Chapter 5

Bitmapped Image

2017.03

Prof. Park Kyusik

Contents

- Basic about Bitmap image
- JPEG image compression
  - Entropy coding
  - Hybrid JPEG encoding
- Some image processing
  - Pixel Point Processing
  - Pixel Group Processing
  - Geometrical Transformation

Bitmapped Images

- Naturally created from an external source in an analog form
  - Scanner, Digital camera, Satellite Image, X-ray
  - Basic Elements - Pixel, Resolution, Color
  - Smallest image element : Pixel or Pel
  - Pixel based image is stored as Bitmap format

Image Compression

- Image files may be too big for network transmission
  - Image compression - **Discard some redundant information** to reduce data file size
  - Effectiveness of compression will depend on the structure of actual image data
  - Lossy or Lossless compression (JPEG?)

Lossless Compression

- No loss on data compression
  - Sometimes called Entropy Coding
  - Run-length encoding (RLE)
  - Huffman coding
  - Dictionary-based schemes – LZ77, LZ78, LZW (LZW used in GIF)

Run Length Encoding (RLE)

- Consider a character runs of **15 'A'**
  - Require 15 bytes to store
  - RLE counts number of repetition of the same characters

- Example)

AAAAAAAAAAAAAAACCCB // 19 bytes
→
15A3C1B // 7 bytes

- Fax transmission
  - Counts white(1) and black(0) pixel runs

- Compression is normally measured with the compression ratio :
  - **Compression Ratio =**

    (original size / compressed size) : 1

- Example with 16 characters string
  - **000ppppppXXXXaaa → 3(0),6(p),4(X),3(a)**

    16 byte string → 8 bytes
  - In this case, RLE yields a compression ratio of 2:1

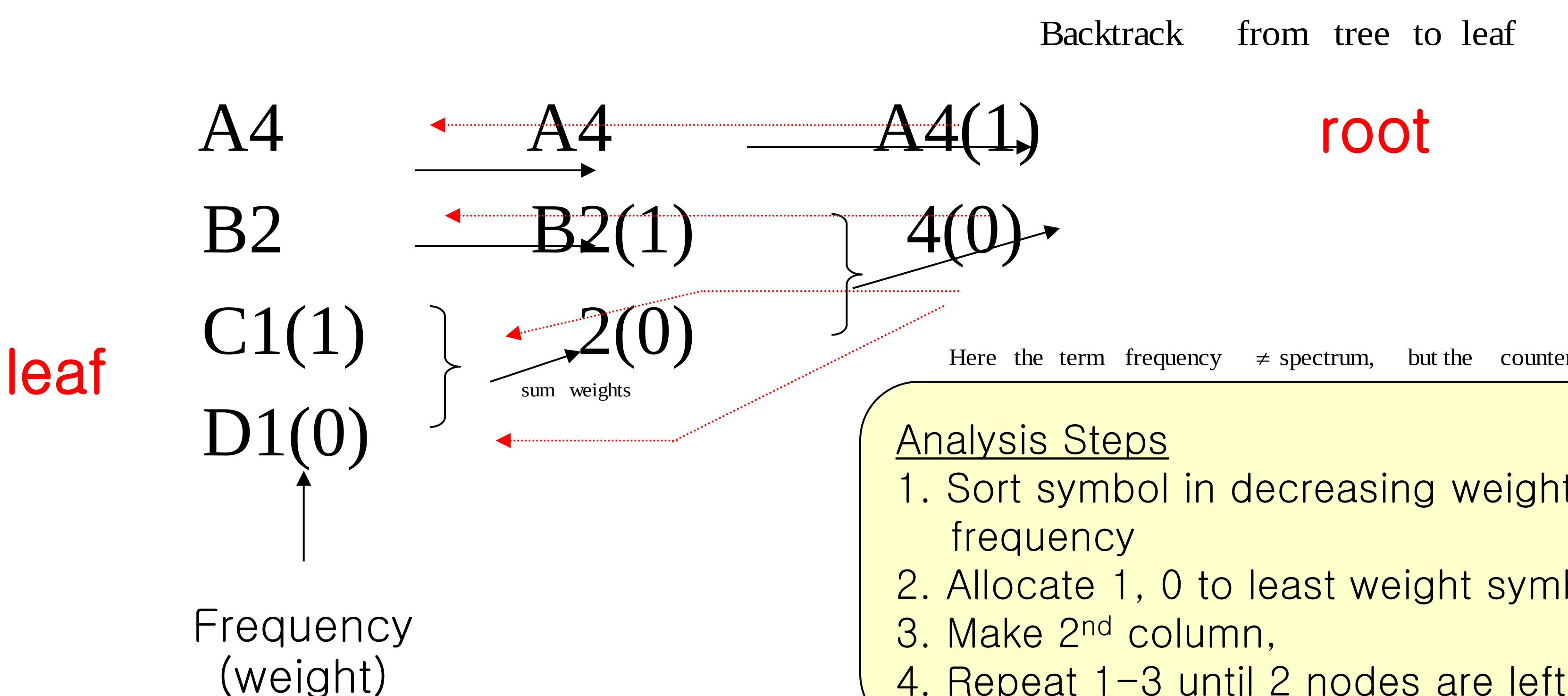
Huffman Coding

- Basic principle
  - Allocate less bits to the more frequently occurring symbols

Step 1) Analyze message symbols to have occurring frequency or probability for each symbol

Step 2) Generate binary Huffman code tree by taking account of symbol frequency

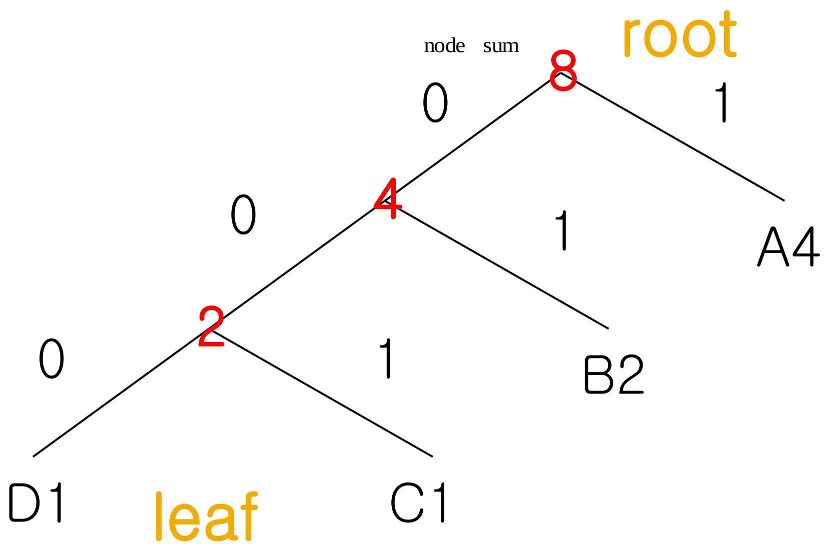
- Encoding Ex)
  - String of symbols “AAAABBCD”
  - Tree derivation



Analysis Steps

1. Sort symbol in decreasing weight frequency
2. Allocate 1, 0 to least weight symbol
3. Make 2nd column,
4. Repeat 1-3 until 2 nodes are left

Generate Binary Huffman Tree



Follow tree to get encoding results

Encoding result

A= 1
B= 01
C= 001
D= 000

Binary 1 means right path, 0 means left path

- (Checking!) weight : D1 C1 2 B2 4 A4 8
  - If weight list lies in increasing order, correct coding!
  - Note – allocate less bits to more occurring symbols

Lossy Compression

- There is loss on data compression
- JPEG, JPEG 2000
  - Analyze and compress image on the frequency domain
- Main Idea
  - Peoples are not very sensitive to high frequencies area of the image!
  - Some high frequency information can be discarded without perceptible loss of quality

Key Compression - DCT

❖ **Compression using spatial correlation in image**

spatial = 2D space

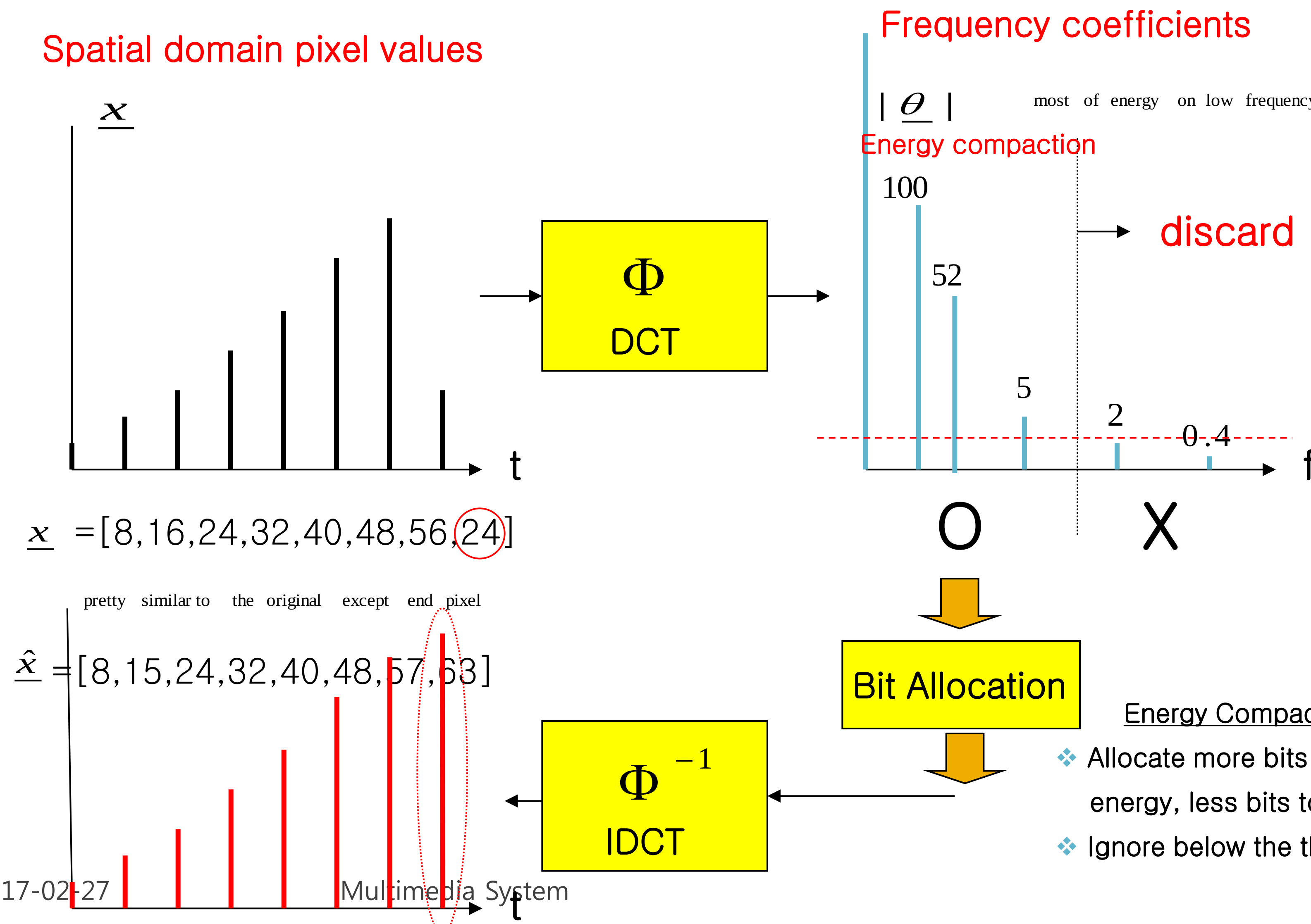
Image pixel is located at 2D position (x_0, y_0)

- Low frequency area of image (Background)
  - Neighboring pixels are changing smoothly
    → High spatial correlation between pixels

- High frequency area of image (Edge)
  - Neighboring pixels are changing rapidly
    → Low spatial correlation between pixels

DCT (Discrete Cosine Transform)

- Transform spatial-domain image pixels to the frequency domain by using DCT
  - After transform, most of energy is concentrated on the low frequency area (Energy Compaction)
  - Less energy on high frequency
- Bit allocation principle
  - Allocate bits depending on energy
  - Allocate more bits to low frequency area, and allocate less bits to high frequency area
  - Ignore the component below some threshold



Energy Compaction

- Allocate more bits to large energy, less bits to small
- Ignore below the threshold

energy compaction

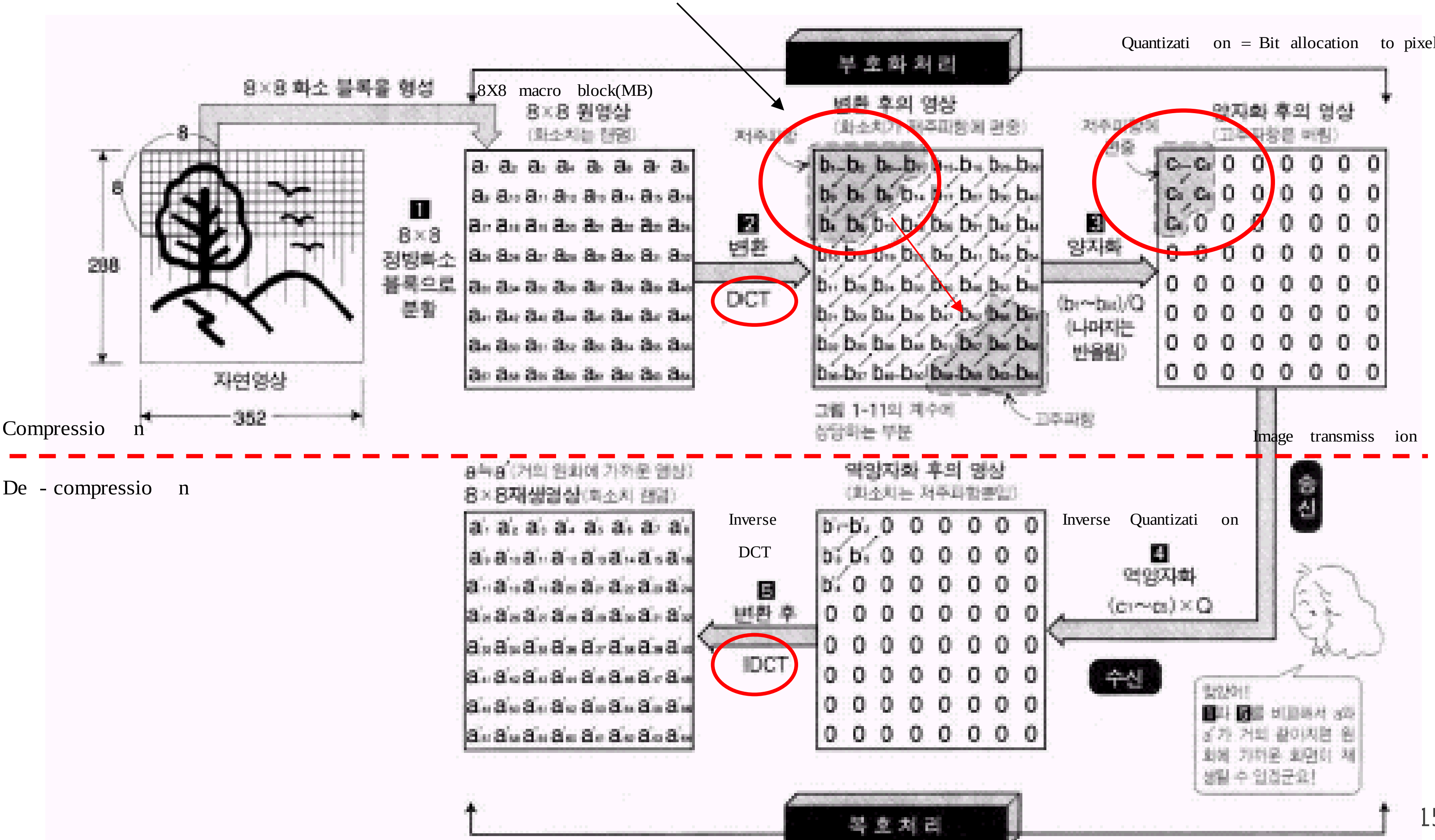


Image (Video) 압축 표준(IS)

	IS	Serial	Name	Year
Fax	ITU-T	T.4	G3 Fax(PSTN)	1980
		T.6	G4 Fax(ISDN)	1984
Still Image	ISO/IEC	10918-1	JPEG, JPEG2000	1992
		11544-1	JBIG (Binary Image)	1993
Video	ITU-T	H.261	pX64Kbps, ISDN Video conferencing	1990
		H.263	PSTN, ISDN Video Conf.	1995
	ISO/IEC	11172-2	MPEG1(VHS)	1992
		13818-2 H.264	MPEG2 (HDTV) MPEG4 (Mobile)	1994 2002

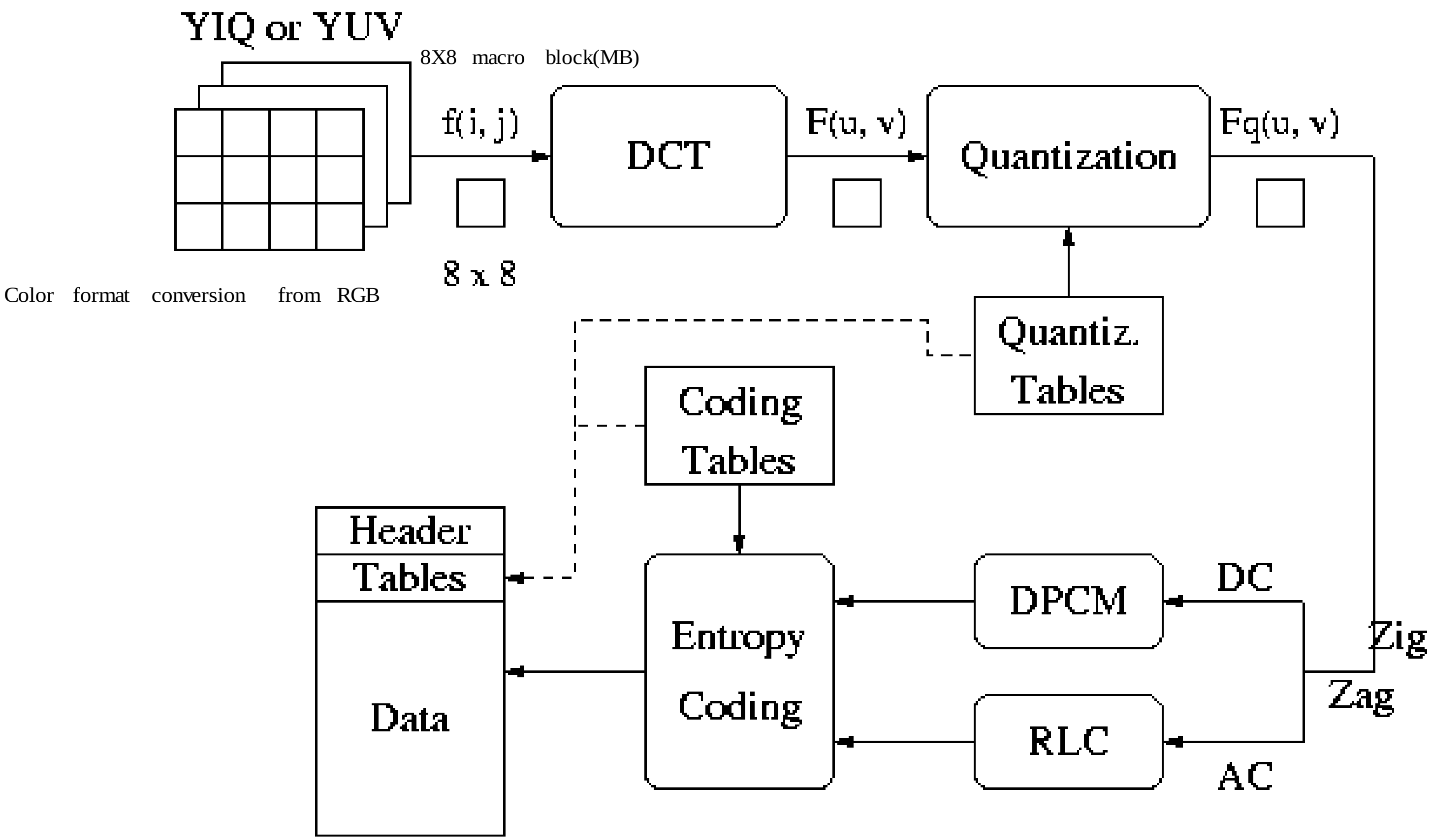
JPEG (Joint Picture Expert Group)

- ISO/IEC JTC 1 : SC29 – WG1
  - Compression algorithm for still color image up to 20:1

- Feature
  - Hybrid coding - Lossless + Lossy Coding
  - Supporting format : B/W, Color format- YUV, YIQ, YCbCr
  - Applications- Still Camera, Image Database,…

- Now moving to JPEG 2000 (Wavelet coding)

JPEG Encoder structure

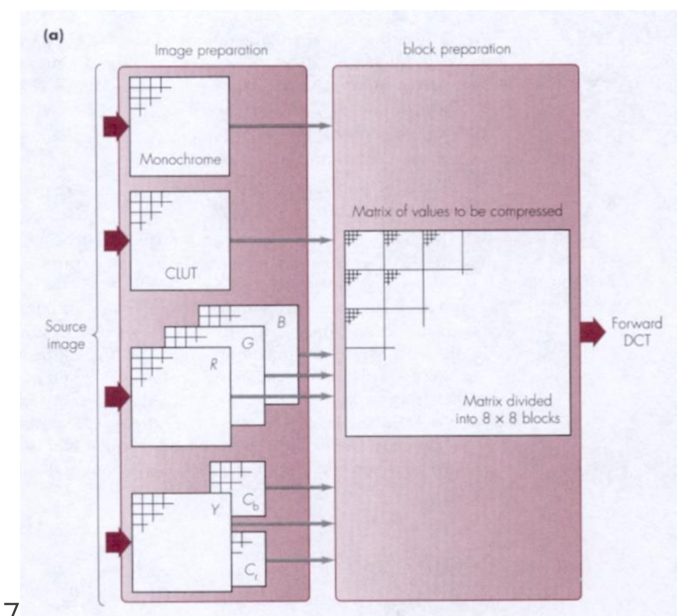


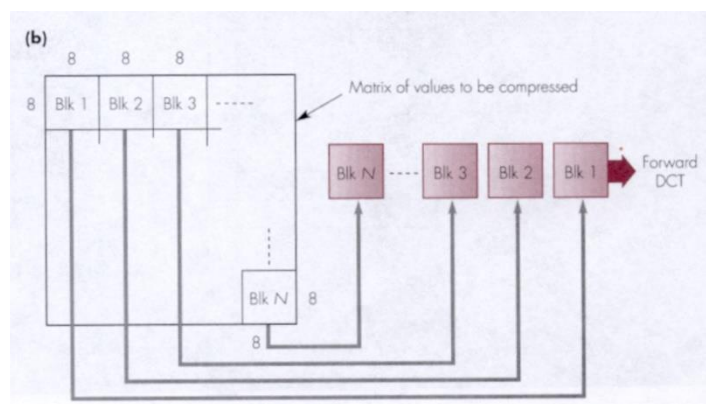
Major Steps in JPEG

- Image preparation (Color format)
- DCT
- Quantization
- Zigzag Scan
- DPCM coding for DC component
- RLE coding for AC component
- Entropy (Huffman) Coding

Image preparation

- Transform color format of image (optional)
  - RGB → YIQ or YUV or YCbCr (see later!)
- Segment into Macro Block (MB)
  - 8 X 8 size block segmentation





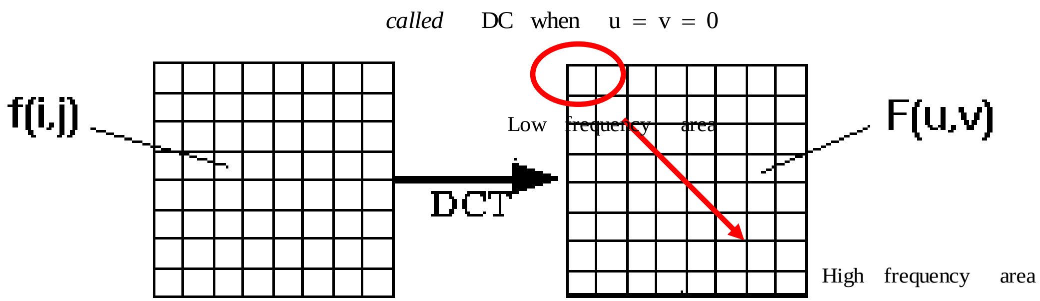
DCT (Discrete Cosine Transform)

- Similar to FFT, transform a spatial-domain signal into frequency components

  - Takes array of pixel values → produces an array of coefficients of frequency components
  - Identify high frequency components and discard
  - Allocate fewer bits for higher frequency area

❖ Conversion 8X8 Block

- Spatial-domain $\Rightarrow$ Frequency-domain



$$F(u,v) = \frac{\Lambda(u)\Lambda(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)\cdot u\pi}{16}\cdot\cos\frac{(2j+1)\cdot v\pi}{16}\cdot f(i,j)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & for\ \xi = 0 \\ 1 & otherwise \end{cases}$$

Quantization

- Data Reduction
  - Reduce number of bits to represent pixel value after DCT
  - $F'[u,v] = \text{round}[\ F[u,v] / q[u,v])$
    - q[u, v] is quantization coefficient
  - Q. Error is main source of Lossy coding

Quantization Example

Low frequency area

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

High frequency area

DCT coefficients

DC

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Quantized

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$\mathrm{round}\left(\frac{-415}{16}\right) = \mathrm{round}(-25.9375) = -26$$

⇨ Same as allocating more bits to low frequency area and less bits to high frequency area

Quantization Table

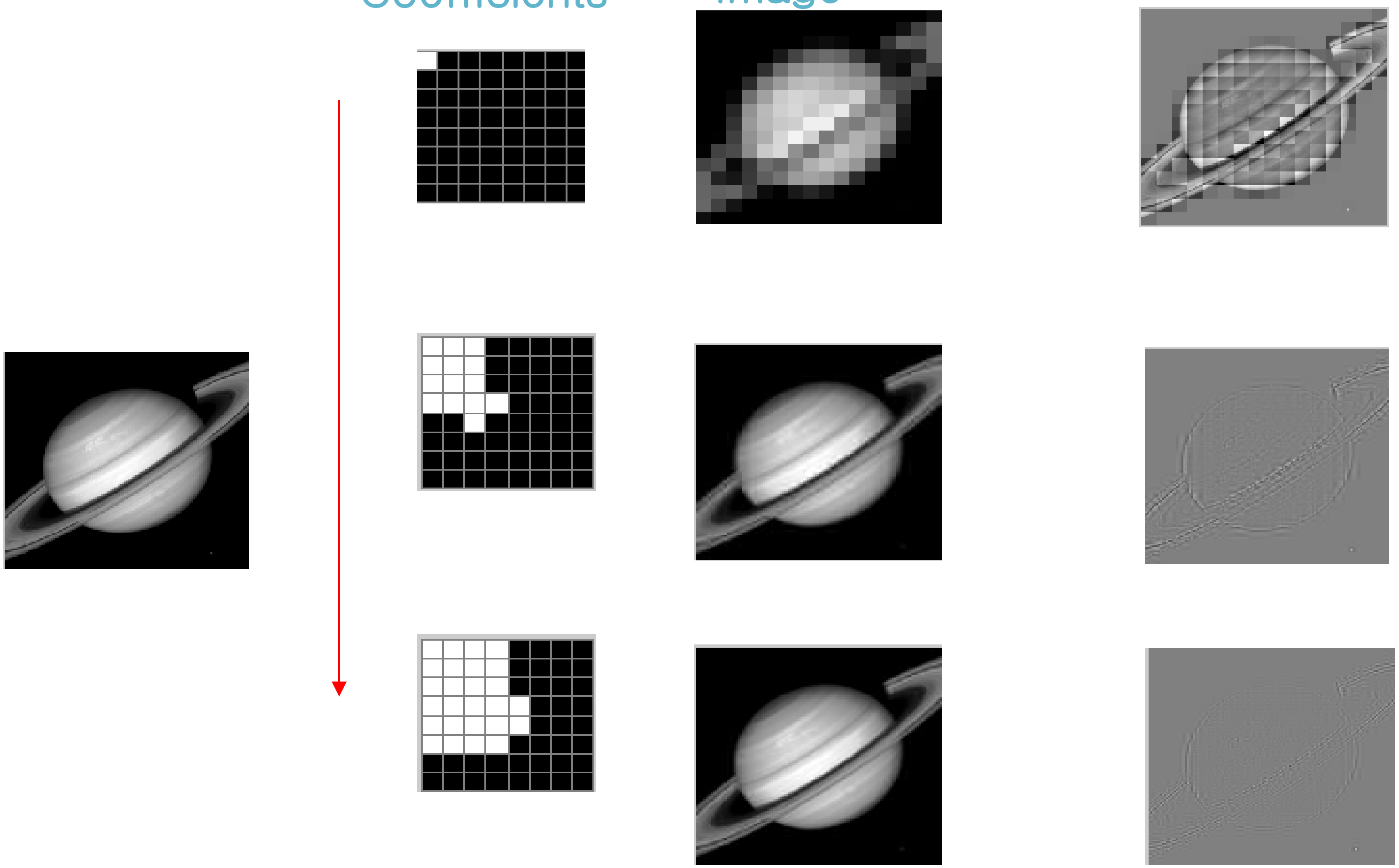
DCT Examples)

Original Image

DCT Coefficients

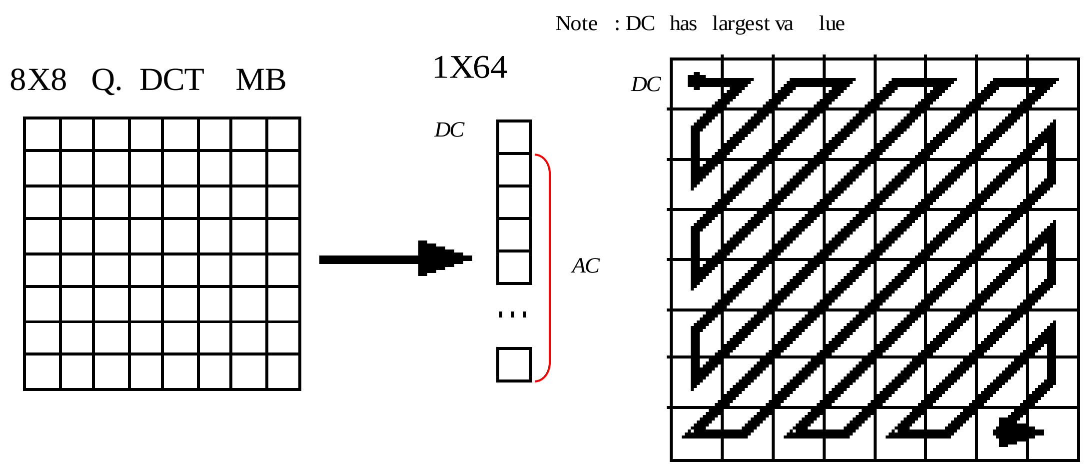
Reconstructed Image

Error Image



Zigzag Scan

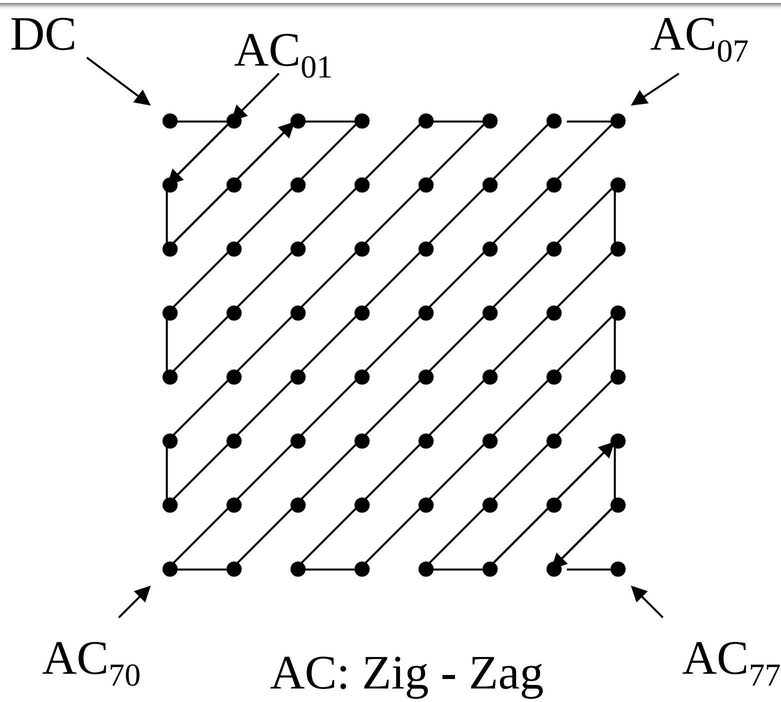
- Maps 8X8 Q. DCT MB into a 1 X 64 vector
- Group low frequency coefficients in top of vector



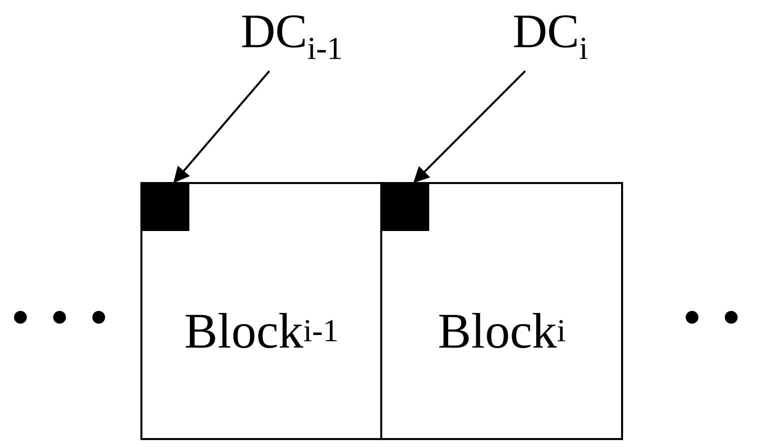
Encoding of Q. DCT Coefficients

- AC Component
  - Do Run Length Encoding (RLE)

- DC component of Quantized DCT
  - DC values of neighboring MB are large and similar
  - Use DPCM (Differential PCM) Coding
    - Encode only the difference between the current 8X8 Macro Block and previous Macro Block



AC: Zig - Zag
RLE

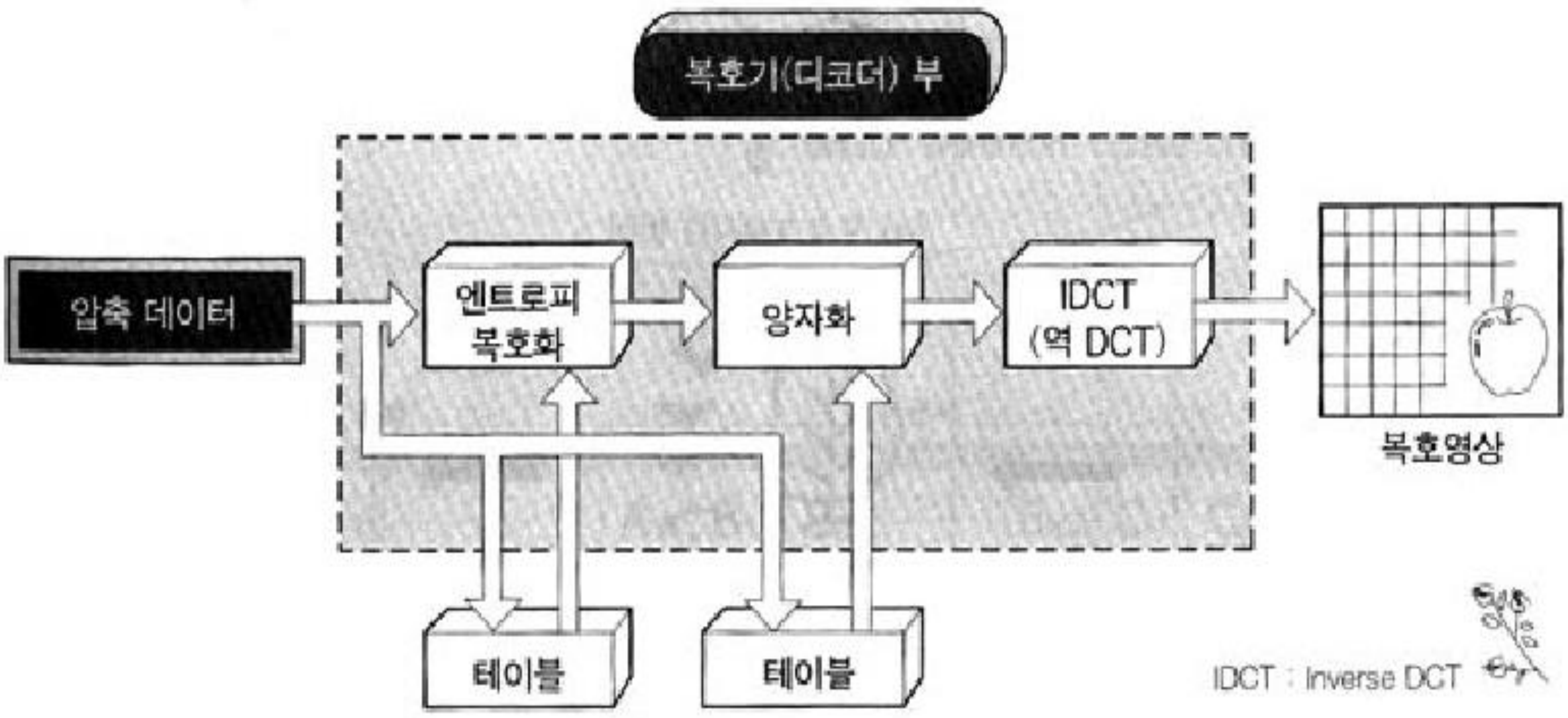


$\triangle DC_i = DC_i - DC_{i-1}$

DC – DPCM coding

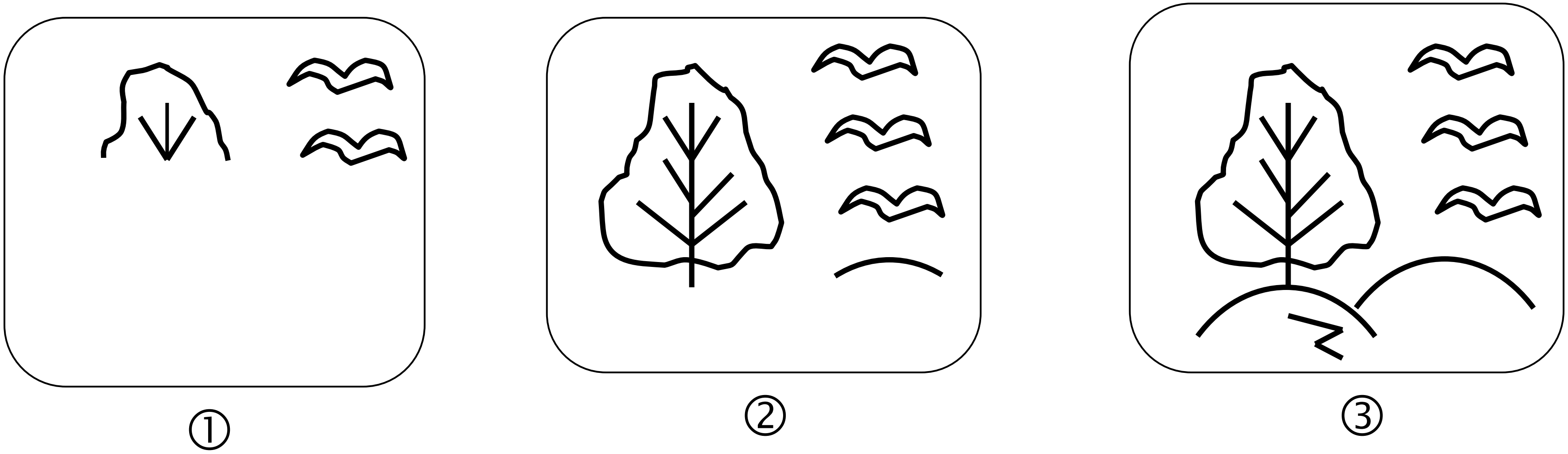
- Entropy Coding- Final stage of JPEG
  → Huffman coding

JPEG Decoder (Inverse of Encoder)



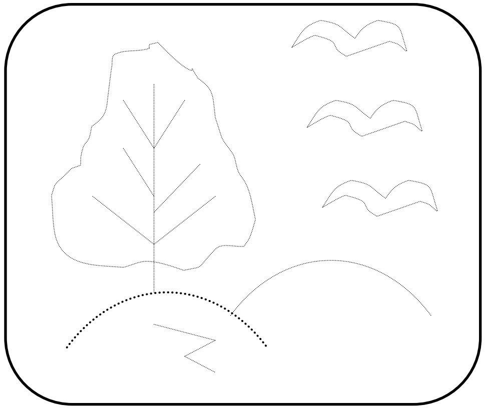
JPEG - Sequential Mode

- Each image is encoded in a single left to right, top to bottom

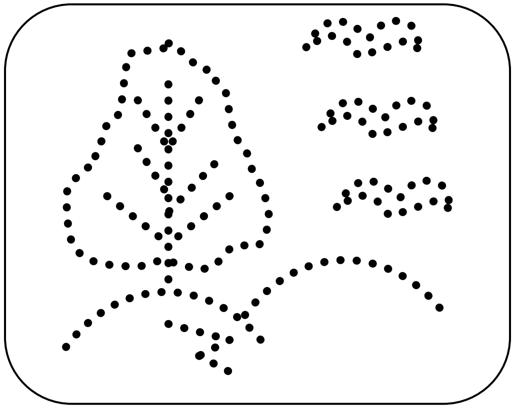


JPEG - Progressive Mode

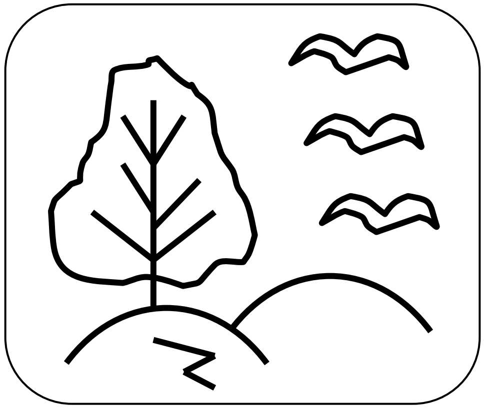
- Goal : Display low quality image and successively improve



①



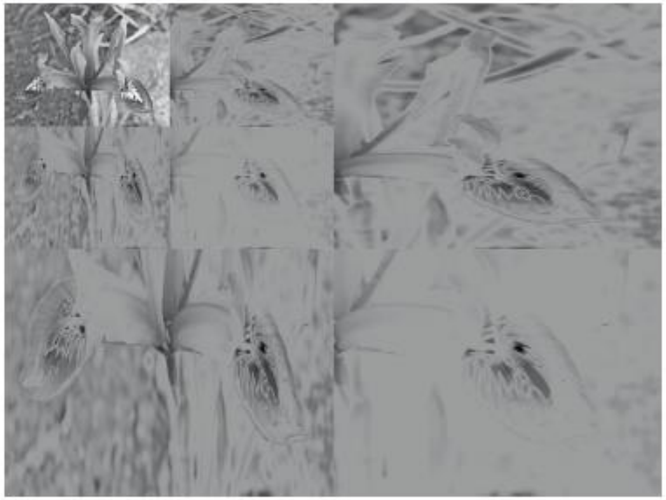
②



③

JPEG2000

- Discrete Wavelet Transform (DWT)
  - Image is divided into tiles. It can be any size up to the entire image
  - Wavelet decomposition : **coarse → fine**
    - Add progressively more detail to the image



Basic Image Processing

- Pixel - Point Processing
  - Histogram, Histogram equalization, Brightness control, Contrast, S-curve
- Pixel - Group Processing
  - Blurring, Sharpening, Noise filtering, Edge detection, Mosaic
- Geometrical transformation
  - Interpolation, reflection, rotation

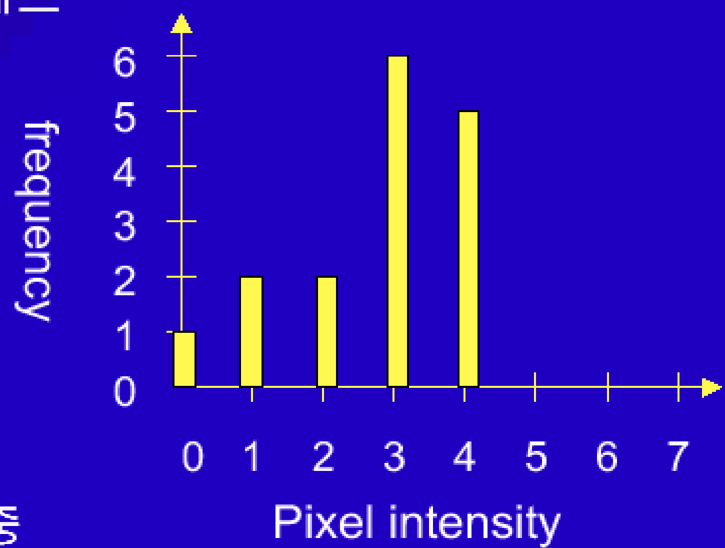
Pixel Point Processing

- Histogram definition

  - 화소가 가진 명암값에 대한 막대 그래프
  - 명암값 : X 축
  - 빈도수 : Y 축

image

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



  - 전체적으로 어두운 영상 => 왼쪽 편중
  - 전체적으로 밝은 영상 => 오른쪽 편중
  - 이상적인 영상 => 균일한 분포
  - 물체의 인식과 구분 (일반적으로 물체는 비슷한 밝기를 가짐)

Image Histogram

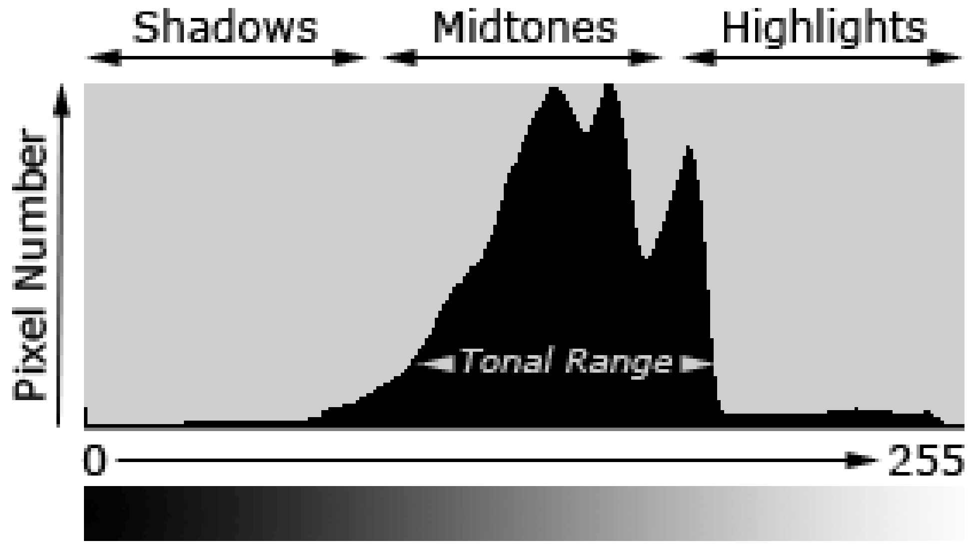
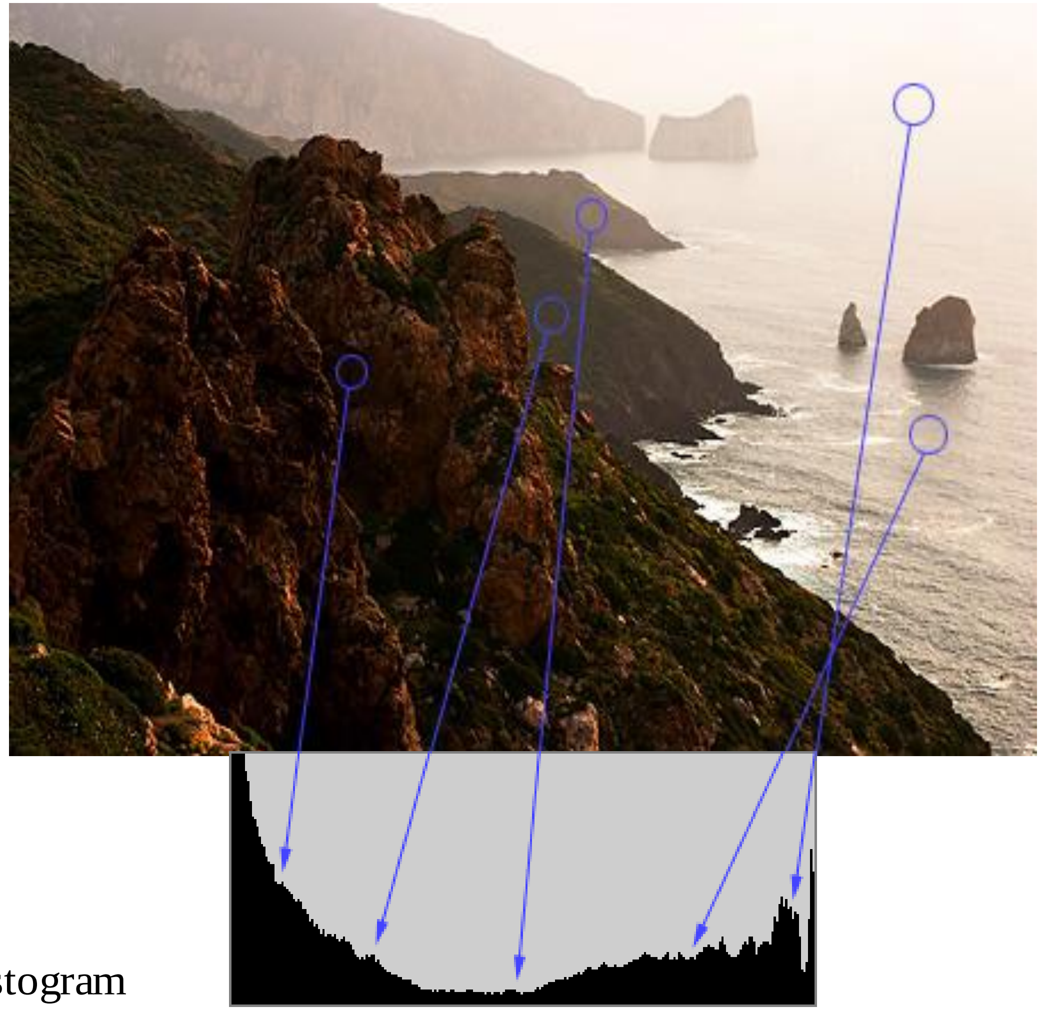


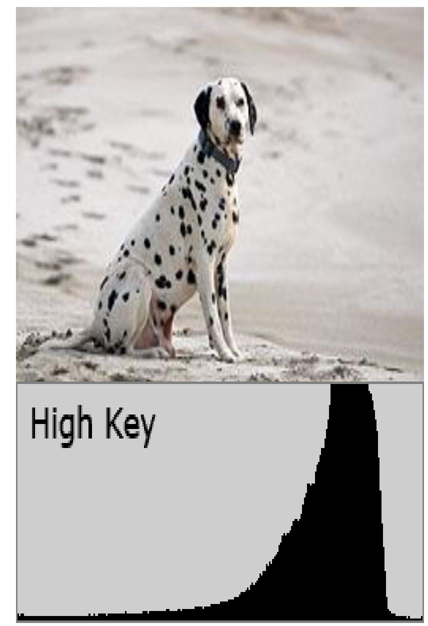
Image tone explains Brightness, Contrast

Image Tone Example



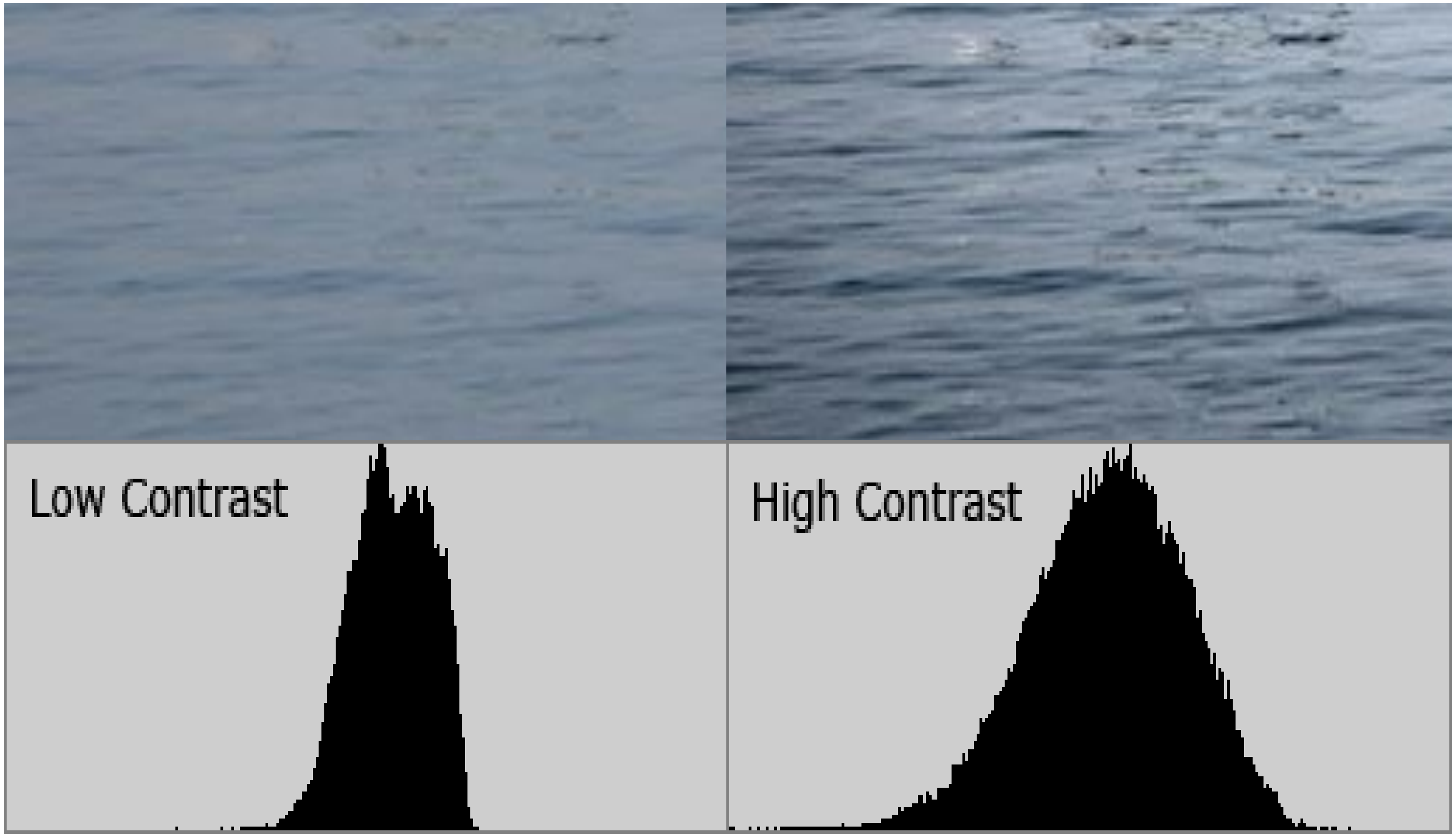
Histogram

Image Histogram Example



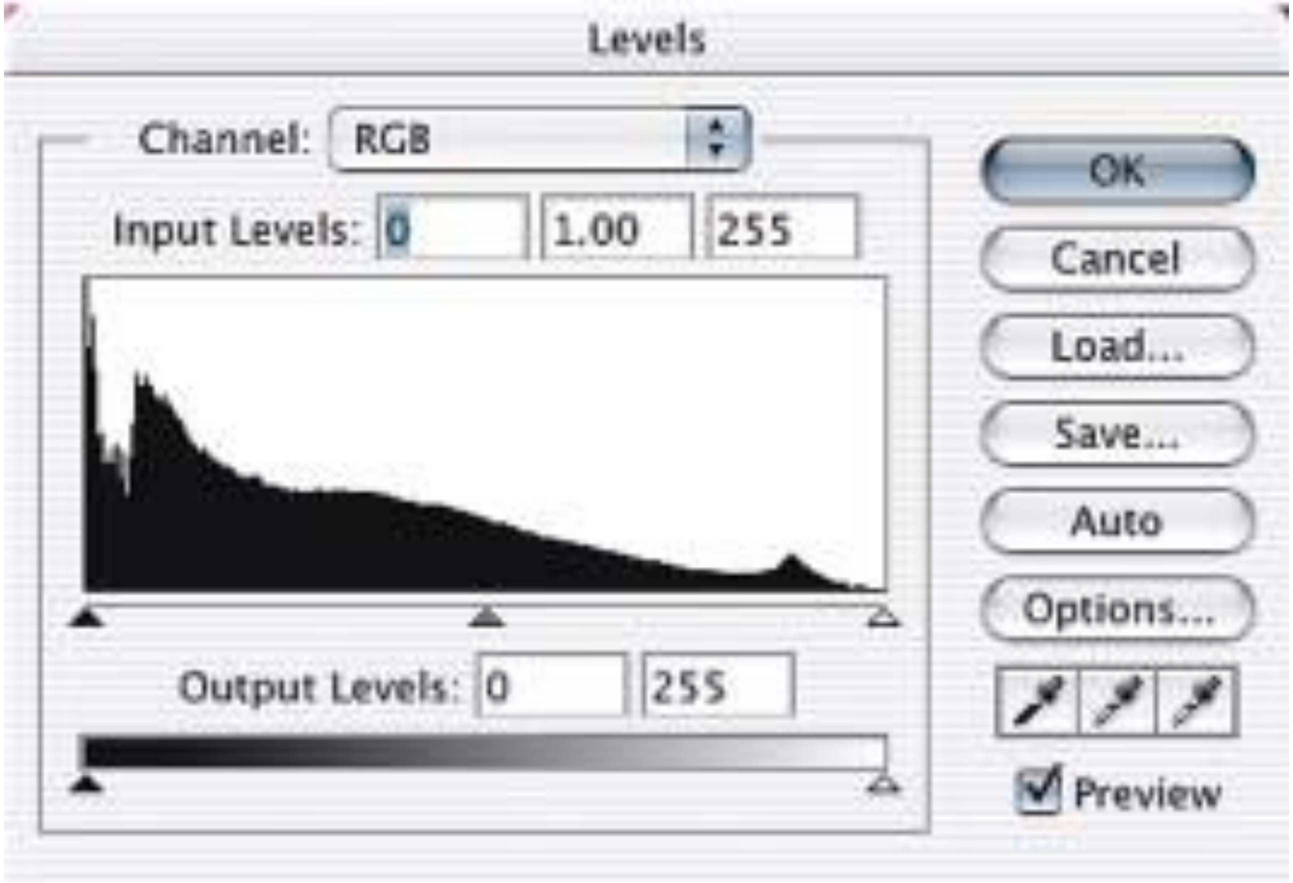
brightness depends on the distributi on of histogram

Image Histogram Example



contrast depends on the width of histogram

Adjustments in Photoshop



Brightness and Contrast Control

- Brightness (pixel processing)
  - Brighter : pixel + Δ
  - Darker : pixel − Δ

- Contrast
  - Stretch or compress the histogram

Brightness Control

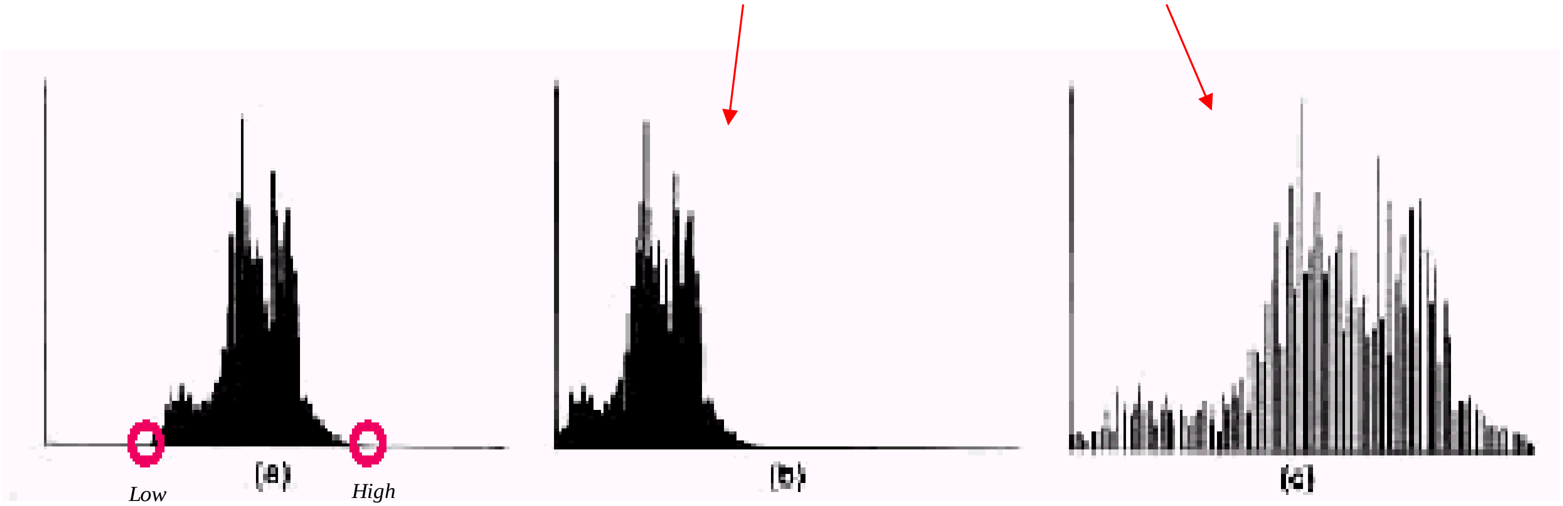
$pixel + 30$

$pixel - 30$

Contrast Stretching

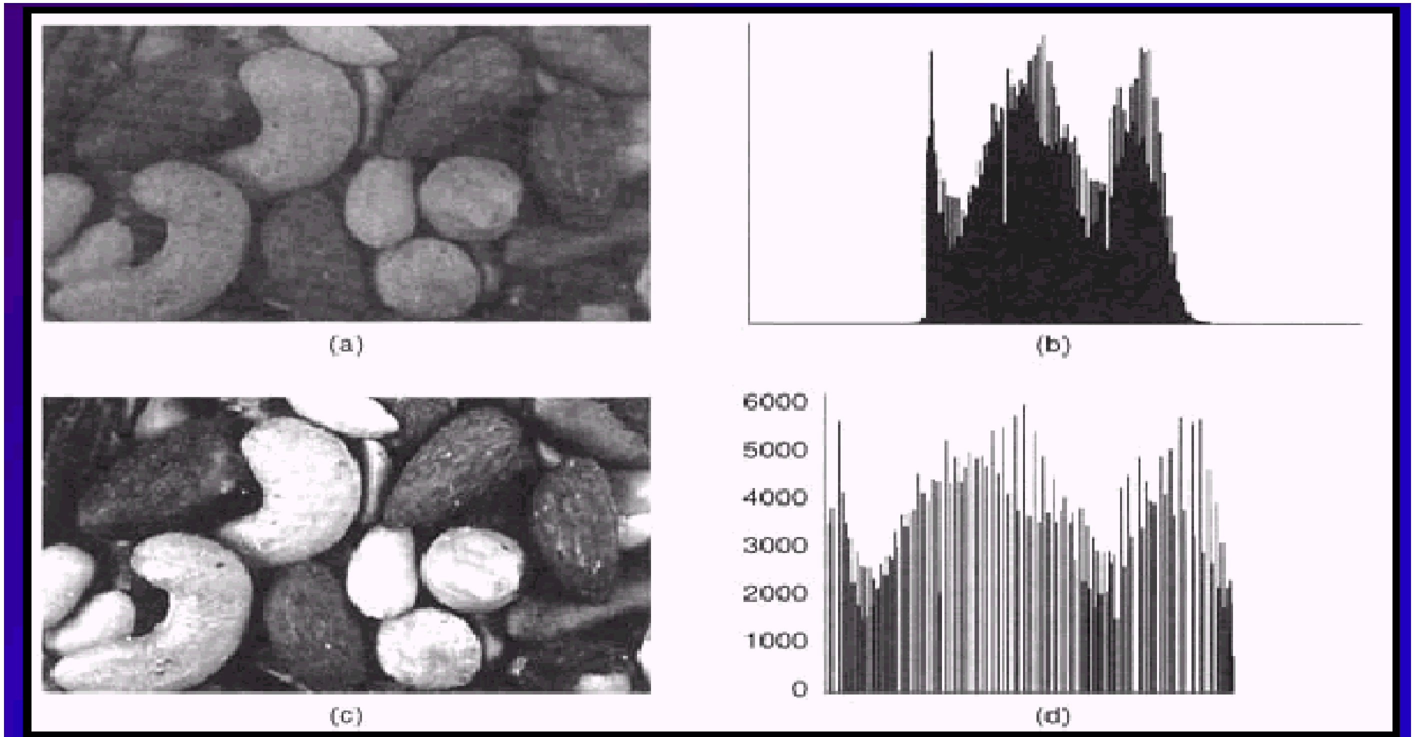
- **Called Histogram Equalization**
  - **Improve image contrast**

$$new\ pixel = [(oldpixel - low)/(high - low)] * 255$$



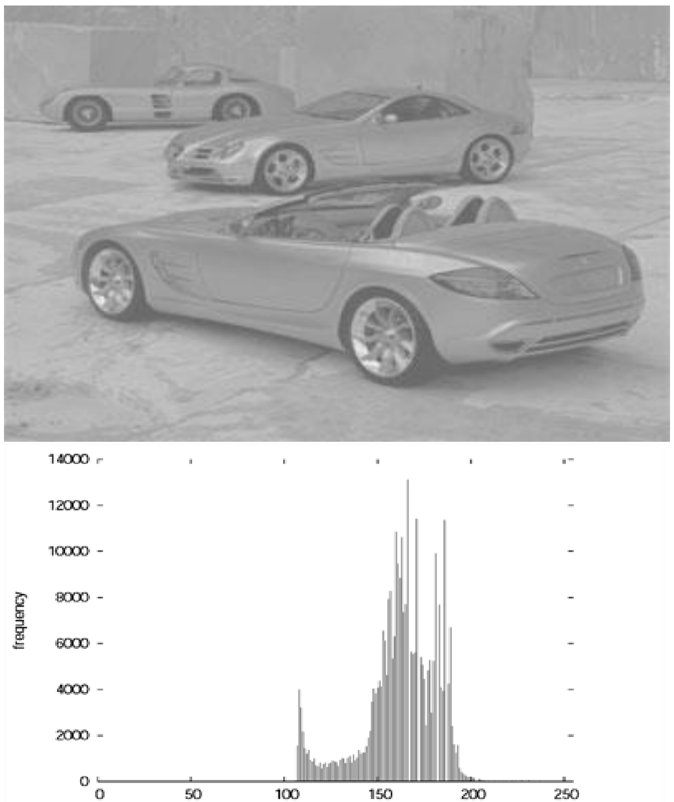
a) Original Histogram :b) histogram – low :c) (histogram – low) * 255 / (high – low)

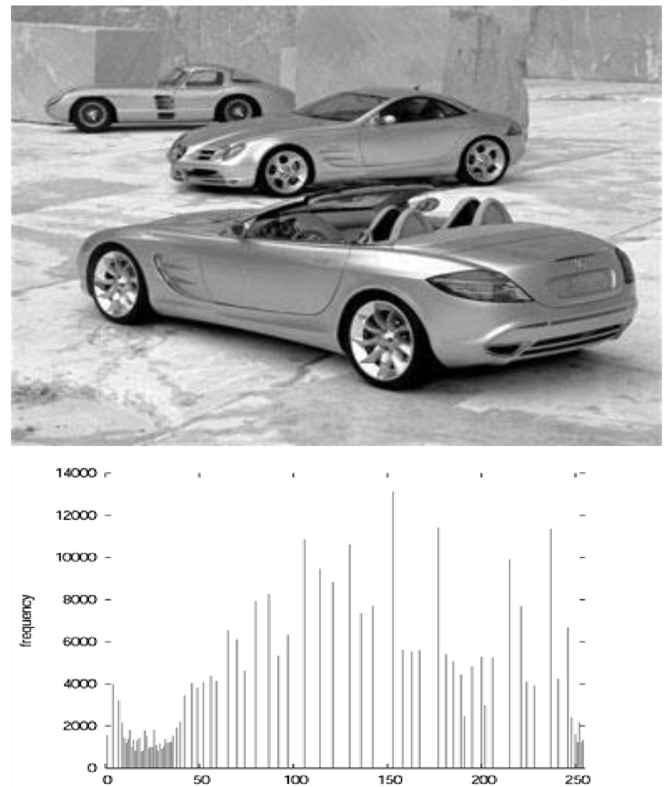
Histogram Equalization



(a) Original Image (b) Histogram
(c) After histogram equalization (d) histogram

Histogram Equalization

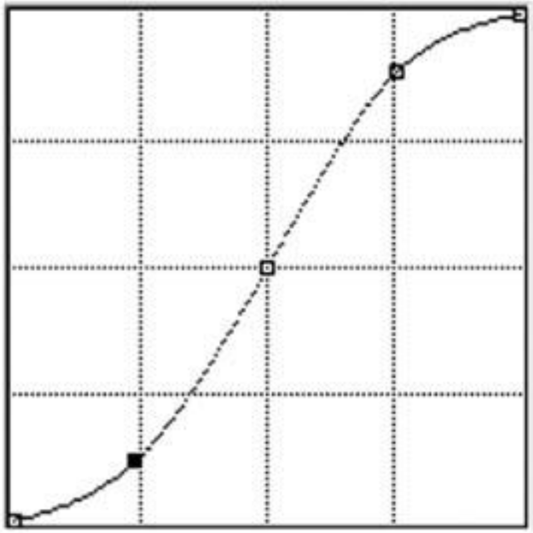




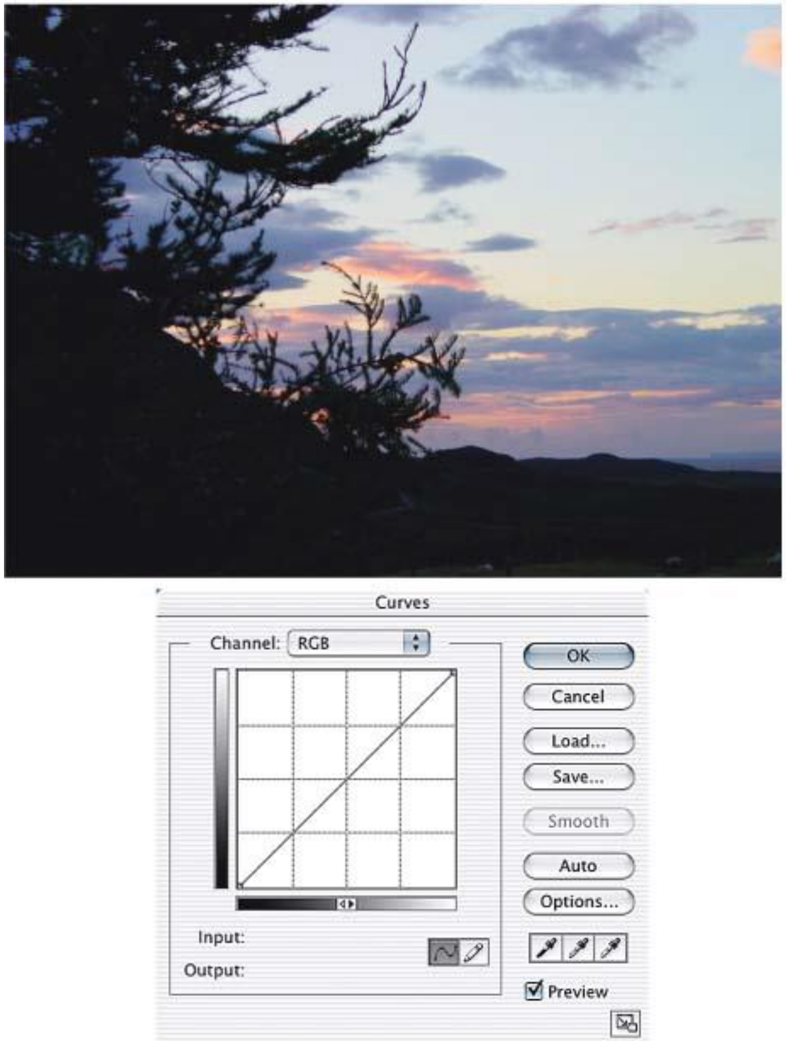
S-curve mapping

- Easy way of
  - Brightness and contrast control
  - Giving special effects

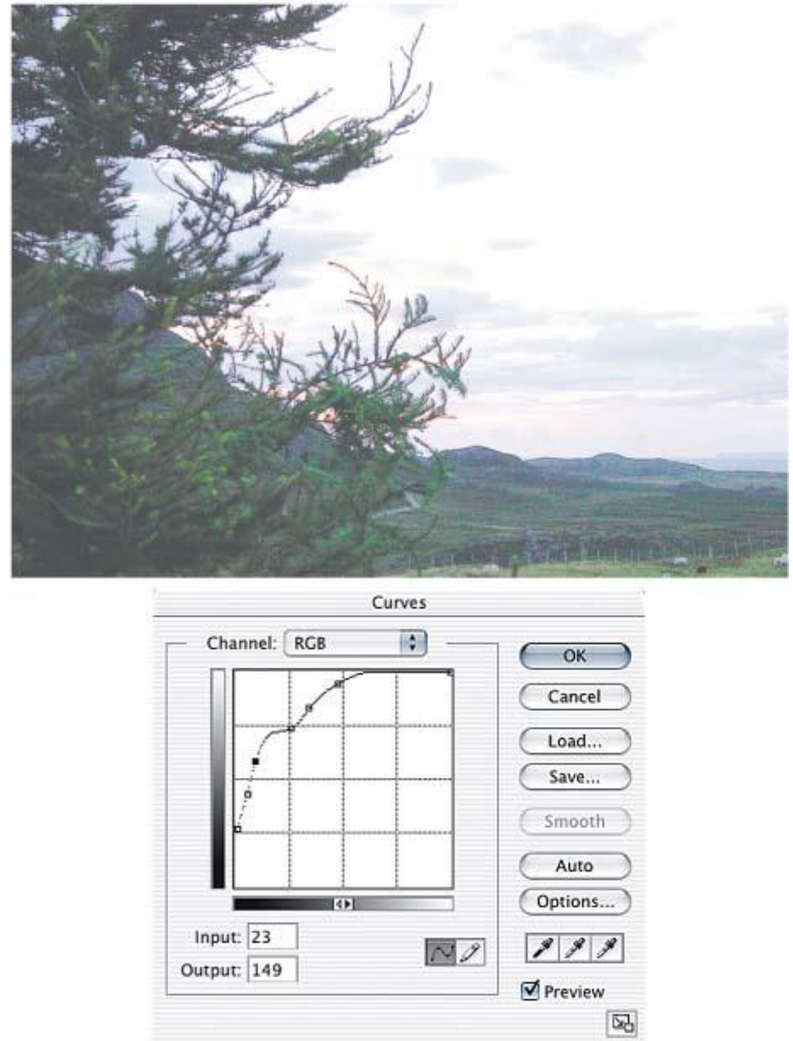
output



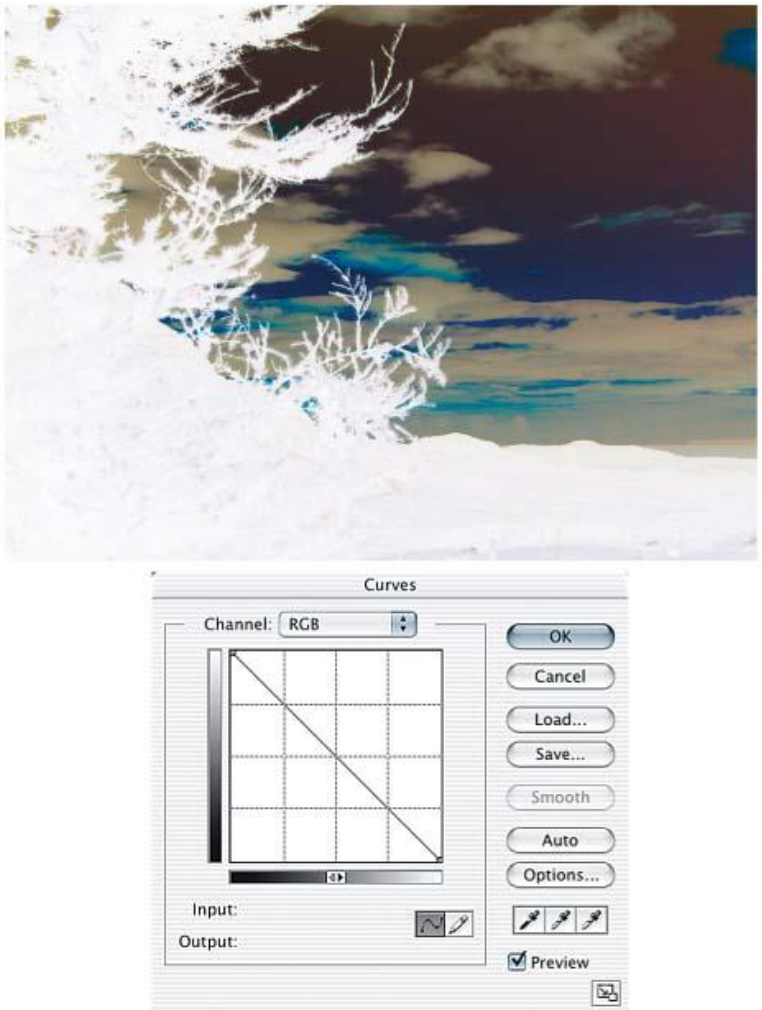
input



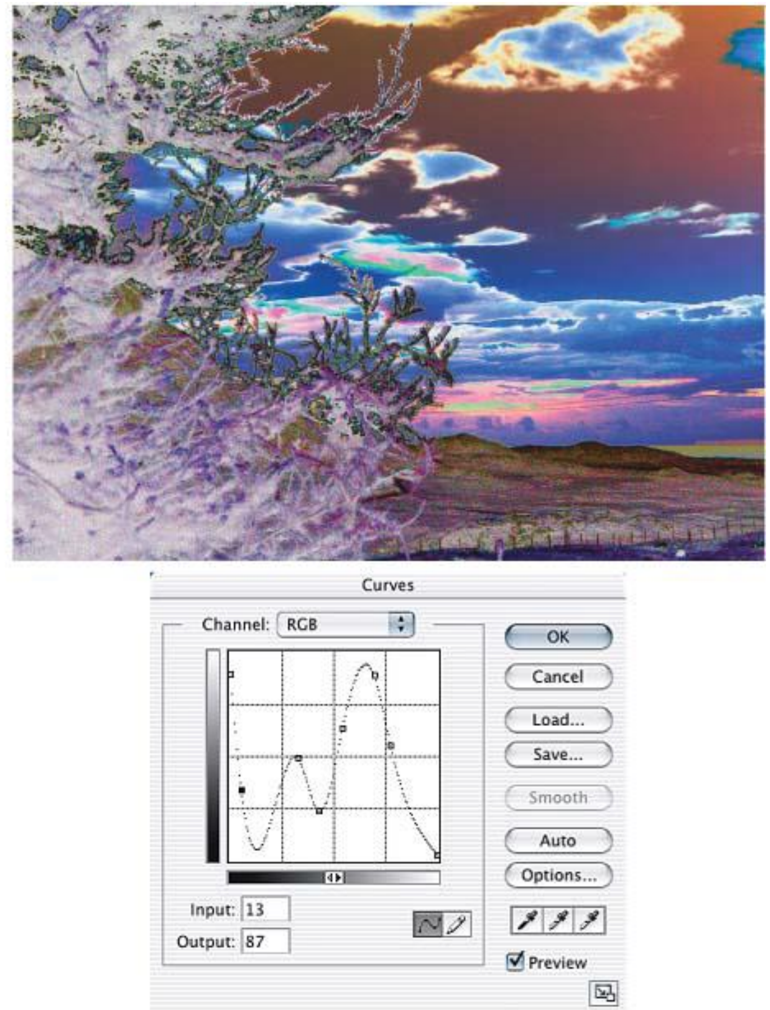
Original photograph



Adjusted to simulate daytime



Reversed



Special effects

Pixel Group Processing

- Image processing based on the Mask
  - Applied to pixel group
  - Mask = Template or window, kernel, filter
  - Convolution = spatial filtering with mask
  - Smoothing, Blurring, Sharpening
  - Noise filtering, Edge detection, Embossing, Mosaic

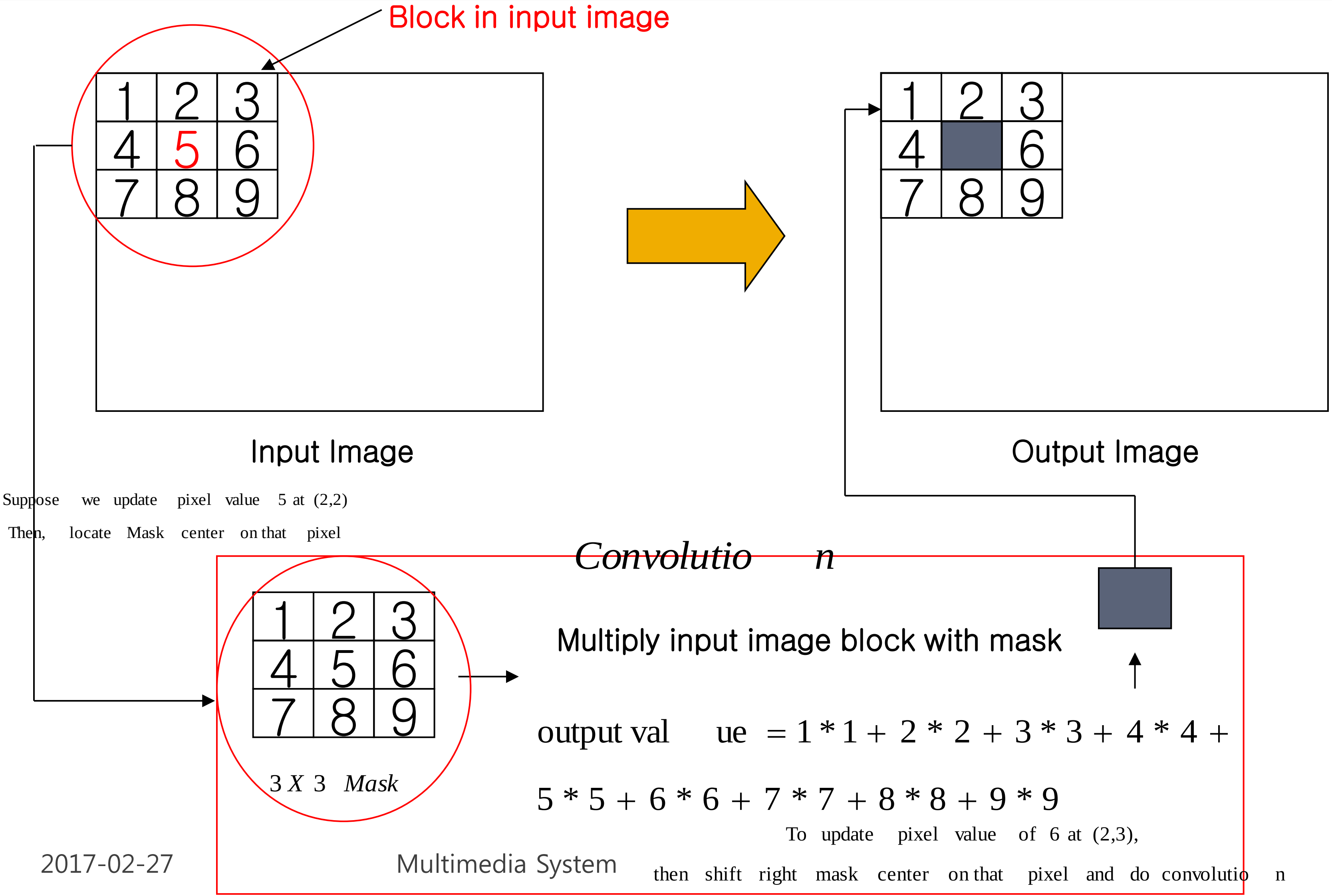

Block in input image
1 2 3
4 5 6
7 8 9
Input Image
1 2 3
4 6
7 8 9
Output Image
Suppose we update pixel value 5 at (2,2)
Then, locate Mask center on that pixel
Convolution
1 2 3
4 5 6
7 8 9
3 X 3 Mask
Multiply input image block with mask
output value = 1 * 1 + 2 * 2 + 3 * 3 + 4 * 4 + 5 * 5 + 6 * 6 + 7 * 7 + 8 * 8 + 9 * 9
To update pixel value of 6 at (2,3),
then shift right mask center on that pixel and do convolution

Image Smoothing - LPF

- Results
  - sharp edge → smooth edge (Blurring)
  - detailed part → blurred

sharp edge

117	117	27
117	117	27
117	117	27
117	117	27
117	117	27

→

smooth edge

117	57	27
117	57	27
117	57	27
117	57	27
117	57	27

Blurring - LPF

- Convolution mask with equal weights
  - Note sum=1

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix}$$

Gaussian Blurring

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

Sharpening

- Eliminate low freq. component.
  - 3x3 convolution mask coefficients all equal to minus value except center, sum=1
  - smooth edge → sharp edge

−1	−1	−1
−1	9	−1
−1	−1	−1

Embossing Effects

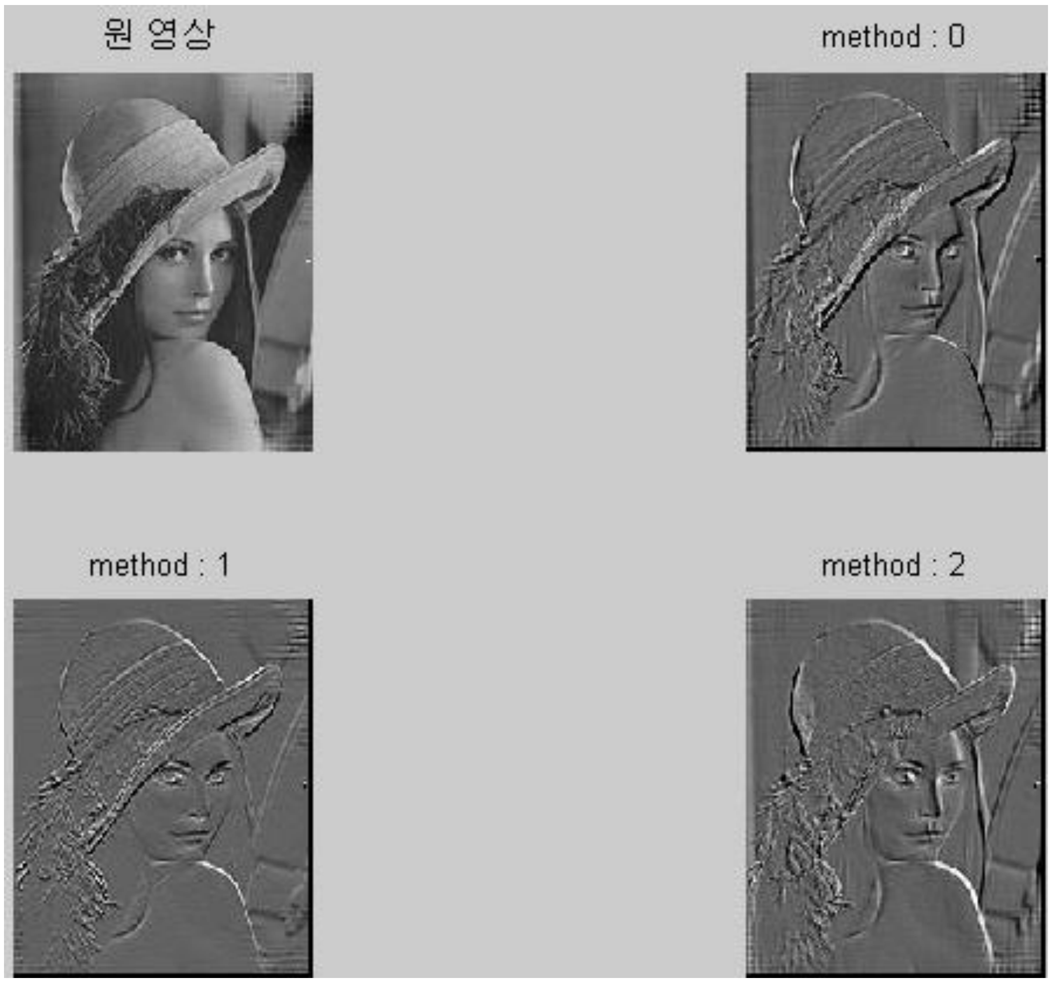
- 3 Mask types (135°, 90°, 45°)

−1	0	0
0	0	0
0	0	1

0	−1	0
0	0	0
0	1	0

0	0	−1
0	0	0
1	0	0

- Mask center value=0, sum= 0


원 영상
method : 0
method : 1
method : 2

Noise Filtering

- Median Filter

4	2	3
2	3	6
7	8	9

Input image

4,2,3,2,3,6,7,8,9

sorting

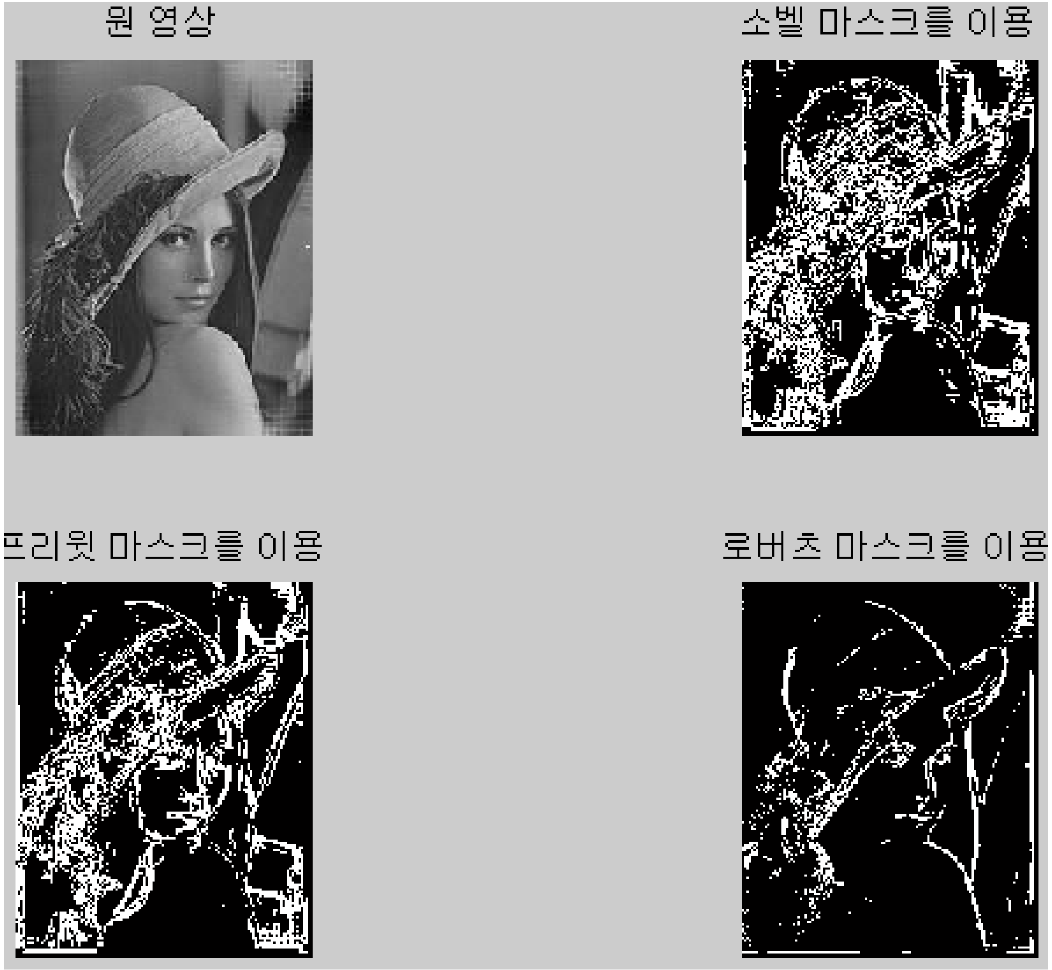
2,2,3,3,**4**,6,7,8,9

This makes smooth change of pixel values

Insert back to the input image

Salt & pepper noise

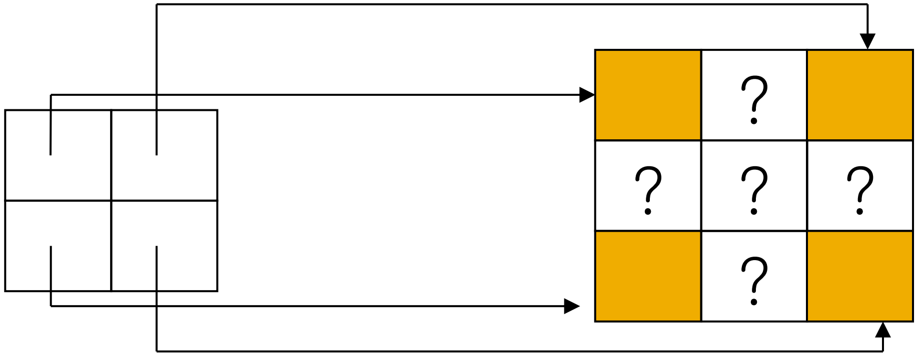
Edge Detection



Interpolation

- 3 Methods
  - Nearest neighbor
    - Use value of pixel in nearest neighbor
  - Bilinear interpolation
    - Use weighted value of all four adjacent pixels
  - Bicubic interpolation
    - Use values of all four adjacent pixels, weighted using cubic splines

- Image zoom in (X 1.5)



- Fill the empty pixel value

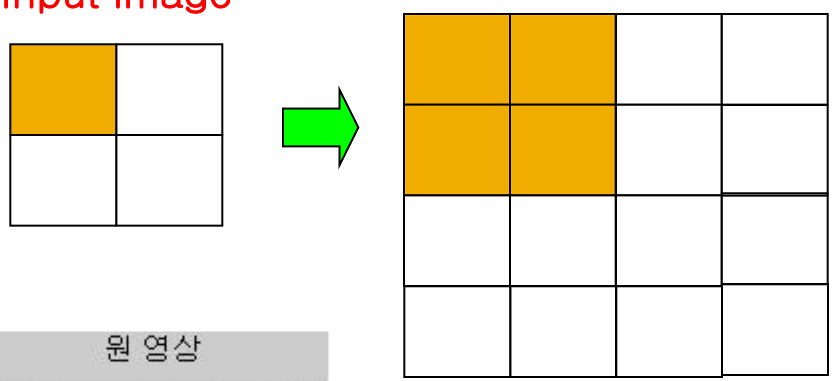
→ Called interpolation

❖ Nearest neighborhood interpolation

- Simplest, but mosaic effect

Input image

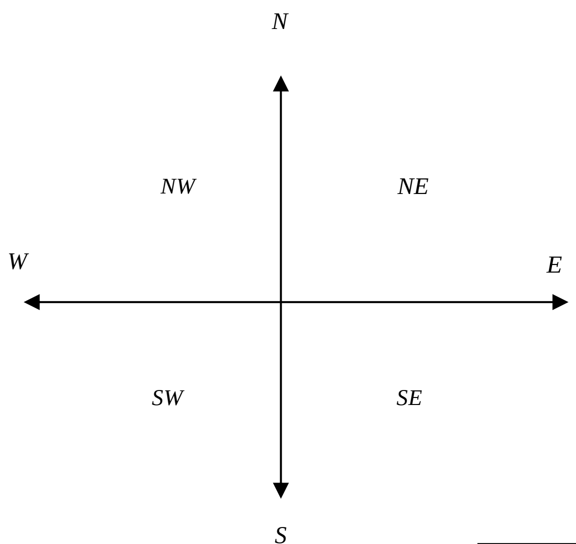
Simple repetition

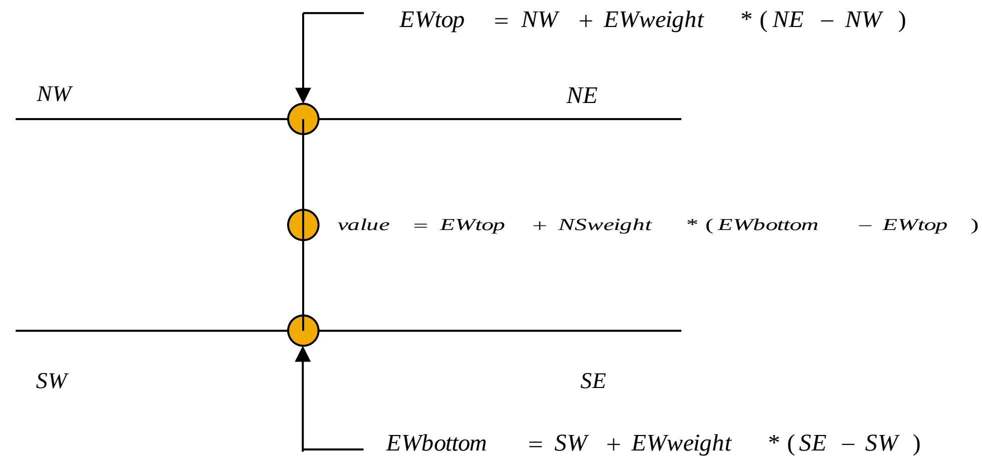


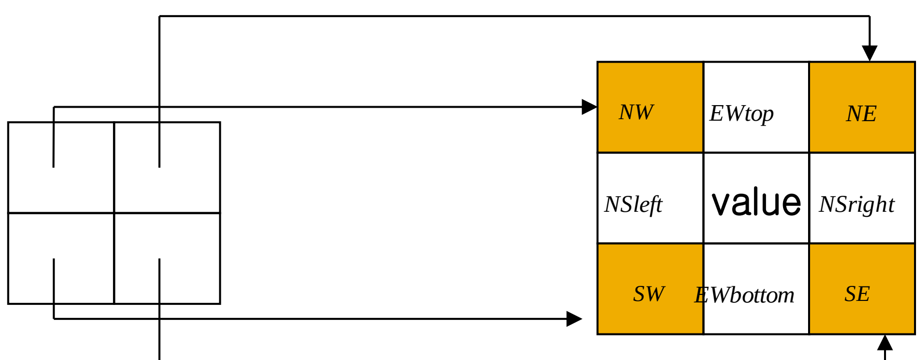
X 2 output image

Bilinear Interpolation

Giving a weight : *EWweight* , *NSweight*







X 1.5

❖ Image zoom with Bilinear method

- Comparison

Nearest-neighbour (left), bilinear (middle) and bicubic (right) interpolation

Image Rotation

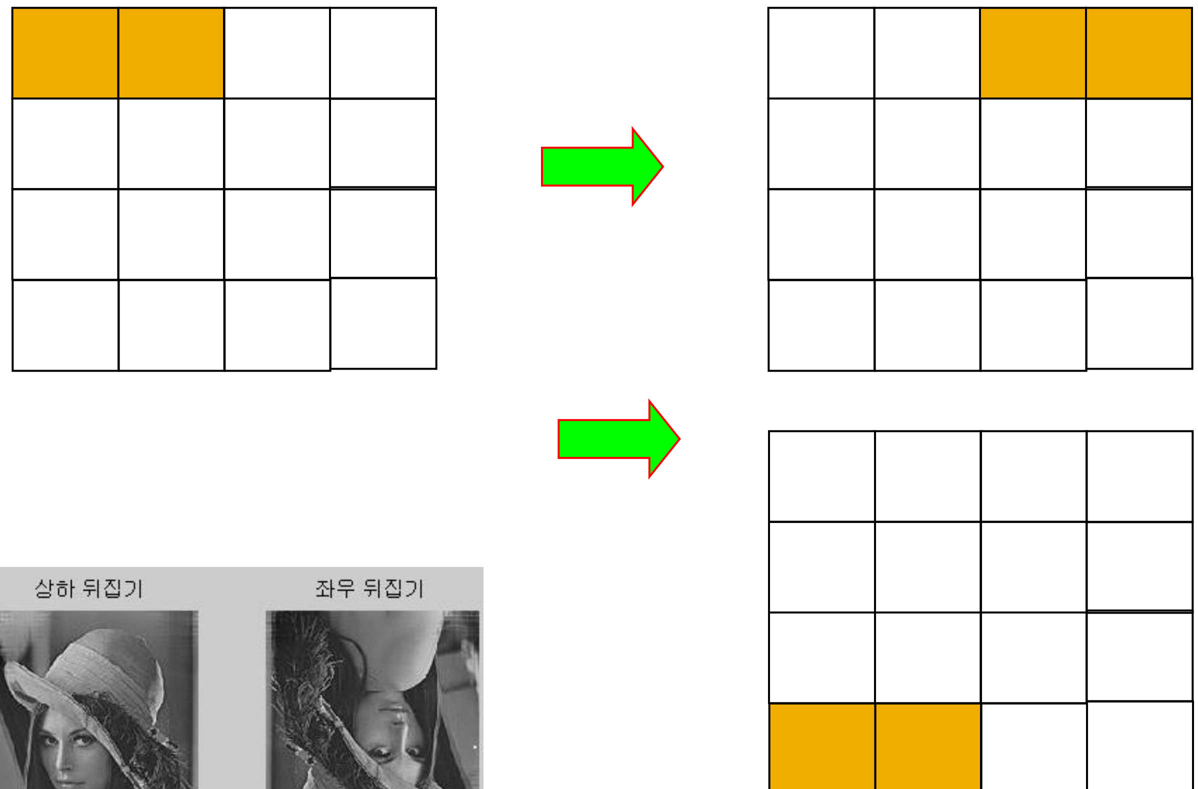
New pixel location

$$x' = (x - centerX) \cdot \cos\theta - (y - centerY) \cdot \sin\theta$$

$$y' = (x - centerX) \cdot \sin\theta - (y - centerY) \cdot \cos\theta$$

Image Reflection

vertical mirroring



Flip, horizontal mirroring

Homework

- Do image operations in lecture PT using image editing program such as Photoshop

Digital Camera

- Picture – 빛의 예술 (Art of light)
  - 노출 (Exposure) - 빛의 양 (light amount)

노출 (Exposure) – 3가지 요소

- 조리개 (Aperture, F stop, Av)
- 셔터스피드 (Shutter speed, Tv)
- ISO (감도)

조리개 (Aperture)

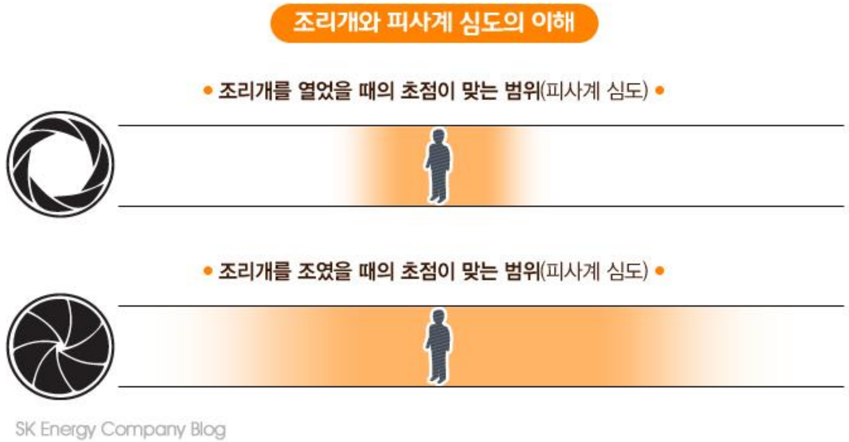
- Camera 렌즈에 있는 움직이는 막(shield)
- 조리개를 조절해 빛을 받아들이는 양 조절

F=32 F=2.8

- Camera 렌즈는 최대 조리개 개방 수치에 따라 성능 및 가격 결정 (F=1.8 > F=2.8)
- 카메라 Av 모드
  - 조리개 설정 수치에 따라 셔터스피드 자동 조절

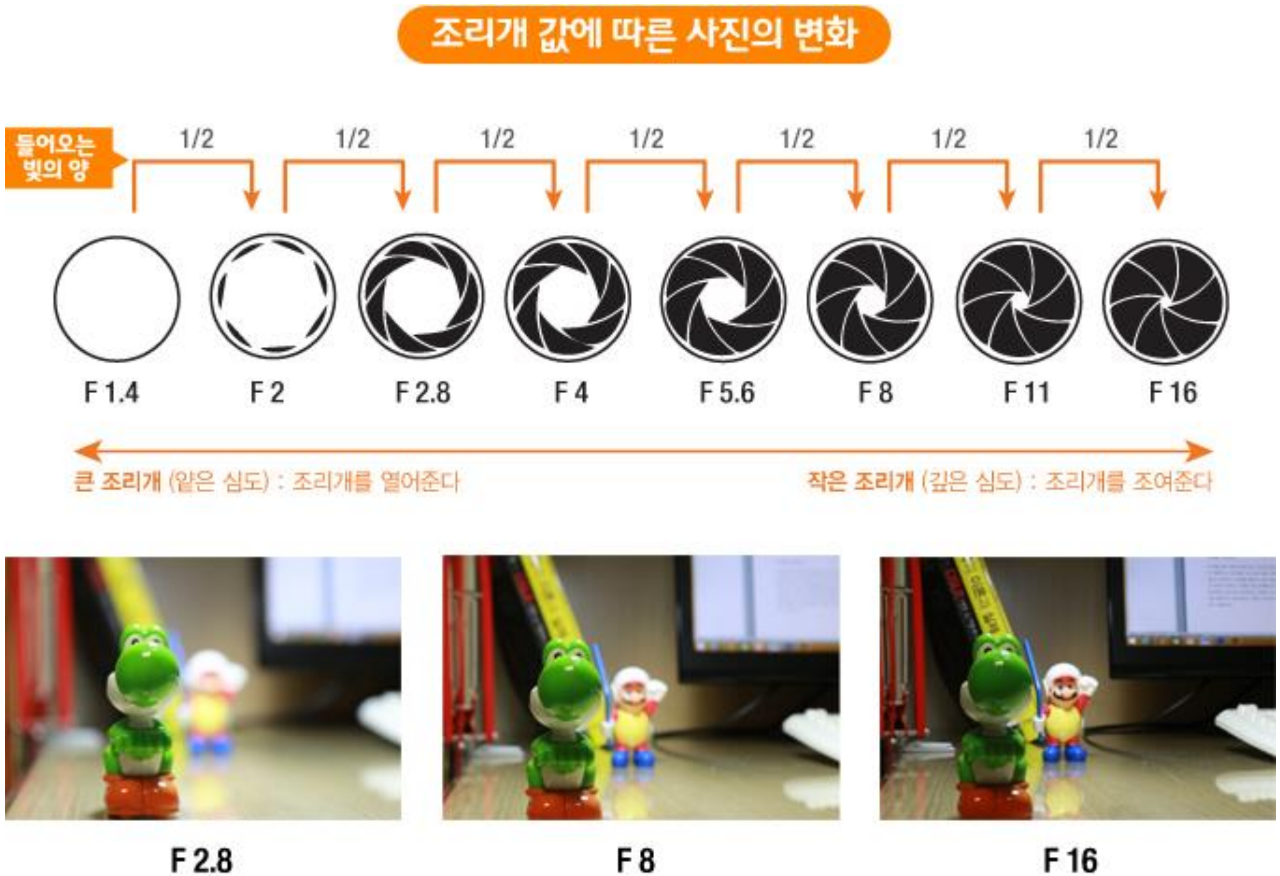
❖ 조리개 - Pan focus and Out focus

- 조리개는 피사계 **심도 조절 기능**
- Pan focus (풍경 사진)
  - 조리개를 조여(F↑)서 심도가 깊은 사진
- Out focus (인물 사진)
  - 조리개를 개방해서(F↓)서 심도가 낮은 사진


조리개와 피사계 심도의 이해
• 조리개를 열었을 때의 초점이 맞는 범위(피사계 심도) •
• 조리개를 조였을 때의 초점이 맞는 범위(피사계 심도) •
SK Energy Company Blog


eneloop


eneloop



SK Energy Company Blog

❖ 셔터 스피드(Shutter speed)

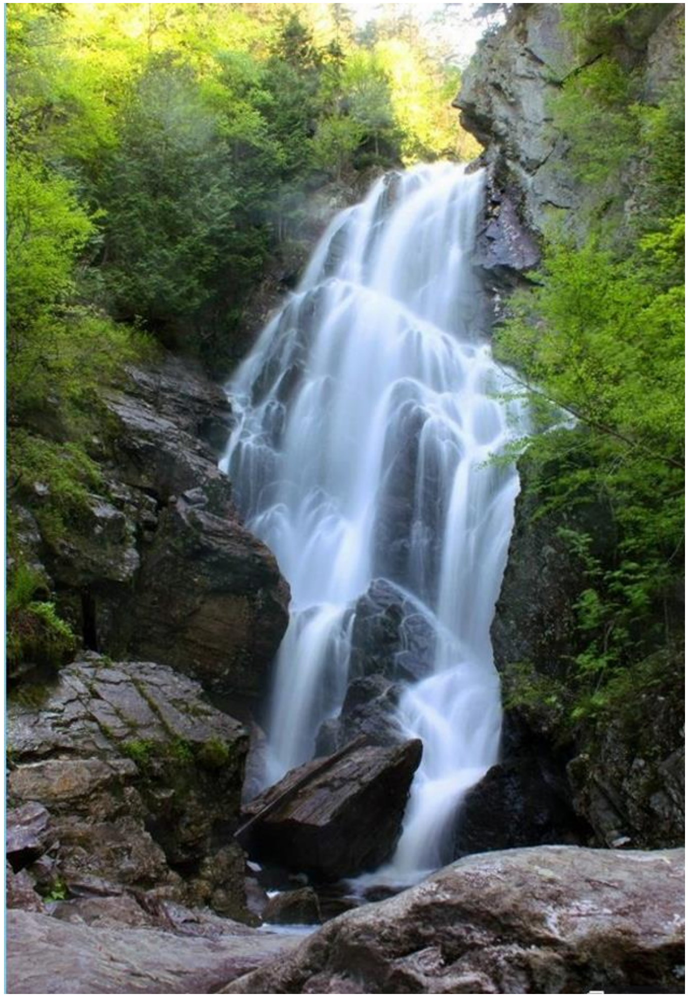
- 카메라에 장착되어 있는 검정색 막. 막 뒤에는 CMOS 센서가 위치하고 있음
- 셔터(Shutter) 버튼을 누르면 순간 shutter가 열리면서 노출 결정
  - Bulb 모드(셔터를 누르는 동안 무한대 노출)
  - 15sec ~ 1/2000sec

- Shutter speed를 설정해 느리게 또는 빠르게 움직이는 물체를 찍을 때 설정


셔터스피드에 따른 사진의 변화
들어오는 빛의 양
1/2
1/2
1/2
1/2
1/2
1/2
1/2
1/2초
1/4초
1/8초
1/15초
1/30초
1/60초
1/125초
1/250초
느린 셔터스피드 : 빛의 양이 많아진다
빠른 셔터스피드 : 빛의 양이 적어진다
60분의 1초
250분의 1초
SK Energy Company Blog

Shutter speed – 떨어지는 pen

- (1/30sec) 손 떨림으로 인해 흐릿함 < 1/60sec
- (1/640sec) 물체를 고정시켜 보이는 효과
- 경주 자동차등의 빠른 물체를 찍을 때는 Shutter speed ↑, 계곡 물 등에 흐릿한 효과 ↓



조리개와 shutter speed를 이용한 노출

- 조리개와 shutter speed는 반비례
  - 조리개를 많이 열면, shutter speed는 빨라야 하고, 조리개를 조이면, shutter speed를 느리게 조절
  - 조리개를 한 스톱(one stop) 높이면, shutter speed를 한 스톱 낮춤.
  - One stop ~ 빛을 받아 들이는 양이 정확히 2배가 되는 정도를 의미하는 용어

❖ 조리개와 shutter speed 수치 조절

- Example) 적정 노출

  조리개 f11, shutter speed 1/125

  ISO = 100

- 조리개와 shutter speed 상관 관계

조리개	2.8	4	5.6	8	11	16	22	32
	↕	↕	↕	↕	↕	↕	↕	↕
Shutter speed	1/2000	1/1000	1/500	1/250	1/125	1/60	1/30	1/15

- ISO –광 감도
  - CMOS 센서가 빛에 반응하는 감도 값
  - 조리개와 shutter speed 만으로 적정 노출을 얻기 어려운 어두운 환경에서 ISO 값을 높여서 인위적을 밝은 사진 가능
    - ISO : 100 ~ 25600
    - 일반적으로 ISO 적정 값은 100 또는 200
    - ISO를 높이면 Noise가 끼게 됨

SK Energy Company Blog

- 카메라 모드
  - 조리개 우선 모드 (Av)
    - 조리개 설정(Pan or Out focus)하면 shutter speed와 ISO는 카메라가 조절
  - 셔터 스피드 우선 모드 (Tv)
    - 셔터 스피드 조절하면 조리개와 ISO 자동 설정
  - M(Manual) 모드
    - 조리개, 셔터 스피드, ISO등 수동 조절
  - 자동 모드
    - 카메라가 모든 수치 자동 조절

- Book
  - Add R programming examples from the internet