

컴퓨터 그래픽스

제9장 3차원 객체의 모델링

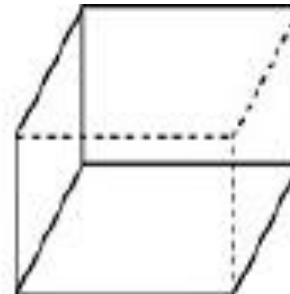
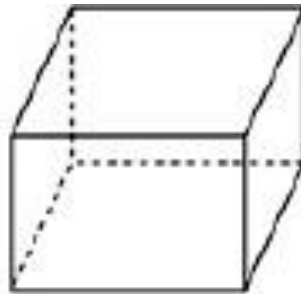
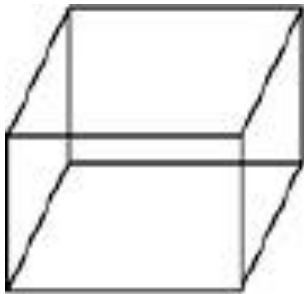
2016년 2학기

9장 학습 내용

- 은면 제거
 - 은면제거 개념
 - 은면제거 알고리즘
- 컬러 모델

은면 제거 (Hidden Surface Removal)

- 3차원 객체를 2차원 평면에 투영
 - 시점에서 가장 가까운 객체가 화면에 나타난다.
 - 관측자의 시점에서 보이는 면만을 나타낸다
 - 입체감 있는 3차원 화면
 - 렌더링 속도 향상



은면 제거 (Hidden Surface Removal)

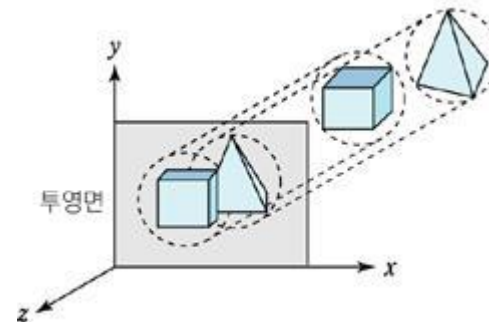
- 알고리즘의 종류

- 객체 공간법 (Object Space Method)

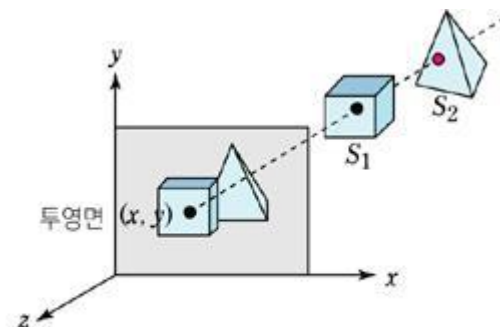
- 공간상에 있는 두 객체의 공간적 위치 관계를 이용하여 은면 결정
 - 객체 수가 적거나 서로 분산된 경우에 효율적
 - 객체수가 많은 경우에는 계산이 복잡하다
 - 깊이 정렬 알고리즘

- 이미지 공간법 (Image Space Method)

- 투영된 픽셀 평면에서 공간상의 객체가 보이는지 여부를 검사하는 방법
 - 투영면의 해상도에 의해 처리속도가 좌우
 - 많은 메모리가 필요
 - Z-buffer 알고리즘



(a) 객체 공간법



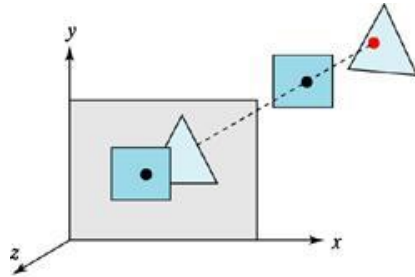
(b) 이미지 공간법

은면 제거 처리 개념

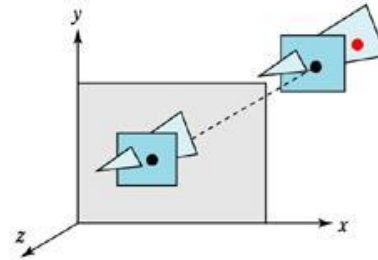
- 은면 제거 알고리즘을 위한 원리 및 성질

- 객체의 각 표면을 정렬 (Sorting)

- 투영면으로부터의 거리 (에 따라 객체들을 정렬 또는 객체를 구성하는 각 면들을 정렬: z 축에 따라 정렬 \rightarrow 투영면에 가장 가까운 객체나 객체의 면들을 선택하여 그린다.
 - 수평방향 또는 수직방향 정렬: x 축, y 축에 따라 정렬 \rightarrow 겹치는지 여부를 판단
 - 깊이 정렬법



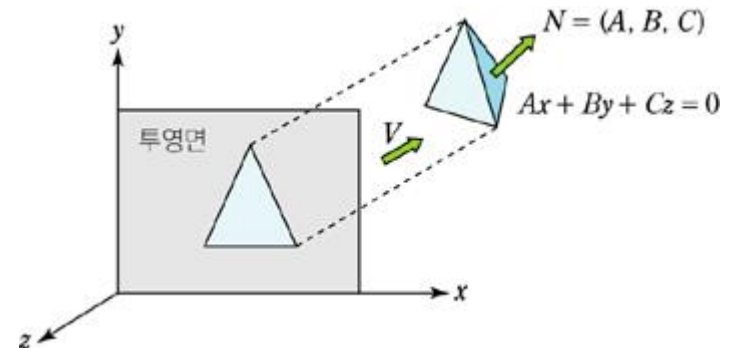
(a) 객체가 정렬되어 있는 경우



(b) 별도의 처리과정이 필요한 경우

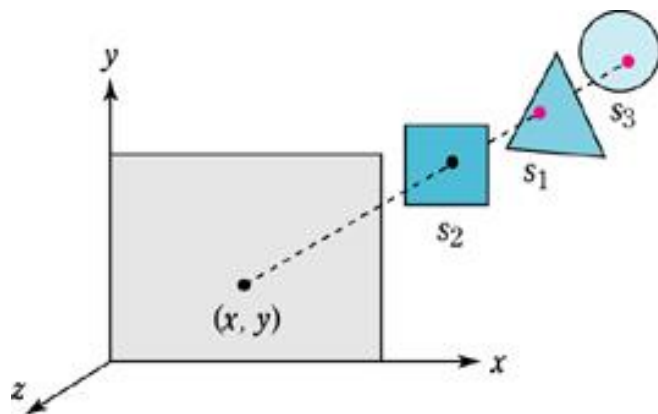
뒷면 제거 (Back-face Removal)

- 은면 제거의 첫 단계에서 이용
 - 빠르고 간단하게 뒷면을 찾는 객체공간 방법
 - 시점에서 보이는 앞면은 화면에 낙타내고 뒷면은 낙타내지 않는다.
 - 일반적인 장면에서 다면체 면의 약 50% 제거
- 평면 방정식 이용
 - $Ax + By + Cz + D = 0$ 에서 법선 벡터 $N=(A,B,C)$
 - 시선 방향 벡터 V 와 법선 벡터 N : $V \cdot N > 0$ 이면 뒷면, $V \cdot N < 0$ 이면 앞면
 - 시선은 $-z$ 방향이므로 $C < 0$ 이면 뒷면
- 시선 방향: z 축의 음의 방향: 벡터 V
- $V \cdot N > 0$: 시점에서 보이지 않는 뒷면
- $V \cdot N < 0$: 시점에서 보이는 앞면
- $V = (0, 0, V_z)$, $N = (A, B, C)$
- $V \cdot N = V_z C$
- $C < 0 \rightarrow V_z C > 0 \rightarrow$ 뒷면
- $C > 0 \rightarrow V_z C < 0 \rightarrow$ 앞면

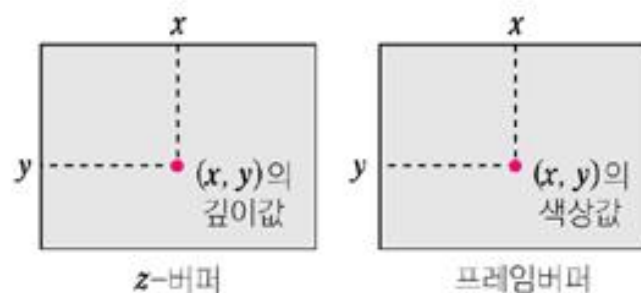


은면 제거: Z-Buffer 알고리즘

- Z-Buffer 알고리즘 (깊이 버퍼 알고리즘)
 - 가장 일반적으로 사용되는 이미지 공간 접근 방법
 - 물체의 가시성을 픽셀 단위로 조사
 - 시점과 투영면의 픽셀을 연결하는 직선에서 투영면에 가장 가까운 객체만이 투영면에 나타난다
 - 2개의 버퍼를 사용한다
 - Z-버퍼: 투영면상의 점인 픽셀 (x, y) 에 투영되는 z값 저장
 - Frame-버퍼: 색상값 저장



(a) z-버퍼 알고리즘의 개념



(b) z-버퍼와 프레임버퍼

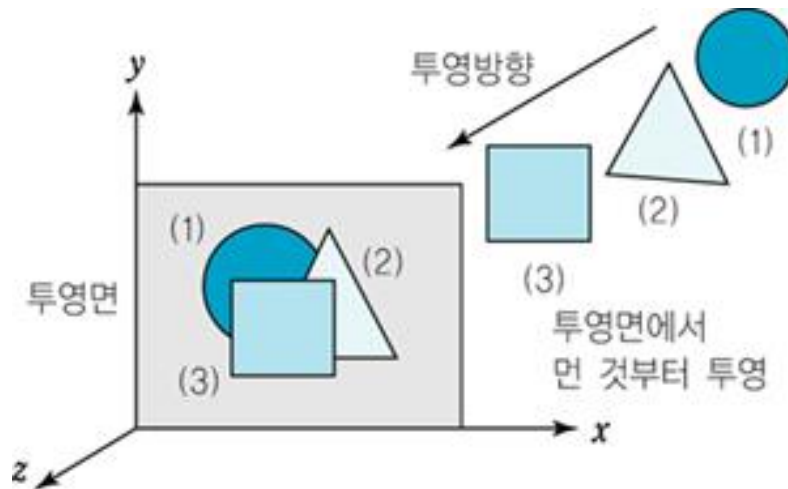
은면 제거: Z-Buffer 알고리즘

- 수행 과정

- 시선의 방향: z축 음의 방향, 투영면: $z=0$
- 공간상의 객체 가운데 시점에 가장 가까운 객체 (z값이 가장 큰 객체)가 투영면에 나타난다.
- 버퍼 초기화
 - Z-버퍼: 최소값 $\text{Depth}(x, y) = -\infty$
 - 프레임-버퍼: 배경의 색상값 $\text{Frame}(x, y) = \text{background}$
- 각 객체 면에서 한번에 한 개의 스캔 라인에서 각 (x, y) 점의 깊이를 비교한다
 - 만약 $z > \text{Depth}(x, y)$ 이면
 - $\text{depth}(x, y) = z$
 - $\text{Frame}(x, y) = \text{intensity}(P)$
- (x, y) 에서 다각형의 방정식은 $Ax + By + Cz + D = 0$
 - $z = (-Ax - By - D) / C$
- (x, y) 에서의 깊이가 z 라면, 다음 점 $(x+1, y)$ 에서의 깊이 z' 는
 - $z' = (-A(x+1) - By - D) / C$

은면 제거: 깊이 정렬 기법 (Depth-sorting)

- 깊이 정렬 알고리즘
 - 페인터 알고리즘(Painter's algorithm)
 - 다각형 면을 깊이 (z값)에 따라 정렬한 뒤, 먼 것부터 투영하여 그린다
 - 뒤쪽에 있는 면은 나중에 그린 면 (가까운 면)에 가려진다

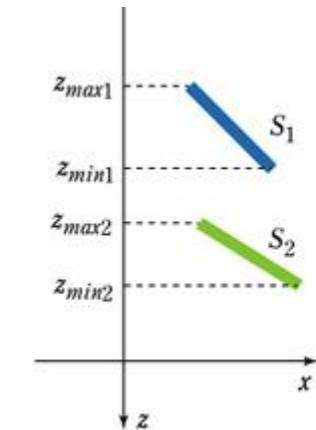


은면 제거: 깊이 정렬 기법 (Depth-sorting)

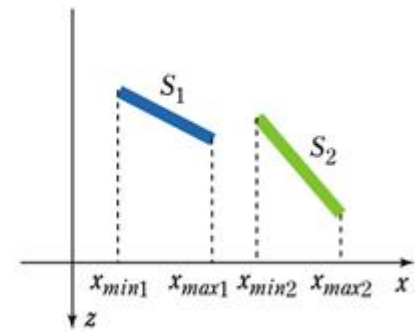
- 공간상의 객체: $z < 0$ 인 공간에 있다.
- 시선: z 축 음의 방향
- 투영면: $z = 0$
- 시점: $z > 0$ 인 공간에 존재
- 다각형을 z 축의 절대값에 따라 정렬: 투영면에서 가장 멀리 떨어진 면부터 순서대로 나열된다.

1) 깊이인 z -영역이 겹치지 않으면 멀리 있는 면부터 그린다. 깊이가 겹치는 경우에는 다음의 검사를 한다.

2) x -방향으로 겹치지 않으면 각각 그린다.



(a) z -영역이 겹치지 않는 경우



(b) x -영역이 겹치지 않는 경우

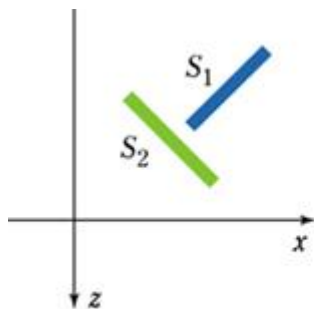
은면 제거: 깊이 정렬 기법 (Depth-sorting)

3) 다각형면 S_1 과 S_2 의 z 영역이 중첩된다

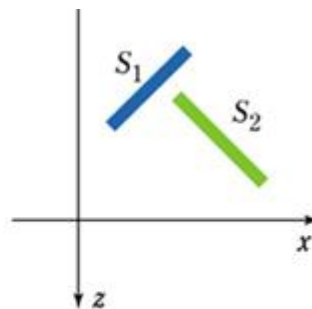
S_1 ($A_1x+B_1y+C_1z+D_1=0$) 이 면 S_2 ($A_2x+B_2y+C_2z+D_2=0$) 의 안쪽 (뒤쪽)에 있으면 → 다각형면 S_1 의 모든 꼭지점 (x, y, z) 에 대해 $A_2x+B_2y+C_2z+D_2 < 0$ → S_1 은 S_2 뒤에 있다 → S_1 부터 그린다.

4) 다각형면 S_1 과 S_2 의 z 영역이 중첩된다

S_2 가 투영하려는 S_1 면의 바깥쪽 (앞쪽)에 있으면 → 다각형면 S_2 의 모든 꼭지점 (x, y, z) 에 대해 $A_1x+B_1y+C_1z+D_1 > 0$ → S_2 는 S_1 의 앞에 있다 → S_1 부터 그린다.



(a) S_1 이 S_2 뒤에 위치

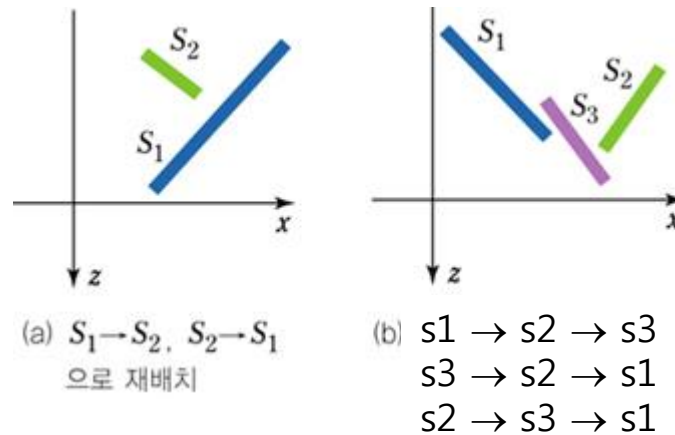


(b) S_1 의 앞에 S_2 가 위치

은면 제거: 깊이 정렬 기법 (Depth-sorting)

5) 면이 중첩될 때 앞의 4가지가 모두 아니면, 두 면의 순서를 바꾼다. (투영하려는 면 S_1 가 더 멀리 있지만 S_2 를 가리고 있는 경우)

6) 3개 이상의 면이 겹치는 경우 각각 2개씩 비교하여 순서를 정한다.

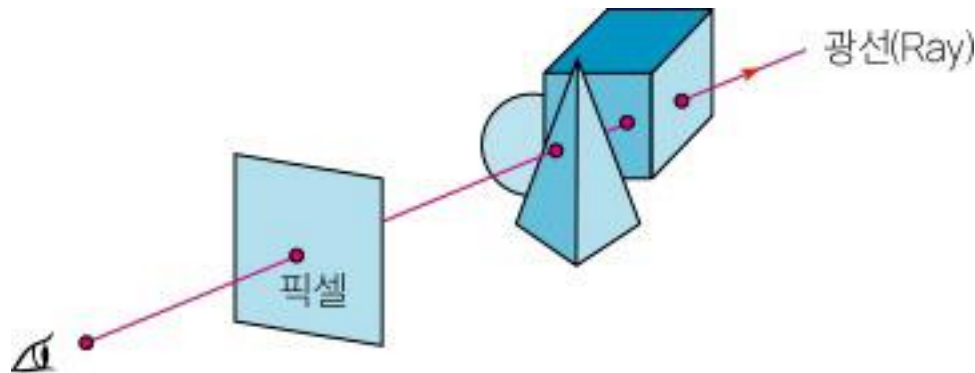


7) 해당하는 경우가 없으면, 일부 다각형 면을 절단하여 다시 적용한다.

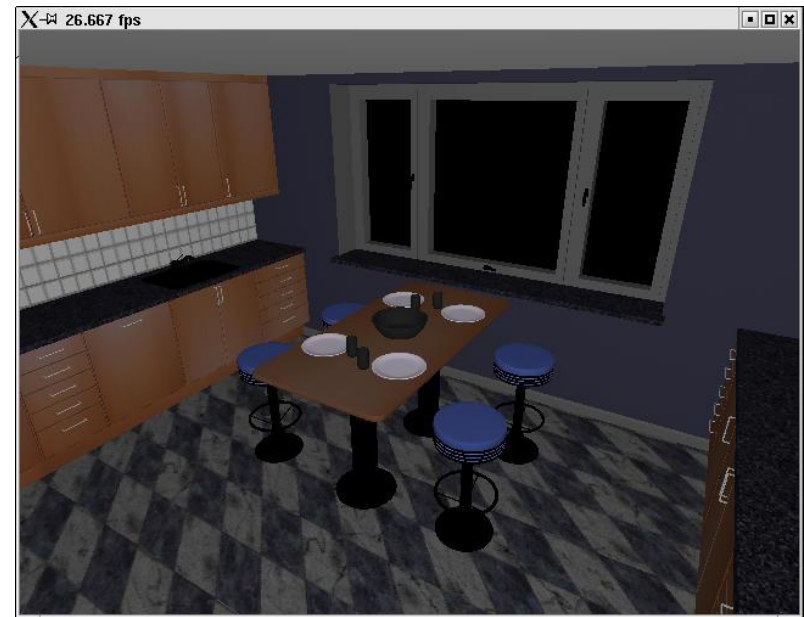
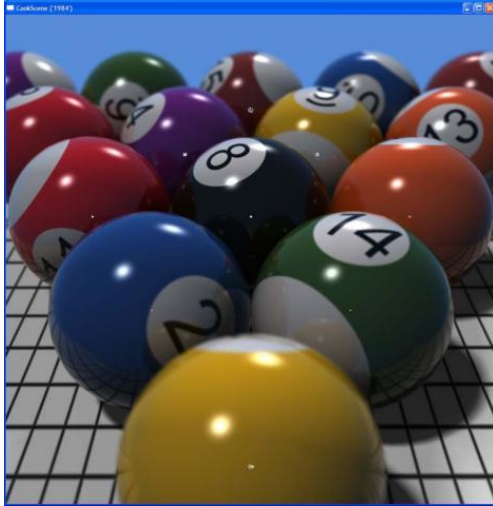
레이 캐스팅 (Ray casting)

- Ray Casting

- 시점에서 투영면의 각 픽셀을 통해 빛(ray)을 투사
 - 이 빛과 처음으로 만나는 객체를 선택하여 해당 픽셀을 그린다.
 - 임의의 곡면과 같은 표면에서 효과적인 은면제거 방법
 - z-버퍼 알고리즘의 변형
 - Z-버퍼 알고리즘은 면의 깊이를 비교하여 처리하나, 레이 캐스팅은 처음 만나는 면만을 선택하여 픽셀 단위로 처리



레이 캐스팅 (Ray casting)



컬러 모델

• 빛의 성질

– 빛이란: 가시광선

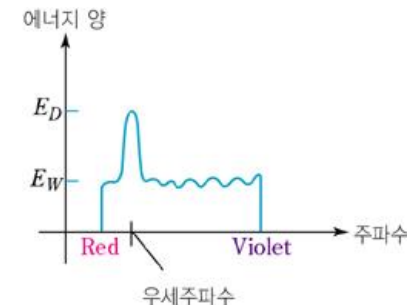
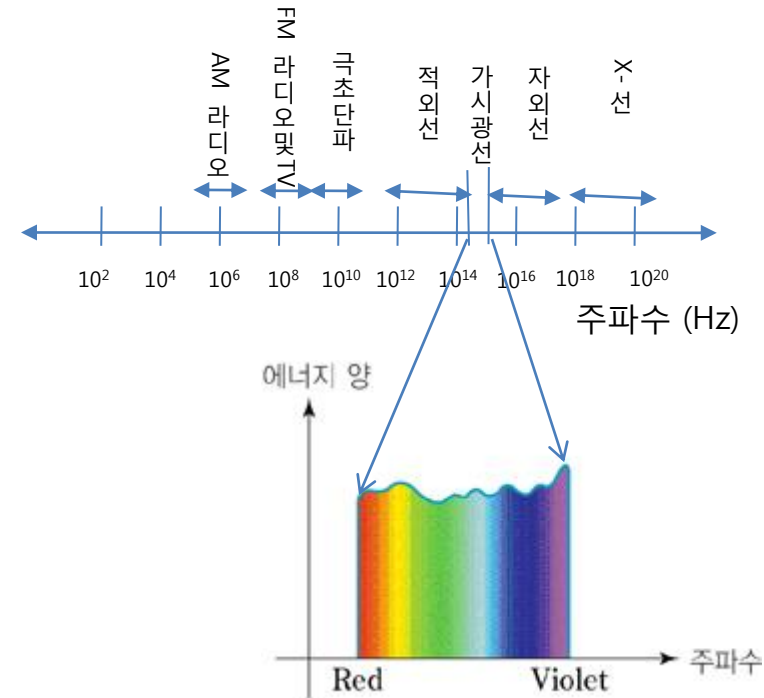
- 빨간색($4.3 \times 10^{14}\text{Hz}$)에서 보라색($7.5 \times 10^{14}\text{Hz}$) 주파수 대역에 걸쳐서 에너지가 분포

– 빛의 3가지 속성

- 색상 (Color): 고유의 주파수
 - 우세 주파수 (Dominant Frequency)에 해당되는 주파수가 색상
- 명도 (Luminance, Brightness): 빛이 가지고 있는 에너지의 양
- 채도 (Purity, Saturation): 빛의 순수도
 - 주파수의 범위가 좁게 집중되면 채도가 높다.

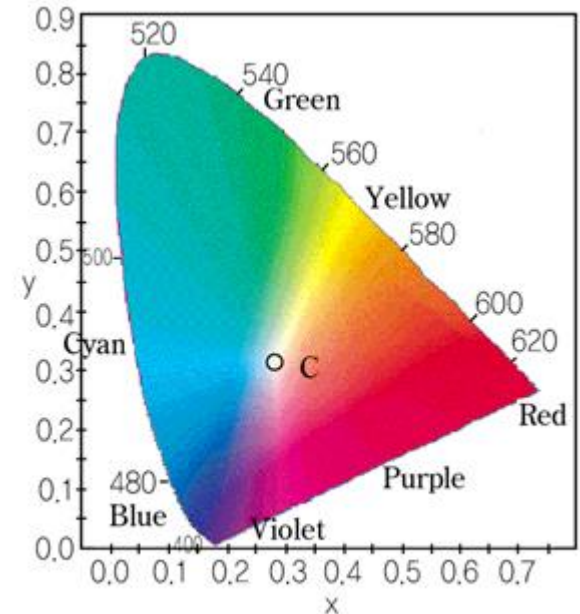
– 색채 (Chromaticity): 빛의 색상과 채도를 통합적으로 표현

– 보색 (Complementary color): 두 빛이 혼합되어 백색광이 될 때 (red와 cyan, green과 magenta, blue와 yellow 는 보색 관계)



컬러 모델: CIE 색채도

- 1931년 CIE (Commission Internationale de l'Éclairage)라는 국제표준기구에서 주색상 역할을 할 수 있는 3가지 색을 지정
 - 3가지 색상 (A, B, C): 가상의 색 → CIE 색 원소 (CIE Primaries)
 - 보이는 색상: $x + y + z = 1$
where $x = A/(A+B+C)$
 $y = B/(A+B+C)$
 $z = C/(A+B+C)$
 - 임의의 색 표시: (x, y, z) (실제로는 (x, y))
 - CIE 색채도에서 x 는 A, y 는 B의 값
 - 보라색 (violet): $(0.18, 0) \rightarrow (0.18, 0, 0.82)$
 - 색채도의 가장자리에 위치한 색 → 순수한 색상

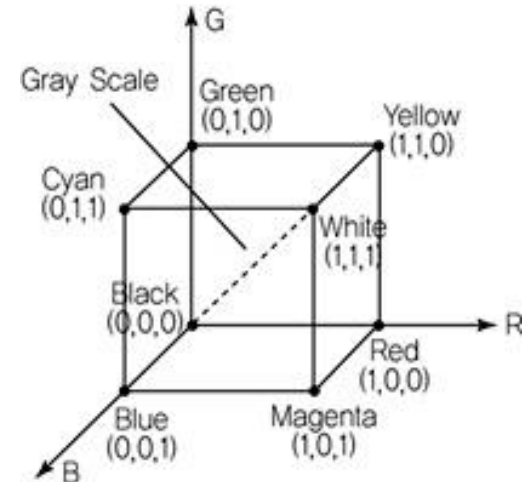


색채도 내의 임의의 색은 가장자리의 색상들의 조합으로 표현
C: 태양의 백색광

컬러 모델: RGB, CMY 컬러 모델

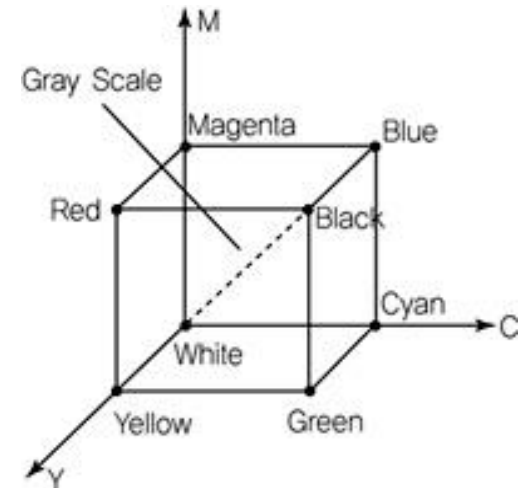
- RGB color model

- Video monitor(또는 사람의 눈)의 원리에 적합
- 가산모델(Additive model) : 3가지 R, G, B 빛을 더하여 원하는 빛을 생성
- 디스플레이 모니터상의 색상 표현에 적절한 모델

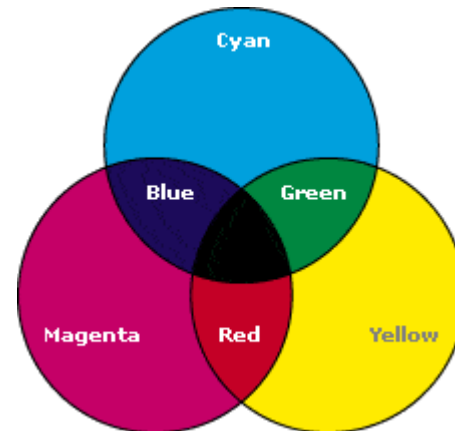
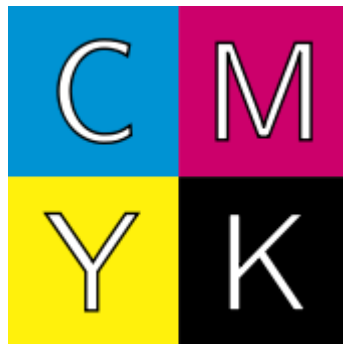
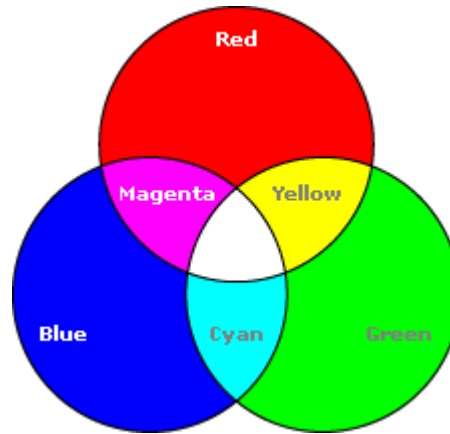
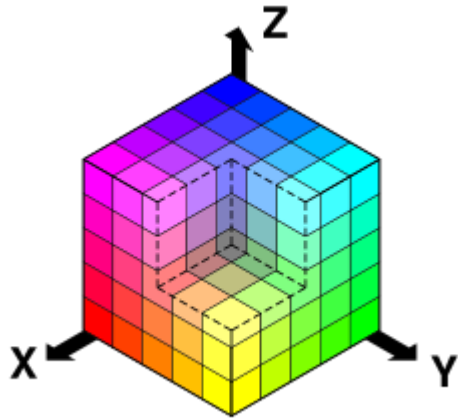


- CMY color model

- 청록색(Cyan), 자홍색(Magenta), 노란색(Yellow)
- Hard-copy 출력장치 (예, Ink-jet printer)에 적당
- 감산모델(Subtractive model) : 빛이 물체에 닿으면 물체의 고유한 색상에 해당하는 빛의 성분은 반사되고 나머지 부분은 흡수 (사람 눈은 반사된 빛 인식)



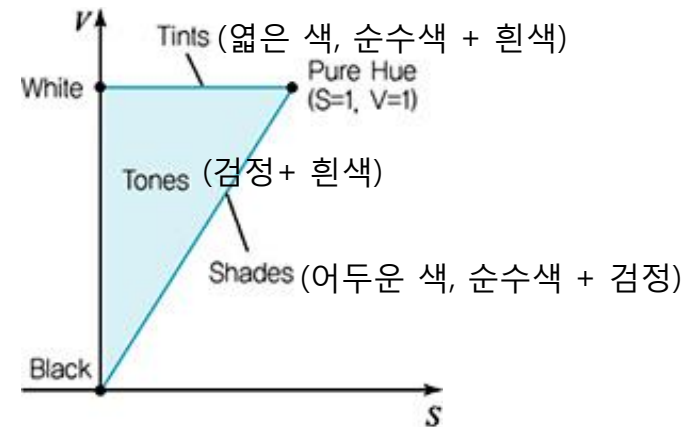
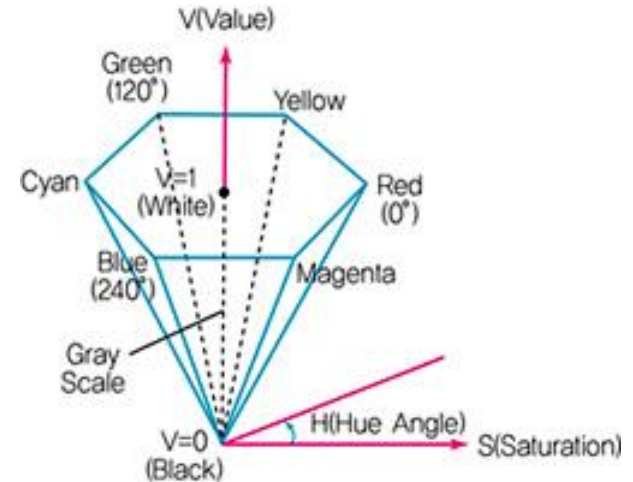
컬러 모델: RGB, CMY 컬러 모델



컬러 모델: HSV 컬러 모델

• HSV 컬러 모델

- 인간이 색을 사용하는 방식과 가장 근접한 모델
- 색조(Hue), 채도(백색과의 혼합도, Saturation), 밝기 (Value)
- HSV 육각뿔
 - 색상 (h): 정육면체의 둘레
 - $0^\circ \leq H \leq 360^\circ$
 - 채도 (s): 정육각형의 중심으로부터 둘레 사이
 - $0 \leq S \leq 1$
 - 정육면체의 중심의 위치는 백색($S = 0$),
 - 둘레의 위치가 색상의 채도는 1 ($S = 1$)
 - 밝기 (v): 상하 방향의 축
 - $0 \leq V \leq 1$
 - $V = 1$ 인 경우는 가장 밝은 색,
 - $V = 0$ 은 가장 어두운 색



컬러 모델: HSV 컬러 모델

H	S	V	색
0	1.0	1.0	Red
120	1.0	1.0	Green
240	1.0	1.0	Blue
	0.0	1.0	White
		0.0	black

H, S, V 값을 각각 0 ~ 255 범위로 사상시키거나 백분율로 사상시켜 사용하기도 한다.

