

GPU를 이용한 물 표면 실시간 렌더러 구현

이재성, 권덕호, 성만규

계명대학교 컴퓨터공학부 게임모바일공학전공

e-mail:kon9383@naver.com, dau9999@kmu.ac.kr, mksung@kmu.ac.kr,

Realtime Rendering of Water Surface using GPU

JaeSung Lee, Dukho Kwon, Mankyu Sung

Dept of Game & Mobile, Faculty of Computer Engineering, Keimyung University

요 약

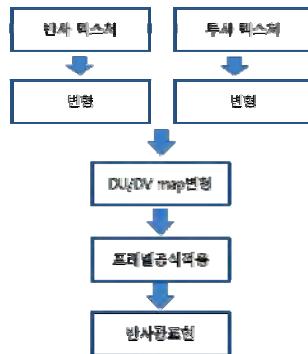
본 연구는 게임을 비롯한 많은 콘텐츠에서 활용하기 위한 GPU기반 사실적 물 애니메이션 기법을 제안한다. 물 표면은 반사 및 투사와 같은 물리적 현상이 일어나며, 시점에 따른 반사와 투사의 정도가 자동적으로 조절되어야 한다. 본 논문에서는 GPU 프레임 버퍼를 이용한 렌더투텍스처 방법을 이용하여 반사 및 투사결과를 텍스처로 저장하였으며, 이 저장된 데이터에 대한 UV좌표 값을 변경함으로써, 자연스러운 물결의 모습을 표현하였다. 또한 투사 및 반사의 정도가 프레넬(Fresnel) 공식을 통해 자동적으로 계산되도록 하였다.

1. 서론

최근 GPU성능의 발전에 따라 실시간 처리가 가장 중요한 게임에서도 자연스럽게 사실적인 3차원 공간에 대한 렌더링이 가능하였다. 하지만, 여전이 물이나 불, 혹은 옷감과 같은 복잡한 객체에 대한 표현은 수작업을 통한 표현이 불가능하며, 수작업을 통해 표현이 가능하더라도 자연스러운 움직임을 표현하는 데에는 한계가 있다. 본 논문에서는 GPU 셰이더를 이용하여 자연스러운 물 표면을 표현하는 기법을 구현하였다[1]. 본 연구에서는 GPU기반의 OpenGL 셰이더를 이용하였으며, 렌더투텍스처를 이용하여 투사와 반사를 각각 하나의 텍스처로 저장한 후, 이를 블렌딩하여 물 표면을 나타내었다. 또한, 시간데이터를 이용하여 텍스처 좌표값을 변형함으로써, 물결 표현을 하였으며, 반사광(specular light)를 이용하여 물결 위의 광원의 모습을 렌더링 하였다.

2. 알고리즘

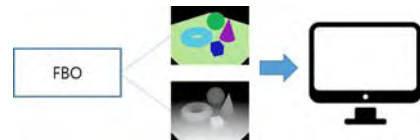
전체 알고리즘은 그림1의 순서에 따른다. OpenGL에서 제공하는 렌더투텍스처 기법을 이용하여 반사 텍스처와 투사 텍스처를 각각 생성하며, 이를 입력된 시간데이터를 이용하여 텍스처 좌표를 변형한다. 후에 카메라의 위치를 고려하여 프레넬 공식을 이용한 반사 텍스처의 가중치와 투사 텍스처의 가중치를 계산하며, 이를 블렌딩하여 물 표면을 나타낸다. 이 후에 반사광 색상을 추가함으로써 최종 렌더링 결과를 얻는다.



(그림 1) 알고리즘 순서도

2.1 프레임 버퍼

반사 및 투사 텍스처는 OpenGL의 프레임버퍼오브젝트(FrameBufferObject)를 이용하여 구하였다. 프레임 버퍼는 화면상의 나타내는 컬러정보와 텍스 정보들을 동시에 하나의 텍스처 형태로 저장 가능하도록 해주며, 실제로 모니터에 나타나지 않고도 버퍼에 저장할 수 있다. 그림2는 컬러와 텍스 텍스처를 가진 프레임버퍼오브젝트를 나타낸다.



(그림 2) 프레임버퍼오브젝트

2.2 절단 플레인을 이용한 투사 및 반사 텍스처

반사와 투사되는 이미지를 얻기 위해서는 렌더링을 수행할 때 물 표면을 기준으로 아래 부분과 위 부분을 구분하여 따로 따로 렌더링 해야 한다. 즉, 반사이미지를 얻기 위해서는 물 표면 위 부분만을 렌더링하고, 투사 이미지를 얻기 위해서는 물 표면 아래 부분만을 렌더링 하여 프레임 버퍼오브젝트에 저장한다.

OpenGL에서는 gl_ClipDistanc라는 셰이더 내장 변수를 제공해 준다. 이를 위해 평면의 방정식을 이용한다.

$$ax + by + cz + d = 0 \quad (1)$$

이 방정식에서 (a, b, c)는 평면의 normal이며, d는 원점까지의 거리를 나타낸다. 대 부분 평면은 horizontal하기 때문에 (a, b, c) = (0,1,0)이며 d는 물 표면의 높이를 의미한다. 이 관계를 이용하면 gl_ClipDistace값은 쉽게 물 표면을 나타내는 사각형 각 정점의 월드 좌표 p_{world} 와 평면방정식의 4개의 coefficient 인 $p=(a,b,c,d)$ 와의 내적을 통해 계산된다. ($p, v_{world} \in R^3$)

$$gl_ClipDistace = p \cdot v_{world} \quad (2)$$

반사와 투사는 바로 p값을 조절함으로써 쉽게 설정할 수 있다. 이 때 반사 텍스처를 위해서는 카메라의 위치를 물 속 안으로 바꾸어야 한다. 카메라 위치를 C라고 하고 물표면의 y축값을 d라고 한다면 C의 높이는 다음과 같이 계산된다.

$$\begin{aligned} \Delta &= 2(C_y - d) \\ C_y &= C_y - \Delta \end{aligned} \quad (3)$$

2.3 텍스처 변형

윗 단계에서 얻은 투사와 반사텍스처는 텍스처 좌표에 대한 쉬프팅을 통해 변형한다. 이 변형단계는 물 표면의 어를거림을 표현하기 위함이다. 본 연구에서는 DU/DV맵을 이용하여 수정하였다. DU/DV map는 텍스처 좌표를 바꾸기 위한 레퍼런스 테이블을 의미한다. 반사와 투사의 텍스처 좌표를 각각 $UV_{reflection}$ 과 $UV_{refraction}$ 이라고 하고, DU/DV map의 값을 DV라고 한다면, 두 텍스처 좌표는 아래와 같이 변형된다.

$$UV_{reflection} += DV \quad (4)$$

$$UV_{refraction} += DV$$

2.4 프레넬 계산

프레넬 값은 현재 카메라의 시점 방향을 이용하여 물 표면에 얼마나 반사와 투사이미지가 중요한지에 대한 가중치를 구하기 위해 필요하다. 이를 위해 물 표면의 normal 벡터(N)와 시점 벡터(V)에 대한 각도 값을 이용한다.

프레넬 값 F는 다음과 같이 계산된다. 이 공식에서 r는 반사의 정도를 강조하기 위한 상수 값이다.

$$F = (N \cdot V)^r$$

2.5 반사광 추가

마지막으로 대표적으로 많이 사용되는 풍 광원 모델의 반사광을 추가하여 물 표면에 나타내는 하이라이트를 표현한다. 자세한 반사광에 대한 구현은 []을 참고한다.

2.6 최종 픽셀 컬러

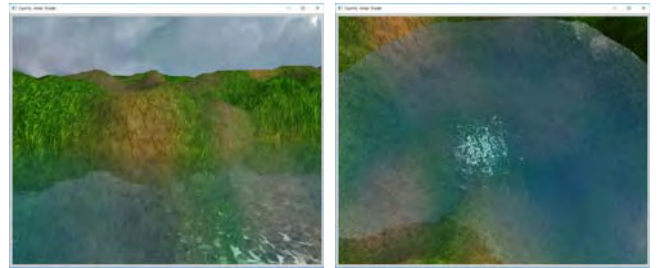
최종 계산된 픽셀 컬러(Final)는 반사 텍스처를 이용한 색상($I(UV_{reflection})$)과 투사 텍스처를 이용한 색상($I(UV_{refraction})$)을 프레넬 값 F 가중치로 활용하여 블렌딩 하며, 이 후에 반사광 L 추가하여 최종 픽셀값을 계산한다.

$$Final = mix(I(UV_{reflection}), I(UV_{refraction}), F) + L$$

3. 실험

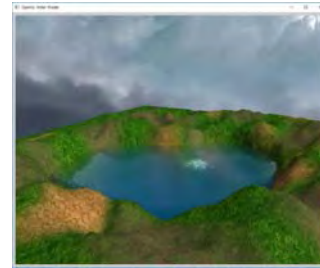
위에서 설명한 알고리즘을 Visual Studio 2013을 이용하여 Windows 7에서 구현하였다. 그림3과 그림4는 각각 반사 텍스처와 투사 텍스처를 실시간 캡처한 스크린 샷이며, 그림 5는 최종 렌더링 결과를 나타낸다. 윈도우 생성

을 위해 FLTK UI라이브러리를 이용하였으며, 지형 모델 데이터를 이용하여 렌더링 하였다.



(그림 3) 반사 텍스처

(그림 4) 투사 텍스처



(그림 5) 최종 렌더링 결과

물 표면은 자연스럽게 반사와 투사의 모습을 나타냈으며 모든 렌더링은 30프레임 이상의 실시간 속도를 나타내었다. 또한, 시점 방향의 변화에 따라 반사와 투사의 정도가 바뀔을 확인하였다.

4. 결론

본 논문은 실시간으로 물 표면을 표현하기 위한 OpenGL를 렌더러를 구현하였다. 현재의 물 표면에 대한 표현은 순수하게 텍스처를 이용하였으므로, 실제 사실적인 유체 애니메이션을 포함하지 못했다. 컴퓨터 그래픽스 연구에서는 유체 시뮬레이션에 대한 많은 연구가 되어 있으며[2], 이를 활용하여 추후 연구를 진행할 예정이다.

Acknowledgment

본 논문은 교육부와 한국연구재단의 대학특성화사업(CK-1)의 지원을 받아 수행된 연구 결과입니다.

참고문헌

- [1] David Wolf, OpenGL 4.0 Shading Lanuage Cookbook, PACT publishing, 2011
- [2] Miles Maclin, Matthia Muller, Position Based Fluid, ACM TOG vol.32 no.4