Image Enhancement Techniques

Eun Yi Kim

Artificial Inteligence
& Computer Vision
Laboratory

Basic Techniques in DIP

- Image Representation and Storage

- image enhancement & filtering
  - To improve the delectability of important image details or objects
  - Contrast stretching, smoothing, sharpening

- Feature extraction
  - Color : color space, histogram
  - Texture: Wavelet trans, Discrete cosine trans.
  - Shape: Edge detector, HOG, SIFT/SURF

I N D E X

Artificial Inteligence
& Computer Vision
Laboratory

Color

- Used heavily in human vision
- Color is a pixel property, making som e recognition problems easy
- Visible spectrum for humans is 400 n m (blue) to 700 nm (red)
- Machines can “see” much more;

  ex. X-rays, infrared, radio waves

Coding methods for humans

- RGB is an additive system (add colors to black) used for displays.
- CMY is a subtractive system for printing.
- HSI is a good perceptual space for art, psychology, and recognition.
- YIQ used for TV is good for compression.

Editing saturation of colors



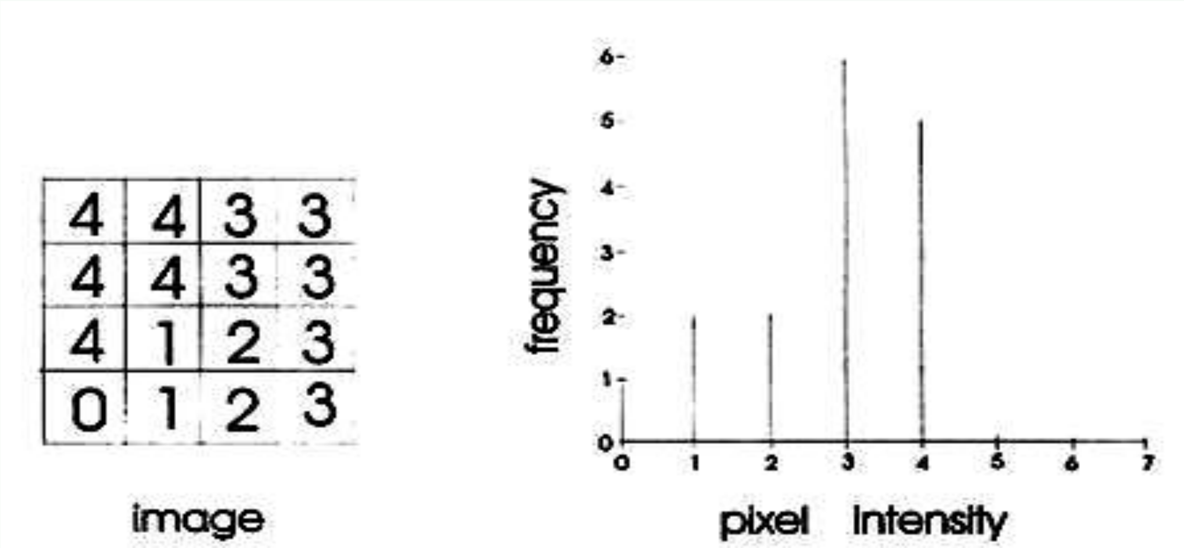
- (Left) Image of food originating from a digital camera;
- (center) saturation value of each pixel decreased 20%;
- (right) saturation value of each pixel increased 40%.

Histogram

Artificial Inteligence
& Computer Vision
Laboratory

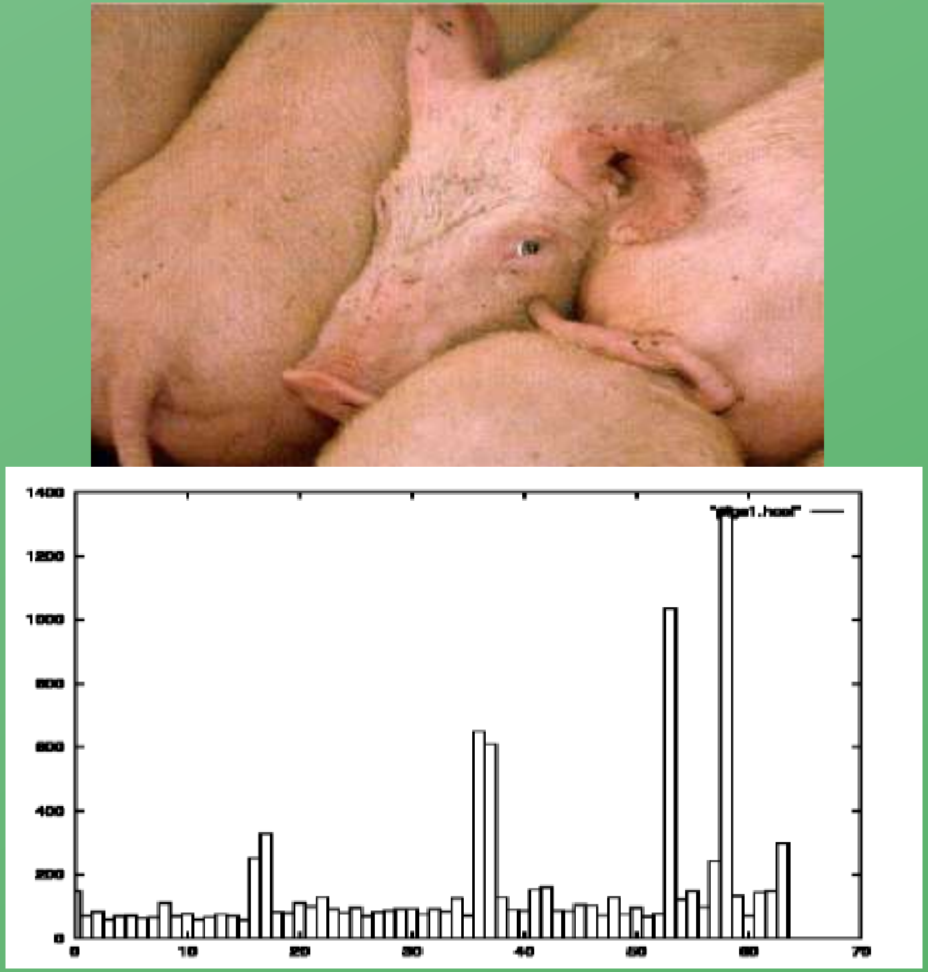
- Color histogram can represent an image
- Histogram is fast and easy to compute.
- Size can easily be normalized so that different image histograms can be compared.
- Can match color histograms for database query or classification.

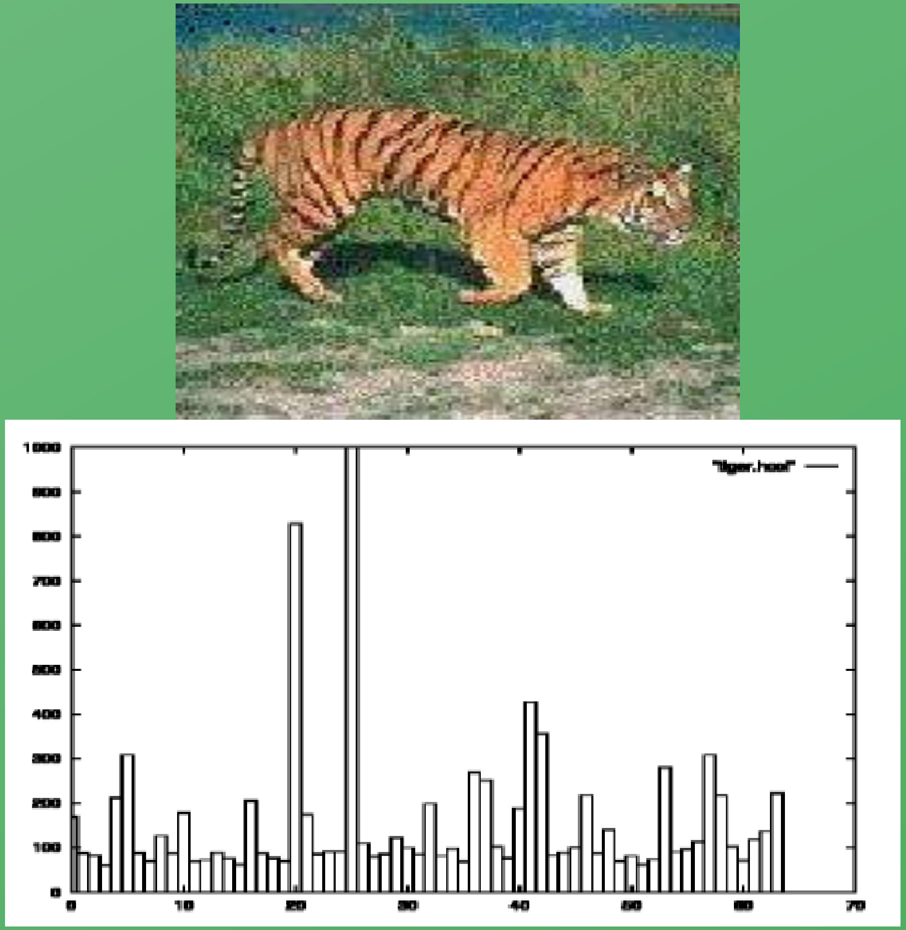
Histogram



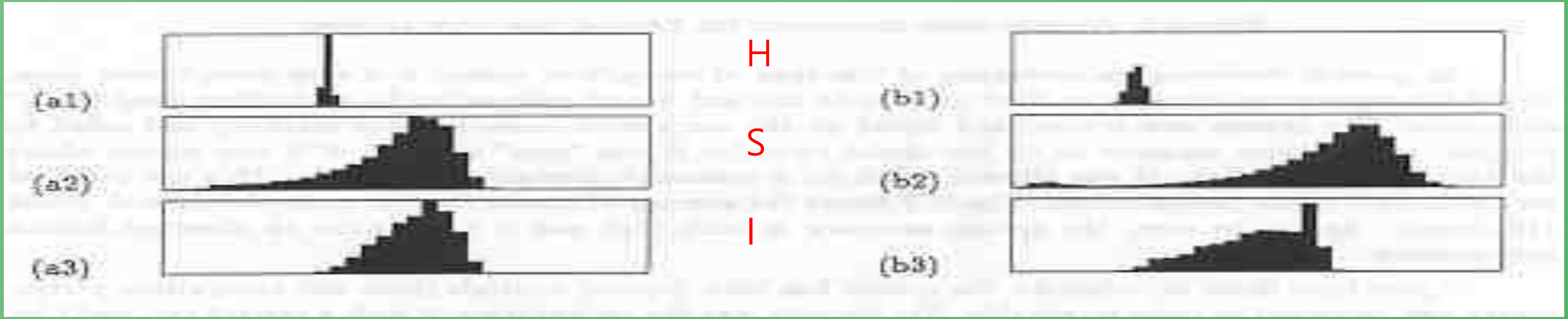
Sample image with histogram

Histograms of two color images





Apples versus Oranges



Separate HSI histograms for apples (left) and oranges (right) used by IBM’s VeggieVision for r
ecognizing produce at the grocery store checkout station.

Contrast Enhancing

Point Processing

Artificial Inteligence & Computer Vision Laboratory

- The simplest kind of range transformations are these independent of positi on x,y:

$$G(x,y) = \mathrm{T}(f(x,y))$$

- This is called point processing.

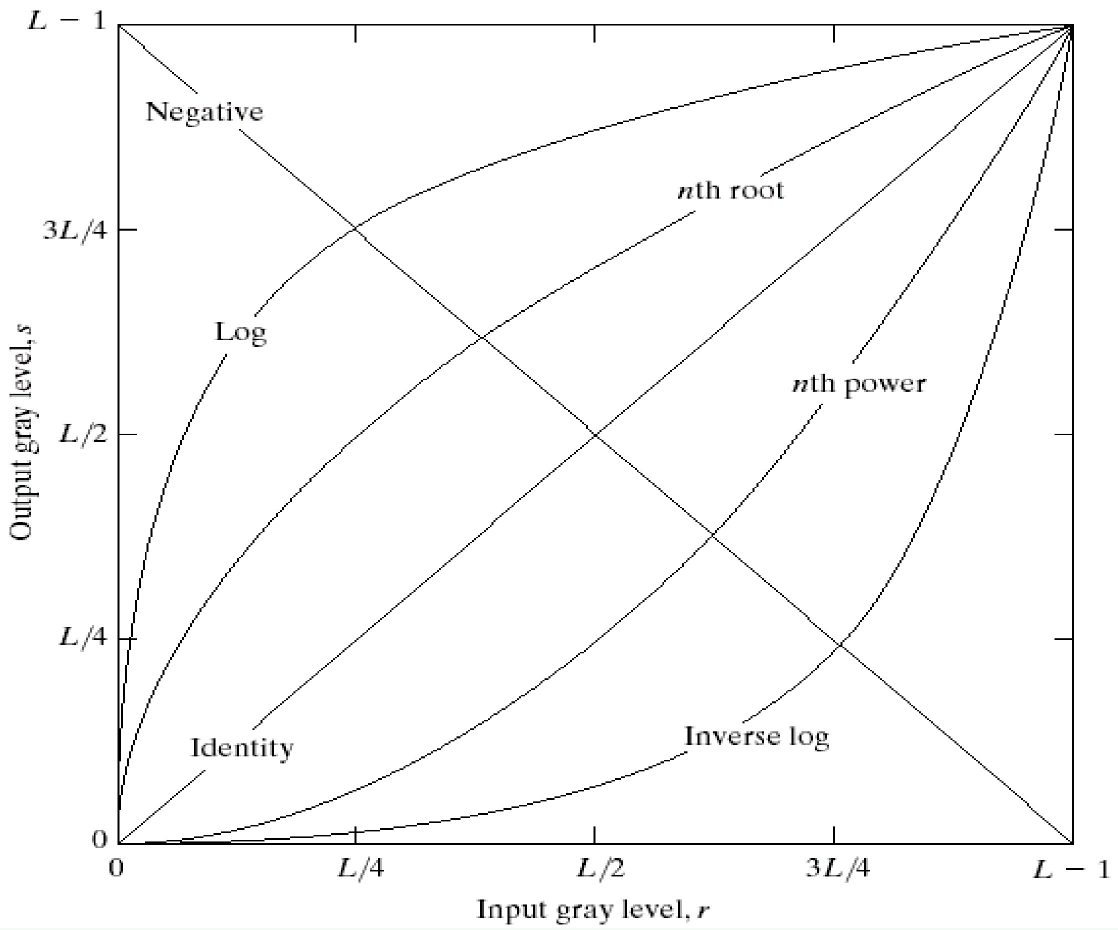
- What can they do?
- What's the form of *T*?

- **Important:** every pixel for himself – spatial information completely lost!

Point Processing

- Basic Point Processing

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Point Processing

Artificial Inteligence & Computer Vision Laboratory

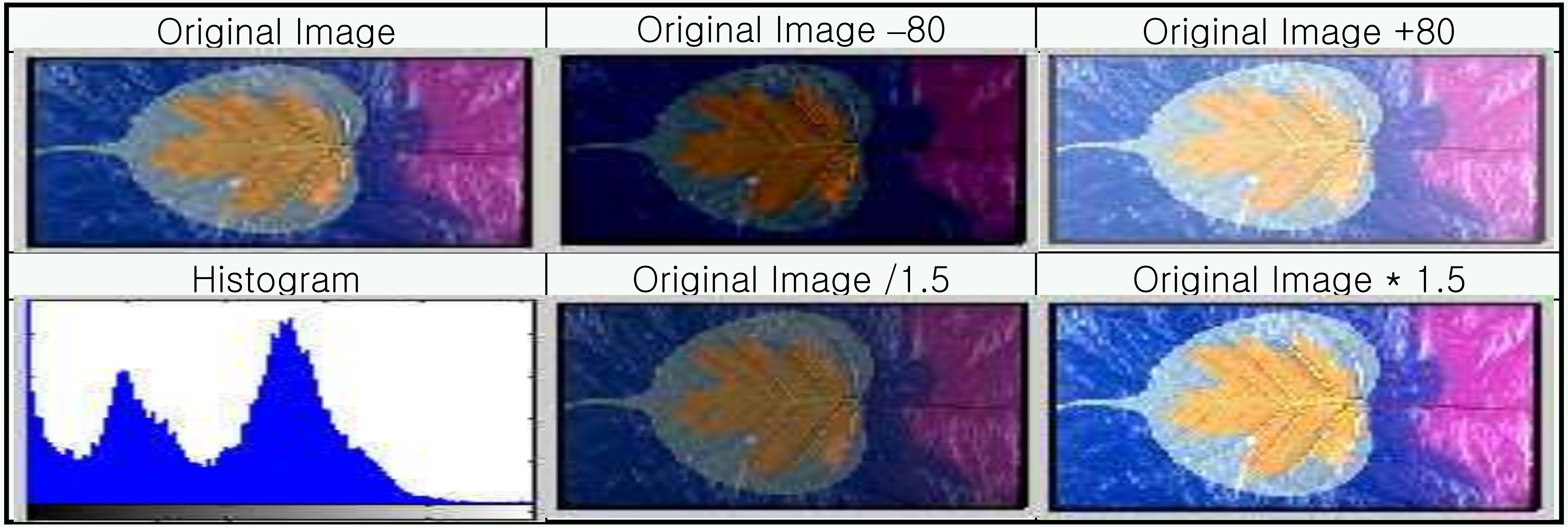
- Arithmetic/logic operations
  - Image addition, subtraction, multiplication, division and averaging

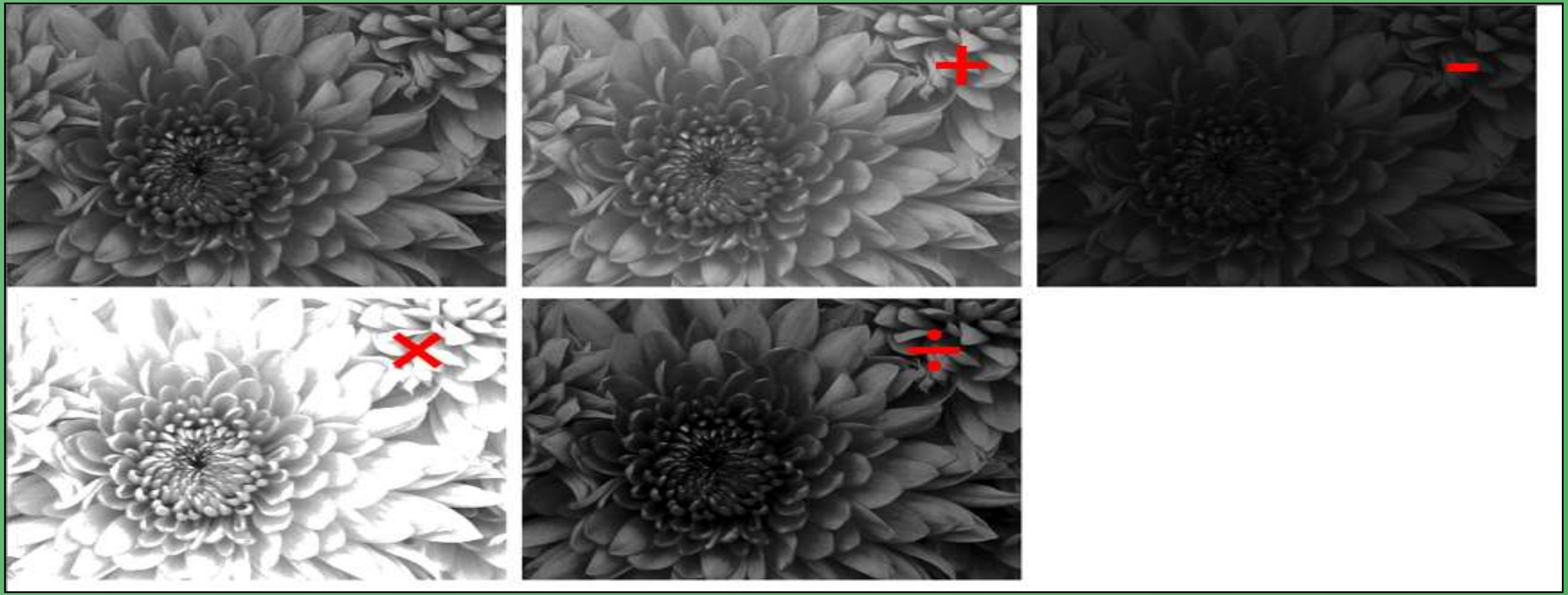
- Simple gray level transformations
  - Image negatives
  - Log transformations
  - Power-law transformations
  - Thresholding
  - Gray-level slicing, Bit-plane slicing
  - **Contrast stretching**

- Histogram processing
  - Histogram equalization

Arithmetic operations

- 화소값에 일정한 상수를 더하고, 빼고, 곱하고, 나누는 연산
  - +, - : 영상의 밝기를 밝게 하거나 어둡게 한다.
  - ×, ÷ : 영상의 명암대기를 높이거나 낮춘다.
- 문제: 음수 또는 화소가 가지는 최대값보다 큰값이 출력가능
- 해결:Clamping : 음수는 0, 255보다 큰값은 화소값 255로 설정





Power-law transformations

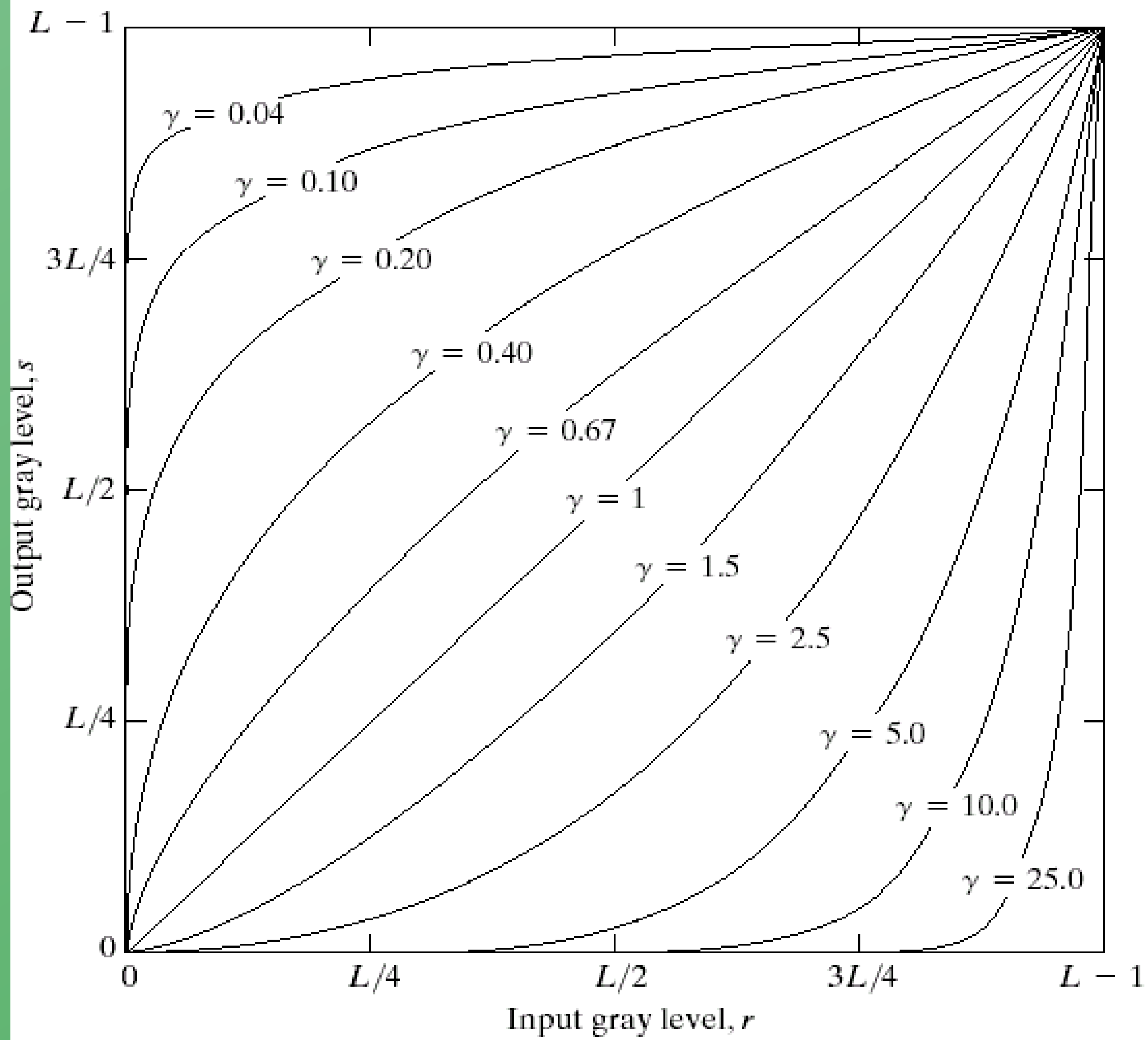
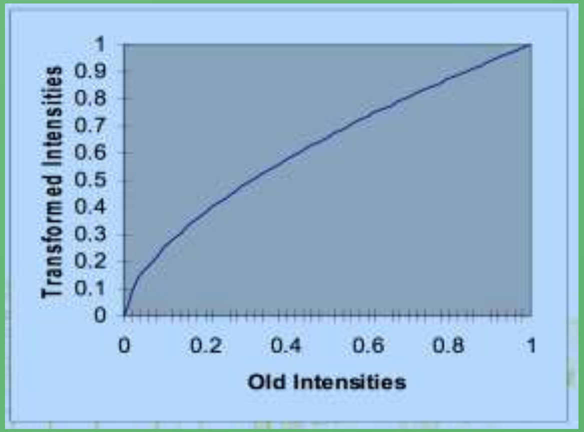
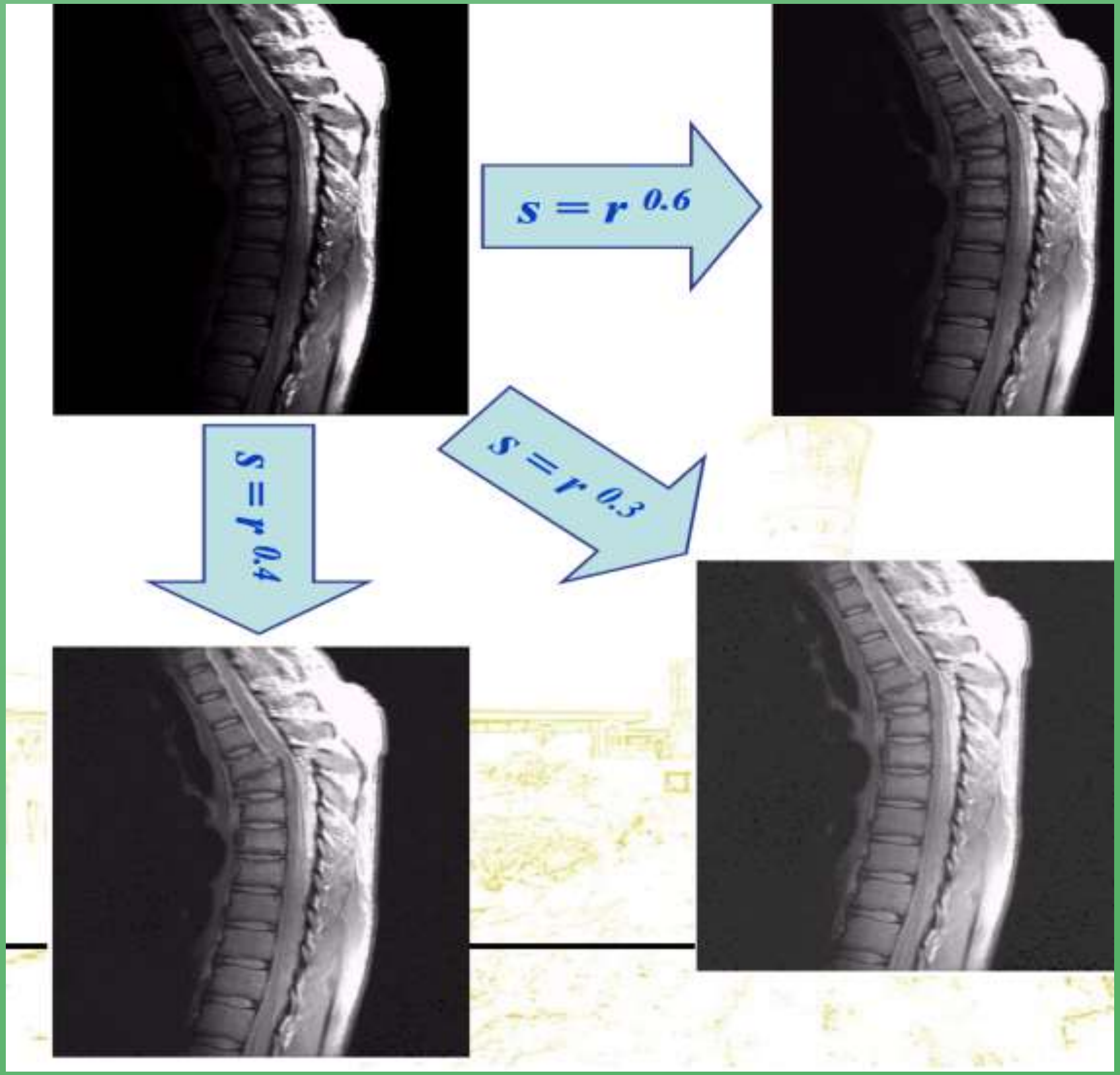


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

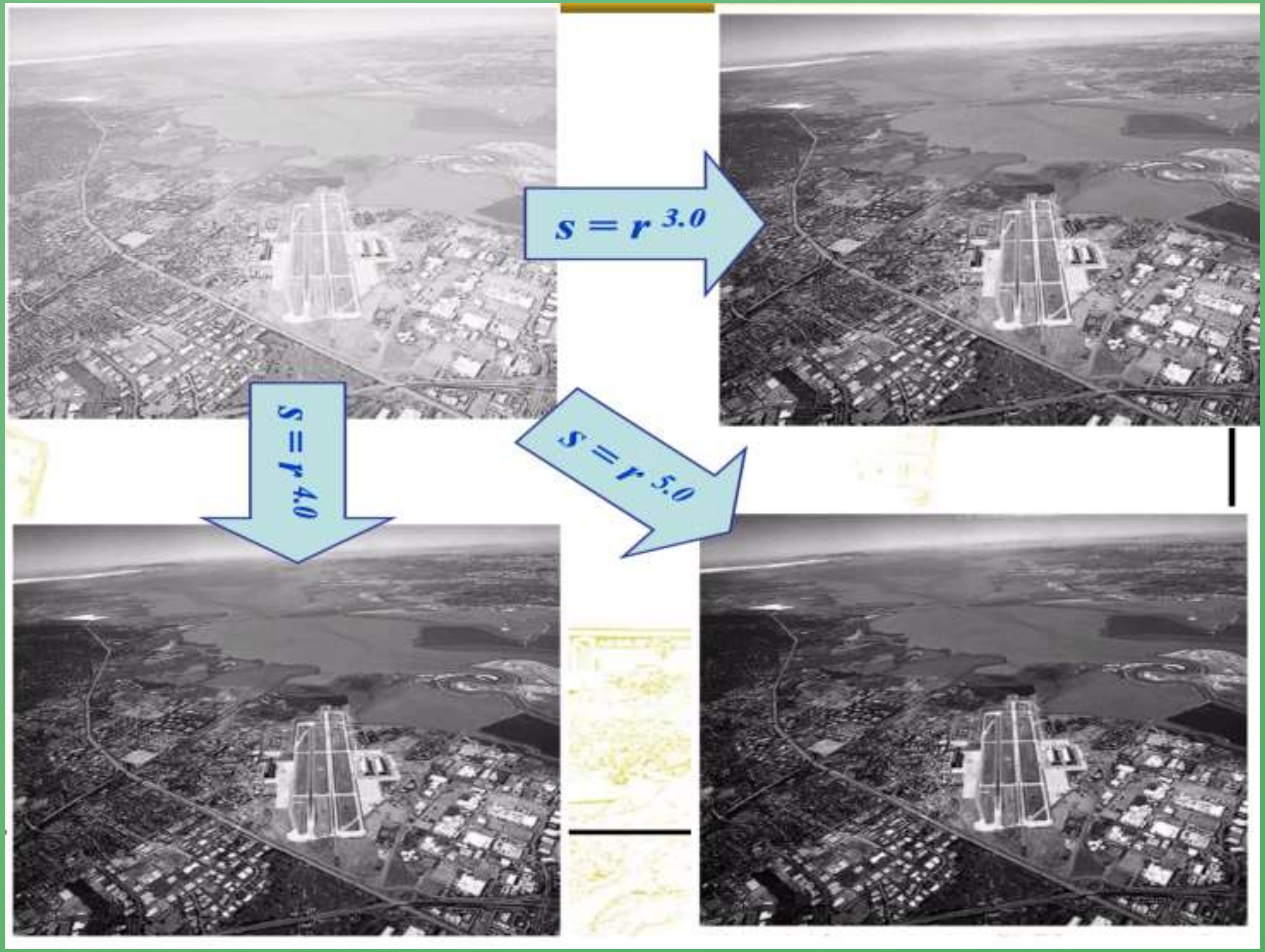
Power Law Example



Power Law Example



Power Law Example

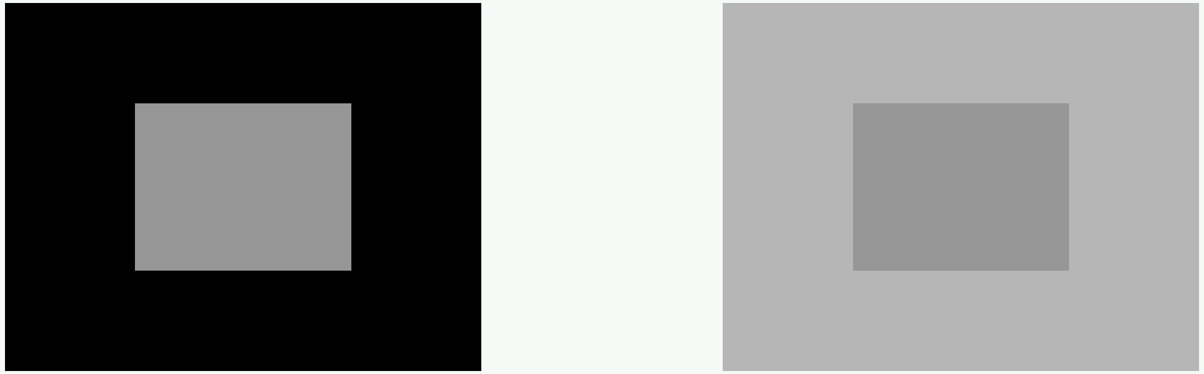


Contrast

- 대비(contrast)
  - 가장 어두운 영역으로부터 가장 밝은 영역의 범위

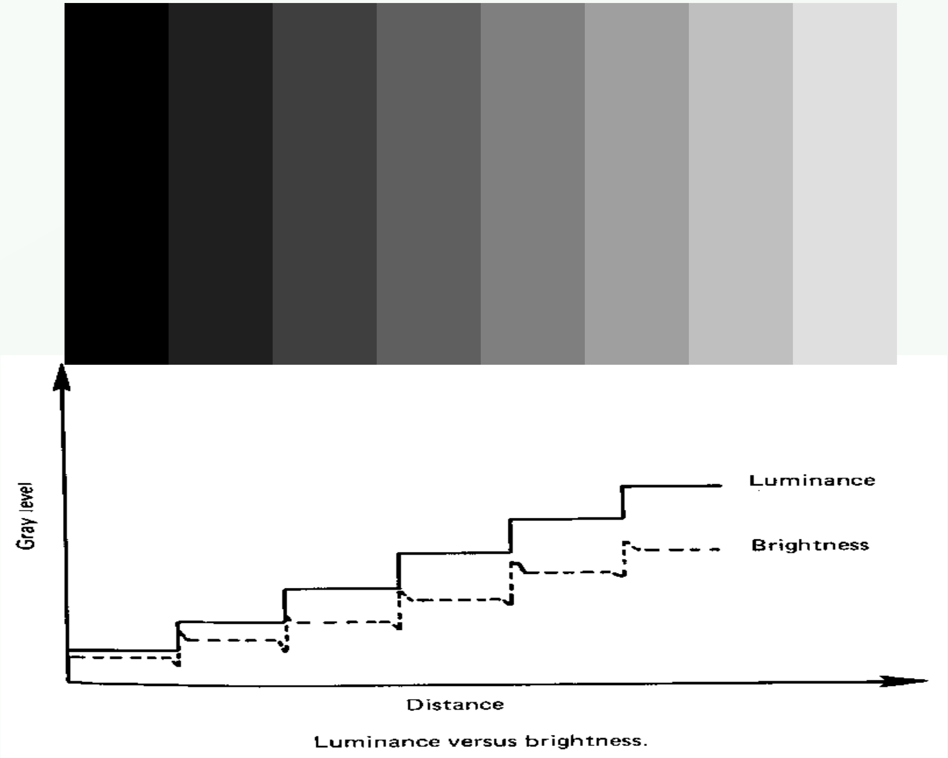
$$\boxed{Contrast = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}}$$

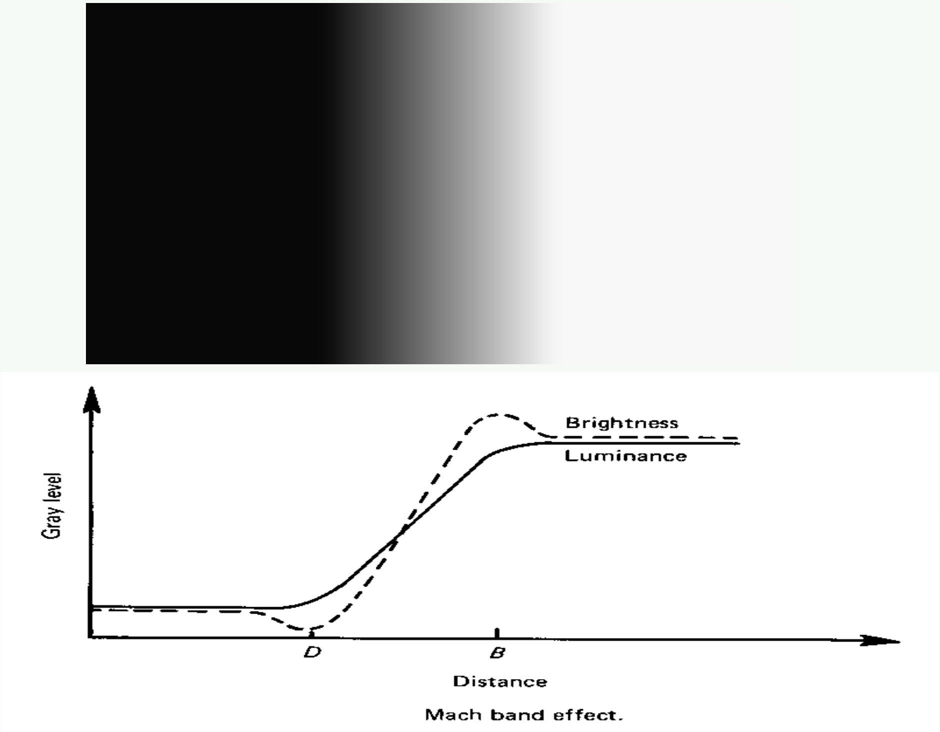
  - 지각 작용은 순수한 광도의 강도에 민감하기보다는 광도의 대비에 더 민감하다(spatial filtering of eyes).



Contrast

- Mach Band:서로 다른 광도가 인접해 있는 경우 발생하는 효과
  - 광도가 급격히 변화하는 데에 대한 시각 시스템의 반응은 경계부분을 강조하여 보는 경향이 있음





Contrast Stretching

- A contrast stretching operator is a point processing that uses a piecewise smooth function T(f{x,y)) of the input gray level to enhance import details of the image

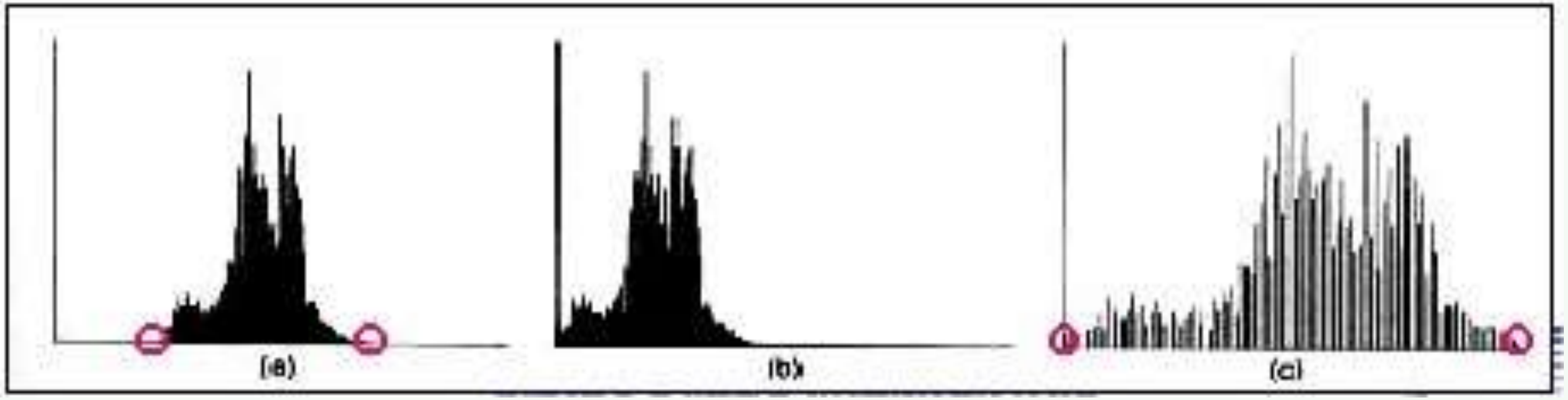
- Basic contrast stretching
- Ends-in-search
- Simple transformation function
- Histogram processing
  - Histogram equalization, histogram specification

Contrast Stretching

- 기본 명암 대비 스트레칭
  - 특정부분, 중앙에 명암값이 치우치는 히스토그램을 가진 영상에 가장 잘 적용
  - 모든 범위의 화소값을 포함하도록 영상을 확장

$$new\ pixel = \frac{old\ pixel - low}{high - low} * 255$$

(a) 입력영상
(b) 히스토그램-low
(c) (히스토그램-low)*255/(high-low)

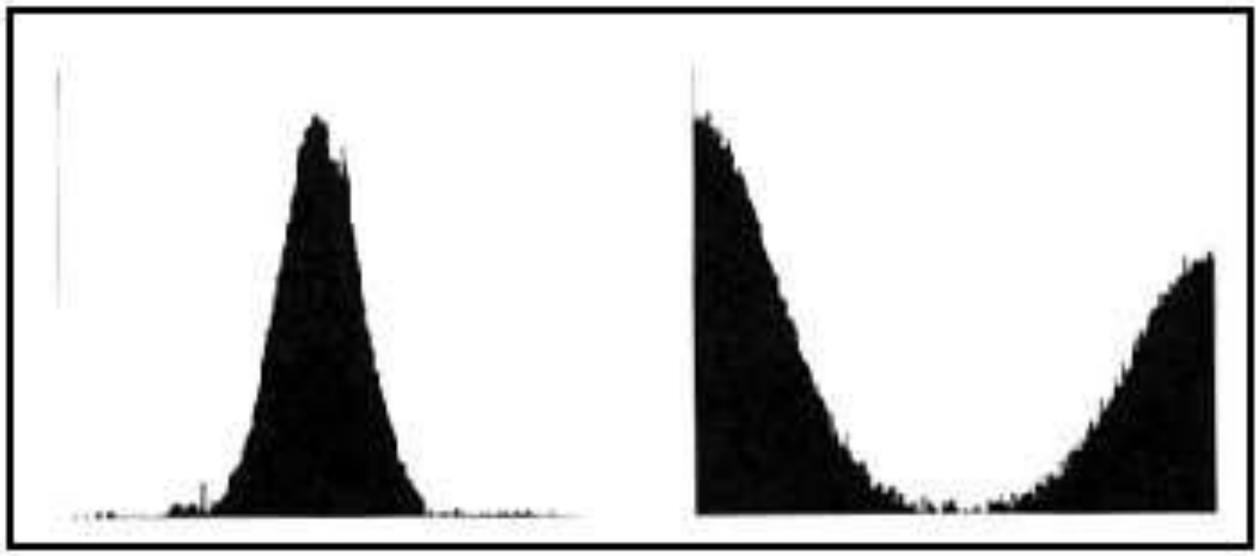


Original Image

Basic Contrast Stretching

Contrast Stretching

- **기본 명암 대비 스트레칭**
  - 낮은 명암대비를 가진 영상의 질을 향상시킬 수있는 유용한 도구로서 가우시안 (Gaussian) 분포를 가질때 가장 잘 적용

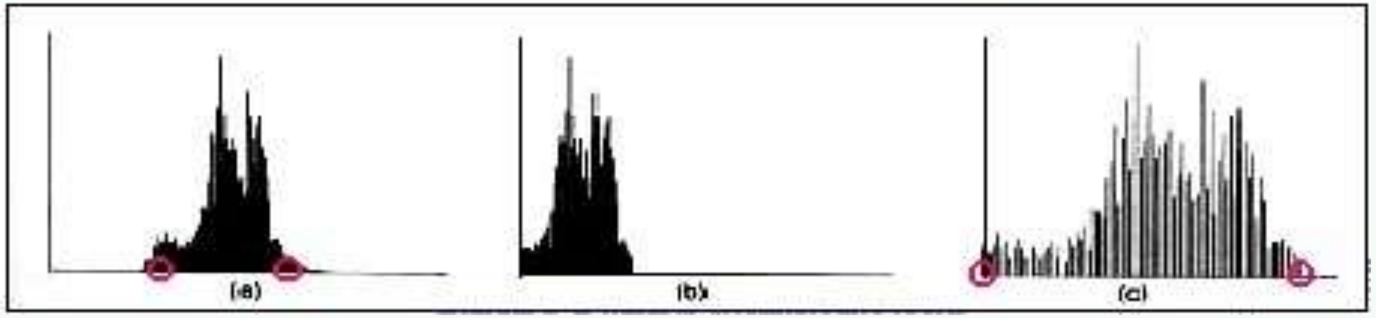


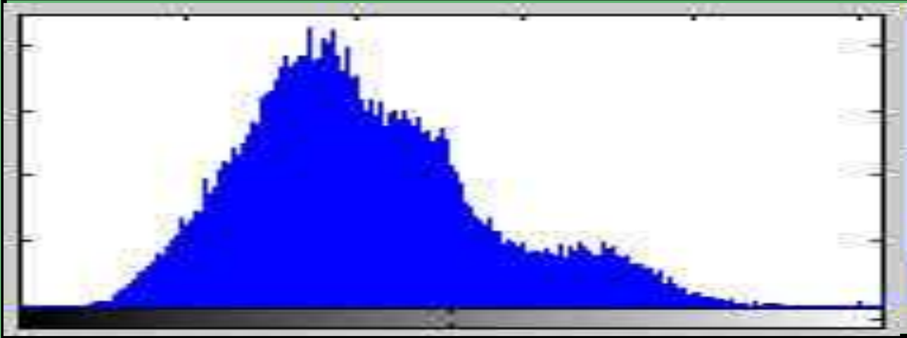
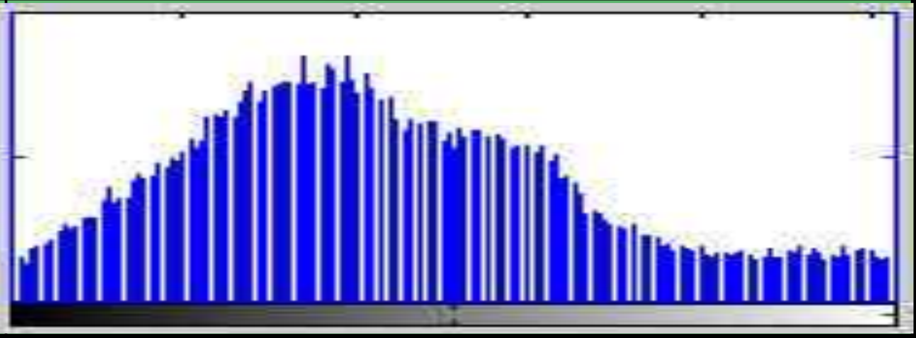
낮은 명암 대비와 높은 명암 대비를 가진 히스토그램

Contrast Stretching

- 엔드인탐색(ends-in search)
  - **모든범위의 명암값을 갖지만 히스토그램의 특정 부분에 화소들이 치우친 영상에 가장 잘 적용**
    - 일정한 양의 화소를 흰색 또는 검은색을 갖도록 지정
    - 알고리즘: 2개의 임계값(low, high)을 사용
      - low : 낮은 범위에서 지정한 양 이후의 화소의 pixel intensity
      - high: 높은 범위에서 지정한 양 이후의 화소의 pixel intensity

$$output(x) = \begin{cases} 0 & for\ x \le low \\ 255 * (x - low) / (high - low) & for\ low < x \le high \\ 255 & for\ high \le x \end{cases}$$



Original Image	Ends-In Searched Image
	
Histogram	Histogram
	

Original Image

End-in Search

Contrast stretching

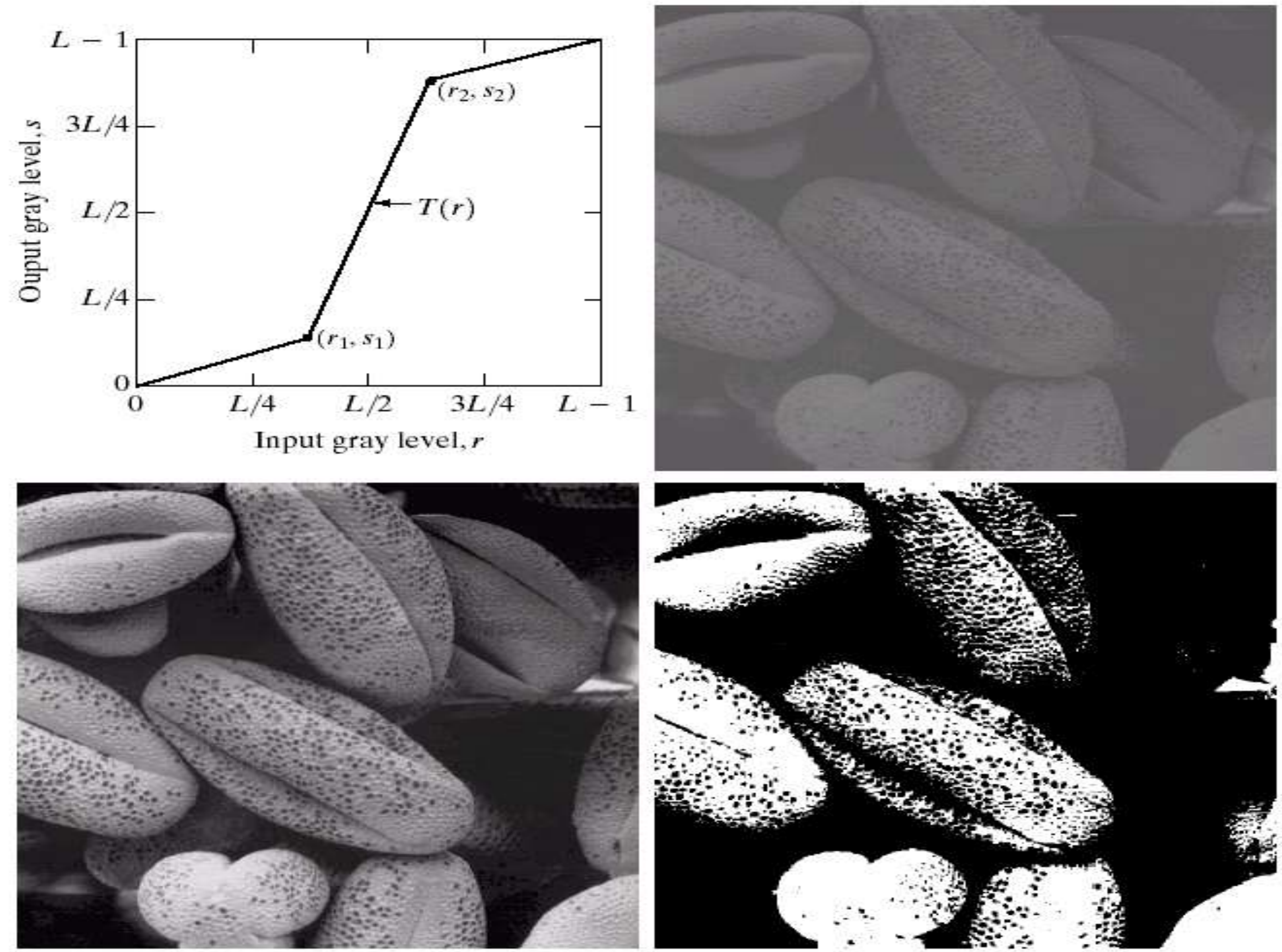


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Contrast stretching

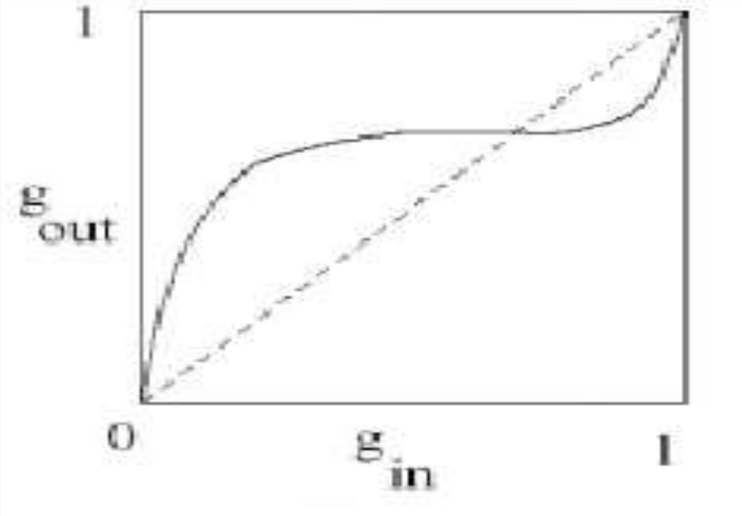
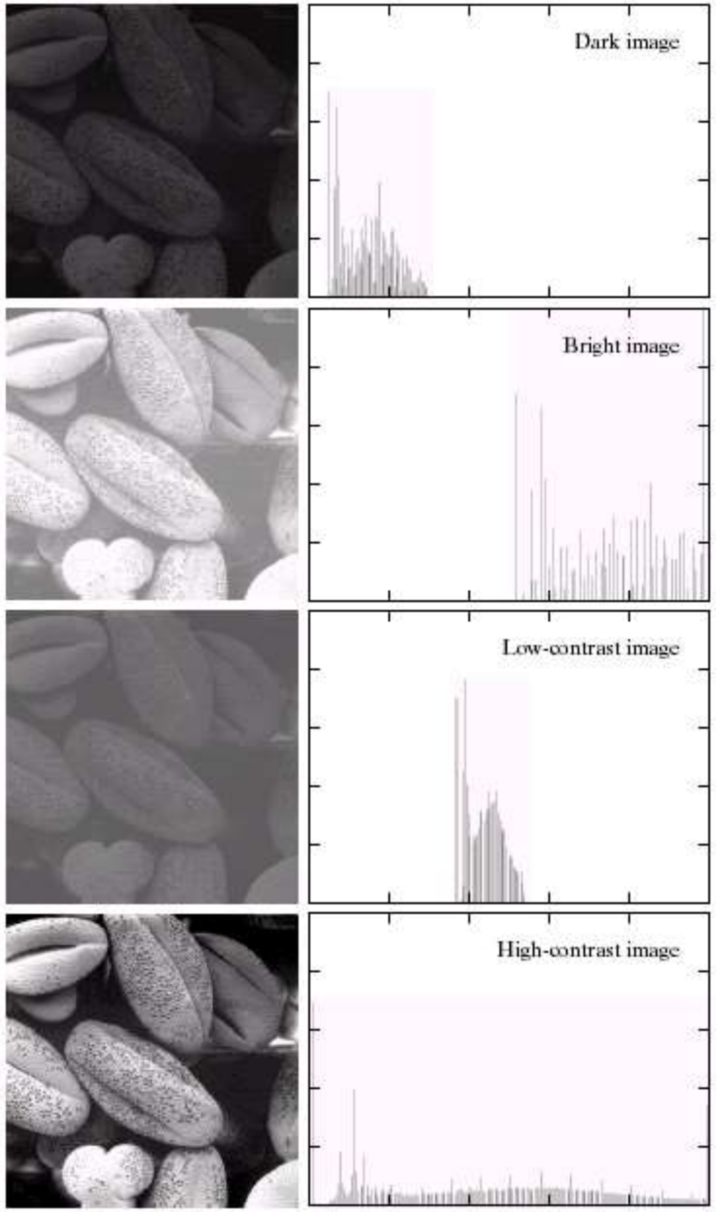


Image Histograms

Artificial Inteligence & Computer Vision Laboratory

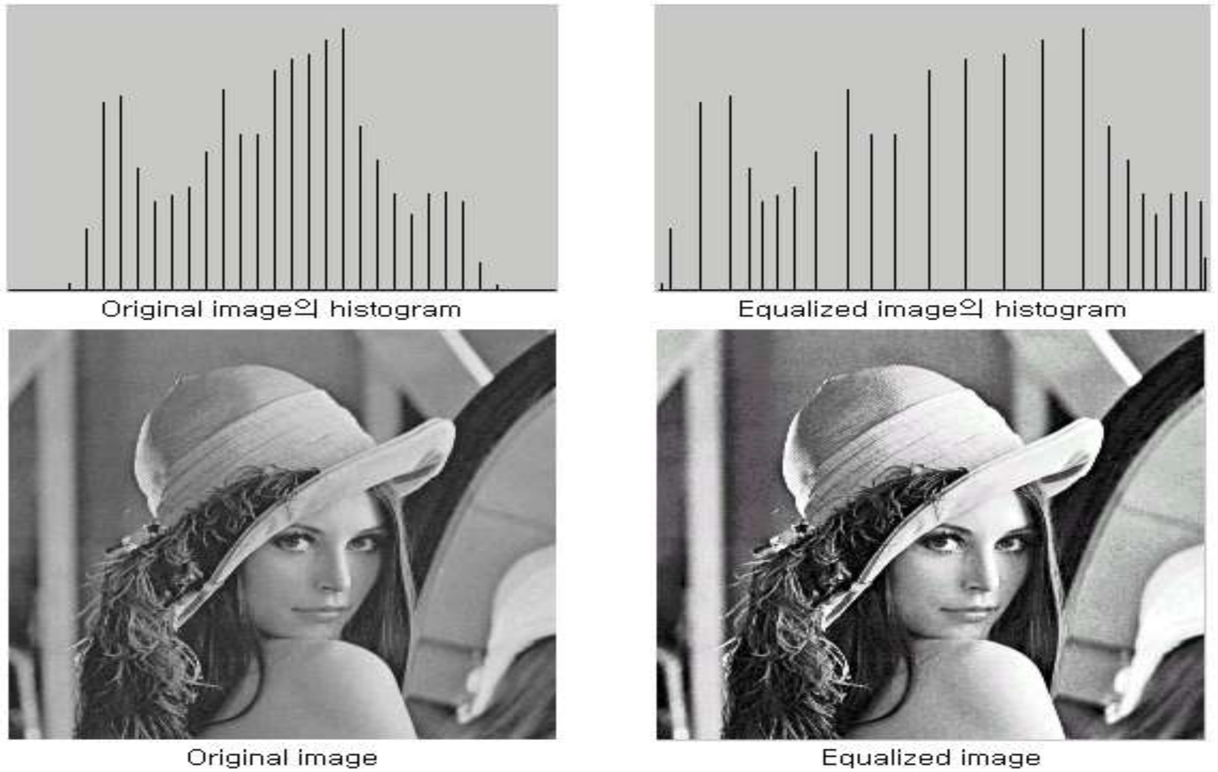


a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

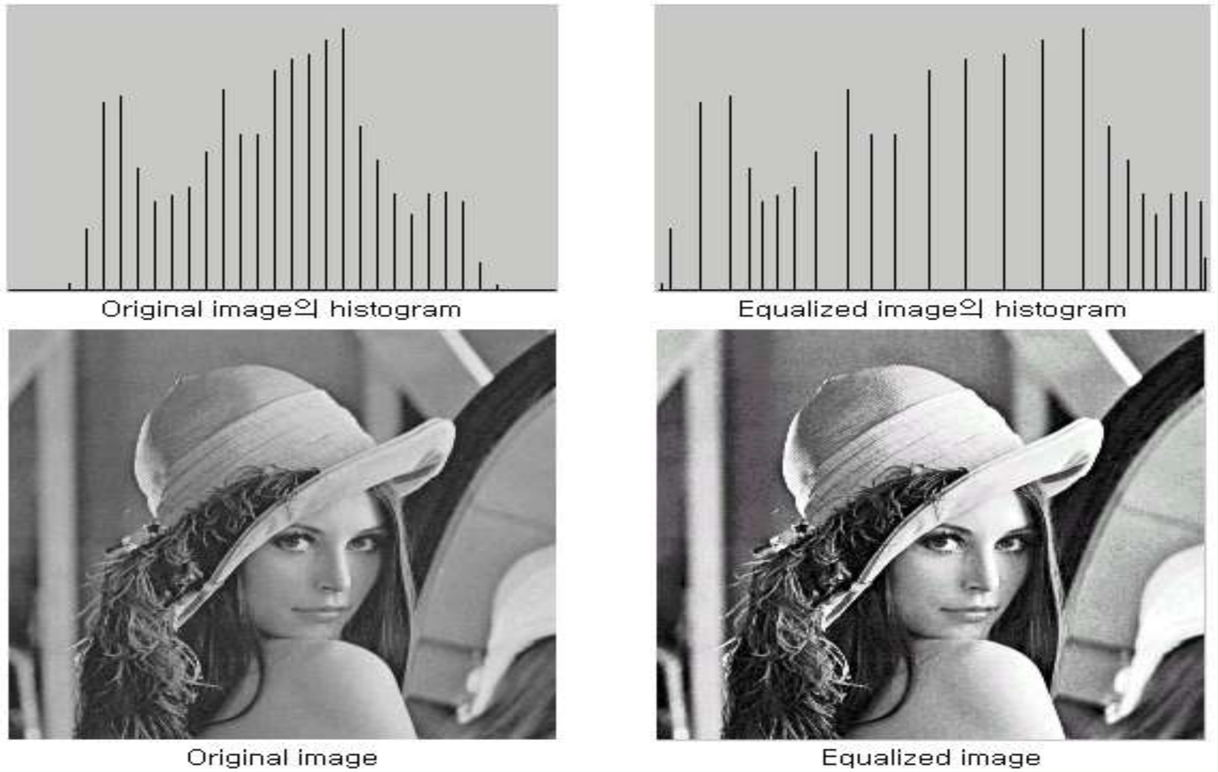
Histogram Equalization

- Two requirements on the operator are
  - (a) The output image should use all available gray levels
  - (b) The output image has approximately the same number of pixels of each gray level



Histogram Equalization

- Two requirements on the operator are
  - (a) The output image should use all available gray levels
  - (b) The output image has approximately the same number of pixels of each gray level



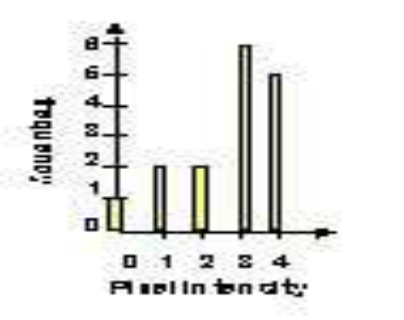
Histogram Equalization

Artificial Inteligence & Computer Vision Laboratory

- Process
  1. Create the histogram about the input image
  2. Make the look-up table by calculating the normalized sum of histogram
  3. Transform the input image using look-up table

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

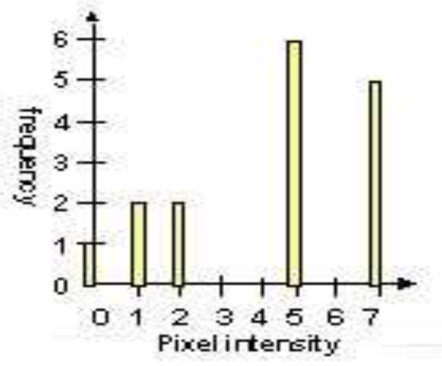
Original Image



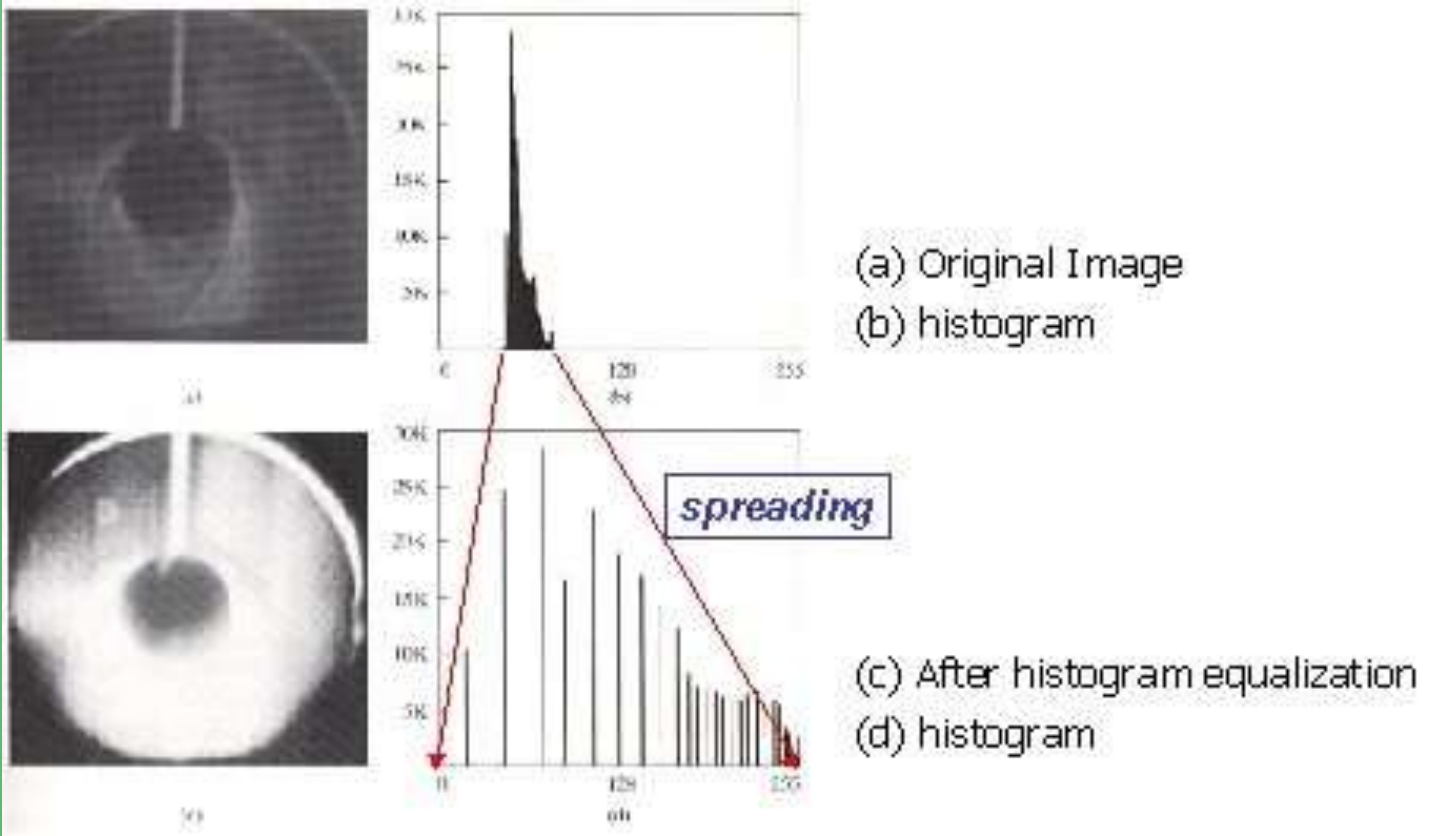
		sum	Normalized sum
0	1	1	0.43 (=1x(7/16))
1	2	3	1.31(= 3x(7/16))
2	2	5	2.18
3	6	11	4.81
4	5	16	7.0
5	0	16	7.0
6	0	16	7.0
7	0	16	7.0

7	7	5	5
7	7	5	5
7	1	2	5
0	1	2	5

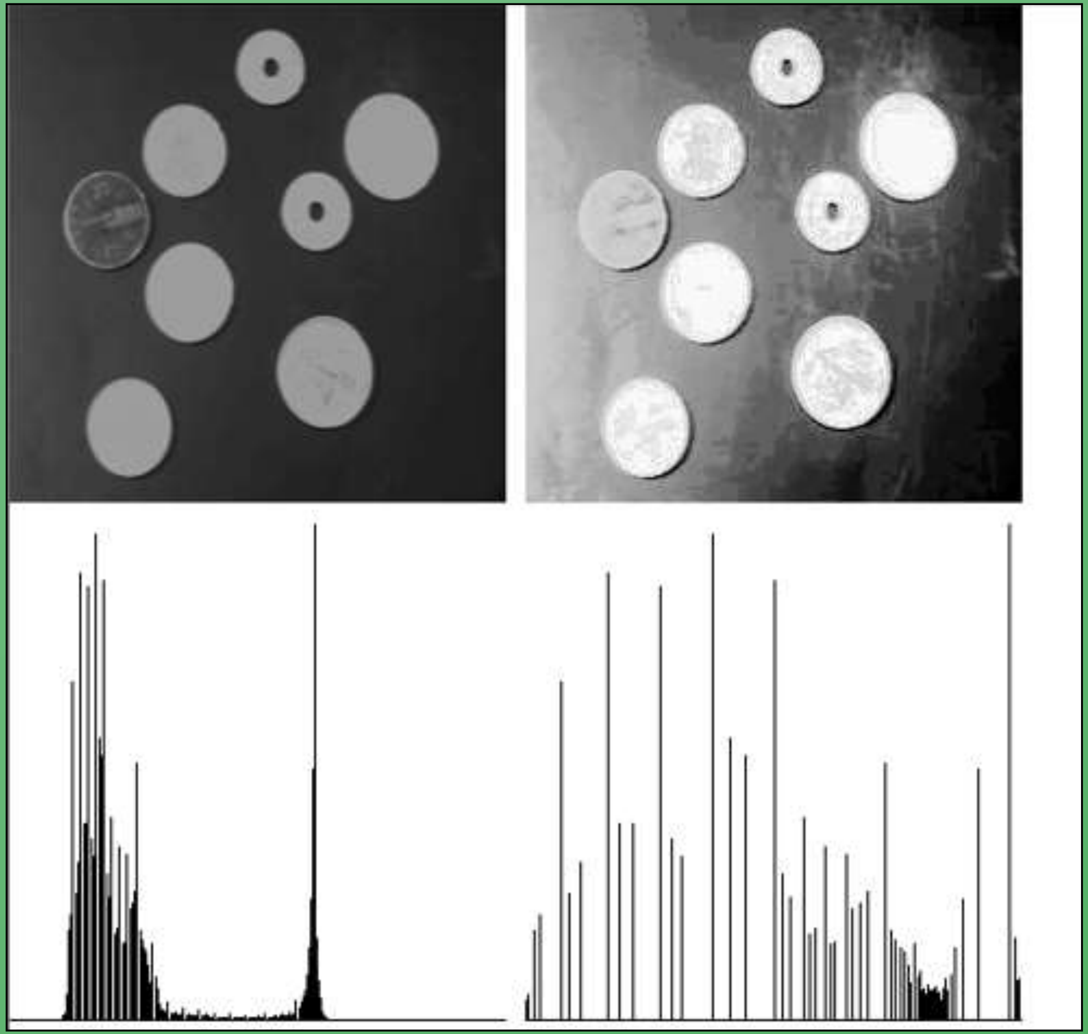
Equalization Image



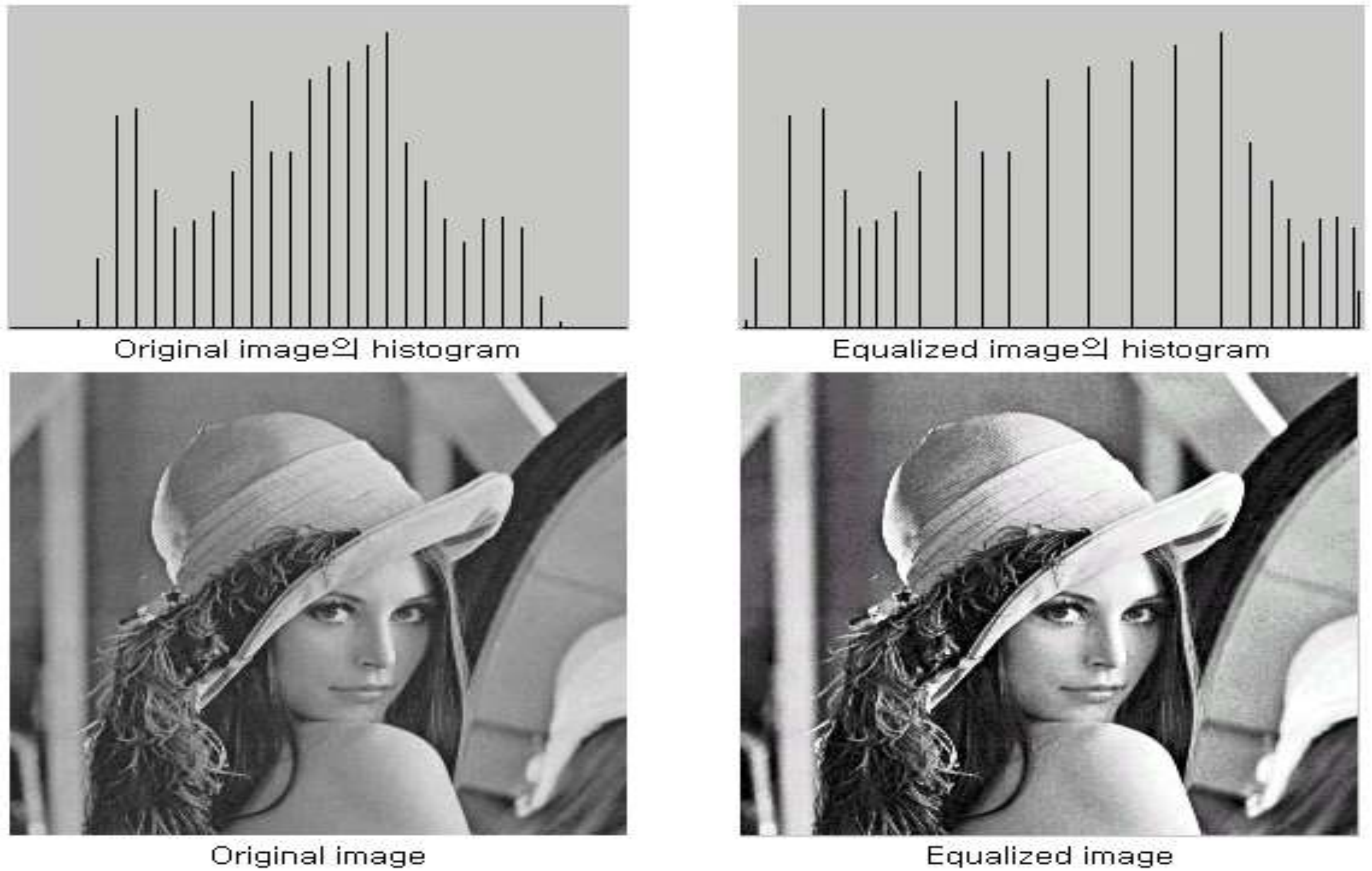
Histogram Equalization



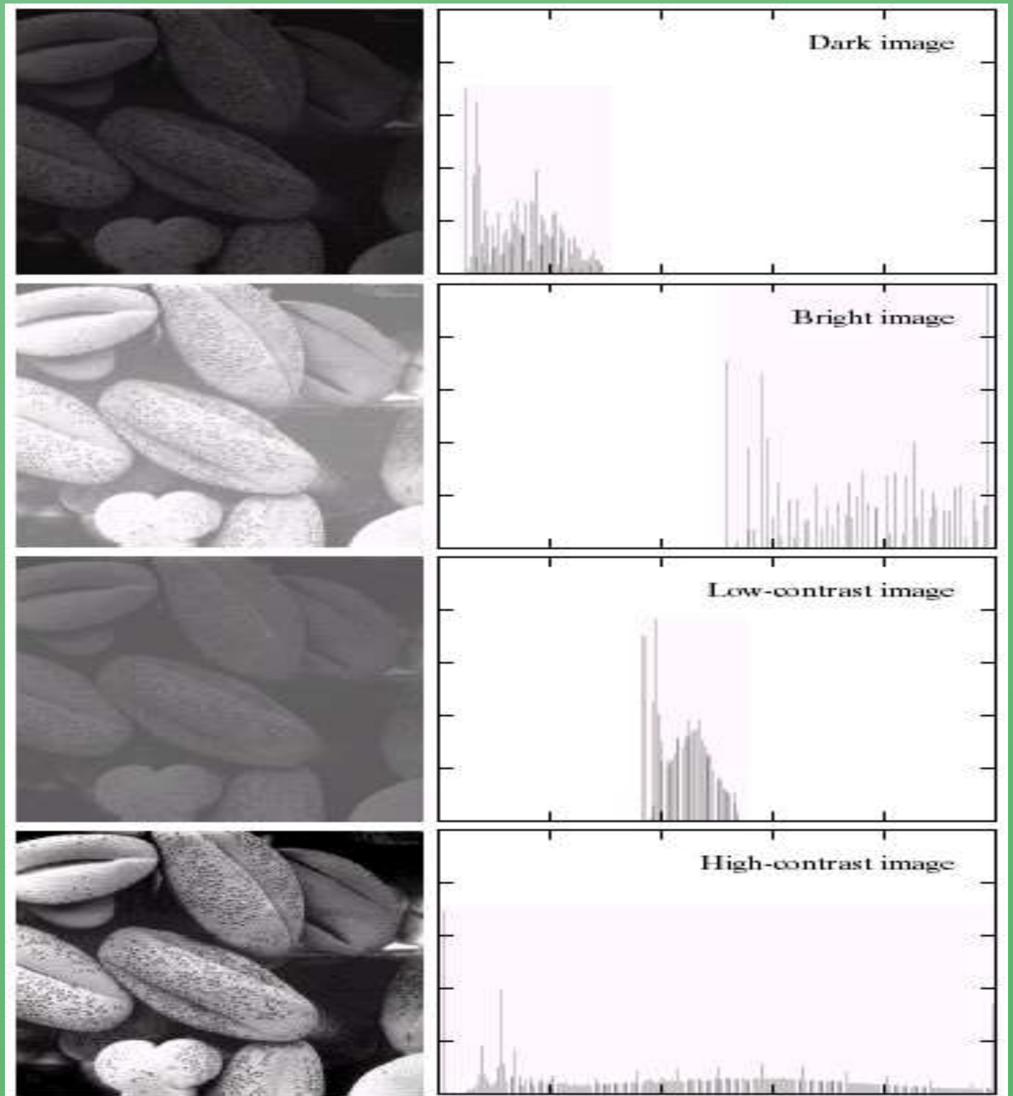
Histogram Equalization

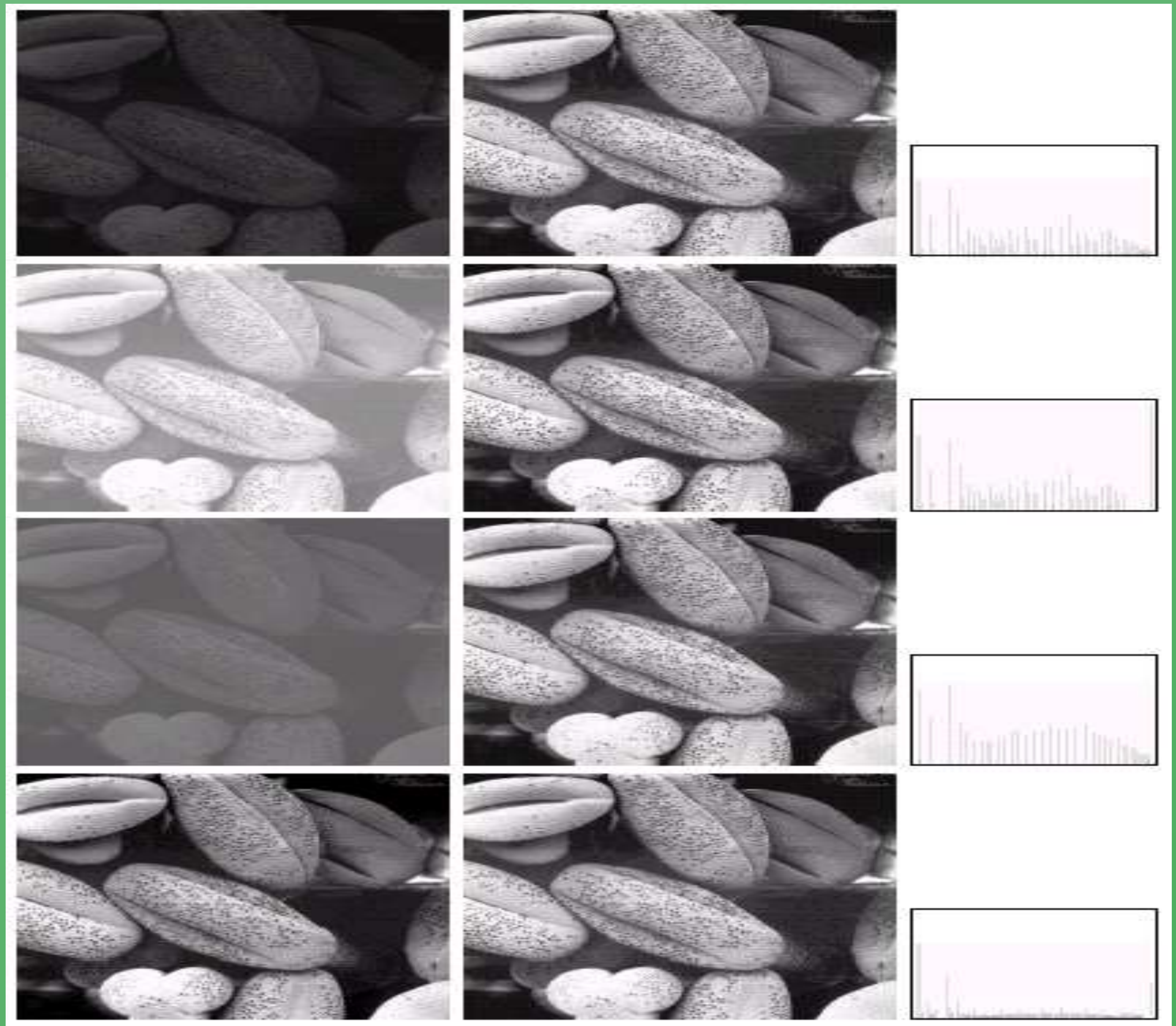


Histogram Equalization

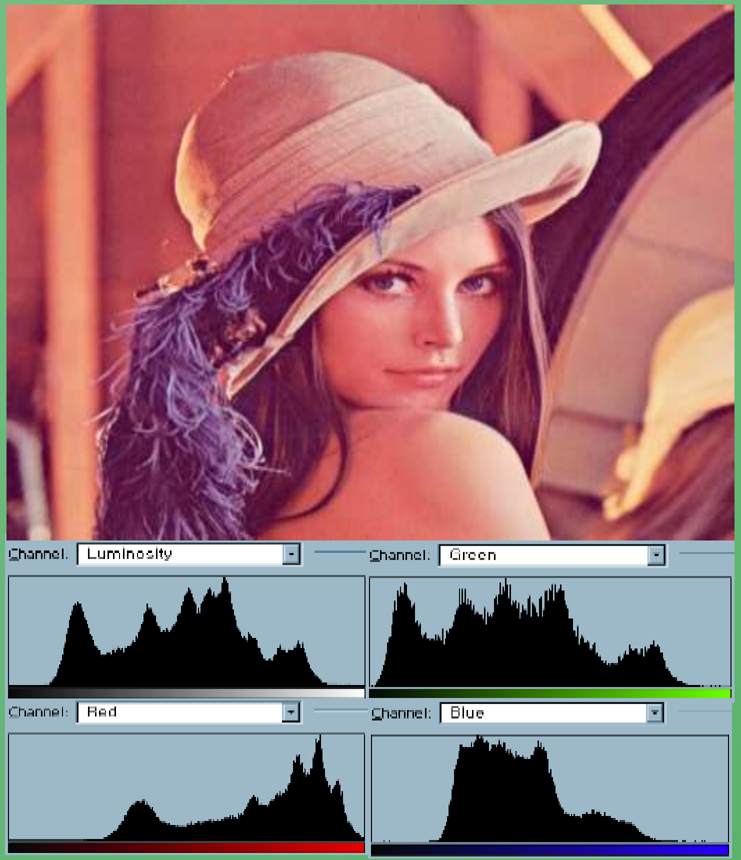


Histogram Equalization

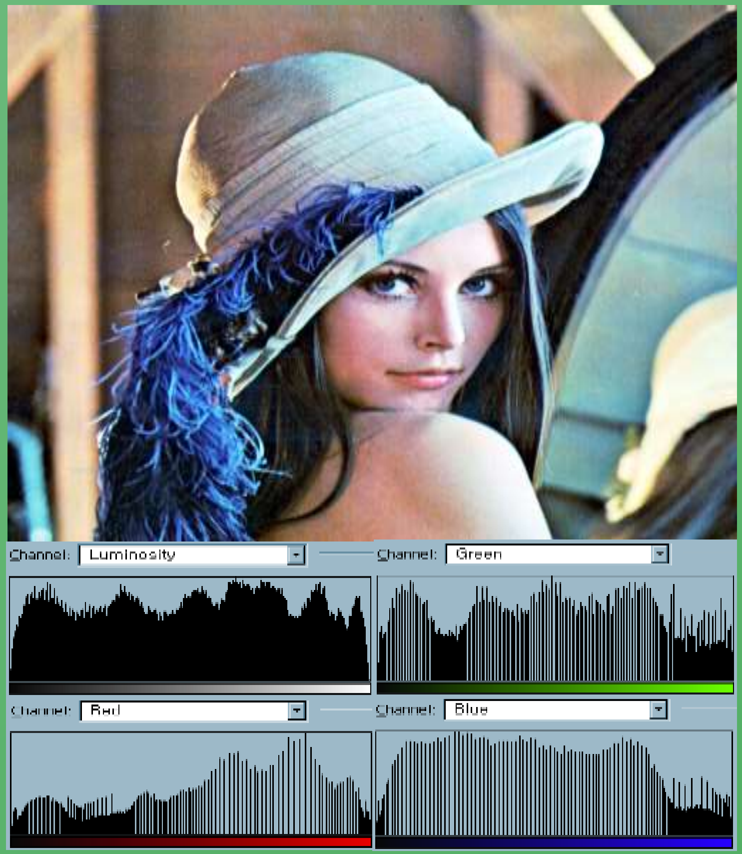




Histogram Equalization on color plane

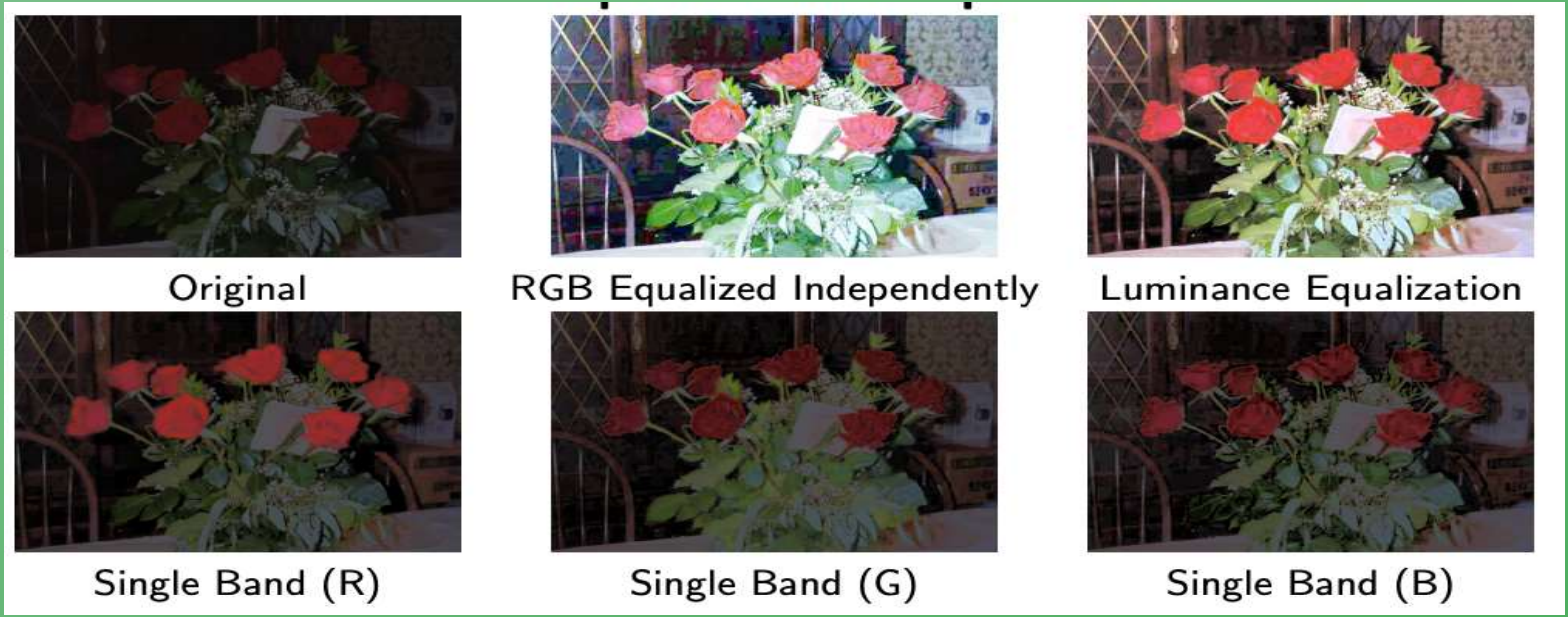


Original Image



Equalized Image

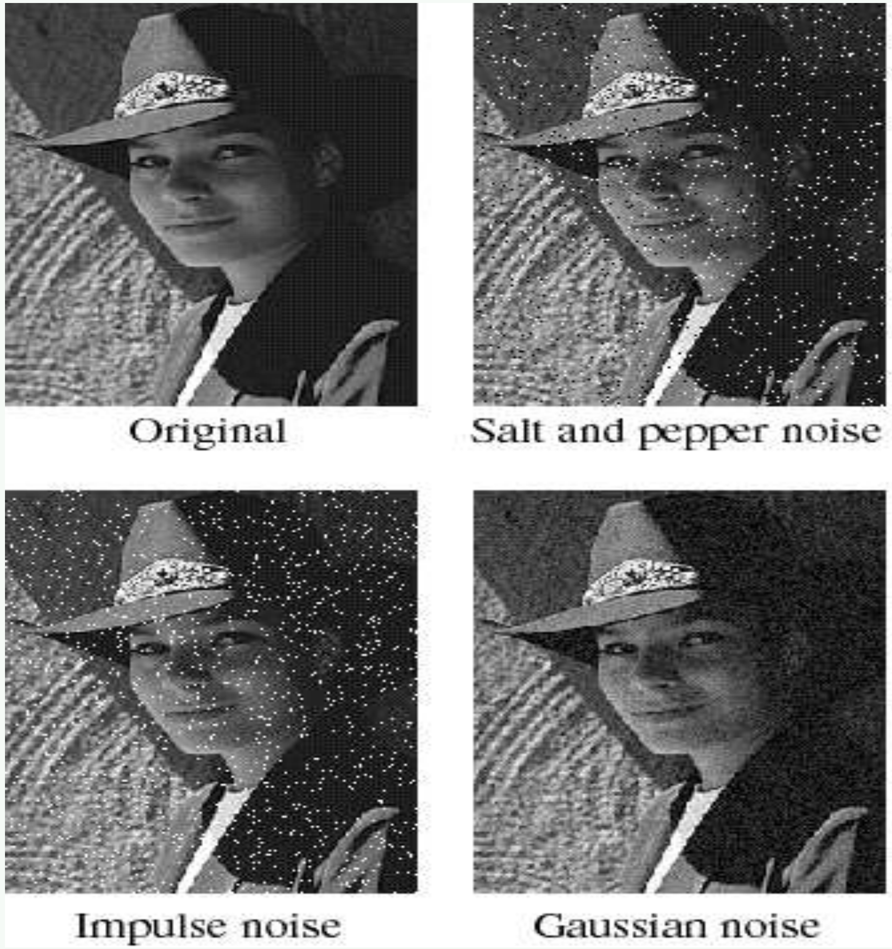
Histogram Equalization on color plane



Noise Filtering

Noise

- Image processing is useful for noise reduction
- Common types of noise
  - **Salt and pepper noise**: contains random occurrences of black and white pixels
  - **Impulse noise**: contains random occurrences of white pixels
  - **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution



Practical Noise Reduction

- How can we “smooth” away noise in a single image?

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$G[x, y]$

Filtering Operations using Masks

- Masks operate on a neighborhood of pixels.
- A mask of coefficients is centered on a pixel.
- The mask coefficients are multiplied by the pixel values in its neighbor hood and the products are summed.
- The result goes into the corresponding pixel position in the output im age.

36	36	36	36	36
36	36	45	45	45
36	45	45	45	54
36	45	54	54	54
45	45	54	54	54

Input Image

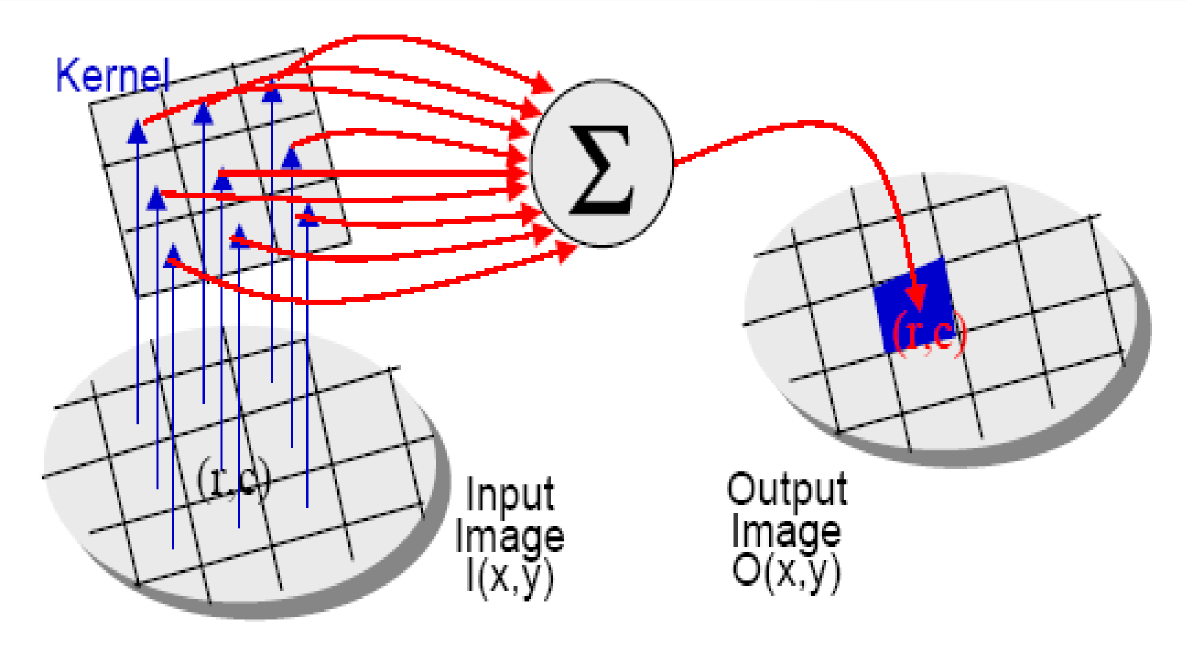
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

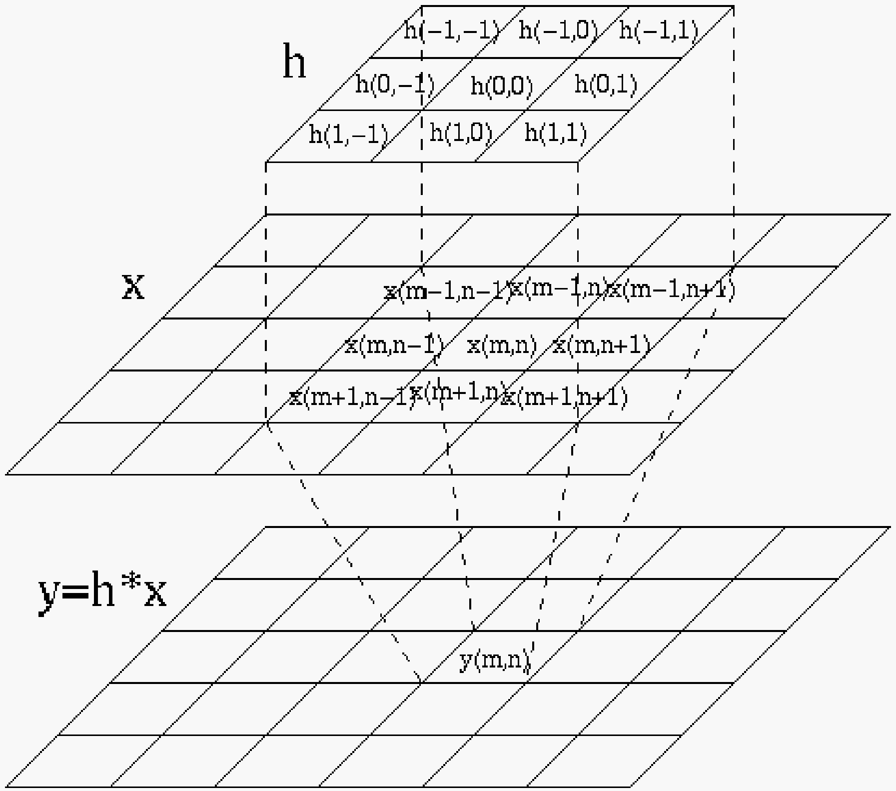
3x3 Mask

**	**	**	**	**
**	39	**	**	**
**	**	**	**	**
**	**	**	**	**
**	**	**	**	**

Output Image

Convolution





Convolution

- A **convolution** operation is a cross-correlation where the filter is flippe d both horizontally and vertically before being applied to the image:

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

- It is written:

$$G = H \star F$$

- Suppose H is a Gaussian or mean kernel. How does convolution diffe r from cross-correlation?

Kernel (Convolution Mask, Window)

- 가중치에 따라 다른 결과를 생성함
  - Convolution mask summed to ‘1’ → same average intensity as the original image.
  - Convolution mask summed to ‘0’→ edge detection.
  - 회선 마스크 또는 회선 커널(2차원 정방행렬)
  - 원시화소가 중앙에 위치하도록 홀수 x 홀수 사용

- 영상스무딩(smoothing), 영상강화(enhancement), 에지검출 (edge detection)등의 효과에 사용

Mean filtering

- How can we "smooth" away noise in a single image?

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$G[x, y]$

Mean Kernel

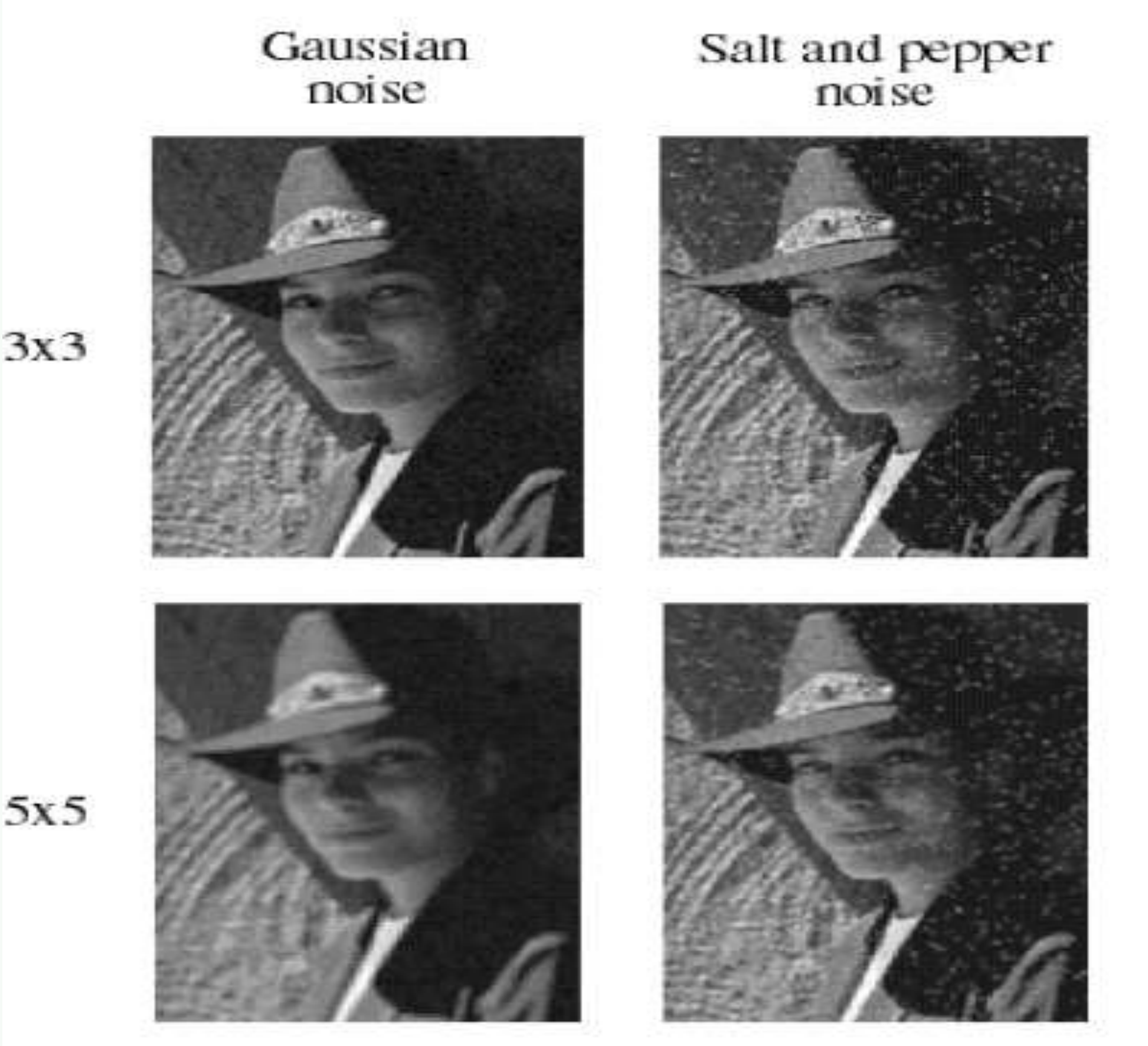
- What's the kernel for a 3x3 mean filter?

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x,y]$

$H[u,v]$

Effects of mean filtering



Gaussian Filtering

- A Gaussian kernel gives less weight to pixels further from the center of the window

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

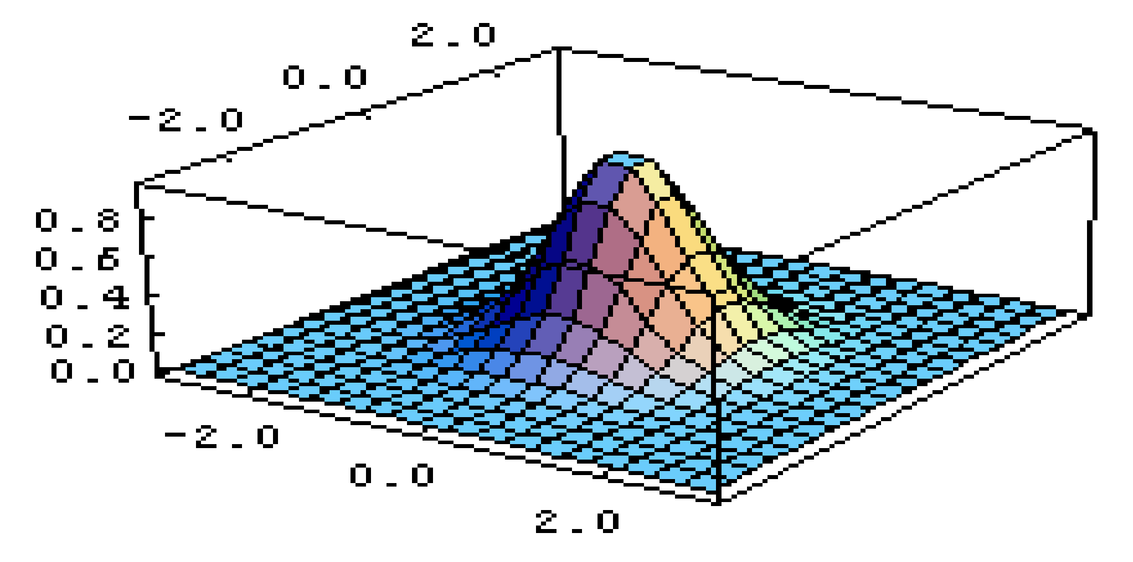
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$H[u, v]$

Gaussian Kernel

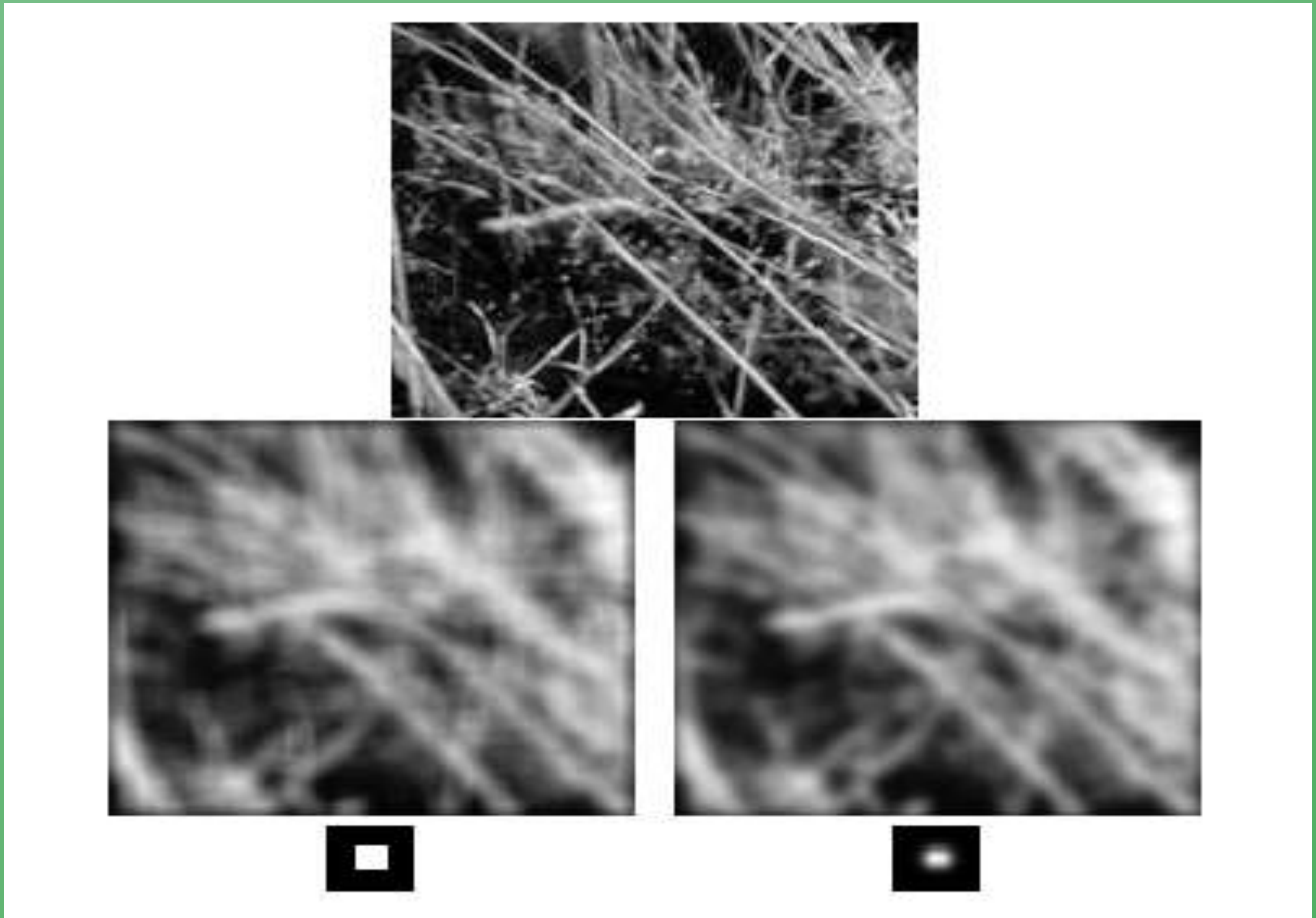
- This kernel is an approximation of a Gaussian function:

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$



$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] F[i-u, j-v]$$

Mean vs. Gaussian filtering



Median Filters

- A **Median Filter** operates over a window by selecting the median inte nsity in the window.

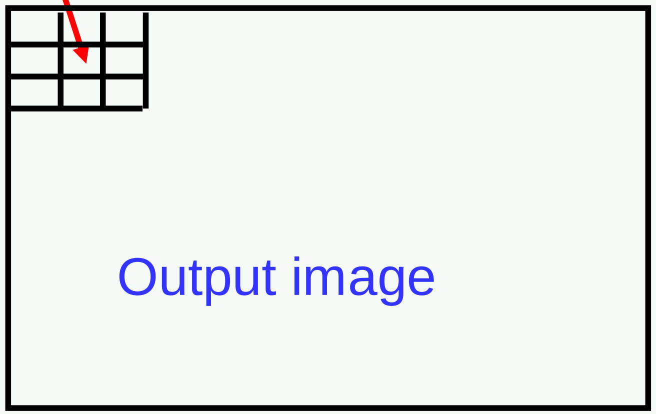
- What advantage does a median filter have over a mean filter?

- Is a median filter a kind of convolution?

Median Filtering

$$\begin{bmatrix} 110 & 110 & 114 \\ 100 & 104 & 104 \\ 95 & 88 & 85 \end{bmatrix} \Rightarrow \{85,88,95,100,104,104,110,110,114\}$$

Median Value



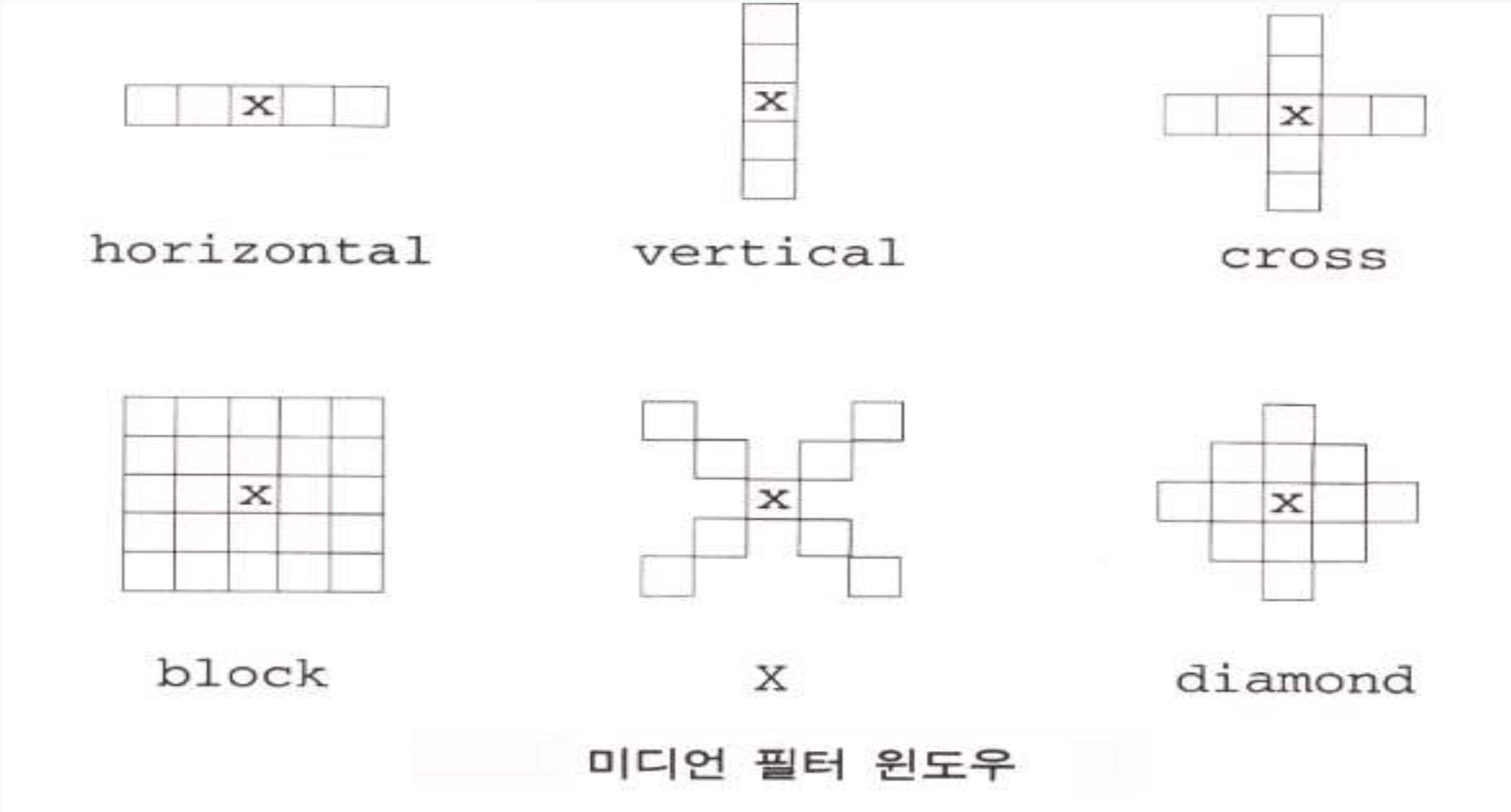
Median Filters

Artificial Inteligence & Computer Vision Laboratory

- A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges.

- Median filtering is a nonlinear operation often used in image processing.

Median Filters



Comparison: salt & pepper noise

Comparison: Gaussian noise

Convolution in Color Space

Artificial Inteligence
& Computer Vision
Laboratory

- RGB영상의 각 채널에 대해 convolution 수행
  - 영상을 블러링하는 경우 유용

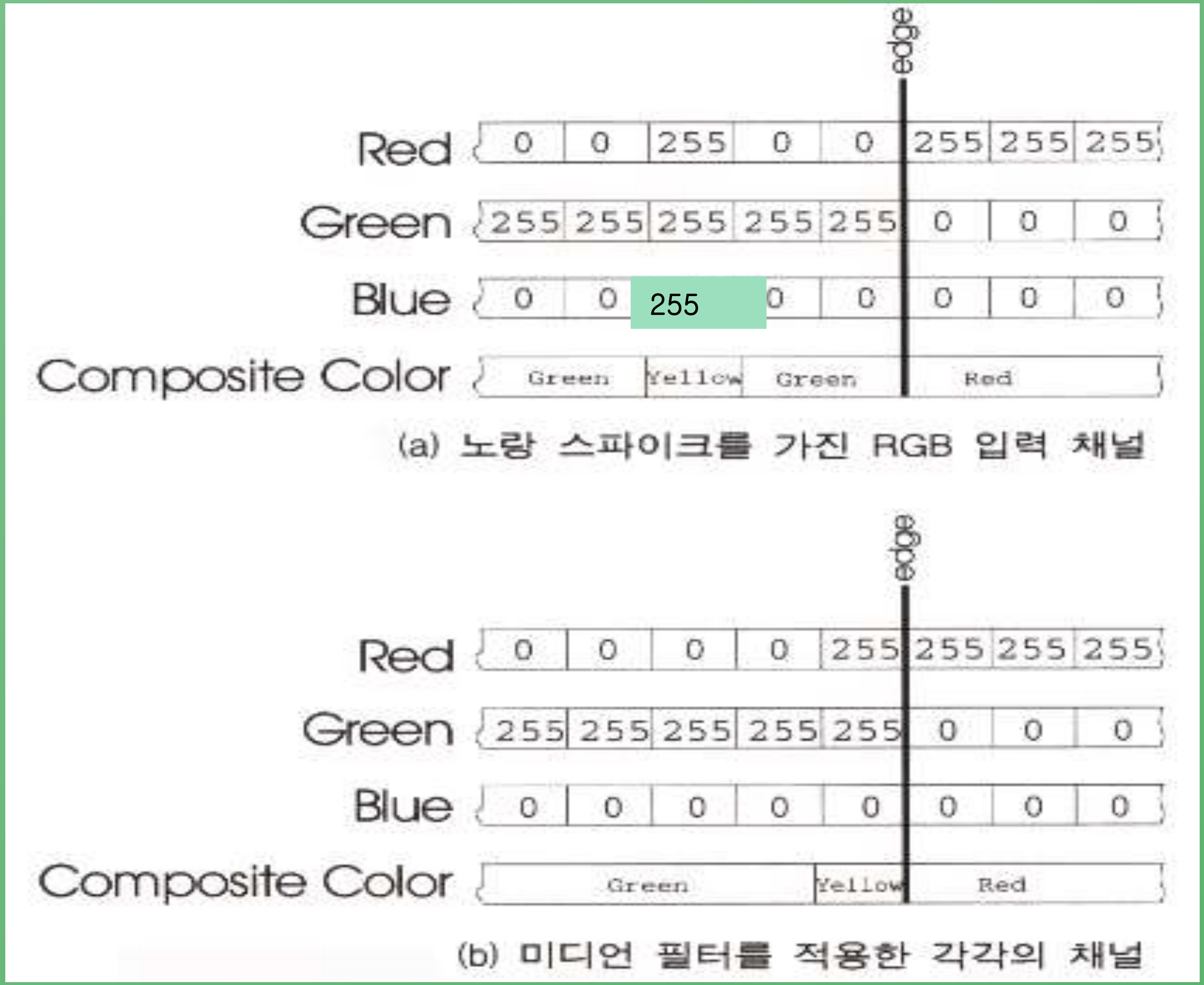
- HSI 컬러 공간에서 영상에 대한 밝기 채널에 대해 회선 수행
  - 밝기 데이터에 대해 convolution하고,RGB 공간으로 변경
  - 원 색상을 보존하고자 하는 경우 유용

Median Filtering in Color Space

- R, G, B 각각의 칼라 요소에 대해 Median Filtering
  - R,G,B 컬러 요소들 각각에 대하여 미디언 필터를 적용한 후, R,G,B 컬러 요소에 대한 미디언 필터의 출력을 결합
  - 문제점
    - 컬러 요소들 사이의 상관관계를 손실

- Boundary Problem
  - Zero-padding
  - Boundary-copying
  - Ignore

Median Filtering in Color Space



$$\sum_{i=1}^{N} | x_{med} - x_i | \leq \sum_{i=1}^{N} | y - x_i |$$

$$Dist_i = \sum_{i=1}^{N} | red_i - red_j | + \sum_{i=1}^{N} | green_i - green_j | + \sum_{i=1}^{N} | blue_i - blue_j |$$

High-Boost Filtering

- In image processing, it is often desirable to emphasize high frequency components representing the image details without eliminating low frequency components (such as sharpening).

- The **high-boost filter** can be used to enhance high frequency component while still keeping the low frequency components.

High-boost = A*Original – Lowpass

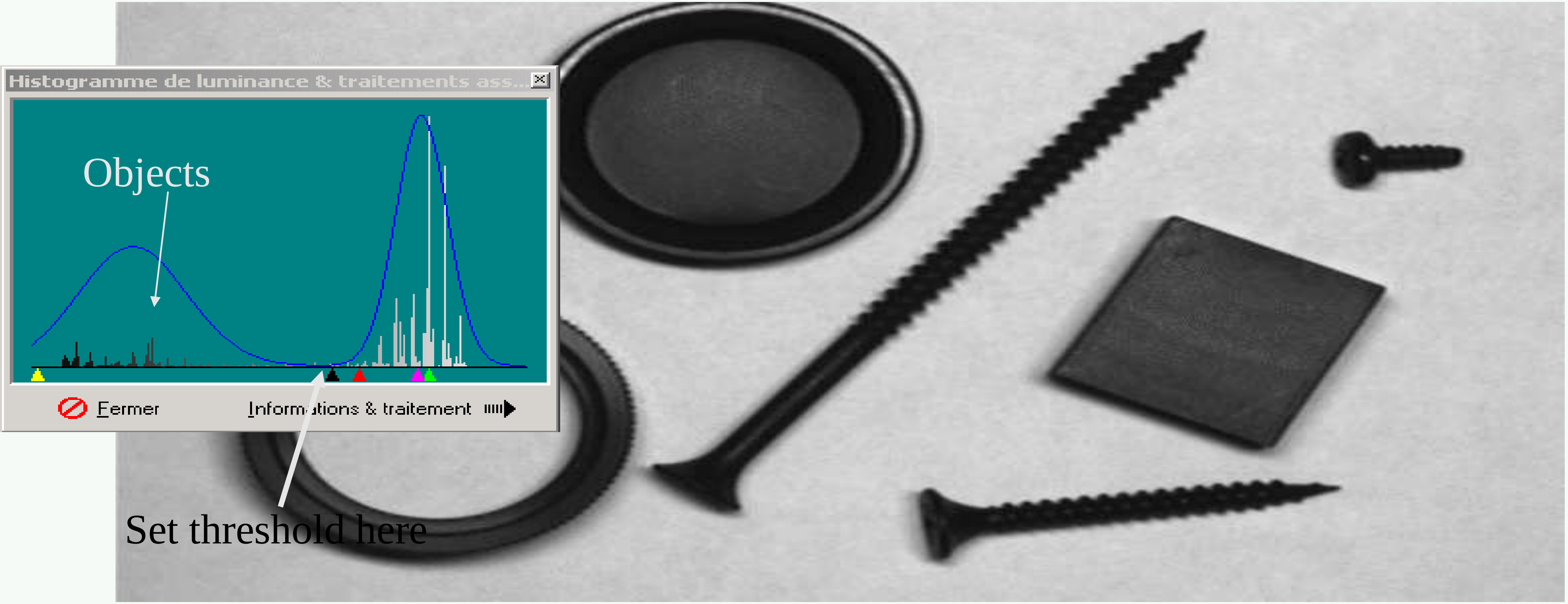
or

High-boost = A*Original + Highpass

- It can be used for animation

Morphological operator

Gray level thresholding



Gray level thresholding

Problem here

How do we fill “missing pixels”?

Mathematical Morphology

mathematical framework used for:

- pre-processing
  - noise filtering, shape simplification, ...
- enhancing object structure
  - Skeletonization
- segmentation
  - watershed,...
- quantitative description
  - area, perimeter, ...

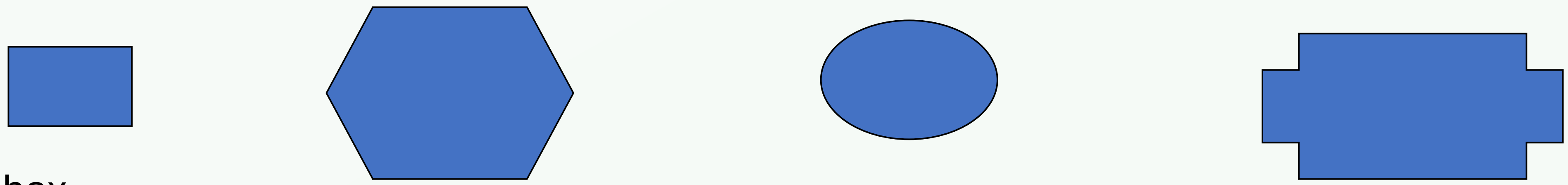
Mathematical Morphology

- Morphological processes logically combine pixel brightness with **a structuring element**, looking for specific patterns.
  - Morphological processes use set theory operations, such as intersection (AND), union (OR), and complement (NOT), to combine the pixels logically into a resulting pixel value.
- Like spatial convolution, the morphological process moves across the input image, pixel by pixel, placing resulting pixels in the output image.
  - At each input pixel location, the pixel and its neighbors are logically compared against a structuring element, or morphological mask, to determine the output pixel's logical value.
  - The structuring element is generally composed of square dimension of size 3x3, 5x5, and greater, depending upon the application.

Structuring Elements

A structuring element is a shape mask used in the basic morphological operations.

They can be any shape and size that is digitally representable, and each has an origin.



box

hexagon

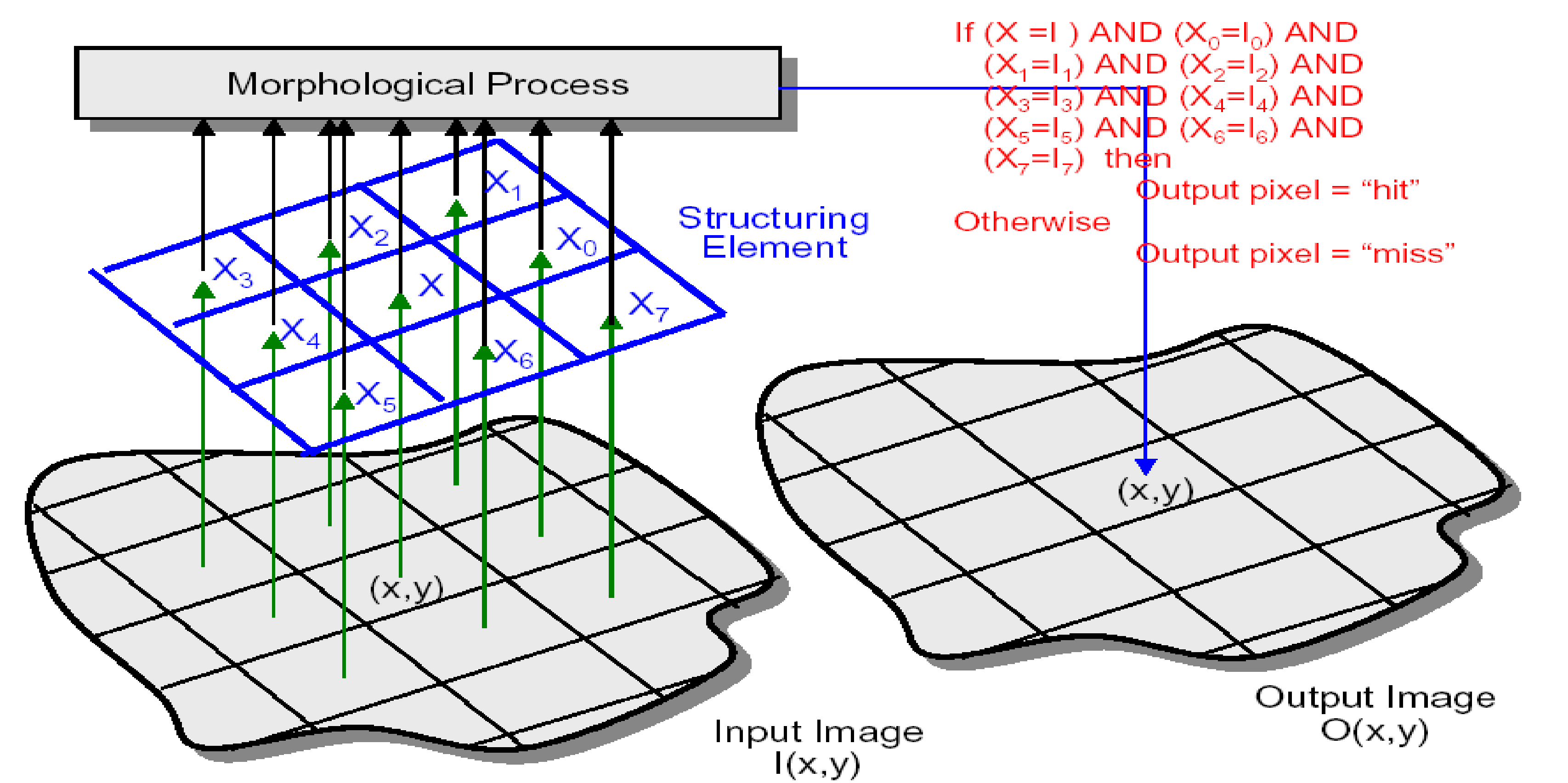
disk

something

box(length,width)

disk(diameter)

Mathematical Morphology with Structuring Elements



Binary Mathematical Morphology

Artificial Inteligence
& Computer Vision
Laboratory

Binary mathematical morphology consists of two basic operations

dilation **and** **erosion**

and several composite relations

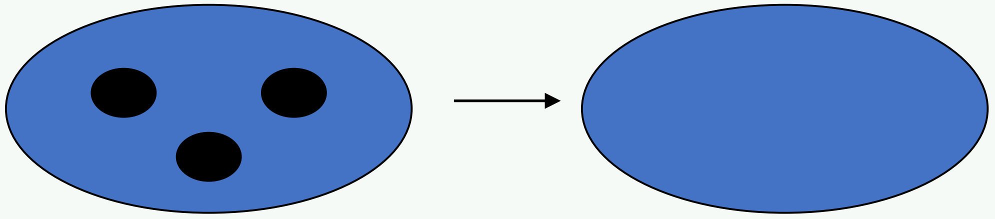
closing **and** **opening**
conditional dilation
. . .

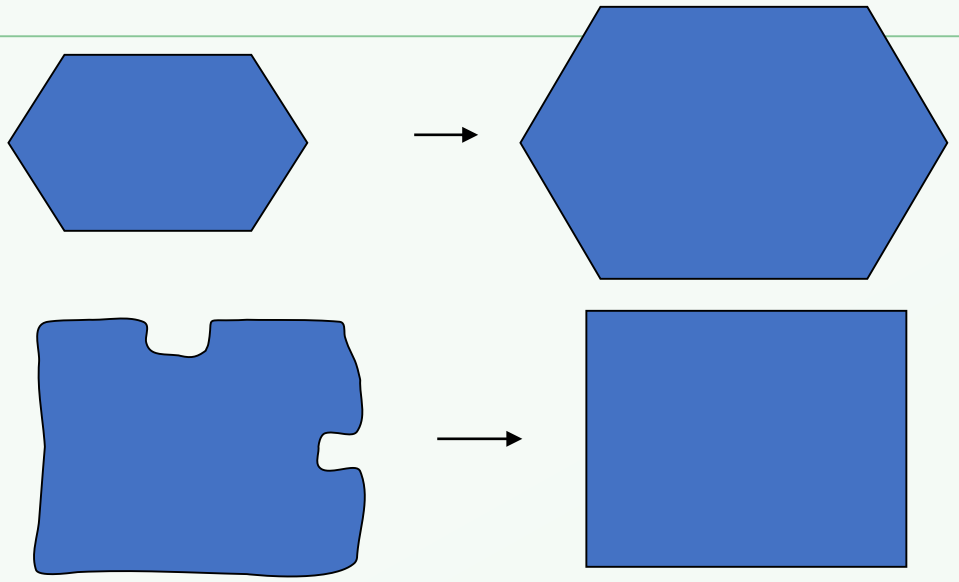
Dilation

Dilation expands the connected sets of 1s of a binary image.

It can be used for

1. growing features
2. filling holes and gaps



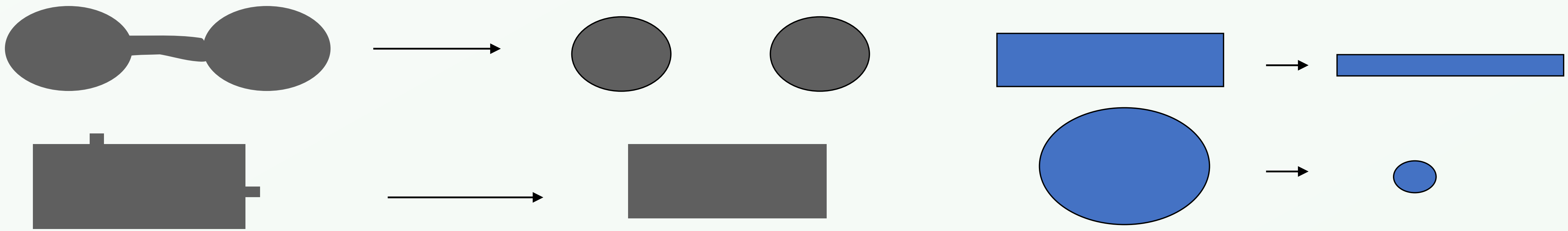


Erosion

Erosion shrinks the connected sets of 1s of a binary image.

It can be used for

1. shrinking features
2. Removing bridges, branches and small protrusions



Dilation with Structuring Elements

Artificial Inteligence & Computer Vision Laboratory

The arguments to dilation and erosion are

1. **a binary image B**
2. **a structuring element S**

dilate(B,S) takes binary image B, places the origin of structuring element S over each 1-pixel, and Ors the structuring element S into the output image at the corresponding position.

$$\underset{B}{\boxed{\begin{matrix}0&0&0&0\\0&1&1&0\\0&0&0&0\end{matrix}}} \qquad \underset{S}{\boxed{\begin{matrix}1&\\ \color{red}{1}&1\end{matrix}}} \xrightarrow{\text{dilate}} \underset{B \oplus S}{\boxed{\begin{matrix}0&1&1&0\\0&1&1&1\\0&0&0&0\end{matrix}}}$$

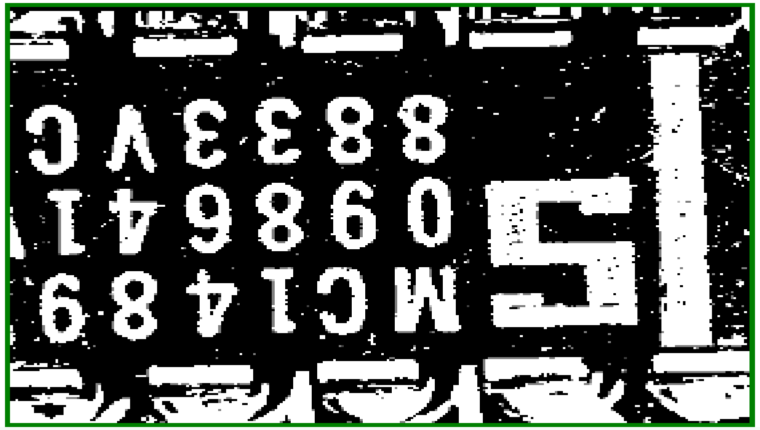
origin

Dilation with Structuring Elements

The arguments to dilation and erosion are

1. **a binary image B**
2. **a structuring element S**

dilate(B,S) takes binary image B, places the origin of structuring element S over each 1-pixel, and Ors the structuring element S into the output image at the corresponding position.



Erosion with Structuring Elements

The arguments to dilation and erosion are

1. **a binary image B**
2. **a structuring element S**

erode(B,S) takes a binary image B, places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image only if every position of S (with a 1) covers a 1 in B.

origin

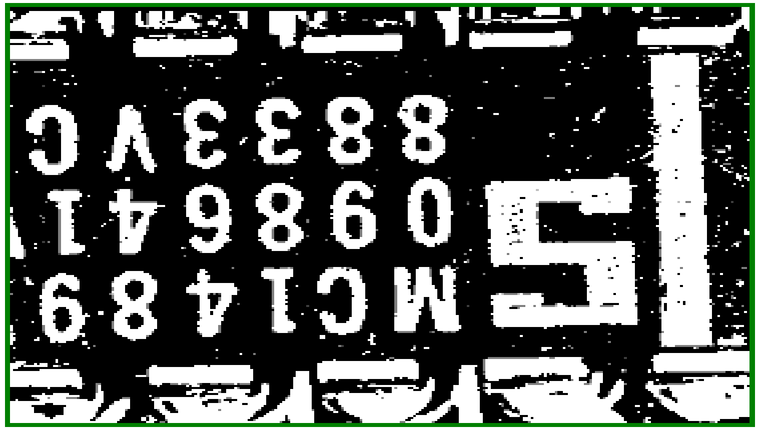
$$\underset{B}{\begin{bmatrix} 0&0&1&1&0\\ 0&0&1&1&0\\ 0&0&1&1&0\\ 1&1&1&1&1 \end{bmatrix}} \qquad \underset{S}{\begin{bmatrix} 1\\ \color{red}{1}\\ 1 \end{bmatrix}} \xrightarrow{\text{erode}} \underset{B \ominus S}{\begin{bmatrix} 0&0&0&0&0\\ 0&0&1&1&0\\ 0&0&1&1&0\\ 0&0&0&0&0 \end{bmatrix}}$$

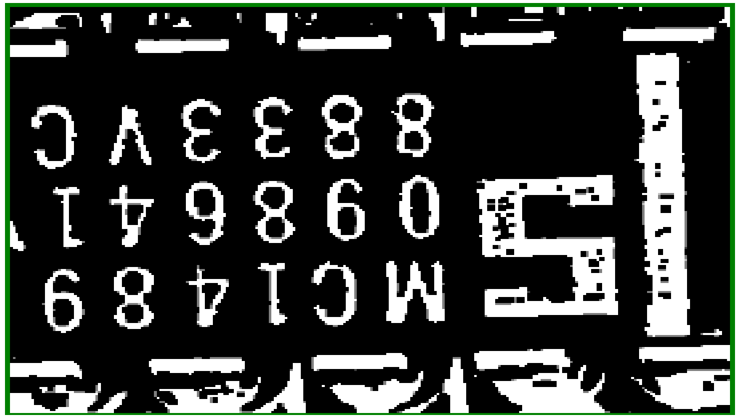
Erosion with Structuring Elements

The arguments to dilation and erosion are

1. **a binary image B**
2. **a structuring element S**

erode(B,S) takes a binary image B, places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image only if every position of S (with a 1) covers a 1 in B.





Example to Try

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

$$S \quad \begin{bmatrix} 1&1&1\\ 1&1&1\\ 1&1&1 \end{bmatrix}$$

erode

dilate with same structuring element

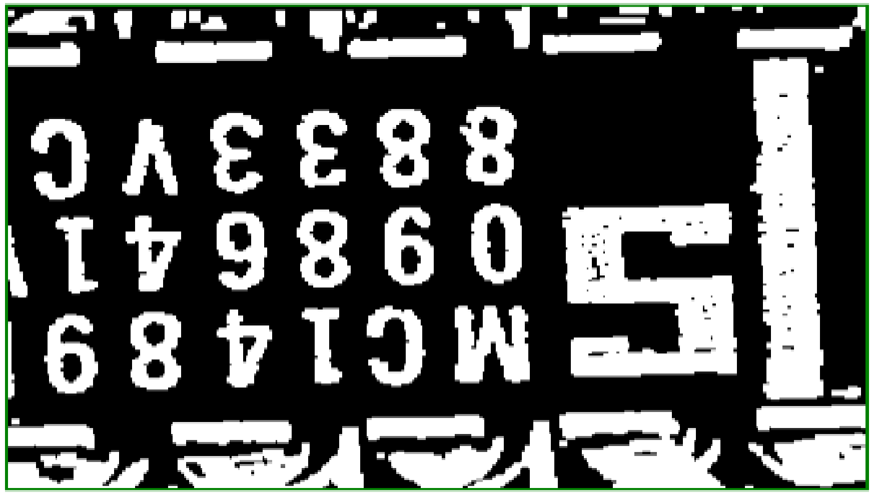
Erosion with Structuring Elements

Artificial Inteligence & Computer Vision Laboratory

- Closing is the compound operation of dilation followed by erosion (with the same structuring element)

- Opening is the compound operation of erosion followed by dilation (with the same structuring element)

Dilated & Eroded Image



Eroded & Dilated Image