

1. 영상처리 소개

다루는 내용

- 1.1: 영상처리 개요 및 응용
- 1.2: 영상의 기본
- 1.3: 영상의 입출력

1.1 영상처리 개요 및 응용

■ 영상이란?

- 넓은 의미: 시각적으로 정보를 기록하고 표현하는 하나의 방식인 그림이나 사진을 뜻함
- 좁은 의미: 화소(Pixel)라고 하는 작은 사각형들의 집합으로 정의
- 영상은 2차원적으로 변하는 어떤 양이 컴퓨터로 처리할 수 있는 형태로 기록되어 있는 것
- 수학적으로는 행렬을 영상으로 정의



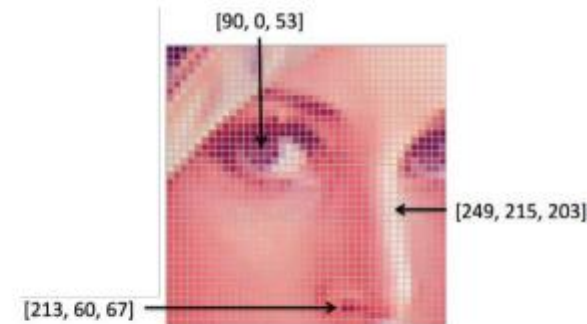
1.1 영상처리 개요 및 응용

■ 디지털 영상

- 디지털 영상은 0과 1로 이루어진 이진수의 세계
 - 영상의 기본 요소는 화소(pixel)
 - 화소는 0부터 255의 값을 가지는 8비트(=1byte) 이진수
 - 명암도(gray) 영상은 1바이트, 컬러영상은 [blue, green, red] 3바이트로 구성됨

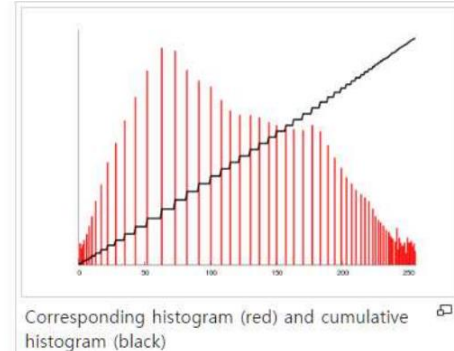
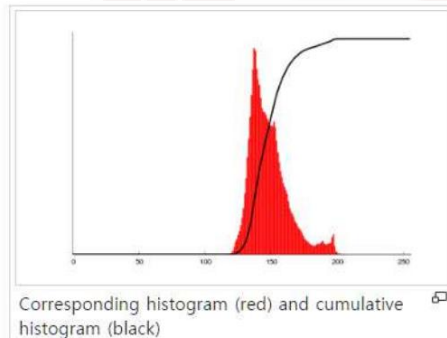


187	187	187	194	197	173	77	25	19	19
190	187	190	191	158	37	15	14	20	20
187	182	180	127	32	16	13	16	14	12
184	186	172	100	20	13	15	18	13	18
186	190	187	127	18	14	15	14	12	10
189	192	192	148	16	15	11	10	10	9
192	195	181	37	13	10	10	10	10	10
189	194	54	14	11	10	10	10	9	8
189	194	19	16	11	11	10	10	9	9
192	88	12	11	11	10	10	10	9	9



■ 영상처리(Image Processing)란?

- 영상처리는 출력 영상을 얻기 위해 입력 영상(digital image)을 다루는 알고리즘을 연구하는 것
- 영상의 정보를 손쉽게 이해할 수 있도록 영상의 특성을 변경하는 것
- 자동화 시스템에서 영상의 인식에 적합하도록 영상을 변형하는 처리의 과정



■ 영상분석(Image Analysis)이란?

- 고수준의 정보를 추출하기 위해 영상을 다루는 알고리즘을 연구하는 것

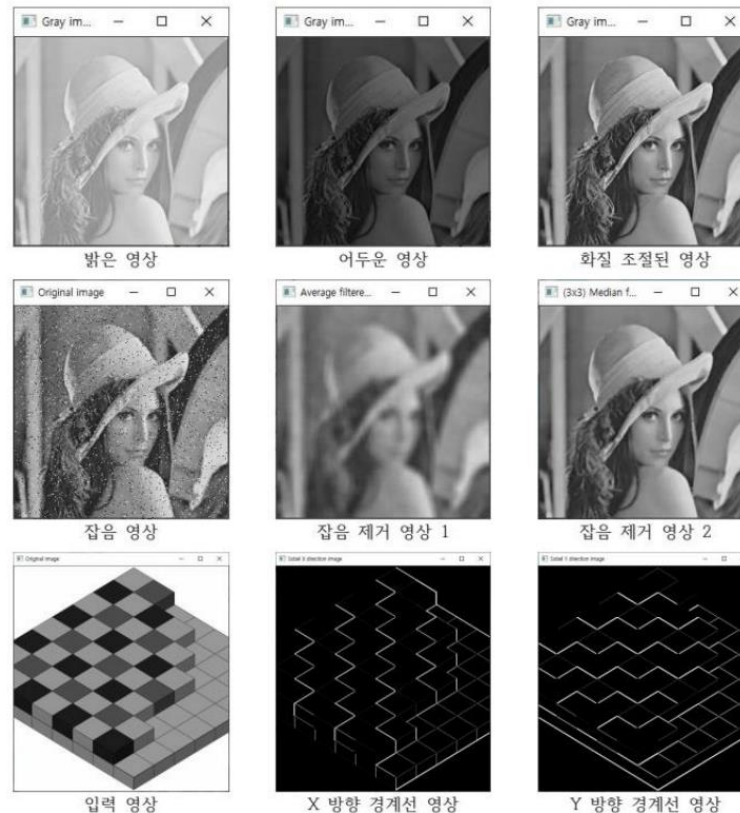
※ 고수준의 정보?

- 영상 속 객체가 무엇인지 인지하거나, 위치를 추정, 영역을 구분하는 것

1.1 영상처리 개요 및 응용

■ 영상처리의 예시

- **영상의 화질 개선:** 시각적인 개선을 위해 영상을 다른 영상으로 변환하는 것
- **기하학적 변환:** 화소의 위치는 입력 영상에서 출력 영상으로 바뀌지만 화소의 값은 변경되지 않는 것
- **영상의 특징 부각:** 영상 속에서 주요 객체들의 경계 또는 영역과 같은 주요 특징을 강조하는 것



■ 화질개선

- **향상(enhancement)**은 시각적인 개선을 위해 영상을 다른 영상으로 변환하는 것을 말함
 - 어두운 영상을 밝게 하는 것이나, 자세함을 높이기 위해 영상의 대비(contrast)를 증가
 - 물체의 경계를 밝게 표시하거나 에지(edge)를 탐지

■ 복원

- **복원(restoration)**은 잡음으로 훼손된 영상을 회복하는 목적
 - 훼손은 센서에 의한 잡음, 신호의 전송 중에 생긴 잡음, 혹은 외부처리에 의한 잡음 등이 원인이 될 수 있음
 - 영상의 일부분의 훼손 또는 전체적인 노이즈를 감소시킬 수 있는 알고리즘을 연구

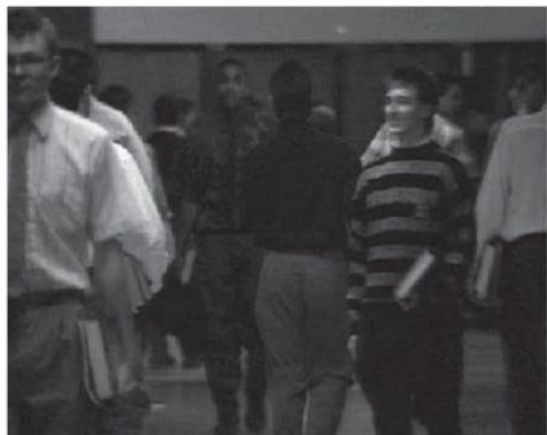
■ 압축

- **압축(compression)**은 가능한 압축에서 영상의 질이 떨어지지 않으면서 원래 신호가 필요한 비트보다 훨씬 적은 비트로 영상을 저장하는 것
- 영상을 표현하고 저장하는데 필요한 많은 데이터를 줄이는 처리

1.1 영상처리 개요 및 응용

- (Enhancement) 어두운 영상 -> 대조(contrast)개선
- (Restoration) 잡음으로 오염된 영상 -> 영상복원 영상
- (Compression) 깨끗한 영상 -> 처리의 효과를 잘 보이기 위해 의도적으로 낮은 품질 변환

입력



출력



Enhancement

Restoration

Compression

■ 화질 개선의 중요성

- 화질 개선은 기본적으로 시각적인 개선을 위해 원래 영상으로부터 개선된 새로운 영상으로 변환하는 작업이 수행
- 사람의 인지력에 도움을 주는 화질 개선은 의료영상의 판독과 같이 사람이 직접 영상에 대한 판단이 필요할 때 큰 도움을 줄 수 있음
- 오늘날 화질개선은 사람의 인지능력에서 나아가 AI의 자동화 판독에 도움이 될 수 있도록 하는 것에 중점적인 연구가 이루어지고 있음 (전처리 기술로 활용)
 - ex) 자율주행 자동차를 위한 안개 상황에서의 디노이징 기술



(a) 안개 낀 도로 영상



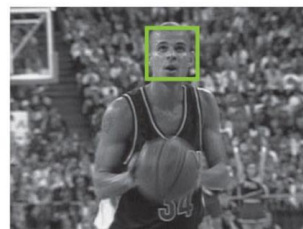
(b) 히스토그램 평활화로 개선한 영상

■ 영상분석의 예시

- 분류(Classification)
 - 영상이 어느 클래스에 해당되는지 지정하는 작업
 - 예) 얼굴인식, 숫자인식, 지문인식, 보행자 인식 등등.
- 탐지(Detection)
 - 주요 객체가 어느 위치에 있는지 찾아내는 작업
 - 분할과 달리 대략적인 영역을 사각형의 박스형태로 찾음
- 분할(Segmentation)
 - 영상에서 같은 객체에 속한 화소들을 구분하는 과정
 - 화소들을 그룹화하는 처리라고 할 수 있음



Segmentation



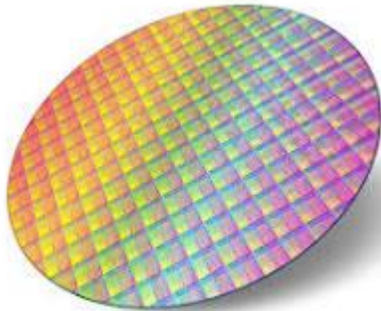
Classification



Shape from X

■ 응용의 예들

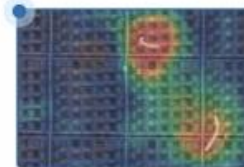
- 산업 검사(industrial inspection): 제품의 결함을 찾기 위해 사용
 - 반도체 산업에서 웨이퍼의 결함 또는 이물질 검출등에 활용



뉴로클 AI 딥러닝 비전검사

Applications

웨이퍼 / 다이 검사



웨이퍼 표면 결함 검출



웨이퍼 표면 결함 검출

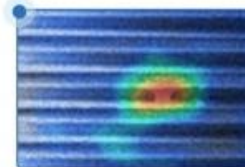


다이 표면 결함 검출

마스크 검사



마스크 기판 이물 검출

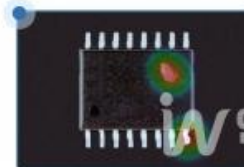


마스크 패턴 불량 검사



마스크 패턴 불량 검사

반도체 부품 검사



IC Chip 외관 검사



MLCC 결함 검사



PCB 커넥터 핀 검사

■ 응용의 예들

- 문자 영상분석(character image analysis): 문서 자동 분석에 활용
 - 편지 분류를 위하여, 종이에 쓰인 문자와 숫자 읽기에 문자 인식기술로 사용
 - 자동차번호판 인식 시스템



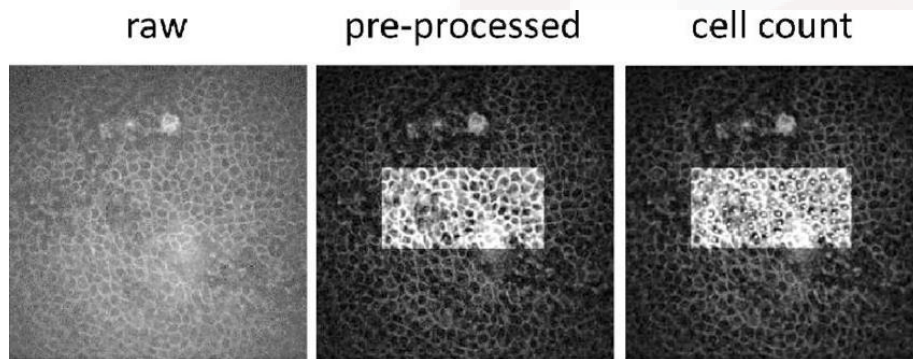
■ 응용의 예들

- 원격 감지(remote sensing)
 - 인공 위성이나 특정 위치에 비행하는 항공기의 센서들로 얻은 여러 스펙트럼 대역의 데이터로부터 데이터의 획득 및 처리
 - 군사 목적을 위한 위성 영상 처리
 - 육상 특징, 식물 양, 광산의 위치, 수질의 온도 측정, 해수면의 변화 등 자연의 변화 탐지



■ 응용의 예들

- 과학적 이미징(scientific imaging)
 - 현미경 영상, 생체영상으로 보이는 세포 추적
 - 빛 또는 전자 현미경을 사용한 세포 분석
 - 천문학 연구를 위한 망원경의 해상도 향상 방법 및 영상 개선

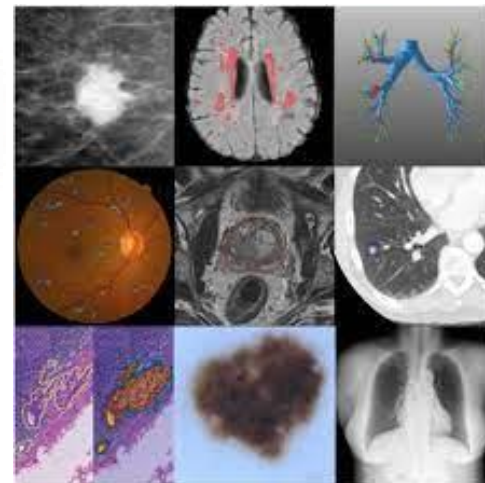
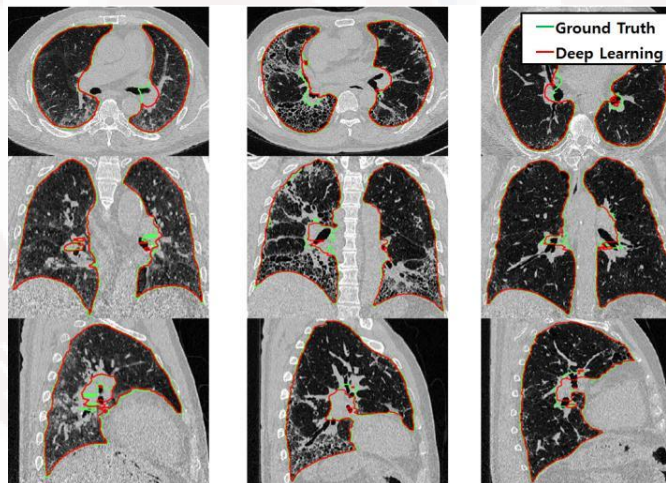


화성 표면

■ 응용의 예들

■ 의료 이미징(medical imaging)

- 종양 감지, 질병 진단, 뼈가 부러졌는지 유무 확인 등등.
- 수술 안내, 동맥의 막힘 감지 등 의학적 영상분석의 자동화에 활용



■ 기타 응용

■ 추적

- 차량 추적, 보행자 추적, 혈류 추적

■ 행동 분석

- 스포츠 동작 인식, 스포츠 자세 코칭, 쓰러짐, 서있는 자세



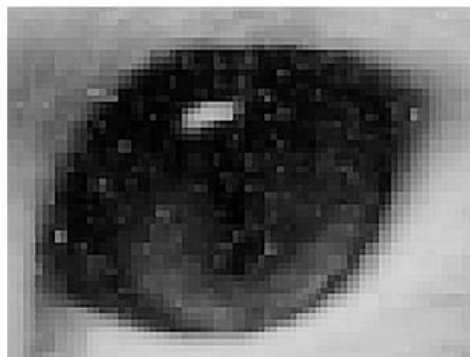
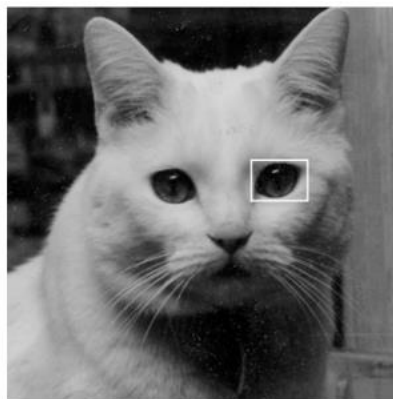
(d) 추적(<https://motchallenge.net/vis/MOT17-09-SDP>)



(e) 행동 분석(<https://github.com/mostafa-saad/deep-activity-rec#dataset>)

■ 화소 (Pixel)

- 디지털 영상을 구성하는 기본적인 영상의 원소
- 영상을 이루는 배열의 각 요소는 **화소(pixel)**라 부른다
- Picture Element, 점, 화소

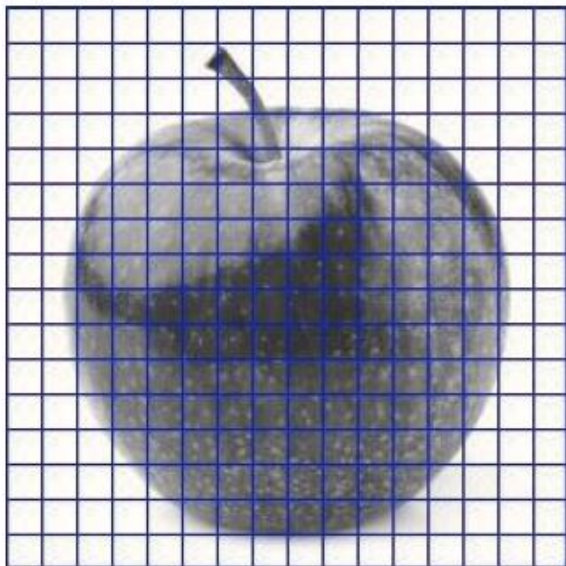


1	33	78	60	55	1	76	99
56	45	90	91	88	3	80	87
77	90	23	1	99	34	90	68
80	65	40	100	8	7	8	49
23	55	71	0	19	200	33	45
21	44	207	65	18	33	12	77
22	33	45	66	78	89	0	77
13	57	89	88	90	200	208	100

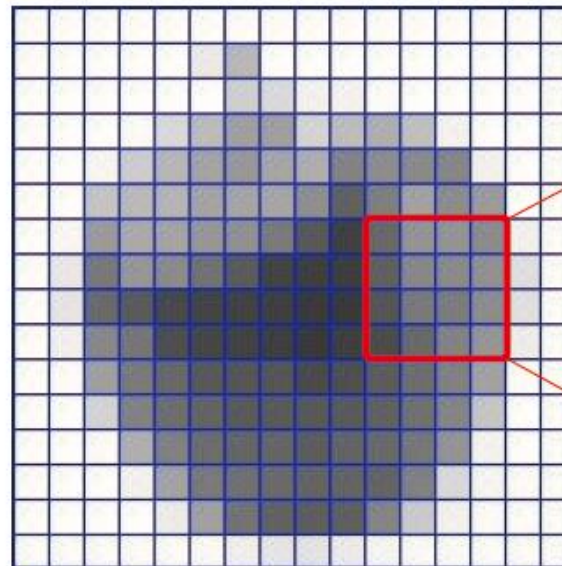


■ 디지털 변환

- 피사체가 반사하는 빛 신호를 CCD센서에서 샘플링과 양자화를 통해 디지털 영상으로 변환
- 디지털 영상으로 변환 구성요소는 화소로 이루어져 있다
- 카메라로 캡처한 영상은 일반적으로 2차원 센서 입력의 디지털화로, **디지털 영상**이라 부르며, 행렬(matrix)과 같이 값의 이산된 2차원 배열이다.
 - 영상의 **너비**는 행렬의 **열** 수와 같다
 - 영상의 **높이**는 행렬의 **행** 수와 같다



(a) 샘플링



105	149	149	141
97	137	139	146
86	123	126	142
76	106	132	150

(b) 양자화

■ 영상의 해상도

- 해상도는 물리적 단위 공간에서 식별 가능한 점(화소)의 개수를 뜻한다.
 - 인치 당 점의 개수를 뜻하는 DPI(dot per inch) 또는 PPI(pixels per inch)는 해상도를 의미
ex) 72 DPI는 1인치(2.54cm)에 1/72 크기의 점이 72개 들어가 있음을 의미
ex) 17.3인치의 노트북 액정화면이 1920x1080 픽셀로 구성되어 있다면, 해상도는 127.34PPI 가 된다



[잘못상식] 왜 디스플레이 크기는 대각선으로 나타낼까?



6.1형

Super Retina XDR 디스플레이

6.1형(대각선) 전면 화면 OLED 디스플레이

2532 x 1170 픽셀 해상도(460ppi)

iPhone 12 Pro 디스플레이는 모서리가 둥근 형태로, 기기의 아름다운 곡면 디자인을 반영합니다. 이 모서리는 기기의 전체적인 모양인 직사각형 내부에 위치합니다. 직사각형 기준으로 측정했을 때, 화면은 대각선 길이 기준 153.97mm(6.06형)입니다 (실제로 보이는 영역은 이보다 좁음).



6.7형

Super Retina XDR 디스플레이

6.7형(대각선) 전면 화면 OLED 디스플레이

2778 x 1284 픽셀 해상도(458ppi)

iPhone 12 Pro Max 디스플레이는 모서리가 둥근 형태로, 기기의 아름다운 곡면 디자인을 반영합니다. 이 모서리는 기기의 전체적인 모양인 직사각형 내부에 위치합니다. 직사각형 기준으로 측정했을 때, 화면은 대각선 길이 기준 169.55mm(6.68형)입니다 (실제로 보이는 영역은 이보다 좁음).

[짤막상식] 왜 디스플레이 크기는 대각선으로 나타낼까?

- 대각선으로 화면 크기를 측정하는 방법은 원형 면이 있는 그림관이 일반적으로 사용되었던 1세대 CRT 텔레비전에 사용된 방법에서 계승...

-위키피디아-

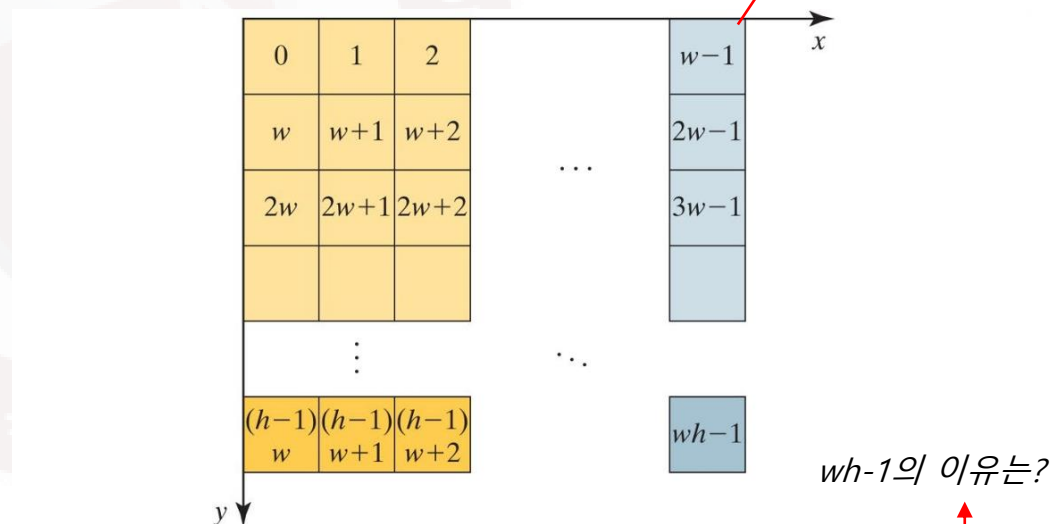
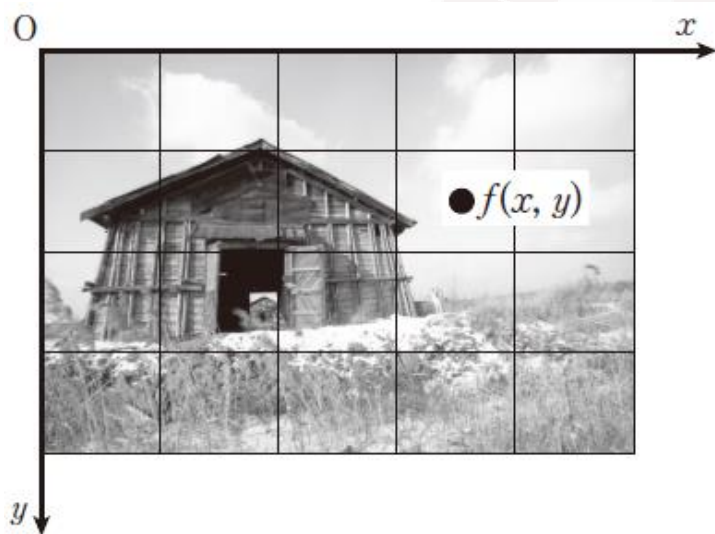


1927년 [미국](#)의 필로 판스워스(Philo Taylor Farnsworth)가 세계 최초로 완전한 전자식 브라운관 텔레비전을 발명

■ 영상 데이터 접근

- 화소 값은 한 쌍의 좌표(x, y)로 접근된다.
 - x 와 y 는 음수가 아닌 정수다.
- 그레이 스케일 영상 f 의 좌표(x, y)에 있는 화소의 값 v 는 아래와 같이 표기할 수 있다.

$$v = f(x, y) \quad (\text{열, 행})$$



2차원 행렬 영상을 1차원 배열로 변환된 결과 →

0	1	2	...	w-1	w	w+1	w+2	...	2w-1	2w	...	wh-1
(0, 0)	(1, 0)	(2, 0)		(w-1, 0)	(0, 1)	(1, 1)	(2, 1)		(w-1, 1)	(0, 2)		(w-1, h-1)

■ 영상 종류

- 회색조(명암) 영상 (Grayscale image)
- 컬러 영상(Color image, RGB image)
- 이진 영상(Binary image)
- 인덱스 영상 (Indexed image)



컬러 영상



회색조 영상



이진 영상

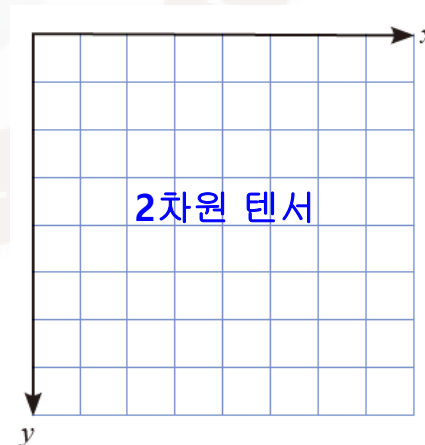


인덱스 영상

1.2 영상의 기본

■ 회색조 영상(gray scale image)

- 그레이 스케일 영상(gray scale image)은 각 화소의 값이 캡처된 빛의 양을 나타내는 스칼라로 나타낸다.
- 1픽셀은 밝기 값을 의미하는 1개 채널로 구성되며 8비트를 할당
- 1개 픽셀은 $0 \sim 2^8 - 1$ 의 총 256개 밝기값
- 의료영상, 군사 영상 등의 경우에는 1픽셀에 더 많은 비트를 할당, 그러나 영상을 화면에 출력할 때에 일반적인 디스플레이 장치의 한계로 컬러 매핑을 통해 출력
- 용량
 - 가로(1920)x세로(1080)x깊이(8비트)
 - Full HD: $1920 \times 1080 \times 8 = 16,588,800 \text{ bits} = 2,073,600 \text{ bytes}$ (약 2MBytes)



■ 컬러 영상

- RGB 컬러 영상은 화소 값이 빨강(R), 녹색(G), 파랑(B)의 세 가지 색상 채널에서 캡처된 빛의 양을 포함하는 형태이다.
 - 컬러영상은 화소당 24비트를 필요로 하거나, 3개의 컬러 채널 각각에 대해 1바이트를 필요로 함 (※ 1개 화소에 $2^{24}=16,777,216$ colors)
 - **인터리브(interleaved)** 형태: 하나의 화소에 대한 3개의 값은 다음에 오는 3개의 값 이전에 저장

$$B_0 G_0 R_0 B_1 G_1 R_1 B_2 G_2 R_2 \cdots B_{n-1} G_{n-1} R_{n-1}$$

- **평면(planar)** 채널 방식: 빨강, 녹색, 파랑 채널이 별도의 화소당 1바이트 영상으로 저장

$$B_0 B_1 B_2 \cdots B_{n-1} G_0 G_1 G_2 \cdots G_{n-1} R_0 R_1 R_2 \cdots R_{n-1}$$

- **알파 값 또는 불투명도(alpha value or opacity)**: 여러 영상을 블렌딩하는 방식

$$B_0 G_0 R_0 A_0 B_1 G_1 R_1 A_1 B_2 G_2 R_2 A_2 \cdots B_{n-1} G_{n-1} R_{n-1} A_{n-1}$$



Color image (RGB channel)



Red channel



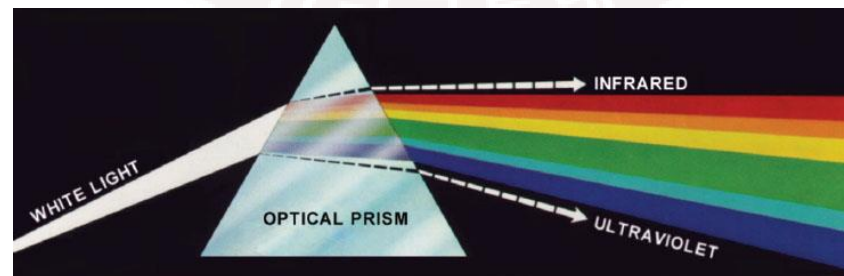
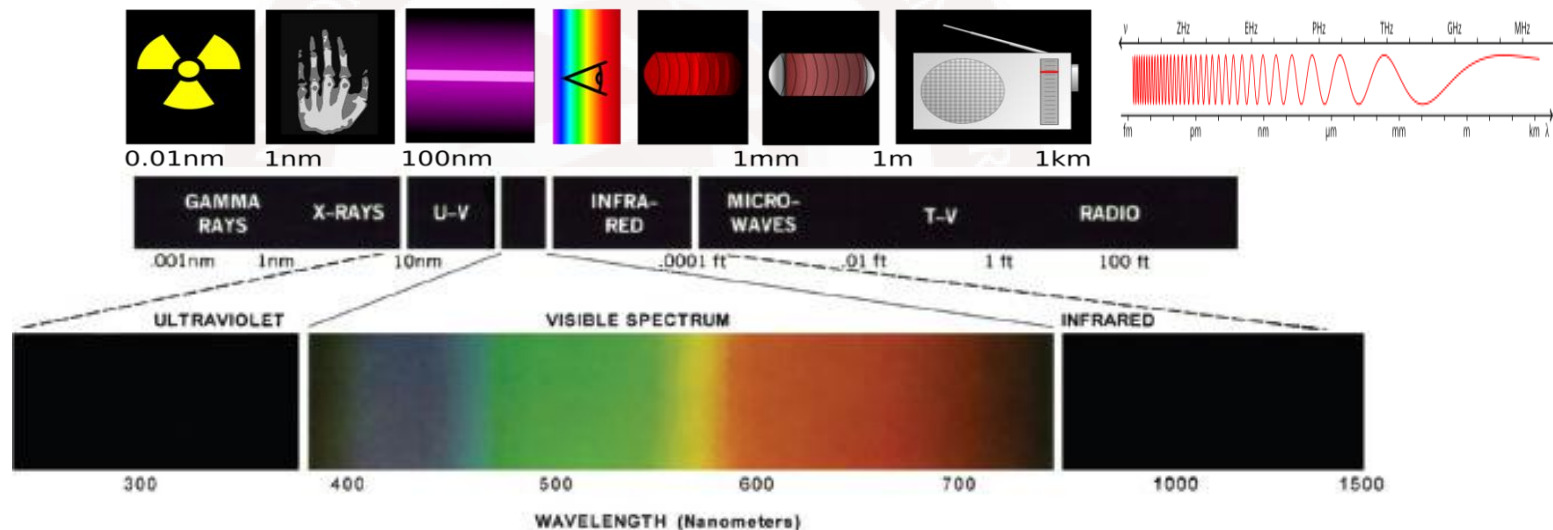
Green channel









Blue channel

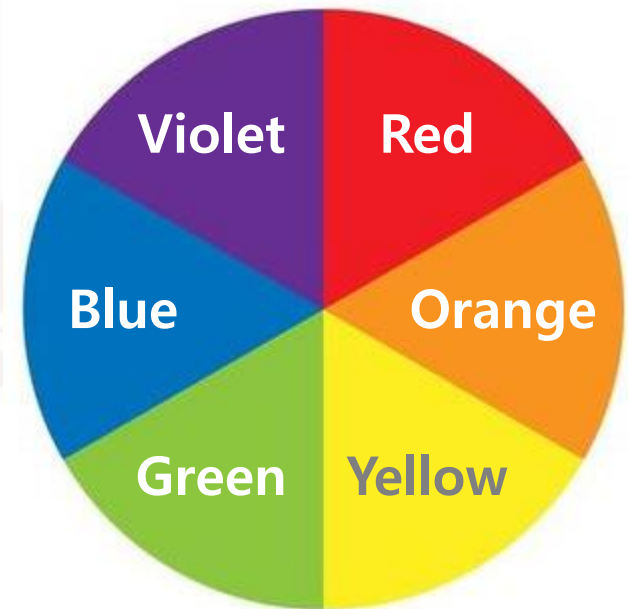
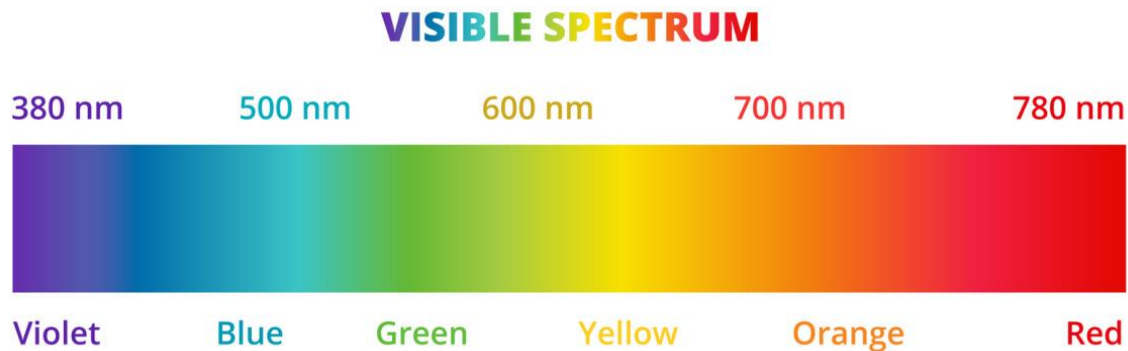
■ 빛

- 백색으로 생각하는 빛이 실제로는 무지개 색에 해당하는 다양한 파장으로 구성된다는 것을 18세기 초에 아이작 뉴턴의 광학에 관한 연구에서 발표됨
- 가시광선
 - 인간이 인지할 수 있는 약 380nm ~ 780nm 사이의 파장



■ 컬러 휠(컬러 서클)

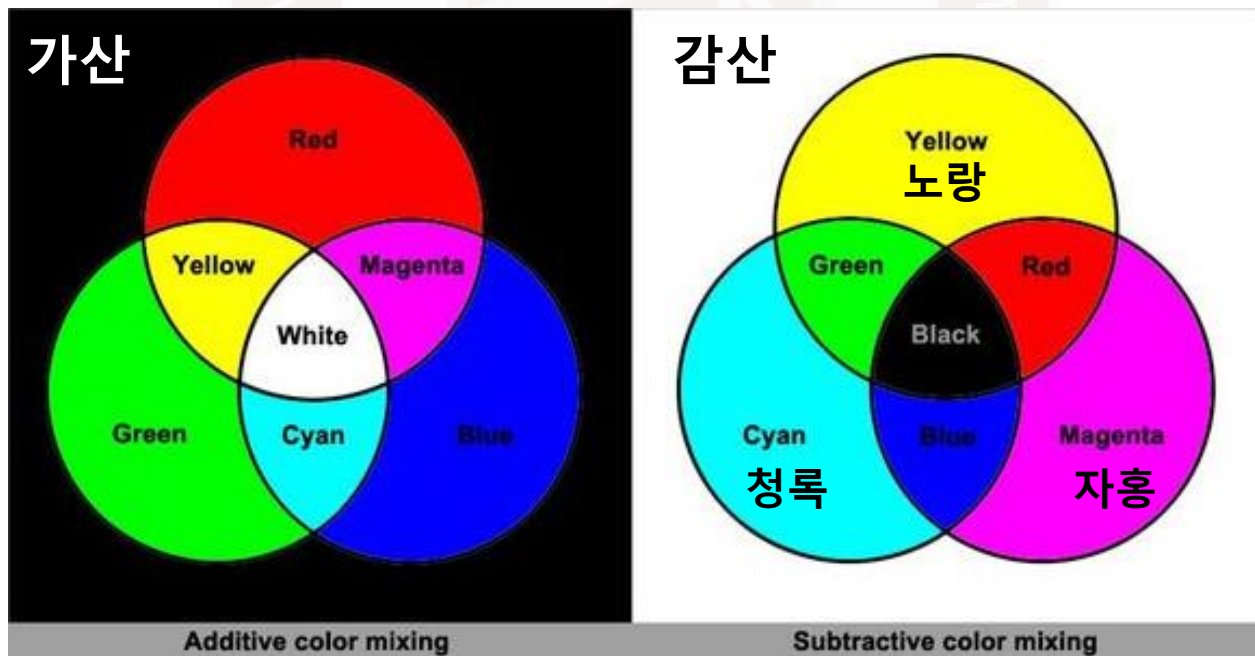
- 가시 스펙트럼은 파장에 의해 매개 변수화된 선형 직선에 놓여 있는 물리적(physical)인 세계에 존재하는 색
- 컬러 휠은 무지개에서 서로 가장 멀리 떨어져 있는 빨강과 보라색이 컬러 휠에서 인접하도록 무지개의 색상을 원처럼 감싸서 표현한 모델링으로, 심리적(psychological)인 세계에 존재하는 색
 - ex) 녹색은 두 개의 인접한 색상인 파란색과 노란색의 혼합 ( =  + )
 - ex) 주황색은 빨간색과 노란색의 혼합 ( =  + )



Color Wheel

■ 가산 및 감산 컬러 모델

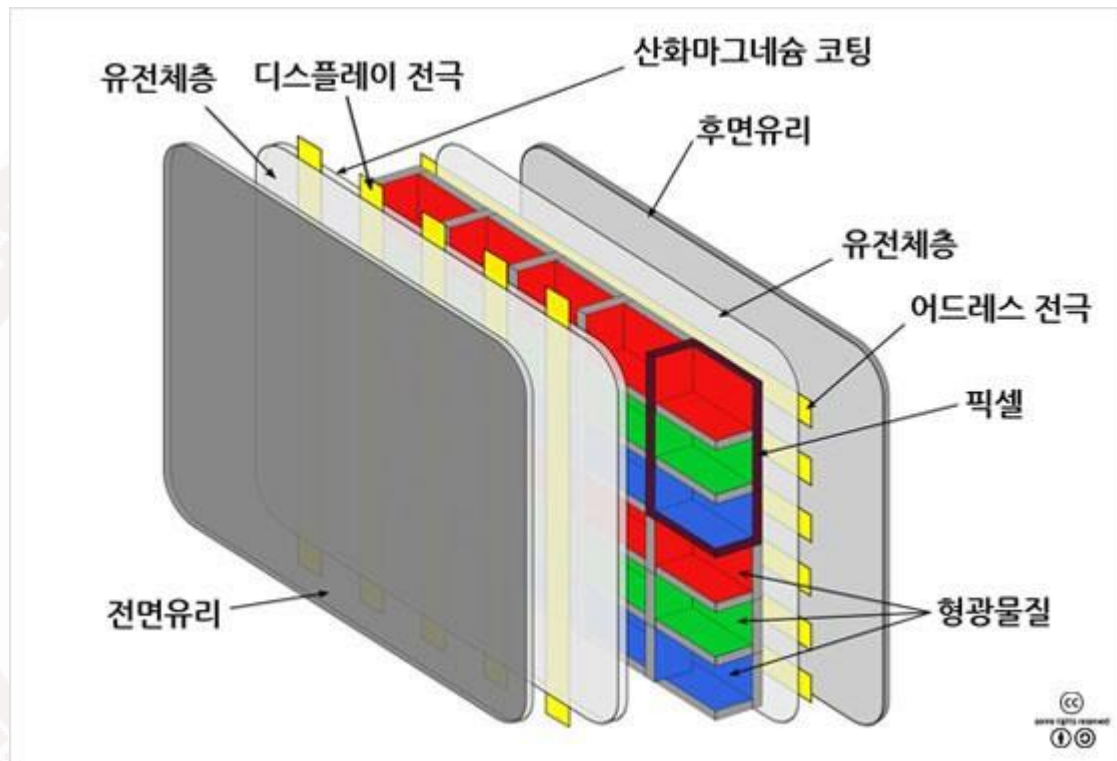
- 여러 개의 **조명**을 함께 비추어 중첩에 의해 장면이 밝아지는, **혼합된 모든 컬러가 흰색**이 되는 경우를 **가산 컬러 모델**이라고 한다. (※ 대상으로부터 직접 우리 눈으로 빛이 도착하는 경우)
- 청록, 자홍, 노랑은 각각 특정한 파장의 빛을 흡수하고, 나머지 파장의 빛을 반사하여 색을 만들어내며, 색상이 혼합될수록 더 많은 빛이 흡수되기 때문에 색상이 더 어두워진다. 즉, **함께 혼합되는 모든 컬러가 검정색**이 된다. 따라서 **잉크 또는 페인트 혼합으로 인해 발생하는 컬러**를 설명하는 컬러 모델은 **감산 컬러 모델**이라고 한다. (※ 1차적으로 반사된 빛이 우리 눈으로 빛이 도착하는 경우)



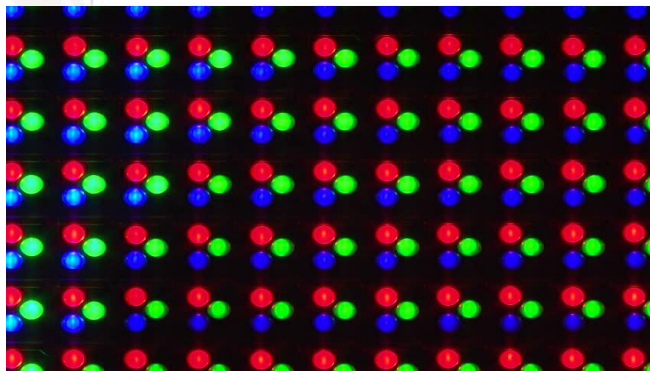
1.2 영상의 기본

■ 디스플레이 장치에서 RGB

Display Panel



LED 확대 영상 (RGB로 이루어진 발광 다이오드)



1.2 영상의 기본

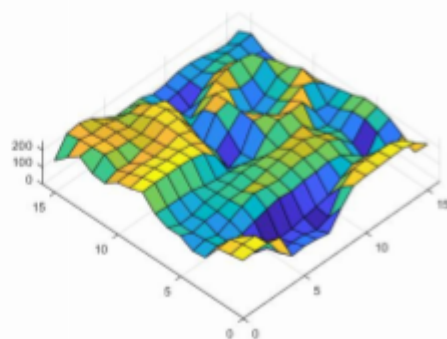
■ 컬러 영상

■ 용량

- 가로(1920)x세로(1080)x깊이(3채널x8비트)
- Full HD: $1920 \times 1080 \times 3 \times 8 = 49,766,400$ bits = 6,220,800 bytes (약 6MBytes)



Clip color sample



241	245	241	225	67	74	65	49	147	227	235	237	241	244	225	168
226	237	237	153	35	47	43	18	44	171	226	231	230	227	193	73
212	217	182	36	14	21	27	43	56	71	120	155	148	120	75	123
110	128	97	100	123	143	158	146	159	161	170	157	106	25	138	170
99	115	94	127	159	188	176	165	181	191	183	195	151	77	168	190
140	108	125	132	149	148	154	172	182	192	199	173	136	101	164	191
137	107	112	136	140	143	160	158	157	152	168	184	107	83	128	162
170	162	160	156	133	120	115	106	84	172	217	215	219	117	107	144
247	251	250	248	243	224	156	34	73	214	202	146	181	218	192	125
221	234	239	235	238	227	187	85	148	220	152	96	133	165	219	113
205	210	213	205	214	233	207	83	208	228	102	77	116	175	181	76
152	179	169	194	203	235	233	131	181	223	200	157	165	229	166	81
161	221	206	200	192	231	235	174	80	145	201	224	225	209	70	65
210	215	205	203	183	202	217	186	45	41	98	144	147	106	92	78
155	210	193	197	176	164	168	111	72	84	101	105	120	113	116	124
88	195	205	182	135	96	98	101	75	81	103	105	93	73	66	92

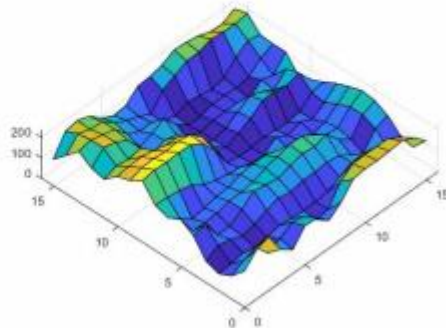
1.2 영상의 기본



Clip color sample



Green
channel

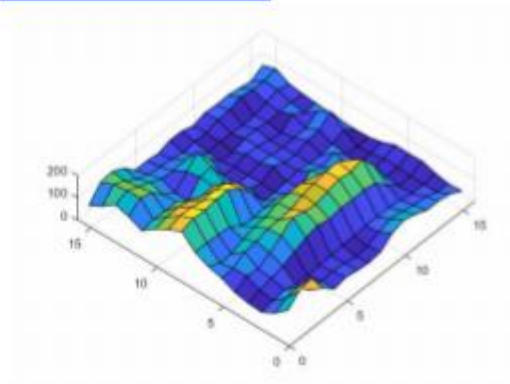


97	159	182	79	35	110	107	54	101	171	180	179	183	190	165	110
35	69	89	22	50	89	75	18	28	119	164	168	167	165	128	39
6	13	24	10	20	33	25	22	26	32	61	91	95	75	40	57
27	64	50	26	43	67	87	74	89	88	94	85	45	14	90	115
113	125	33	33	74	122	111	100	121	138	130	149	83	43	120	147
152	54	55	58	75	77	85	112	126	142	143	106	66	58	119	156
121	75	32	45	57	61	84	92	92	74	39	36	32	48	93	136
143	138	131	117	78	56	45	36	17	12	60	27	51	21	62	113
221	233	229	234	230	196	121	18	8	11	19	12	8	19	72	80
184	204	208	201	208	198	148	56	13	42	71	118	118	108	126	52
164	171	176	166	149	147	96	16	11	27	64	101	79	46	107	33
111	134	125	138	70	93	80	7	11	25	27	27	6	56	36	109
126	184	169	137	63	87	118	48	4	5	22	25	19	31	80	108
168	182	166	150	87	58	59	23	10	35	26	13	28	86	127	115
109	174	152	149	102	71	51	20	94	124	154	156	177	169	169	178
51	145	157	136	85	57	64	116	122	136	168	164	154	132	130	161

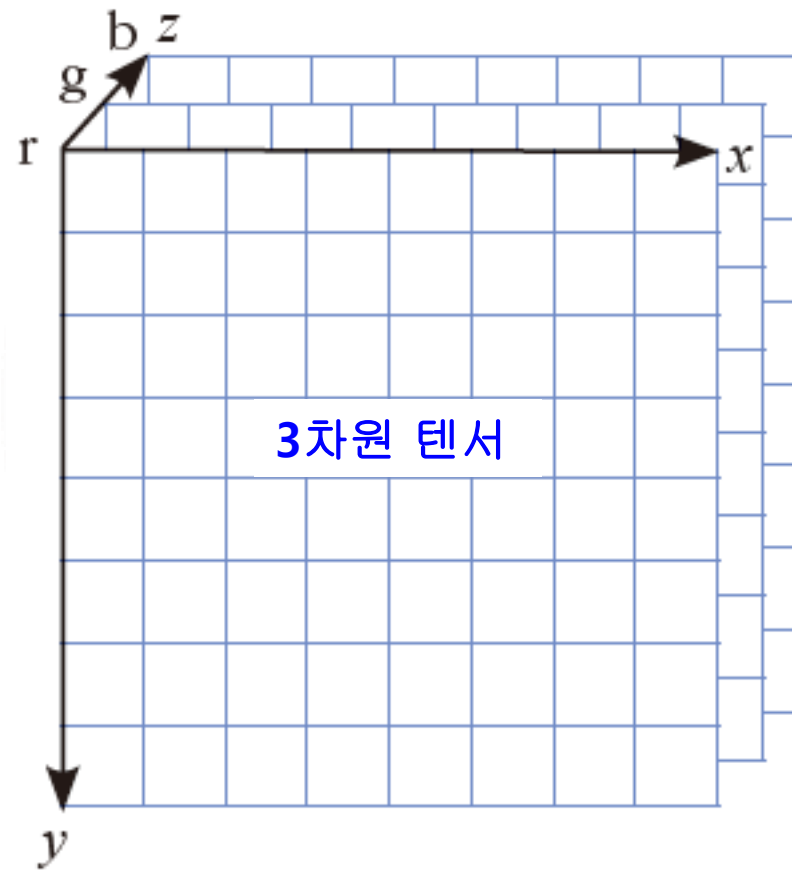
1.2 영상의 기본



Clip color sample



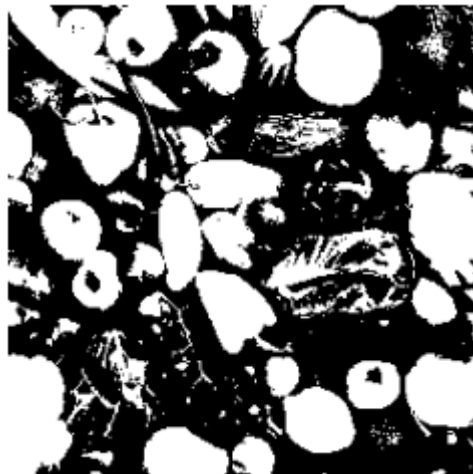
86	133	149	64	19	31	22	16	24	66	84	69	48	49	39	12
37	66	73	19	18	23	18	7	14	37	54	48	33	32	12	9
7	18	26	8	10	13	17	22	24	29	29	29	22	11	12	21
12	28	43	55	79	102	117	103	113	109	114	102	55	9	13	40
25	58	49	79	116	156	143	134	152	166	158	174	106	15	18	38
53	56	89	94	112	113	117	140	153	165	167	135	87	13	21	41
69	46	46	71	83	90	111	116	118	100	64	56	41	7	10	18
98	99	95	89	54	48	54	53	32	21	61	32	57	9	4	6
155	172	169	178	179	145	85	10	6	17	22	15	13	22	46	4
128	149	146	137	143	135	93	31	12	48	25	21	41	49	89	6
96	109	114	101	79	84	61	12	12	27	22	15	20	20	55	12
53	64	59	63	24	90	96	17	14	32	27	15	6	48	14	17
81	124	106	58	24	80	124	56	7	6	25	27	18	23	24	23
98	122	97	77	52	48	56	29	7	16	14	8	12	30	46	32
56	113	88	57	36	30	25	13	23	20	38	42	62	51	51	64
25	86	94	70	21	22	15	33	33	16	34	29	32	24	13	44



1.2 영상의 기본

■ 이진 영상(binary image)

- 1픽셀은 **밝기 값을 의미**하는 1개 채널로 구성되며 2비트를 할당
- 1개 픽셀은 0과 1의 총 2개 밝기값
- 용량
 - 가로(1920)x세로(1080)x깊이(1비트)
 - Full HD: $1920 \times 1080 = 2,073,600 \text{ bits} = 259,200 \text{ bytes}$



Q. 이진 영상은 몇 차원 텐서로 구성되는가?

1.2 영상의 기본

■ 인덱스 영상(indexed image)

- 1개 픽셀은 RGB 3채널이 아닌 1채널로 구성
 - 1개 채널 값은 컬러 영상, 회색조 영상, 이진 영상과 같은 밝기 값이 아니라, 인덱스 값으로, RGB 밝기 값을 갖고 있는 색상표(Palette)에서 위치에 대한 인덱스로 활용
- 1채널에 8비트를 할당
- 일반적으로 적은 수의 비트로 컬러 영상을 표현할 때 활용
- 컬러 영상에 비해 **인덱스 영상의 화질은 저하되지만 영상 용량도 낮음**



Color image (RGB channel)



Indexed image (1 channel)

1.2 영상의 기본

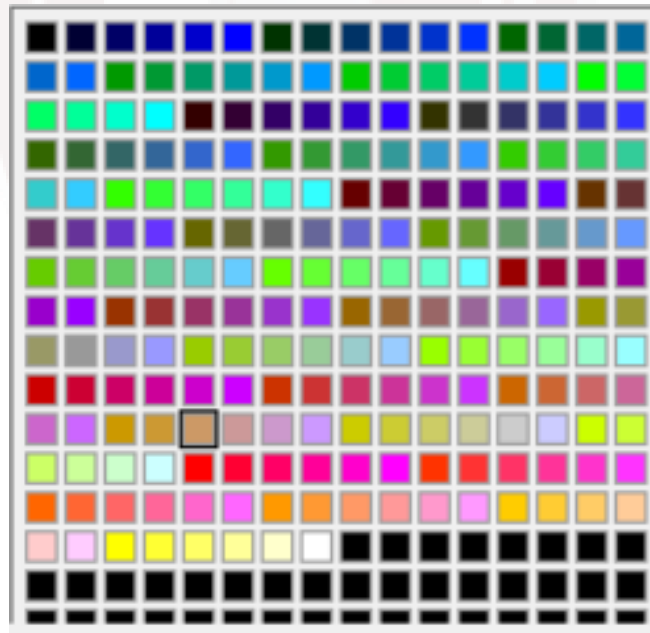
■ 인덱스 영상(indexed image)

■ 색상표

- 팔레트(Palette)
- 컬러맵(Color map)

■ 용량

- 가로(1920)x세로(1080)x깊이(8비트)+RGB(3채널x8비트)x256개
- Full HD: $1920 \times 1080 \times 8 + 3 \times 8 \times 256 = 16,594,944$ bits = 2,074,368 bytes (약 2MBytes)



남자 : “이게 뭐가 달라??? 똑같잖아!”



순서대로

스윗데블 루즈, 돌리레드, 베이비플레임, 래커레드, 스파이키 로즈, 몽슈레드, 리얼레드, 모던레드, 못된핑크

VS

여자 : “이게 뭐가 달라??? 똑같잖아!”



1.2 영상의 기본

■ 인덱스 영상(indexed image)



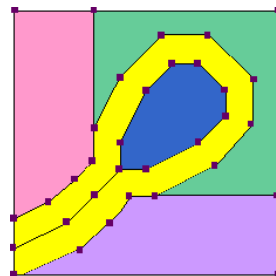
194	201	207	157	42	49	48	42	120	163	206	205	205	205	163	120
151	187	193	108	42	48	42	0	42	121	163	199	199	163	162	42
144	144	145	36	0	6	36	36	42	43	79	121	120	78	42	78
78	115	79	79	80	116	122	116	122	122	122	79	0	120	121	121
84	85	79	80	116	159	123	123	159	165	165	165	122	78	120	163
127	79	80	116	116	122	122	123	159	165	165	123	116	78	120	163
121	79	79	115	116	116	122	122	122	116	115	151	79	78	120	126
128	128	128	122	121	79	79	79	73	108	151	151	151	72	78	120
207	213	207	213	214	171	122	36	36	144	144	108	144	144	151	84
171	207	207	207	207	171	164	79	108	151	114	84	121	121	158	78
164	164	164	164	164	200	157	72	144	151	78	84	84	114	157	42
121	163	121	163	150	194	194	108	144	145	151	114	108	151	114	84
122	170	164	163	150	194	194	115	72	108	144	145	144	150	48	48
164	170	164	164	157	151	151	145	36	42	78	108	114	85	85	85
121	164	164	163	121	115	114	72	48	84	91	91	91	91	91	91
78	164	164	163	120	78	78	85	49	90	91	91	91	54	54	91

색상표

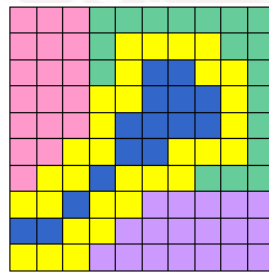
Index	R	G	B
0	51	255	204
1	51	255	255
...			
36	102	0	51
37	102	0	102
38	102	0	153
39	102	0	204
40	102	0	255
41	102	51	0
42	102	51	51
43	102	51	102
44	102	51	153
45	102	51	204
46	102	51	255
47	102	102	0
48	102	102	51
...			
253	102	102	153
254	102	102	204
255	102	102	255

■ 이미지 파일(image file)

- 대부분의 영상 파일들은 래스터 (raster) 그래픽 영상 파일
- 래스터 그래픽스(Raster graphics) 이미지란 비트맵 이미지라고 부르기도 하며, 직사각형 격자의 화소, 색의 점을 모니터, 종이 등의 매체에 표시하는 자료 구조
 - 래스터 파일 확장자명: .tiff / .psd / .pdf / .jpg / .png / .gif / .bmp
- 벡터 (vector)영상 파일은 선과 점, 면 그리고 곡선으로 이루어진 영상으로 해상도가 커져도 화질이 저하되지 않고 항상 매끄럽고 선명한 이미지를 보여주는 영상
 - 영상 속 색상 정보를 점, 선 면으로 표현하려면 비트맵 방식보다 더 많은 용량과 위치정보를 갖고 있어야 하기 때문에 복잡한 이미지일 수록 래스터/비트맵 방식보다 효율이 떨어짐
 - 특수 이미지에 가까우며 Adobe Illustrator등의 특정 SW에서만 생성할 수 있음
 - 벡터 파일 확장자명: .ai / .eps / .svg / .pdf



Vector



Raster



Raster



Vector

1.3 영상의 입출력

■ 영상 입출력 코드

- 지정한 컬러 영상 파일을 회색조(gray scale) 형태로 읽어서 화면에 출력
- bmp 및 jpg로 저장을 수행
 - .bmp는 미압축, 미가공 파일로 파일 크기가 큰 고품질 이미지 파일 포맷
 - .jpg는 손실 압축 방법으로 매우 높은 해상도의 이미지를 처리할 때 품질이 저하될 수 있음

```
import cv2

if __name__ == '__main__':
    img = cv2.imread('./images/Lenna.jpg', cv2.IMREAD_GRAYSCALE)
    cv2.imshow("Image", img)
    cv2.imwrite('Lena_1.bmp', img)
    cv2.imwrite('Lena_2.jpg', img)
    cv2.waitKey(0)
```

■ 실행 결과

- 원본의 파일크기: 463KB
- 회색조 bmp 저장 크기: 258KB
- 회색조 jpg 저장 크기: 81KB



Lenna.jpg
(원본)



Lena_1.bmp



Lena_2.jpg

■ 영상 불러오기 imread 함수

- 래스터 그래픽 영상 파일 형식들을 메모리로 읽어들이м
- 파일 시그니처(file signature)를 읽어 적절한 코덱을 결정하고, 압축 해제된 영상 데이터 구조에 필요한 메모리 할당과 같은 복잡한 작업들을 처리
- 확장자가 아닌 파일 시그니처를 읽어 파일의 포맷을 분석

함수명	cv2.imread(fileName, flags)
매개변수	<ul style="list-style-type: none">- fileName (str) 읽어들일 영상 파일의 경로(상대 경로 또는 절대 경로 이용)- flags (int) 영상 파일을 읽을 때 옵션<ul style="list-style-type: none">cv2.IMREAD_COLOR: 컬러로 읽어 들이기 (3채널 BGR 영상) (1)cv2.IMREAD_GRAYSCALE: 회색조로 읽어 들이기 (0)cv2.IMREAD_UNCHANGED: alpha channel까지 포함하여 읽기 (-1)cv2.IMREAD_REDUCED_GRAYSCALE_2: 1채널 1/2크기 grayscale 영상cv2.IMREAD_REDUCED_GRAYSCALE_4: 1채널 1/4크기 grayscale 영상cv2.IMREAD_REDUCED_GRAYSCALE_8: 1채널 1/8크기 grayscale 영상cv2.IMREAD_REDUCED_COLOR_2: 3채널 1/2크기 BGR영상cv2.IMREAD_REDUCED_COLOR_4: 3채널 1/4크기 BGR영상cv2.IMREAD_REDUCED_COLOR_8: 3채널 1/8크기 BGR영상
리턴값	image 객체 행렬(numpy.ndarray)

■ 영상 저장 imwrite 함수

- 영상을 래스터 그래픽 영상 파일 형식으로 저장함
- 저장하는 파일의 형식은 파일 확장자에 따라서 결정됨
 - .jpg / .bmp / .png

함수명	cv2.imwrite(fileName, image, flags)
매개변수	<ul style="list-style-type: none">- fileName (str) 저장할 영상 파일의 경로(상대 경로 또는 절대 경로 이용)- image (numpy.ndarray) 저장할 영상 데이터- flags (int) 영상 파일을 저장할 때 옵션
리턴값	True 또는 False (bool)

■ 영상 화면 출력 imshow 함수

- 영상을 지정한 윈도우에 출력
- "윈도우 이름"에 따라서 출력되는 윈도우가 결정
 - ※ 일치하는 윈도우 이름이 없으면 새로운 윈도우를 생성하여 출력

함수명	cv2.imshow(window_name, image)
매개변수	<ul style="list-style-type: none">- window_name (str) 윈도우 창의 Title 내용- image (numpy.ndarray) 출력할 영상 데이터
리턴값	N/A

■ 영상정보 출력

- 입력 영상에 대한 넓이, 높이, 채널 수 정보 출력

```
import cv2

if __name__ == '__main__':
    # color image
    img = cv2.imread('./images/Lenna.jpg', cv2.IMREAD_UNCHANGED)
    height, width, channel = img.shape
    print("height: {}, width: {}, channel: {}".format(height, width, channel))

    # grayscale image
    img = cv2.imread('./Lena_2.jpg', cv2.IMREAD_UNCHANGED)
    height, width = img.shape
    print("height: {}, width: {}".format(height, width))
```

1.3 영상의 입출력

■ 영상 정보 확인: shape 속성

- image 데이터는 numpy.ndarray 형태로서 shape 속성 포함
- 영상에 대한 높이(height), 넓이(width), 채널 수(channel) 정보

■ shape 속성

- 컬러영상: shape 속성이 height, width, channel 정보로 구성
- 회색조 영상: shape 속성이 height, width 정보로 구성
- `len(image.shape)` 함수로 3채널인지 2채널인지 구분하는 처리가 필요

■ waitkey 함수

- 지정된 시간 동안 사용자가 키를 입력할 때까지 프로그램을 대기
- 대기하는 시간은 밀리 초(milliseconds) 단위로서 1,000이 1초에 해당

함수명	cv2.waitKey(delay)
매개변수	<ul style="list-style-type: none">- delay (int) 대기할 시간 (밀리 초 단위) delay가 0보다 작거나 같은 경우 무한 (infinite) 대기를 의미
리턴값	사용자가 입력한 키 값 (int)

■ waitkey 함수

- 지정된 시간 동안 사용자가 키를 입력할 때까지 프로그램을 대기
- 대기하는 시간은 밀리 초(milliseconds) 단위로서 1,000이 1초에 해당

```
import cv2

img = cv2.imread("./images/Lenna.jpg", cv2.IMREAD_UNCHANGED)
cv2.imshow('Image', img)

# cv2.waitKey(1000)
key = cv2.waitKey(0)
print("key = {0} ({1})".format(key, chr(key)))
```