오목한 물체의 그림자를 포함하는 그림자 텍스처

류태규[O] 오경수
숭실대학교 미디어학과
{mordor[O], oks}@ssu.ac.kr

A shadow texture to represent the shadow including a non-convex object

Taegyu Ryu[0] Kyungsoo Oh
Dept. of Media, Soongsil University

요 약

본 논문은 오목한 물체의 그림자 표현이 가능한 그림자 텍스처를 생성하기 위해, 기존의 그림자 맵 (shadow map) 방법으로 그림자 정보를 생성하고 아틀라스 텍스처 (atlas texture)에 저장하는 방법을 제안한다. 기존 그림자 텍스처 방법은 그림자 정보를 이미지 텍스처에 저장하므로 그래픽스 하드웨어의 색상 처리 기능을 사용하여 고품질의 그림자를 효율적으로 표현하는 장점이 있지만 오목한 물체의 그림자를 표현하지 못하고 그림자 텍스처 생성에 많은 시간이 소요되는 문제점을 가지고 있다. 실험 결과, 새로운 방법은 그림자 텍스처의 문제점을 해결하고 고품질의 그림자를 표현하는 것을 알 수 있다.

1. 서 론

그림자는 빛이 사물에 가려 도달 하지 못하는 곳에 발생한다. 사람은 실생활에서 그림자가 있는 풍경에 익숙하다. 그렇기 때문에 그림자가 있으면 영상의 사실성이 높아지고 물체간의 공간적 관계를 알기 쉽게 된다. 하지만 실시간 그래픽스 프로그램에서의 그림자 표현은 높은 연산 부담으로 인해 흔하지 않다. 실시간 그림자 표현 방법으로는 그림자 맵(shadow map), 그림자 볼륨(shadow volume), 그림자 텍스처(shadow texture) 등이 있다.

그림자 맵 방법은 광원으로부터 장면의 물체까지의 최단 거리를 그림자 맵에 저장하고 렌더링 대상의 광원까지의 거리와 그림자 맵에 저장된 거리를 비교하여 그림자 여부를 판단한다 [1]. 하지만 텍스처에 거리를 저장하기 때문에 색상 처리를 위한 그래픽스 하드웨어의 기능들을 사용할 수 없다.

그림자 볼륨 방법은 물체에 의해 빛이 가려진 영역을 기하정보로 표현하고 이를 사용하여 그림자를 생성한다 [2]. 기하정보를 기반으로 하므로 샘플링에 의한 화질 저하 문제가 없는 방면 실제 장면의 기하 정보 보다 더 많은 그림자 기하정보가 추가 될 수 있기 때문에 렌더링에 심한 부하를 가져오는 문제가 있다.

그림자 텍스처 방법은 그림자 정보가 저장되어 있는 이미지 텍스처를 렌더링 대상에 입히는 방법이다 [3]. 텍스처에 그림자 정보를 색상 형태로 저장 하므로 그래픽스 하드웨어 에서 제공하는 이미지 처리 기능(입맵, 텍스처 샘플링 등)을 사용하여 높은 화질의 그림자를 표현 할 수 있다. 하지만 기존의 그림자 텍스처 방법은 오목한 물체의 그림자를 표현 할 수 없으며 그림자 텍스처를 생성하기 위해 많은 시간이 걸린다.

본 연구는 그림자 텍스처 방법을 개선하여 오목한 물체의 그림자를 표현 할 수 있도록 하며 선형 시간에 그림자 텍스처를 생성하는데 목적이 있다.

3D 모델의 오목한 부분에 발생하는 그림자를 그림자 텍스처에 표현 하기 위해서 아틀라스 텍스처(atlas texture) 형식의 그림자 텍스처를 사용 했으며 기존 그림자 맵 방법으로 그림자 텍스처의 그림자 정보를 생성 하므로 선형 시간에 그림자 텍스처를 생성 할 수 있게 되었다.

2. 오목한 물체의 그림자를 포함하는 그림자 텍스처 생성과 이를 사용한 렌더링

제안 하고자 하는 그림자 텍스처 방법은 그림 1과 같이 2단계의 생성 과정과 1단계의 렌더링 과정으로 이루어 진다. 우선 광원을 시점으로 그림자 맵을 생성하고 그림자 맵을 참조하여 장면의 3D 모델 별로 그림자 텍스처를 생성한다. 마지막으로 그림자 텍스처를 사용하여 렌더링 한다. 광원이나 물체의 이동이 없는 경우 2개의 생성 단계는 생략될 수 있다.

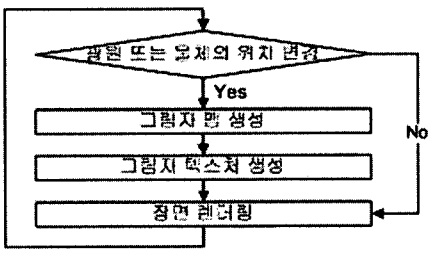


그림 1. 전체 렌더링 순서도

2.1 그림자 맵 생성

그림자 맵은 일반적인 그림자 맵 알고리즘과 같이 광원과 장면의 3D 모델간 최단 거리를 저장하는 텍스처다. 그림자 맵을 생성하기 위해 광원을 시점으로 하여 Z-버퍼 알고리즘을 수행한다. 그 결과, 깊이 버퍼에는 광원과 가장 가까운 모델의 거리가 저장된다.

2.2 그림자 텍스쳐 생성

그림자 텍스처는 텍셀에 그림자 정보를 저장한 이미지 텍스처다. 기존의 그림자 텍스처 방법은 3D 모델의 그림자 정보를 저장하는 텍셀을 결정 하기 위해 광원을 투영 점으로 하는 평면 원근 투영법을 사용 한다. 하지만 이 방법은 올바른 그림자 정보를 저장 할 수 없다. 예를 들어 그림 2와 같이 모델의 B, C는 텍셀 A 에 투영된다. 이때, B는 그림자가 아니고 C는 그림자이기 때문에 텍셀 A는 올바른 그림자 정보를 저장 할 수 없다.

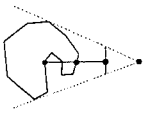


그림 2. 평면 원근 투영에 의해 만들어 진 텍셀은 모델의 여러 부분에 해당된다.

그림 2의 문제는 3D 모델의 모든 표면이 일대일 대응하는 텍셀을 갖도록 아틀라스 텍스처를 사용하여 해결한다. 아틀라스 텍스처는 그림 3과 같이 텍스처 아틀라스(texture atlas) [5]를 사용하여 3D 모델의 전개도 모양으로 텍스처 좌표가 설정된 이미지 텍스쳐이다. 그림자 영역에 대응하는 텍셀은 그림자 색으로, 그렇지 않은 텍셀은 조명 색으로 지정하면 그림자 텍스처가 된다. 이 텍스처를 3D 모델에 매핑하면 올바른 그림자가 표현된다. 이와 같은 그림자 텍스처는 그래픽스 하드웨어로 생성한다.

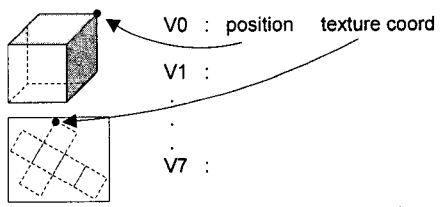


그림 3. 아틀라스 텍스쳐 (atlas texture)

그림 4 는 그림자 텍스처를 그래픽스 하드웨어로 렌더링 하는 과정이다. 3D 모델의 각 정점에는 3D위치 값과 아틀라스 텍스쳐 좌표 값이 저장 되어 있다고 가정 한다. 정점 쉐이더는 아틀라스 텍스쳐 좌표를 위치 값으로 출력 하고 정점의 3 차원 위치 값은 텍스쳐 좌표 값으로 출력한다. 따라서 픽셀 쉐이더는 픽셀에 해당하는 3 차원 위치 값을 텍스쳐 좌표로 전달 받게 된다.

픽셀 쉐이더는 입력된 world 좌표계의 3 차원 위치를 그림자 맵의 좌표계로 변환한다. 변환된 위치의 z(깊이) 값은 그림자 맵에서 얻은 z 값과 비교한다. 그림자 맵에는 광원과 가장 가까운 3D 모델의 위치가 저장 되 있으므로 거리가 더 먼 경우 그림자 색을, 반대의 경우 조명 색을 출력한다.

2.3 그림자 텍스처를 사용한 장면 렌더링

정점에 저장된 위치와 아틀라스 텍스처 좌표 값을 사용하여 그림자 텍스처를 매핑하고 3D 모델을 렌더링 한다. 그림자 텍스처는 하드웨어 밉맵(mipmap)이 적용되어 축소(minification) 시 그림자의 경계 부분에 발생하는 계단 현상이 완화된다.

3. 구현 및 실험 결과

오목한 물체의 그림자를 포함하는 그림자 텍스처 방법은 DirectX 9.0, HLSL, VC++을 사용하여 구현 되었고 펜티엄 4 3.2Ghz, 1GB RAM, ATI 9800XT 시스템에서 실험 하였다. 사용된 모델은 스텐포드 버니의 지오메트리 이미지 모델이다.

3D 모델의 지오메트리 이미지는 모델 표면상의 위치와 일대일 대응하는 이미지 픽셀에 그 위치 정보를 저장한 이미지이다. GU의 지오메트리 이미지 생성 방법 [4]는 3D 모델의 표면 모양 변화와 모델의 절개를 최소화는 장점이 있다.

지오메트리 이미지는 아틀라스 이미지로 볼 수 있기 때문에 3D 모델 정점의 지오메트리 이미지 상 좌표를 아틀라스 텍스쳐 좌표로 사용할 수 있다.

그림 5. 오목한 물체의 그림자를 표현하는 그림자 텍스처 결과

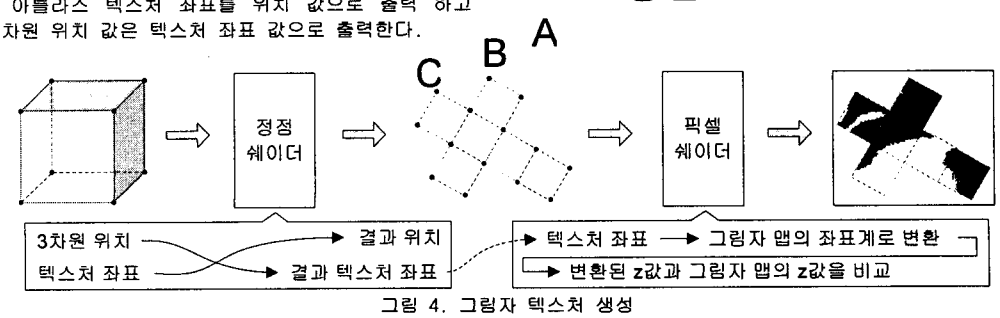
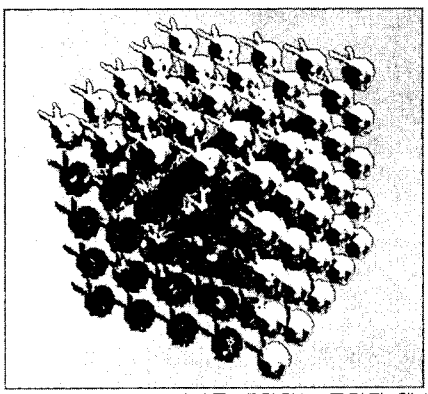


그림 4. 그림자 텍스처 생성

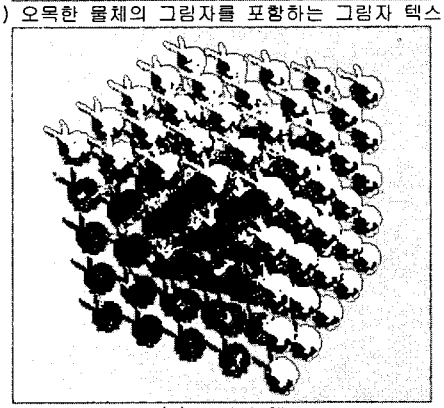
그림자 텍스처를 생성하기 위해 사용된 그림자 맵의 크기는 512x512 이며 그림자 맵 방법에 의한 결과 역시 동일한 그림자 맵을 사용한다.

그림 5는 오목한 물체의 그림자를 포함 하는 그림자 텍스처 방법으로 렌더링 한 결과 이다. 사용된 3D 모델은 257x257 크기의 지오메트리 이미지로 66049개의 정점과 131072개의 면을 갖는다. 그림자 텍스처의 크기는 3D 모델의 크기와 비슷한 256x256 이다.

그림 6. 오목한 물체의 그림자를 표현하는 그림자 텍스처(좌측) 와 그림자 맵(우측)



(a) 오목한 물체의 그림자를 포함하는 그림자 텍스처



(b) 그림자 맵

그림 7. 125개의 모델 렌더링 결과

그림 6과 그림 7은 그림자만을 렌더링한 결과이다. 자기 그림자를 표현 하는 그림자 텍스처 방법은 축소 시에 밉맵이 적용되어 계단 현상이 줄어든 것을 확인할 수 있다.

그림 6에서 사용된 3D 모델이나 그림자 텍스처의 크기는 그림 5와 동일 하고 그림 7에서 사용된 3D 모델은 257x257 크기의 원본을 64x64 크기로 축소하여 만든 저해상도 지오메트리 이미지이다. 장면에서 이 저해상도 3D 모델은 125개가 사용됐으며 각각 4096개의 정점과 7938개의 면을 갖는다. 또한 3D 모델 별로 64x64크기의 그림자 텍스처를 사용한다.

표 1. 각 결과들의 그림자 표현 방법에 의한 수행 속도

장면	그림자 텍스처	그림자 맵
그림 6	64.53(127.50)	65.58(127.32)
그림 7	12.33(27.04)	13.44(27.06)

* 단위 : fps
* 괄호 안의 값은 광원이 고정된 상태에서의 속도

표 1은 그림 6과 그림 7의 수행 속도를 그림자 표현 방법에 따라 측정하고 정리 한 것이다. 추가로 광원이 고정되어 매번 그림자 맵과 그림자 텍스처를 생성하지 않아도 되는 상황의 결과를 측정 하였다. 그림자를 텍스처로 처리하는 방법은 그래픽스 하드웨어에서 충분한 가속을 받을 수 있기 때문에 그림자 맵 방법에 근접한 속도를 보이고 있다.

4. 결론 및 향후 연구

그림자 텍스처는 그래픽스 하드웨어에 적합한 방법이지만 오목한 물체의 그림자를 표현 하는데 어려움이 있으며 그림자 텍스처의 생성에 많은 시간이 소요된다. 본 연구에서는 실험 결과와 같이 아틀라스 텍스처 형식을 사용하여 오목한 물체의 그림자를 표현하며 선형 시간에 그림자 텍스처를 생성한다. 이렇게 생성된 그림자 결과는 밉맵이 적용되어 축소 시 계단 현상이 완화 되는 것을 확인 할 수 있다.

향후에는 그림자 텍스처에 부드러운 그림자(soft shadow)를 구현하고 그림자 맵으로 그림자를 판단할 때 발생하는 화질의 저하를 완화 하기 위해 [6]의 "Adaptive shadow maps" 와 같은 방법을 적용 하는 연구가 필요하다.

참고 문헌

[1] Lance Williams, "Casting curved shadows on curved surfaces", Computer Graphics (Proc. of SIGGRAPH 78), 12(3):270- 274, 1978.

[2] Franklin. C. Crow, "Shadow algorithms for computer graphics", Computer Graphics (Proc. of SIGGRAPH 77), 11(2):242- 248, 1977.

[3] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, Paul Haeberli, "Fast shadows and lighting effects using texture mapping", ACM SIGGRAPH Computer Graphics, v.26 n.2, p.249-252, July 1992.

[4] Xianfeng Gu, Steven J. Gortler, Hugues Hoppe, "Geometry images", Proceedings of the 29th annual conference on Computer graphics and interactive techniques, July 23-26, 2002.

[5] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, Jérome Maillot, "Least squares conformal maps for automatic texture atlas generation", ACM Transactions on Graphics (TOG), v.21 n.3, July 2002.

[6] Randima Fernando, Sebastian Fernandez, Kavita Bala, Donald P. Greenberg, "Adaptive shadow maps", Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p.387-390, August 2001.