



3D MAX

1장. 학습 내용

- 컴퓨터 그래픽스 개요
- 모델링과 렌더링
- 인간의 시각과 색
- 컬러 모델

2

1.1 컴퓨터 그래픽스 개요



- 컴퓨터 그래픽스 개요
 - 컴퓨터 그래픽스의 정의
 - 컴퓨터 그래픽스 관련 학문 분야
 - 컴퓨터 그래픽스의 응용분야
 - 컴퓨터 그래픽스의 발전 과정

3

컴퓨터 그래픽스의 정의

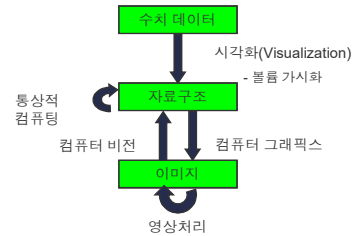
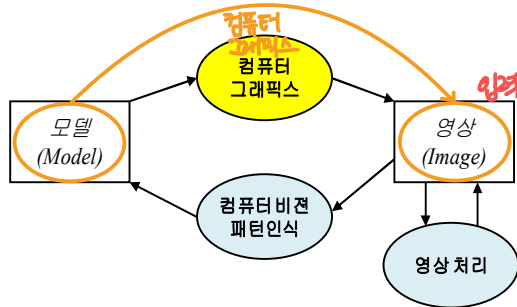


- 컴퓨터 그래픽스
 - “컴퓨터를 사용하여 그림을 생성하는 기술”
- 국제 표준화 기구(ISO/IEC JTC 1/SC 24)
 - Methods and techniques for converting data to and from a graphic display via computer.

4

★
샘

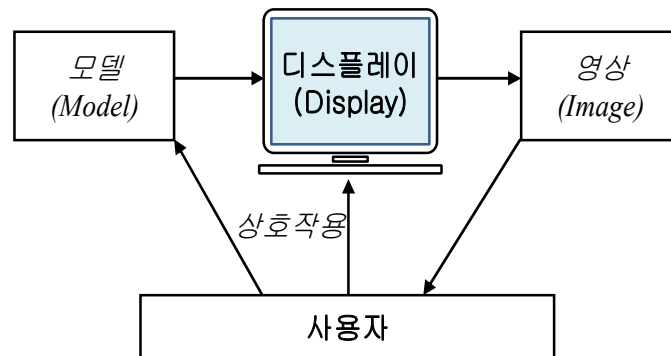
관련 학문 분야



5

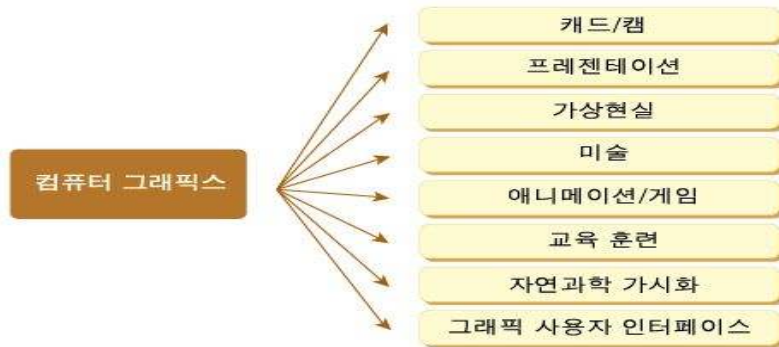
대화형 그래픽스 시스템

- 예) PowerPoint 발표자료 만들기



6

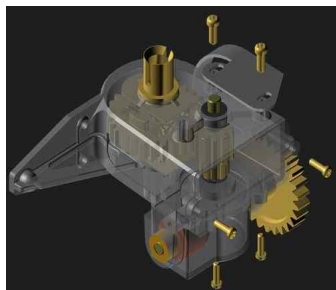
컴퓨터 그래픽스의 응용분야



7

CAD

- Computer-Aided Design
- 설계에 필요한 인력, 시간, 노력 등을 단축함으로써 설계 효율을 향상
- Computer-Aided Manufacturing
 - Numerical Control Machine
 - 자동 생산에 따른 효율과 가공의 정밀도 향상



기계부품 렌더링

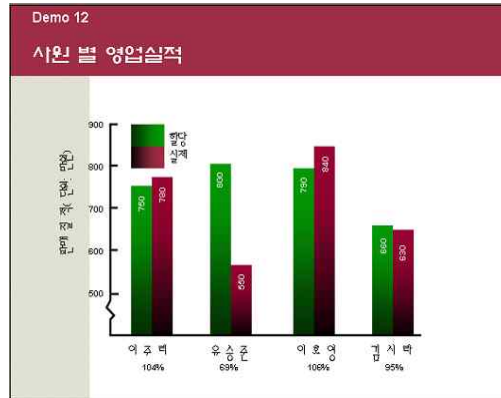


자동차 겔모습 렌더링

8

프레젠테이션 그래픽스

- 막대 차트(Bar Chart), 선 그래프(Line Chart), 파이 차트(Pie Chart), 입체 그래프(Surface Graph)
- 백문(百聞)이 불여일견(不如一見)
- 시선을 붙잡을 것



9

가상, 증강현실

- Virtual Reality
 - “존재하지 않는 가상의 환경을 구성하되 그것이 마치 현실과 똑같이 느껴지도록 만드는 데 주안점”
- 사이버 클래스, 가상환경의 구축, 아바타 생성
- 현실에서 경험하지 못하는 것을 가능하게 하여 인간 경험의 폭을 크게 확장하는 데 기여할 수 있음.
- 교육, 의료 등에 다양하게 활용될 것으로 예상됨



[그림 1-5] 레이 트레이싱에 의한 렌더링

10

가상, 증강현실

- 기술적 요소 *음향, 소리, 거리*
 - 입체화면, 3차원 입체 음향, 데이터 장갑
 - 장면 데이터베이스, 그래픽 소프트웨어
- 인지과학, 전자공학, 기계공학, 음향학



[그림 1-6] HMD *입체화면*



[그림 1-7] 데이터 글로브

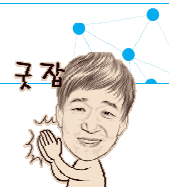


[그림 1-8] 데이터 글러브

11

미술

- 무선 스타일러스 펜
- 그래픽 소프트웨어*



[그림 1-9] 촬영 영상

[그림 1-10] 파스텔 처리



(b) 그래픽스로 재현한 고희의 그림

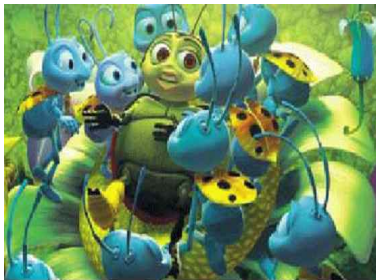
12

애니메이션

- 프레임들의 빠른 연속적인 디스플레이를 말하는데, 영화, 게임, 뮤직 비디오, TV 프로그램 등에 활용됨



- Ants



Ants의 제작에 소요된 자원

소요 내역	양
총 프레임(정지화면) 수	119,592 개
주당 렌더링에 소요된 시간	275,000 시간
평균 정지 프레임 크기	6 MB
렌더링에 사용된 실리콘 그래픽 서버 수	270 대
프로세싱에 사용된 데스크 탑 컴퓨터 수	166 대
프로세서 당 평균 메모리 용량	156MB
1개의 프로세서로 제작할 경우의 소요시간	약 54년
영화 저장을 위한 보조기억 장치 용량	3.2 TB
매 순간은 라인으로 공유된 프레임 수	75,000 개

13

게임

- 캐릭터, 배경화면, 애니메이션
- 2차원 게임에서 3차원 게임으로 발전하고 있음
- 사용자와 프로그램 사이의 상호작용에 대한 설계가 매우 중요함
- 상호작용에 걸리는 시간을 최소화
- 시장성 면에서 볼 때 무한한 가능성



[그림 1-13] 스타크래프트 게임



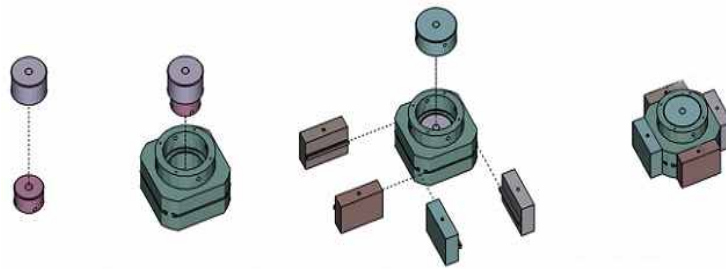
(a) 3차원 게임 "Dead or Alive"

14

교육

- CAI(Computer Aided Instruction)

- 학습 보조도구로서 컴퓨터를 활용
- 텍스트와 함께 시각 정보를 활용하여 교육 훈련의 효과를 높임
- Ex. Exploded View



[그림 1-14] 기계부품의 폭발 조망

15

훈련

- 시뮬레이션



[그림 1-15] 운전 시뮬레이션

[그림 1-16] 비행 시뮬레이션

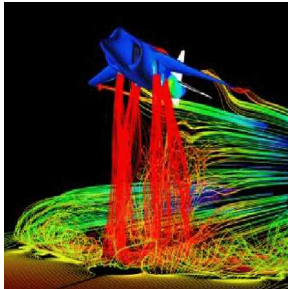


16

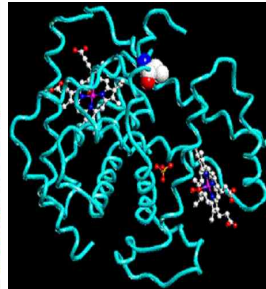
과학분야 가시화

- SCI VIS(Scientific Visualization)

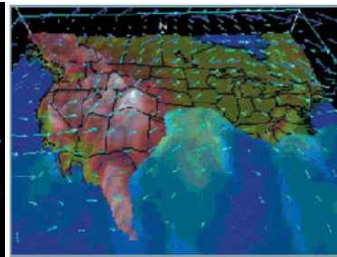
- 대용량 정보분석 **+빅데이터**
- 자연현상을 시각화. 현상 내부의 패턴이나 추세를 직관적으로 파악



[그림 1-17] 기류 분석



[그림 1-18] 분자 구조

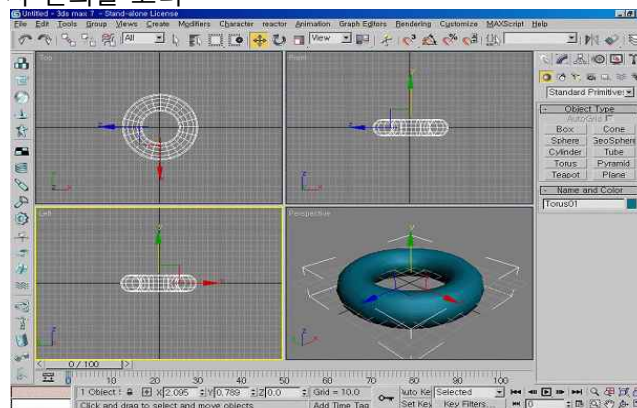


(b) 일기예보에 적용한 그래픽스

17

그래픽 사용자 인터페이스

- 컴퓨터 소프트웨어에서 항상 사용하는 Window, Icons, Menu, scroll bar 등 편리한 사용자 인터페이스를 위한 방법을 제공함
- 사용자 편의를 고려

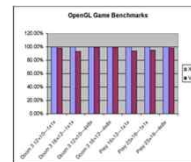
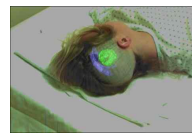
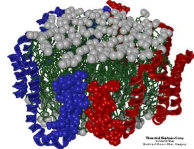
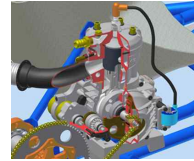


[그림 1-22] 3D 스튜디오 맥스

18

멀티미디어 분야

- 멀티미디어 응용에서 가장 자주 이용하는 매체가 그래픽스이며, 디지털 방송에서 그래픽스 기술의 적극 활용되고 있음



[그림 1-22] 3D 스튜디오 맥스

19

그래픽스 기술의 발전과정

- 1950년대
 - 컴퓨터 그래픽스의 태동기로 수동적인 컴퓨터 그래픽스가 주류
 - 문자를 이용해 원하는 그림을 그리고 라인프린터와 같은 출력장치를 이용하여 단순히 출력하는 수준
 - Line printer, Teletype
 - CRT Monitor (MIT의 Whirlwind 컴퓨터)

20

1960년대

1960	William Fetter	“컴퓨터 그래픽”이란 용어를 최초로 사용
1963	Ivan Sutherland	컴퓨터 그래픽의 제반개념을 확립
1963	Douglas Englebart	최초의 마우스 프로토타입
1965	Jack Bresenham	선분 그리기 알고리즘을 개발

- 그래픽스 산업의 태동기
- Ivan Sutherland의 drawing system (MIT, 1963)
 - CRT와 Light pen 사용했으며, 현대 대화형 그래픽스의 시초
- DVST(Direct View Storage Tube)를 이용한 CAD시스템 (1960년대 말)
- Tektronix 4010

21

1960년대

- 이반 서더랜드(Ivan Sutherland)
 - 컴퓨터 그래픽의 창시자
 - 대화형 컴퓨터 그래픽 개념: 라이트 펜으로 의사전달
 - 최초의 상호대화식 그래픽스 시스템 스케치패드(1963년)
 - 직선, 원호 등 기본적 그래픽 요소를 사용하여 물체를 표현하는 방법
 - 기본물체를 조합하여 큰 물체를 모델링한다는 계층구조 모델링
 - 물체를 선택하여 이동하는 방법
 - 팝업 메뉴에 의한 사용자 입력



22

1970년대

• 그래픽 알고리즘의 시기

[표 1-3] 1970년대 사건

1971	Gouraud	구로 셰이딩 알고리즘
1973	John Whitney Jr.	컴퓨터 그래픽에 의한 최초의 영화 "West World"
1974	Edwin Catmuff	텍스처 매핑, 지-버퍼 알고리즘
1974	Bui-Tong Phong	전반사에 의한 하이라이트 알고리즘
1975	Martin Newell	베지어 표면 메쉬를 사용한 차 주전자 모델
1975	Benoit Mandelbrot	프랙탈 이론
1976	Jim Blinn	주변 매핑, 범프 매핑 이론
1977	Steve Wozniak	컬러 그래픽 PC: Apple II
1977	Frank Crow	앤티 에일리어싱 알고리즘
1979	Kay, Greenberg	최초로 투명한 물체 면을 그려냄

- 벡터 그래픽스 장치가 주종인 시기
- CAD 시스템이 사용되기 시작 : 예) 비행기 제조회사
- Raster Display Monitor(Raster-Scan Monitor)의 개발

23

1980년대

- PC의 시대,
- 래스터 그래픽 하드웨어
- 기하 엔진 출현



[그림 1-24] Luxo Junior



[그림 1-25] Tin Toy

1980	Turner, Whitted	광선 추적 알고리즘
1982	Steven Lisberger	3차원 그래픽 애니메이션 "Tron"
1982	John Walkner, Dan Drake	"AutoCAD"
1983	Jaron Lanier	데이터 장갑을 사용한 가상현실 영화
1985	Pixar	"Luxo Junior"
1985	NES	가정용 게임 "Nintendo"
1986	Steve Jobs	Lucasfilm사의 Pixar 그래픽 그룹을 인수
1987	IBM	VGA 그래픽 카드
1989	IBM	SVGA 그래픽 카드
1989	Pixar	"Tin Toy" 아카데미상 수상

[표 1-4] 1980년대 사건

24

- 래스터 그래픽스가 일반화 됨 (Raster-Scan Monitor)
 - Mac, GKS(1985), PHIGS(1988)
- Special purpose hardware 개발
 - Silicon Graphics geometry engine
 - VLSI implementation of graphics pipeline
- Color Graphics와 광선 추적법(Ray Tracing)이 제안됨
- 다양한 HCI(Human-Computer Interaction) 기술이 발전
 - Window, Menu, Icon, Mouse, 등
- Graphics Art가 나타남.
- 출판업에 WYSIWYG (What You See Is What You Get) 개념이 나타남

25

1990년대

- 사실적(Photo-realistic) 그래픽 영상에 주력



[그림 1-26] NVIDIA GeForce 256

1990	Pixar: Hanrahan, Lawson	렌더링 소프트웨어 "Renderman" 개발
1990	Gary Yost	3-D Studio 개발
1991	Disney and Pixar	"Beauty and the Beast"
1992	Silicon Graphics	OpenGL 사양 발표
1993	Steven Spielberg	"Jurassic Park"
1995	Pixar	"Toy Story"
1995	Microsoft	DirectX API 사양 발표
1996	John Carmack, Michael Abrash	Quake 그래픽 엔진 개발
1999	NVIDIA	GeForce 256 GPU

[표 1-5] 1990년대 사건

26

- 영화, 광고, 게임, 시뮬레이션 등 다양한 분야에 활발히 활용됨
- 3차원 그래픽 가속기(Graphics Accelerator)
 - - Texture mapping, Blending, Accumulation, stencil buffers 등
 - - Parallel Processor, Pipe-line Architecture
- 3D 그래픽스가 발전하고 3D UI 기술이 발전함
- 가상현실의 등장 및 인터넷 시대의 시작(VRML)



[그림 1-26] NVIDIA GeForce 256

27

2000년대

- 고속, 고품질의 렌더링이 가능해 지며, 실시간 렌더링 기술이 발전함
- 물체의 사실감과 자연스러움을 증가시키는 그래픽스 기술이 발전
- 비사실적 렌더링(Non-Photorealistic rendering) 기술이 활용됨
- 모바일 환경 및 무선 환경: PDA, 휴대폰, 스마트폰 등
- 가상환경에서의 3차원 아바타 활용이 활발해 짐
- 증강현실 기술의 활용

28

2000년대

[표 1-6] 2000년대 사건

2001	Square	"Final Fantasy: The Spirits Within"
2003	Timothy Purcell	광선추적 기법을 GPU에 적용
2004	Id Software	Doom Engine 발표
2004	DirectX, OpenGL	New Version 발표



[그림 1-27] Doom

29

2010년~현재

- Mobile Computing: iPhone
- Cloud Computing: Amazon Web Services (AWS)
- Virtual Reality: Oculus Rift
- Artificial Intelligence
 - Big Data/Deep Learning
 - Google Car

30

1.2 모델링과 렌더링



- 모델링과 렌더링 **3차원**
 - 그래픽스를 이용해 영상을 만드는 과정
 - 다양한 모델링 기법들
 - 3D 그래픽 파이프라인과 영상이 만들어지는 과정
 - 2차원 그래픽스

31

3D 그래픽스를 이용해 영상을 만드는 과정



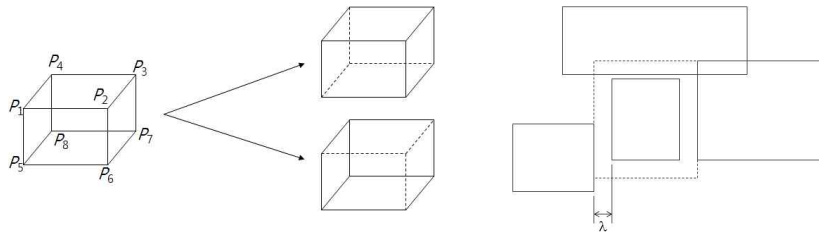
- 모델링(Modeling)
 - 물체의 형태를 3차원 좌표계에서 표현하는 과정
 - 일반적으로 **고도로 대화적**이며, **상세한 이미지 작업 필요하지 않**으며, 보통 **그래픽 워크스테이션**에서 실행됨 **렌더링**
 - **3차원 스캔**에 의해서도 모델링이 가능함
 - Wireframe 모델, Polygon(Surface) 모델, Solid 모델 등
+ **스캐너**
- 렌더링(Rendering)
 - 3차원 객체를 2차원 화면에 투영하고 투영된 물체의 면에 색상과 명암의 변화와 같은 **3차원적인 질감을 더해 현실감을 추가**하는 과정
 - **방대한 계산 과정이 필요**. 계산 전용 컴퓨터에서 실행됨
 - Hidden Surface Removal, Shading, Texture Mapping, Shadow, Ray Tracing 등

32

(1) Wire-frame Model

바다의 표현 (점과 선)

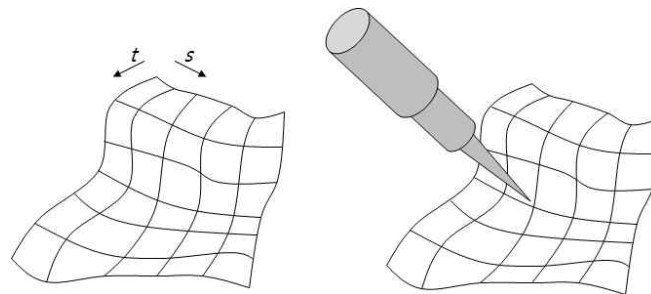
- 단순하고 효율적이지만 모호성
- 객체의 정확한 형태를 이해하기가 혼란



33

(2) Surface Model

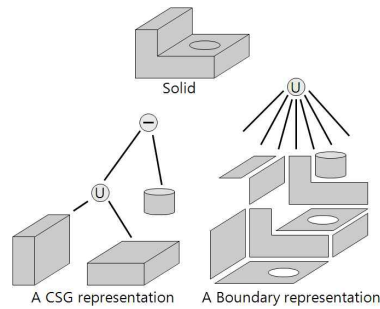
- 점, 선과 함께 표면 조각이 기본 요소에 추가
- 대부분의 3차원 모델들은 최종적으로 표면 모델로 변환되어 렌더링
- 내부/외부를 표현할 수 없으므로 물체의 완전한 표현이라고 볼 수는 없다



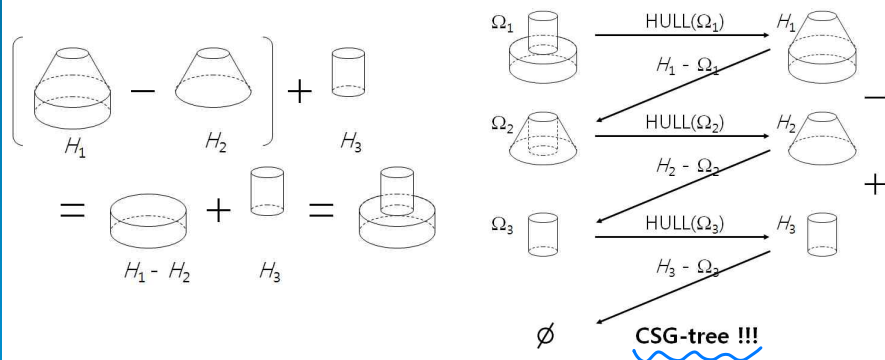
34

(3) Solid Model

- 솔리드 모델은 물체의 내부와 외부의 구분이 가능
- CSG(Constructive Solid Geometry)
- 체적을 가진 기본 원소들을 더하거나 빼는 등의 연산을 통해 모델을 표현
- 표면 모델로 변환해야 최종적으로 렌더링할 수 있다.



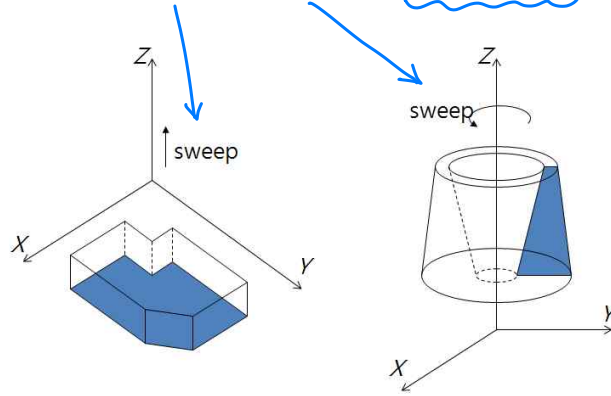
35



36

(4) Swept Volume

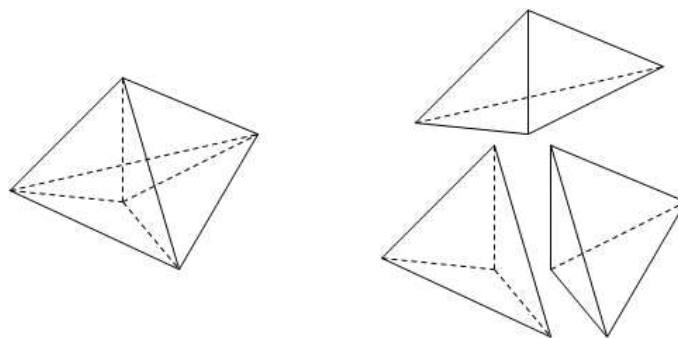
- 기본 원형을 직선이나 원형으로 쓸어서 볼륨을 표현



37

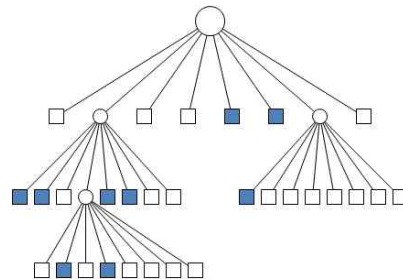
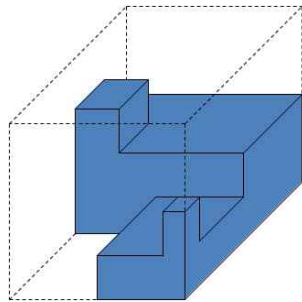
(5) Object Decomposition

- 객체를 단위 사면체나 육면체 등의 기본 원형들로 분해 (decomposition)하여 표현



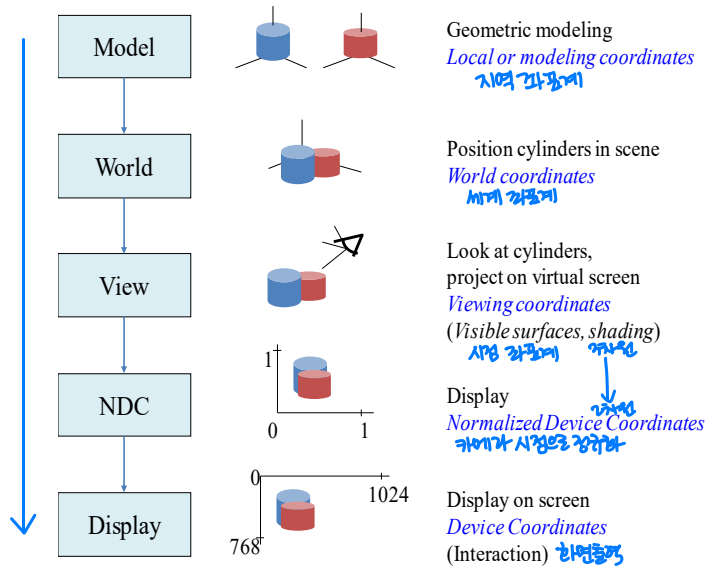
38

(6) Octree



39

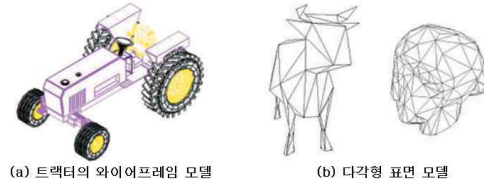
그래픽 파이프라인



40

3차원 그래픽 생성과정

- 물체의 기하학적인 형상을 모델링(Modeling)

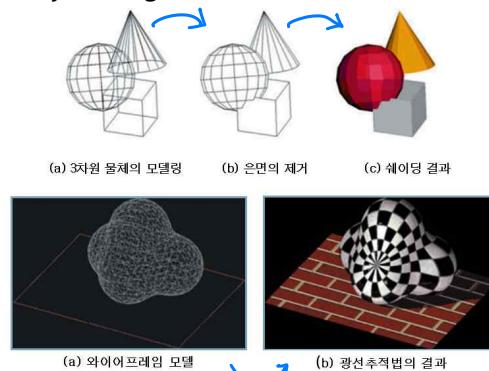


- 3차원 물체를 2차원 평면에 투영(Projection)
 - 평행 투영법과 원근 투영법
- 생성된 3차원 물체에 색상과 명암을 부여(Rendering)

41

Rendering

- 색상과 명암의 변화와 같은 3차원적인 질감을 더하여 현실감을 추가하는 과정
- 은면의 제거(Hidden Surface Removal)
- 셰이딩(Shading), 텍스처 매핑 (Texture Mapping), 그림자(Shadow)
- 광선추적법(Ray Tracing)



42

2차원 그래픽스

- 점, 선, 원, 곡선 등과 같은 기본 도형을 이용하여 2차원 평면상에 그림
- 결과물을 픽셀의 형태로 표현: 각 픽셀은 적색(Red), 녹색(Green), 청색(Blue)의 농도 값을 배합
- 벡터 그래픽(Vector Graphics)
- 래스터 그래픽(Raster Graphics)



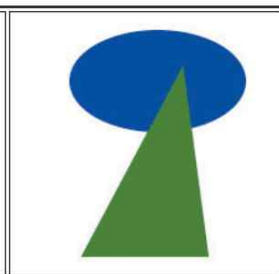
[시연] 래스터 그래픽과 벡터 그래픽의 비교

43

벡터 그래픽(Vector Graphics)

- 그래픽에 사용된 객체들을 수학적 함수로 표현하여 기억 공간에 저장하는 방식
- 파일의 크기가 래스터 그래픽 방식으로 저장한 것보다 작음
- 기하적 객체를 수식의 형태로 표현하므로 화면 확대시에도 화질의 변화가 없음 ex) 폰트 활용

```
<?xml version = "1.0"?>
<svg width="200" height="200">
  <ellipse cx="110" cy="50" rx="70"
  ry="40">
    style="fill:blue; stroke:blue; stroke-
    width:2"/
  <polygon style="fill:green;stroke:green;"
  stroke-width:2" points="130,40 140,190
  50,190"
  />
</svg>
```

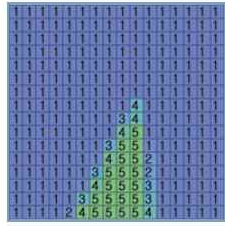


SVG로 작성된 그래픽스 명령문과 그 수행결과

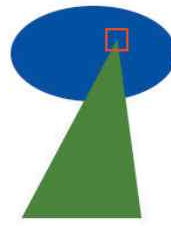
44

래스터 그래픽(Raster Graphics)

- 래스터 그래픽 출력장치에 표시하기 위한 그래픽 데이터를 픽셀단위로 기억 공간에 저장
- 저장된 파일의 크기는 출력장치의 해상도에 비례
- 화면을 확대하면 화질이 떨어짐 $\text{파일 크기} \propto \text{해상도}$



(a) 메모리에 저장된 픽셀 값(붉은 사각형 내)



(b) 화면에 그려진 결과물

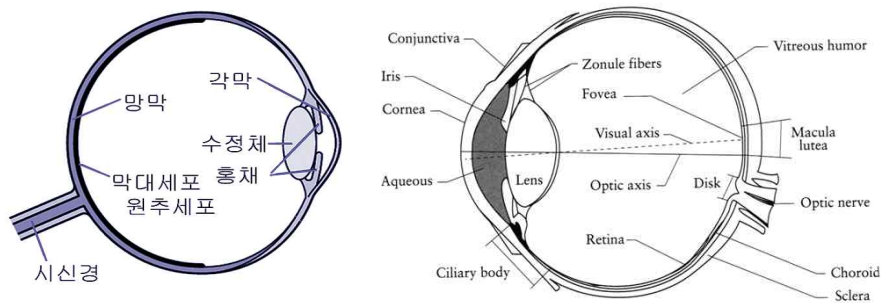
45

1.3 인간의 시각과 색

- 인간의 시각 시스템
- 가시광선의 이해

46

인간의 시각 시스템

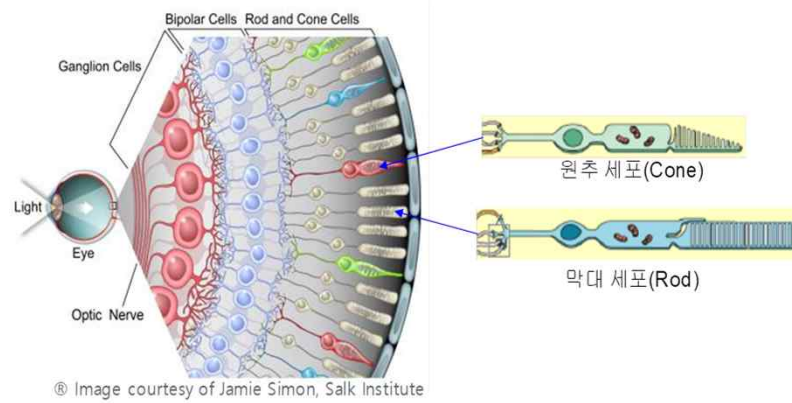


47

망막 빛을 감지하기 세포

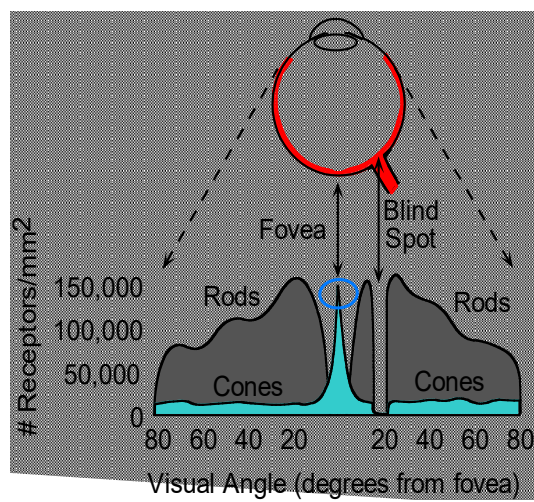
- Cones (원추세포)
 - 원추형으로 생겼으며 색(color)에 민감하게 반응함(color vision)
 - 상대적으로 덜 민감하여 어두운 곳에서는 구분이 어려움 밝은 곳에서 시각을 담당함
- Rods (막대세포) *4장은 막대세포, 원추세포보다 더 많다*
 - 막대와 같이 생겼으며 밝기(intensity)에 민감하게 반응함(gray-vision)
 - 매우 민감하며 어두운 곳에서도 민감하게 반응하여 야간 시각을 담당함. 어두운 곳에서 물체의 색은 감지가 안되지만 형태는 잘 파악할 수 있음.

48



49

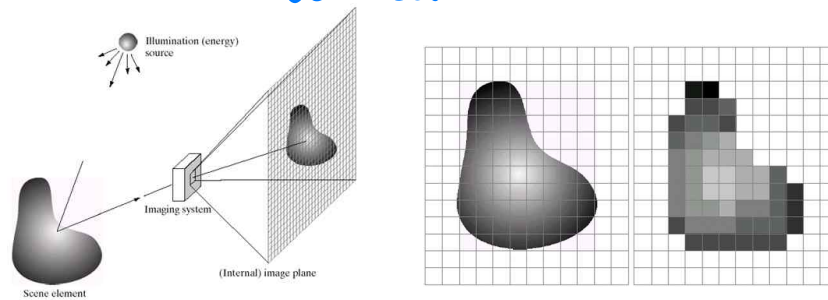
세포들의 분포



50

디지털 카메라 모델

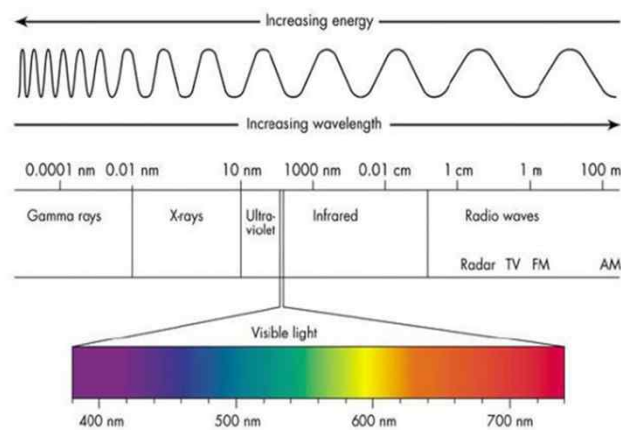
- 망막이 센서 배열로 대체
- 센서의 종류
 - CCD(Charge Coupled Device) **노이즈↓** **속도↓** **비용↑**
 - CMOS **해상도↑** **잡음↑**
- 최종적으로 상이 가로와 세로 방향으로 샘플링



51

가시광선의 이해 사람으로 볼수있는 영역 전자기파의 일종

- 전자기파의 파장별 분류와 가시광선 대역

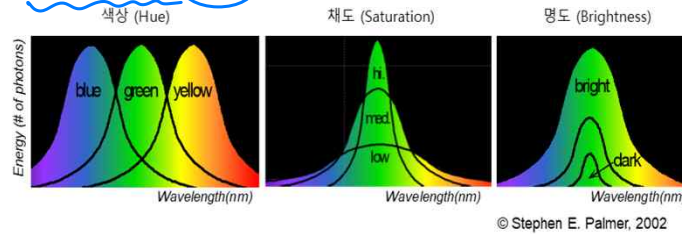


- 가시광선의 파장: 390nm – 720nm

52

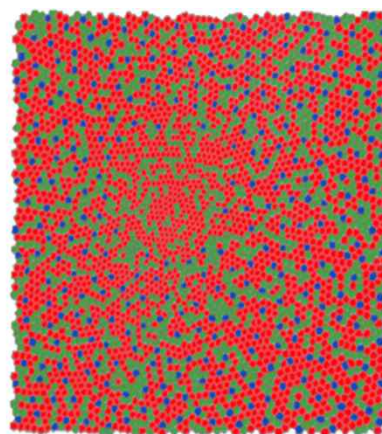
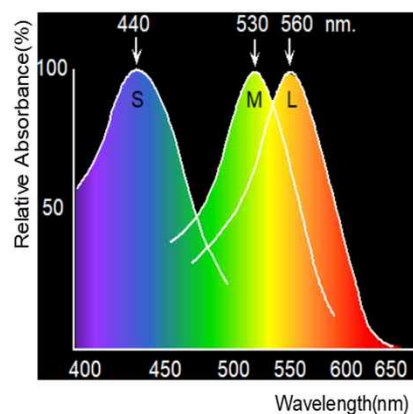
색상, 명도, 채도 차이점 시험

- 색상(Hue)
 - 전자기파 에너지의 평균(mean)을 나타내는 주파수
 - 왼쪽 그림은 blue, green, yellow의 에너지 분포의 예
- 채도
 - 에너지 분포의 분산(variation)
 - 분산이 ↑면 채도가 ↓고, 분산이 ↓이면 채도가 ↑ 높아진다.
- 명도(Brightness)
 - 에너지 분포의 면적이 넓으면 밝은 빛



53

세 종류의 원추세포와 이들의 분포



© Stephen E. Palmer, 2002

54

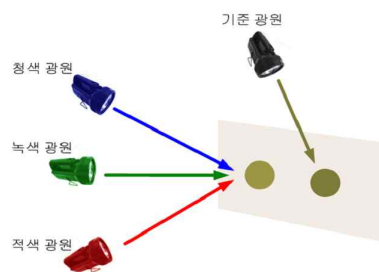
1.4 컬러 모델

- CIE, 색범주
- RGB, CMY, CMYK, YUV, ...
- RGB 컬러모드/인덱스 컬러모드
- 하프 토크닝, 디더링

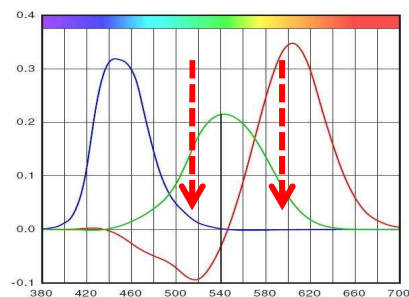
55

컬러 매칭

- 580nm 황색 = 적색(0.25) + 녹색(0.13) + 청색광원(0.0)
- 500nm 근처: 적색은 음의 값??
 - G, B를 합성한 색상에서 적색 성분을 빼야 함.
 - 현실적으로 불가능 → RGB로 모든 자연색 표현 불가



[그림 3-5] 컬러매칭

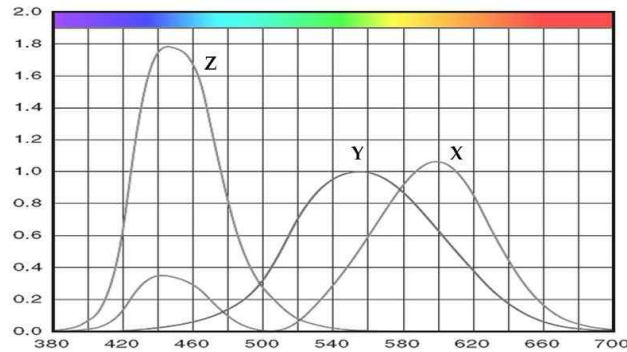


[그림 3-6] 컬러매칭 함수

56

CIE 컬러 모델

- Commission Internationale d'Eclairage (1931년)
 - X, Y, Z: 가상의 삼원색, 수학적으로 유도, 모두 양의 함수
 - Y = 명도. X, Z가 색을 결정



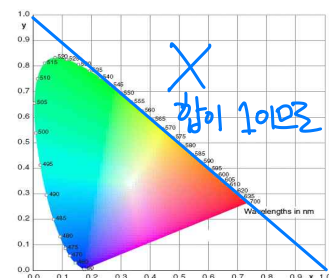
[그림 3-7] CIE 가상 삼원색

57

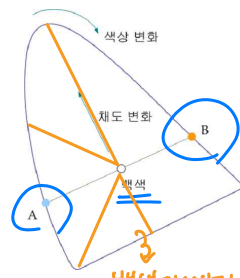
CIE 정규화

- $x+y+z = 1$ 이 되도록 x, y, z 를 설정

$$x = X / (X + Y + Z) \quad y = Y / (X + Y + Z) \quad z = Z / (X + Y + Z)$$
- x, y 가 결정되면 z 는 자동으로 결정됨: x, y 의 함수로서 색을 표현
 - CIE 색 범주
 - 순색(단일 파장): 경계선, 내부색: 순색의 혼합, 보색: 예. A와 B

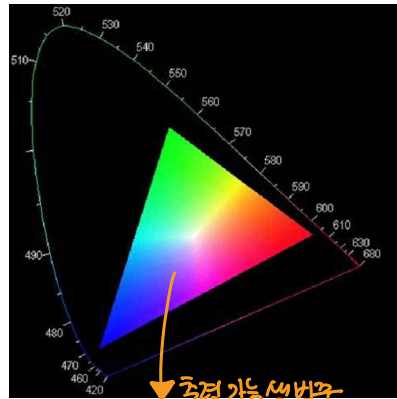


CIE Color Gamut(색 범주) 사람의 눈에 보이는 색

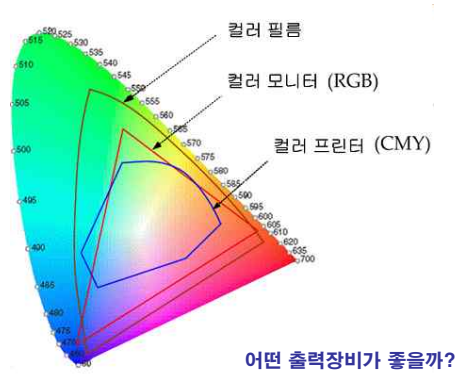


백색에서부터 일직선은 모두 같은색이고 채도만 변화

색 범위(Color Gamut)



[그림 3-10] 컬러 모니터 색 범주



[그림 3-11] 출력 장비별 색 범주

59

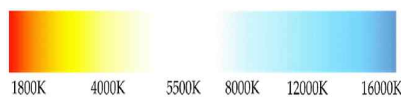
색 온도(Color Temperature)

- 가열된 물체의 온도에 의한 색 표현
 - 이상적인 흑체 (完全黑體)가 방출하는 빛의 색은 플랑크의 복사법칙에 의해 온도에 의해서만 정해짐

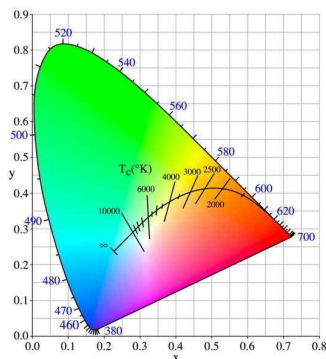
[표 3-1] 온도별 색

1200 K	촛불	6000 K	밝은 대낮 태양
2800 K	전구, 일출과 일몰	7000 K	약간 흐린 하늘
3000 K	스튜디오 램프	8000 K	흐린 하늘
5000 K	평균 태양광	10,000 K	매우 흐린 하늘

가장 뜨거운



[그림 3-12] 색 온도 I



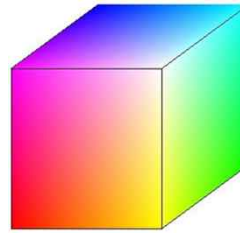
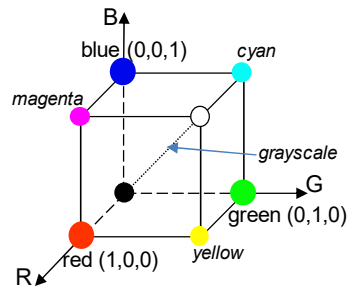
[그림 3-13] 색 온도 II

60

RGB 컬러모델

삼중 자극이론(Tri-Stimulus Theory)

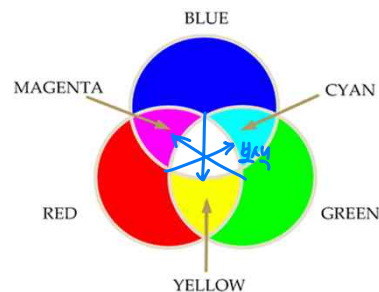
- 원추세포는 파장 630nm(빨강), 530 nm(녹색), 450nm(청색)에 가장 민감하게 반응



61

가산모델 ⊕

- 빛의 합성(예: 모니터)
- $R+G = Y$, $G+B = C$, $B+R = M$
- RGB의 보색은 CMY

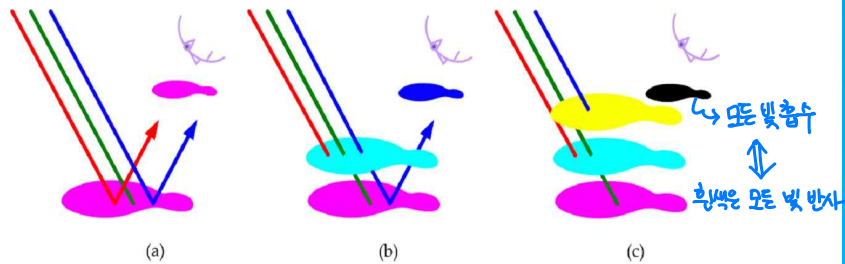


62

CMY 컬러모델

• 감산모델 ⊖

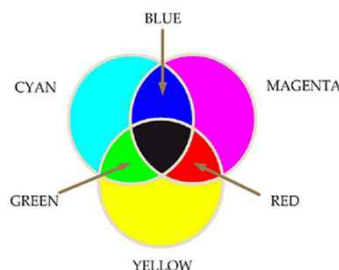
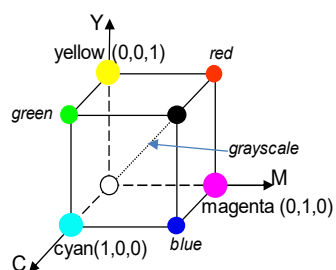
- 물감의 합성 (예: 프린터) *CMY*
- $W - G(\text{Magenta의 보색}) = R + B = \text{Magenta}$
- $W - B(\text{Yellow의 보색}) = R + G = \text{Yellow}$
- $W - R(\text{Cyan의 보색}) = G + B = \text{Cyan}$



[그림 3-18] 감산모델

63

- RGB는 CMY의 합성으로 표현
- $C + M + Y = \text{Black}$ cf. $R + G + B$
- 물감의 삼원색은 **CMY**. 빨강노랑파랑이 아님.



64

CMYK 컬러모델

컴퓨터용 색재현력

→ key=black

- K for Chromatic (회색농도)
 - C = 120, M = 80, Y = 200
 - C = 40, M = 0, Y = 120, K = 80
- 장점
 - 잉크 건조시간 ↓ 잉크 비용 ↓
 - 정밀한 회색농도를 표현(완벽한 보색차단)

65

HSV 컬러모델 푸른인덱스에 누런 선하

컬러영상처리에 사용

- RGB 모델의 단점
 - 직관적이지 않다. 보라색 = R, G, B 각각 얼마?
쉬이 힘들
- HSV(Hue, Saturation, Value)
 - 또는 HSB(Hue, Saturation, Brightness)
 - 색상(Hue), 채도(Saturation), 명도(Value, Brightness)
 - 화가의 직관
 - 셰이드 = 어떤 색상에 흑색을 섞음. 채도와 명도를 동시에 낮춤.
 - 틴트 = 어떤 색상에 백색을 섞음. 채도는 낮추고 명도는 높임.



RGB는 각각 색 표현 방법

Blue Screen ?



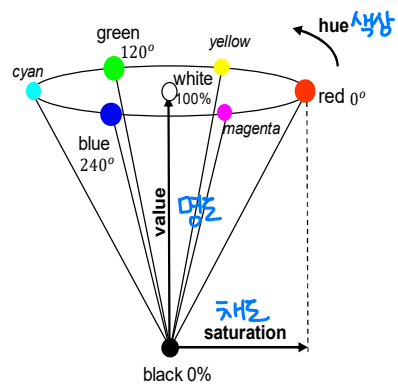
[그림 3-21] 셰이드와 틴트

66

• 육각별

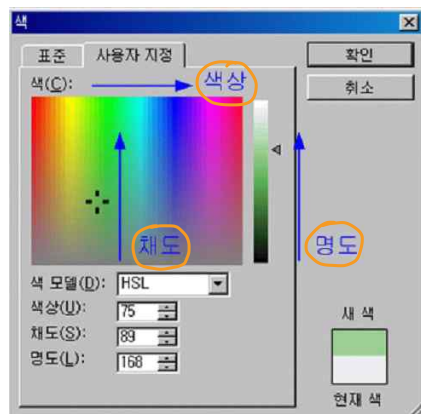
H	S	V	색
0	1.0	1.0	Red
120	1.0	1.0	Green
240	1.0	1.0	Blue
	0.0	1.0	White
		0.0	Black
90	0.5	0.25	

[표 3-2] HSV 컬러 좌표

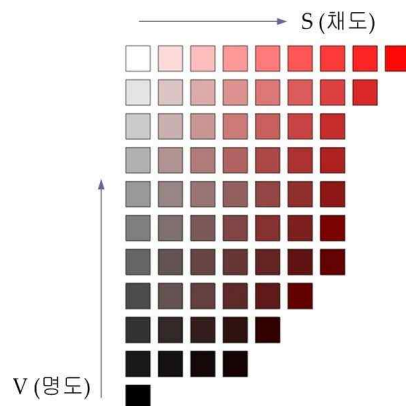


67

• 대화상자 예시, 적색 단면 예시



[그림 3-23] HSV 대화상자

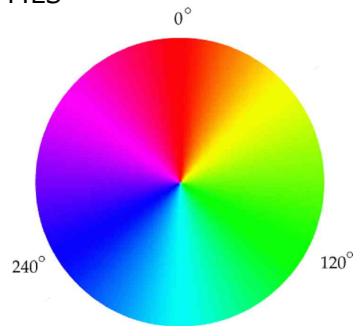


[그림 3-24] HSV 단면

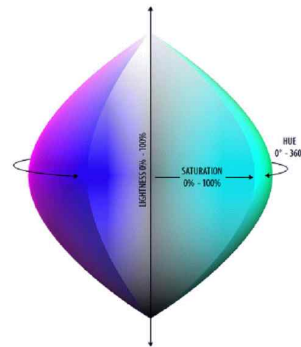
68

HSV 컬러모델의 변형

- 원뿔형 HSV
 - 색상을 둥글게 배치
- HLS



[그림 3-25] HSV 모델 I



[그림 3-26] HLS 모델 II

69

YUV, YCbCr, YIQ 통신, 압축, 방송

- 컬러 TV의 흑백 TV 호환성

YUV (명도, 방향)

$$\begin{aligned}
 Y &= 0.213R + 0.715G + 0.072B \\
 U &= 0.299R + 0.587G + 0.114B \\
 V &= 0.492(B - Y) \quad V = 0.877(R - Y)
 \end{aligned}$$

명도
색(색상, 채도)

- 디지털 TV

YCbCr

$$\begin{aligned}
 Y &= Y \\
 Cb &= (B - Y)/1.772 + 0.5 \\
 Cr &= (R - Y)/0.402 + 0.5
 \end{aligned}$$

- NTSC TV 표준

YIQ

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.11)$$

- I는 주황-청색(Orange-Blue), Q는 자주-녹색(Purple-Green)축
- Y, I, Q 순서대로 민감도가 낮아짐

70

• 흑백 TV에서 컬러 TV로의 호환성

- 기존의 밝기(Y) 성분에 대한 신호는 그대로 두고 추가로 색상정보를 다른 주파수 대역으로 전송
- 흑백 TV는 그대로 사용, 컬러 TV는 추가된 컬러 신호를 복원

• 인간 시각의 특성

- 영상 압축 등 통신을 위해 사용
- 사람은 색상보다 밝기에 훨씬 민감 (막대 세포가 훨씬 많음)
- Y 신호는 그대로 사용하고 색상신호는 줄여 사용
- 예) 밝기(Y)와 색상(U, V)을 4:1:1 등의 비율로 사용
4 1 1 6byte \Rightarrow 데이터가 반으로 줄음

71

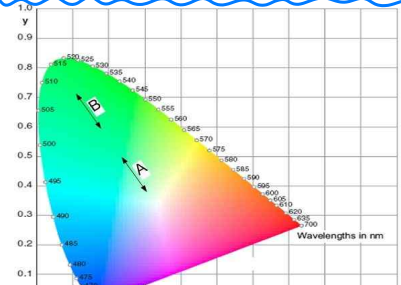
CIE L*a*b*

• 웨버의 법칙

- 자극이 강할수록 상대적 감도는 낮아짐
- 절대 명도 I일 때, 인지된 명도는 $\log(I)$ 에 비례

• CIE 모델의 문제점

- 인지된 색차가 그림의 거리에 비례하지 않음 인지된 색차 \propto 그림의 거리



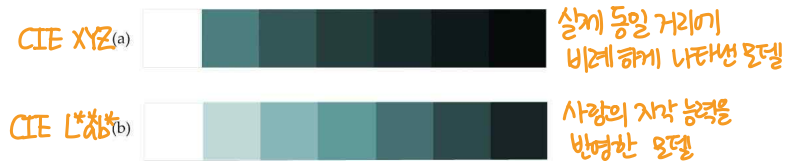
인간의 지각 능력을 반영 ?

[그림 3-29] CIE 모델에서의 색차

72

CIE L*a*b*

- CIE의 변형
 - 인지 컬러모델(Perceptual Color Model)
 - 사람의 지각 능력을 반영한 컬러 모델
 - 인지된 색차가 맵상의 거리에 비례하도록
- CIE XYZ와 CIE L*a*b*와의 비교

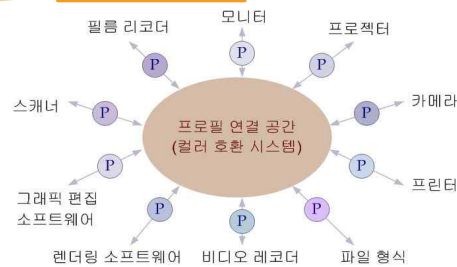


[그림 3-30] CIE 모델과 CIE L*a*b* 모델

73

CMS(Color Management System)

- 동일한 컬러 모델을 사용하더라도 출력 장비에 따라 보이는 색이 달라짐. Why?
- 컬러 호환성 CMS
 - 컬러모델, 장비특성에 따른 오차를 최소화
 - 개별 장비의 컬러모델, 장비특성을 프로파일 형태로 나타냄
 - 장비에 무관한 컬러공간으로 사상

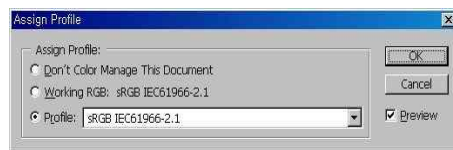


[그림 3-33] 컬러처리 시스템

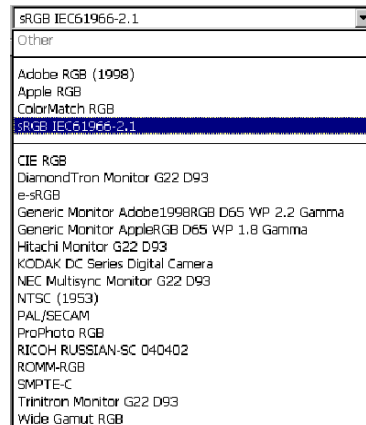
74

프로필 연결공간

- 일종의 세계어에 해당
 - 장비 프로필 정보를 감안
 - sRGB(Standard RGB) 또는 CIE L*a*b*으로 사상



[그림 3-34] 포토샵의 프로파일 타입 할당



[그림 3-35] 프로파일 타입

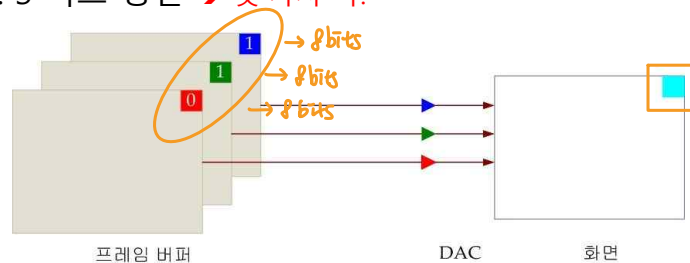
75

RGB 컬러모드 → 요즘은 대부분 이것 사용

- 프레임 버퍼 내용이 색을 결정
 - 용량에 따라 색 종류가 결정됨
 - 하이컬러, 풀컬러, 트루컬러, ...

→ 8bits x 3 = 24bits

- 예: 3 비트 평면 → 몇 가지 색?



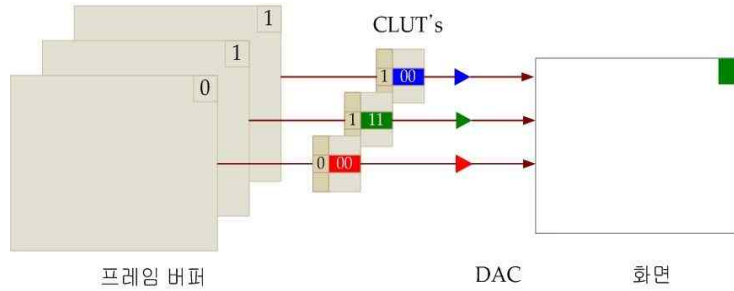
[그림 3-36] RGB 컬러모드

76

인덱스 컬러모드

• 컬러 번역기

- 프레임 버퍼 내용은 CLUT의 인덱스
- 실제 컬러는 CLUT의 우측 칼럼에 의해 결정



[그림 3-37] 인덱스 컬러모드 I

77

Color lookup table CLUT

- 한 화면에 보일 수 있는 컬러는 여전히 8가지
- 그 컬러가 어디서 왔는가의 문제

프레임버퍼(RGB)	보기표(R)	보기표(G)	보기표(B)
000(0)	00011100	11000000	10010011
001(1)	11001001	00010100	01011100
010(2)	10010000	10010011	00010101
011(3)	00110001	00111001	00110000
100(4)	11110101	01010011	11001111
101(5)	01011000	10110100	10110101
110(6)	00100011	01010101	01011100
111(7)	10111100	11111100	11111001

[표 3-3] 컬러 보기표의 예

78

Color palette



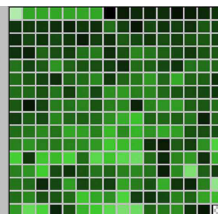
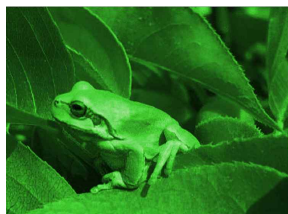
[그림 3-39] 인덱스 컬러모드



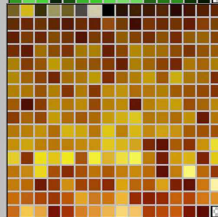
[그림 3-40] 팔레트

79

- PNG, BMP, TGA, TIFF
 - 파일 내부에 팔레트 정보를 포함



[그림 3-43] 녹색 위주 팔레트



[그림 3-44] 금색 위주 팔레트

80

하프 토투닝 출력장치 한계 극복

- 우리 눈의 종합적 인식능력을 이용
- 인쇄물의 하프토투닝
 - 흑백이지만 회색처럼 보임
 - 화면 화소에는 이것이 불가능



(a)



(b)

출력장치의 한계를 극복

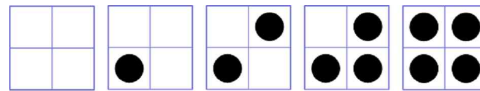
[그림 3-45] 인쇄물의 하프 토투닝

아날로그 하프 토투닝
화면(모니터)에서는???

81

디지털 하프 토투닝

- 점의 크기 대신 개수를 조절
 - 우리 눈이 4개의 화소를 한 단위 평균적 밝기를 인식
 - 실질적 해상도는 감소



(a)

(b)

(c)

(d)

(e)

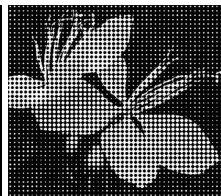
[그림 3-46] 흑백 모니터의 하프 토투닝



[그림 3-47] 원 영상



[그림 3-48] 흑백영상



[그림 3-49] 하프톤

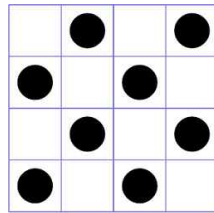
82

- 하프톤 행렬

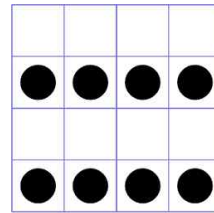
$$\text{Halftone Pattern 1} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$

$$\text{Halftone Pattern 2} = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}$$

- 줄 무늬에 유의



(a)



(b)

[그림 3-52] 하프 톤 패턴 적용

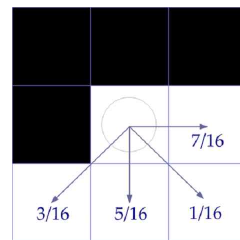
83

디더링

하프 톤 패턴의 차이점

- ➔ 해상도 감소 없음 (동일 해상도 유지)

- 화소에서 화소로의 사상
- Ex. 오류 확산(Error Diffusion Dither)



[그림 3-53] 오류 확산 방법

- Ex. 패턴 디더(Patterned Dither)
 - 모든 화소의 회색도를 최대 8로 정규화
 - 정규화 값이 행렬 값보다 크면 화소에 가장 가까운 색으로, 작으면 배경색으로

$$\text{Dither Pattern} = \begin{bmatrix} 6 & 8 & 4 \\ 1 & 0 & 3 \\ 5 & 2 & 7 \end{bmatrix}$$

84

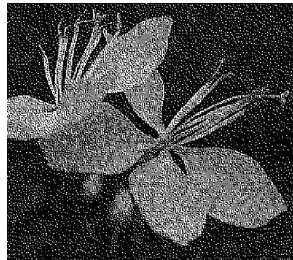
오류확산, 패턴 디더

• 디더링

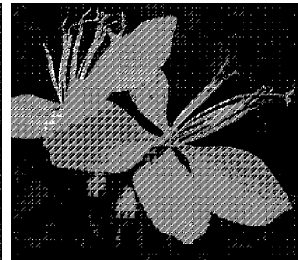
- 오류확산: 오류를 전파
- 패턴디더: 인위적인 잡음 삽입



[그림 3-54] 원영상



[그림 3-55] 오류확산



[그림 3-56] 패턴 디더링

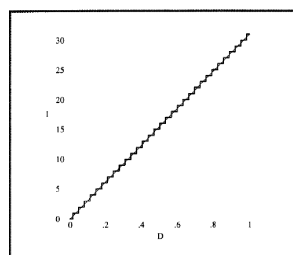
85

감마수정

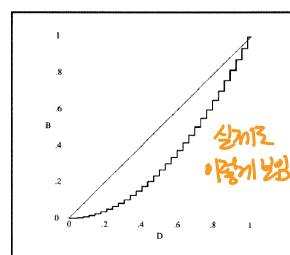
- 인점의 밝기는 전자빔의 밝기에 비 선형적으로 반응
 - (0.255)에서 128은 중간 회색도가 아님
 - 인점 밝기와 회색도와의 관계

$$B = \alpha \delta^\gamma$$

- 알파: 비례상수, 델타: 정규화 회색도, 감마: 1이 아니고 (1.7-2.8)



[그림 3-57] 전자빔/회색도

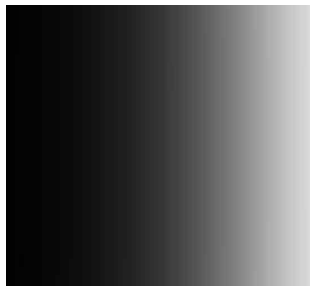


[그림 3-58] 밝기/전자빔

86

감마수정의 필요성

- 프로그래머의 기대치 대 실제 밝기
 - 회색도 128은 실제로 더욱 어둡게 나타남



[그림 3-59] 기대값



[그림 3-60] 실제값

실제로 어둡게 보임

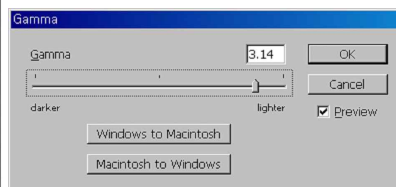
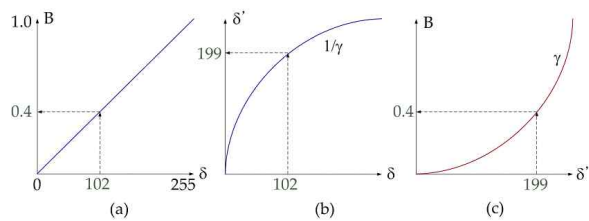
$$\bullet V_C \sim V_S^{(1/\gamma)}$$

- CRT의 비선형성
- 인간 시각의 비선형성

87

감마수정

$$\begin{aligned} \bullet \therefore B &= \alpha (\delta')^\gamma \\ &= \alpha (\delta^{1/\gamma})^\gamma \\ &= \alpha (\delta) \end{aligned}$$



88

1장 연습문제



89



감사합니다!

90