버택스 셰이더에 의한 Screen Space Billboard Effect

이상원[O]
[O]청강문화산업대학교 게임콘텐츠스쿨
e-mail: one@ck.ac.kr[O]

Screen Space Billboard Effect by Vertex Shader

SangWon Lee[O]
[O]School of Game, ChungKang College of Cultural Industries

● 요 약 ●

본 논문은 실시간 3D 그래픽을 사용하는 Unity 3D나 Unreal Engine 등에서 이펙트를 표현하는 일반적인 방법인 Billboard의 문제점을 보여주고, 범용적이고 가벼운 연산으로 해결할 수 있는 방법을 제시한다.

키워드: 버택스 셰이더 (Vertex Shader), 빌보드 (Billboard), 이펙트 (Effect), 유니티 3D (Unity 3D)

I. Introduction

Fig 1. Billboard Effect

Billboard Effect는 실시간 3D 그래픽 엔진 환경에서 이펙트를 최대한 가볍게 렌더링 함으로써 높은 프레임 레이트(frame rate)를 확보하기 위해 일반적으로 사용되는 방식이다.

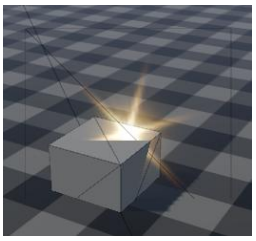


Fig. 2.

언제나 카메라를 정면으로 바라보는 속성의 이 사각형 폴리곤은, 허공에 있으면 문제가 없으나 주변에 다른 물체가 가까이 있으면 사각 평면이라는 사실이 시각적으로 드러나는 문제를 안고 있다.(그림 2)

이 문제는 다양한 방법으로 보완할 수 있는데, 본 논문에서는 최대한 연산이 가벼운 방식으로 접근함으로써 낮은 사양의 기기에서도 적용할 수 있는 방법을 제시한다.

II. The Proposed Scheme

1. 간단한 해결 방법과 한계

[그림 2]의 경우 Billboard를 카메라 방향으로 조금만 이동하더라도 문제가 해결된다. 그리고 이동하는 거리는 사각형 박스의 두께면 충분하다.

다음 코드는 HLSL 버택스 셰이더에서 카메라 방향으로 버택스를 이동하는 방식을 보여준다.

```
half3 viewDir =
normalize(_WorldSpaceCameraPos - worldPos);
worldPos += viewDir * offsetAmount;
```

worldPos는 버택스의 월드 위치이다.
offsetAmount는 이동할 거리이다.

이 방법은 매우 가볍고 직관적이어서 실제로 많은 이펙트 작업에 적용했을 때 대부분 만족스러운 결과를 얻을 수 있었다.(그림 3)

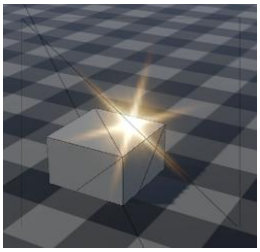


Fig. 3.

그러나 이 방식은 오차가 있는데, offsetAmount 값이 커질수록, 그리고 화면 중앙에서 벗어날수록 오차는 점점 커진다.

빌보드 이펙트가 지면에 근접해서 발생할 경우 거의 카메라 바로 앞까지 버텍스를 이동할 정도로 offsetAmount 값이 커져야하는데, 이런 경우 오차가 매우 심하게 발생한다.

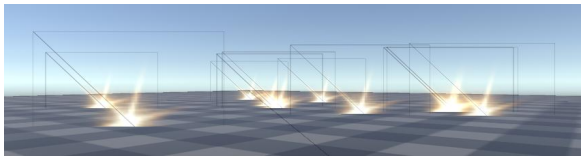


Fig. 4.

그림 4는 지면에 가까운 곳에서 발생한 빌보드 이펙트들의 모습이고 offset을 적용하기 전이다.

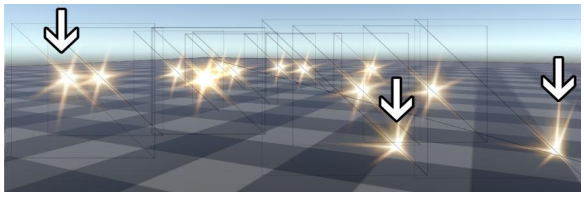


Fig. 5.

그림 5는 offset을 적용한 모습이다. 빌보드가 지면에 가까이 있기 때문에 offsetAmount 값을 크게 적용해야했고 화면 바깥쪽에서 오차가 심하게 발생했다.

오차가 발생하는 이유는 다음과 같다.

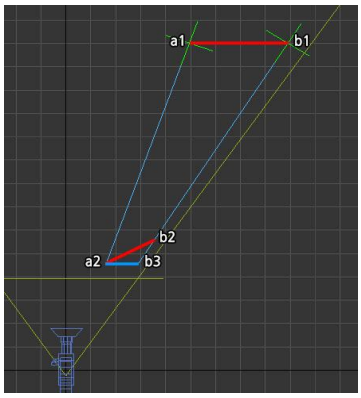


Fig. 6.

a1과 b1은 offset을 적용하지 않은 빌보드의 위치이다. a1과 b1이 카메라를 향해 동일한 거리를 이동하면 a2와 b2의 위치가 된다. a2와 b2 위치의 빌보드는 카메라로부터 비스듬한 각도가 되며 오차가 발생한 결과가 된다. 시각적으로 적절한 위치는 b3이다. 사실 a2 위치도 약간의 오차가 발생한 상태이지만 이해를 돕기 위한 그림임을 참고하기 바란다.

2. 오차 보정

다음 코드는 오차를 보정하기 위한 HLSL 버텍스 셰이더 코드이다. (Unity 3D shader)

```
half3 viewDir =
  normalize(_WorldSpaceCameraPos - worldPos);
half3 viewDirOrth = UNITY_MATRIX_V[2].xyz;
half outer = 1 - dot(viewDir, viewDirOrth);
worldPos += viewDir *
  (offsetAmount + offsetAmount * outer);
```

viewDirOrth는 카메라의 forward 벡터이다.

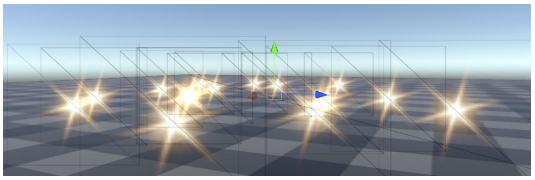
3. 카메라 공간으로 이동

다음 코드는 오차 보정과 카메라 공간으로의 이동을 수행한다.

```
half3 viewDir =
  normalize(_WorldSpaceCameraPos - worldPos);
half3 viewDirInverse =
  normalize(worldPos - _WorldSpaceCameraPos);
half3 viewDirOrth = UNITY_MATRIX_V[2].xyz;
half outer = 1 - dot(viewDir, viewDirOrth);
worldPos = _WorldSpaceCameraPos +
  viewDirInverse * (_ProjectionParams.y +
  _ProjectionParams.y * outer +
  offsetAmount + offsetAmount * outer);
```

_ProjectionParams.y는 카메라의 near plane이다.

카메라 공간을 사용하여 이펙트를 카메라의 near plane 바로 앞까지 근접시킴으로써, 마치 포스트 이펙트의 스크린 스페이스 이펙트와 유사한 결과를 만들었다.



III. Conclusions

본 논문에서 제시한 방법은 버텍스 셰이더에 간단하게 적용할 수 있어서 기존 프로그램 로직을 변경할 필요 없이 셰이더 수정만으로도 원하는 결과를 얻을 수 있다는 장점이 있다. 또한 빌보드 이펙트 뿐만 아니라 모든 폴리곤 연산에도 적용이 가능하기 때문에 응용 범위가 매우 넓다.