Digital Image Processing

Morphological Image Processing (2)

Color Representation

Topics

- Morphological Operations
  - Connected Component Extraction
  - Convex Hull
  - Thinning
  - Thickening
  - Skeleton
  - Pruning
  - Extension to gray level images
  - Matlab Examples

Dilation and Erosion

- Dilation and Erosion are two basic operations in morphological processing.
- Dilation of a set A in Z^2 by a set B in Z^2 is denoted by $A \oplus B$ and given by

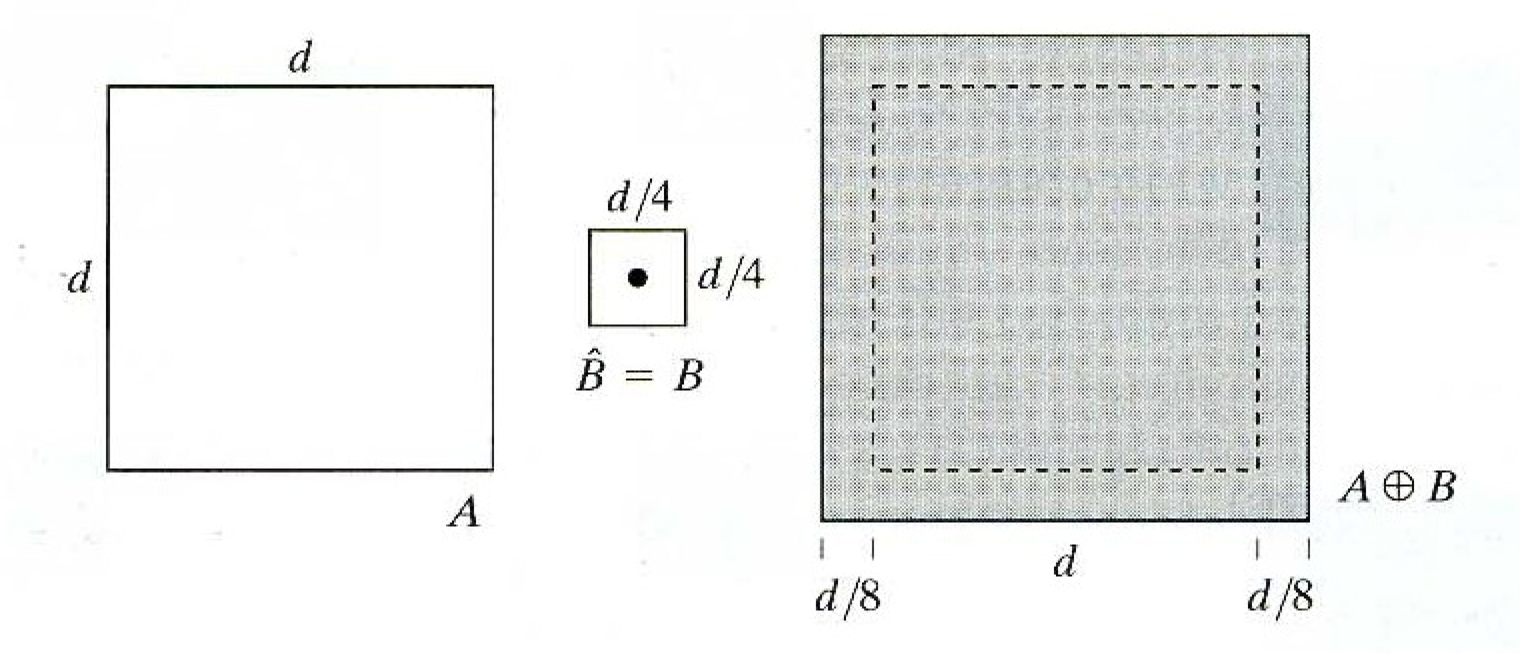
$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \varnothing \}$$

Dilation

- The dilation of A by B is the set of all displacements such that A and $\hat{B}$ overlap with at least one point

- B is called structuring element

Dilation



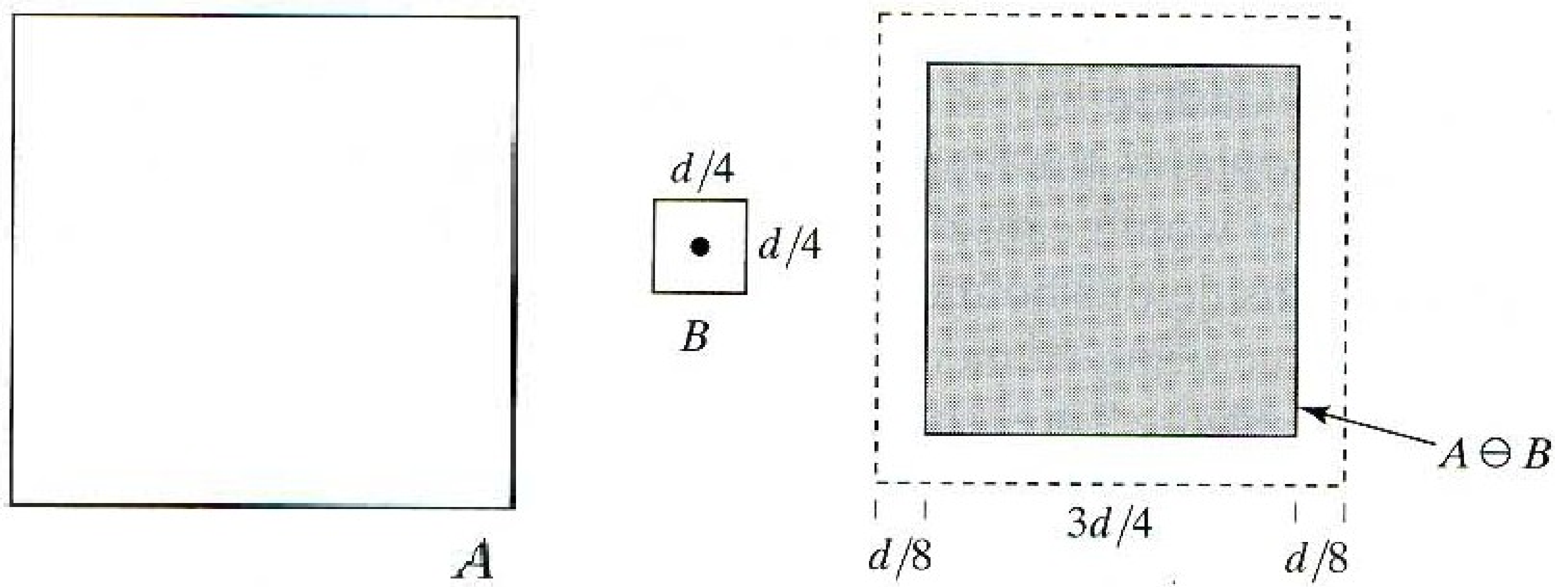
Erosion

- Erosion of a set A in Z^2 by a set B in Z^2 is denoted by $A \ominus B$ and given by:

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

Erosion of A by B is the set of all points z such that B translated by z is contained in A

Erosion



Opening

- Opening smoothes the outer contours, breaks narrow connections, and eliminates small protrusions.

- Opening is defined as :

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

Closing

- Closing smoothes the object contour, fuses narrow connections, eliminates small holes and gaps.

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

Extraction of Connected Components

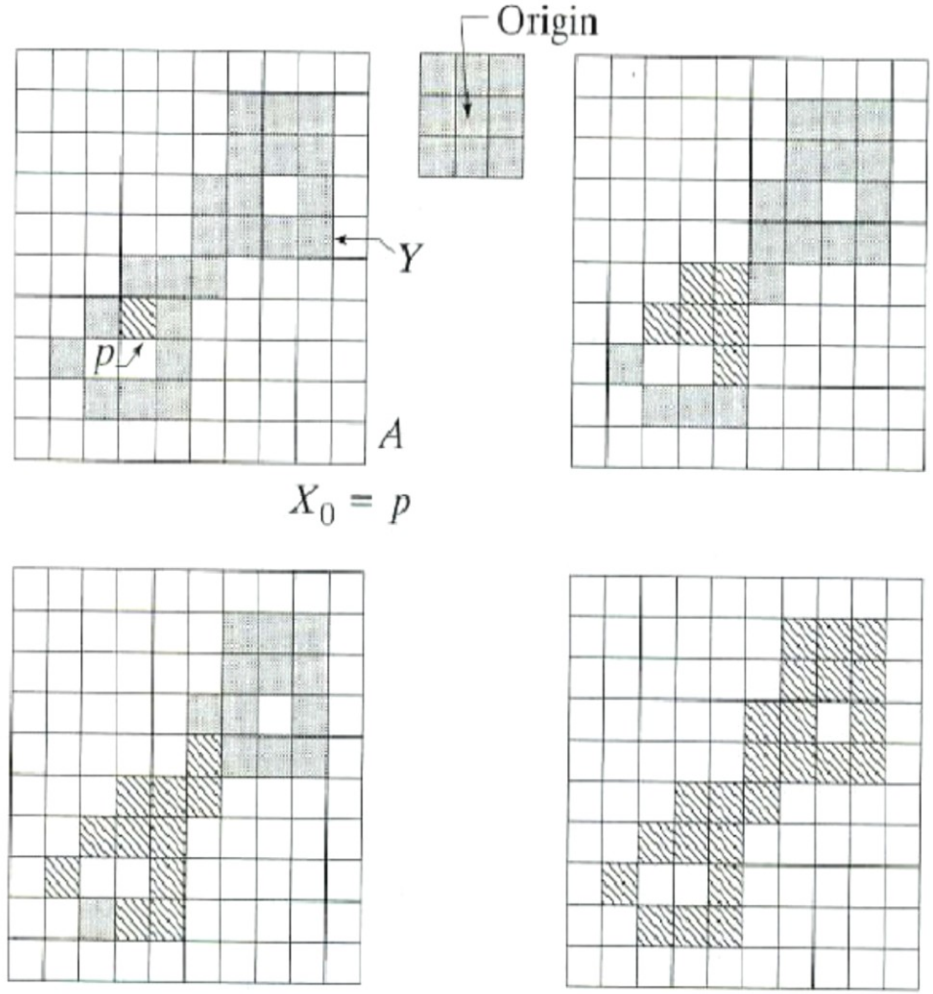
- Begin with a point P inside the connected component, iterate:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \ldots$$

Until $X_k = X_{k-1}$

Initially $X_0 = P$

Connected Component Extraction

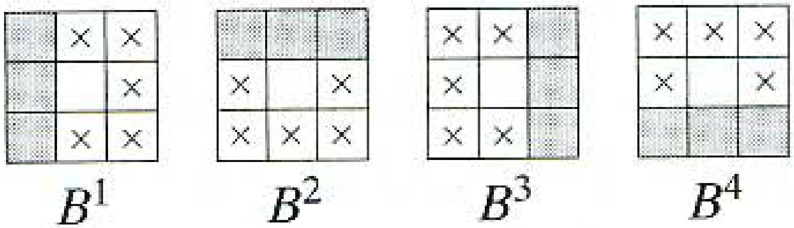


Convex Hull

- A set is said convex if the straight line connecting any two points of the set lies entirely within A.
- Convex Hull of set S is the smallest convex set A that contains S
- The set difference A-S is called the convex deficiency of S

Computing Convex Hull

- Let B_i for i=1,2,3,4 represent the structuring elements shown below



Convex Hull

- Repeat the following equation until converge

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \quad \text{and} \quad k = 1, 2, 3, \ldots$$

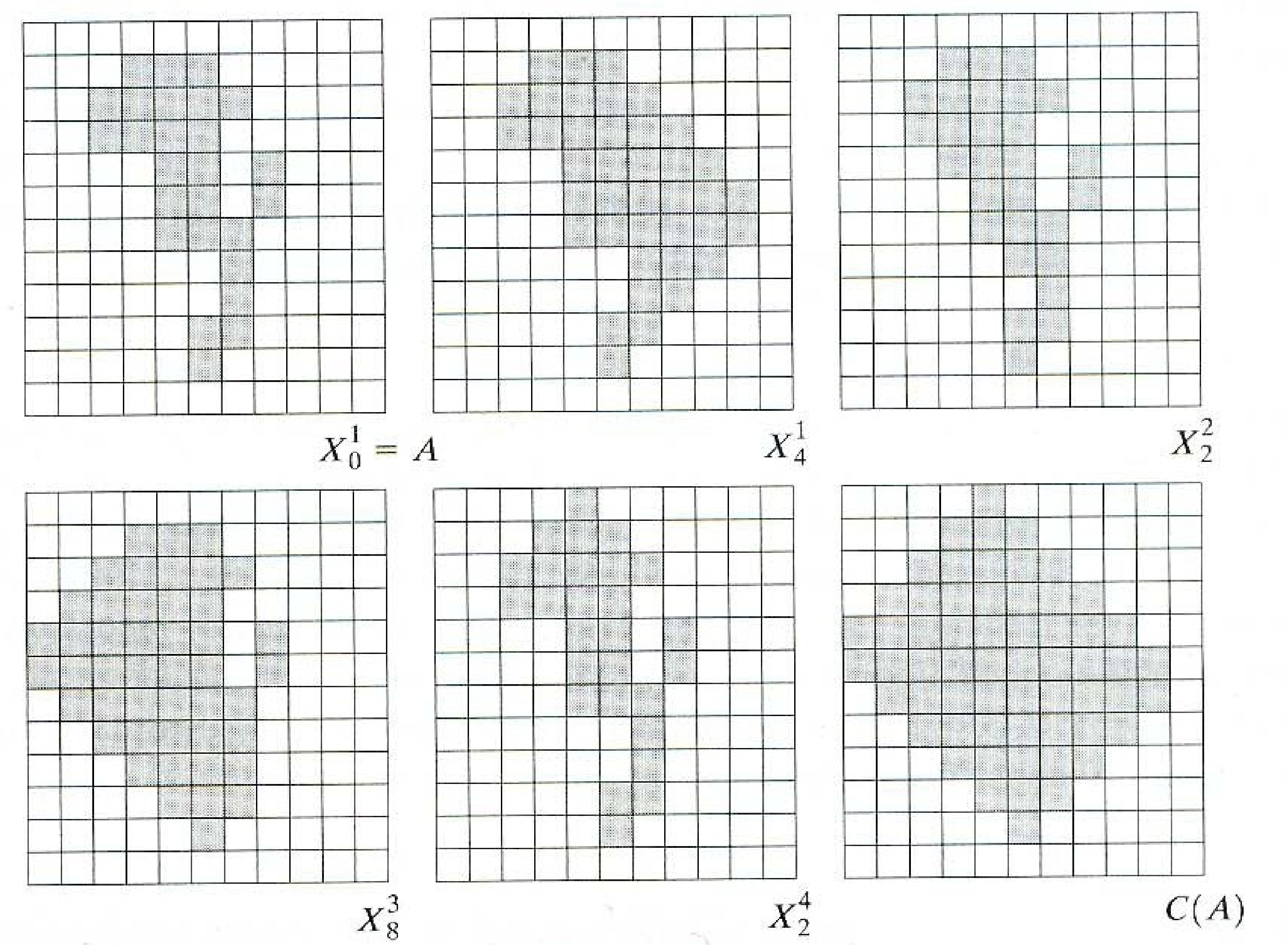
with

$$X_0^i = A$$

$\circledast$ is the Hit-or-Miss operator

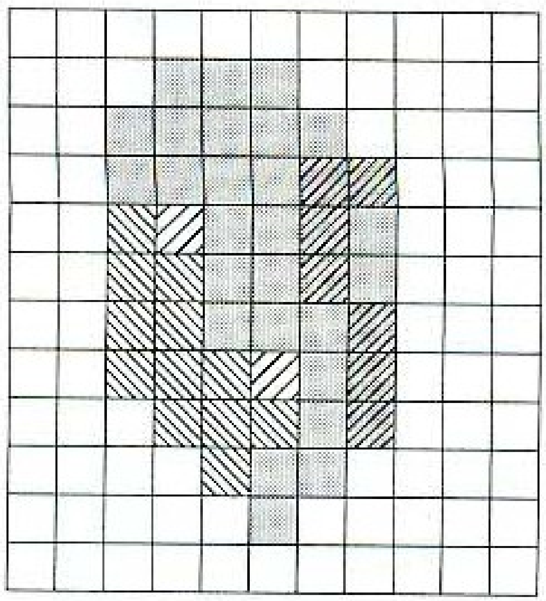
Assuming $D^i = X_{\text{conv}}^i$ Convex Hull is $C(A) = \bigcup_{i=1}^{4} D^i$

Example (Convex Hull)



Improving Convex Hull Algorithm

- The algorithm can be improved by limiting the growth of the algorithm beyond the maximum dimensions of the original set.



Thinning and Thickening

- Thinning is an image-processing operation in which binary valued image regions are reduced to lines
- The purpose of thinning is to reduce the image components to their essential information for further analysis and recognition
- Thickening is changing a pixel from 1 to 0 if any neighbors of the pixel are 1.
- Thickening followed by thinning can be used for filling undesirable holes.
- Thinning followed by thickening is used for determining isolated components and clusters.

Thinning

- Thinning is defined in terms of hit or miss as

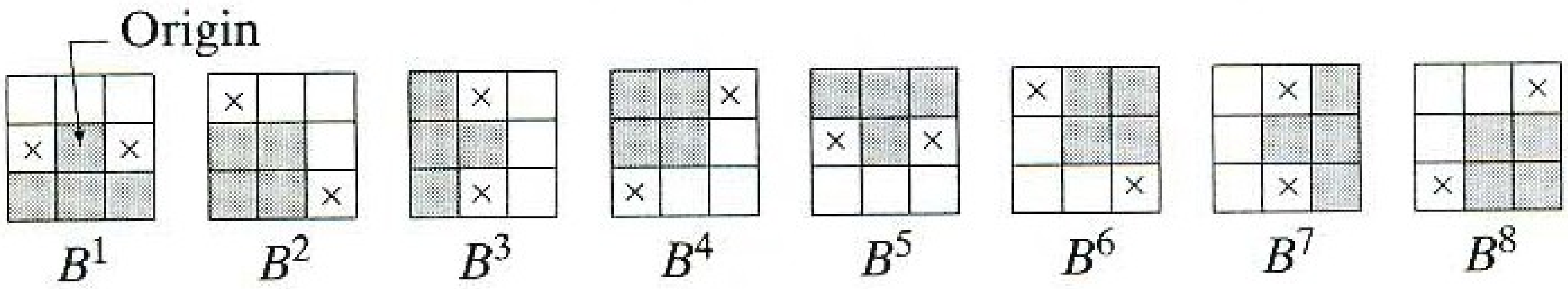
$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c.$$

where B is a sequence of structuring elements like $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$ and the operation can be given as

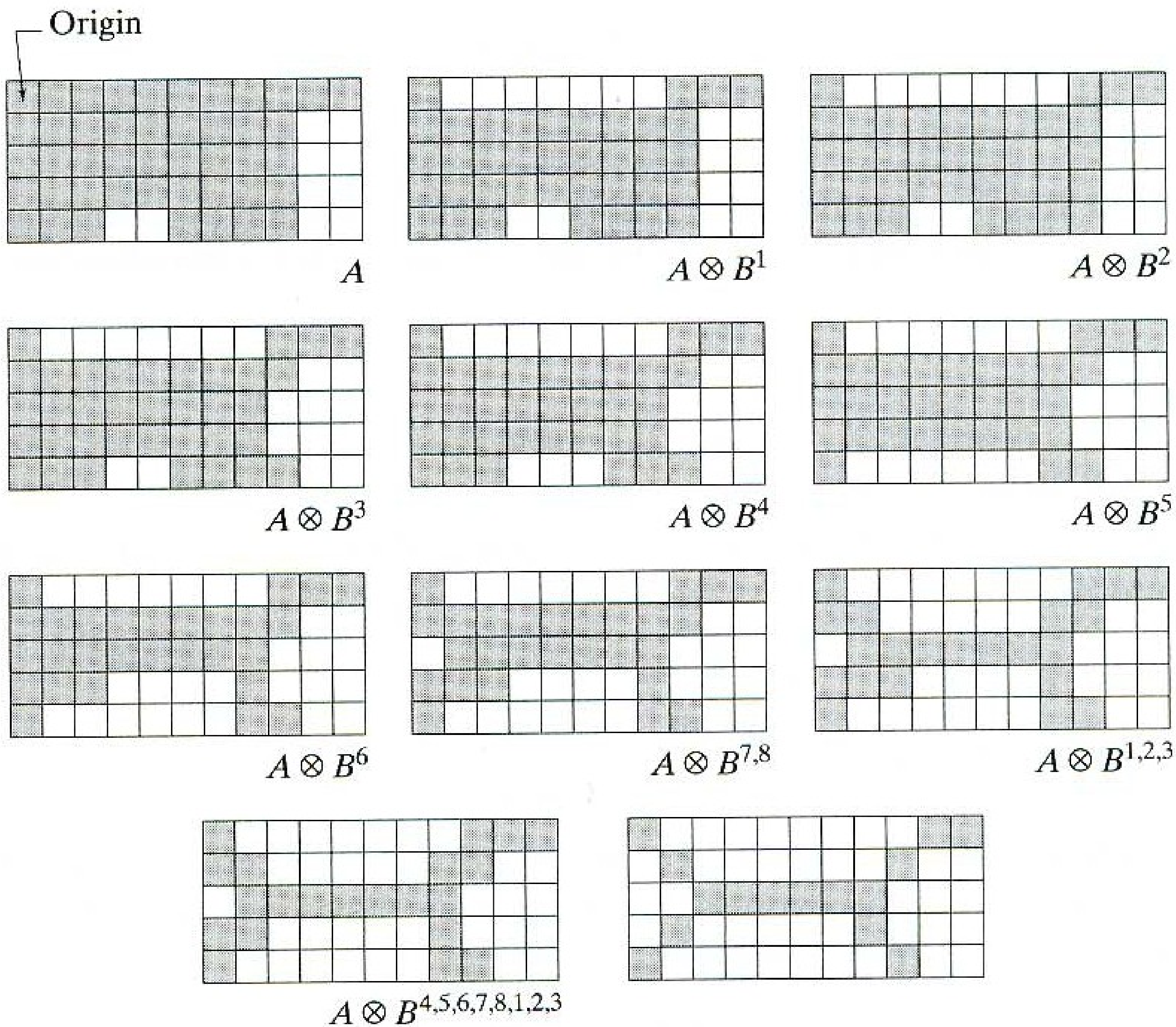
$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$$

Thinning

- Sample set of structuring elements



Thinning Example



Thickening

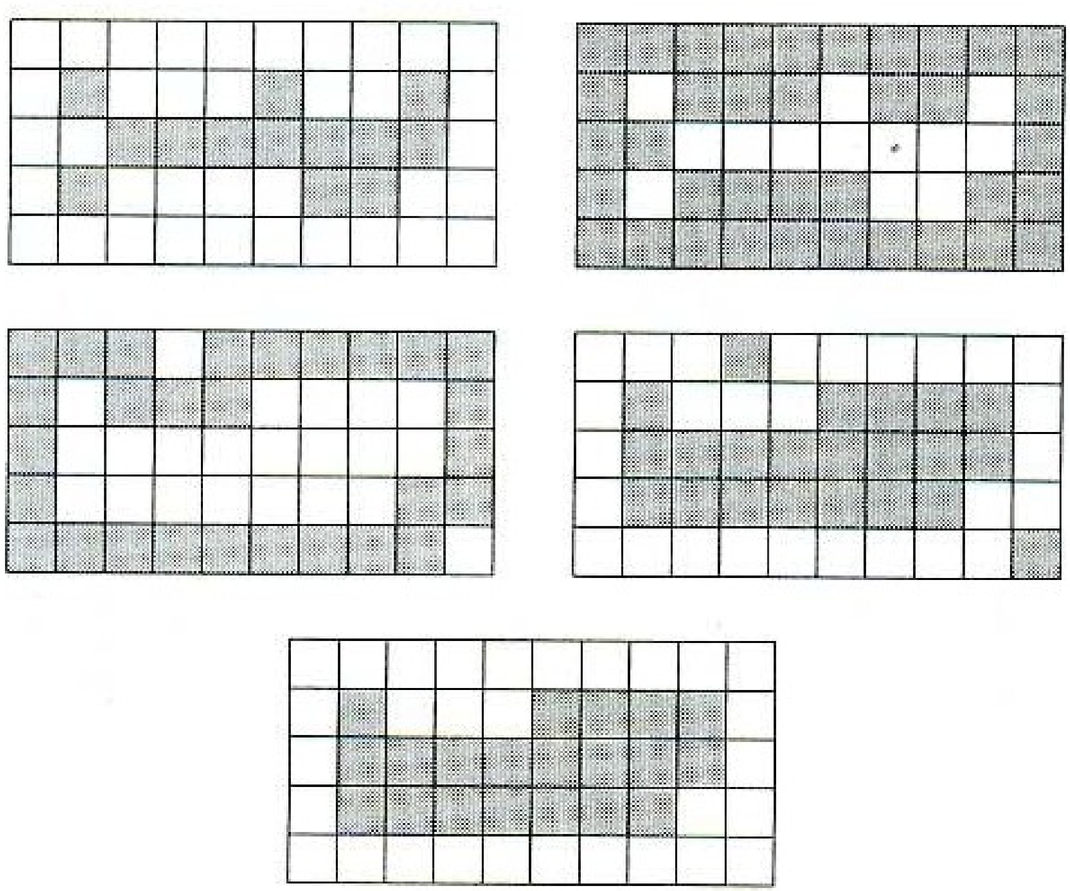
- Thickening is the morphological dual of thinning and defined as

$$A \odot B = A \cup (A \circledast B)$$

or

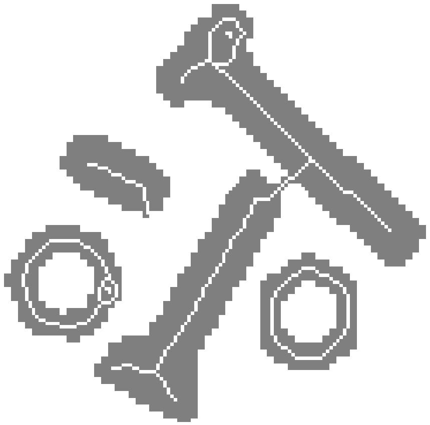
$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2) \ldots) \odot B^n)$$

Thickening Example



Skeleton

- The informal definition of a skeleton is a line representation of an object that is:
  - one-pixel thick,
  - through the "middle" of the object, and,
  - preserves the topology of the object.



Skeleton

- Skeleton is defined by

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

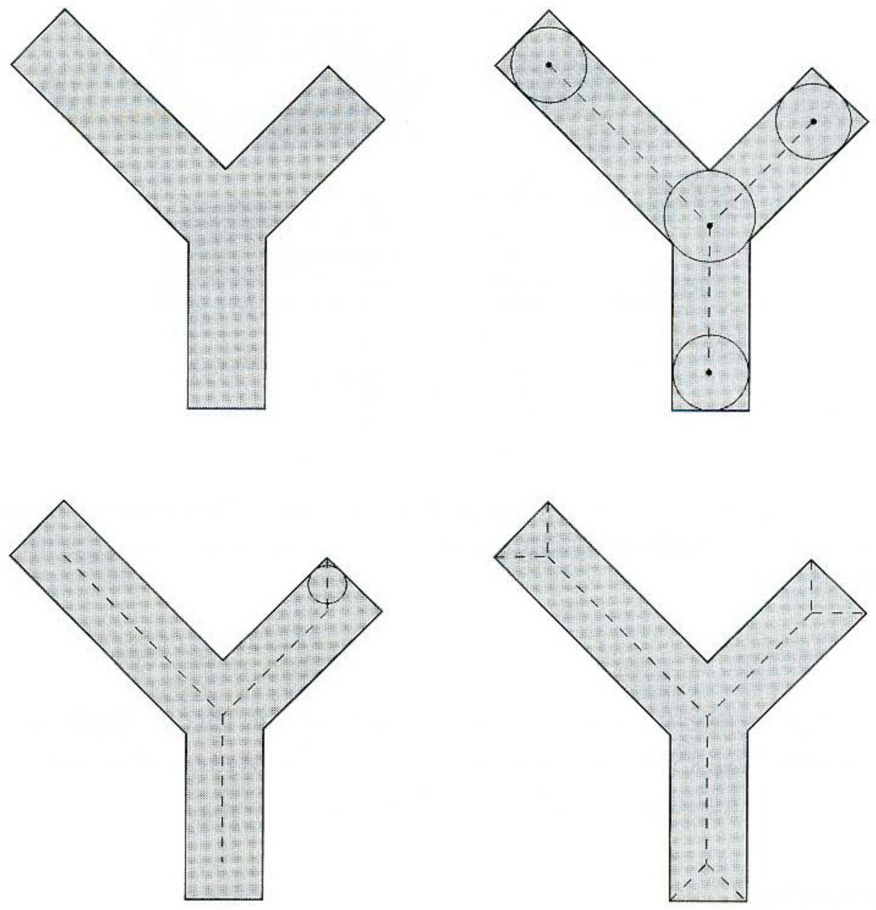
where

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

k is the last iterative step before A erodes to an empty set

$$K = \max\{k \mid (A \ominus kB) \neq \varnothing\}$$

Skeleton Example



Pruning

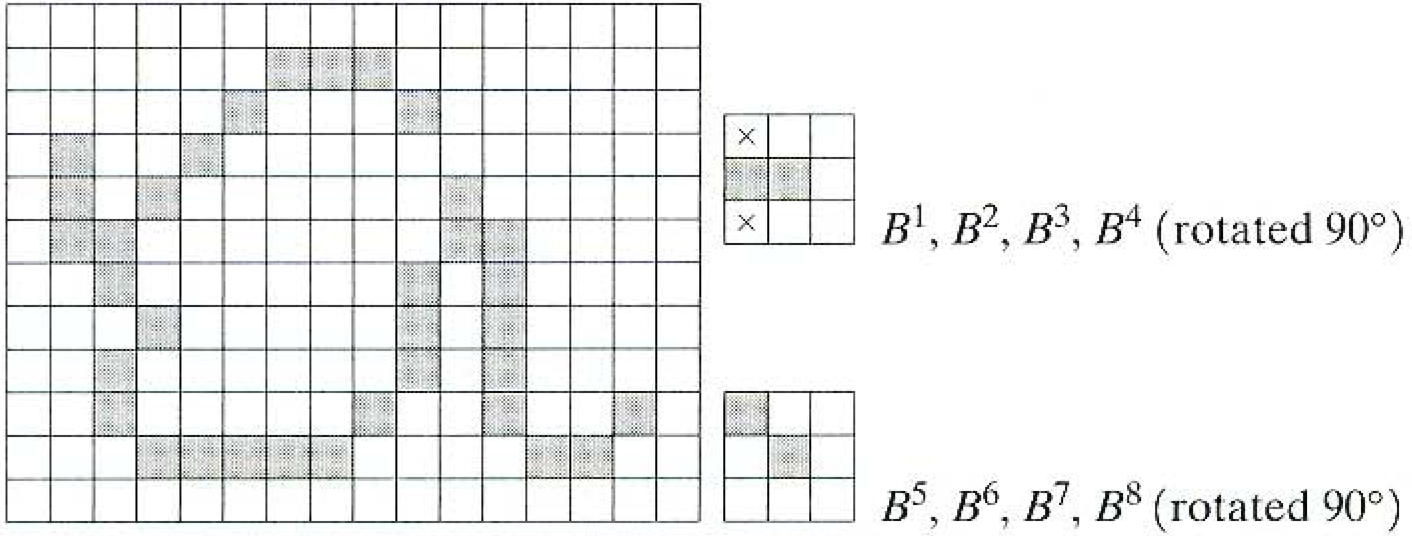
- Thinning and skeletonizing algorithms need a clean-up post-processing
- The following steps are used for pruning:

  - Thinning $$X_1 = A \otimes \{B\}$$
  - Find the end points $$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$$
  - Dilate end points $$X_3 = (X_2 \oplus H) \cap A$$
  - Find the union of X1 and X3

$$X_4 = X_1 \cup X_3$$

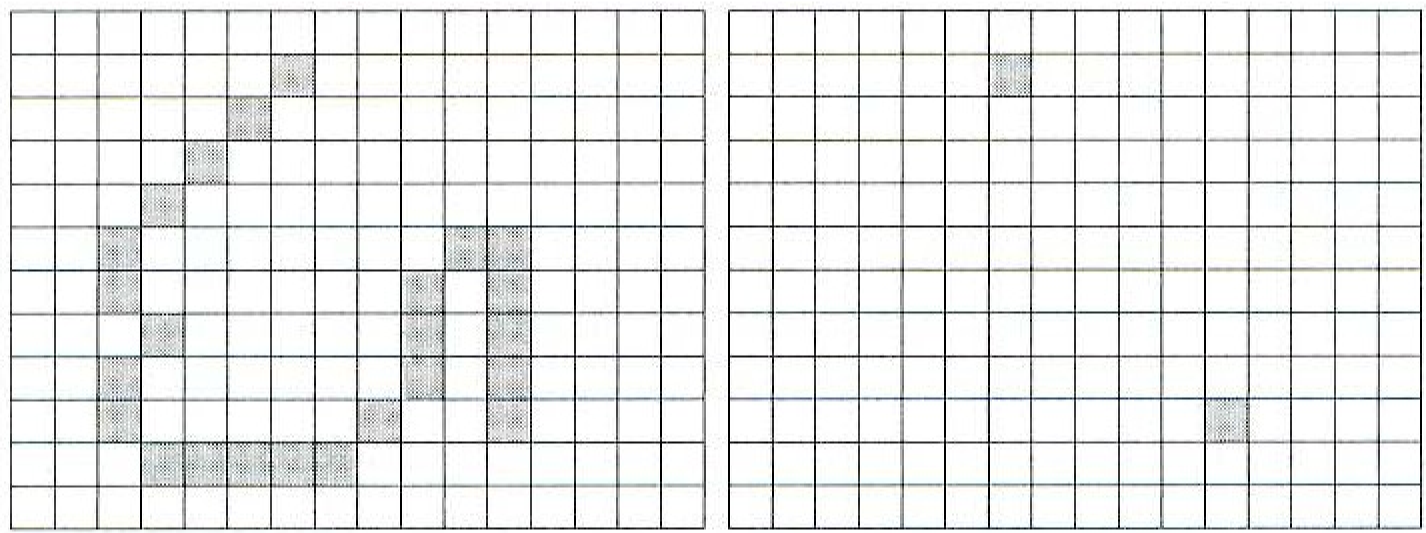
Pruning Example

- Original image and structuring elements



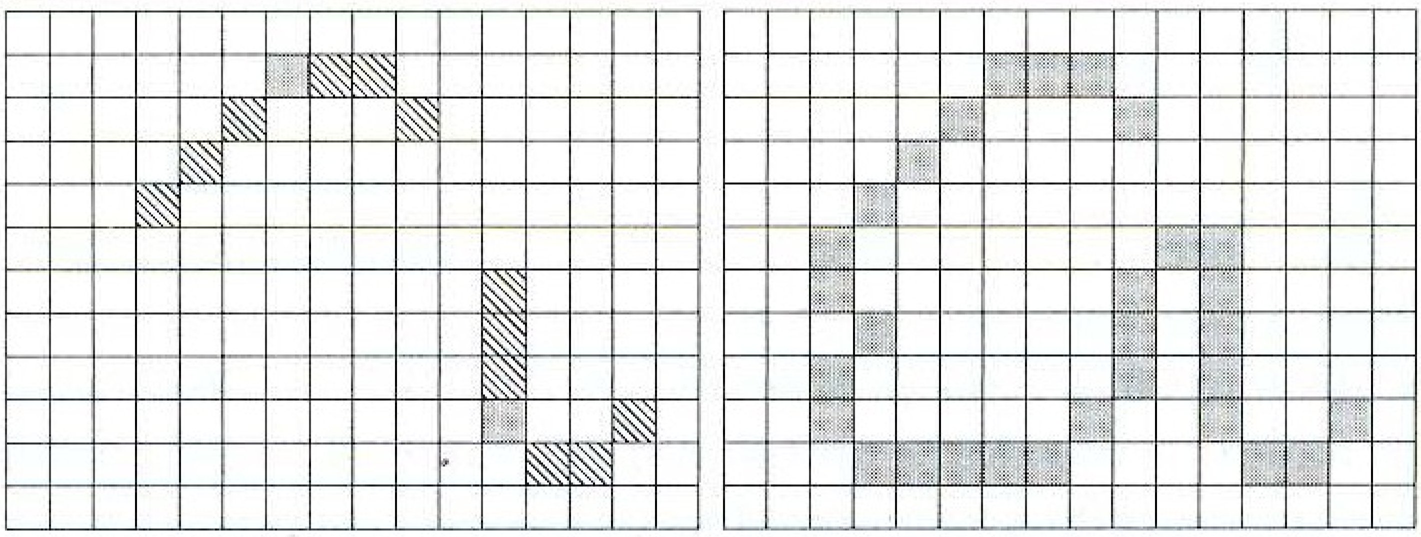
Pruning Example

- Result of thinning and end points detected



Pruning Example

- Dilation of end points and the pruned image



Extension to Gray Level

- Dilation is expressed in 1D as

$$(f \oplus b)(s) = \max\{f(s-x) + b(x) \mid (s-x) \in D_f \quad \text{and} \quad x \in D_b\}$$

- Erosion is given by

$$(f \ominus b)(s) = \min\{f(s+x) - b(x) \mid (s+x) \in D_f \ \text{and} \ x \in D_b\}$$

Extension to Gray Level (2D Case)

- Dilation

$$(f \oplus b)(s,t) = \max\{f(s-x, t-y) + b(x,y) \mid (s-x),(t-y) \in D_f; (x,y) \in D_b\}$$

- Erosion

$$(f \ominus b)(s,t) = \min\{f(s+x, t+y) - b(x,y) \mid (s+x),(t+y) \in D_f; (x,y) \in D_b\}$$

Morphological Operations in MATLAB

- To create structuring element use strel(.)

  SE = strel(*shape*, parameters)

Examples:

SE = strel('arbitrary', NHOOD)
SE = strel('diamond', R)
SE = strel('disk', R, N)
SE = strel('line', LEN, DEG)
SE = strel('octagon', R)
SE = strel('pair', OFFSET)
SE = strel('periodicline', P, V)
SE = strel('rectangle', MN)
SE = strel('square', W)

Morphological Operations in MATLAB

- SE=strel(NHOOD) is also a valid call for the function
- Use imerode(Im,SE) and imdialte(Im,SE) for erosion and dilation respectively
- Use imopen(Im,SE) and imcolose(Im,SE) for openning and closing
- For hit-or-miss use bwhitmiss(.)
  - BW2 = bwhitmiss(BW1,SE1,SE2)
    BW2 = bwhitmiss(BW1,INTERVAL)

Hit or Miss Example

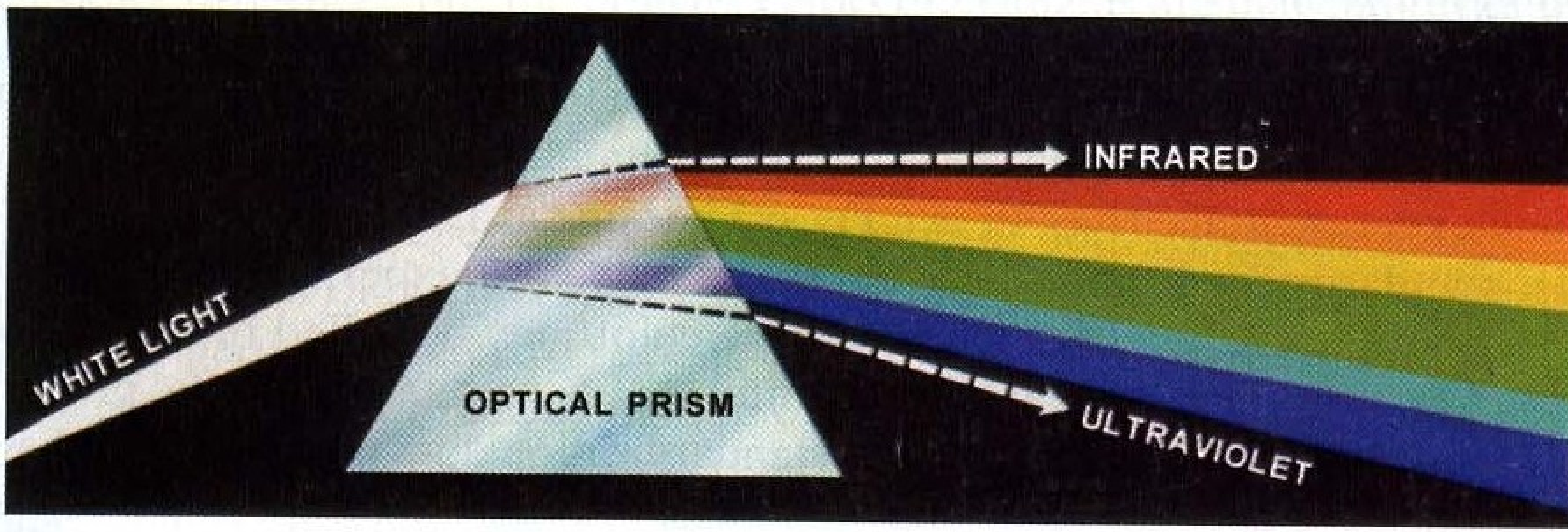
```
bw = [0 0 0 0 0 0
      0 0 1 1 0 0
      0 1 1 1 1 0
      0 1 1 1 1 0
      0 0 1 1 0 0
      0 0 1 0 0 0]

interval = [0 -1 -1
            1  1 -1
            0  1  0];

bw2 = bwhitmiss(bw,interval)

bw2 =

     0     0     0     0     0     0
     0     0     0     1     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
```

Color Representation

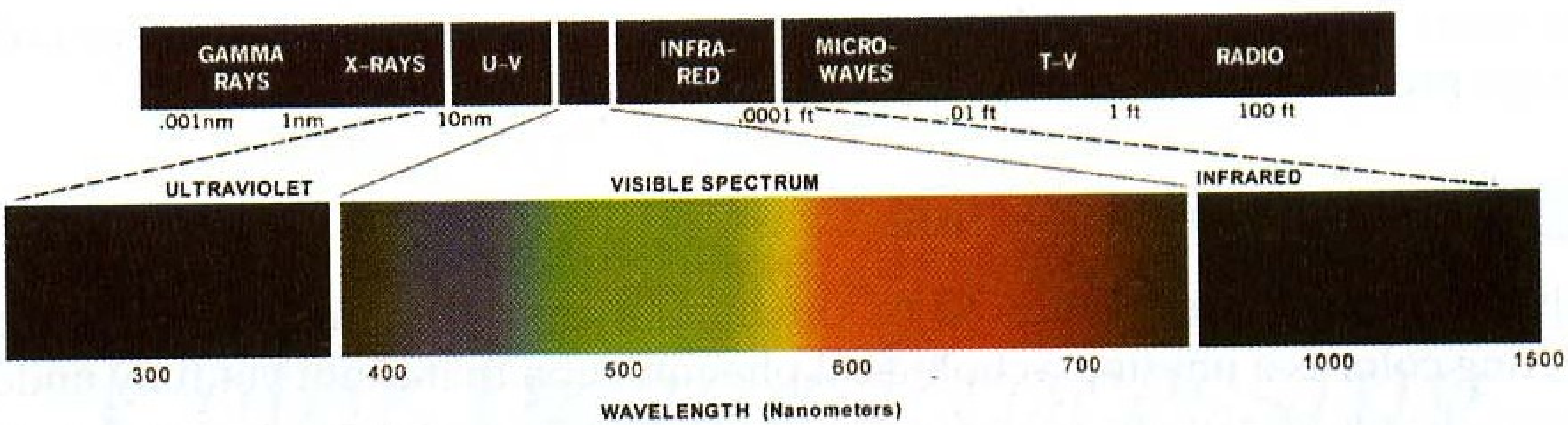
Color Representation: Motivation

- Color is an important feature for detecting and identifying objects
  - Human face detection
  - License plate detection
  - Content based image retrieval systems
- Image segmentation can be performed base on color values
- Tracking moving objects in video (surveillance)
- Etc.

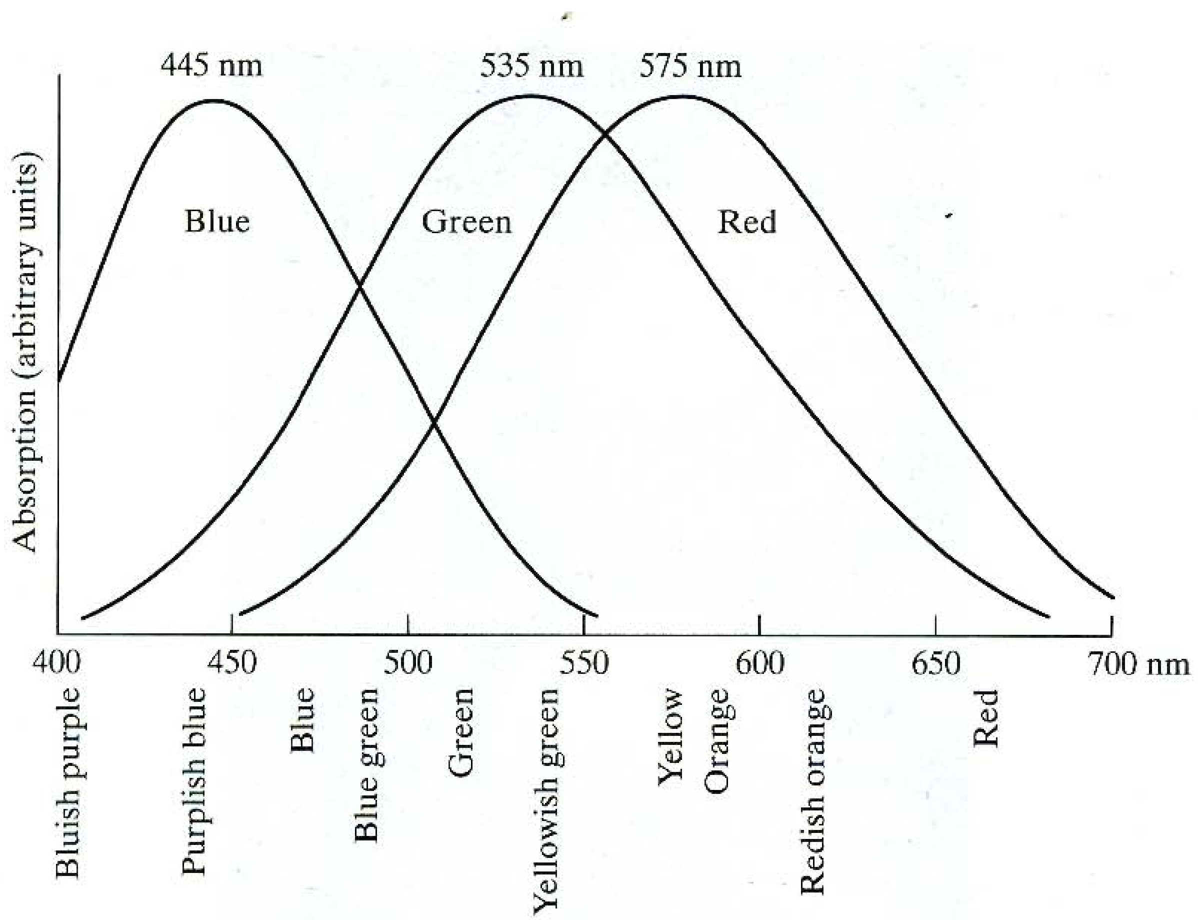
Color Spectrum



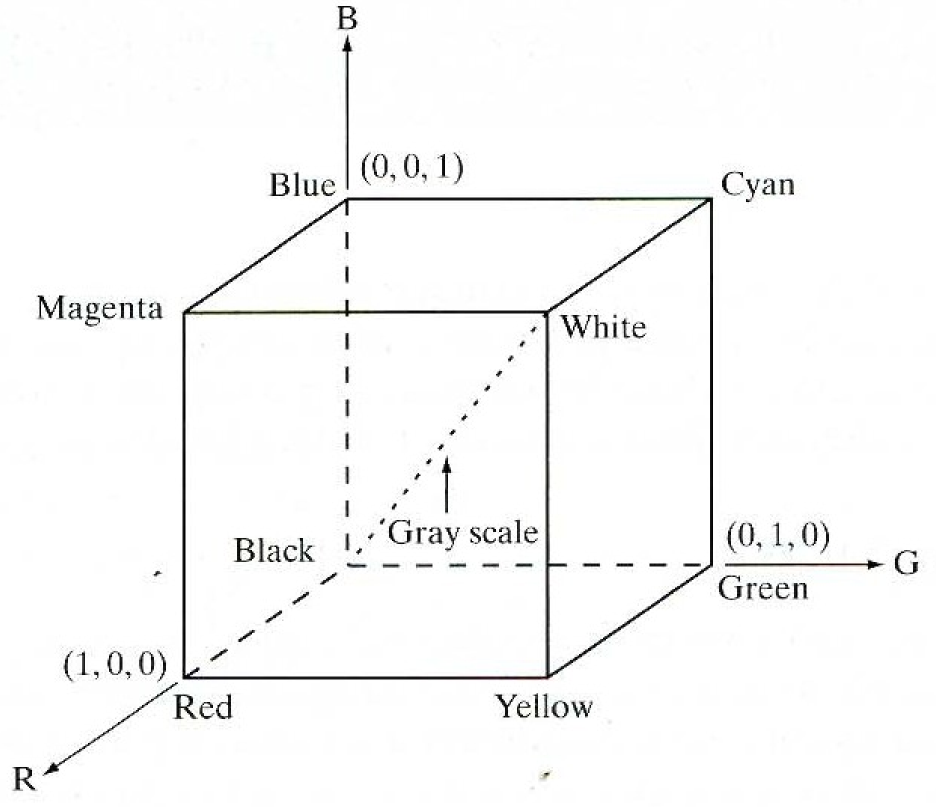
Visible Wavelength



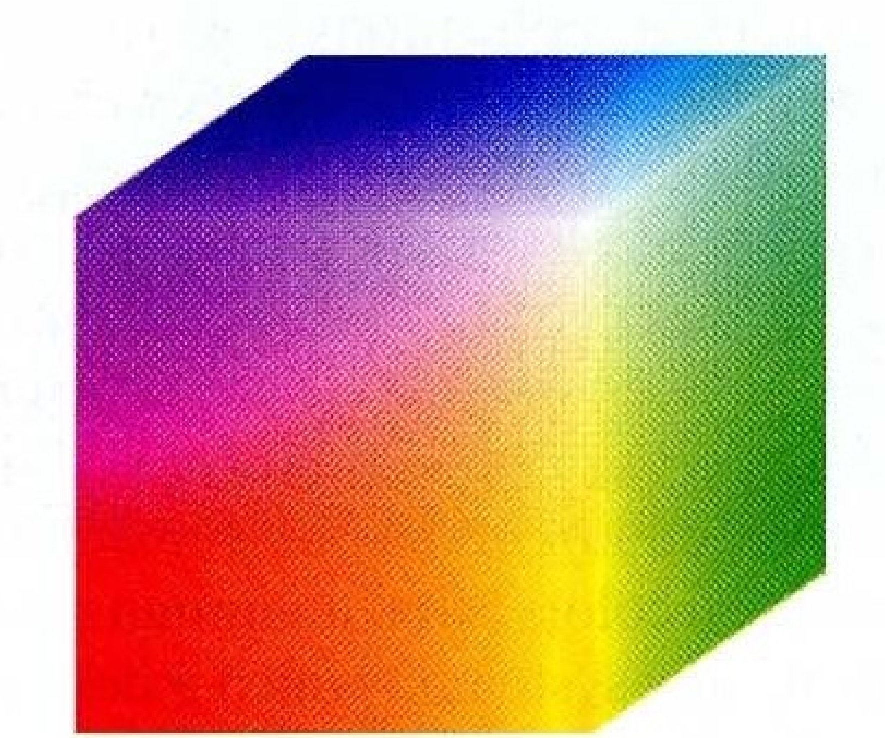
Absorption of light by human eye



Red, Green, Blue Color Cube



RGB Color Cube



YIQ Color Model

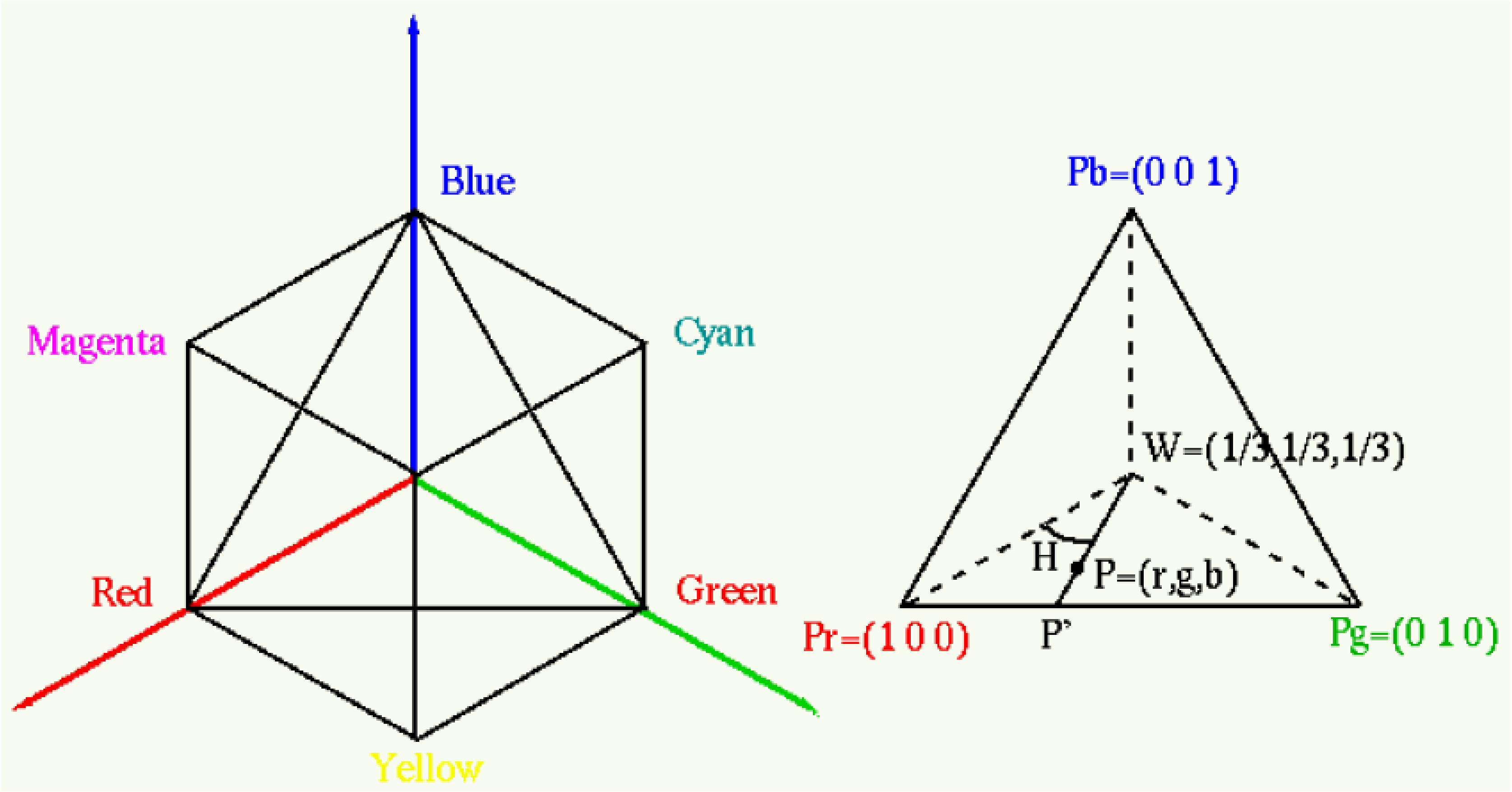
- YIQ is the color space used by the NTSC color TV system
- The Y component represents the luma information, and is the only component used by black-and-white television receivers
- I and Q represent the chrominance information
- For example, applying a histogram equalization to a color image is done by Y component only

YIQ Color Model Conversion

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.9563 & 0.6210 \\ 1 & -0.2721 & -0.6474 \\ 1 & -1.1070 & +1.7046 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

HSI (or HSV) Color Model



Conversion from RGB to HSI

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

with

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

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

$$I = \frac{1}{3}(R + G + B)$$

Enhancement using HSI Model

Color Space Conversions in MATLAB (1)

- To convert from RGB to HSV use rgb2hsv
- For HSV to RGB conversion use hsv2rgb
- E.g.
  - I = imread('test.bmp');
  - H = rgb2hsv(I);

Color Space Conversions in MATLAB (2)

- To convert from RGB color space into YIQ color space use rgb2ntsc
  - RGB = imread('sample.png');
  - YIQ = rgb2ntsc(RGB);
- ntsc2rgb convert from YIQ to RGB

- Images can be displayed in RGB space only

Color Space Conversions in MATLAB (3)

- I component in YIQ (the first component) is equivalent to a gray scale conversion. However, you may also use rgb2gray either.
  - I=imread('test.bmp');
  - YIQ = rgb2ntsc(I);
  - imshow(YIQ(:,:,1))

Questions?