

# 여러 장의 Depth-Map 을 이용한 반투명 물체의 실시간 렌더링

이재영\*, 이관행\*

\*광주과학기술원 기전공학과

e-mail : [jylee81@gist.ac.kr](mailto:jylee81@gist.ac.kr)

## Real-Time Rendering of Translucent Objects using Multiple Depth-maps

Jae-Young Lee\*, Kwan-Heng Lee\*

\*Dept. of Mechatronics, Gwangju Institute of Science and Technology

### 요 약

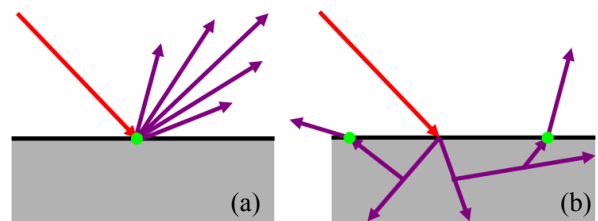
반투명 물체(Translucent Object)는 불투명한 물체와는 달리 물체 내부에서 산란이 일어난다. 따라서 반투명 물체의 한 표면(Surface)을 렌더링하기 위해서는 그 표면의 정규 벡터뿐만 아니라 그 표면의 주변 기하 정보가 필요하다. 그러나 그래픽 하드웨어 구조는 불투명한 표면의 렌더링에만 적합하게 설계 되었기 때문에, 반투명 물체의 실시간 렌더링의 구현에는 많은 제약이 따른다. 이 논문에서는 영상 공간 접근 방법(Image Space Approach)의 연장선에서 기존의 방법론이 가지고 있는 한계점을 해결함으로써, 반투명 재질의 실시간 렌더링을 위한 효과적인 방법을 제시한다.

### 1. 서론

일반적으로 물체는 그 재질의 성질에 의해 불투명(Opaque), 반투명(Translucency)으로 나눌 수 있다. 불투명, 투명 물체와는 달리 물체 내부에서 산란이 일어나는 물체를 반투명 물체라고 한다. 우리 주변에서 흔히 볼 수 있는 우유와 같은 혼합물이나 양초, 대리석, 사람의 피부와 같은 물체들은 모두 반투명한 성질을 띤다[1]. 불투명 물체는 표면 위 산란(Surface Scattering)만 일어나며 한 지점에서 방출되는 빛의 양은(Light Intensity)은 표면의 정규 벡터(Surface Normal), 관측자의 시선, 입사되는 빛의 각도 등 세가지 요인에 의해서만 결정된다(그림 1 - (a)). 반투명 물체의 경우 빛이 표면을 투과한 이후 표면 아래의 산란(Subsurface Scattering)이 일어나므로 빛이 방출되는 표면의 주변 기하 정보가 필요하다(그림 1 - (b)). 따라서 이러한 반투명 물체를 표현하기 위해서는 기존의 폰 라이팅 모델(Phong Lighting Model)과 같은 불투명한 물체의 반사 모델(Reflectance Model)을 반투명 물체에 적용할 수 없다[1][2].

반투명 재질의 렌더링에 관한 연구는 비실시간 렌더링에서부터 이루어져 최근 실시간 렌더링을 구현하

고자 하는 연구가 이루어지고 있다. 일반적으로 그래

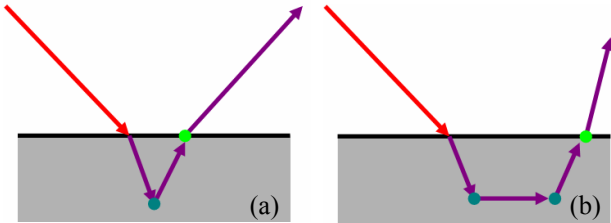


(그림 1) 표면 산란의 두 종류. (a) 표면 위 산란(Surface Scattering), (b) 표면 아래 산란(Subsurface Scattering)

픽 하드웨어에서 반투명 물체에 대한 실시간 렌더링은 구조적으로 불가능하다. 하지만 몇 가지 제한된 조건 아래에서는 성과가 이루어졌다. 본 논문에서는 최신 프래그래머블 그래픽스 하드웨어(Programmable Graphics Hardware)의 구조에 알맞은 반투명 물체의 실시간 렌더링 방법론을 제시한다.

2. 관련 연구 및 배경

빛의 전달(Light Transport)에 대한 물리학 지식을 바탕으로 반투명 물체에 대한 비실시간 렌더링(Off-Line Rendering)을 위한 방법론은 1970 년도부터 꾸준히 제시되었다 [3][4]. 2001 년 이후 표면 아래의 산란을 효과적으로 표현하기 위해서 쌍극자 근사법(Dipole Approximation)이 적용되었다[3]. 이러한 비실시간 렌더링을 위한 연구를 기반으로 최근 실시간 렌더링을 위한 노력이 있다.



(그림 2) 산란 횟수에 따른 표면 아래 산란의 두 종류. (a) 단일 산란 (Single Scattering), (b) 복수 산란 (Multiple Scattering).

2.2 쌍극자 근사법(Dipole Approximation)

표면 아래의 산란을 계산의 편의를 위해 단일 산란(Single Scattering)과 복수 산란(Multiple Scattering)으로 나눌 수 있다[3]. 단일 산란은 기존의 광선추적법(Ray Tracing)을 적절히 사용하여 표현한다[5]. 복수 산란은 쌍극자 근사법(Dipole Approximation)에 의해서 근사적으로 표현한다[3](표 1).  $S$  는 단위 면적 당 산란되는 빛의 양을 나타내고,  $\sigma_a$  는 물질의 흡수 계수(absorption coefficient),  $\sigma'_s$  는 산란 계수(reduced

$$S(x_i, w_i; x_o, w_o) = \frac{1}{\pi} F_t(\eta, w_o) R_d(x_i, x_o) F_t(\eta, w_i).$$

$$R_d(x_i, x_o) = \frac{\alpha'}{4\pi} \left[ z_r(1 + \sigma_s) \frac{e^{-\sigma_r}}{s_r^3} + z_v(1 + \sigma_s) \frac{e^{-\sigma_v}}{s_v^3} \right]$$

$$z_r = 1/\sigma'_t, \quad z_v = z_r + 4AD, \quad \alpha' = \sigma'_s / \sigma'_t$$

$$s_r = \|x_r - x_o\|, \text{ with } x_r = x_i - z_r \cdot N_i$$

$$s_v = \|x_v - x_o\|, \text{ with } x_v = x_i + z_v \cdot N_i$$

$$A = \frac{1 + F_{dr}}{1 - F_{dr}}, \quad D = 1/3\sigma'_t, \quad \sigma = \sqrt{3\sigma_a\sigma'_t}$$

$$F_{dr} = -\frac{1.440}{\eta} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta$$

$$\sigma'_t = \sigma_a + \sigma'_s$$

<표 1> 쌍극자 근사법

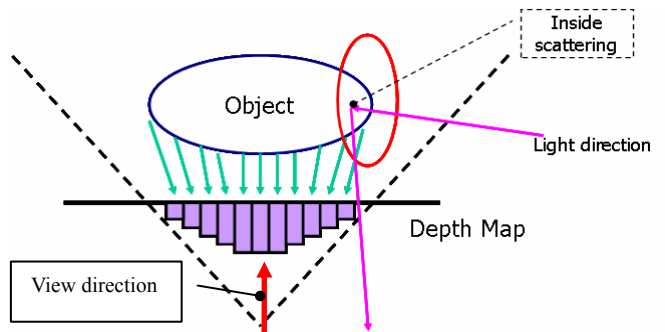
scattering coefficient),  $\eta$  는 굴절률,  $F_t$  는 표면을 투과한 빛의 양을 나타내는 프레넬 항(Fresnel factor)이다 [3]. 쌍극자 근사법을 이용하면 빛이 입사되는 지점과 방출되는 지점의 거리가 산란되는 빛의 양을 결정한다. 이 논문에서는 내부 산란이 매우 많이 일어나는 물체로 가정하여 단일 산란에 대하여 고려하지 않고 복수의 산란만으로 반투명 물체를 표현한다[5].

2.3 반투명 물체의 실시간 렌더링

반투명 물체의 실시간 렌더링을 위해서 PRT(Precomputed Radiance Transfer) 방법과 영상 공간 접근 방법(Image Space Approach)이 제시되었다. PRT는 정적인 3 차원 물체가 하나의 Vertex 에 주는 영향을 미리 계산하여 각 Vertex 에 저장한 후 실행 시(Run-Time)에 저장된 정보를 이용하여 표면 아래의 산란을 표현한다[9]. 이 방법은 실시간으로 모양이 변형되는 물체에는 적합하지 않으며 많은 전처리 시간을 필요로 한다. 영상 공간 방법론(Image Space Approach)은 3 차원 기하 정보를 2 차원 영상으로 저장한 다음에 그 영상을 이용하여 하는 산란에 따라 방출되는 빛의 양을 계산한다[6][7]. Depth-Map 을 이용한 영상 기반 렌더링(Image Based Rendering)과 기본적으로 동일한 방법을 취한다. 최신 그래픽 하드웨어의 Pixel Shader 가 이미지 프로세싱에 유용하게 설계 되었기 때문에 영상 공간 방법론은 구현이 쉽고 전처리가 필요없다. 하지만 3 차원 정보를 2 차원 이미지로 변환하는 과정에서 많은 제약이 따른다.

3. 여러 장의 Depth-Map 을 이용한 영상 공간 접근 방법 (Image Space Approach)

Depth-Map 을 참조하면 이러한 접근방법을 반투명 재질에 이용하면 볼록한(Convex) 물체에 대해서 Viewer 의 방향에 대한 모든 기하 정보를 알 수 있다 [6][7](그림 3).



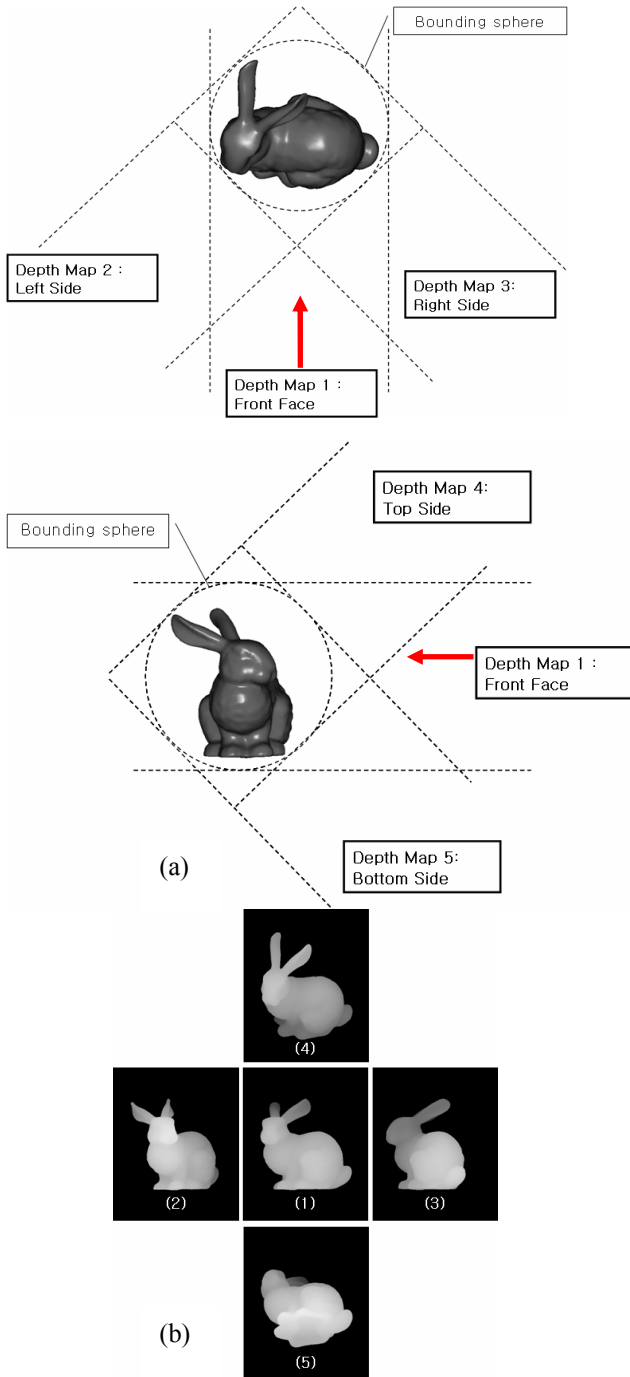
(그림 3) Depth Map 을 이용한 산란 표현

3.1 Depth-Map 의 실시간 생성

최신 그래픽하드웨어의 성능을 효율적으로 이용하여 Depth-Map 을 실시간으로 생성할 수 있다. 프레임 버퍼 대신 Texture 메모리에 2 차원 영상을 렌더링하고 두 번째 패스의 Pixel Shader 에서 이전 과정의 영상을

Texture 로서 사용하여 렌더링할 수 있다. 이미지의 RGBA 4 개의 채널에 각 픽셀 포인트에 해당하는 Normal 벡터의 성분  $x, y, z$  와 Depth  $z$  를 저장한다.

### 3.2 각도를 달리한 여러 장의 Depth-Map 생성



(그림 4) 여러 각도에서 얻은 Depth-Map.

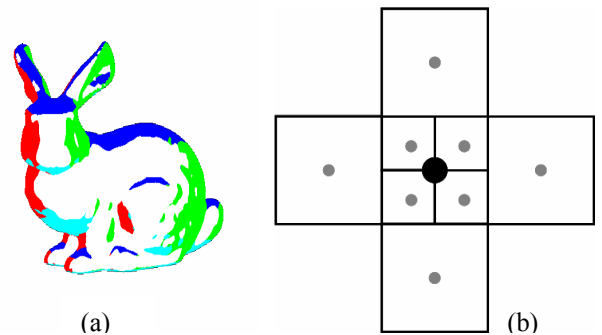
- (a) 방향에 따른 Depth-Map 생성.
- (b) 결과 이미지 (1) 정면, (2) 45도 각도의 좌측면, (3) 45도 각도의 우측면, (4) 45도 각도의 윗면, (5) 45도 각도의 아랫면.

기존 연구와 동일한 방법으로 Depth-Map 을 관측자와 같은 방향으로 하나만 생성하게 되면 관측자의 옆

쪽에서 입사되는 빛에 대한 문제가 발생한다[7](그림 3). 왜냐하면 물체의 실루엣 쪽으로 갈수록 주변의 깊이 정보가 부족하기 때문이다. 문제를 해결하기 위해 정면을 포함한 5 장의 Depth-Map 을 생성한다(그림 4). Depth-Map 생성에 5 개의 Pass 와 최종 이미지 생성에 1 개의 Pass, 총 6 개의 Pass 가 필요하다.

### 3.3 적응형 Depth-Map 참조

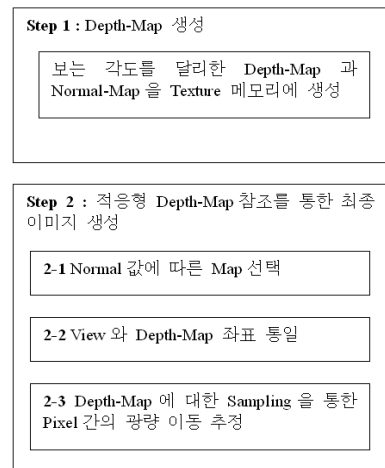
여러 장의 Depth-Map 중 주변 영역에 대한 정보를 가장 잘 포함하고 있는 것을 선택한다. 이때는 각 Pixel 의 Normal 값을 이용한다. Normal 값과 Depth-Map 의 방향이 가장 일치하는 것을 고른다. 그림 5 - (a)는 Normal 값에 따라서 구분되는 영역을 나타낸다. Normal 방향에 따라서 5 개의 영역으로 나뉘고 각각의 영역 내의 Pixel 의 값을 결정할 때에 그에 따라서 선택된 Depth-Map 을 참조한다.



(그림 5) (a) Normal 값에 따라 분할된 영역  
(b) 샘플링 패턴

### 3.3 샘플링

Depth Map 의 정보를 이용하여 주변 영역에서 들어오는 빛의 양을 샘플링 할 수 있다. 이 과정에서는 중요도 샘플링(Importance Sampling)을 사용한다. 관심 Pixel 에서 가까운 영역은 많은 수를 먼 영역은 적은 수의 샘플을 취한다[7]. 그림 5 - (b)는 샘플링 패턴을 나타낸다. 가운데 큰 점이 그리고자 하는 Pixel 이고 나머지 주변 점이 샘플링을 수행하는 지점이다. 하나



(그림 6) 프로그램 흐름

의 Pixel 을 그리기 위해 총 8 번의 샘플링을 수행하므로 총 8 번 Depth-Map 을 참조한다. Pixel 에 가까운 지점에서는 4 번의 샘플링이 이루어지고 나머지 네 지점에서는 각각 한번의 샘플링만 이루어진다.

그림 5 은 이상의 과정을 나타낸다.

#### 4. 결과

실험은 nVidia GeForce6600 GPU(4 개의 Vertex Pipeline, 6 개의 Pixel Pipeline)와 그래픽 메모리 126Mb 을 장착한 그래픽하드웨어와 3.0GHz Intel Pentium4 CPU 와 1Gb 메모리를 장착한 PC 를 사용하여 이루어졌다. Vertex 개수 38,620 개, Polygon 개수 68,328 의 모델을 사용하였고, 물리 계수로는 산란 계수 2.19/mm, 흡수 계수 0.0021, 굴절률 1.5, 모델의 전체 크기는 7.28cm × 7.24cm × 5.64cm 로 설정하였다. Viewport 512×512 의 화면에 DirectX9c 와 HLSH 로 구현하였다.

기존 방법에 따라 관측자 방향의 Depth-Map 한장을 사용하면 반투명 물체의 실루엣 경계부분에서 균일한 샘플링이 이루어 지지 않음으로 인해서 깨지는 등의 왜곡이 생긴다(그림 7 - (a)). 이것은 앞서 설명했듯이 샘플링을 할 때에 실루엣이나 Normal 값이 측면을 향하는 면에 대해서는 Depth-Map 에서 표면 주변의 적절한 샘플을 얻을 수 없기 때문에 생긴다. 제안된 방법으로 수행 후 실루엣 경계부분에서도 적절한 샘플링이 이루어 짐으로써 더 정확한 이미지를 얻을 수 있다(그림 7 - (b)).

렌더링 속도는 4.7 fps 로 기존 연구에 비해서 저하되었다. 여러 장의 Depth-Map 을 만들고 적응적으로

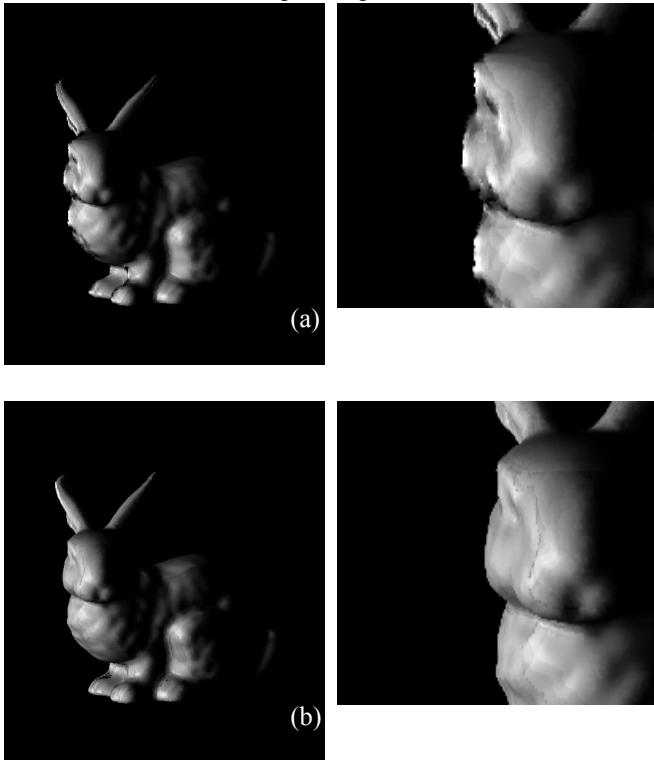


그림 7. 렌더링된 결과 (a) View 방향의 Depth-Map 한 장만을 이용한 결과 (b) 여러 장의 Depth-Map 을 사용한 결과

선택하는데 드는 시간 때문이다. 그러나 Pixel Shader 의 속도가 빠르게 향상되는 최신 그래픽스 하드웨어 발전의 영향을 감안하면 근래에 속도 향상을 기대할 수 있다.

#### 5. 결론 및 향후 연구 계획

본 연구는 반투명 재질의 실시간 렌더링을 위하여 그래픽스 하드웨어 구조에 적합한 방법을 제안하였다. 기존 연구에 한계점인 실루엣의 깨짐을 해결하고, 더욱 균일한 효과적인 샘플링을 가능하게 하였다. 그러나 여전히 불룩한 물체에만 물리적으로 적합하므로 오목한(Concave) 물체의 적용을 위한 개선이 필요하다.

#### 6. 감사의 글

본 연구는 광주과학기술원 실감방송 연구센터를 통한 정보통신부 대학 IT 연구센터(ITRC) 사업의 지원과 광주과학기술원의 실감 콘텐츠 연구센터(ICRC)를 통한 과학기술부 특정 연구개발 사업의 지원에 의한 것입니다.

#### 참고문헌

- [1] Jan J. Koenderink, and Andrea van Doorn, "Shading in the Case of Translucent Objects" In Proceedings of SPIE, 4299, p312-320, 2001.
- [2] Roland W. Fleming, Henrik Wann Jensen, and Heinrich H Bülthoff, "Perceiving translucent materials", Proceedings of the 1st Symposium on Applied perception in graphics and visualization, Los Angeles, California, pp.127-134, 2004.
- [3] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan, "A Practical Model for Subsurface Light Transport", In Proceedings of SIGGRAPH 2001, ACM, New York, pp511-518, 2001.
- [4] Henrik Wann Jensen, Juan Buhler, "A Rapid Hierarchical Rendering technique for Translucent Materials", In Proceedings of SIGGRAPH 2002, ACM, New York, 21, 3, pp576-581, 2002
- [5] Hendrik P.A. Lensch, Michael Goesele, Philippe Bekaert, Jan Kautz, Marcus A. Magnor, Jochen Lang, and Hans-Peter Seidel, "Interactive Rendering of Translucent Objects", In Proceedings of the 10th Pacific Conference on Computer Graphics and Applications 2002, pp214-224, October 2002
- [6] Carsten Dachsbacher, and Mare Stamminger. "Translucent Shadow Maps", In the Proceedings of the 14th Eurographics Symposium on Rendering 2003, pp197-201, 2003.
- [7] Tom Mertens, Jan Kautz, Philippe Bekaert, Frank Van Reeth, and Hans-Peter Seidel. "Efficient Rendering of Local Subsurface Scattering", In Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, p51-58, October08-10, 2003.
- [9] Xuejun Hao, and Amitabh Varshney, "Real-Time Rendering of Translucent Meshes", ACM Transactions on Graphics, Vol. 23, No.2, April 2004, pp.120-142, 2004