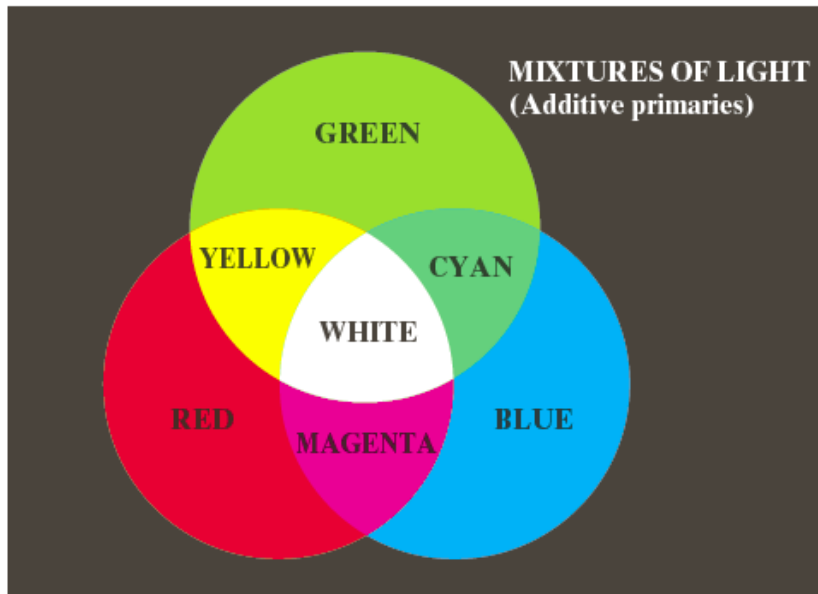
- Cyan = White – Red
- Magenta = White – Green
- Yellow = White – Blue
- Black = White – Red - Green - Blue

Human eye perception-based!  
(e.g., metamerism, 同色异谱)

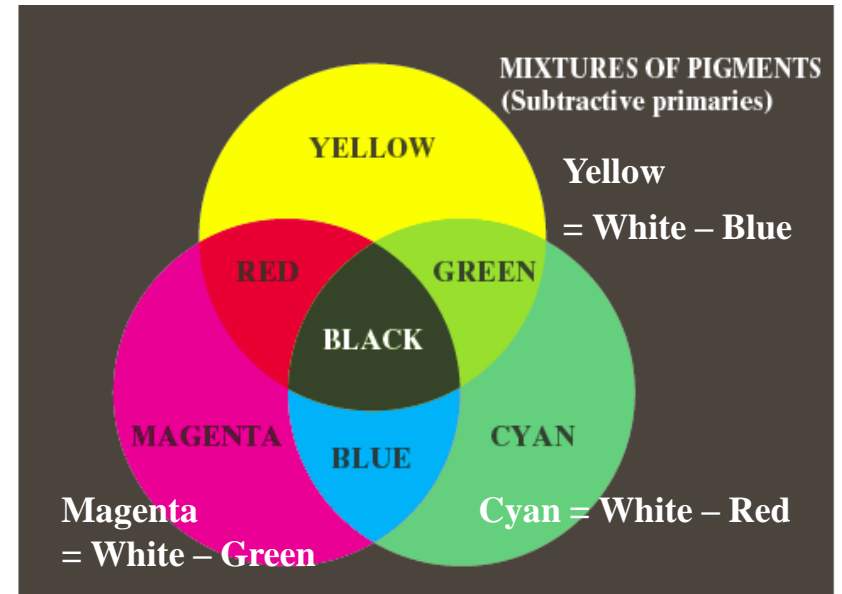


# Secondary colors

## RGB color



## CMY and CMYK color



**Black = White - Red - Green - Blue**



# CMY Color Model

## ➤ RGB to CMY conversion

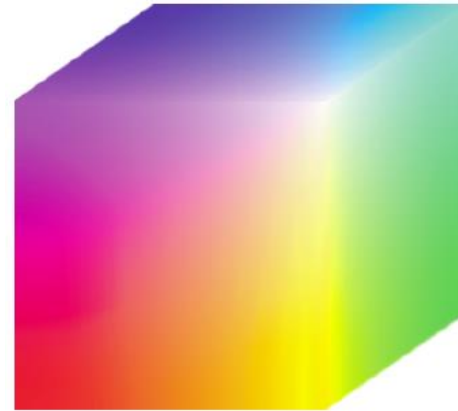
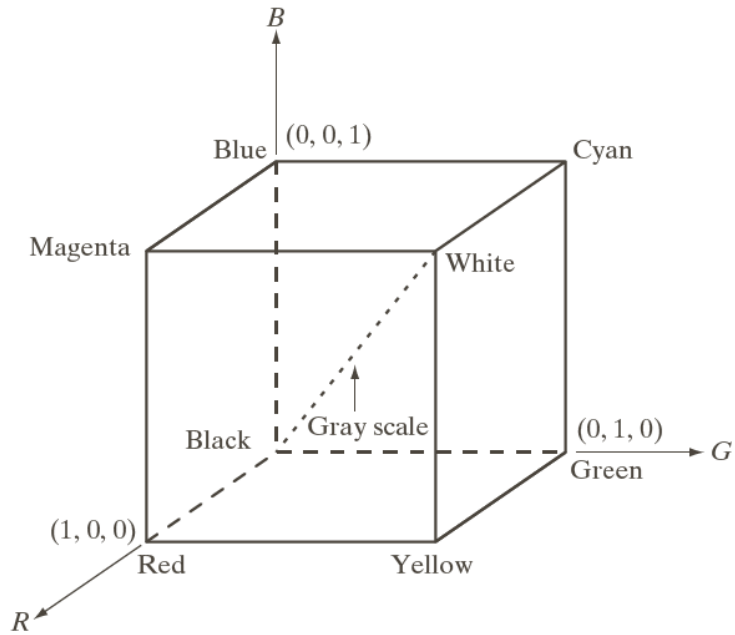
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

In order to produce true black in printing, a fourth color, black, is added into the CMYK color model



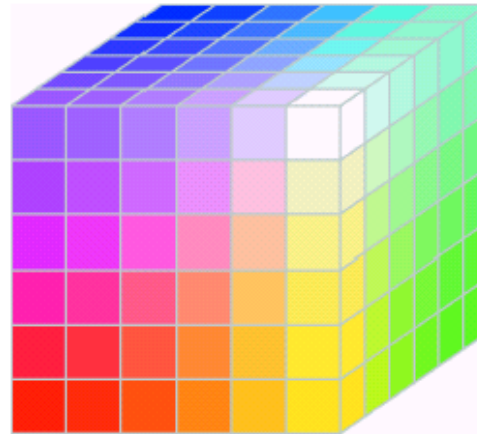
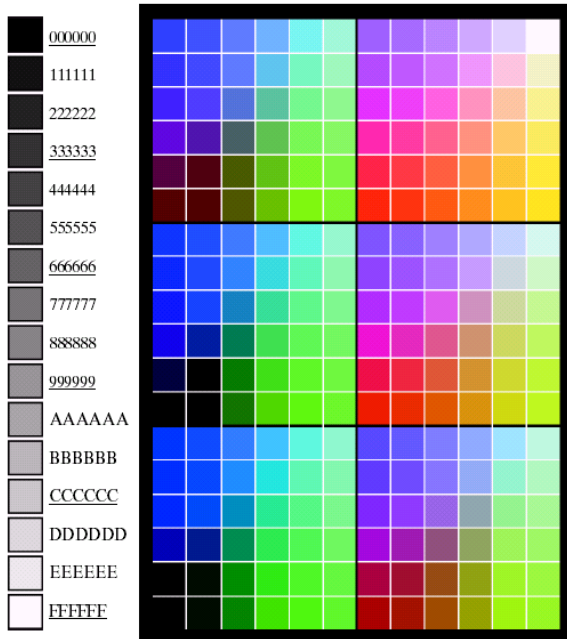


# RGB Color Model

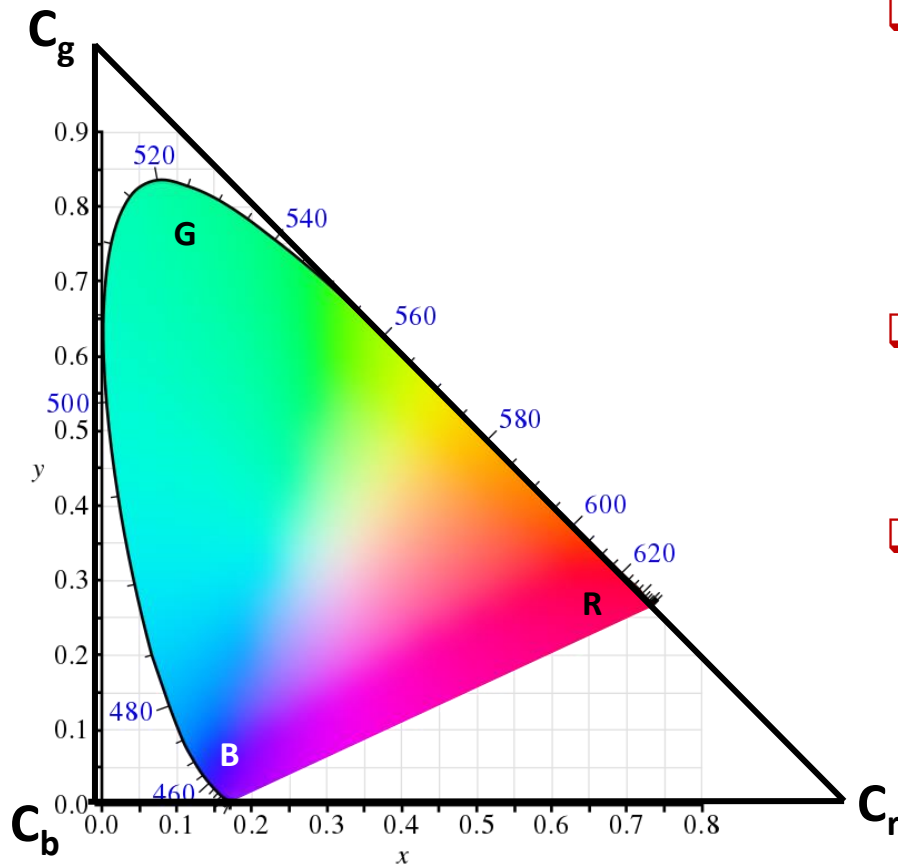


# Safe/Standard RGB Color (24 bytes)

Number System		Color Equivalents					
Hex	00	33	66	99	CC	FF	
Decimal	0	51	102	153	204	255	



# Chromaticity diagram



□  $x = \frac{R}{R+G+B}, y = \frac{G}{R+G+B}, z = \frac{B}{R+G+B}$

Then  $z = 1 - x - y$ .

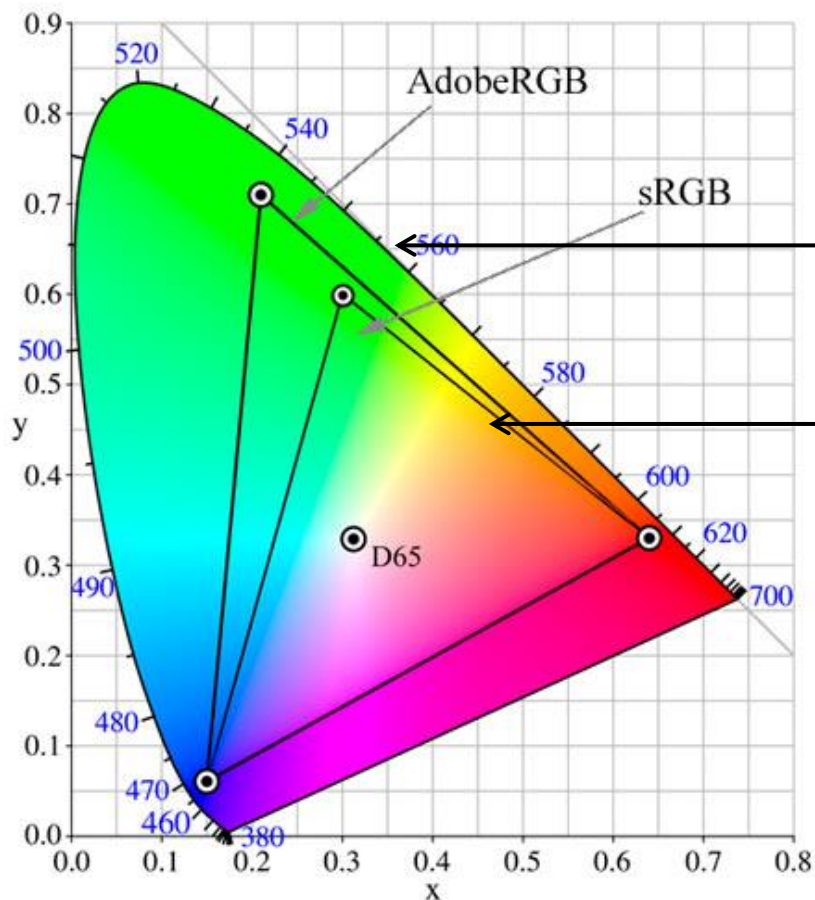
The color cube turns to a 2-D color gamut.

□ It is seen that all visible chromaticities correspond to non-negative values of  $x$ ,  $y$ , and  $z$ .

□ An equal mixture of two equally bright colors will not generally lie on the midpoint of that line.



# Color Gamut



White: D65 [0.3127,0.3290]

Red: [0.6400, 0.3300]

Green: [0.3000, 0.6000]

Blue: [0.1500, 0.0600]

CIE Chromaticity diagram

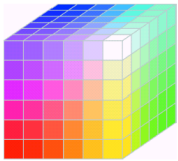
Color gamut for monitor

- **sRGB (standard Red Green Blue)** is an RGB color space and Microsoft created cooperatively in 1996 to use on monitors, printers, and the Internet.
- The **Adobe RGB (1998) color space** is an RGB color space developed by Adobe System, Inc. in 1998.

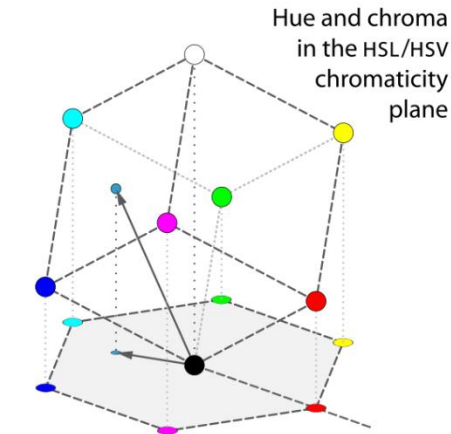


# HSI Color Model

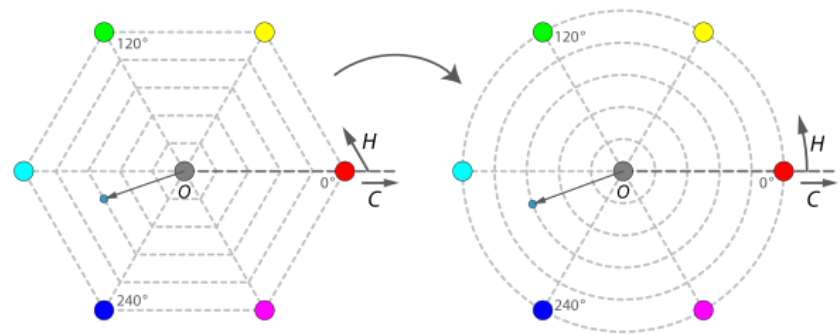
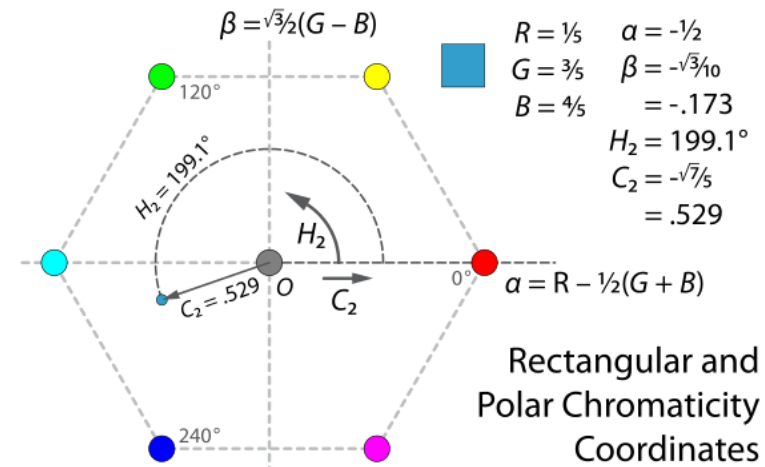
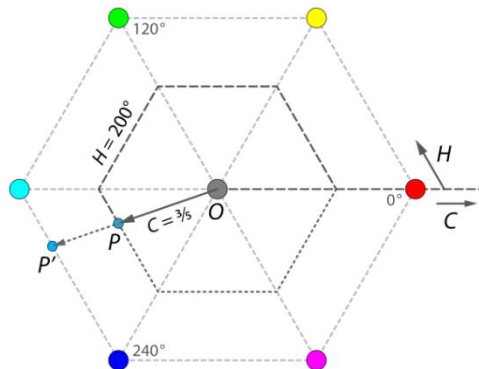
RGB color cube



Hue and chroma  
in the HSL/HSV  
chromaticity  
plane



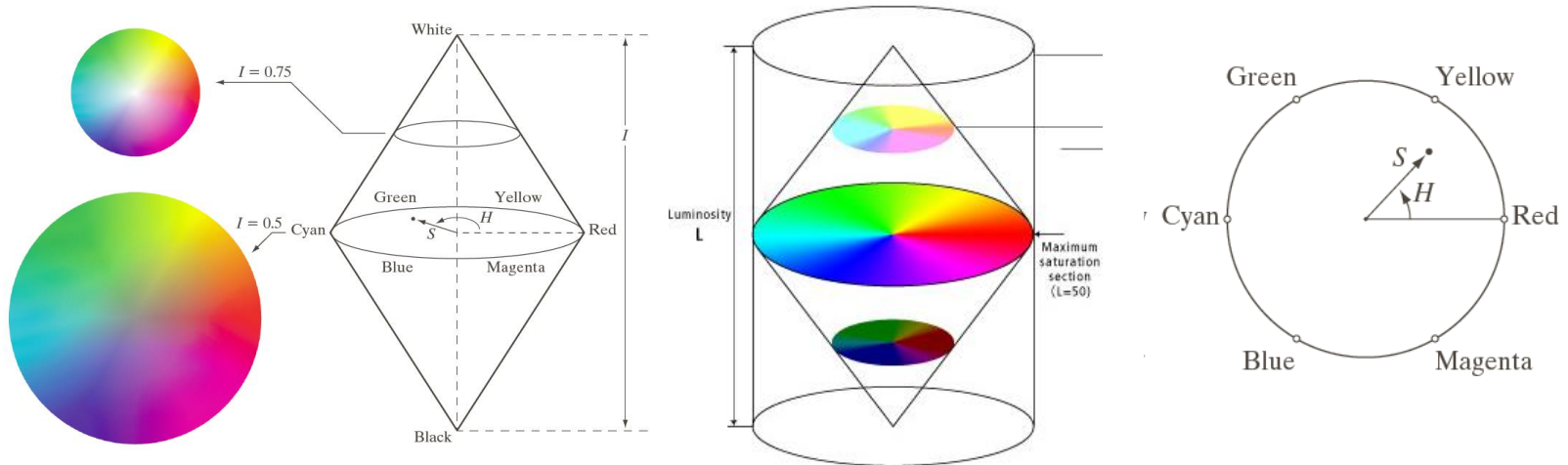
$C = \frac{OP}{OP'} = B - R = \frac{4}{5} - \frac{1}{5} = \frac{3}{5} = .6$   
 $R = \frac{1}{5}$   
 $G = \frac{3}{5}$   
 $B = \frac{4}{5}$   
 $H = 60^\circ \times \left(4 + \frac{R-G}{C}\right) = 60^\circ \times \left(4 - \frac{2}{3}\right) = 200^\circ$



# HSI Color Model

## ❑ HSI Color Model

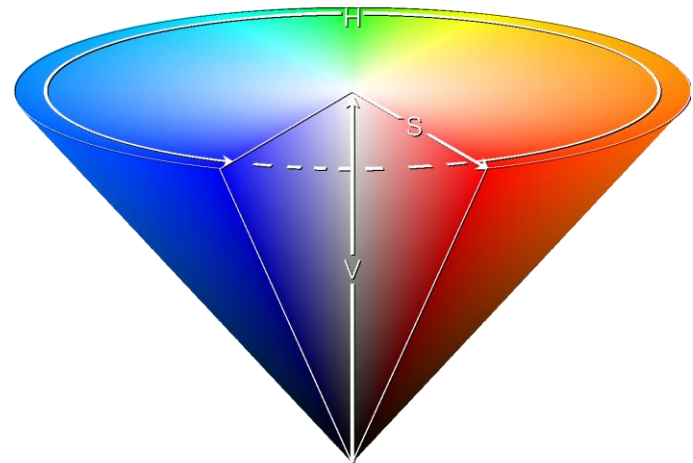
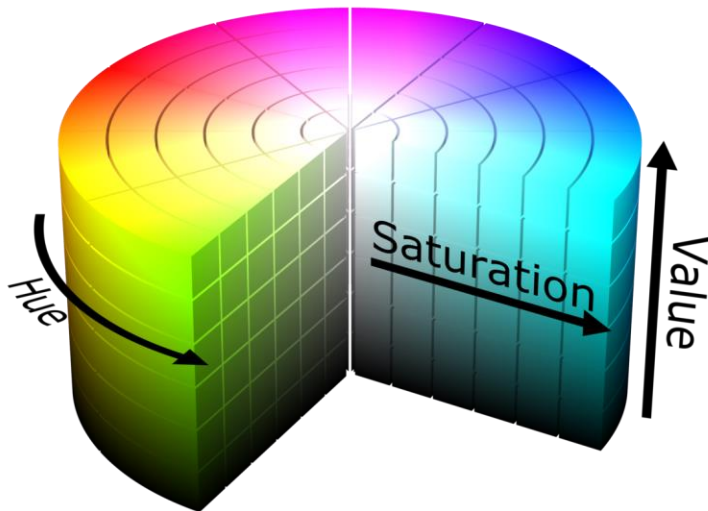
- **Hue:** Dominant color associated with wavelength.
- **Saturation:** relative purity, the amount of white light mixed with a hue
- **Intensity/Lightness.**  $I = (r + g + b) / 3$



# HSV Color Model

## ➤ HSV Color Model

- **Hue:** Dominant color associated with wavelength.
- **Saturation:** relative purity, the amount of white light mixed with a hue
- **Value.**  $v = \max(r, g, b);$



# Color image processing: Outline

---

- ❑ Standard color spaces
  - RGB
  - CMYK
  - HSI/HSV
  - Lab
- ❑ Transform between color spaces
  - RGB to gray scale
  - RGB to HSI
- ❑ Color balance





# RGB to Gray scale

➤ **Maximum value:**

$$g_R(x, y) = g_G(x, y) = g_B(x, y) = \max[f_R(x, y), f_G(x, y), f_B(x, y)]$$

➤ **Average value**

$$g_R(x, y) = g_G(x, y) = g_B(x, y) = [f_R(x, y) + f_G(x, y) + f_B(x, y)]/3$$

➤ **Weighted value**

$$g_R(x, y) = g_G(x, y) = g_B(x, y) = 0.299f_R(x, y) + 0.587f_G(x, y) + 0.114f_B(x, y)$$



# Transform to CMY Color Space

$$\square \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



$$\square \begin{bmatrix} g_R(x, y) \\ g_G(x, y) \\ g_B(x, y) \end{bmatrix} = \begin{bmatrix} 255 - f_R(x, y) \\ 255 - f_G(x, y) \\ 255 - f_B(x, y) \end{bmatrix}$$



# Transform from RGB to HSI/HSV

$$\theta = \arccos \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - G)(G - B)]^{\frac{1}{2}}} \right\}$$

$$H = \begin{cases} \theta, & G \geq B \\ 360 - \theta, & G < B \end{cases}$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)]$$

$$I = \frac{R + G + B}{3} \quad V = \max(r, g, b);$$

$$S = 0 \rightarrow H = 0, \quad I = 0 \rightarrow S = 0, \quad H = 0$$



# Transform from HSI to RGB

➤  $0^\circ \leq H < 120^\circ$

$$B = I(1 - S), \quad R = I \left[ 1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right], \quad G = 3I - (R + B)$$

➤  $120^\circ \leq H < 240^\circ$

$$R = I(1 - S), \quad G = I \left[ 1 + \frac{S \cos(H - 120^\circ)}{\cos(180^\circ - H)} \right], \quad B = 3I - (R + G)$$

➤  $240^\circ \leq H < 360^\circ$

$$G = I(1 - S), \quad B = I \left[ 1 + \frac{S \cos(H - 240^\circ)}{\cos(300^\circ - H)} \right], \quad R = 3I - (G + B)$$



# Color Balance

## ➤ White balance:

$$I(x, y) = 0.299f_R(x, y) + 0.587f_G(x, y) + 0.114f_B(x, y)$$

$$k_R = \frac{\bar{I}}{f_R} \quad k_G = \frac{\bar{I}}{f_G} \quad k_B = \frac{\bar{I}}{f_B}$$

$$\begin{bmatrix} g_R(x, y) \\ g_G(x, y) \\ g_B(x, y) \end{bmatrix} = \begin{bmatrix} k_R & & \\ & k_G & \\ & & k_B \end{bmatrix} \begin{bmatrix} f_R(x, y) \\ f_G(x, y) \\ f_B(x, y) \end{bmatrix}$$

## ➤ Maximum value balance

$$S_{RGB} = \min[R_{max}, G_{max}, B_{max}]$$

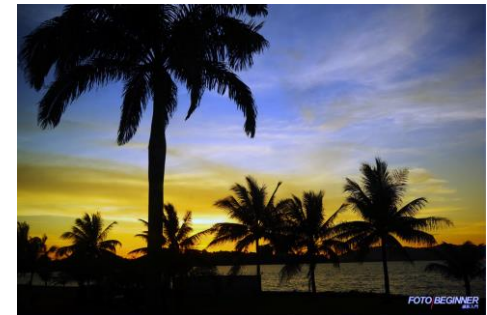
$$k_R = \frac{S_{RGB}}{T_R} \quad k_G = \frac{S_{RGB}}{T_G} \quad k_B = \frac{S_{RGB}}{T_B}$$

$$\begin{bmatrix} g_R(x, y) \\ g_G(x, y) \\ g_B(x, y) \end{bmatrix} = \begin{bmatrix} k_R & & \\ & k_G & \\ & & k_B \end{bmatrix} \begin{bmatrix} f_R(x, y) \\ f_G(x, y) \\ f_B(x, y) \end{bmatrix}$$

1. Find the smallest max value  $S_{RGB}$  in each color channel.

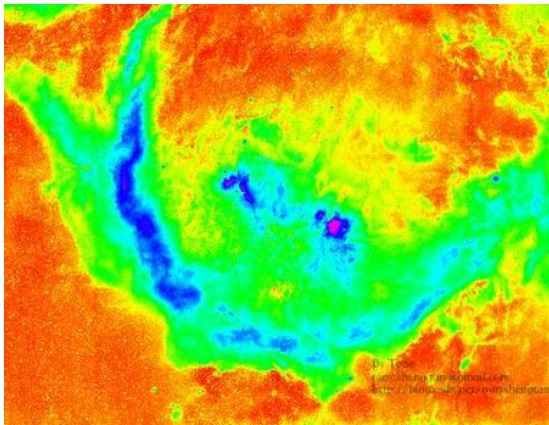
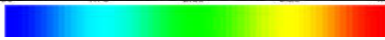
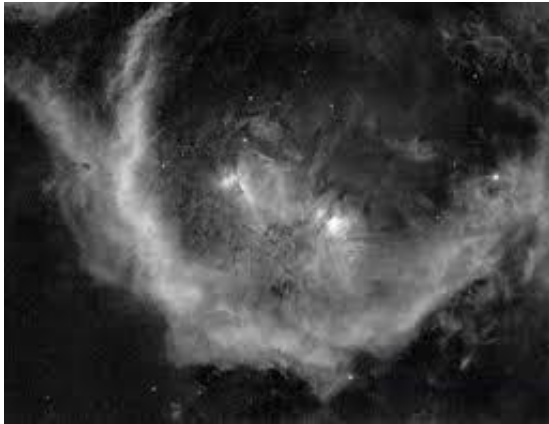
2. Calculate the number of intensities  $N_r, N_g, N_b$  that larger than  $S_{RGB}$  in each color channel. Then find the largest number  $N_{max} = \max[N_r, N_g, N_b]$ .

3. Sort the intensities in each channel and find the the  $N_{max}_{th}$  intensity value  $[T_r, T_g, T_b]$  in each color channel.





# Pseudo color enhancement



# Take home message

---

- ❑ The color that you perceived depends on the cone cells in your eye.
- ❑ There are variety of different color space defined by CIE. Each color space has its unique advantage.
- ❑ When the intensity in each color channel is unbalance, the color looks weird. Try to practice color space transform by implementing a color balance correction method.



# Morphological image processing: Outline

---

## Morphology Image Processing （形态学图像处理）

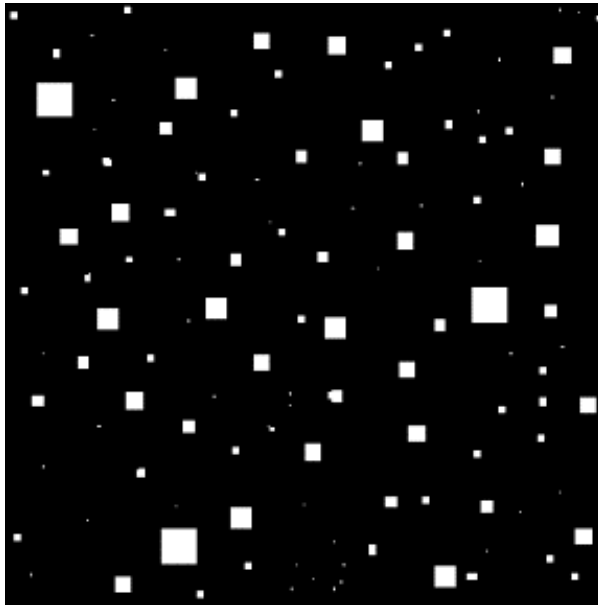
- Morphological operation.
- Morphological algorithms.





# Problem try to solve

- ❑ Imperfect image segmentation.



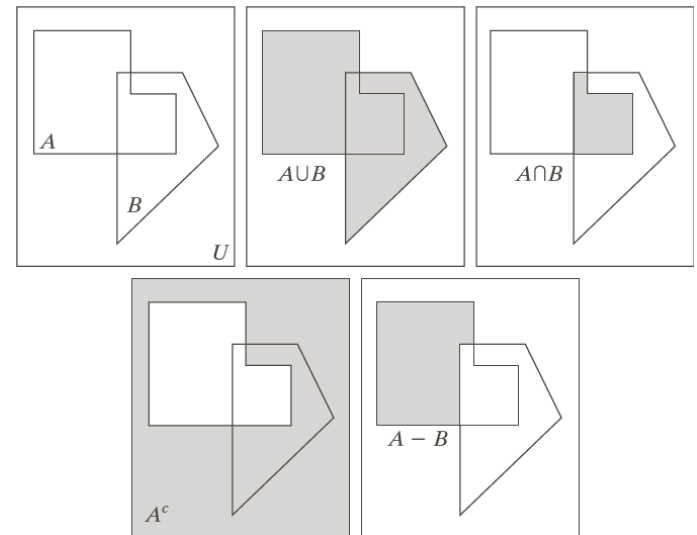
Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



# Preliminaries: Set Operation

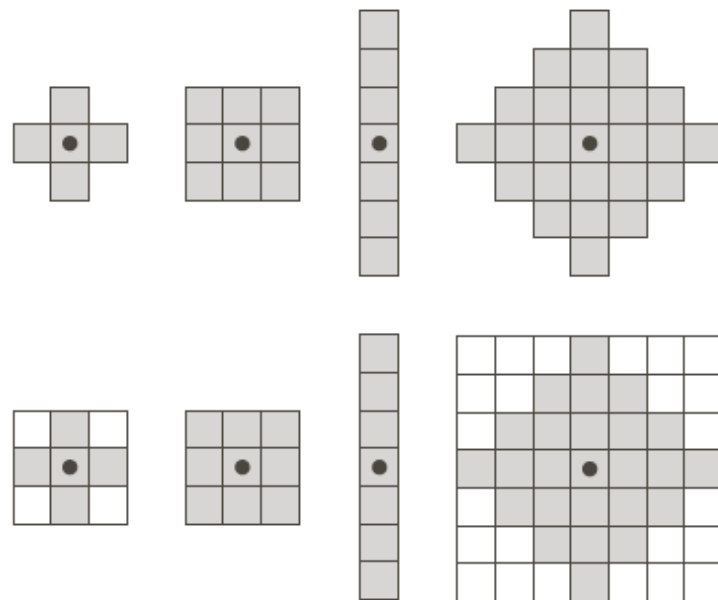
A digital image  $f(x, y)$  can be considered as a set  $A$ , if  $w(x, y)$  in 2D integer space  $Z^2$ , then

- $w \in A$ :  $w$  is an element of  $A$ .
- $w \notin A$ :  $w$  is not an element of  $A$ .
- $B = \{w | \text{condition}\}$ : all elements which meet the specific condition.
  - $A \cup B = \{w | w \in A \text{ or } w \in B\}$ : union (并集)
  - $A \cap B = \{w | w \in A \text{ and } w \in B\}$ : intersection (交集)
  - $A^c = \{w | w \notin A\}$ : complement (补集)
  - $A - B = \{w | w \in A \text{ and } w \notin B\}$ : difference (差集)



# Structuring element (结构元)

- **Structuring Element (SE):** small sets or sub-images used to probe an image under study for properties of interest.
- **SE Selection**
  - Simpler than the image
  - With boundary
  - Convex
- **Structures**
  - Origin
  - Rectangular



# Erosion (腐蚀)

## □ Definition:

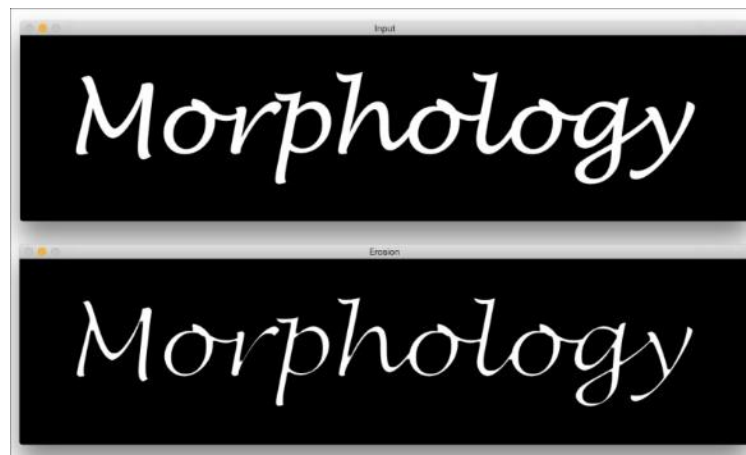
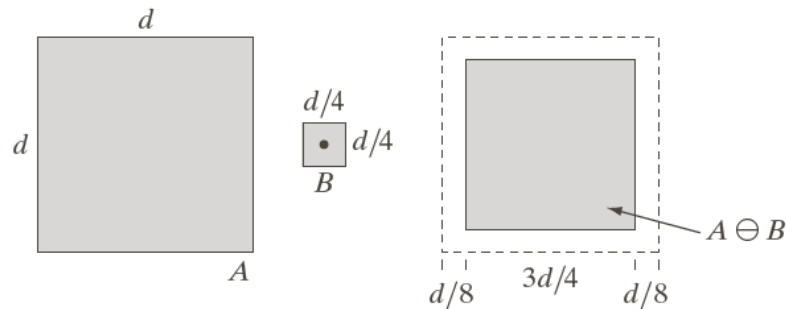
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

or

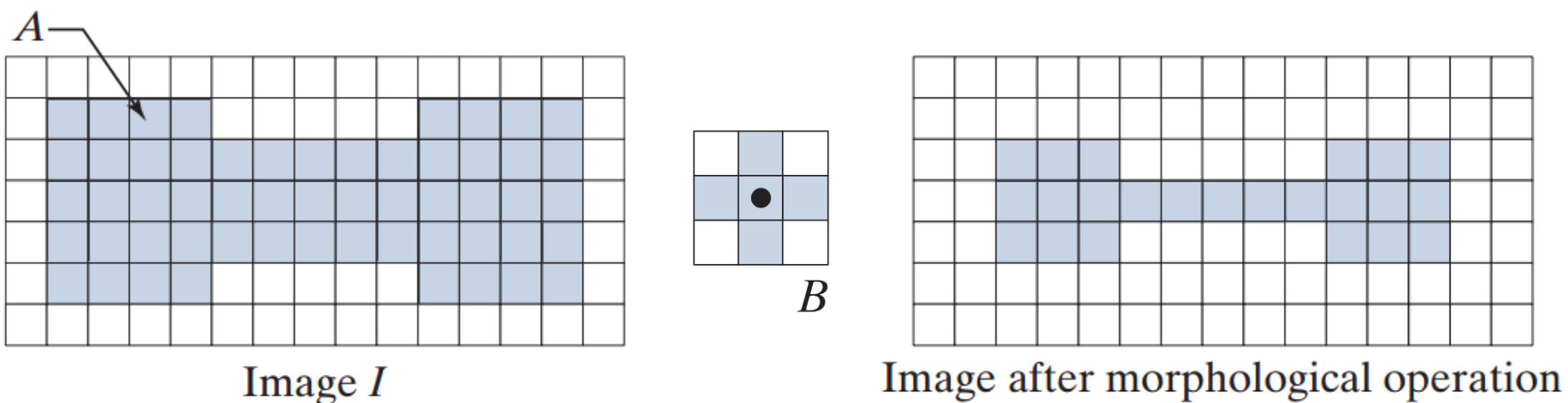
$$A \ominus B = \{z \mid (B)_z \cap A^c = \emptyset\}$$

## ➤ Erosion will do:

- removes thin lines
- isolate dots
- leaves gross details
- “Peeling away” layers
- Is always a sub-set of A



# Erosion (腐蚀), example



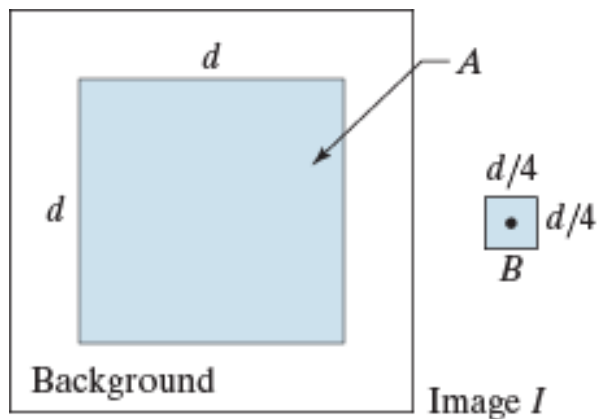
a b c

**FIGURE 9.3**

(a) A binary image containing one object (set),  $A$ . (b) A structuring element,  $B$ . (c) Image resulting from a morphological operation (see text).



# Erosion (腐蚀), example



a b c  
d e

**FIGURE 9.4**

(a) Image  $I$ , consisting of a set (object)  $A$ , and background.

(b) Square SE,  $B$  (the dot is the origin).

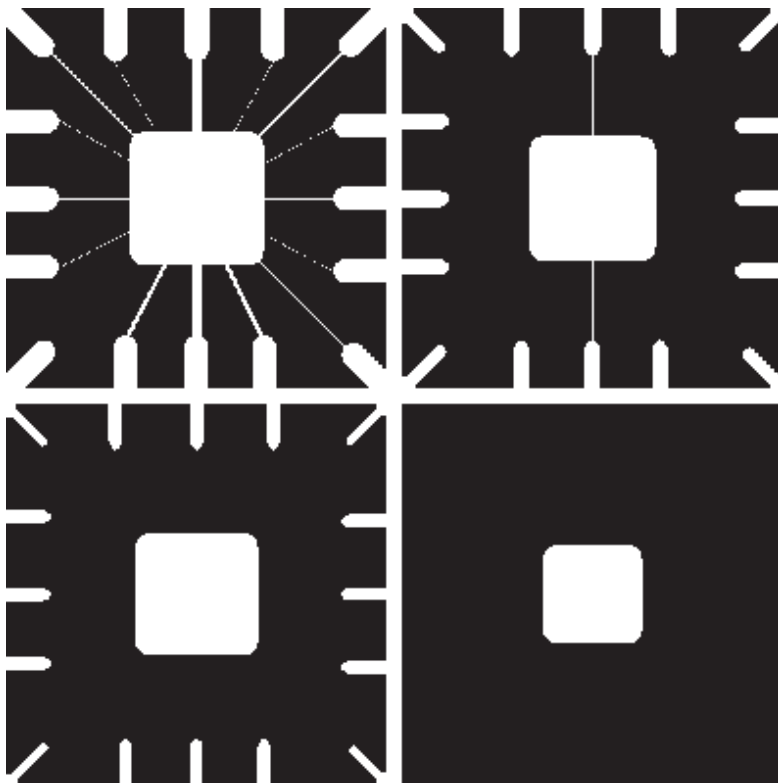
(c) Erosion of  $A$  by  $B$  (shown shaded in the resulting image).

(d) Elongated SE.

(e) Erosion of  $A$  by  $B$ . (The erosion is a line.) The dotted border in (c) and (e) is the boundary of  $A$ , shown for reference.



# Erosion application



a b  
c d

**FIGURE 9.5**

Using erosion to remove image components.

(a) A  $486 \times 486$  binary image of a wire-bond mask in which foreground pixels are shown in white. (b)–(d) Image eroded using square structuring elements of sizes  $11 \times 11$ ,  $15 \times 15$ , and  $45 \times 45$  elements, respectively, all valued 1.



# Dilation (膨胀)

## □ Definition

$$A \oplus B = \{z | \widehat{B_z} \cap A \subseteq A\}$$

or

$$A \oplus B = \{z | \widehat{B_z} \cap A \neq \emptyset\}$$

- **Dilation** will do:
  - Fatten up. Kind of opposite of Erosion.
  - Bridge gaps, fill holes, without change overall size of object.

$\widehat{B_z}$  overlap at least one element of  $A$



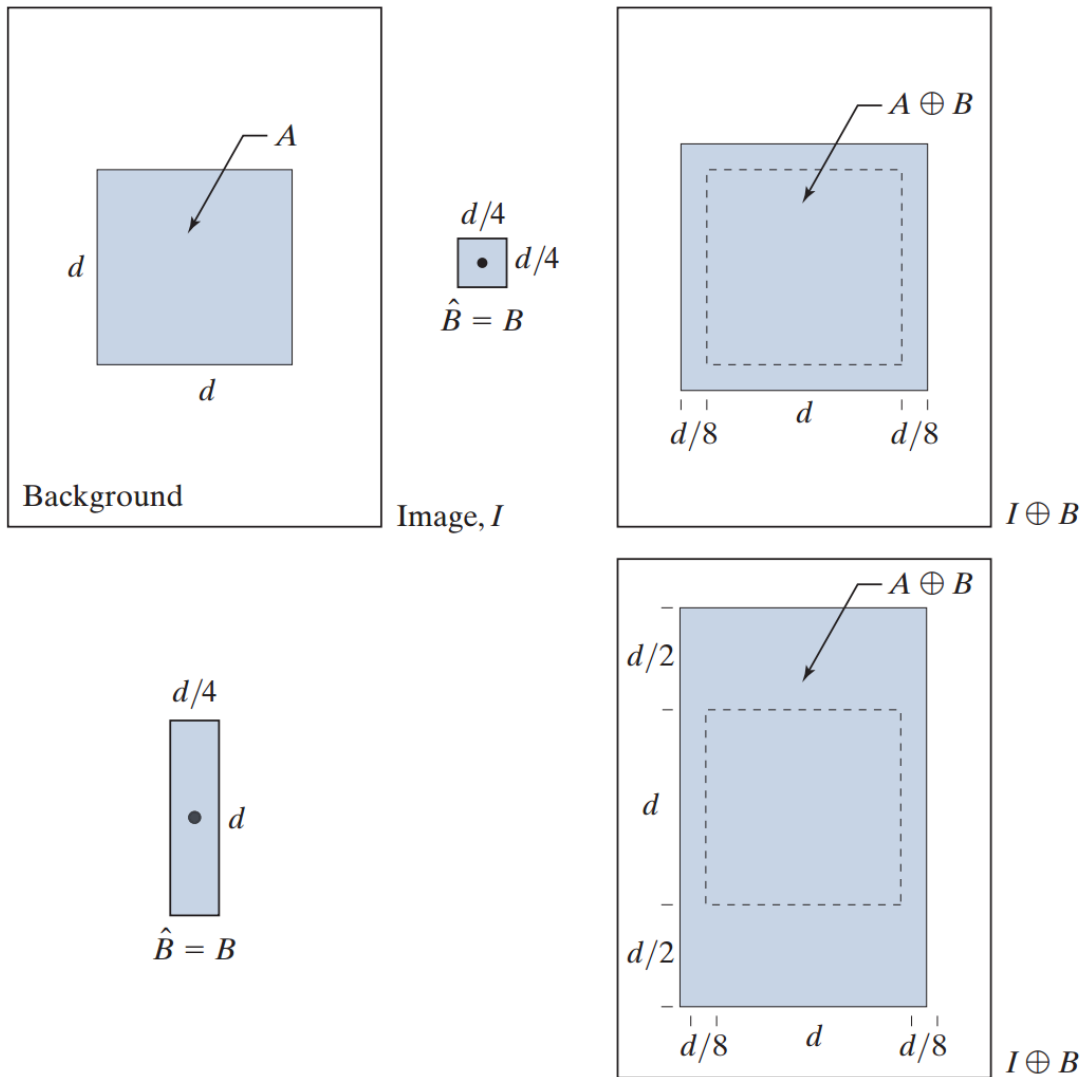


# Dilation (膨胀), example

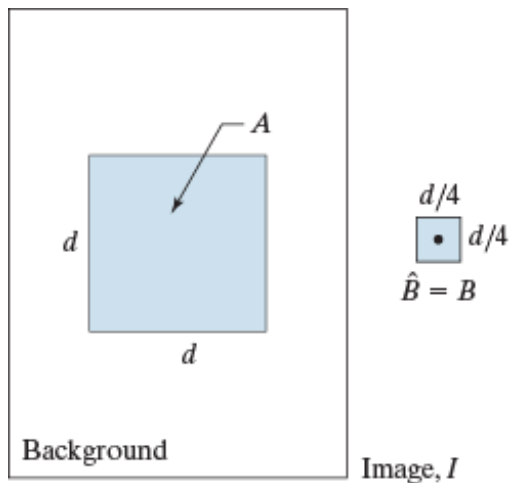
a b c  
d e

**FIGURE 9.6**

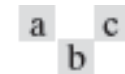
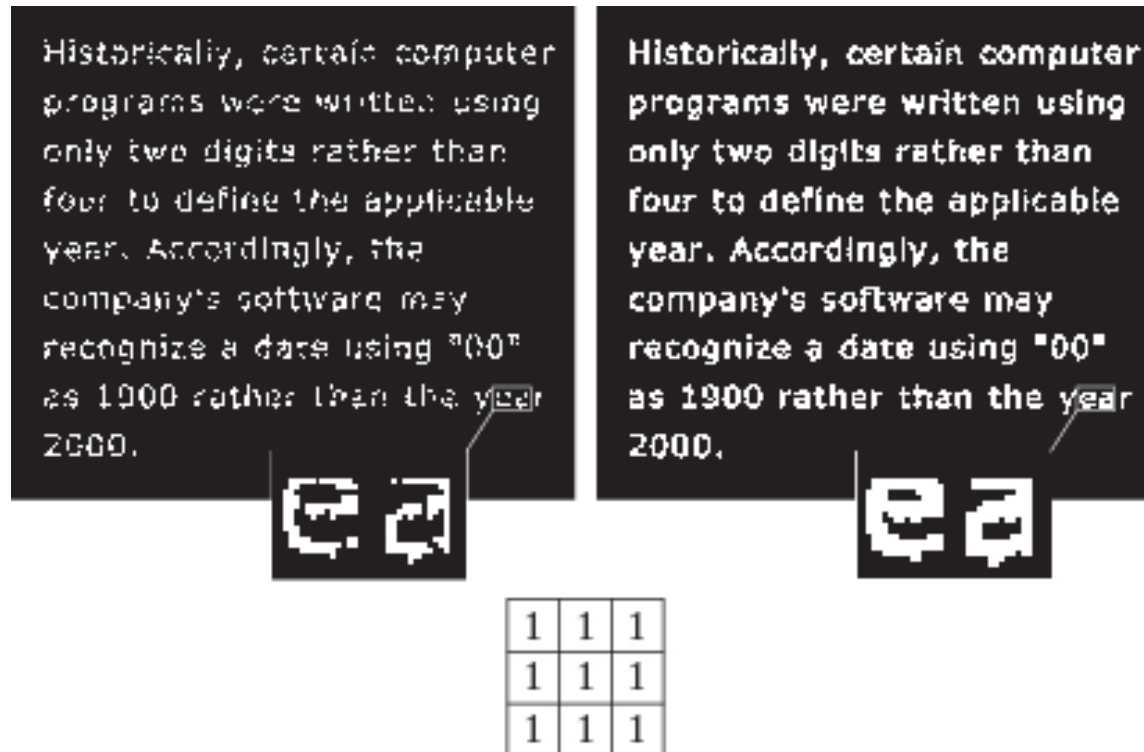
(a) Image  $I$ , composed of set (object)  $A$  and background.  
(b) Square SE (the dot is the origin).  
(c) Dilation of  $A$  by  $B$  (shown shaded).  
(d) Elongated SE.  
(e) Dilation of  $A$  by this element. The dotted line in (c) and (e) is the boundary of  $A$ , shown for reference.



# Dilation (膨胀), example



# Dilation application



**FIGURE 9.7**

(a) Low-resolution text showing broken characters (see magnified view).

(b) Structuring element.

(c) Dilation of (a) by (b). Broken segments were joined.



# Opening (开操作)

➤ **Definition:**

$$A \circ B = (A \ominus B) \oplus B$$

or

$$A \circ B = \bigcup \{ (B)_z \mid (B)_z \subseteq A \}$$

➤ **Erode then dilate: break narrow bridges, eliminate thin structures**

➤ **Matlab Function:**  $J = \text{imopen}(I, SE)$

➤ **Properties:**

①  $A \circ B$  is a subset (subimage) of  $A$

② If  $C$  is a subset of  $D$ , the  $C \circ B$  is a subset of  $D \circ B$

③  $(A \circ B) \circ B = A \circ B$



# Closing (闭操作)

➤ **Definition:**

$$A \bullet B = (A \oplus B) \ominus B$$

➤ **Dilate, then erode: fuse narrow breaks, eliminate small holes**

➤ **Matlab Function:** `J = imclose(I,SE)`

➤ **Properties:**

① A is a subset (subimage) of  $A \bullet B$

② If C is a subset of D, the  $C \bullet B$  is a subset of  $D \bullet B$

③  $(A \bullet B) \bullet B = A \bullet B$



# Opening & Closing

## ❑ Opening

- Smooth the contour of an object
- Break narrow bridges
- Eliminate thin structures

## ❑ Closing

- Smooth the contour of an object
- Fuse narrow breaks and long thin gulfs
- Eliminate small holes
- Fill gaps in the contour



# The Hit-or-Miss Transformation

$$A \odot B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

```

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

```

$B_1$

```

    1
  1 1 1
    1

```

```

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

```

$B_2$

```

    1 1
  1 1
    1

```

```

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

```

```

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

```

```

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

```



# The Hit-or-Miss Transformation

- A method to find the location of a shape  $B_1$  in an image  $A$ .
  - Erosion of  $A \ominus B_1$  gives all places where  $B_1$  fits in  $A$ .
- So also require the boundary around the shape,  $B_2$  to be empty.
  - Erosion of  $A^c \ominus B_2$  gives all places where  $B_2$  fits in empty places of  $A$ .
- Then take the intersection:

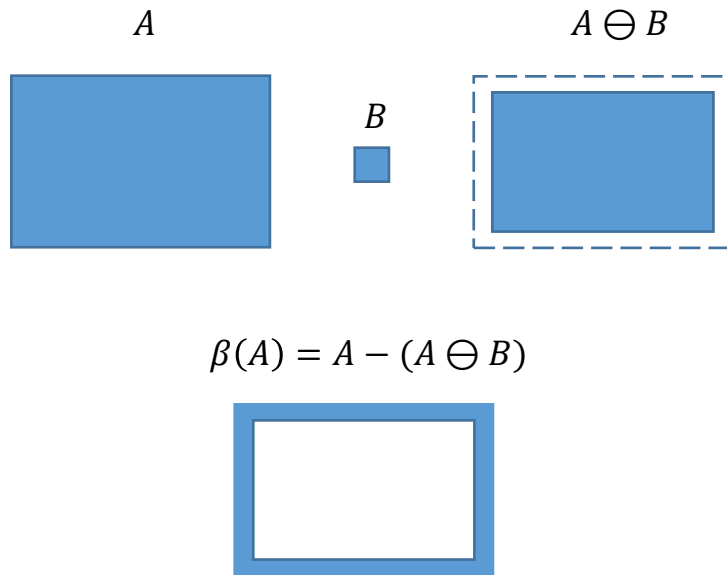
$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$





# Boundary Extraction

Morphological algorithm:  $\beta(A) = A - (A \ominus B)$

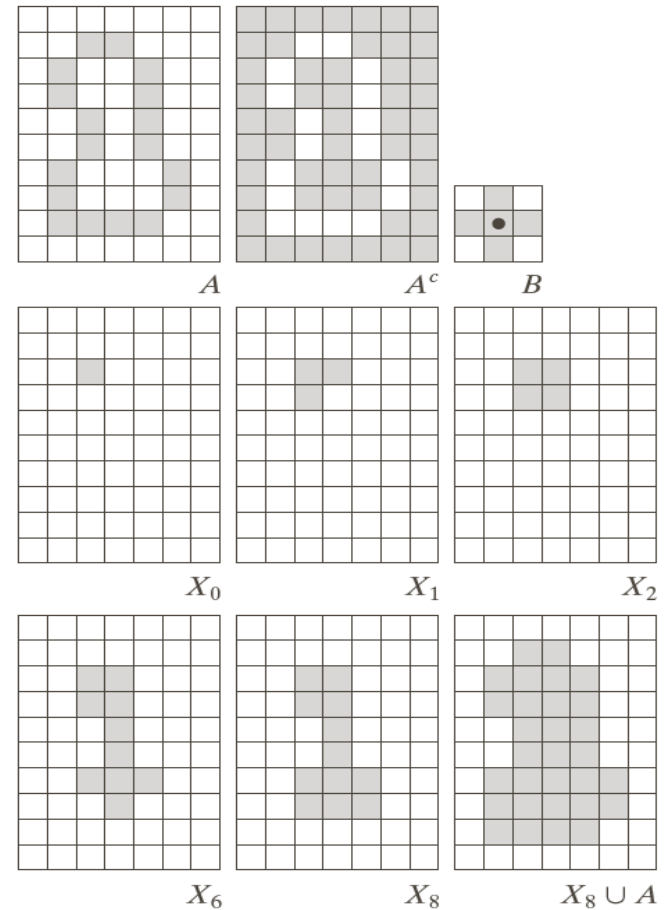


# Hole Filling

- ❑ Let  $A$  be the set of 8-connected boundary points of a region
- ❑ Start with a point inside the region
- ❑ Repeatedly dilate
- ❑ At each step, the points corresponding to the region boundary are set to zero :

$$X_k = (X_{k-1} \oplus B) \cap A^c, k = 1, 2, 3, \dots$$

- ❑ Stop when no more changes



# Take home message

- ❑ Morphological Language: Set theory (集合)
- ❑ Morphological operations take a set of pixels
- ❑ Key element: “structuring element”
- ❑ Insensitive to noise & Smooth edge
- ❑ Key operations
  - Erosion
  - Dilation
  - Opening
  - Closing
  - HMT (Hit or Miss Transformation)

