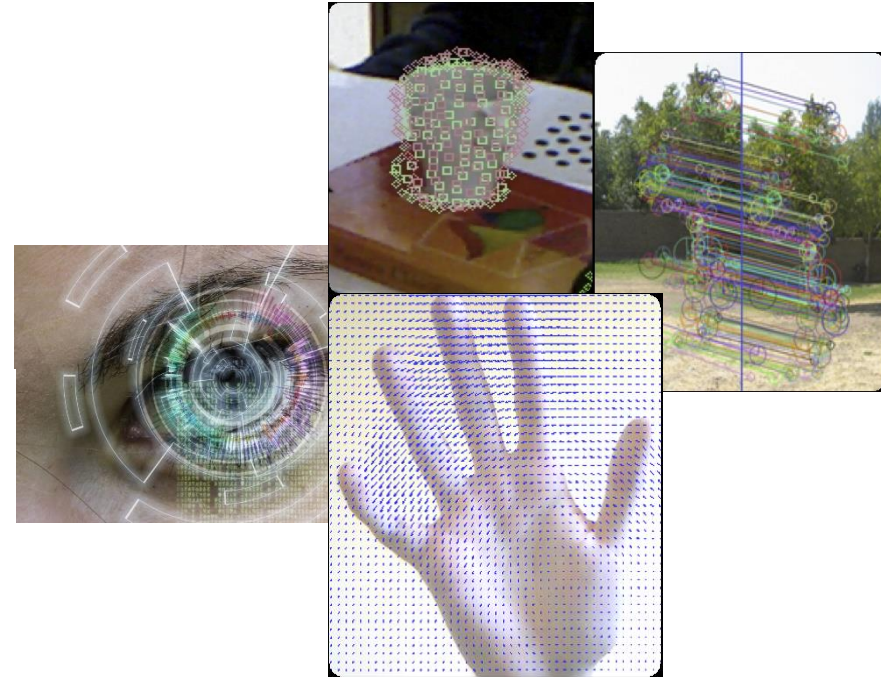


2023 Fall

COMPUTER VISION

비전
프로그래밍



chap.2 영상 처리(image processing) 소개 및 화
소 처리 (pixel processing)

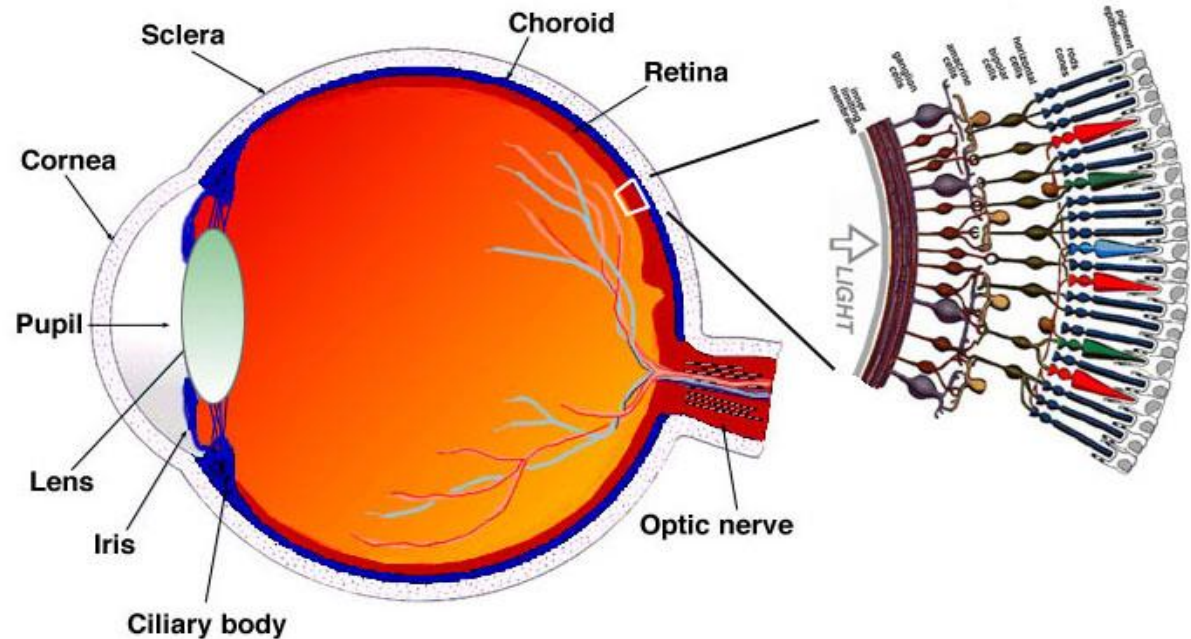
강의 목표

- 사람의 시각적 데이터 센싱 구조 이해한다.
- 화소(pixel)의 정의를 이해한다.
- 영상처리 기술의 전반적 내용을 이해 한다.
- 화소 점처리(pixel point processing)가 무엇인지 이해한다.
- 다양한 화소 점처리 (pixel point processing) 기법에 대해서 이해한다.

2.1 Structure of the Human Eye (1)

■ Eye

- Nearly sphere, average diameter approximately of 20 mm.



requires some knowledge of how we see colors

Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

2.1 Structure of the Human Eye (2)

- Some Elements
 - Cornea and sclera: outer cover
 - Choroid: a network of blood vessel that serve as the major source of nutrition to the eye.
 - Iris: control the amount of light that enter the eye.

- Lens: layers of fibrous cells (60 to 70% water)

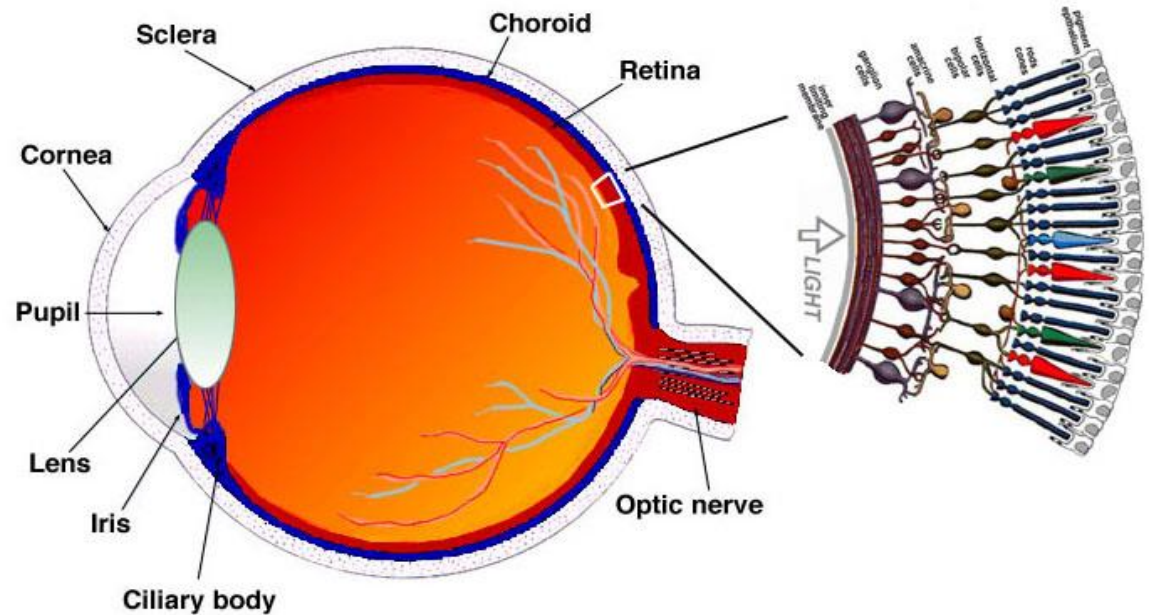


Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

2.1 Structure of the Human Eye (3)

- Pupil: the central opening of the iris, varies in 2mm to 8mm.
- Retina
 - Innermost membrane of the eye
 - When the eye is focused, light from object is imaged on this retina.
 - Distribution of discrete light receptors (**cones** and **rods**)

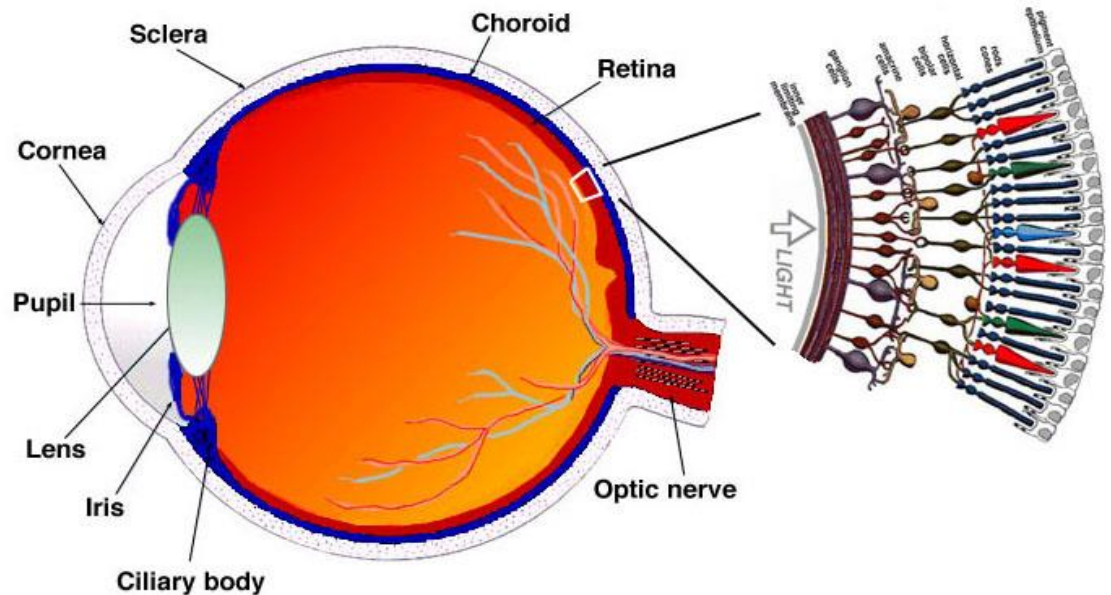


Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

2.1 Structure of the Human Eye (4)

- Retina
 - Cones
 - » Usually number of between 6 and 7 million.
 - » Located in the central of the retina (called the **fovea**) and highly sensitive to color.
 - » Can resolve fine details largely (one→its own nerve end)
 - Rods

- Much larger: 75 to 150 million
- Several rods are connected to a single nerve end.
- General, overall picture of the field of view.
- They are not involved in color vision and are sensitive to low levels of illumination.

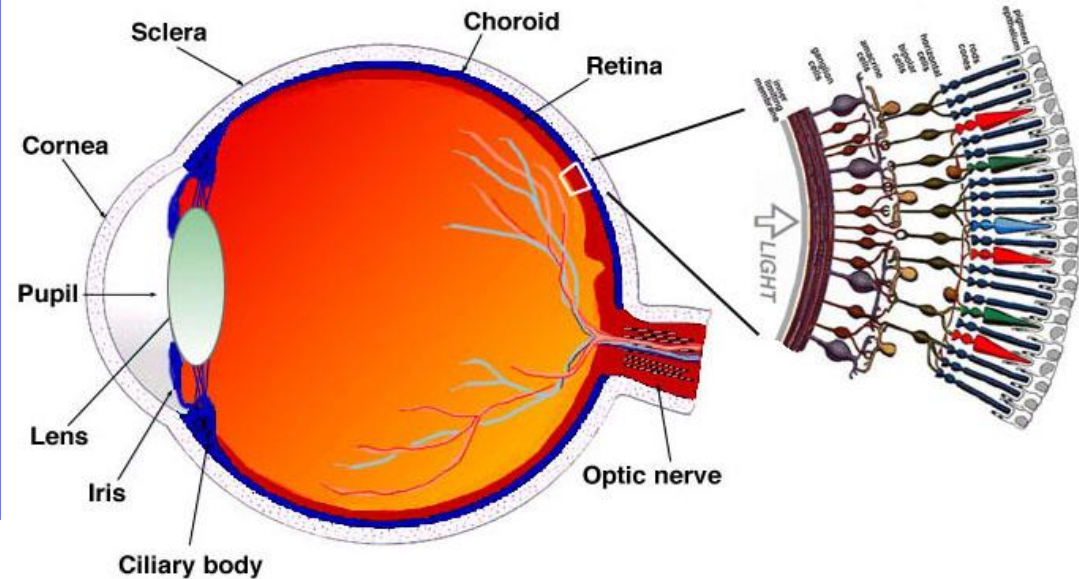
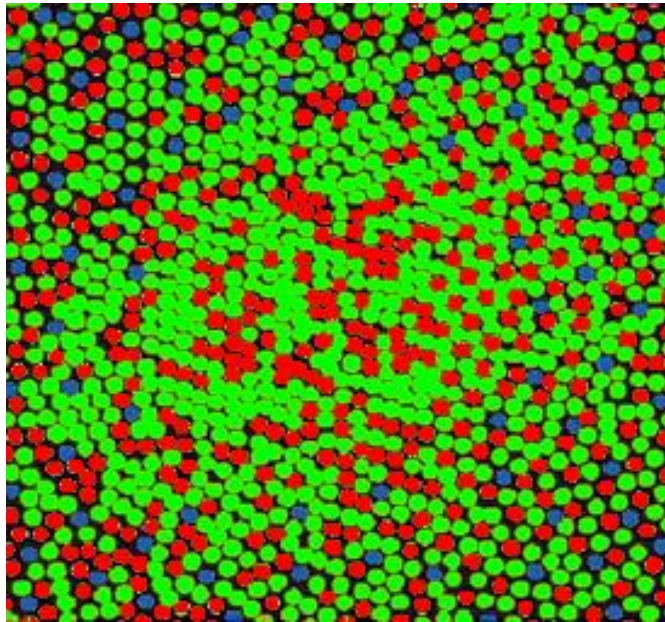


Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

2.1 Structure of the Human Eye (5)

- Eye's Light Sensors

cone density near fovea



#(blue) << #(red) < #(green)

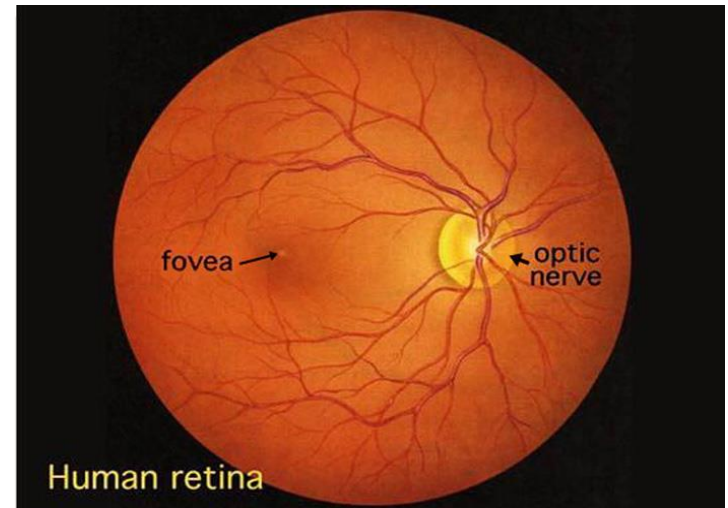
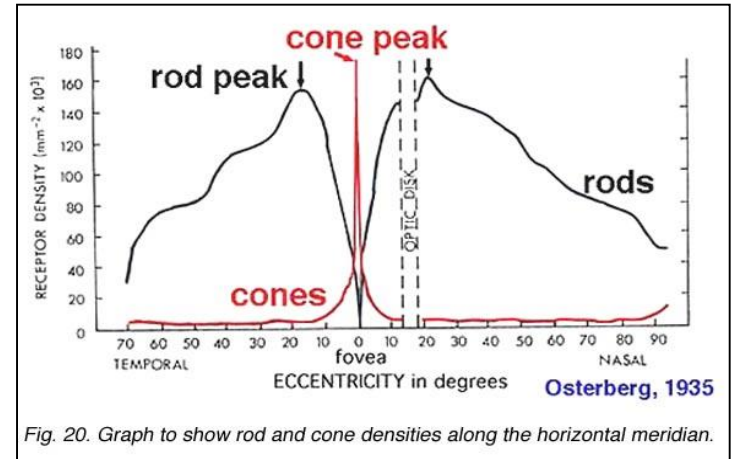
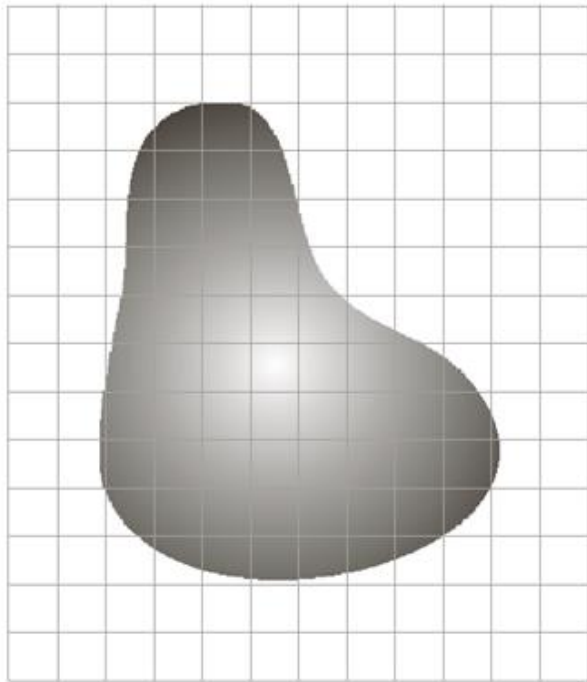


Fig. 1. Human retina as seen through an ophthalmoscope.

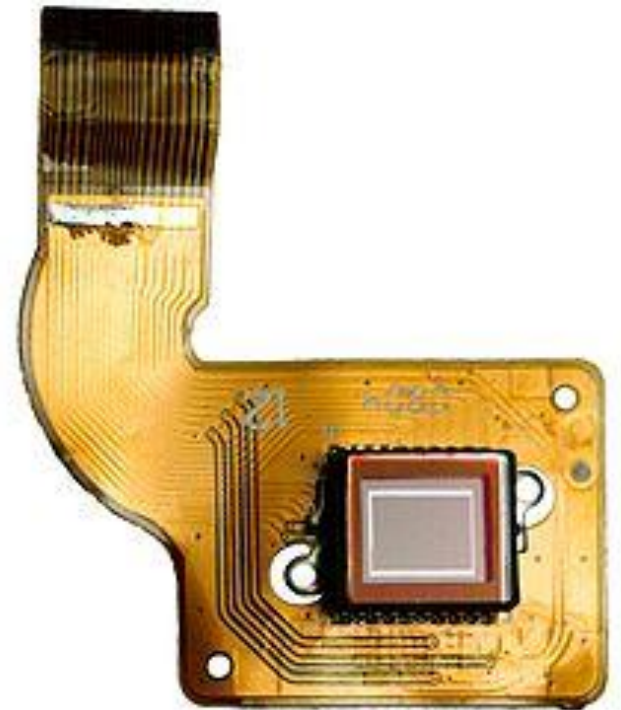
2.1 화소(Pixel)이란?

■ 화소(pixel = picture element)

- The *pixel* (a word invented from "picture element") is the basic unit of programmable color on a computer display or in a computer image.
- The address of a pixel corresponds to its physical coordinates on imaging sensor.



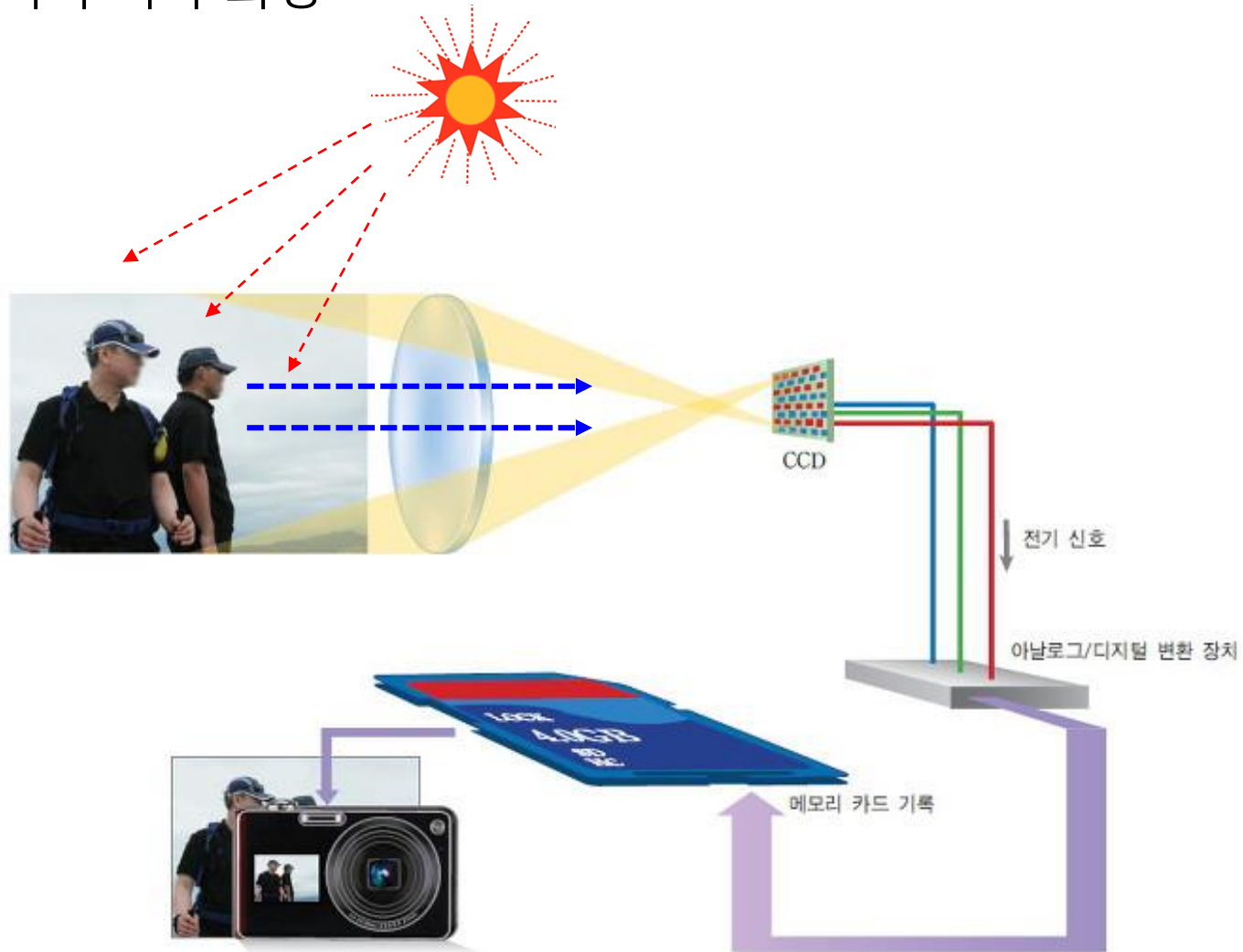
센서에 투영된 영상



CCD 센서

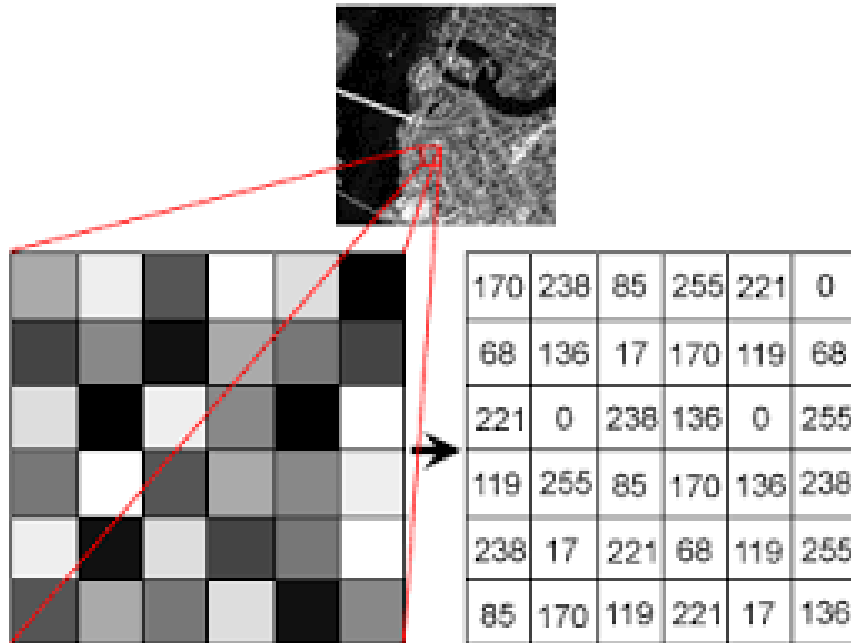
2.1 화소(Pixel)이란?

■ 영상 데이터 획득 과정



2.1 화소(Pixel)이란?

- 화소(pixel = picture element)의 표시 방법
 - Usually, 8-bits resolution is used
 - Pixel value : [0 ~ 255] (integer or BYTE type value)

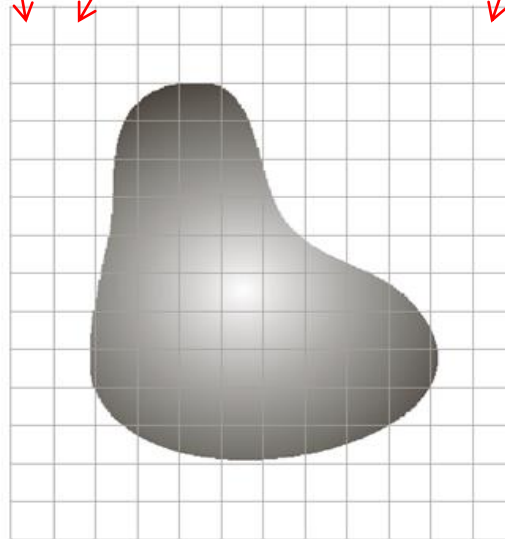


2.1 화소(Pixel)이란?

■ 디지털 영상의 표현 방법: 샘플링과 양자화

- Matrix representation is very natural.
- The representation of an $M \times N$ numerical array as

$$\hat{f}_{\text{gray}}(i, j) \text{ or } f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix}$$



영상처리 (Image Processing) 기술 소개(1)

■ 영상처리(image processing)

- 주어진 목적을 달성하기 위해 원래 디지털 영상을 새로운 디지털 영상으로 변환
- 컴퓨터비전의 **전처리 과정**



(a) 원래 영상



(b) 어둡게



(c) 블러링



(d) 회전

2.2 영상처리 (Image Processing) 기술 소개(2)

영상처리 기술 분야

컬러 공간(모델)(Color Space(model))

화소 점처리(pixel point processing) 기술

히스토그램 처리(Histogram processing) 기술

영역 처리(region-based processing) 기술

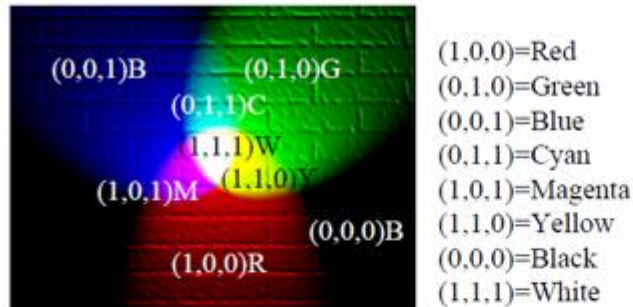
기하 연산(Geometric processing) 기술

다해상도(Multi-resolution) 기술

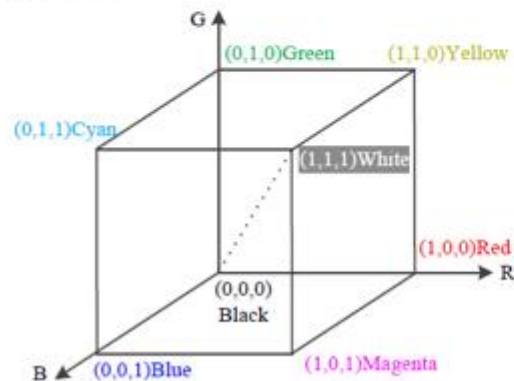
2.2 영상처리 (Image Processing) 기술 소개(3)

■ 컬러 공간/모델(Color Space/Model)

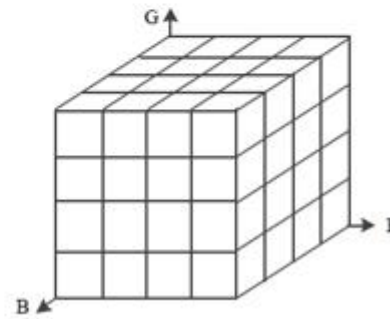
- 기본적으로 3개의 컬러 성분을 정의하여 컬러를 만들어 냄
- RGB, CMYK, HSI, HSY 등이 정의되어 사용됨



(a) 빛의 혼합



(b) RGB 큐브



(c) 양자화된 4×4×4 RGB 큐브

2.2 영상처리 (Image Processing) 기술 소개(4)

- 화소 점처리(pixel point processing) 기술
 - 개별 화소(pixel)를 단위로 값을 변화시키는 방법

$$f_{out}(j, i) = t(f(j, i))$$

$$= \begin{cases} \min(f(j, i) + a, L - 1), & \text{(밝게)} \\ \max(f(j, i) - a, 0), & \text{(어둡게)} \\ (L - 1) - f(j, i), & \text{(반전)} \end{cases}$$



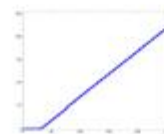
(a) 원래 영상



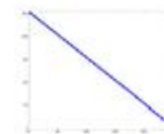
(b) 밝게(a=32)



(c) 어둡게(a=32)



(d) 반전



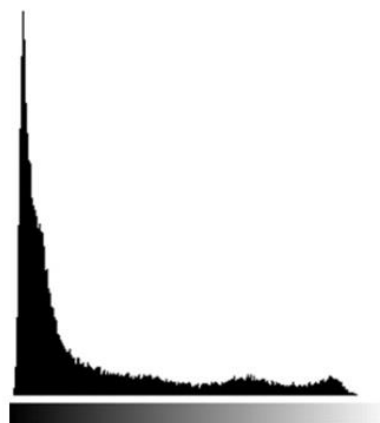
2.2 영상처리 (Image Processing) 기술 소개(4)

■ 히스토그램 처리(Histogram processing) 기술

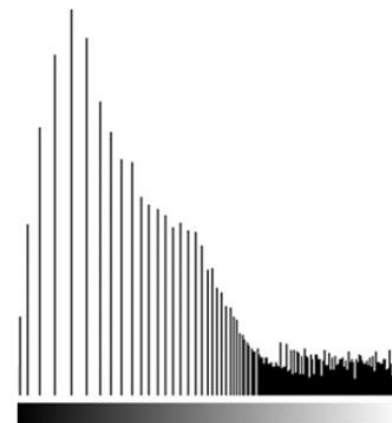
- 히스토그램 = 영상의 밝기의 확률적 분포
- 영상의 밝기 특성을 확인 가능

$$h(l) = |\{(j, i) | f(j, i) = l\}|$$

$$\hat{h}(l) = \frac{h(l)}{M \times N}$$



(a) 원본 영상

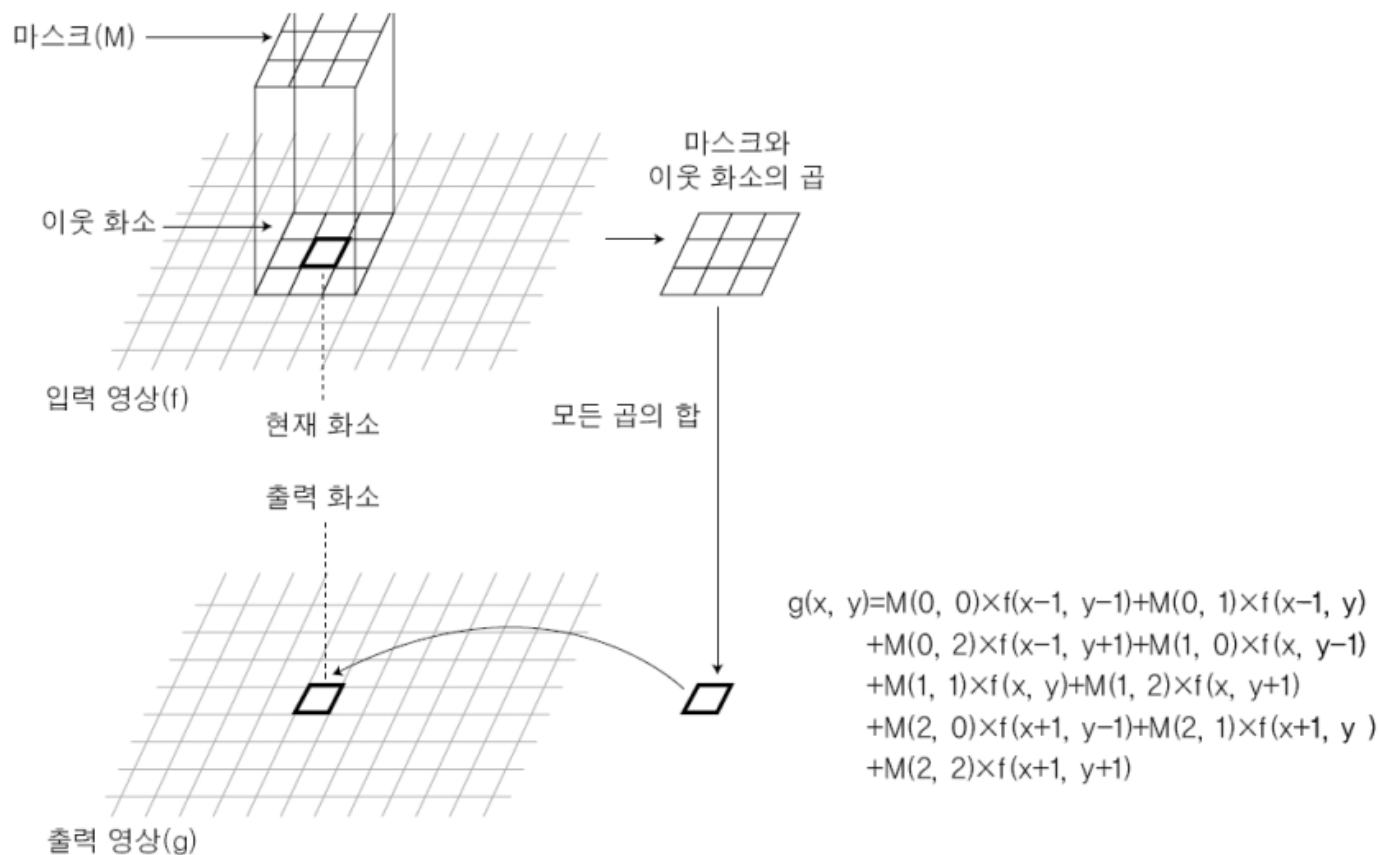


(b) 평활화 영상

2.2 영상처리 (Image Processing) 기술 소개(5)

■ 영역 처리(region-based processing) 기술

- 현재 처리하고자 하는 **화소 값과 인접한 주위 화소 값을 함께 고려하여 처리**
- 마스크(mask) 정의가 필요함 → Convolution processing가 이론적 배경임



2.2 영상처리 (Image Processing) 기술 소개(6)

■ 기하 연산(Geometric processing) 기술

- 화소 값의 변화가 아닌 현재 화소 값의 원하는 위치로 이동시키는 기술
- 회전, 확대/축소, 위치 왜곡 등

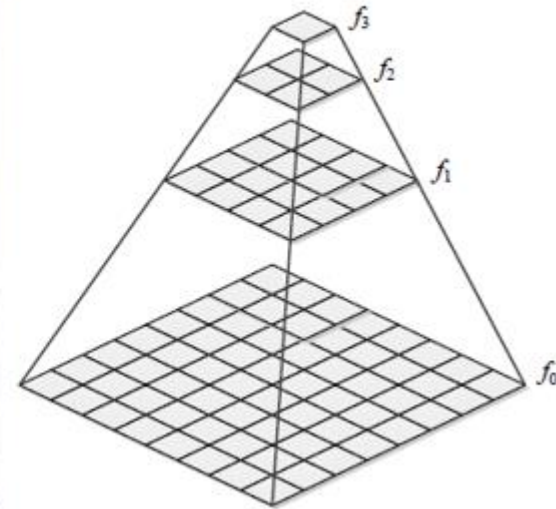
변환	동차 행렬 \hat{H}	설명
이동	$T(t_y, t_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_y & t_x & 1 \end{pmatrix}$	y방향으로 t_y , x방향으로 t_x 만큼 이동
회전	$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	원점을 중심으로 시계방향으로 θ 만큼 회전
크기	$S(s_y, s_x) = \begin{pmatrix} s_y & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$	y방향으로 s_y , x방향으로 s_x 만큼 확대
기울임	$Sh_y(h_y) = \begin{pmatrix} 1 & 0 & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, Sh_x(h_x) = \begin{pmatrix} 1 & h_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Sh_y : y방향으로 h_y 만큼 기울임 Sh_x : x방향으로 h_x 만큼 기울임



2.2 영상처리 (Image Processing) 기술 소개(7)

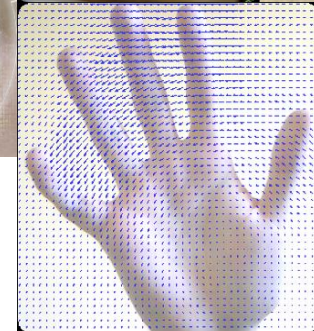
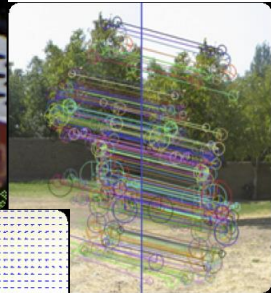
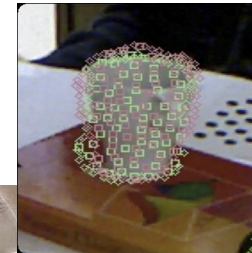
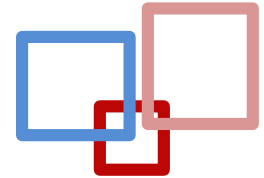
■ 다해상도(Multi-resolution) 기술

- 화면 해상도를 줄이거나 늘리는 연산을 영상 해석을 시도하는 방식
- 업샘플링과 다운샘플링: Pyramidal approach



COMPUTER VISION 비전 프로그래밍

화소 점처리(pixel point processing)의 이해



2.3 화소 점 처리(pixel point processing): 개념

■ 화소 점 처리

- 원 화소의 값이나 위치를 바탕으로 단일 화소 값을 변경하는 기술
- 다른 화소의 영향을 받지 않고 단순히 화소 점의 값만 변경하므로 포인트 처리(Point Processing)라고도 함.
- 산술연산, 논리연산, 반전, 광도 보정, 히스토그램 평활화, 명암 대비 스트레칭 등의 기법이 있음.
- 디지털 영상의 산술연산은 디지털 영상의 각 화소 값에서 임의의 상수 값으로 덧셈, 뺄셈, 곱셈, 나눗셈을 수행하는 것
- 그레이 레벨 영상에서 화소 값이 작으면 영상이 어둡고, 화소의 값이 크면 밝음.

2.3 화소 점 처리(pixel point processing): 산술연산/논리연산

■ 화소의 밝기 값

- 밝기의 단계 수는 화소를 표현하는 양자화 비트 수가 결정
- 그레이 레벨 영상에서는 색은 없고 밝기만 있음.
- 보통, 화소는 밝기를 나타내는데, 주로 양자화 비트 수를 8비트로 표현

■ 명암 대비

- 대비(Contrast): 영상 내에 있는 가장 밝은 값과 가장 어두운 값의 차이로, 영상의 품질을 결정하는 중요한 요소임.
- 높은 대비를 보이는 디지털 영상: 어두운 명도와 밝은 명도의 차이가 너무 커서 시각적으로 좀더 명확하게 보임.
- 낮은 대비를 보이는 디지털 영상: 밝기의 차이가 크지 않아 시각적으로 명확하지 못함.

2.3 화소 점 처리(pixel point processing): 산술연산/논리연산

■ 산술연산

- 덧셈
- 뺄셈
- 나눗셈
- 곱셈

■ 논리연산 (bit operation)

- OR
- AND
- NAND
- XOR

2.3 화소 점 처리(pixel point processing): 산술연산/논리연산

- For example,



입력 영상 상수 더하기(원영상+60) 상수곱하기(원영상*1.4) 상수 빼기 (원영상-60)

출력 = 원본 영상값 (230) + 60 ??? 출력값은 290으로 8bit로 표현할 수 있는 한계인 255보다 큼. 표시에 문제가 발생..!!!

2.3 화소 점 처리(pixel point processing): 산술연산의 문제점과 해결 방법

■ 문제점

- 결과 값이 화소의 최대값과 최소값을 넘을 수 있음.

■ 해결 방법

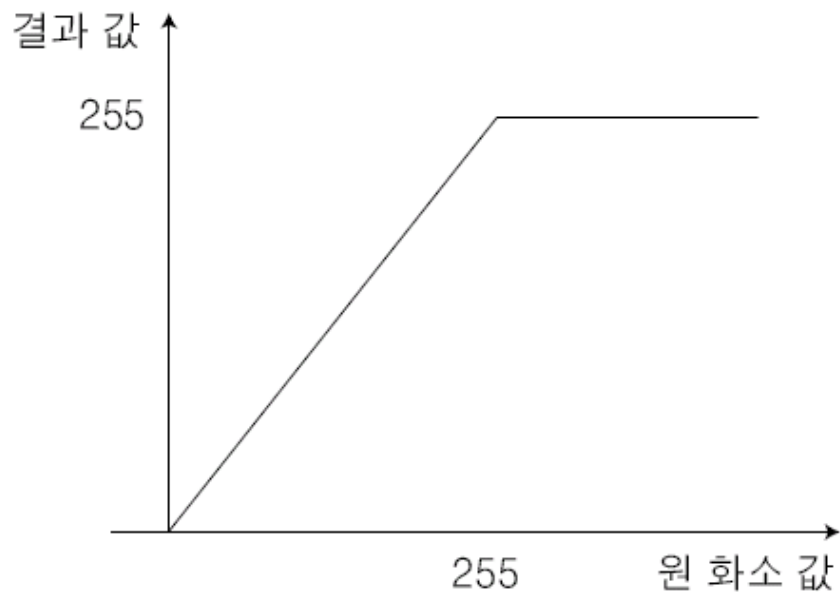
▪ 클래핑(Clamping) 기법

- 연산의 결과 값이 최소값보다 작으면 그 결과 값을 최소값으로, 최대값보다 크면 결과 값을 최대값으로 하는 기법
- 8비트 그레이 영상의 최소값은 0, 최대값은 255
- 음수는 0으로 설정하고, 255보다 큰 값은 255로 설정함.

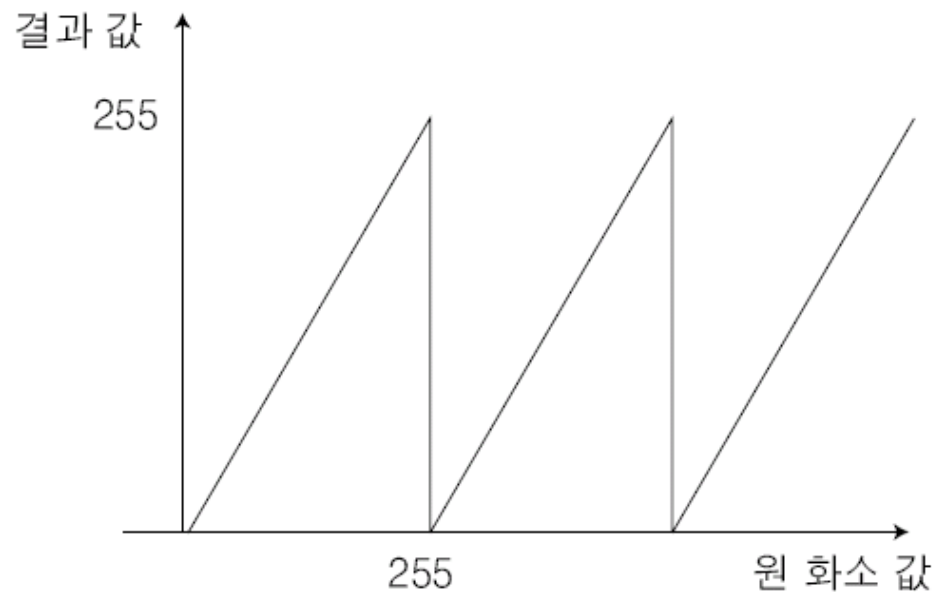
▪ 랩핑(Wrapping) 기법

- 연산의 결과 값이 최소값보다 작으면 그 결과 값을 최소값으로, 최대값보다 크면 최소값부터 최대값까지를 한 주기로 해서 이를 반복하는 기법
- 최대값+1은 최소값이 되고, 연산의 결과 값이 최대값+상수 값일 때는 계속 상수 값-1로 설정함.
- 8비트 그레이 영상의 최소값은 당연히 0이고, 최대값은 255
- 음수는 0으로, 255보다 큰 결과 값 256은 0으로, 257은 1로 설정한 후 이런 방식으로 주기를 계속 반복

2.3 화소 점 처리(pixel point processing): 산술연산의 문제점과 해결 방법(계속)



(a) 클래핑 기법



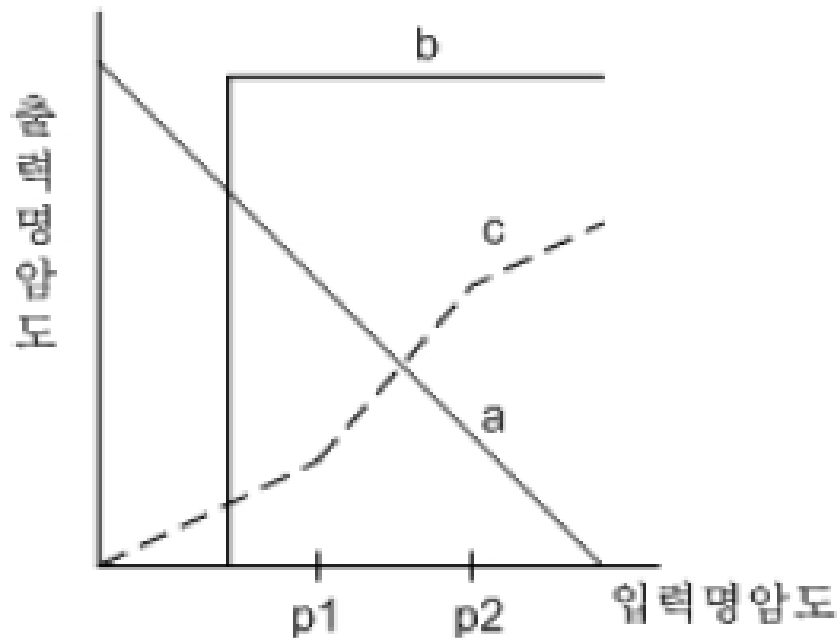
(b) 랩핑 기법

[그림 4-9] 산술연산의 문제점 해결 기법

2.3 화소 점 처리(pixel point processing): 일반적인 화소 점 처리

■ 화소 점 처리를 일반화된 수식으로 표현

- $Q = T(p)$, Q =출력 화소값, p =입력 화소값
- 선형(linear) 혹은 비선형(non-linear) 변환



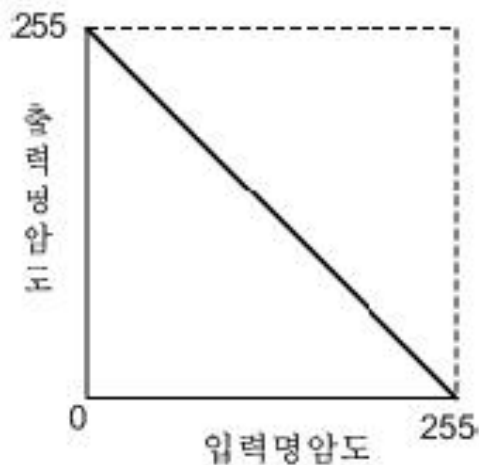
[다양한 변환 관계식]

2.3 화소 점 처리(pixel point processing): 영상 반전(Image Negative)

- 사진의 음화와 같은 영상 구함



(a) 원영상



(b) 반전 변환 함수



(c) 반전된 영상

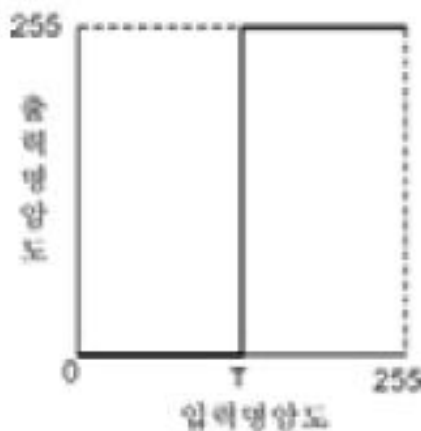
[그림 4-10. 영상 반전]

2.3 화소 점 처리(pixel point processing): 영상 이진화(Binarization)

- 명암 대비가 매우 낮은 경우에 배경과 물체를 확실하게 구분



(a) 원영상



(b) 이진화 변환 함수



(c) 이진화된 영상

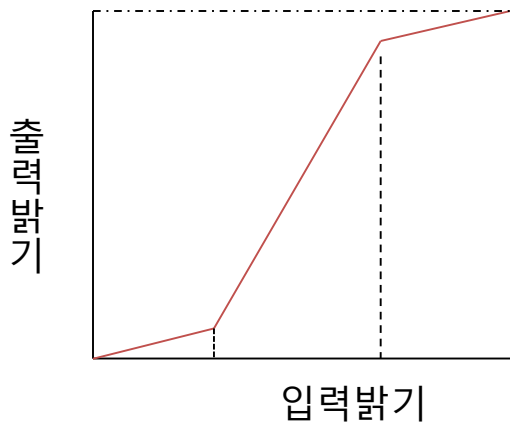
[그림 4-11. 영상 이진화]

2.3 화소 점 처리(pixel point processing): 영상 명암 대비 변환(contrast enhancement)

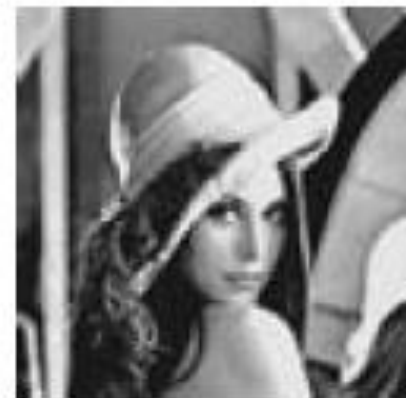
- 특정 구역의 명암도의 생동 폭을 증가 또는 감소 시킴



(a) 원영상



(b) 명암대비 변환 함수

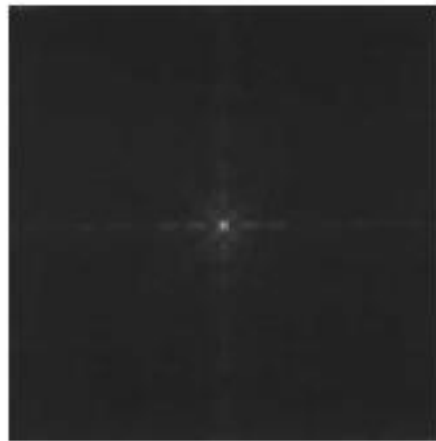


(c) 대비된 영상

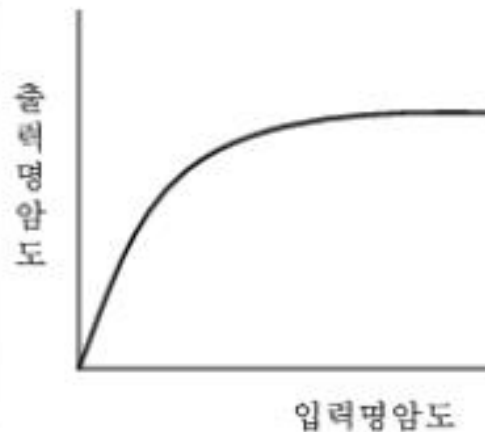
[그림 4-12. 영상 명암대비 향상 결과]

2.3 화소 점 처리(pixel point processing): 로그 변환

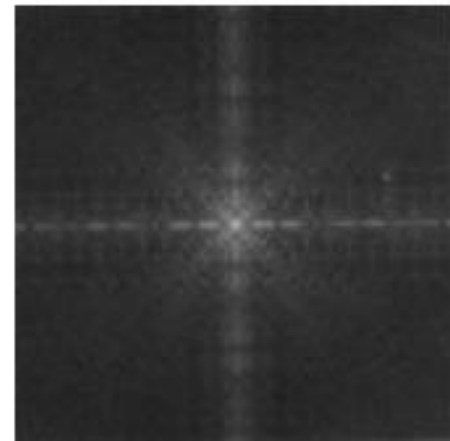
- 명암도의 최대값을 초과하는 값들이 발생하는 경우 화소값의 동적 범위를 명암도의 허용범위 내로 압축
- 관계식: $q = c \log(1 + |p|)$



(a) 원영상



(b) 로그 변환 함수

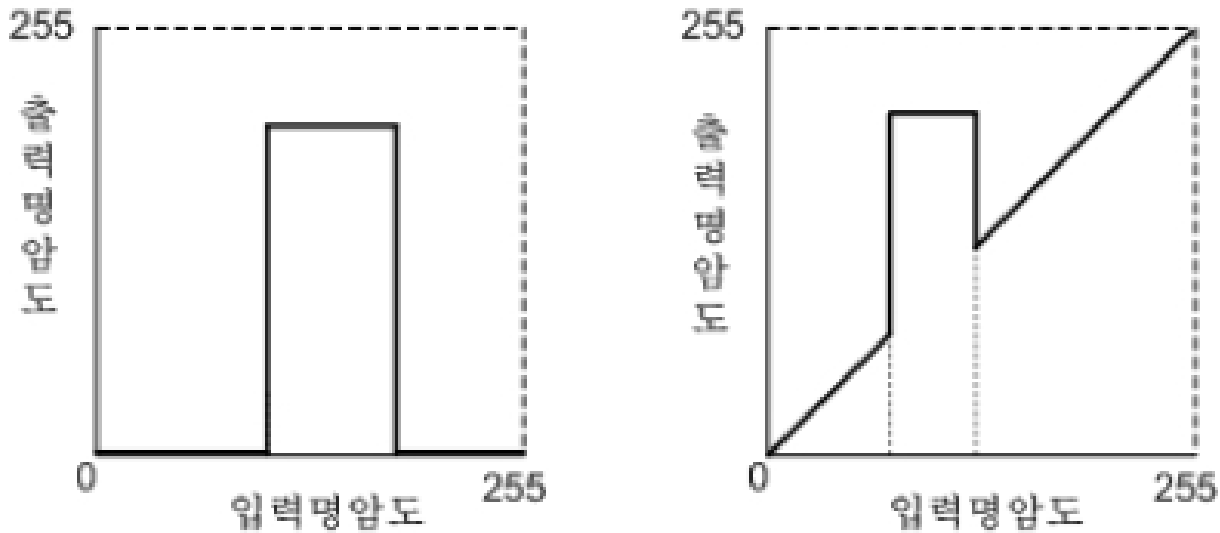


(c) 로그 변환 적용된 영상

[그림 4-13. 로그 변환 결과]

2.3 화소 점 처리(pixel point processing): 명암도 분할 (Gray level slicing)

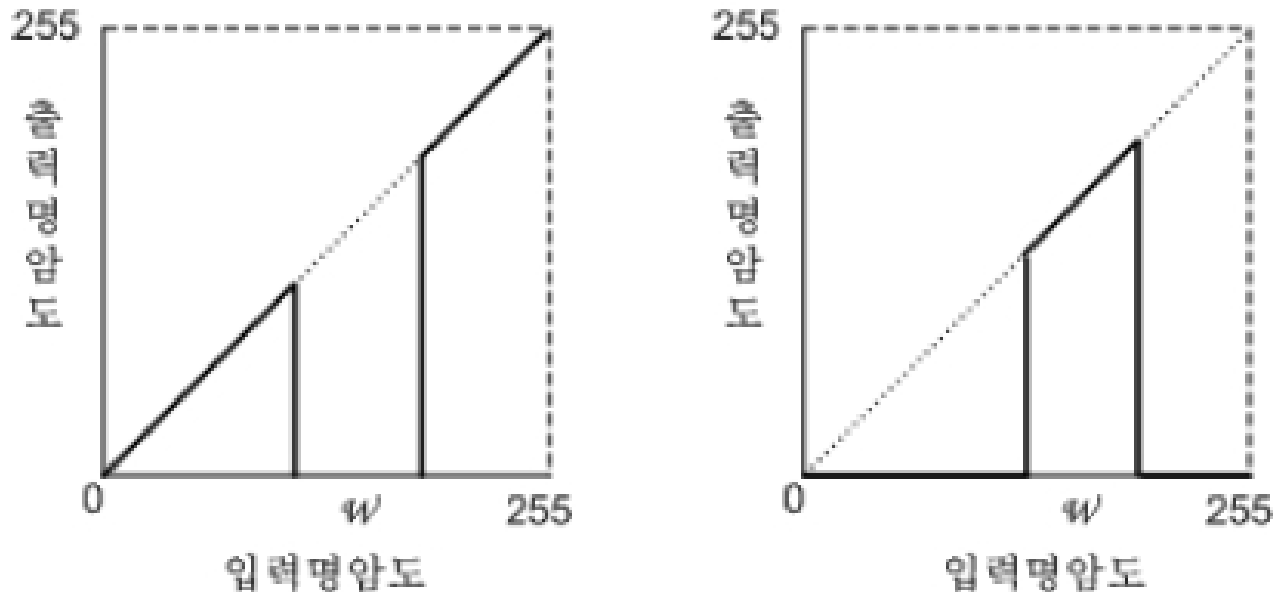
- 영상의 특정 범위의 밝기를 강조



[그림 4-14. 명암도 분할 변환 함수]

2.3 화소 점 처리(pixel point processing): 화소 대역 제거 또는 통과 변환

- 중간 제거는 입력 영상의 명암도 W 부분은 0으로 제거하고 나머지 부분은 보존
- 중간 통과는 입력 영상의 명암도 W 부분은 보존하고 나머지 부분은 0으로 제거



[그림 4-15. 화소 대역 제거 및 통과 변환 함수]

2.3 화소 점 처리(pixel point processing): 감마 보정(Gamma Correction)

- 입력 값을 조정하여 출력을 제대로 만드는 과정
- 감마 보정 함수

$$\text{Output}(q) = [\text{Input}(p)]^{(1/\gamma)}$$

- 함수의 감마 값(γ)에 따라 영상을 밝게 하거나 흐리게 조절할 수 있음
- 감마 값이 1보다 크면 영상이 어두워지고, 1보다 작으면 영상이 밝아짐

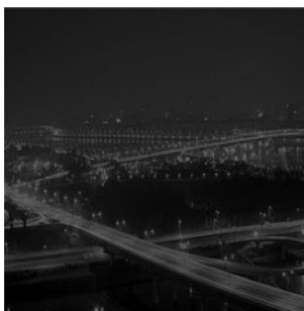
(a) 입력 영상



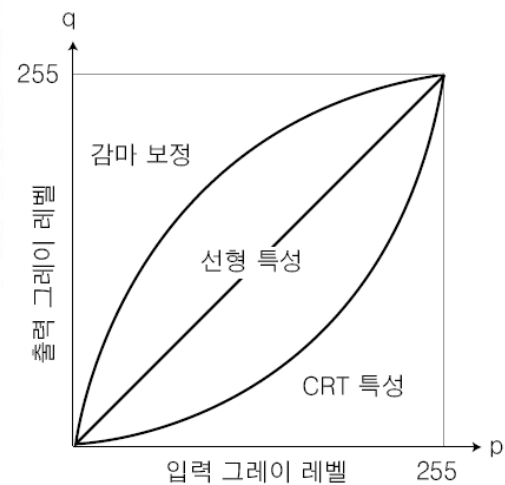
(b) 감마 값(γ) 0.8로 보정한 영상



(c) 감마 값(γ) 1.2로 보정한 영상



(d) 감마 보정 변환 함수 그래프



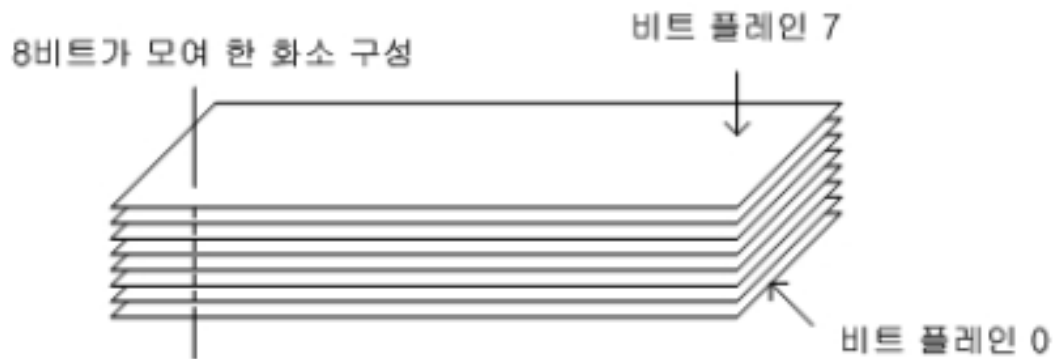
[그림 4-16] 감마 보정 영상과 감마 함수의 그래프

2.3 화소 점 처리(pixel point processing): Bit plane slicing

■ 비트 평면(plane) 분할 기법(Bit Plane Slicing)(1)

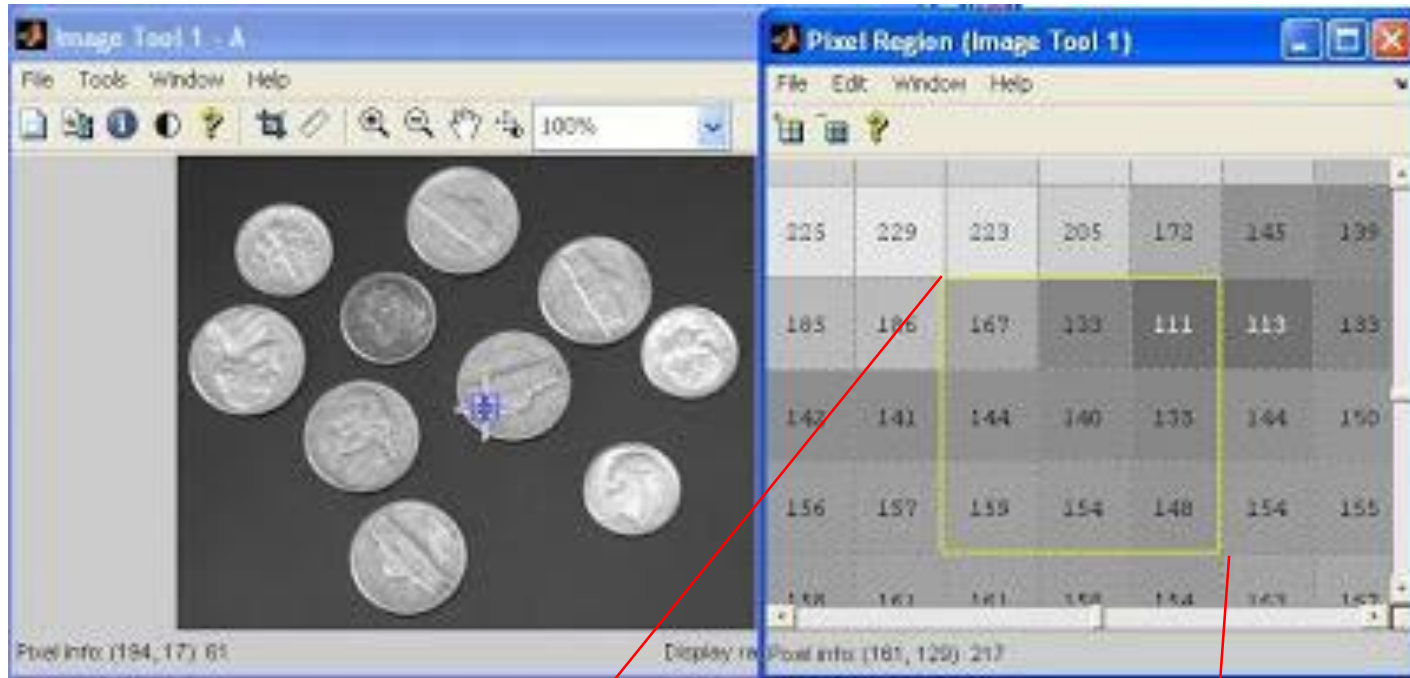
- 영상에서 각 비트의 값들이 어떤 영향을 미치는지 알아보기 위해 각각의 비트들로만 이루어진 영상을 만드는 방법
- m 비트의 경우

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_12^1 + a_02^0$$



[그림 4-20. 영상의 비트 plane 분할 개념]

2.3 화소 점 처리(pixel point processing): Bit plane slicing



10100111	10000101	01101111
10010000	10001100	10000111
10011111	10011010	10010100

2.3 화소 점 처리(pixel point processing): Bit plane slicing

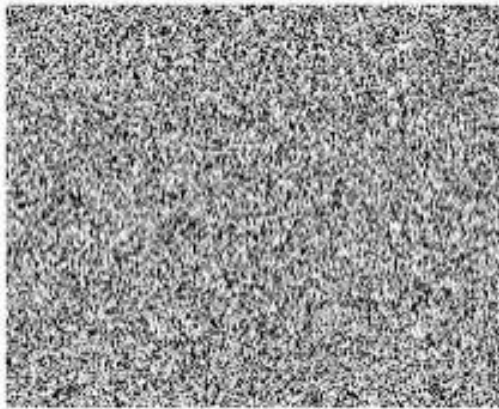
1-th bit

1010011 ¹	1000010 ¹	0110111 ¹
1001000 ⁰	1000110 ⁰	1000011 ¹
1001111 ¹	1001101 ⁰	1001010 ⁰

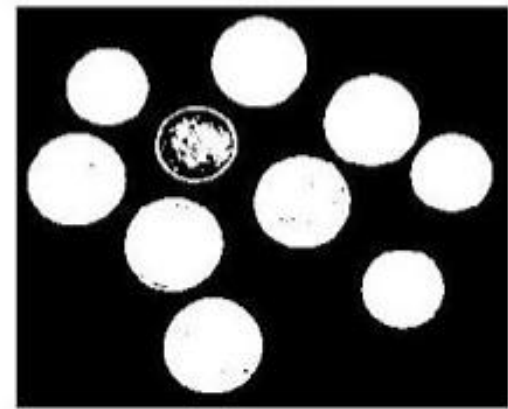
8-th bit

¹ 0100111	¹ 0000101	⁰ 1101111
¹ 0010000	¹ 0001100	¹ 0000111
¹ 0011111	¹ 0011010	¹ 0010100

Bit plane 1

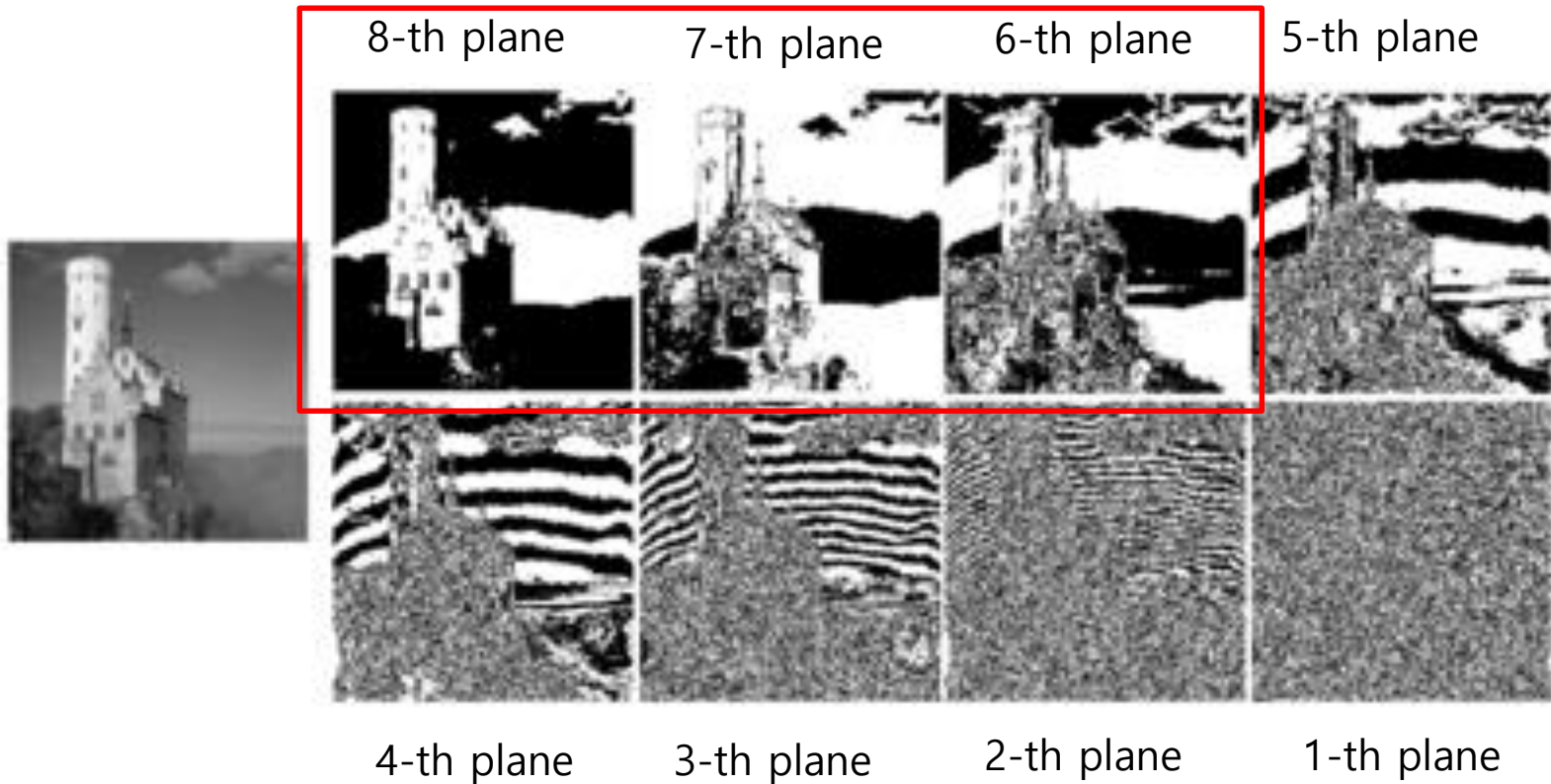


Bit plane 8

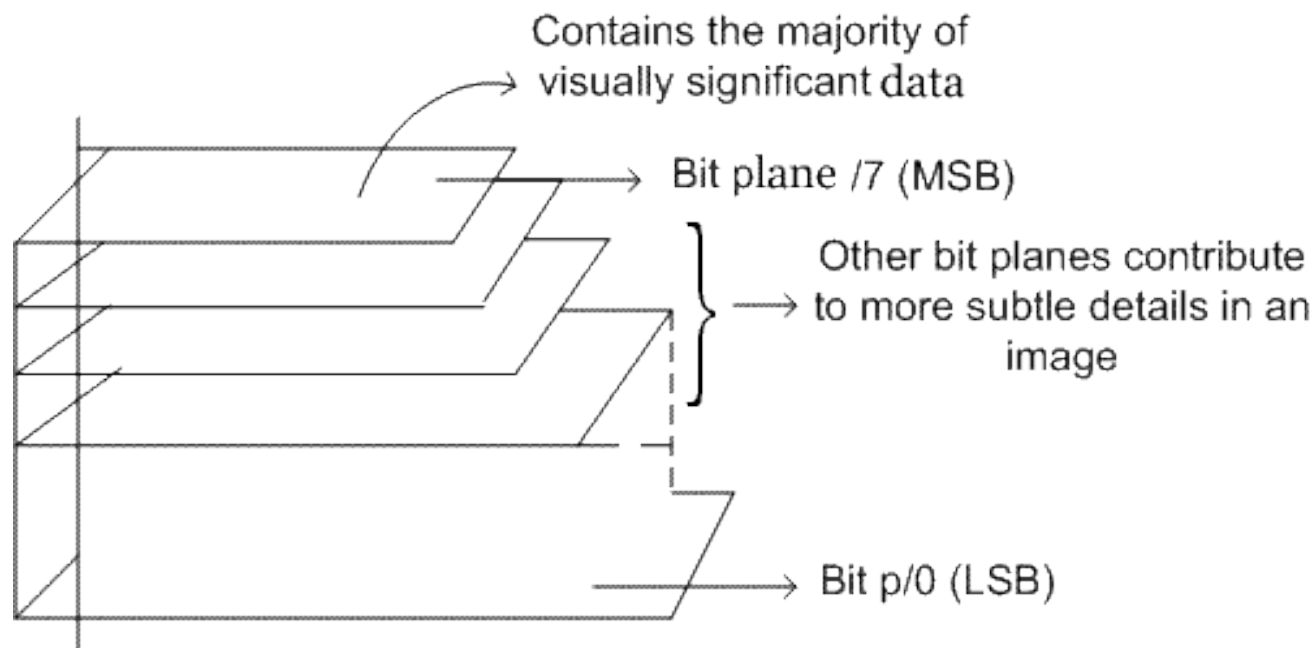


2.3 화소 점 처리(pixel point processing): Bit plane

- Which plane contains more information?

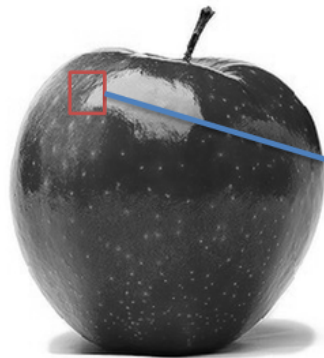


2.3 화소 점 처리(pixel point processing): Bit plane slicing



2.4 컬러영상(color image) 화소 데이터 구조

■ 흑백 영상(Gray scale image)



Gray-scale Image

	j																
i	206	225	193	185	182	174	151	137	132	161	140	119	132	120	141	121	125
	230	194	191	197	189	160	164	143	156	157	99	160	127	113	121	122	107
	195	188	213	185	148	178	153	160	148	116	123	123	155	142	107	151	117
	183	191	190	170	177	167	148	164	145	134	127	158	140	112	128	97	146
	194	177	189	184	171	139	179	149	108	142	146	140	122	114	115	109	164
	177	171	170	175	161	165	172	121	149	153	127	116	131	148	133	133	140
	180	177	157	156	168	171	137	145	153	134	138	149	133	128	137	123	119
	185	160	171	149	157	142	132	147	124	129	118	138	132	118	165	138	104
	165	159	145	176	159	125	159	137	131	142	152	152	116	135	147	106	122
	153	180	186	168	139	160	151	158	114	155	172	83	125	154	107	124	152
	176	191	153	127	166	140	144	149	164	158	71	184	166	81	147	150	132
	177	145	124	151	152	154	140	179	156	92	161	201	108	101	165	128	131
	139	131	152	146	140	158	173	159	92	170	171	89	123	161	124	136	99
	145	136	169	150	141	134	175	106	158	155	142	121	144	137	102	112	107
	141	157	158	121	139	169	137	135	165	124	145	129	105	104	118	112	118
	158	149	122	135	153	140	107	156	121	152	156	118	124	129	118	104	94
165	142	145	132	156	117	135	146	127	138	107	95	116	120	102	94	93	
130	168	151	132	132	134	125	139	116	132	126	111	129	106	99	102	123	
171	173	149	136	133	111	130	121	120	102	104	127	120	111	106	102	118	
185	171	150	109	133	125	120	114	105	121	109	111	111	103	115	100	96	
181	138	124	129	102	123	107	138	119	101	108	109	114	95	102	109	125	
155	137	131	109	114	105	128	119	104	102	103	121	104	129	103	124	110	
140	120	139	128	103	116	110	122	110	106	103	112	110	108	124	120	104	
119	111	136	112	125	125	122	115	90	119	105	98	132	101	126	91	122	
125	127	132	91	134	121	82	117	109	96	97	112	130	109	113	126	129	

```
for(i=0;i<500;i++){
    for(j=0;j<500;j++){
        value= image.at<uchar>(j,i);
    }
}
```


2.4 컬러영상 (color image) 화소 데이터 구조

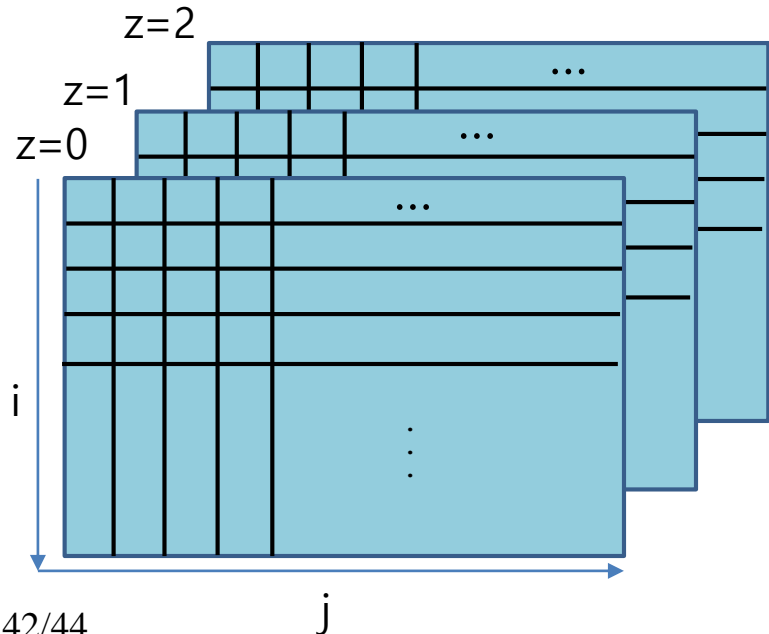
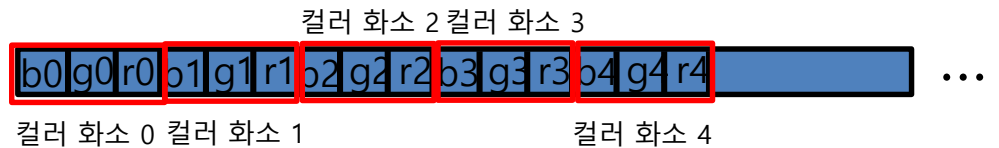
■ 컬러 영상(Color image)

- 1 컬러 화소 = 3개의 컬러 화소 성분(red, green, blue)으로 구성됨



2.4 컬러영상 (color image) 화소 데이터 구조

- 실제 메모리 상에서의 데이터 구조
 - 1차원 구조로 메모리에 저장됨.
 - OpenCV 2.x 이후로는 2차원 Matrix 구조의 버퍼를 사용하므로 2차원 좌표로 접근 가능



2.4 컬러영상 (color image) 화소 데이터처리 구조(in OpenCV)

■ 컬러 영상(Color image)

```
for(i=0;i<500;i++){
    for(j=0;j<500;j++){
        for(k=0;k<3;k++){
            Vec3b &intensity = image.at<Vec3b>(j,i);
            intensity.val[k]=0;
            //intensity.val[0], intensity.val[1], intensity.val[2]에 (b,g,r)값 저장
        }
    }
}
```

← 영상 높이만큼
← 영상 폭 만큼
← 화소 당 컬러의 개수 만큼

```
for(i=0;i<500;i++){
    for(j=0;j<500;j++){
        b= image.at<Vec3b>(j,i)[0]; ← 3개의 성분으로 분해
        g= image.at<Vec3b>(j,i)[1];
        r= image.at<Vec3b>(j,i)[2];
    }
}
```

본 강의 요약

- 영상처리(Image Processing) 기술 분류 소개
- 사람의 시각적 데이터 센싱 구조
- 화소(pixel)의 정의
- 화소 점처리(pixel point processing)에 대한 소개
- 다양한 화소 점처리 (pixel point processing) 기법 소개
 - 산술/논리 연산
 - 반전, 이진화, 향상, 대역통과/제거 등등
 - 색상 대비, 감마 보정 기법, 비트 슬라이싱 기법
 - OpenCV에서 화소 접근법

COMPUTER VISION 비전 프로그래밍

Thank you and question?

