3D Game Programming 06

afewhee@gmail.com

- Texturing
- Addressing
- Filtering
- Multi Texturing
- Blending
- 실습

- Texturing
  - 폴리곤(Polygon) 만으로도 가상 세계를 표현
    - Detail이 높아질 수록 컴퓨터의 자원이 많이 필요
  - 텍스처링
    - 폴리곤을 줄이고 오브젝트의 질감을 높이기 위해 이미지를 폴리곤에 적용하는 매핑(Mapping) 기술
  - 샘플링(Sampling)
    - 조건에 맞는 색상을 이미지에서 추출하는 방법
- 텍스처링 기술
  - Addressing
  - Filtering
  - Multi-Textureing

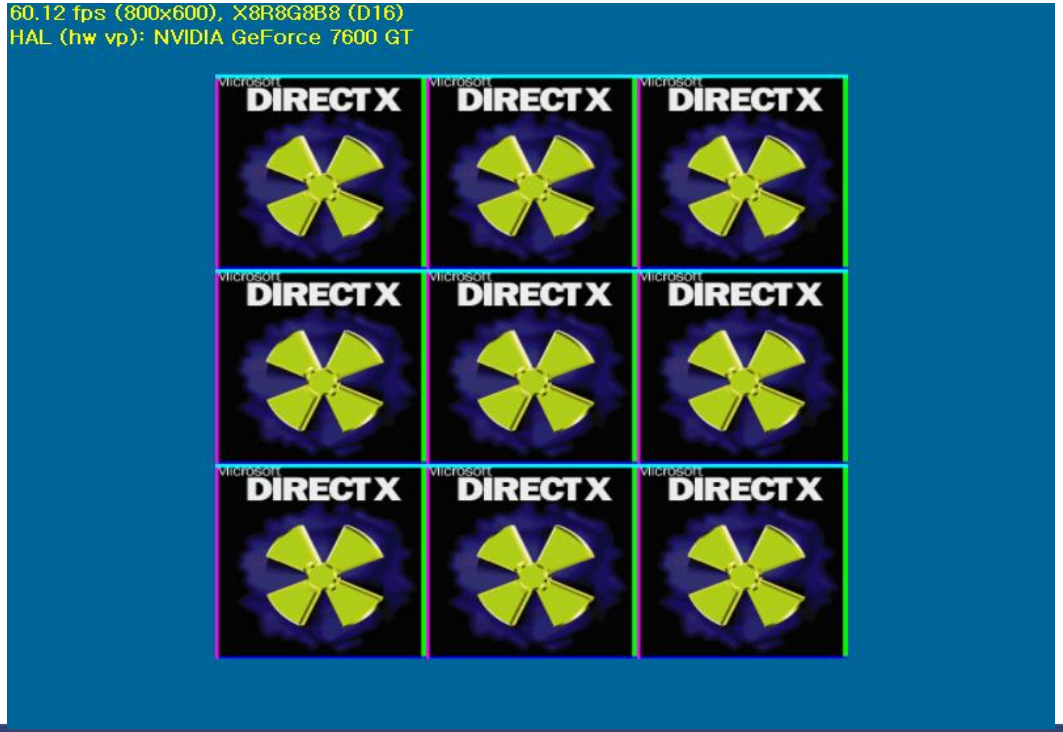
- 텍스처 좌표:
  - Direct3D: UV 좌표계 - y 값이 화면좌표계와 동일(상단-0, 하단-1)
  - OpenGL: ST 좌표계(수학의 오른손 좌표계 사용)
  - 모든 텍스처 좌표는 [0,1]의 범위를 가짐(정규화)

- Addressing
  - 텍스처 좌표의 값이 [0,1] 범위 이외의 값에 대해서 Direct3D가 처리하는 방법

  - 종류
    - Wrap
    - Mirror
    - Clamp
    - Border-color

- Wrap:
  - 1보다 크면 [0,1] 범위 안에 올 때까지 1씩 감소 시켜 결정
    Ex)2.4 → 0.4, 10.5 → 0.5
  - 0보다 작으면 [0,1] 범위 안에 올 때까지 1씩 증가 시켜 결정
    Ex) -1.0 → 0.0, -0.4 → 0.6
  - 실외 지형에서 타일링(Tiling)으로 자주 사용

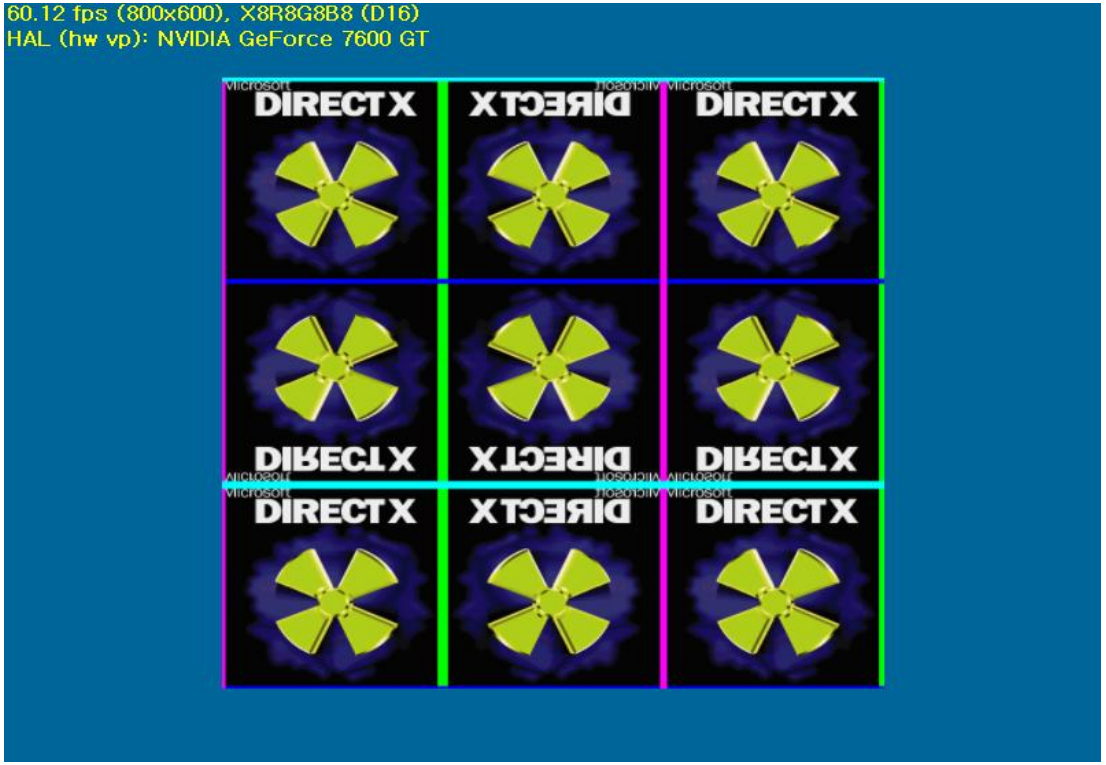
```
pDevice->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DTADDRESS_WRAP);
pDevice->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DTADDRESS_WRAP);
```



- Mirror:
  - Wrap과 비슷하나 거울처럼 텍스처를 적용
  - 1.1→ 0.9, 2.0→0.0

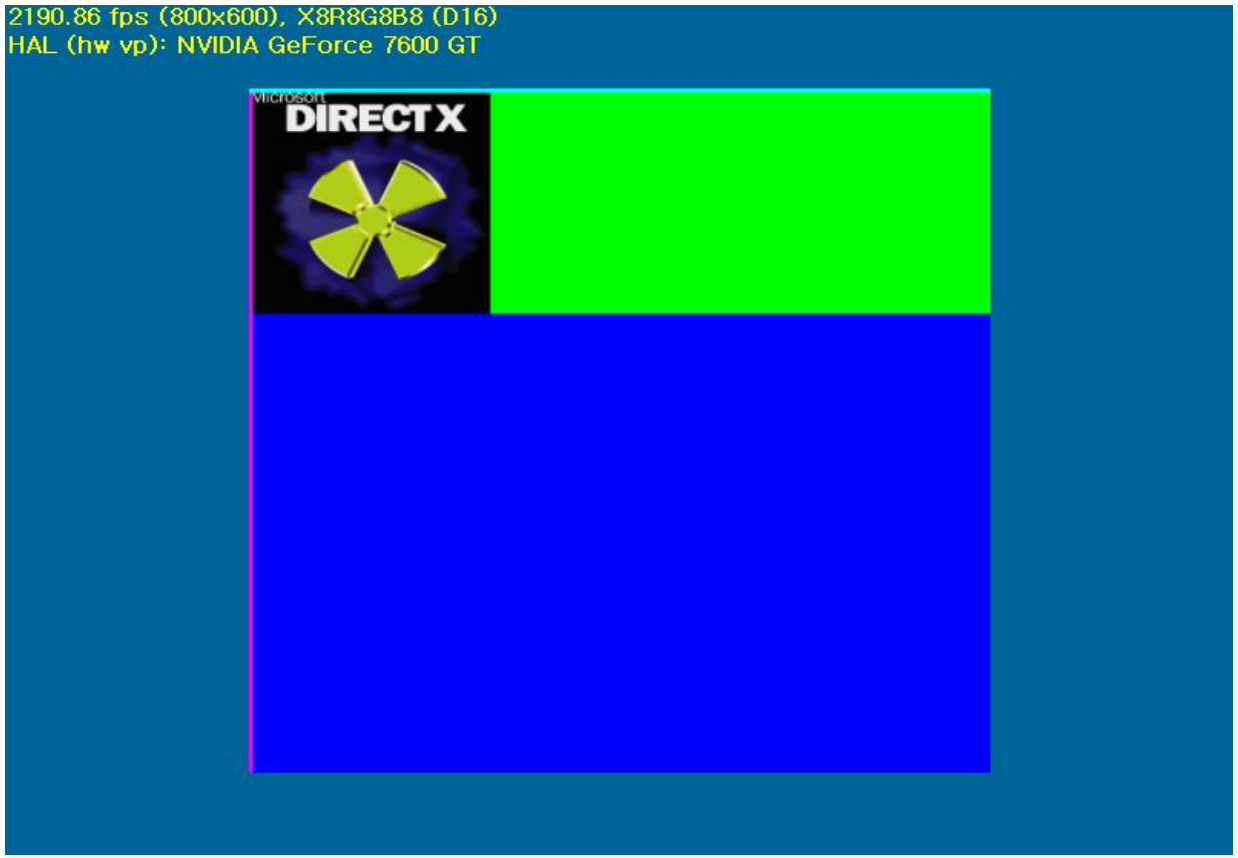
  - 경계가 자연스럽게 이어지는 텍스처 매핑에서 주로 사용

```
pDevice->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DTADDRESS_MIRROR);
pDevice->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DTADDRESS_MIRROR);
```



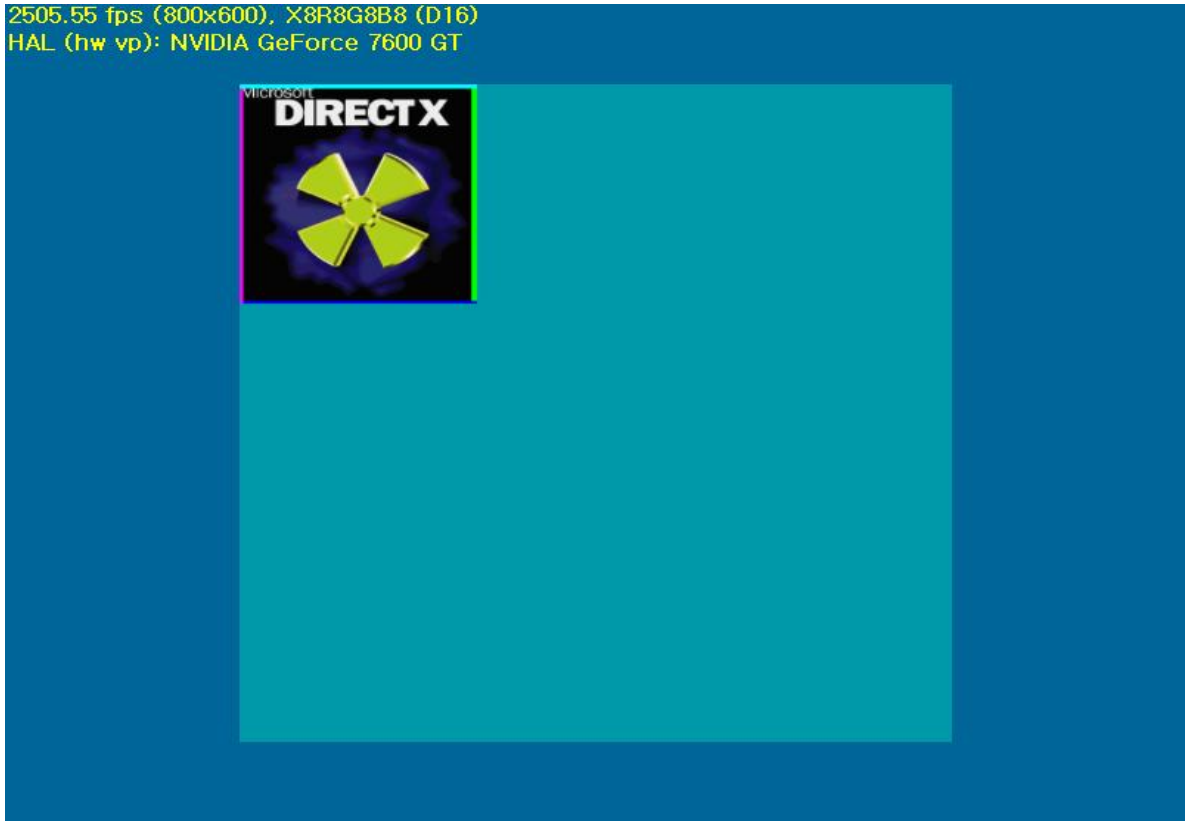
- Clamp:
  - [0,1]범위 밖의 색상은 마지막 픽셀 값으로 결정
  - Clamp모드는 그림자 맵과 같이 불 필요하게 픽셀이 반복 되는 것을 막는데 주로 사용

  - pDevice->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DTADDRESS_CLAMP);
    pDevice->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DTADDRESS_CLAMP);



- Border-Color:
  - 렌더링 머신에 주어진 값으로 [0,1]범위 밖의 색상을 결정

  - pDevice->SetSamplerState(0, D3DSAMP_BORDERCOLOR, 0x0000ffff);
  - pDevice->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DTADDRESS_BORDER);
    pDevice->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DTADDRESS_BORDER);



- 필터링
  - 텍스처를 적용한 폴리곤을 화면에 출력하는 경우 텍스처가 화면에서 확대 되거나 축소되었을 때 픽셀을 보정하는 방법

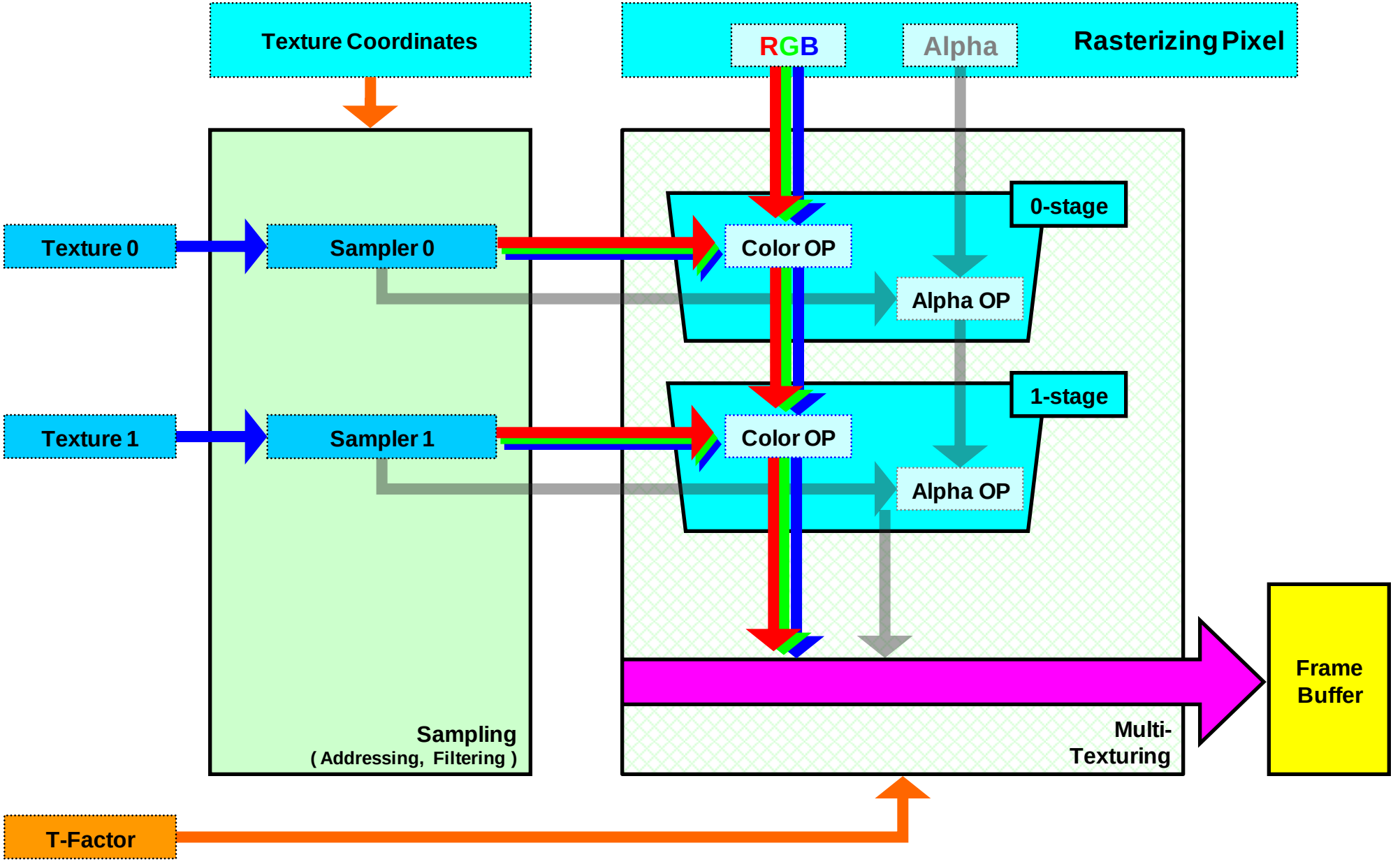
- 필터링 종류
  - Magnification
    - 텍스처가 확대 됐을 때 Sampling 방법 ➔ D3DSAMP_MAGFILTER
  - Minification
    - 텍스처가 축소 됐을 때 Sampling 방법 ➔ D3DSAMP_MINFILTER
  - MIPMAP (Latin: *multum in parvo* -"much in a small space")
    - 카메라에서 멀리 떨어진 객체는 낮은 해상도의 텍스처를 적용하고, 가까이에 있는 객체는 높은 해상도의 텍스처를 적용하기 위해 텍스처를 여러 개의 작은 해상도의 텍스처들을 생성하는 것.

      ➔ D3DSAMP_MIPFILTER
    - 카메라의 거리에 따라 적절한 텍스처를 선택
    - 카메라에서부터 먼 거리의 오브젝트에 높은 해상도의 텍스처를 적용하는 경우 노이즈 발생
    - D3DXCreateTextureFromFileEx () 함수로 밉맵 레벨 조정

- 방법
  - 근접점 샘플링(nearest-point sampling)
    - 해당 UV좌표에서 가장 가까운 픽셀을 샘플링
    - 가장 빠름
    - 텍스처의 경계에서 문제 발생
  - 선형 필터링(Bilinear interpolation filtering)
    - 해당 UV좌표에서 가장 가까운 네 픽셀을 샘플링, 가중치를 적용해서 픽셀을 결정
    - 대부분의 3D 프로그래머가 선호
  - 삼중 필터링(Trilinear Filtering)
    - Direct3D는 MIPMAP이 설정이 되면 연관이 있는 MipMap에 선형 필터링을 계산하고 다시 이를 가중치를 적용해서 최종 픽셀을 설정
  - 비등방형 필터링(AF: Anisotropic Filtering)
    - 폴리곤에 적용된 텍스처와 스크린 평면간의 각도의 차이를 이용해 왜곡을 보정
    - Trilinear Filtering과 비슷하나 텍스처의 가로 세로에 다른 가중치를 적용
    - 가장 우수 가장 느림

- Multi-Texturing
  - 그래픽 파이프라인에서 기하 변환에 의해 만들어진 픽셀, 디바이스의 T-Factor, Sampling한 텍스처 픽셀 등을 이용해서 각각의 스테이지(Stage)를 단계적으로 거치면서 픽셀을 만들어 가는 과정
  - 텍스처는 총 8개의 Stage에 연결 가능
  - 하나의 Stage에는 하나의 텍스처만 연결
  - 픽셀의 RGB와 Alpha에 대한 Multi-Texturing은 독립적으로 처리
  - RGB ➔ Vector Multi-Texturing
  - Alpha ➔ Scalar Multi-Texturing

4. Multi-Texturing (2Stage)



4. Multi-Texturing - 프로그램 작성

- 정점 구조체 설정

```
struct VtxUV
{
    float4      p;      // 위치
    ...
    float       u,v;    // 텍스처 좌표
};
```

- 텍스처 생성: D3DXCreateTextureFromFileEx() 함수 이용
- 디바이스의 스테이지에 텍스처 적용: pDevice->SetTexture(스테이지, 텍스처 포인터);
- 텍스처 어드레스 모드 설정
  - pDevice->SetSamplerState(스테이지, D3DSAMP_ADDRESSU, 어드레스 옵션);
  - pDevice->SetSamplerState(스테이지, D3DSAMP_ADDRESSV, 어드레스 옵션);
  - pDevice->SetSamplerState(스테이지, D3DSAMP_ADDRESSW, 어드레스 옵션);
- 텍스처 필터링 Mag, Min, Mip 설정
  - pDevice->SetSamplerState(스테이지, D3DSAMP_MAGFILTER, 필터 옵션);
  - pDevice->SetSamplerState(스테이지, D3DSAMP_MINFILTER, 필터 옵션);
  - pDevice->SetSamplerState(스테이지, D3DSAMP_MIPFILTER, 필터 옵션);
- 멀티 텍스처링에 대한 스테이지 상태 설정
  - pDevice->SetTextureStageState(스테이지, D3DTSS_COLORARG1, ...);
  - pDevice->SetTextureStageState(스테이지, D3DTSS_COLORARG2, ...);
  - pDevice->SetTextureStageState(스테이지, D3DTSS_COLOROP, ...);
  - pDevice->SetTextureStageState(스테이지, D3DTSS_ALPHAARG1, ...);
  - pDevice->SetTextureStageState(스테이지, D3DTSS_ALPHAARG2, ...);
  - pDevice->SetTextureStageState(스테이지, D3DTSS_ALPHAOP, ...);
- 정점버퍼를 연결해서 렌더링
  - pDevice->SetStreamSource();
  - pDevice->SetFVF();
  - pDevice->DrawPrimitive();

- Alpha Blending
  - 두 픽셀을 가중치에 따라 섞는 방법
  - Pixel의 Alpha 값을 해당 색상의 Weight와 동일
  - Shader 에서 [0,1] 범위 값을 가짐 →1 완전 불투명 , 0: 완전 투명

- 종류
  - 멀티 텍스처링에 의한 블렌딩 → Vertex, Material, Texture Alpha 사용
  - Alpha Blending ➔ Frame Buffer or Render Target Alpha Blending

- Frame Buffer Alpha Blending
  - Back buffer에 쓰여져 있는 픽셀과 새로운 픽셀을 섞는 방법
  - 최종 픽셀 = Source Pixel(새로운 픽셀) ⊗ Source Blend Factor
    + Dest Pixel(이미 저장되어 있는 픽셀) ⊗ Dest Blend Factor

  ⊗ 은 Pixel의 R, G, B에 적용될 연산

  - Default Blend Factor:
    - Source: D3DBLEND_SRCALPHA
    - Dest: D3DBLEND_INVSRCALPHA

Ex)

```
pDevice->SetRenderState(D3DRS_ALPHABLENDENABLE, TRUE);
pDevice->SetRenderState(D3DRS_SRCBLEND, D3DBLEND_SRCALPHA);
pDevice->SetRenderState(D3DRS_DESTBLEND, D3DBLEND_INVSRCALPHA);
```


- 실습
  - Diffuse Map, Detail Map 이미지를 만들어서 32*32 격자에 Multi-Texturing 을 적용하시오.