

컴퓨터 그래픽스

제1장 컴퓨터 그래픽스 개요 및 시스템

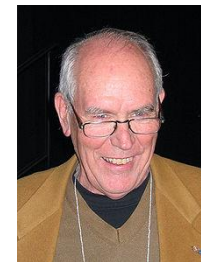
2018년 2학기

컴퓨터 그래픽스의 발전

- 컴퓨터 그래픽스
 - 컴퓨터를 사용하여 그림을 생성하는 기술
- 1950~60년대

1960	William Fetter	“컴퓨터 그래픽”이란 용어를 최초로 사용
1962	Jack Bresenham	선분 그리기 알고리즘을 개발
1963	Ivan Sutherland	컴퓨터 그래픽의 제반 개념을 확립
1963	Douglas Englebart	최초의 마우스 프로토타입

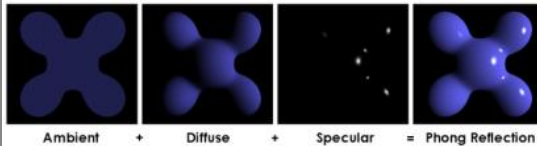
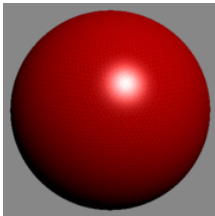
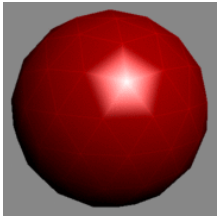
- 윌리엄 페터 (William Fetter)
 - 미국의 그래픽 디자이너로 컴퓨터 그래픽스의 창시자
 - 사람 모습의 컴퓨터 애니메이션 개념을 개발
- 이반 서더랜드(Ivan Sutherland)
 - 컴퓨터 그래픽스의 개념 확립
 - 대화형 컴퓨터 그래픽 개념: 라이트 펜으로 의사전달
- 잭 브레즌함 (Jack Bresenham)
 - Winthrop 대학 교수
 - 1962년 IBM 재직 시 직선 그리기 알고리즘 개발



컴퓨터 그래픽스의 발전

• 1970년대

1971	Gouraud	구로 셰이딩 알고리즘
1974	Bui-Tong Phong	전반사에 의한 하이라이트 알고리즘
1975	Martin Newell	베지어 표면 메쉬를 사용한 차 주전자 모델
1975	Benoit Mandelbrot	프랙탈 이론
1977	Steve Wozniak	컬러 그래픽 PC: Apple II



- 베노이트 맨델브롯 (Benoit Mandelbrot)
 - 폴라드 출신의 미국 수학자
 - 프랙탈 이론과 자연의 “roughness and self-similarity” 개념 확립
- 스티브 워즈니악 (Steve Wozniak)
 - 미국의 발명가, 전기공학자, 프로그래머
 - 스티브 잡스와 함께 애플 공동 창시자



컴퓨터 그래픽스의 발전

- 1980년대
 - PC시대
 - 래스터 그래픽 모니터
 - HCI 기술 발전(윈도우, 메뉴, 아이콘, 마우스 등)

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

컴퓨터 그래픽스의 발전

- 1990년대
 - 3D 그래픽스 발전
 - 인터넷 환경

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	Id Software: John Carmack, Michael Abrash	Quake 그래픽 엔진 개발 (샘플)
1999	NVIDIA	GeForce 256 GPU

컴퓨터 그래픽스의 발전

- 2000년 이후
 - 실시간 렌더링
 - 물체의 사실감과 자연스러움 증가
 - 모바일 환경

2003	Timothy Purcell	광선추적 기법을 GPU에 적용
2004	Id Software	Doom Engine 발표
2004	DirectX, OpenGL	New Version 발표
2007	OpenGL	OpenGL ES 2.0 발표
2009	DirectX	DirectX 11 발표
2014	OpenGL	OpenGL 4.5 발표
2017	OpenGL	OpenGL 4.6 발표

2차원 그래픽스

- 2차원 그래픽스

- 점, 선, 원, 곡선 등과 같은 기본 도형을 이용하여 2차원 평면상에 그림 그리는 것
- 결과물을 픽셀의 형태로 표현: 각 픽셀은 적색(Red), 녹색 (Green), 청색(Blue)의 농도 값을 배합
- 그래픽의 표현방법과 처리방식에 따라 벡터 그래픽스와 래스터 그래픽스로 구분
- 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 확대 시에도 화질의 변화가 없음

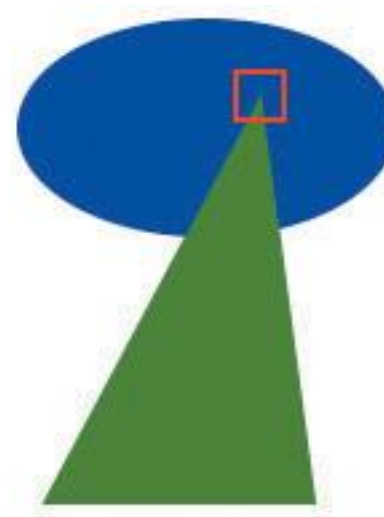
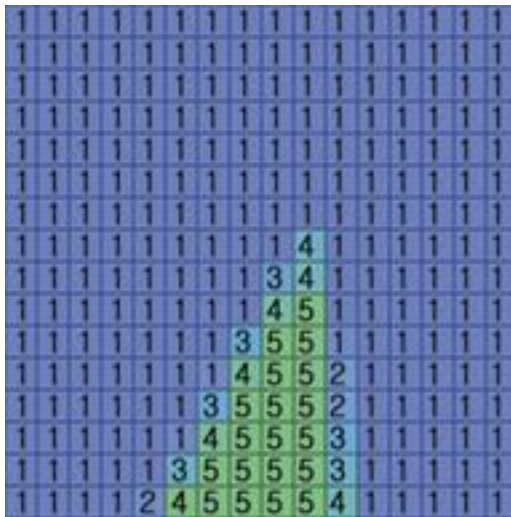
```
<?xml version = "1.0"?>
<svg width="200" height="200">
<ellipse cx="110" cy="50" rx="70" ry="40">
style="fill:blue; stroke:blue; stroke-width:2"/
<polygon style="fill:green;stroke:green;"
stroke-width:2" points="130,40 140,190 50,190"
/>
</svg>
```



2차원 그래픽스

- 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



3차원 그래픽스

- 3차원 그래픽스
 - 3차원 그래픽 생성과정 :
 - 물체의 기하학적인 형상을 모델링(Modeling)
 - 3차원 물체를 2차원 평면에 투영(Projection)
 - 생성된 3차원 물체에 색상과 명암을 부여(Rendering)

3차원 그래픽스

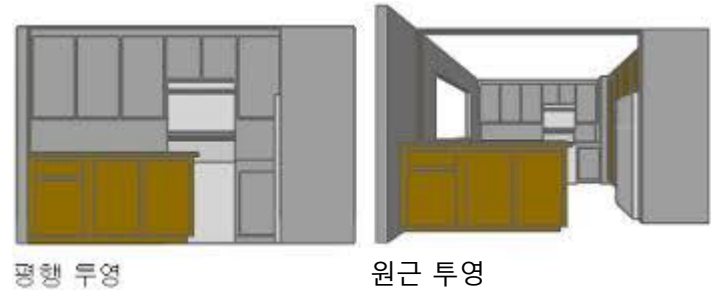
- 모델링(Modeling) 과정
 - 3차원 좌표계에서 물체의 모양을 표현하는 과정
 - 와이어프레임(Wireframe)모델
 - 다각형 표면(Polygon Surface)모델
 - 솔리드(Solid)모델링
 - 3차원 스캔에 의한 모델링



3차원 그래픽스

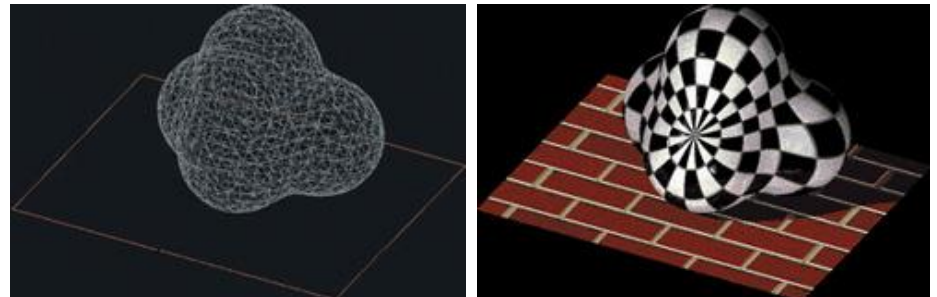
- 투영(Projection) 과정

- 3차원 객체를 2차원 화면에 투영
 - 평행 투영법, 원근 투영법



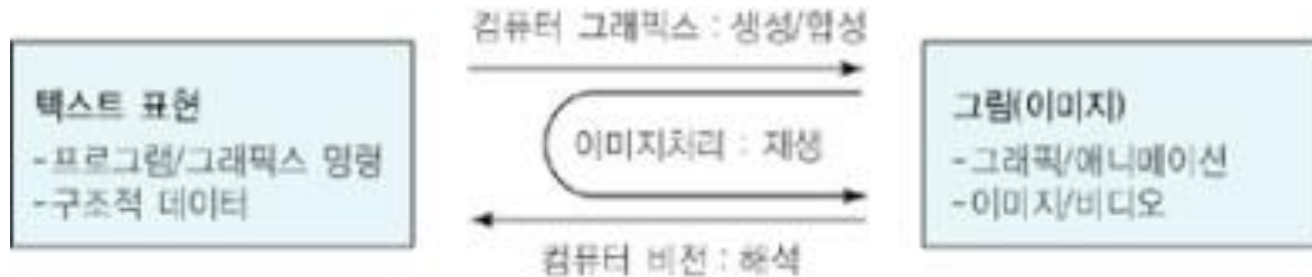
- 렌더링(Rendering) 과정

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



그래픽스, 이미지 처리

- 그래픽스와 이미지 처리



- 컴퓨터 그래픽스

- 인공적인 그림의 생성
- 일반적으로 그림의 품질이 중요함

- 이미지 처리(Image Processing)

- 실물 사진을 디지털 카메라나 스캐너를 이용하여 디지털화

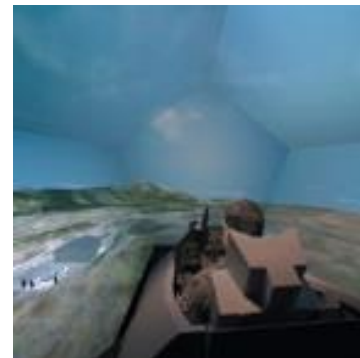
컴퓨터 그래픽스 활용

- CAD (Computer Aided Design)
 - 부품설계 및 도면작성(Drafting), 기계설계
 - VLSI 설계, 전자회로 설계
 - Communication network, Water/Electricity supply system
 - 자동차, 비행기, 선박의 설계: Wireframe model
 - 건축 설계



컴퓨터 그래픽스 활용

- 컴퓨터 애니메이션과 시뮬레이션(Simulation)
 - 프레임들의 빠른 연속적인 디스플레이
 - Education, Training(Flight simulator), Physical system 의 Behavior 연구



컴퓨터 그래픽스 활용

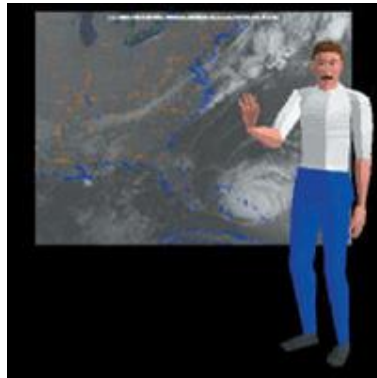
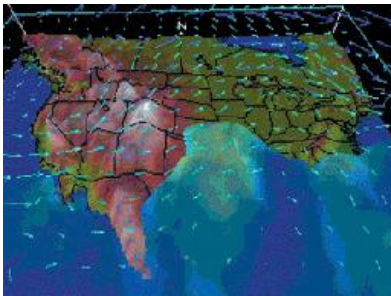
- 컴퓨터 디자인 및 아트
 - 상업 디자인(Commercial art)
 - 창작 미술(Creative art)



- 게임 및 엔터테인먼트
 - 영화, 게임, 뮤직 비디오, TV 프로그램 등
 - 컴퓨터 게임: 2차원 게임에서 3차원 게임으로 발전

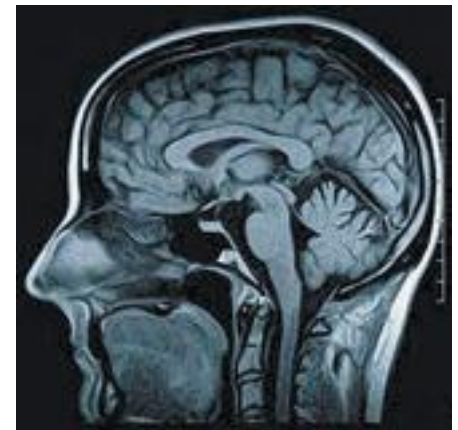
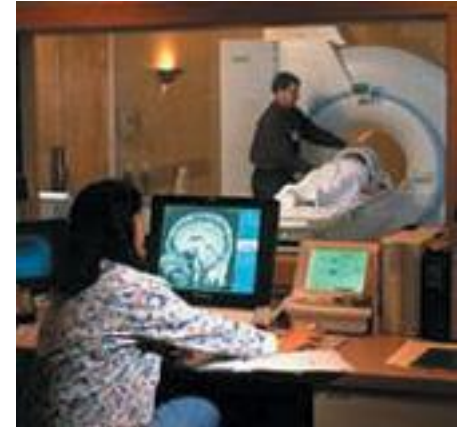
컴퓨터 그래픽스 활용

- 프레젠테이션 및 데이터 시각화(Data Visualization)
 - 프레젠테이션 그래픽스(Presentation Graphics) : Graph, Chart, Business graphics, Project management
 - Computer generated model (Visualization) - Physical, Financial, Economic model
- 멀티미디어 분야에서의 활용
 - 그래픽은 멀티미디어 응용에서 가장 자주 이용하는 매체
 - 웹페이지, 디지털 방송, 휴대폰, 사이버 클래스, 가상환경의 구축, 아바타 생성



컴퓨터 그래픽스 활용

- GUI(Graphical User Interface)
 - Window
 - Icons
 - Menu
- 전자 출판
 - Document Preparation System
- Publication (DTP : Desk-Top Publishing) 공간 정보의 표현
 - 지리정보시스템(GIS: Geographic Information System)
 - 차량 주행안내 시스템(Car Navigation System)
- 이미지처리(Image Processing)
 - Feature Detection
 - Pattern Recognition
 - 3D Reconstruction(예: MRI, CT)



컴퓨터 그래픽스 시스템

- 그래픽스 시스템 구성:

- 그래픽스 하드웨어:

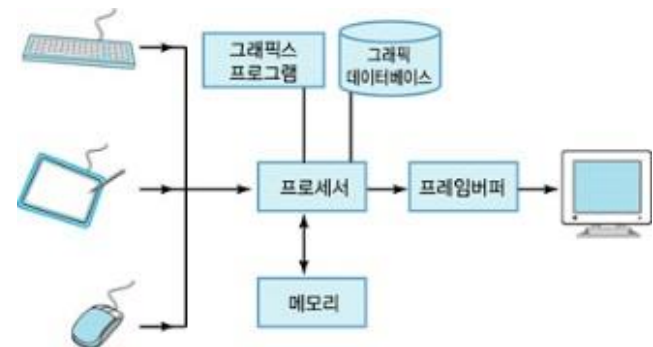
- 그래픽 데이터를 입력 또는 생성하는 입력 장치
 - 그래픽 데이터를 처리하는 프로세서인 CPU와 그래픽스 가속기
 - 결과물 출력하는 출력장치
 - 그 외, 시스템 메모리와 프레임 버퍼

- 그래픽스 프로그램:

- 그래픽 데이터를 생성하고 처리하여 원하는 형태의 그래픽 결과물을 디스플레이 하기 위해서 사용되는 프로그램
 - 그래픽스 라이브러리 및 응용 프로그램

- 그래픽스 데이터베이스

- 방대한 양의 그래픽 데이터를 관리하고 검색

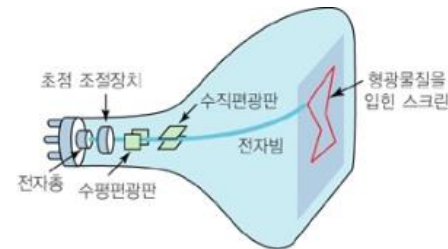


출력 장치

- 모니터

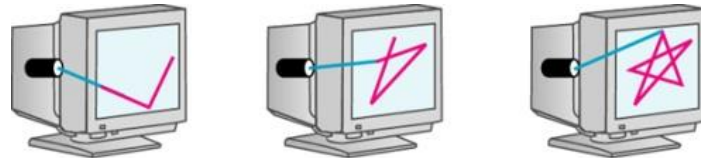
- CRT 모니터

- 주사 방식에 따라 벡터모니터, 래스터 모니터로 구분된다.



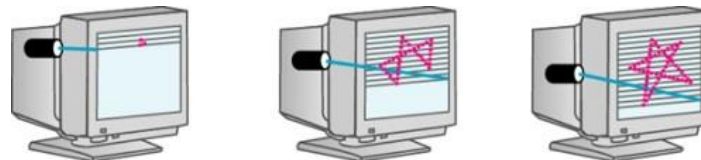
- 벡터 모니터:

- 전자총이 선이나 도형의 위치를 따라가며 직접 전자 빔을 스캐닝하여 그림 그리는 형태



- 래스터 모니터

- 명암 및 색상 표현에 적합, 대부분의 모니터에서 사용



출력 장치

– 평판 디스플레이

- LCD (Liquid Crystal Display)

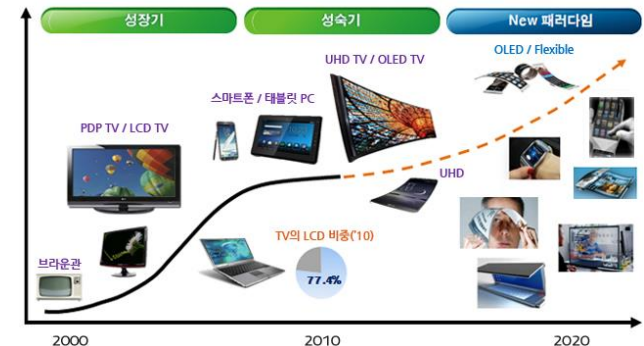
- 시계, 컴퓨터, TV, 자동차/항공기 속도표시판 등에 폭넓게 사용
- 2장의 얇은 유리판 사이에 고체와 액체의 중간물질인 액정을 주입하여 상하 유리판 위 전극의 전압차로 액정의 분자배열을 변화시켜 명암을 발생시켜 구동하는 방식
- 소비전력이 적고 가볍고 슬림해서 휴대하기 편리한 장점을 가지고 있으나 스스로 빛을 내지 못하고 시야각과 휘도가 좋지 않고 응답속도가 느린 단점

- LED (Lighting Emitting Diode)

- 발광 다이오드
- 에너지가 가해졌을 때 가시광선이나 비가시광선(적외선)을 발산하는 다이오드
- 반도체라는 특성으로 인해 처리속도, 전력소모, 수명 등에서 큰 장점

- OLED (Organic Light Emitting Diode)

- 유기발광다이오드
- 특정한 반도체 물질에서 전기적 충격으로 분리된 전자와 정공이 재결합하면서 발광하는 현상을 이용해서 색상을 표시하는 방식이며 고휘도, 배터리의 긴 수명, LCD에 비해 응답속도가 1000배 빠르고 경량, 박형의 저전력



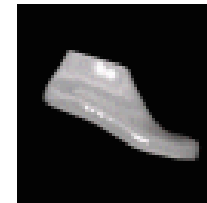
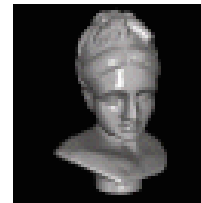
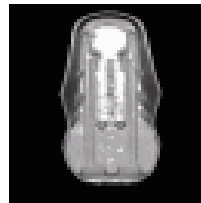
입력 장치

- 2차원 입력 장치

- 마우스
- 그래픽스 태블릿
- 조이스틱
- 라이트 펜

- 3차원 입력 장치

- 3D 디지털라이저, 3D 스캐너
- 모션 캡처 장치

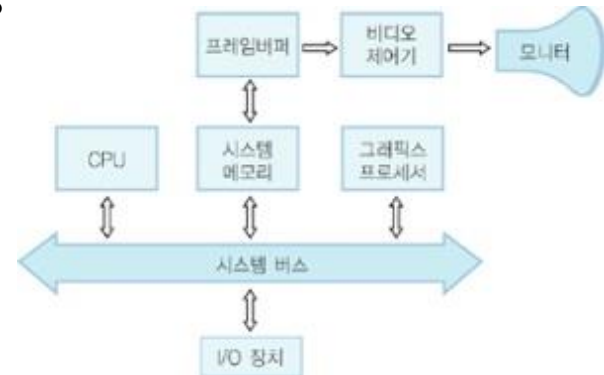


- 출력 장치

- 프린터
- 플로터
- 3D 프린터 ([샘플](#))

그래픽스 프로세서

- 그래픽스 프로그램의 처리결과가 그래픽스 프로세서에 의해 수행되는 방식
 - 랜덤 스캔 방식 (Random Scan)
 - 벡터 모니터에서 그래픽스 프로세서가 그림을 처리하는 방식
 - 그래픽스 응용 프로그램 실행 → 디스플레이 파일 생성 → 시스템 메모리 저장 → 매 활성 주기마다 모니터에 출력
 - 래스터 스캔 방식 (Raster Scan)
 - 래스터 모니터에 적용되는 방식
 - 그래픽스 응용 프로그램 실행 → 그래픽스 명령어 형태로 처리 결과 생성 → 명령어들이 그래픽스 프로세서에 의해 수행 → 픽셀 이미지 형태로 프레임 버퍼 내에 저장 [스캔 변환]
 - 빠른 활성화를 위해 이중 프레임버퍼를 사용



그래픽스 소프트웨어

- 그래픽스 소프트웨어

- 그래픽스 라이브러리:

- 그래픽스 응용 프로그램을 개발하는데 필요한 기능과 알고리즘을 제공하는 라이브러리
 - OpenGL, Direct3D 등

- 그래픽스 응용프로그램:

- 특정한 분야의 그래픽을 제작하거나 특수한 목적으로 개발된 경우
 - CAD, 3D Max 등