

# Graphics Systems and Models

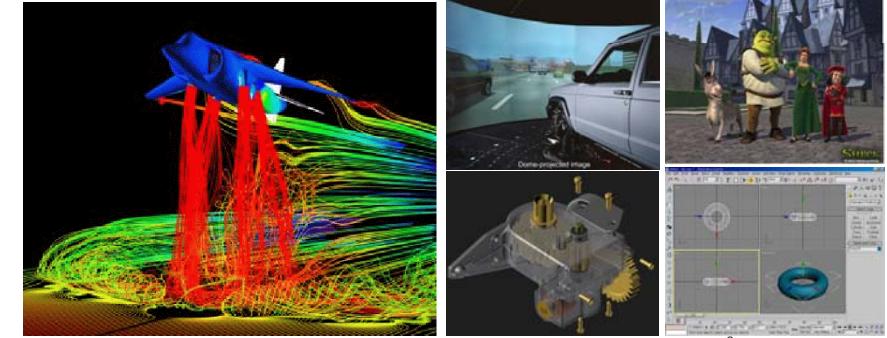
321190  
2013년 봄학기  
3/7/2013  
박경신

## Computer Graphics Main Theme

- ▣ 이미지 (Imaging)
  - 2차원 이미지를 효과적으로 표현
- ▣ 모델링 (Geometric Modeling)
  - 가상 3차원 물체를 효과적으로 표현
- ▣ 렌더링 (Rendering)
  - 3차원모델에서 2차원 이미지로 사진과 같이 사실적으로 표현
- ▣ 애니메이션 (Animation)
  - 시간에 따른 움직임을 자연스럽게 표현

## Computer Graphics Applications

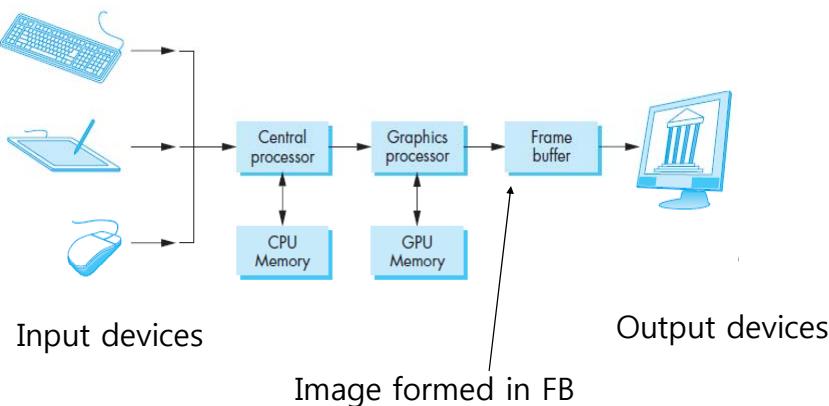
- ▣ 정보의 표시 (Information Visualization)
- ▣ 설계 (Design)
- ▣ 시뮬레이션 (Simulation), 애니메이션 (Animation)
- ▣ 사용자 인터페이스 (User Interfaces)



## A Graphics System

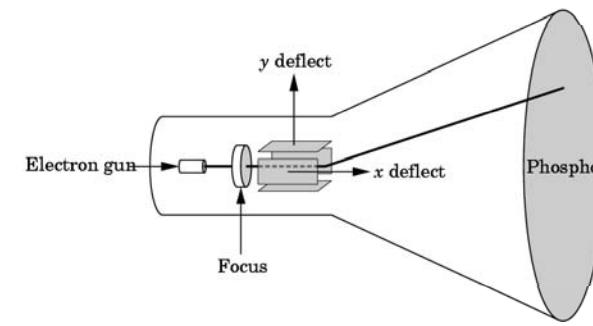
- ▣ PC
  - CPU
  - Memory
  - Disk
  - Graphics card
  - Sound card
  - Peripherals: keyboard, mouse, monitor
  - Network
- ▣ Graphics card
  - GPU
  - Memory
  - Video output

## A Graphics System



5

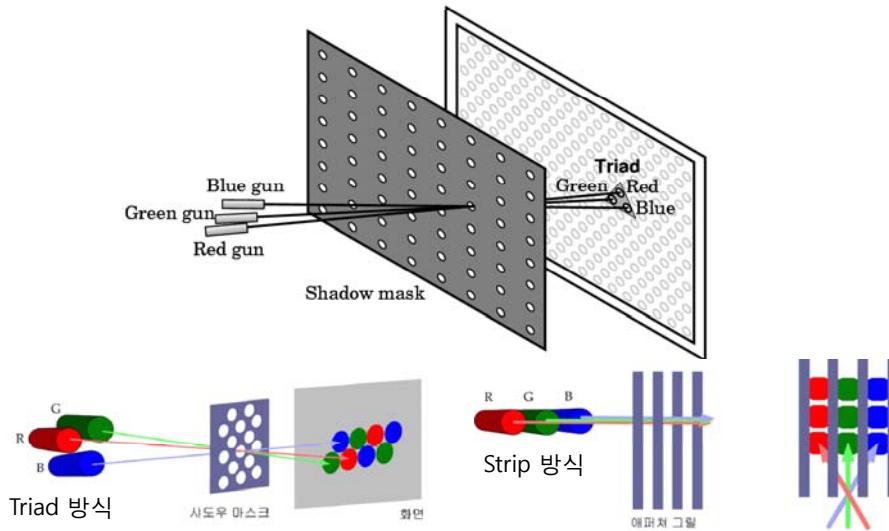
## Cathode-Ray Tube (CRT)



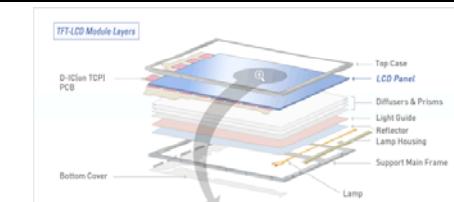
컴퓨터의 출력은 디지털 아날로그 변환기 (digital-to-analog converter)에 의해 x,y (수직, 수평) 편향판 사이의 전압으로 변환. 충분한 양의 전자선이 형광 물질에 도달하면 CRT 표면에서 빛이 방출. CRT는 a line-drawing device (calligraphic)로 사용되거나 프레임 버퍼 (raster mode) 출력에 사용

6

## Shadow Mask CRT



## Liquid Crystal Display (LCD)



[www.lgphilips-lcd.com](http://www.lgphilips-lcd.com)

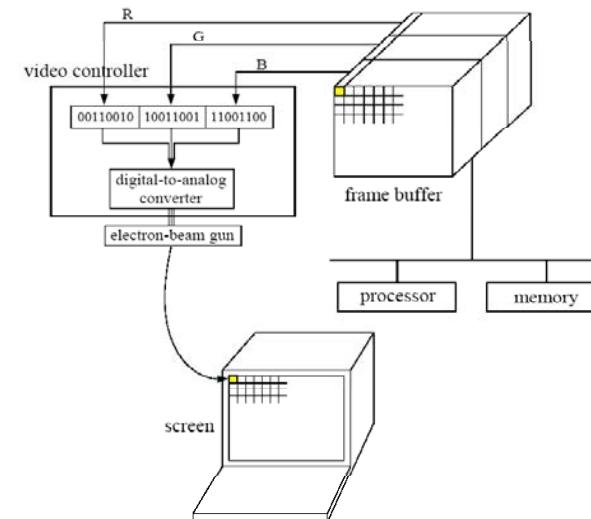
최근 널리 쓰는 TFT LCD는 박막트랜지스터와 화소 전극이 배열되어 있는 하판과 색상을 나타내기 위한 컬러 필터 및 공통 전극으로 구성된 상판, 그 사이에 채워져 있는 액정으로 구성. 그리고 두 유리기판의 양쪽 면에는 가시광선을 선형 편광하여주는 편광판이 각각 부착되어 있고, 상하판 전극 사이의 액정으로 축전기(capacitor)가 형성되고 이곳에 영상정보가 저장.

## Vector Graphics System

- ▣ Vector display, Calligraphic Display
- ▣ Oscilloscope, Radar, Plotter, Evans/Sutherland PS Series
- ▣ 화소 (pixel) 개념이 없음
  - 무한 해상도 (전자빔의 폭이 해상도)
  - Aliasing 없음
  - 프레임버퍼 없음
  - 디스플레이 리스트 명령어 집합



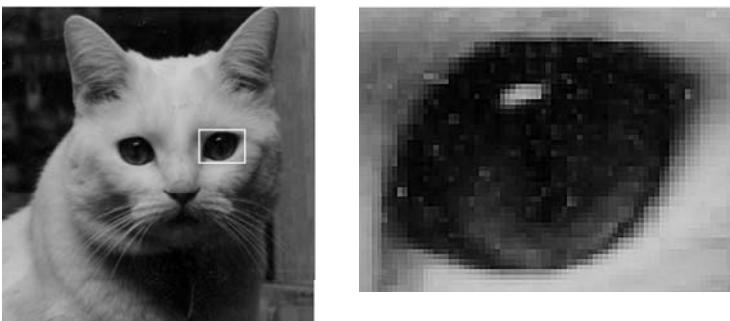
## Raster-based Graphics System



10

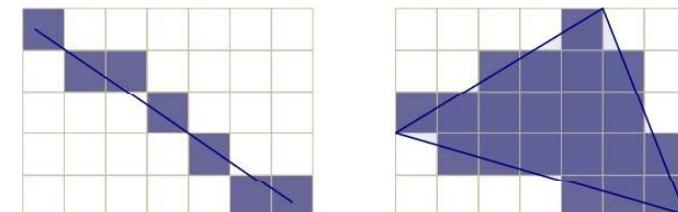
## Raster-based Graphics System

- ▣ 그래픽스 시스템 안의 프레임버퍼 (frame buffer)에서 화소 (pixel)의 배열 (array)인 래스터 (raster)로 생성



## Rasterization

- ▣ 래스터화 (Rasterization) = 주사 변환 (Scan Conversion)
  - 기하학적인 도형을 프레임버퍼 안의 픽셀의 색과 위치로 변환시키는 작업
  - 물체 좌표에서 화면 좌표로
  - 부동소수 좌표에서 정수 좌표로

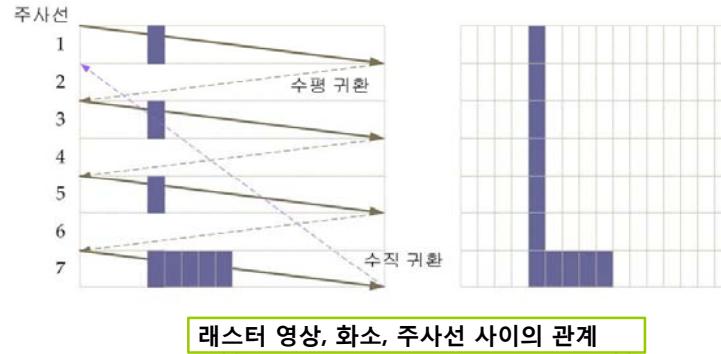


Aliasing (계단모양의 거친 경계선) 발생

## Interlacing

### ▣ 비월주사 (Interlacing, Interleaving)

- NTSC TV 표준: 60 Half Frame/Sec, 525 Scan Line, 4:3 Aspect Ratio

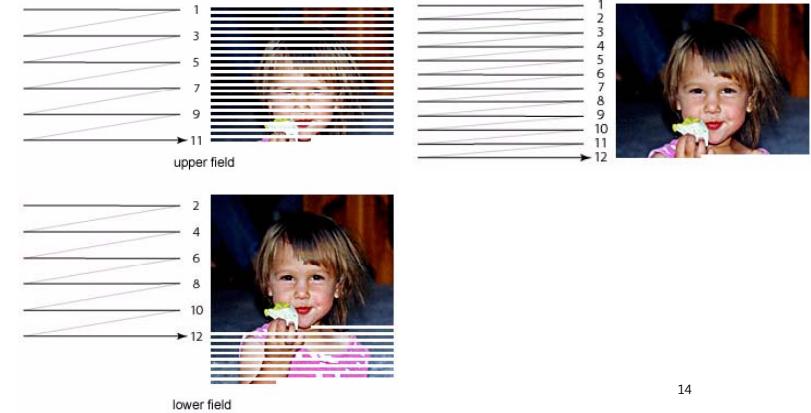


13

## Interlacing

### ▣ Interlaced vs. Non-interlaced (Progressive)

- TV에서 화면을 표시할 때 쓰는 방식인 인터레이싱
- 컴퓨터의 모니터에서 화면을 표시할 때 쓰는 방식인 프로그래시브



14

## Raster Image

### ▣ 래스터 이미지 (Raster Image)

- 직사각형 형태의 이미지 영역을 화소 (pixel)라 부르는 조그마한 영역으로 나누어 각 화소를 해당 영역을 대표하는 색깔로 칠함
- 연속적인 영상을 유한개의 화소를 사용하는 영역으로 표현하므로 오차가 발생 - aliasing

### ▣ 화소 (Pixel = picture element)

### ▣ 프레임버퍼 (Frame buffer)

- 화면에 출력되는 래스터 이미지에 대한 픽스맵 데이터는 프레임 버퍼(frame buffer)라고 불리는 메모리 한 부분에 저장되어야 함
  - 해상도 (resolution)은 프레임버퍼의 픽셀 수 – 640x480, 1024x768 ..
  - 프레임버퍼의 깊이 (depth of frame buffer)는 시스템이 얼마나 많은 색을 한 픽셀에서 표현할 수 있는지를 결정
- Eg., 8 bits: 256 colors

15

## Frame Buffer

### ▣ 넓은 의미로 프레임 버퍼(Frame Buffer)는 화면에 도시할 래스터 이미지뿐만 아니라 그러한 이미지를 생성하는데 필요한 여러 부류의 정보를 저장해주는 포괄적 의미의 그래픽스 전용 메모리(Video Memory)를 뜻함.

### ▣ 색깔 버퍼 (Color buffer)

- 더블 버퍼 (double buffer)
- 스테레오 버퍼 (stereo buffer)
- 알파 버퍼 (alpha buffer)

### ▣ 깊이 버퍼 (Depth buffer)

### ▣ 스텐실 버퍼 (Stencil buffer)

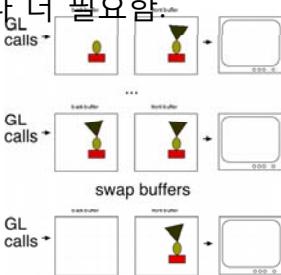
### ▣ 축적 버퍼 (Accumulation buffer)

### ▣ 픽셀 버퍼 (Pixel buffer) 등

16

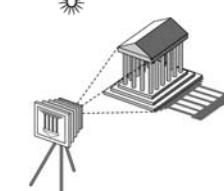
## Double Buffering

- 비디오 제어기가 항상 완성된 이미지를 도시하도록 함.
- 이를 위하여 프로세서는 이미지의 내용을 계산하여 후면 버퍼에 축적
- 그동안 비디어 제어기는 전면 버퍼의 내용을 읽어 화면에 이미지 도시함
- 싱글 버퍼링보다 훨씬 부드러운 애니메이션 생성
- 더블 버퍼링을 사용할 경우 버퍼가 하나 더 필요함. 필요에 따라 주어진 색깔 버퍼를 두개로 나누어야 함.
- 실제 frame rate이 낮아질 수 있음.

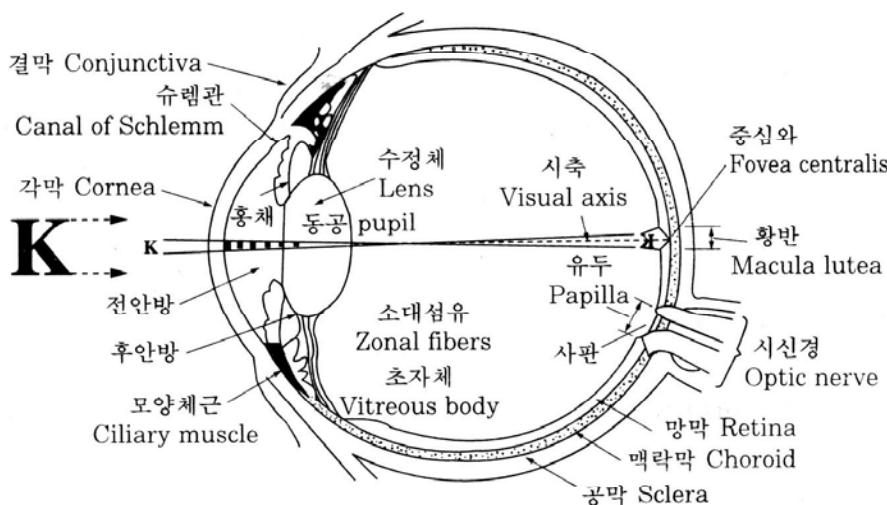


## Images

- 컴퓨터 영상 (Computer-generated images)은 물체가 실제 존재하지 않는다는 점에서 인공영상임.
- 컴퓨터가 영상을 생성하는 방법은 인간의 시각 시스템과 같은 전통적인 영상생성 방법과 흡사함.
- Object (객체)는 영상생성 과정이나 관측자와 관계없이 공간에 존재.
- Viewer (관측자)는 물체의 영상을 형성하는 것. 인간 시각 시스템에서는 망막, 카메라에서는 필름에 영상 형성.
- Light (광원)이 없다면 객체는 어둡게 되고 영상에서는 아무 것도 보이지 않는다.



## Human Visual System



## Human Eye

- 빛은 각막(cornea)을 통해 눈에 들어옴.
- 홍채(iris)는 동공을 통해 눈으로 들어오는 빛의 양을 조절. 동공(pupil)은 눈의 중앙에 검게 보이는 부분으로 열린 틈.
- 동공을 통과한 빛은 수정체(lens)를 통과하여 눈 후면의 감광 표면인 망막(retina)의 중심부근, 황반 (fovea)에 초점을 맞춤.
- 사람의 수정체는 관찰자와 물체 사이의 거리에 따라 수정체에 붙어 있는 모양근이 수축하거나 이완하여 초점에 맞도록 함. 물체가 가까우면 이 근육이 수축하여 수정체가 불룩해짐. 물체가 멀면 이 근육이 이완하여 수정체가 평평해짐.
- 망막은 카메라의 필름 같은 역할.
- 망막은 두 가지 광수용기(Photoreceptor), 즉 간상체(Rods)와 원추체(Cones)로 되어 있음.
- 망막은 빛을 신경신호로 바꿔서 뇌로 보내줌.

## Photoreceptors (Rods & Cones)

### 간상체(rods)

- 주로 어두울 때 활동
- 흑백의 음영(Luminance)망을 구분함
- 밝은 곳에 있다가 어두운 곳에 갑자기 들어가면 처음에는 아무것도 보이지 않다가 차차 주위가 보이기 시작. 처음 7~8분동안은 원추체가 작용하여 어두운 곳의 대상들을 보게 되며, 그 다음 약 40분 정도까지 간상체가 작용하여 대상들을 분명히 식별하게 됨.
- 간상체는 파란색 스펙트럼에 민감함

### 원추체 (Cones)

- 밝은 환경에서 작용
- 원추체의 밀집이 바로 시력(visual acuity)을 결정
- 원추체에는 세 가지가 있는데, 각각 삼원색(red, green, blue)에 대응하는 빛의 파장 범위에 대하여 민감함. [Young-Helmholtz설]
- 색깔을 보려면 원추체가 활성화되어야 하므로, 어두운 상태에서는 색각(color vision)이 존재하지 않게 됨.

## Visual Acuity

### 최소분간시력

- 가장 통용되는 시력 척도는 최소분간시력(minimum separable acuity), 눈이 검출할 수 있는 과녁의 최소 특징 또는 과녁의 부분 사이의 최소 공간을 말함.

### Vernier 시력

- Vernier 시력은 한 선과 다른 선의 측방향 변위(lateral displacement), 즉 미소한 치우침을 분간하는 능력임.

### 최소지각시력, 입체 시력

- 최소지각시력(minimum perceptible acuity)은 배경으로부터 한 점(가령 둥근 점)을 분간하는 능력. 또 깊이가 있는 단일 물체의 차이를 분간하는 능력을 입체시력(stereoscopic acuity)이라 함.

### 2 feet 떨어진 곳에서 1280x1024, 17" 모니터를 보면 1 pixel 당 1.4 arc-minutes을 시력을 제공해준다.

## Rods/Cones Distribution

### 간상체와 원추체의 분포

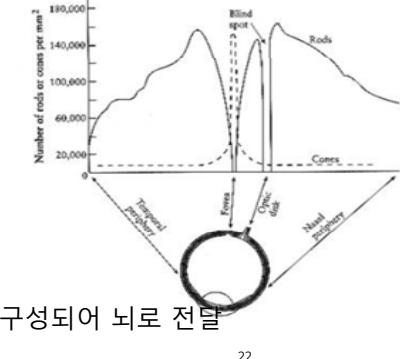
- 망막 표면에서 원추체와 간상체의 분포는 균일하지 않음. 600만~700만 개의 원추체가 황반(fovea)에 집중.
- 간상체는 황반으로부터 약 이십도 떨어진 곳에서부터 말초에 이르기까지 밀집해 있음

### 황반 (Fovea)

- 원추체만 다양 존재
- 간상체는 없음
- S-cones (blue 원추체)는 없음
- 시력이 가장 예민함

### 맹점 (Blind Spot)

- 간상체와 원추체 없음
- 신경절 세포(ganglion cells)만 구성되어 뇌로 전달



22

## Color Perception

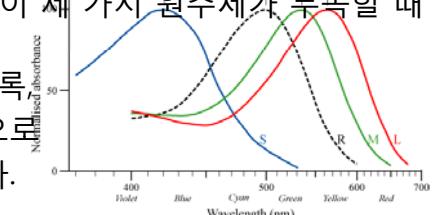
### 사람이 인식할 수 있는 가시분광(Visible spectrum)은 400 nm(보라색)~700 nm (빨강색)사이에 펼쳐져 있다.

### 세상의 물체들은 극단적인 경우를 제외하고 여러 파장들을 반사하므로 망막의 동일 지점이 이들에 의해 자극을 받는다.

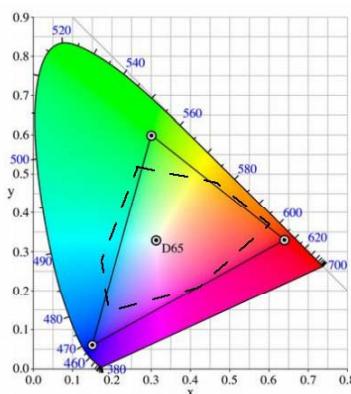
### S, M, L 세 가지 원추체가 있는데 각각 파랑색, 초록색, 빨강색에 대응하는 빛의 파장 범위에 대하여 민감한데 색깔의 지각은 이 세 가지 원추체의 동시작용의 결과이다.

### 색맹 (Color blindness)은 이 세 가지 원추체가 부족할 때 나타나는 현상이다.

### 컬러 텔레비전은 빨강, 초록, 파랑점들의 가산적 혼합으로 대부분의 색깔을 나타낸다.



## Chromatic Color



### 색상 (Hue)

- 우리가 색깔이라고 부르는 시각 경험

### 채도 (Saturation)

- 색깔의 채도란 그 색이 흰색/회색 때문에 그 순수성이 떨어지는 정도
- 진한색 (빨강색, 파랑색)은 채도가 높은 것. 회색으로부터 멀리 있음.
- 파스텔톤 (분홍색, 하늘색)은 채도가 낮은 것. 회색에 가까워짐.

### 명도 (Brightness)

- 빛 자극의 물리적 강도가 감각되는 명도를 결정.
- 명도의 차원은 흑색에서부터 색상이 없고 명도가 최대인 백색까지 변함.

25

## Color Model

### 색깔을 어떤 모델을 사용하여 수치적으로 표현할 것인가?

- RGB
- CMY
- HSV/HSB
- YUV/YIQ – PAL/NTSC Television

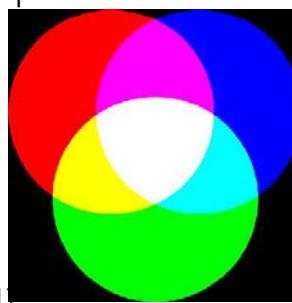
26

## RGB

### 빛의 삼원색으로 컬러 디스플레이 시스템에 적합

### RGB 컬러 모델은 각 화소 값이 그 화소에 칠할 (RED, GREEN, BLUE)값을 더해 표현하는 방식

- $n_R$ : # of bits for R channel
- $n_G$ : # of bits for G channel
- $n_B$ : # of bits for B channel
- $n = n_R + n_G + n_B$
- $2^n = 2^{(nR + nG + nB)}$  개의 색깔 표현 가능



### Examples

- Black (0, 0, 0), White (1, 1, 1)
- Red (1, 0, 0), Green (0, 1, 0), Blue (0, 0, 1)
- Cyan(0, 1, 1), Magenta (1, 0, 1), Yellow (1, 1, 0)

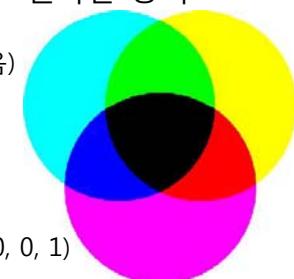
27

## CMY

### 색의 삼원색으로 프린팅에 적합.

### CMY 컬러 모델은 RGB의 보색인 Cyan, Magenta, Yellow을 사용. 흰색에서 RGB를 빼서 표현하는 방식.

- Cyan = 1 – red (green& blue만 남음)
- Magenta = 1 – green (red & blue 만 남음)
- Yellow = 1 – blue (red & green 만 남음)



### Examples

- Black (1, 1, 1), White (0, 0, 0)
- Red (0, 1, 1), Green (1, 0, 1), Blue (1, 1, 0)
- Cyan (1, 0, 0), Magenta (0, 1, 0), Yellow (0, 0, 1)

### RGB/CMY 변환

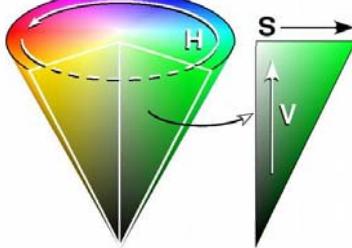
- CMY = (1, 1, 1) – RGB
- RGB = (1, 1, 1) – CMY

### CMYK는 검정색(Black, K)를 추가, 인쇄에서 주로 사용.

## HSV/HSB

- HSV 컬러 모델은 색상(Hue), 채도 (Saturation), 명도 Value/Brightness)

- 색상은 0 ~ 360도의 범위 각도
  - 0 - 빨간색, 120 - 녹색, 240 - 파란색
- 채도는 0 ~ 1 범위의 반지름
  - 0 - 명도의 명암도 (무채색)
  - 1 - 컬러는 원뿔모형 기반의 꼭대기 가장자리 100% 채도
- 명도는 z축에서의 위치
  - 0 - 검정
  - 1 - 흰색



## YUV/YIQ

- YUV 컬러 모델은 TV에서 RGB 신호를 영상의 명암을 나타내는 명도 (Luminance) (Y)와 두 개의 색차 (Chrominance) (U, V)로 변환하여 표시하는 방법.

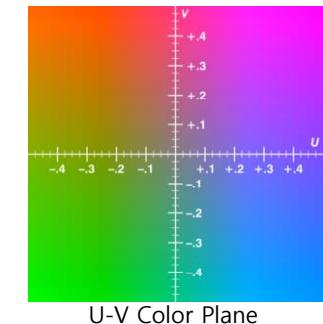
- Y: 밝기를 0 ~ 1의 수로 표현
- U: 푸른 정도를 -0.5 ~ 0.5의 수로 표현
- V: 블루 정도를 -0.5 ~ 0.5의 수로 표현

- RGB->YUV

$$\begin{aligned} Y &= 0.299 R + 0.587 G + 0.114 B \\ U &= -0.147 R - 0.289 G + 0.436 B \\ V &= 0.615 R - 0.515 G - 0.100 B \end{aligned}$$

- YUV->RGB

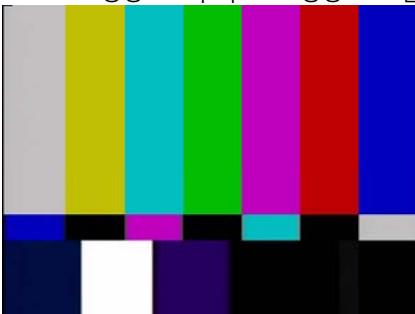
$$\begin{aligned} R &= Y + 1.140 V \\ G &= Y - (0.396 U + 0.581 V) \\ B &= Y + 2.029 U \end{aligned}$$



30

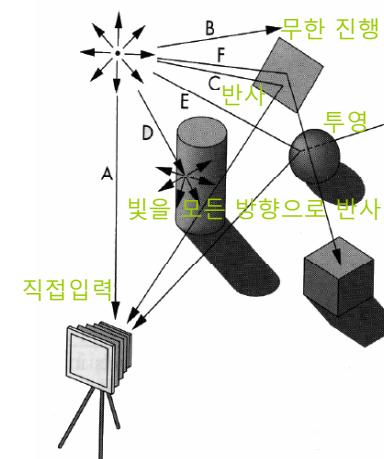
## Luminance

- 색의 밝기 (brightness)
- RGB 컬러에서 명암도로 변환
  - NTSC 표준 명암도 =  $0.299 * R + 0.587 * G + 0.114 * B$ 
    - 명도에 대한 NTSC 표준에 따름
  - 평균 명암도 =  $0.33 * R + 0.33 * G + 0.33 * B$ 
    - RGB 영상으로부터 HSV 영상으로 변환



31

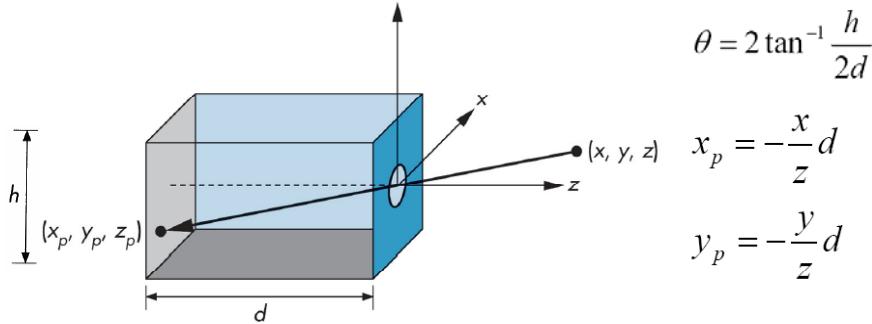
## Synthetic Imaging Process



- 점 광원 (point light source)은 한 지점에서 모든 방향으로 빛을 방출
- 관측자 (Viewer)는 카메라
- 광선 (ray)은 한 점에서 시작하여 임의의 방향으로 무한히 진행하는 반직선.
- 광선추적 (Ray tracing) 또는 광자매핑 (photo mapping)은 이런 영상모델을 기반으로 하여 제작한 기법
- 라디오시티 (radiosity)는 면에 입력된 광선이 모든 방향으로 똑같이 반사되는 물체에 가장 적합. 에너지 바탕 기법.

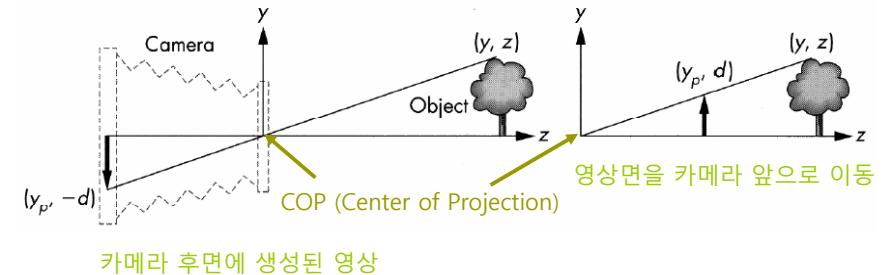
## Pinhole Camera

- Pinhole camera는 상자의 한쪽 면에 작은 구멍이 나있고 필름이 안쪽의 반대쪽 면에 놓인 상자
- $(x_p, y_p, -d)$ 는  $(x, y, z)$ 가 투영된 점
- 카메라의 시야는 필름 면에 맷힐 수 있는 가장 큰 크기의 객체에 의해 만들어지는 각도



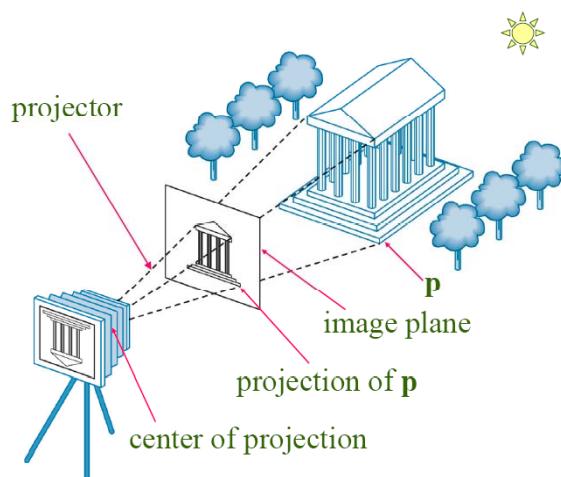
## Synthetic-Camera Model

- Pinhole camera model에서 image plane을 앞으로 움직인 것
- 클리핑 윈도우 (Clipping window)
  - 합성카메라에서는 시야에 따른 영상 크기의 제한을 고려해야 함



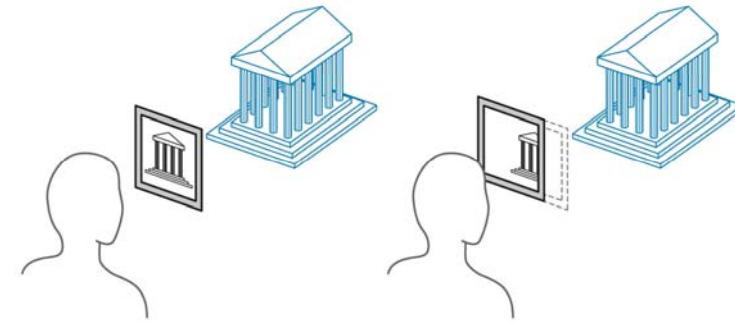
34

## Synthetic-Camera Model



35

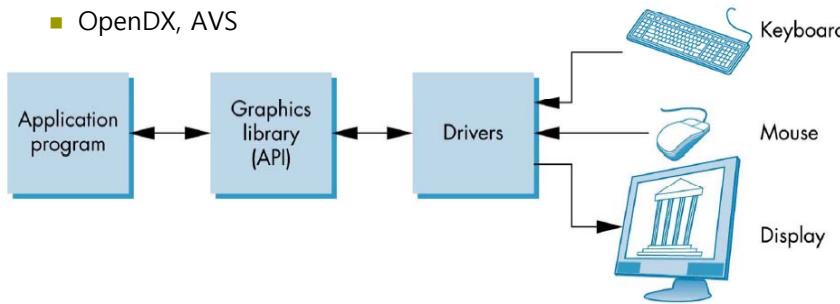
## Clipping



36

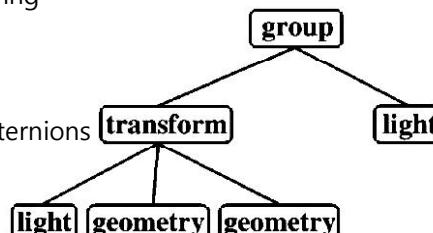
## 3D Graphics API

- ▣ Low-level graphics libraries
  - OpenGL, Direct3D
- ▣ Scene graph libraries
  - SGI Performer, Open Inventor, Open Scene Graph, Java3D
- ▣ Scientific visualization & advanced graphics toolkits
  - OpenDX, AVS



## Scene Graph API

- ▣ 장면 그래프 (scene graph)는 그래픽을 표현하는데 쓰이는 트리 (tree)임
- ▣ 노드 (node)는 그룹, 변환, 조명, 기하 (group, transformation, light, geometry) 등을 표현하는데 쓰임
- ▣ 주요 요소:
  - Standard data structures for geometry & graphics state
  - Automatic, optimized rendering
  - View culling
  - Level of detail
  - Model loaders
  - Math: vectors, matrices, quaternions
  - Intersection testing
  - Multiprocessing



39

## Low-level 3D Graphics API

- ▣ 기본적으로 제공하는 명령어:
  - 점, 선, 폴리곤, 곡선과 곡면 (points, lines, polygons, curves and surface) 기하학적 모델링
  - 위치, 회전, 크기 지정
  - 컬러 지정
  - 조명 (light sources) 지정
  - 재질 (material properties) 지정
  - 관측 (view) – 카메라 위치 (camera position: center of projection), 방향 (camera coordinate system), 초점 거리 (size of the image), 필름 면 (height and width of the back of the camera)
  - 텍스쳐 로딩 (texture loading)

38

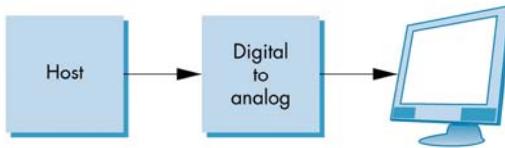
## Graphics Pipeline

- ▣ 그래픽스 파이프라인은 PC 메모리에 있는 프로그램과 데이터
- ▣ CPU가 프로그램을 실행하여 그래픽스 명령어를 처리함
- ▣ 명령어와 데이터가 그래픽스 카드로 보내짐
- ▣ 영상 (Images)는 그래픽스 카드 메모리에서 렌더링됨
- ▣ CPU와 그래픽스 카드를 연결해주는 인터페이스
  - PCI
  - AGP
  - PCI Express
- ▣ 그래픽스 카드와 디스플레이를 연결해주는 인터페이스
  - VGA
  - DVI
  - Composite, S-Video

40

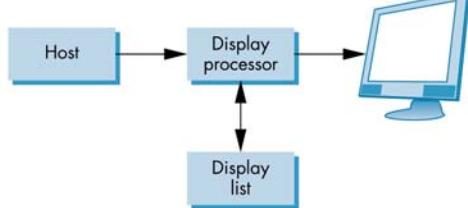
## Graphics Architecture

### ▣ 초기 그래픽스 시스템



### ▣ 디스플레이 프로세서 구조

- 디스플레이 프로세서에서 리스트 안에 있는 프로그램을 실행하여 scan conversion하여 디스플레이로 보내줌

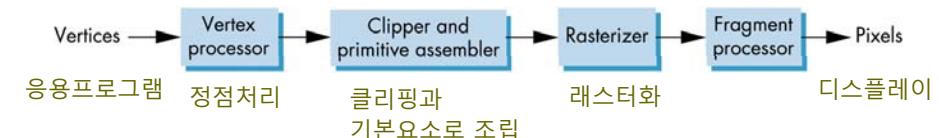


41

## Pipeline Architecture

### ▣ 그래픽 파이프라인 (Graphics pipeline) 구조

- 정점 처리 (vertex processing)
- 클리핑과 기본요소로 조립 (clipping and primitive assembly)
- 래스터화 (rasterization)
- 단편 처리 (fragment processing)

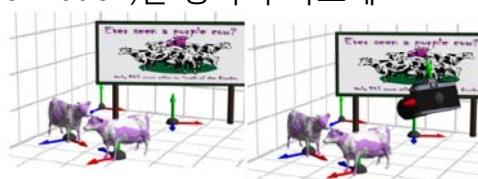


## Vertex Processing

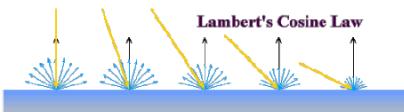
### ▣ 각 정점의 좌표변환을 수행하고 색을 계산 처리

### ▣ 행렬 변환(matrix transformation)을 통하여 좌표계 변환을 계산

- Model transformation
- Viewing transformation



### ▣ 광원의 특성과 객체 표면의 물리적인 특성을 고려한 물리적인 음영모델을 이용하여 색을 계산



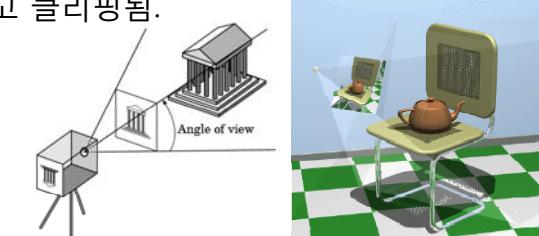
43

## Clipping and Projection

### ▣ 투영 (projection)은 관측자가 구도를 잡았을 때 물체를 구성하는 3차원 공간의 점이 2차원 평면인 화면의 어느 지점으로 투영되는지 결정

- 원근 투영 (Perspective projection)
- 평행 투영 (Parallel projection)

### ▣ 카메라 앞에 피라미드 같은 클리핑 볼륨(clipping volume)을 두어 윈도우 밖에 투영되는 객체는 영상으로 나타나지 않고 클리핑됨.



## Primitive Assembly

- ▣ 클리핑은 정점 단위로 이루어지기 보다는 기본요소 단위로 이루어짐
- ▣ 파이프라인의 클리핑 단계에서는 클리핑이 이루어지기 전에 정점의 집합을 아래와 같은 기본요소 단위로 조립함
  - 선분 (Line segments)
  - 다각형 (Polygons)
  - 곡선과 곡면 (Curves and surfaces)

45

## Rasterization

- ▣ 클리핑기로부터 나온 기본 요소는 아직 정점으로 표현되어 있는데 프레임 버퍼의 픽셀로 변환되어야 함.
- ▣ 래스터기의 출력은 각 기본 요소 단편 (fragments)의 집합
- ▣ 단편은 색과 위치 정보를 전달하는 잠정적인 픽셀
- ▣ 단편은 주어진 픽셀에 대해서 현재의 단편이 이미 래스터화된 단편의 뒤에 놓여있는지를 결정하기 위한 깊이(depth) 정보

46

## Fragment Processing

- ▣ 래스터기에서 생성한 단편 (fragments)을 받아들여 프레임 버퍼 안에 있는 픽셀을 갱신
- ▣ 단편에 대응하는 픽셀의 색은 프레임 버퍼로부터 읽혀지거나 또는 반투명의 효과를 주기 위하여 단편의 색과 혼합
- ▣ 단편의 색은 텍스쳐 매핑 혹은 범프 매핑으로 바꾸거나, 또는 정점의 색으로 보간.
- ▣ 카메라에서 가까운 단편은 다른 단편을 보이지 않게 할 수 있음.
  - 은면 제거 (Hidden-surface removal)

47

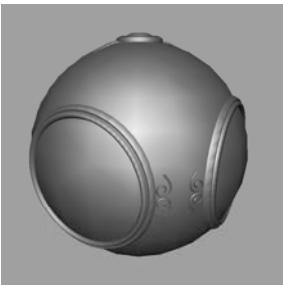
## Programmable Pipeline

- ▣ 최근 NVIDIA와 ATI 등 그래픽스 카드 제조회사에서 발표하고 있는 GPU (Graphics Processing Unit)에서 사용하고 있는 구조
- ▣ 렌더링 파이프라인의 일부분을 프로그래머가 **vertex shader**와 **pixel/fragment shader**를 통하여 원하는 방식으로 프로그래밍을 할 수 있음.
- ▣ 과거에는 불가능하였던 다양한 실시간 렌더링 효과를 유연하게 생성할 수 있음.

48

## Computer Graphics: 1980-1990

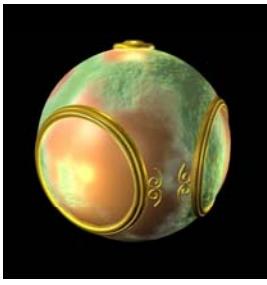
Realism comes to computer graphics



Smooth shading



Environment mapping



Bump mapping

49

## Computer Graphics: 1980-1990

- Special purpose hardware
  - Silicon Graphics geometry engine
    - VLSI implementation of graphics pipeline
- Industry-based standards
  - PHIGS
  - RenderMan
- Networked graphics: X Window System
- Human-Computer Interface (HCI)

50

## Computer Graphics: 1990-2000

- OpenGL API
- Completely computer-generated feature-length movies (Toy Story) are successful
- New hardware capabilities
  - Texture mapping
  - Blending
  - Accumulation, stencil buffers



## Computer Graphics: 2000-

- Photorealism & Non-Photorealism
- Graphics cards for PCs dominate market
  - Nvidia, ATI
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry
  - Maya, Lightwave
- Programmable pipelines
  - Cg, GLSL, HLSL

