

3차원 그래픽스의 동영상에 대한 반사 효과의 고속처리

High Performance Reflection Effect Processing for Moving Pictures in 3 Dimensional Graphics

이승희 · 이건명*

Seung Hee Lee and Keon Myung Lee*

충북대학교 전자정보대학 컴퓨터공학전공

E-mail: kmlee@cbnu.ac.kr

요 약

컴퓨터 하드웨어 고성능화와 함께 실시간 컴퓨터그래픽스를 활용하는 응용 시스템이 많이 출현하고 있다. 이 논문에서는 3차원 컴퓨터그래픽스에서 모델 내에 애니메이션 동영상이 재생되고 있는 상황에서 반사효과를 효율적으로 구현하는 방법을 제안한다. 제안한 방법은 동영상에 재생영역이 특정 반사평면의 어느 위치에 어떤 형태로 반사되는지 기하학적으로 계산한 다음, 텍스처 매핑하는 방법으로 반사효과를 구현하는 것이다. 제안한 방법이 기존의 스텐실 버퍼를 이용한 반사평면 효과 기법에 비하여 동영상과 반사평면의 영상이 함께 나타나는 경우에 처리 속도에는 큰 영향이 없고, 최소 30%에서 최대 127% 정도의 처리율이 개선되는 것을 실험 데이터에서 확인하였다.

키워드 : 3차원 컴퓨터그래픽스, 반사효과, 애니메이션, 렌더링

Abstract

With the advance of high performance computing hardware, many applications have been emerging which exploit real-time computer graphics capabilities. This paper is concerned with an effective realization method for reflection effect for the situations in which moving pictures are played in 3D computer graphics modeling world. The method determines in a geometric way the locations of the projection plan into which the playing areas of moving pictures are mapped, and then realizes the reflection effect with texture mapping. Compared with the traditional stencil buffer-based reflection method, the processing time of the proposed method does not significantly deteriorate for the models with moving pictures and reflection surfaces, and its throughput was improved by 30% at minimum and 127% at maximum for the models used in the comparative studies.

Key Words : 3D Computer graphics, reflection effect, animation, rendering

1. 서 론

3차원 컴퓨터 그래픽스의 기술의 발달로 실사와 거의 같은 가상환경의 구현이 가능해지고 있다. 3차원 그래픽스에서는 가상환경을 구성하는 모든 물체를 3차원 공간상의 좌표로 표현하는 모델링을 한 다음, 3차원 물체를 2차원 화면에 대응하는 평면으로 투영(projection)을 하고 나서, 대응되는 화면상의 각 픽셀의 색상값을 결정하는 렌더링(rendering) 과정을 거쳐서, 이미지를 생성한다.[1] 이러한 3차원 컴퓨터 그래픽스 기술은 교육, 영화, 시뮬레이션, 게임, 시뮬레이션, 의료, 데이터의 시각화 등 다양한 분야에서 활용되고 있다.

실사와 같은 가상 환경을 컴퓨터 그래픽스를 통해 만들기 위해서 반드시 필요한 것 중의 하나는 실세계처럼 자연

스럽게 색상을 표면에 재현하는 것이다. 실세계에서 물체 표면의 색상은 물체의 빛에 대한 반사특성, 투과특성, 표면의 거칠기, 광원의 위치, 방향, 세기 등에 의해서 결정된다. 3차원 모델의 표면이 유리, 거울 등과 같이 반사특성을 갖는 경우, 이를 반영하여 렌더링하는 것이 필요하다. 이러한 현실감있는 렌더링 기법으로, 빛의 물리적인 특성을 반영하여 광원으로 부터의 빛이 어떤 반사, 굴절을 통해서 면에 어떤 세기로 도달하는지 추적하여 계산하는 광추적(ray tracing) 기법[1]이 있다. 이 방법은 양질의 결과를 얻을 수 있는 반면, 이론적으로 가능한 모든 광의 반사 및 굴절을 계산해야 하기 때문에 많은 시간과 비용을 필요로 한다. 광선추적법의 시간적 부담을 회피하는 렌더링 기법으로 환경 매핑(environment mapping)기법[1]이 사용되기도 한데, 이 방법은 미리 만들어둔 주위 환경 이미지를 물체 표면에 입혀서 투영시키는 것이다. 이 방법은 하늘 영역과 같이 원거리의 환경에 대해서는 효과적으로 이미지를 만들어내지만, 거울이나 대리석 표면과 같은 근거리 반사는 효과적으로 표현하지 못한다.[2] 한편, 근거리 반사 효과를 효과적으로 재현하는 기술로서 스텐실 버퍼(stencil buffer)를 사용하는 평면반사효과 기법[2,3]이 있다. 이 기법에서는 실제 화면에

접수일자 : 2008년 7월 21일

완료일자 : 2009년 5월 7일

* 교신 저자

이 논문은 2007년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음

어떤 영상의 일부분이 렌더링되는 것을 막는 스텐실 효과를 이용하는 것이 있다. 이 방법에서는 예를 들어 거울을 표현하고자 한다면 거울면을 기준으로 전체 모델을 대칭변환한 다음, 거울면에 해당하는 부분에 대해서만 다시 한번 렌더링하여 거울면에 대응하는 이미지를 만들어서 거울효과를 반영하지 않은 이미지와 합성하여 반사효과를 구현한다. 이 방법은 반사효과를 효과적으로 표현할 수 있지만, 동일한 렌더링을 두 번 수행해야 하기 때문에 비용이 큰 방법이다. 3차원 그래픽스에서는 많은 계산을 수행하게 되면 현실감을 향상시킬 수 있지만, 응용에 따라서는 또는 자원의 제약 등으로 인해서 많은 계산 비용을 부담할 수 없는 경우가 많다. 따라서 가능하면 많은 계산은 피하면서 현실감을 줄 수 있는 절충하는 접근 방법이 사용되고 있다.

이 논문에서는 컴퓨터그래픽스를 통해서 표현되는 공간상에 대형 스크린이 있고, 여기에서 동영상에 재생되고 있고 재생되는 동영상에 대리석 바닥 등과 같이 반사 특성이 있는 평면에 반사되고 있는 효과를 효과적으로 구현하는 방법을 대상으로 한다. 기존의 광선추적법이나 스텐실 버퍼를 사용하는 평면반사효과 기법을 적용하면 이러한 모델 중에 재생되는 동영상의 반사효과를 구현할 수 있지만, 계산 비용이 크기 때문에 하드웨어적인 제약이나 실시간 제약 때문에 적용하기 곤란할 수 있다. 따라서 이 논문에서는 이러한 반사효과를 적은 비용을 구현할 수 있는 방법을 제안한다.

2. 관련 기술

2.1 3차원 컴퓨터 그래픽스

3차원 컴퓨터 그래픽스는 실세계에서 사진을 찍어 2차원 이미지가 만들어지는 것과 같은 효과를 컴퓨터를 사용하여 계산적으로 구현하는 기술로서, 사진을 찍을 대상이 실제 존재하는 것이 아니라 컴퓨터 내부에 데이터로 존재하고, 사진을 찍는 것이 광학적인 현상에 의하는 것이 아니라 계산을 통해 수행된다.

3차원 그래픽스에서는 이미지를 생성할 대상이 되는 3차원 물체로 구성된 대상 환경을 만드는 것이 우선적으로 필요한데 이러한 작업을 모델링(modeling)이라고 한다. 모델을 구성하는 가장 일반적인 방법은 3차원 물체의 표면을 다각형(polygon)으로 나타내는 것으로, 다각형은 이를 정의하는 꼭지점(vertex)의 3차원 좌표값들로 명세하여 모델의 형태를 정의한다. 3차원 물체를 구성하는 다각형은 가상의 3차원 공간좌표계를 기준으로 꼭지점의 좌표값들에 의해서 기술되고, 이들 다각형은 꼭지점에 색상정보가 부여되거나, 또는 반사특성 정보가 부여되며, 외부면을 나타내기 위한 법선 벡터(normal vector)가 부여된다. 한편, 물체가 반사하는 색상은 물체를 비추는 빛에 의해 결정되기 때문에, 추가적으로 광원(light source)의 세기, 색상, 방향, 위치 등의 정보를 환경에 설정하게 되면 3차원 그래픽스를 위한 모델이 완성된다.

3차원 모델을 2차원 이미지로 나타내기 위해서는 3차원 모델의 각 면이 2차원 이미지 면에 어떻게 사상(mapping)되는지 결정하는 투영(projection) 하는 단계를 거쳐야 한다. 투영은 카메라, 즉 관찰자의 위치, 방향, 카메라 면의 방향에 의해서 결정되며, 대표적인 투영 방법으로는 원근감을 느낄 수 있도록 거리에 따라 투영되는 크기에 달라지도록 하는 원근 투영법(perspective projection)과, 거리에 무관하게 투영되는 평행 투영법(parallel projection) 메라,다. 원근

투영법에서는 카메라의 위치가 하는 공간상의 임의의 장소에 위치하도록 설정되(ion)평행 투영인 경우에는 카메라가 무한대 거리에 하도록것으로 가정되어, 3차원 물체 상의 어떤 점이 2차원 이미지가 투영될 때 위치는, 해당 점과 카메라 위치를 연장한 직선과, 투영면이 만나는 점이 된다.

투영된 위치의 색상값을 결정하는 것을 렌더링이라고 하는데, 렌더링을 하기 전에 실제 카메라에 비춰지는 면을 선택하기 위한 은면(hidden surface) 제거 등의 과정을 거치게 된다. 물체 표현에 대한 색상은 표면에 지정된 색상, 광원의 색상 및 방향과 강도, 표면의 거칠기 및 반사특성 등에 의해서 결정된다. 이러한 물리적인 특성을 얼마나 충분히 반영하여 색상 값이 결정되는가에 따라 현실감이 좌우된다. 또한 현실감을 높이기 위해서 다양한 무늬를 모델의 표면에 벽지 바르듯이 대응시키는 텍스처 매핑(texture mapping) 기법도 많이 활용되고 있다. 3차원 모델에 투명하거나 반투명한 평면이 있거나 거울이나 미끄러운 바닥면이 있을 경우에는 투과나 반사를 반영하여 렌더링하는 것이 필요하다. 빛의 투과나 반사를 고려하여 색상값을 계산하는 데는 많은 비용이 요구된다.

2.2 반사효과 구현 방법

이 논문에서 대상으로 하는 반사효과를 반영하여 렌더링할 수 있는 대표적인 방법으로는 광추적 기법[1], 환경 매핑 기법[1], 스텐실 버퍼를 사용하는 평면반사효과 기법[2] 등이 있다.

광추적 기법은 관찰자가 물체를 볼 수 있는 것은 물체에서 반사된 빛을 감지하는 것이라는 사실에 입각한 것으로, 빛 입자 하나가 물체 표면에 반사되어 관찰자의 눈에 들어오는 경로를 역으로 추적하여 해당 경로에 의해서 빛이 어느 정도의 세기로 들어오는지 계산하는 방법이다. 광추적 기법은 다음과 같은 과정을 통해서 렌더링을 한다.

- 단계 1. 매 프레임(frame) 이미지를 생성할 때 마다 시점(viewpoint)에서 이미지의 각 픽셀 방향으로 광선을 쏜다.
- 단계 2. 이 광선이 광원에 도달하거나 정해진 횟수 만큼 반사 또는 투과할 때까지 광자의 이동 경로를 추적한다.
- 단계 3. 광자의 이동경로를 따라서 해당 픽셀로 들어오는 빛의 세기를 계산하여 색상을 결정한다.

이 방법은 실제 환경에서 일어나는 물리적인 빛의 반사 모델을 그대로 적용한 것이기 때문에 3차원 가상 환경에서도 실제 환경과 같이 렌더링된 이미지를 얻을 수 있다. 하지만 각 픽셀별로 경로를 추적해야 하기 때문에 많은 계산 비용이 요구되는 방법이다.

환경 매핑 기법은 광선 추적을 사용하지 않고, 3차원 공간 중심에서 주변 환경을 반사하는 표면을 갖는 모델을 렌더링하기 위해 제안된 방법이다. 이 방법에서는 렌더링할 물체가 전반사(全反射) 표면을 가지고 있다고 전제하고, 이 물체의 상, 하, 좌, 우, 전, 후 면 방향에 비추어질 장면을 계산하여 정육면체 형태의 장면을 구하게 되는데, 이러한 정육면체의 표면에 대응하는 전체 이미지를 환경 맵(environment map)이라고 한다. 다음은 환경 매핑 기법의 과정을 보인 것이다.

- 단계 1. 반사 특성을 갖는 면을 갖는 물체를 3차원 공간의 중앙으로 간주하고 주변 환경에 대한 영상을 상, 하, 좌, 우, 전, 후에 해당하는 여섯 방향에서 얻어낸다. 이를 위해 해당 물체가 위치할 자리에 대신 카메라를 놓고 이들 여섯 방향에 대한 렌더링을 하

여 구한다.

단계 2. 이들 여섯 방향에 대한 장면 이미지로 하나의 텍스처를 구성하는데, 이것이 환경 맵이다.

단계 3. 만들어진 환경 맵을 반사특성을 갖는 모델의 표면에 텍스처 매핑한다.

단계 4. 텍스처 매핑된 객체를 3차원 모델 공간에 두고, 렌더링 연산을 수행한다.

환경매핑 방법은 광추적 기법에 비해서는 계산량이 상대적으로 적긴 하지만, 각 프레임 이미지를 생성할 때마다 환경 맵을 구성하기 위해 여섯 번의 렌더링을 하고 최종적으로 한번의 렌더링을 해야 하기 때문에 총 일곱 번의 렌더링을 해야 한다. 반사 면을 갖는 물체가 여러개인 경우에는 이러한 물체마다 환경 맵을 별도로 생성해야 하기 때문에 계산 비용이 크게 증가하는 문제가 있다.

평면 반사 효과 기법은 반사효과를 구현하기 위해서 스텐실 버퍼라는 특수한 버퍼를 사용한다. 스텐실 버퍼는 0 또는 1값으로 채워질 수 있고 크기는 후면 버퍼의 크기와 같고, 스텐실 버퍼의 (i, j)번째 픽셀은 후면 버퍼의 (i, j)번째 픽셀에 대응한다. 이 방법은 먼저 전체 모델을 후면 버퍼에 렌더링하고, 스텐실 버퍼에 반사평면 영역을 표시한다. 반사 평면을 중심으로 대칭된 전체 모델을 스텐실 버퍼에 있는 값을 비교하면서 후면 버퍼에 그려 넣는다. 이 과정에서는 스텐실 버퍼에 표시된 반사 평면 영역에 대응하는 후면버퍼 영역에만 영상을 그려넣는다. 스텐실 버퍼를 이용하는 평면반사 효과 기법은 다음과 같은 과정을 통해서 반사효과를 구현한다.

단계 1. 전체 모델을 보통 때와 마찬가지로 렌더링한다.

단계 2. 스텐실 버퍼의 값은 모두 0으로 설정하고, 반사평면을 스텐실 버퍼에 렌더링하여 반사평면에 해당하는 스텐실 버퍼의 픽셀은 1로 설정한다.

단계 3. 반사 평면을 기준으로 전체 모델을 대칭시키고 후면 버퍼에 렌더링을 하는데, 이 때 스텐실 버퍼의 대응되는 픽셀의 값이 1인 경우에만 반사된 모델이 후면 버퍼에 반영되도록 한다.

평면 반사 효과 기법은 반사 효과를 반영하기 위해서 두 번의 전체 모델에 대한 렌더링과 한번의 대칭변환을 해야 한다.

이 논문에서는 3차원 모델의 표면에 동영상이 상연되고 있고 이것이 반사면에 비춰지는 상황에서의 효과적인 렌더링에 대해서 효과적인 방법을 모색한다. 동영상은 최소 초당 15 프레임정도로 재생이 되어야 자연스러워 보이기 때문에 하나의 이미지를 생성하는데 1/15초 이내에 시간만이 허용된다. 이러한 제약하에서 모델이 복잡해 질 경우 광추적 기법은 비용이 너무 많이 들어서 저렴한 하드웨어 상에서는 아직은 적용할 수 없다. 한편 환경 매핑의 경우에도 1/15초 이내에 최소 7번의 렌더링을 하는 것은 무리이고, 스텐실 버퍼를 사용하는 평면 반사 효과 기법은 1/30초 이내에 하나의 프레임 이미지만 만들 수 있으면 되기 때문에 적용가능할 수도 있지만, 일반 개인용 컴퓨터에서 게임 등의 응용에서 이러한 효과를 활용하기에는 적용하기 어려울 수도 있다. 이 논문에서는 스텐실 버퍼를 사용하는 평면 반사 효과 기법보다 효과적으로 거울, 스크린 등의 평면에 상연되는 동영상이 대리석 표면과 같이 매끈한 반사평면에 반사되는 효과를 구현하는 방법을 제안한다.

3. 제안한 반사효과 구현 기술

복잡하게 구성된 가상 공간에서 동영상이 상연되고 있는

반사평면이 있는 상황을 실시간 컴퓨터 그래픽스로 구현하는 것은 많은 계산 비용을 요구한다. 특히 반사효과 등은 현실감을 높여주는 부수적인 효과로서 모든 반사특성을 갖는 표면에 대해서 반사효과를 구현하지 않더라도 현저하게 눈에 띄는 일부분에만 적용하더라도 효과가 있다. 앞 절에서 기술한 광추적 기법, 환경 매핑 기법, 스텐실 버퍼를 이용한 평면 반사효과 기법은 반사효과를 실현할 수 있지만, 계산 시간적인 비용이 문제가 된다. 이중 스텐실 버퍼를 이용한 기법이 효과적이기는 하지만 두 번의 렌더링을 하는 것은 애니메이션으로 재현되어야 하는 3차원 그래픽스에서는 부담이 크기 때문에 적용하는데 제약이 있다. 이 논문에서는 모든 반사 특성이 있는 표면에 대한 반사효과를 구현하는 것보다는 관찰자에게 가장 크게 영향을 줄 수 있는 반사평면에 대해서 상연되는 동영상의 반사효과를 구현하는 방법을 제안한다.

3.1 동영상 반사효과 구현 절차

제안한 방법의 기본 개념은 영상의 재현영역의 네 꼭지점이 반사특성이 있는 각 물체의 표면의 어떤 위치에 대응하는지 계산하여, 동영상을 구성하는 각 정지영상을 상기 계산된 대응점을 꼭지점으로 하는 사각형에 직접 텍스처 매핑(texture mapping)하도록 하여 렌더링하도록 것이다. 동영상의 프레임속도(frame rate)에 따라서 주기적으로 동영상을 구성하는 정지영상을 직접 반사특성이 있는 평면에 텍스처 매핑을 하게 되면 동영상의 반사효과가 적은 처리비용으로 실현할 수 있다.

(그림 1)은 제안한 방법이 상연중인 동영상에 특정 반사평면에 반사효과가 나도록 하는 처리과정을 보인 것이다. 그림에 보는 바와 같이 3차원 컴퓨터그래픽스 모델에서 동영상의 재현과 반사효과 표현을 위해서 우선 동영상의 표시영역 꼭지점들에 대해서 이들이 각 반사평면의 어떤 위치에 대응되는지 계산한다. 동영상을 구성하는 정지영상들을 프레임속도에 따라 동영상 표시영역에 텍스처 매핑하면서, 반사평면별로 동영상의 표시영역 꼭지점에 대응하는 점들을 사각형의 꼭지점으로 하는 영역에 프레임속도에 동기화시켜 해당 정지영상을 텍스처 매핑을 실현한다. 이러한 처리결과를 3차원 컴퓨터그래픽스의 일반 렌더링 모듈에 적용하면, 동영상 재현과 함께 반사평면상에서 동영상의 반사효과가 실현된다.

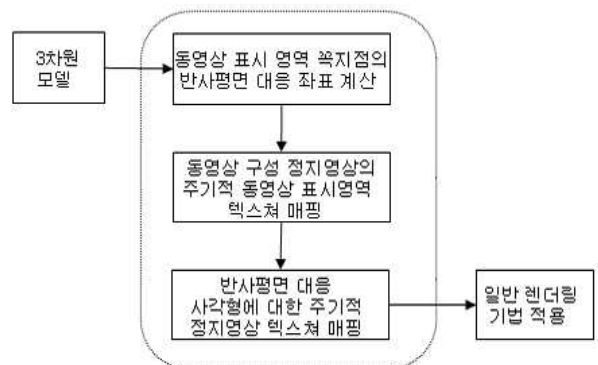


그림 1. 제안한 동영상의 특정 반사평면에 대한 반사효과 구현 방법의 처리 과정

Fig. 1 The proposed processing procedure to implement reflection effect of moving pictures on a reflexive surface

3.2 동영상 표시 영역의 반사평면 상의 위치 결정

반사평면에 반사되어 나타나는 이미지는 관찰자의 위치 및 관찰방향, 반사평면 및 물체의 방향 등에 의해서 결정된다. 제안한 방법을 적용하기 위해서는 동영상이 사각형 영역에 상연되고 있는 것을 전제하고, 이 동영상 영역이 반사평면 상의 어디에 나타나는 지 계산해야 한다. (그림 2)는 물체의 반사면에 대한 반사특성을 보인 것으로, 관찰자의 시점 O 에 대해서 물체 상의 어떤 점 P 가 법선벡터 \vec{n} 를 갖는 반사평면에 맺히는 위치 R 은, \vec{n} 와 평행이면서 점 P , O 를 포함하는 평면상에서 입사각(入射角)과 반사각(反射角)이 동일해지는 위치 R 이 된다.

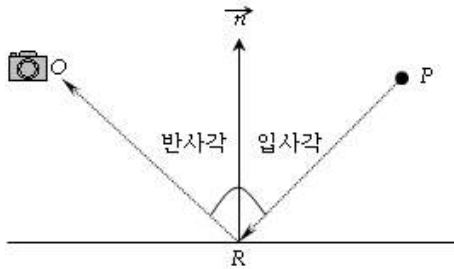


그림 2. 반사면에 대한 반사특성

Fig. 2 Reflection characteristics on a reflection surface

이와 같은 반사면의 특성을 이용하여 동영상이 상연되고 있는 사각형 표시영역의 꼭지점이 현재 관찰자의 시점을 기준으로 지정된 반사평면의 어떤 위치에 대응되는지 계산한다. (그림 3)은 동영상 표시영역의 꼭지점들에 대응하는 반사평면에서의 좌표들을 도식한 것이다. 편의상 다음과 같은 기호를 사용하여 대응 위치의 계산방법을 기술한다.

- O : 투영면(projection plane)의 상향벡터 방향에 있는 좌표값 $(0,1,0)$ 에 해당되는 점
- U : 투영면의 상향벡터 방향에 있는 좌표값 $(0,1,0)$ 에 해당되는 점
- \vec{n} : 반사평면의 법선벡터
- P_i ($i = 1,2,3,4$) : 3차원 컴퓨터그래픽스 모델에서 동영상이 표시되는 영역의 꼭지점 좌표값
- B_i ($i = 1,2,3,4$) : 반사평면상의 임의의 좌표값
- R_i ($i = 1,2,3,4$) : P_i 는 동영상 표시영역의 꼭지점 P_i 가 반사평면에 대응되는 좌표값

관찰자 시점 O 를 원점 $(0,0,0)$ 으로 하고, 관찰자 시점의 상향벡터(up vector) 방향의 좌표값 $(0,1,0)$ 을 U 로 하여, 이들 두 점과 동영상 표현 영역의 한 꼭지점 P_i 를 포함하는 평면과, 점 B_1, B_2, B_3, B_4 등을 포함하는 반사특성이 있는 평면이 만나는 곳들 중에서, P_i 로부터 입사되어 O 로 반사하는 것 중에 입사각과 반사각의 크기가 동일한 점이, P_i 에 대응하는 점 R_i 가 된다. 아래 식은 동영상 표시영역의 꼭지점 P_i 에 대응하는 R_i 의 좌표를 결정하는 식을 나타낸 것이다. 입사각과 반사각이 같아야 하는 조건에서 다음 식이 만들어진다.

$$\frac{\vec{OR}_i \cdot \vec{n}}{\vec{OR}_i} = \frac{\vec{P}_i R_i \cdot \vec{n}}{\vec{P}_i R_i} \quad (1)$$

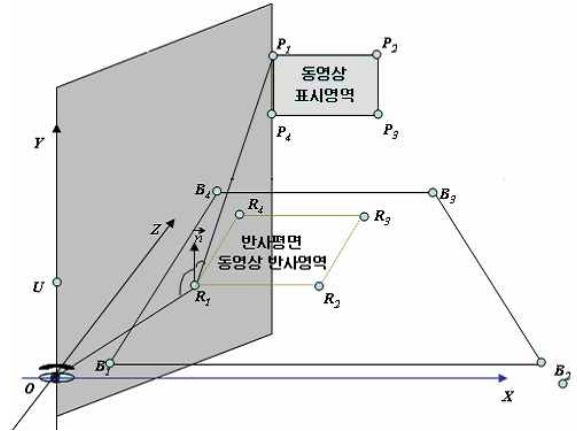


그림 3. 동영상 표시영역과 반사영역의 관계

Fig. 3 Correspondence of moving picture presentation area and reflection surface

R_i 가 점 O, U 를 포함하는 평면상에 존재해야 하고, R_i 가 점 B_1, B_2, B_4 를 포함하는 평면상에 위치해야 하기 때문에 다음 두 조건을 만족해야 한다.

$$(\vec{OR}_i \times \vec{OU}) \cdot \vec{OR}_i = 0 \quad (2)$$

$$(\vec{B}_1 B_2 \times \vec{B}_1 B_4) \cdot \vec{B}_1 R_i = 0 \quad (3)$$

동영상 표시영역의 각 꼭지점 P_i ($i = 1,2,3,4$)에 대해서 식(1)-(3)의 조건을 만족하는 좌표 R_i ($i = 1,2,3,4$)를 계산하면, 이들 좌표가 반사평면상에 동영상이 반사되는 영역의 꼭지점이 된다.

3.3 동영상의 반사평면 효과 렌더링

동영상 표시영역의 꼭지점에 대한 반사평면상의 대응 위치가 계산되면, 이들 위치를 꼭지점으로 하는 사각형 모양으로 현시점에서 재생되는 동영상 프레임을 변형하여 해당 반사영역에 텍스처 매핑하면 된다. 이때 반사평면의 특성을 반영하여 반사평면의 현재 레더링 결과와 알파 블렌딩(alpha blending)과 같은 기법을 이용하여 반사효과를 구현한다. 다음은 제안한 동영상 반사효과 구현 방법을 의사코드로 표현한 것이다.

procedure Animation_Reflection_in_3D

begin

반사평면별로 동영상의 꼭지점 좌표의 반사위치 좌표를 계산한다.

do

동영상을 구성하는 현재시점의 정지영상 프레임을 가져온다.

해당 정지영상 프레임을 동영상 표시영역에 텍스처 매핑한다.

해당 정지영상 프레임을 계산된 좌표를 꼭지점으로 하는 사각형 모양으로 변형한다.

변형된 프레임을 계산된 반사영역 위치에 반사율을 반영하여 텍스처 매핑한다.

시스템에 지정된 렌더링 방법에 따라 3차원 모델을 렌더링한다.

while (재생할 동영상 프레임이 남아 있음)
end.

4. 실험 및 평가

제안한 방법의 성능 및 효과를 평가하기 위해서 현재 실시간 반사효과 처리에 사용되고 있는 스텐실 버퍼를 이용하는 평면반사효과 기법과 비교실험을 하였다.

4.1 실험 환경

실험에서는 보급형 개인용 컴퓨터에서 많이 사용되고 있는 SiS사의 SiS600 그래픽 카드를 사용하는 Microsoft Windows XP Professional 운영체제가 설치된 환경에서, 3차원 컴퓨터 그래픽스 API인 Direct3D[6]를 Microsoft Visual C++에서 사용하여 작성한 응용프로그램을 구현하여 평가를 하였다. 실험에서 사용된 3차원 모델의 정점(vertex) 수는 10,494개이고, 모니터의 화면 재생 빈도는 60Hz로 설정하였다. 3차원 모델은 1280x1024, 1024x768, 800x600 해상도에서 각각 이루어졌고, 각 해상도에 대해서 픽셀 깊이를 32bit, 16bit로 설정하여 실험을 하였다. 제안한 방법과 평면반사효과 기법의 성능은 3차원 모델을 화면에 렌더링하는 프레임 수를 측정하여 비교하였다. (그림 4)는 실험에서 사용된 3차원 모델의 네 개 시점에서 렌더링한 스냅샷을 보인 것이다.

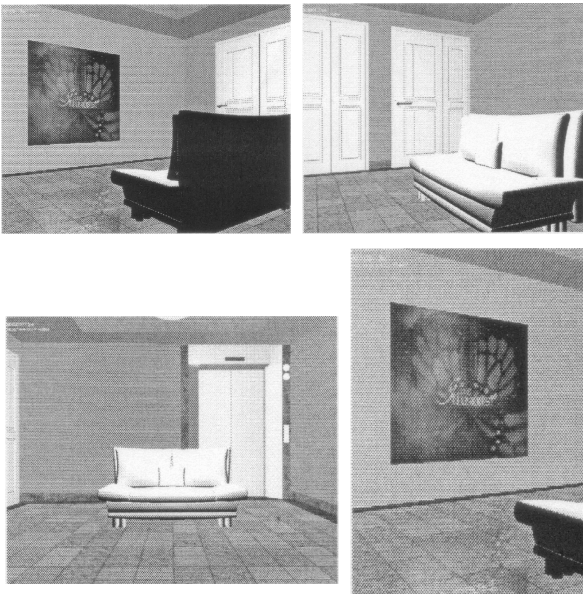


그림 4. 실험에서 사용된 3차원 모델
Fig. 4 A3-D model used in the experiments

4.2 실험 결과 및 비교

3차원 컴퓨터그래픽스에서는 하나의 영상이 렌더링될 때, 깜박거리는 현상을 막기 위해 후면버퍼를 사용하게 되는데 후면 버퍼의 크기는 화면의 해상도, 화면 품질을 결정하는 픽셀깊이에 의해서 결정된다. 다음의 표 1은 해상도와 픽셀 깊이별 요구되는 후면버퍼의 크기를 나타낸다. 후면버퍼의 크기는 (해상도)×(픽셀깊이)가 된다.

표 1. 해상도 및 픽셀깊이에 따른 후면버퍼 크기

Table 1. Back buffer size according to resolution and pixel depth

픽셀깊이	해상도	후면버퍼크기
16bits	800×600	960KB
	1024×768	1.6MB
	1280×1024	2.6MB
32bits	800×600	1.9MB
	1024×768	3.1MB
	1280×1024	5.2MB

스텐실 버퍼를 이용하는 반사평면 모델에서는 후면버퍼와 함께 동일한 크기의 스텐실 버퍼를 함께 사용해야 한다. 스텐실 버퍼는 1비트만 있어도 되지만, 실험에서 그래픽 API로 사용한 Direct3D[6]에서는 D3DFMT_D24S8이라는 8비트를 픽셀별로 사용하는 버퍼 포맷을 이용하기 때문에 해상도별로 표 2와 같은 크기의 스텐실 버퍼 공간을 추가적으로 사용하지만, 제안하는 방법에서는 이러한 공간을 사용하지 않는다.

표 2. 해상도별 스텐실 버퍼 크기

Table 2. Stencil buffer size to resolution

해상도	버퍼 크기
800×600	480KB
1024×768	786KB
1280×1024	1.3MB

앞 절에서 기술한 하드웨어 실험환경에서 제안된 기법과 스텐실 버퍼를 사용하는 평면반사효과 기법의 렌더링 속도를 초당 렌더링 가능한 프레임수(fps)로 측정하여 비교하였다. 실험에서는 (그림 5)에서 보이는 것과 같이 (a)의 벽에 동영상상이 재생되면서 아래 반사평면에 비춰지는 경우와 (b)의 바닥에 동영상만 비춰지는 경우에 대해서 두 기법을 비교하였다.

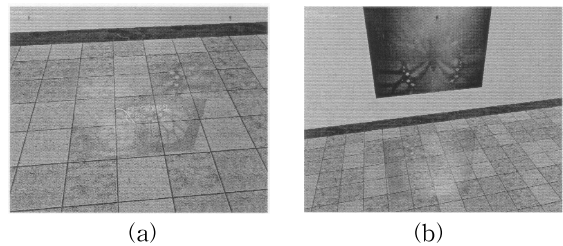


그림 5. 실험에서 사용된 렌더링 화면. (a) 바닥에 반사되는 동영상만 보이는 경우 (b) 동영상상이 벽에 재생되면서 바닥에 반사되는 경우

Fig. 5 Rendered scenes used in the experiments. (a) case with only reflected scene on the floor (b) case with scene on the wall and its reflection on the floor

표 3. 비교기법의 렌더링 속도 (단위: fps)
Table 3. Rendering rate of the compared methods

	픽셀깊이 렌더링화면 해상도	32bits		16bits	
		반사평면	동영상과 반사평면	반사평면	동영상과 반사평면
반사 평면 효과	800×600	46.27	30.14	30.15	30.13
	1024×768	28.97	19.68	27.07	19.99
	1280×1024	15.03	12.01	16.75	15.00
제안 기법	800×600	60.26	60.28	60.28	59.29
	1024×768	30.00	29.99	59.99	57.96
	1280×1024	26.57	23.62	29.85	30.05

표 3에서 보는 바와 같이 지금까지 대표적으로 반사효과 구현에 사용되는 스텐실 버퍼를 이용하는 평면반사효과 기법은 반사평면이 있는 경우 기본적으로 반사평면 영역에 대해서 렌더링이 한번 더 일어나야 하기 때문에 반사평면이 있는 경우 프레임 처리 속도가 늦어지고, 해상도가 커질수록 픽셀깊이가 커질수록 프레임 처리 속도가 늦어진다. 반면에, 제안한 방법은 동영상과 반사평면의 영상이 함께 나타나는 경우에도 처리 속도에는 큰 영향이 없고, 최소 30%에서 최대 127% 정도의 처리율이 개선되는 것을 확인할 수 있었다.

5. 결 론

3차원 컴퓨터 그래픽스로 재현된 모델의 품질을 결정하는 중요한 요인은 렌더링된 영상으로부터 현실 세계에서 느끼는 것과 같은 현실감을 느끼게 하는 것과 애니메이션인 경우에는 끊김 현상이 없도록 하면서 최소 프레임 재생률을 보장하도록 하는 것이다. 이 논문에서는 3차원 컴퓨터그래픽스에서 모델 내에 애니메이션 동영상이 재생되고 있는 상황에서 반사효과를 효율적으로 구현하는 방법을 제안하였다. 제안한 방법은 동영상이 재생영역이 특정 반사평면의 어느 위치에 어떤 형태로 반사되는지 기하학적으로 계산한 다음, 텍스처 매핑하는 방법으로 반사효과를 구현하는 것이다. 제안한 방법을 구현하여 실제 모델에 대한 실험을 통해서 평가할 결과 기존의 방법인 스텐실 버퍼를 이용하는 평면반사효과 기법에 비해서 동영상과 반사평면의 영상이 함께 나타나는 경우에 처리 속도에는 큰 영향이 없고, 최소 30%에서 최대 127% 정도의 처리율이 개선되는 것을 실험 데이터에서 확인하였다.

평면반사효과 기법은 3차원 그래픽스의 모델에 대한 투영과 렌더링을 통해서 반사평면에 비추어질 반사이미지를 정확히 반영할 수 있어서 반사영상의 품질이 제안한 방법에 비하여 더 현실감이 있을 수는 있지만, 3차원 모델이 복잡해지고 해상도가 커지면, 표 6의 결과에서 볼 수 있듯이 애니메이션을 위해 최소 필요한 초당 15 프레임을 처리하기도 어려워지기 때문에, 동영상이 재생되는 3차원 환경에 적용하기에는 하드웨어적인 부담이 크다. 따라서 제안한 방법은 3차원 모델 내에서 동영상이 재생되고 반사평면이 있는 애니메이션을 컴퓨터그래픽스를 통해서 구현하는 경우 효과적으로 사용될 수 있다.

참 고 문 헌

- [1] H. Bungartz, M. Griebel, and C. Zenger, "Introduction to Computer Graphics", Charles River Media, 2004.
- [2] J. Kilgard, "Improving Shadows and Reflections via the Stencil Buffer", <http://developer.nvidia.com/attach/6641>, 1999.
- [3] Mark J. Kilgard, "Creating Refelctions and Shadows with Stencil Buffers", *GDC 99*, 1999.
- [4] R. Bastos, W. Stuerzlinger, "Forward Mapped Planar Mirror Reflections", *University of North Carolina at Chapel Hill, Computer Science Technical Report TR98-206*, 1998.
- [5] W. F. Engel, A. Geva, "Beginning Direct3D Game Programming"(2nd Ed.), Prima Tech, 2000.

저 자 소 개

이승희(Seung Hee Lee)

15권 3호 참조

이건명(Keon Myung Lee)

15권 3호 참조