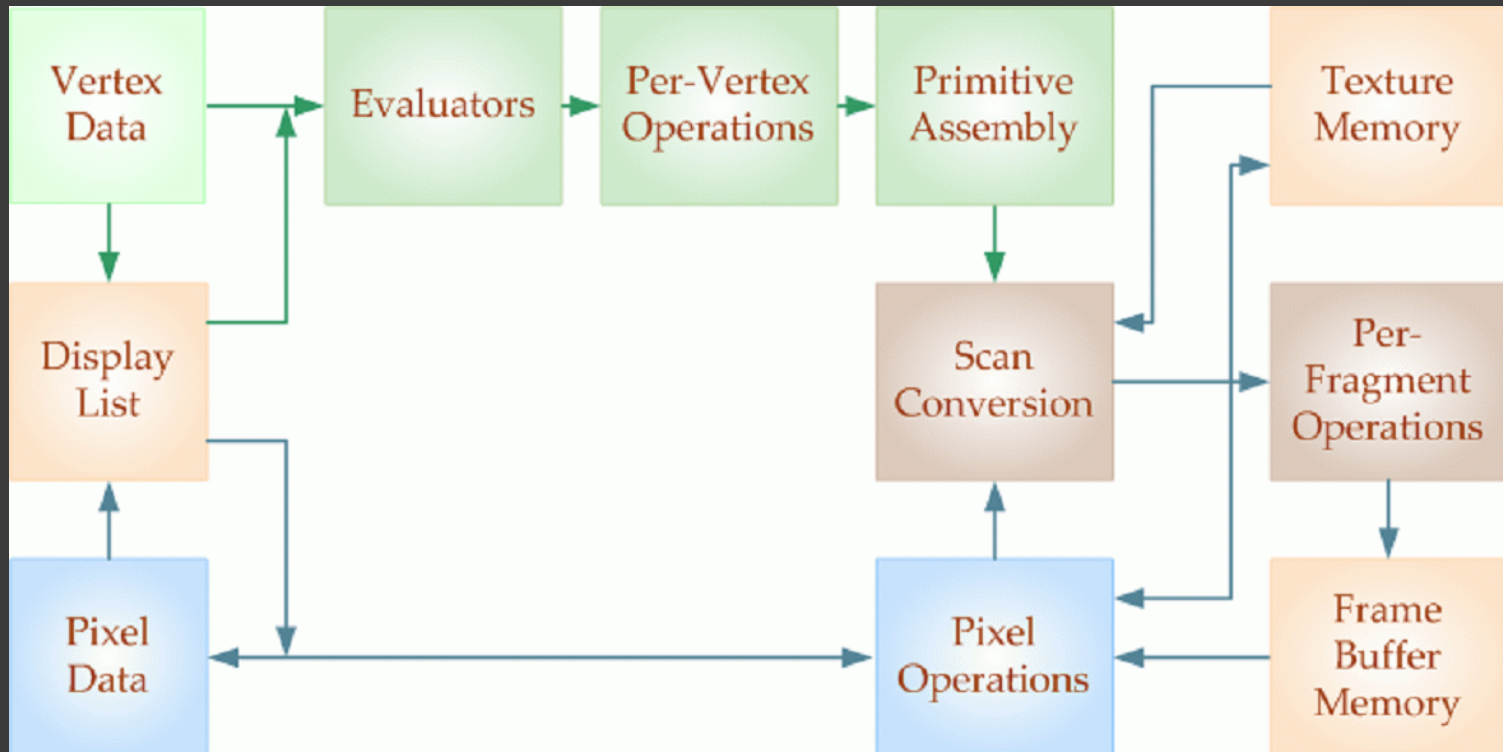


동명대학교
강영민

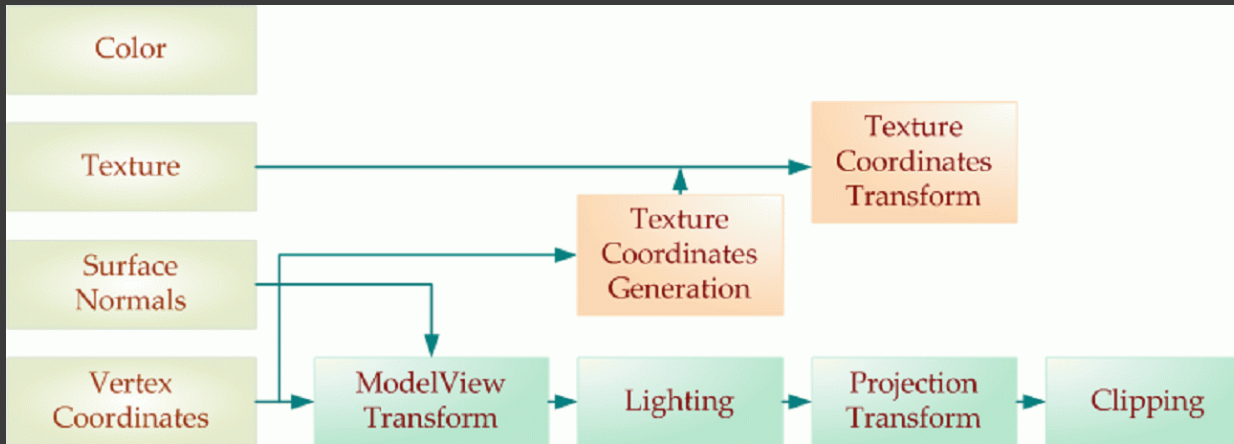
FRAGMENT

지엘 파이프라인

- 녹색 화살표: 3차원 기하 물체 처리(Geometry Processing)
- 청색 화살표: 2차원 영상 처리 (Image Processing)



지엘 파이프라인



- ◎ 정점 연산 프로세스: 정점 좌표로부터 투상변환까지.
- ◎ 기본요소 조합 프로세스: 절단
- ◎ 텍스처 정점 생성 프로세스
 - 주어진 정점 좌표로부터 텍스처 좌표를 계산하는 작업
 - 함수에 따라서는 생성된 텍스처 좌표를 변환.
 - 별도의 파이프라인을 따라 진행
- ◎ 법선벡터: 모델뷰 변환이 가해진 후 시점 좌표계에서 정점의 조명 계산에 사용

프래그먼트(Fragment)

◎ 프래그먼트(Fragment, Pixel Fragment, Fragment Square)

- 화면 화소의 색은 프레임 버퍼에 그 값이 기록되는 순간 결정됨.
- 프레임 버퍼에 기록되기 이전 단계의 화소로서 추상적인 개념
- 화소 위치, 색, 깊이, 텍스처 등 하나의 화소를 채우기 위해 필요한 모든 정보를 지닌 기본단위

◎ 래스터 변환

- 기하 기본요소와 화소 기본요소를 프래그먼트로 변환하는 프로세스

지엘 버퍼



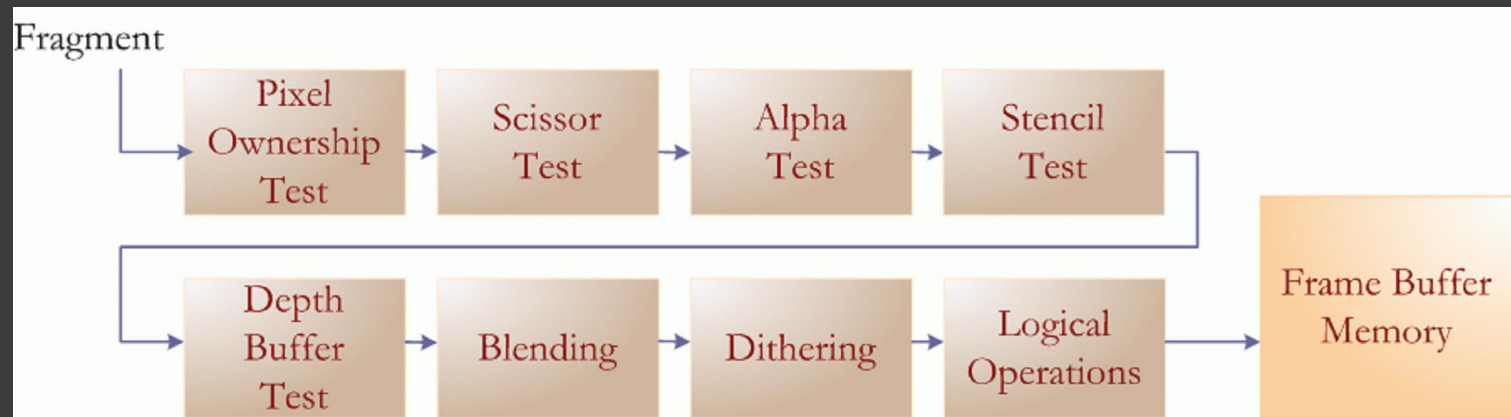
- 프런트 버퍼, 백 버퍼, 보조 버퍼: 컬러 영상의 저장에 사용
- 좌측 버퍼와 우측 버퍼: 입체 영상을 위한 스테레오 버퍼
- 컬러 인덱스: 인덱스 컬러 모드에서 컬러 보기표(CLUT)를 저장
- 보조 버퍼: 직접 화면 디스플레이에 사용할 수는 없음. 임시 저장 장소로 사용 하되 화면에 디스플레이 하려면 프런트 버퍼로 옮겨야 함.

지엘 함수

- ◎ `glutInitDisplayMode(int modes);`
 - 버퍼 활성화: `GLUT_SINGLE`, `GLUT_DOUBLE`, `GLUT_DEPTH`, `GLUT_ACCUM`, `GLUT_STENCIL`
- ◎ `glClearColor(), glClearIndex(), glClearDepth(), glClearStencil(), glClearAccum()`
 - 초기화 값을 지정
- ◎ `glClear(GLbitfield mask);`
 - 실제로 버퍼 내용을 초기화 값으로 초기화:
`GL_COLOR_BUFFER_BIT`, `GL_DEPTH_BUFFER_BIT`,
`GL_STENCIL_BUFFER_BIT`, `GL_ACCUM_BUFFER_BIT`
- ◎ `glDrawBuffer(GLenum mode); glReadBuffer(GLenum mode);`
 - 버퍼 쓰기과 읽기: `GL_FRONT`, `GL_BACK`, `GL_RIGHT`, `GL_LEFT`,
`GL_FRONT_RIGHT`, `GL_FRONT_LEFT`, `GL_BACK_RIGHT`,
`GL_BACK_LEFT`, `GL_AUX1`, `GL_FRONT_AND_BACK`,
- ◎ `glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha); glDepthMask(GLboolean flag); glStencilMask(GLuint mask);`
 - 버퍼 쓰기 금지. 마스킹

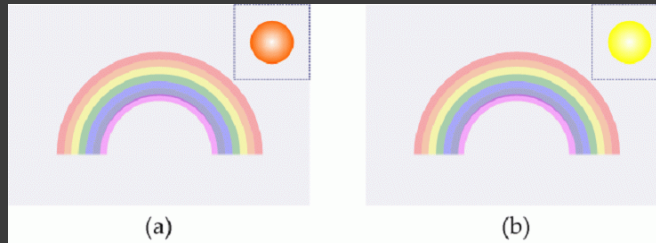
프래그먼트 테스트

- ◎ 테스트 결과
 - 프래그먼트를 제외. 이후 테스트로 입력되지 않음
- ◎ 모든 테스트를 통과한 프래그먼트에 대해서
 - 블렌딩, 디더링, 논리연산이 실행
 - 최종 결과가 프레임 버퍼에 기록됨.

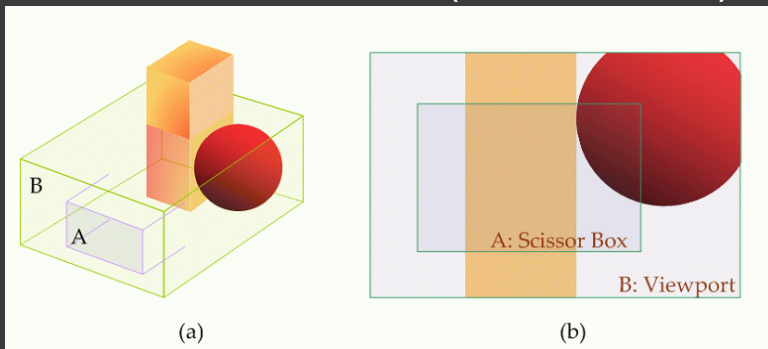


시저 테스트(Scissor Test)

- 그리기 범위를 뷰포트 일부로 제한: 마스크(Mask)

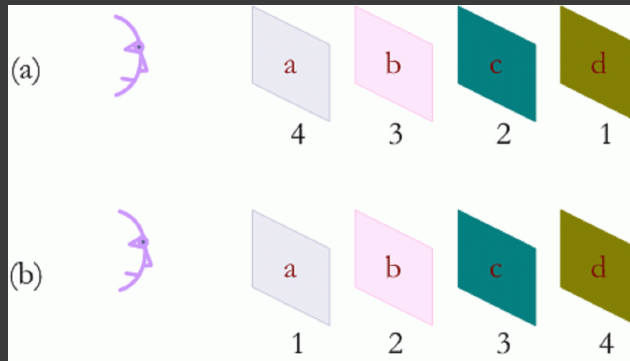


- 넓은 범위의 가시부피 설정: 절단작업은 1번만 일어남
 - 뷰포트 내부에 시저박스를 이동
 - 처리속도 가속화
 - 숨겨진 부분: 가드 밴드(Guard Band)



알파 테스트(Alpha Test)

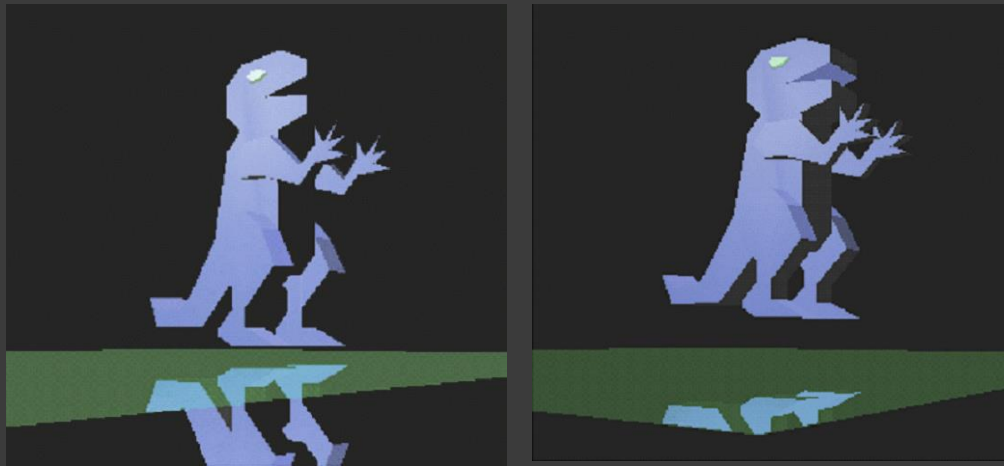
- 알파 값을 기준으로 해당 프래그먼트를 제외
 - `glAlphaTest(GL_LESS, 0.2)`
 - 불투명도가 0.2보다 작은 프래그먼트는 모두 제외
- 투명성 테스트(Transparency Test)



- b)의 순서로 지-버퍼 알고리즘을 가하면 투명성이 적용 안 됨
- 지-버퍼를 활성화한 상태에서 완전히 불투명한 것을 먼저 그림.
- 지-버퍼를 비활성화 한 후 알파가 1.0보다 작은 것을 그림.

스텐실 테스트(Stencil Test)

- ◎ 스텐실 버퍼에 저장된 임의 모양을 기준으로 마스킹
 - cf. 시저 박스: 사각형 모양을 기준
- ◎ 예: 간단한 그림자 그리기



- 그림자 = 원래 물체를 `glScalef(1, -1, 1)`, 이후 아래쪽 반사
- 스텐실 테스트에 의해 바닥 밖의 그림자는 테스트 결과 모두 제외

깊이버퍼 테스트(Depth Buffer Test)

- ◎ 지-버퍼 알고리즘(Z-Buffer Algorithm)이 가해지는 부분
 - glEnable(GL_DEPTH_TEST)에 의해 활성화
- ◎ void glDepthFunc(GLenum func);
 - GL_NEVER, GL_ALWAYS, GL_LESS, GL_LEQUAL, GL_EQUAL, GL_GEQUAL, GL_GREATER, GL_NOTEQUAL
 - 기본값은 GL_LESS: 새로운 z가 이전 z보다 작을 때 갱신

블렌딩(Blending)

- ◎ 프래그먼트와 프레임 버퍼 내용을 혼합(Blending)
 - 입력 프래그먼트 = 소스(Source) 프래그먼트
 - 현재 프레임 버퍼에 있는 화소 = 대상(Destination) 화소
 - 연산 결과는 다시 대상화소에 기록됨.

블렌딩(Blending)

- ◎ (R_s, G_s, B_s, A_s) : 소스 프래그먼트의 색
- ◎ (R_d, G_d, B_d, A_d) : 대상 화소의 색
- ◎ (S_r, S_g, S_b, S_a) : 소스 프래그먼트의 혼합인수
- ◎ (D_r, D_g, D_b, D_a) : 대상 화소의 혼합인수

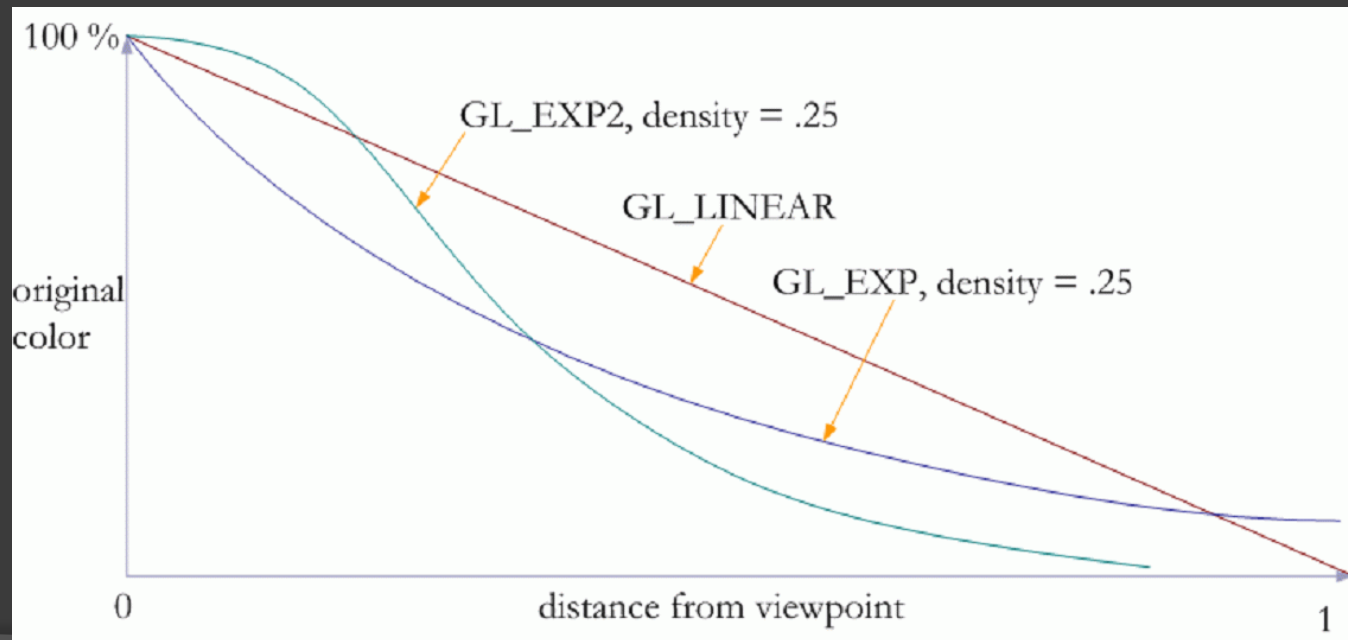
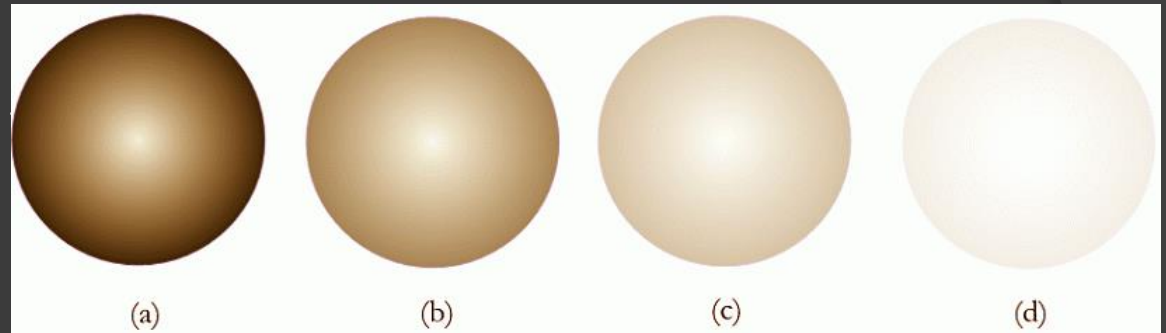
$$(R, G, B, A) = (R_s S_r + R_d D_r, G_s S_g + G_d D_g, B_s S_b + B_d D_b, A_s S_a + A_d D_a)$$

◎ 혼합인수 값에 따른 변환

- (GL_ONE, GL_ZERO)
- (GL_ONE, GL_SRC_ALPHA)
- (GL_SRC_ALPHA, GL_ONE)
- (GL_SRC_ALPHA, GL_DST_ALPHA)



안개효과(Fog Effect)



지엘의 A-버퍼

- ◎ 단일 버퍼로서 복수개의 영상을 누적적으로 저장하는 버퍼
- ◎ 화면 안티-에일리어싱
 - 변위된 영상들을 평균하면 에일리어싱이 약화
- ◎ 필드 깊이 조절
 - 초점거리로부터 물체가 멀어질수록 흐려짐
- ◎ 모션 블러
 - 물체를 이동하면서 `glAccum(GL_MULT, decayFactor)` 함수 호출

