

# 투영 텍스처를 이용한 만화적인 반사 효과의 렌더링

황 순재<sup>1</sup> 최 정주<sup>1</sup> 이 승용<sup>2</sup>

<sup>1</sup>아주대학교 미디어학과, <sup>2</sup>포항공과대학교 컴퓨터공학과

<sup>1</sup>(lilyloh, jungju)@ajou.ac.kr, <sup>2</sup>leesy@postech.ac.kr

## Rendering cartoon-style specular reflection using projective textures

Soon-Jae Hwang<sup>1</sup> Jung-Ju Choi<sup>1</sup> Seungyong Lee<sup>2</sup>

<sup>1</sup>Department of Digital Media, Ajou University

<sup>2</sup>Department of Computer Science and Engineering, POSTECH

### 요 약

물체의 반사(specular reflection)는 물체의 재질을 이해하는 중요한 실마리임과 더불어, 반사 영역의 모양은 국소적으로 물체의 기하학적 정보를 표시하는 중요한 부분이다. 기존의 비사실적 렌더링 분야에서 다루어져 왔던 만화적 표현 관련 연구들은 극도로 단순화된 셰이딩 모델을 사용하여 모델의 상세도를 줄이는 방법을 사용하기 때문에 반사 효과와 같은 상세함을 표현하기에는 부족하다. 본 논문에서는 물체의 반사 영역을 만화적으로 표현하는 새로운 비사실적 렌더링 기법을 제안한다. 물체의 국소 반사 모델(local reflection model)을 사용하여 물체의 모든 정점에 대한 빛의 세기를 계산한 후, 반사 영역을 추출한다. 해당 반사 영역에 만화적으로 패턴화된 텍스처를 투영하여 반사 영역을 만화적인 느낌으로 표현한다. 본 논문에서 제시하는 결과는 국소 반사 모델의 계산을 사용하고 투영 텍스처(projective texture)를 이용하는 모델 공간(object space)의 계산으로, 오늘날 보편화되고 있는 그래픽 하드웨어의 가속 성능을 최대한으로 사용할 수 있는 장점이 있다. 따라서 본 논문의 연구결과는 실시간성이 매우 중요한 게임 등의 분야에서 만화적인 렌더링 기법을 적용하는데 사용할 수 있다.

### 1. 서론

컴퓨터 그래픽스 연구의 지속적 발전에 따라서 사진품질의 사실적 렌더링(photorealistic rendering) 결과물이 실사와 분간하기 어려울 정도로 좋아지고 있으며, 실제로 많은 영화와 3차원 애니메이션, 3차원 게임 등에서 실사와 유사한 품질의 화면을 보여주고 있다. 특히 3차원 게임과 같이 실시간성이 매우 중요한 응용분야에 적용되는 사실적 렌더링의 결과들은 오늘날 그래픽 하드웨어 가속 장치의 눈부신 발달에 도움을 받아 급속히 화면 품질이 향상되고 있다. 이는 하드웨어의 렌더링 파이프라인이 사실적 렌더링을 수행하기에 적합하게 설계되어 있다는 반증이기도 하다. 실제로 오늘날 렌더링 가속 하드웨어는 국소 반사 모델(local reflection model)에 근거한 계산을 빠르게 수행해 주는 역할이 주를 이루고 있다.

비사실적 렌더링(non-photorealistic rendering) 분야의 연구는 대개 컴퓨터를 사용하여 회화적 혹은 만화적인 느낌을 얻기 위하여 숙련된 아티스트의 작품에 사용된 기법을 최대한 흉내 내는 것을 목표로 하는 경우가 많

다. 따라서 회화에 사용하는 붓과 펜의 사용 기법 혹은 사용 효과를 실사에 적용하는 이미지 공간에서의 처리에 초점이 맞추어져 왔다. 이러한 결과들은 몇몇 애니메이션과 게임에 적용된 사례가 있지만 실시간성이 중요한 응용분야에서 하드웨어 가속장치의 도움을 받기 힘들고, 따라서 게임과 같은 응용분야에 적용하기 힘든 것이 현실이다. 비사실적 렌더링 분야에서 만화적 렌더링의 경우는 일부 하드웨어의 가속 성능을 이용하기 위하여 국소 반사 모델에 근거한 모델 공간의 처리를 도입하여 좋은 결과를 얻는 경우도 있다. 하지만 기존 만화적 렌더링 기법은 대개 국소 반사 모델에서 광원에 의한 산란(diffuse reflection) 효과를 위주로 묘사하고 있어, 광원에 따른 물체의 보다 풍부한 질감을 표현하는 데는 한계가 있다.

본 논문에서는 물체의 반사(specular reflection) 효과를 하드웨어 가속 장치를 이용하여 만화적으로 표현하는 새로운 방법을 제시한다. 기존 만화적 렌더링의 결과에 단순히 국소 반사 모델의 근거한 반사광 효과를 추가하는 경우에는 실사와 만화가 단순히 합성되어 일부 어색한 결과를 얻게 되지만, 본 논문에서는 반사 효과 자체도 만화적으로 표현하는 방법을 제시한다. 본 논문에서

제시하는 방법은 만화나 일러스트 작품 등에 등장하는 반사 효과 표현이 대개 해당 반사 영역에 일정한 패턴이 등장하는 경향이 있다는 관찰에 근거하고 있다(그림 1 참조). 따라서 이러한 패턴을 반사 영역에 절절하게 표시할 수 있다면 기존의 연구결과들과 함께 보다 풍부한 표현을 얻을 수 있을 것으로 기대할 수 있다.

본 논문의 개략적인 구성은 다음과 같다. 2장에서는 비사실적 렌더링의 만화적 렌더링과 관련된 기존의 연구 결과와, 본 논문에서 사용하는 국소 반사 모델에 대하여 간략히 살펴보고, 3장에서는 반사 효과의 만화적 렌더링을 위한 자세한 방법을 설명한다. 4장에서는 다양한 실험 결과를 제시하고, 마지막으로 5장에서 결론 및 향후 연구과제에 대하여 언급한다.

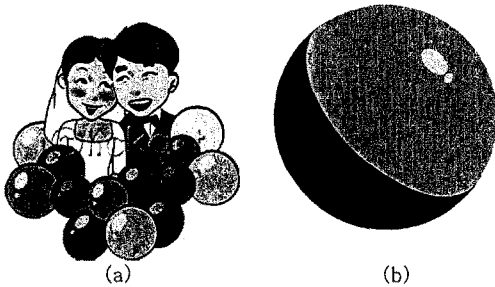


그림 1 : (a) 2D 일러스트에 등장하는 반사 효과에 대한 만화적 표현의 예, (b) 본 논문에서 제시하는 방법으로 구(sphere)를 렌더링한 결과

## 2. 관련 연구

### 2.1 만화적 표현을 위한 비사실적 렌더링

본 장에서는 비사실적 렌더링 분야 중에서 본 논문과 관련이 있는 기존의 연구 결과들을 간략히 살펴본다.

스타일 렌더링(stylized rendering) 분야에서 Gooch 등은 실제 예술가들이 물감을 사용하여 색상을 쓰는 것과 같은 느낌을 주기 위해 전통적인 휘도(luminance) 기반의 셰이딩 톤(shading tone)에 덧붙여 색상(hue)의 변화에 의한 셰이딩 톤 계산 방법을 국소 반사 모델에 추가하였다[1]. 금속성 물체의 반사를 표현하기 위하여 Phong 셰이딩 모델에 근거하여 동일한 기법을 적용한 결과를 제시하였지만, 반사 영역을 만화적인 표현에 이르게 하는 연구에는 미치지 못하는 결과를 보였다. Decaudin은 하드웨어 가속을 받는 다중 텍스처링(multi-texturing) 기법과 이미지 처리 기법을 조합하여 만화적인 느낌을 주는 렌더링 기법을 소개했다[2]. Lake 등은 모델의 법선 벡터와 광원을 향하는 라이트 벡터의 내적 값을 임계값(threshold)으로 하여 구분한 영역에 대한 셰이딩을 하드웨어 가속을 받는 텍스처링 기법을 이용하는 방법을 소개하였다[3]. 임계값에 의하

여 모델은 빛을 받는 밝은 영역과 상대적으로 빛을 못 받는 어두운 영역으로 분리되어, 제한된 개수의 색상만을 셰이딩에 사용하게 되며 이의 결과로 극단적으로 모델의 상세성이 사라져 만화적인 효과를 얻을 수 있게 된다.

물체의 비사실적 표현을 위해 셰이딩 기법과 더불어 물체의 윤곽선(silhouette)을 예술가가 그린 것처럼 표현하는 기법들도 비사실적 렌더링 분야에서 중요하게 다루고 있는 연구 분야이다[4,5,6,7]. 윤곽선을 회화적으로 처리하기 위한 기법들은 모델 공간에서의 처리와 더불어 이미지 공간에서의 처리를 혼합하여 사용하고 있다. 모델 공간에서 주어진 시점(view point)에 따라서 물체의 윤곽선 정보를 찾아낸 후, 그래픽 하드웨어 가속을 이용하여 윤곽선 정보만을 렌더링을 하고, 결과 이미지를 분석하여 화면에 보이는 선분들을 찾아내서 해당하는 선분 정보에 다양한 스타일의 스트로크 효과들을 적용하기 위한 다면체 정보를 구성하는 기법을 일반적으로 사용하고 있다. 하지만 기본적으로 윤곽선에 특정한 효과를 주기 위한 단계는 대체로 이미지 공간에서 이루어지기 때문에 하드웨어의 가속 성능을 최대한 사용하지 못한다는 단점을 가지고 있다.

위에서 소개한 연구들은 물체의 기하학적 정보를 보다 확연하게 표현한다거나, 회화적인 윤곽선을 얻는다는 등의 특정한 응용분야에서 비교적 좋은 결과를 보여주고 있다. 하지만 물체의 반사 효과 표현을 위해서 아직도 기본적인 국소 반사 모델에서 얻어지는 결과를 그대로 사용한 경우가 대부분이기 때문에 반사 영역의 경우 만화나 회화적 느낌의 렌더링 보다는 사실적인 렌더링의 결과에 좀 더 가깝다는 단점이 있다. 물체의 반사 영역에서 좀 더 만화적인 느낌이 나는 연구 결과들은 대부분 이미지 공간에서의 처리가 필요로 했는데 이미지 공간에서의 처리는 생각보다 복잡하고 하드웨어의 가속을 받을 수 없기 때문에 속도의 향상도 쉽게 꾀할 수 없는 단점이 있다.

### 2.2 국소 반사 모델

본 논문에서는 물체의 반사 영역을 그래픽 하드웨어의 가속 성능을 이용하여 만화적으로 표현하는 방법을 제시한다. 따라서 반사 효과를 제외한 부분은 기존의 만화적 렌더링 기법을 어떠한 것이라도 선택하여 사용할 수 있다. 본 논문에서 제시하는 방법은 전통적인 국소 반사 모델의 반사광 계산을 그대로 이용한다. Lake 등은 산란광에 대한 만화적 표현 방법을 얻기 위하여 국소 반사 모델을 그대로 사용하고[3], 따라서 구현의 편리성과 계산의 유사성을 고려하여 Lake 등의 산란 효과를 본 논문에서 거의 그대로 사용하고, 반사 효과를 계산하여 만화적으로 표현하는 부분을 추가하였다.

기본적인 국소 반사 모델은 식(1)과 같다.

$$I = I_a k_a + I_l (k_d (L \cdot N) + k_s (N \cdot H)^n) \quad (1)$$

$I$  : 물체가 빛을 받아 보이는 색상

$I_a$  : 앰비언트 광원의 세기(ambient intensity)

$I_l$  : 광원의 세기(light intensity)

$k_a$  : 물체의 앰비언트 상수(ambient coefficient)

$k_d$  : 물체의 산란 상수(diffuse coefficient)

$k_s$  : 물체의 반사 상수(specular coefficient)

$L$  : 광원을 향하는 단위 벡터

$N$  : 현재 위치에서의 법선 벡터

$H$  : 카메라를 향하는 벡터와  $L$  의 평균 벡터

$n$  : 반사영역 집중도 제어 상수

### 3. 반사 효과의 만화적 렌더링

물체의 반사 효과를 만화적으로 렌더링 하기 위한 개략적인 방법은 다음과 같다. 물체의 모든 정점에 대하여 식(1)을 사용하여 정점의 빛의 세기를 계산한다. 임의로 주어진 두 상수를 임계값으로 하여 모든 정점을 어두운 정점, 밝은 정점과 반사 정점으로 구분한다. 여기까지의 방법은 정점에 대한 빛의 세기를 계산할 때 반사광의 효과를 포함한다는 점, 임계값이 두 개라는 점, 세 종류의 정점으로 나뉜다는 점을 제외하고는 기본적으로 Lake 등의 방법[3]과 동일하다.

일단 반사 정점을 구하면, 모든 반사 정점을 서로 연결되어 있는 군집으로 분할한다. 각각의 군집에 대하여 적절한 텍스처 투영기를 생성하고, 패턴화된 텍스처를 반사 영역에 투영하여 만화적인 반사 효과를 얻는다.

#### 3.1 반사 정점의 군집 구성

반사 영역을 정확하게 찾기 위해서는 물체에 대한 렌더링이 끝난 이미지를 화소 단위로 검색하는 것이 가장 정확한 방법이지만 이미지 기반의 처리는 그래픽 하드웨어의 가속성능을 사용하기 어려워 시간도 많이 걸릴 뿐만 아니라 매우 복잡하기 때문에 본 논문에서는 모델 공간상에서 모델의 정점을 조사하는 방법을 사용하였다.

모델의 정점 단위로 조사한다고 했을 때 가장 정밀한 방법이 모든 정점을 검사하는 것이지만, 본 논문에서는 좀더 빠르고 상대적으로 효율적으로 처리를 위해 다소 정확성을 희생하는 형태인 무작위 검사(randomized test) 방법을 사용한다. 매 렌더링 프레임마다 반사 영역 내에 포함되는 모든 정점을 찾아내기 위해서 모델의 모든 정점을 조사하는 것이 아니라 임의로 선택된 일부의 정점만을 선택하여 조사한 후, 반사 영역 안에 들어 있는 정점들의 일부분을 찾아낸다. 일단 반사 영역 안에 들어가는 정점을 찾아내면, 해당 정점과 연결되어 있는 정점들을 차례대로 검사하여 계속해서 해당 반사 영역에 포함되는 정점들을 확장해 나간다. 결과적으로 모든 반사 정점에 대해서 서로 연결되어 있는 반사 영역의

집합(군집)을 구할 수 있게 된다. 광원, 물체 혹은 시점의 이동에 따른 애니메이션이 있는 경우에는 프레임간의 일관성을 고려하여 이전 프레임에서 반사 정점으로 기록되었던 정점부터 조사하기 시작한다.

#### 3.2 텍스처 투영기 설정

본 연구에서는 각각의 반사 영역을 하나의 패턴화된 텍스처를 사용하여 표현하였다(그림 2 참조). 이러한 텍스처는 실제 만화나 일러스트 화면에 자주 등장하는 예로서 실제 일러스트 및 만화작품 등에서 참고하였다. 본 논문에서는 투영 텍스처(projective textures)를 이용하여 패턴화된 텍스처를 모델에 직접 투영하였다[8].

텍스처 투영기(projector)를 사용하는 방법은 렌더링 파이프 라인에서 뷰 변환(view transform)을 하는 방법과 원리적으로 동일하다. 뷰 변환의 경우 카메라의 위치와 방위(orientation)에 의하여 하나의 고유한 카메라의 특성이 결정된다. 이때 방위는 시선 벡터(look vector), 상향 벡터(up vector)와 우향 벡터(right vector)에 의하여 유일하게 결정되며, 세 벡터는 단위 벡터로서 상호 수직(orthogonal)이라고 가정한다. 이렇게 구성된 세 벡터에 의하여 직교 좌표계가 형성되고 물체의 모든 정점은 해당 직교 좌표계에 의하여 뷰 공간(view space)으로 변환된다. 텍스처 투영기도 이와 같은 원리로 투영기의 위치( $P_{proj}$ ), 투영기의 시선 벡터( $V_{look}$ ), 상향 벡터( $V_{up}$ ) 및 우향 벡터( $V_{right}$ )를 가지고 하나의 고유한 투영기가 표현된다. 이때 투영기에 의하여 형성되는 직교 좌표계로 변환된 좌표 값이 해당 정점의 텍스처 좌표로 사용된다.

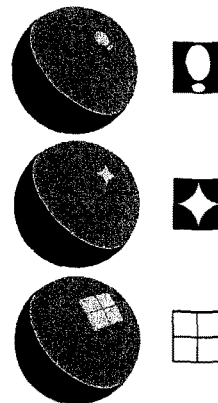


그림 2 : 오른쪽 패턴화된 텍스처를 사용하여 구에 투영한 결과

##### 3.2.1. 투영기의 위치 및 시선 벡터 결정

투영기의 위치를 정하기에 앞서 투영기가 어떠한 지점을 보고 있어야 하는가에 대해서 먼저 결정하도록 한다. 투영기의 위치에서 군집을 이루는 반사 영역 내 정점의 평균 위치( $P_m$ )가 투영의 중심(center of projection)이 되도록 시선 벡터를 결정하는 것이 가장 직관적으로 생

각할 수 있다. 따라서 이제 투영기의 위치( $P_{proj}$ )만 설정하면 시선 방향( $V_{look}$ )은 식(2)와 같이 구할 수 있다.

$$V_{look} = P_m - P_{proj} \quad (2)$$

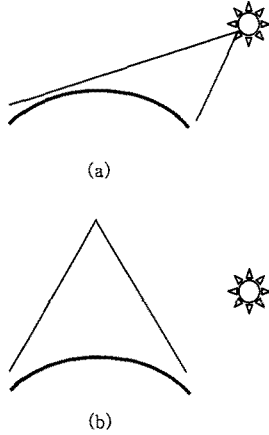


그림 3 : (a) 광원 방향에 투영기를 놓는 경우, (b) 정점의 평균 법선 방향에 투영기를 놓는 경우

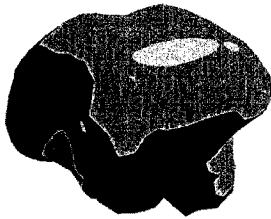


그림 4 : 텍스처 투영기를 광원 방향에 놓고 렌더링한 결과

투영기의 위치가  $P_m$ 으로부터 어느 방향에 놓여 있어야 하는가에 대해서 생각해 보면 몇 가지 쉽게 생각해 낼 수 있는 방향들이 있다. 반사 영역은 광원에서 출발한 빛이 물체에 반사되어 카메라로 직접 들어오는 경우에 해당하는 위치에 생기는 것이므로  $P_m$ 으로부터 광원을 향하는 방향에 투영기의 위치를 설정하는 것이 직관적이다(그림 3(a) 참조). 하지만 실제 구현 시에 투영기로부터 영향을 받지 않아야 하는 정점들도 투영기의 영향을 받는 일이 발생할 수 있다(그림 4 참조). 이러한 현상은 물체의 3차원 정점 위치 정보를 투영기에 의하여 형성된 좌표계로 변환하여 2차원 텍스처 좌표 정보만을 사용하기 때문에 여러 다른 위치의 정점들이 동일한 텍스처 좌표 정보를 갖게 되어 발생하는 문제이다. 일반적으로 뷰 변환에서 투영 변환시에 3차원 정보 중에서 한

좌표 값은 깊이 정보로 사용되어 카메라의 위치에서 보이지 않는 부분 등을 z-버퍼 등을 이용하여 하드웨어적으로 처리할 수 있다. 그러나, 텍스처 투영기에 의한 변환 과정에서는 하드웨어적인 도움을 받을 수 없다.

따라서 이러한 문제를 비교적 쉽게 처리할 수 있는 위치에 투영기의 위치를 설정하는 것이 더 적절하며, 본문에서는 그 위치로 군집을 이루는 정점들의 평균 법선 벡터(average normal vector) 방향을 사용한다(그림 3(b) 참조). 영향을 받지 않는 정점들은 투영기에서 본 좌표의 깊이 값(depth value)이 일정 범위를 벗어나는 경우이기 때문에 평균 법선 벡터 방향에서 정점의 깊이 범위가 상대적으로 작아져서 영향을 받지 말아야 할 정점들이 해당 깊이 범위에 들어올 가능성이 작아진다. 실험결과 렌더링된 결과의 품질에서는 거의 차이가 없었다. 식(3)에 의하여 투영기의 위치가 결정된다.

$$P_{proj} = d N_m + P_m \quad (3)$$

$d$ :  $P_m$ 으로부터 투영기까지의 거리

$N_m$ : 군집을 이루는 정점들의 평균 법선 벡터

### 3.2.2 투영기의 상향 벡터 결정

투영기를 정의하는 요소 중 하나인 상향 벡터는 다른 요소들과는 달리 매우 휴리스틱(heuristic)한 성격을 가지고 있다. 실제 렌더링 시에 상향 벡터를 어떻게 설정하느냐에 따라서 텍스처의 방향이 결정되기 때문에, 패턴화된 텍스처가 특정한 접대칭 구조를 가지고 있지 않다면 매우 다른 형태의 반사 모양을 가지게 될 것이다. 때문에 상향 벡터의 경우에는 장면(scene)의 특성과 모델의 모양 특성 같은 여러 가지 경험적 요소를 고려하여 응용 분야 마다 적절하게 설정을 해주어야 한다.

모델의 국소적인 특징으로부터 상향 벡터를 결정하는 방법 중 하나로 투영의 중심이 되는 정점에서 주 곡률(principal curvature) 방향을 이용하는 방법 등을 사용할 수 있다. 본 논문에서는 구현의 편리성을 고려하여 군집을 이루는 정점들의 OBB(Oriented Bounding Box)를 구한 후 OBB를 구성하는 축 중에서 가장 긴 축을 이루는 방향벡터( $V_p$ )를 이용하여 상향 벡터를 결정한다. OBB의 가장 긴 축 방향을 시선 방향 벡터( $V_{look}$ )에 수직인 평면에 투영한 벡터를 투영기의 상향 벡터( $V_{up}$ )로 사용한다(그림 5 참조). 식(4)에 의하여 투영기의 상향 벡터가 결정된다.

$$V_{up} = V_p - (V_p \cdot V_{look}) V_{look} \quad (4)$$

### 3.2.3 투영 작업

모든 반사 영역의 군집별로 텍스처 투영기가 설정 되면 이제 매 렌더링 프레임마다 실제로 텍스처를 투영하는 작업을 수행한다. 이 작업은 컴퓨터 그래픽스에서 뷰 변환과 투영 변환(projection transform)하는 작업에 해당하는 방법과 완전히 동일하다. 우선 위의 단계에서 설정

이 된 투영기의 파라미터들을 가지고 뷰 행렬(view matrix)과 투영 행렬(projection matrix)을 만들어서 곱하여 최종 변환 행렬을 만들어 낸다. 모델의 모든 정점의 위치 벡터에 대해서 최종 변환 행렬에 단순히 벡터-행렬 곱을 수행하면 모든 정점에 대한 텍스처 좌표가 얻어진다.

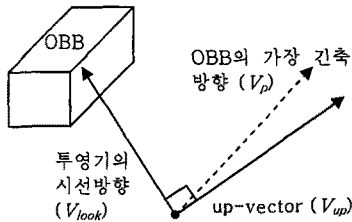


그림 5 : 상향 벡터의 한 예.  $V_{look}$ 에 수직인 평면에  $V_p$ 를 투영한 결과.

#### 4. 실험 결과

본 논문에서 제시한 방법을 Microsoft DirectX 9.0에서 Vertex Shader 2.0과 Pixel Shader 2.0을 이용하여 구현하였다. 실험에 사용한 컴퓨터의 사양은 Intel Pentium 4 2.53G 프로세서에 ATI Radeon 9700이다. 그림 6에는 본 논문에서 제시한 방법을 사용하여 다양한 모델에 대한 반사 영역을 만화적으로 표현한 결과를 제시하였다. 몇 가지 모델의 특성과 렌더링 수행 성능은 다음 표 1과 같다.

모델이름	정점의 수	면의 수	FPS
구	2452	4900	59.87
비너스	711	1418	85.93
사람	3105	4648	52.59
유령	308	564	329.43
그렘린	5873	10374	21.25

표 1 : 반사영역의 만화적 렌더링에 사용된 일부 모델의 특성과 렌더링 수행 성능

#### 5. 결론 및 향후 연구 과제

본 논문에서는 패턴화 된 텍스처를 이용하여 반사 효과를 만화적으로 렌더링하는 새로운 방법을 제시하였다. 기존의 많은 비사실적 렌더링 기법들이 이미지 공간에서의 처리가 필요했던 데에 비해 모델 공간에서의 연산을 통해서 그래픽 하드웨어의 가속 성능을 최대한 이용하였으며, 기존 만화적 렌더링에서 부족하였던 반사 물체의 광원에 대한 반사 효과를 만화적인 느낌으로 표현할 수 있었다.

본 논문에서 제시된 방법에 의하여 생성되는 반사 영역의 모양은 텍스처 투영기의 상향 벡터에 크게 영향을 받는다. 상향 벡터를 다면체 모델의 곡률(curvature) 등 모델의 기하학적 정보에 내재된 특징으로부터 얻을 수 있는 일관된 방법에 대한 추가적인 연구가 필요하다. 또한 현재의 구현은 모델이 애니메이션 되는 경우에는 프레임간의 연관성을 크게 고려하지 못하였다. 애니메이션에서는 모델의 모양이 변하고, 이에 따라서 반사 영역의 변화가 일어나며, 필연적으로 상향 벡터의 변화가 급격히 일어날 가능성이 있다. 이에 따라서 반사 영역의 만화적인 표현이 급격히 변화할 가능성이 발생하고, 이에 대한 해결책으로서 프레임간의 연관성을 고려하는 방법에 대한 추가적인 연구가 필요하다.

#### Acknowledgement

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음.

#### 참고문헌

- [1] A. Gooch, B. Gooch, P. Shirley, and E. Cohen, "A Non Photorealistic Lighting Model for Automatic Technical Illustration," In Proceedings of SIGGRAPH 98, pp. 447-452, 1998.
- [2] P. Decaudin, "Rendu de scènes 3D imitant le style dessin animé." Rapport de Recherche 2919, Institut National de Recherche en Informatique et en Automatique. 1996.
- [3] Lake, C. Marshall, M. Harris, and M. Blackstein, "Stylized rendering techniques for scalable real-time 3D animation," In Proceedings of NPAR 2000, pp. 13-20, 2000.
- [4] L. Markosian, "Art-based Modeling and Rendering for Computer Graphics," Ph.D. thesis, Brown University, 2000.
- [5] L. Markosian, M. Kowalski, S. Trychin, L. Bourdev, D. Goldstein, and J. Hughes, "Real-Time Nonphotorealistic Rendering," In Proceedings of SIGGRAPH 97, pp. 415-420, 1997.
- [6] J. Northrup, and L. Markosian. "Artistic Silhouettes: A Hybrid Approach," In Proceedings of NPAR 2000, pp. 31-38, 2000.
- [7] R. Kalnins, P. Davidson, L. Markosian, A. Finkelstein, "Coherent Stylized Silhouettes," ACM Trans. of Graphics(SIGGRAPH 2003), Vol. 22, No. 3, pp. 856-861, July 2003.
- [8] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haerberli. "Fast shadows and lighting effects using texture mapping," In Proceedings of SIGGRAPH '92, pp. 249-252, 1992.



그림 6 : 만화적인 반사효과를 투영 텍스처를 사용하여 렌더링한 결과