

실시간 응용을 위한 카툰 렌더링 기술

이환직* · 최정주**

1. 서론

컴퓨터 그래픽스의 렌더링 분야는 전통적으로 물리적 현상에 기반을 둔 사실성을 추구하는 사진 품질의 렌더링(photorealistic rendering)에 초점을 맞추어 연구가 진행되어 왔다. 최근 몇 년 사이에 인간 감성의 특수성에 기반을 둔 비사실적 렌더링(non-photorealistic rendering) 분야의 발전이 두드러지고 있다. 삽화, 회화, 만화 등과 같이 우리 생활에 비교적 익숙한 분야에서부터 게임과 같이 최근 주목받는 분야에 이르기까지 비사실적 렌더링의 응용 분야가 나날이 확대되는 추세를 보이고 있다. 비사실적 렌더링의 연구는 회화, 만화 및 삽화 등에서 나타나는 숙련된 작가의 기법을 컴퓨터를 이용하여 흉내 내어, 보다 흥미로운 화면을 얻는데 주요한 목적이 있다.

비사실적 렌더링 분야 중에서 카툰 렌더링(cartoon rendering)은 만화 혹은 애니메이션 작품에서 관찰되는 작가의 의도를 3차원 그래픽스 환경에 적용하여 만화적인 느낌의 화면을 얻는 것을 주된 목적으로 한다. 카툰 렌더링의 주요 연구과제로는 만화적인 명암을 얻는 만화적 셰이딩(toon shading) 기술, 물체의 윤곽선(silhouette)을 얻는 기술 등의 전통적인 주제와, 만화적 표현

을 향상시키기 위한 반사효과의 표현과 같은 새로운 연구 주제가 있다.

본 기고에서는 만화적 표현을 얻기 위한 전통적인 연구 주제를 간략히 살펴보고, 기존의 만화적 렌더링 기법을 향상시키기 위한 만화적 반사효과 표현 기술의 연구와 관련된 새로운 경향을 살펴본다. 특별히 카툰 렌더링 관련 주제 중에서 게임과 같은 실시간 응용에 초점을 맞추어 기존의 연구결과와 새로운 경향을 소개하고 장단점, 개선 방향, 추후 연구방향 등을 살펴본다.

2. 카툰 렌더링의 전통적인 연구 주제

2.1 만화적 셰이딩

만화 혹은 애니메이션에 등장하는 캐릭터의 외관에서 주목할 만한 시각적 특징 중에 하나는 캐릭터를 표현하는 색상의 단순함에 있다(그림 1 참조). 만화 혹은 애니메이션에 등장하는 캐릭터들은 작가의 의지에 따라서 의도적으로 2차원적으로 표현되며, 3차원 모델에 대한 2차원적인 표현은 대개 2~3개의 단순한 색상(solid color)만을 사용하여 셰이딩(shading) 함으로써 달성된다. Lake는 면에서 발생하는 산란광(diffuse lighting)의 단순화를 위해서, Gouraud 셰이딩의 결과로 얻어진 조명도(intensity)를 2개의 색상만으로 극

* 아주대학교 정보통신전문대학원 박사과정

** 아주대학교 미디어학과 교수



그림 1. 게임에 등장하는 카툰 렌더링(2)

단적으로 단순하게 표현하는 방법을 제시하였다 [1]. Gouraud 셰이딩에 의한 조명도 계산은 환경광(ambient lighting)의 효과를 무시하면 전적으로 광원을 향하는 단위 벡터(L)와 정점에서의 단위 법선 벡터(N)의 내적(dot product)으로 표현되는 함수로서 식(1)과 같이 정의된다.

$$I = I_a k_a + I_d k_d (\mathbf{L} \cdot \mathbf{N}) \quad (1)$$

이때, 0.0과 1.0사이의 특정한 값을 임계점으로 하여 두개의 색상으로 분리한 1차원 텍스처를 준비하고, Gouraud 셰이딩의 조명도를 텍스처 좌표로 사용하여 렌더링하면 명암 영역이 확연히 구분되는 단순화된 만화적 표현을 얻을 수 있다 (그림 2 참조).

Lake의 연구결과는 오늘날 보편적인 수준의 그래픽 가속장치의 기능을 그대로 사용할 수 있으므로, 빠른 수행성능을 보인다는 장점을 가지고 있다. 구현이 대단히 간단하고, 빠른 수행성능으로 인해서 게임과 같은 실시간 응용분야에 적용하기 매우 용이하다.

2.2 윤곽선 추출

만화와 애니메이션에서 관찰되는 만화적 렌더링의 주요한 특징 중에 또 다른 하나는 캐릭터의

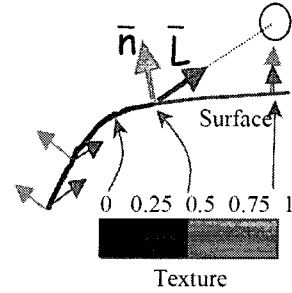


그림 2. 카툰 셰이딩의 결과(1)

윤곽선이 두드러지게 표현된다는 점이다. 캐릭터의 윤곽선은 배경으로부터 캐릭터의 외양을 분리하는 기본적인 역할과 더불어, 캐릭터의 질감, 무게감, 분위기의 전달 등을 위한 수단으로 사용되기도 한다. 만화 혹은 애니메이션에서 윤곽선이 강조되는 이유는 전통적인 셀 애니메이션(cell animation) 제작의 특징에 기인하는 것으로 볼 수 있다. 대개 셀 애니메이션에서는 스케치로 부터 원화를 얻는 과정에서 라이닝 과정을 거치게 되는데, 이 단계에서 윤곽선이 뚜렷이 드러나게 된다.

3차원 다면체 모델에 대한 윤곽선을 얻기 위한 연구들은 크게 이미지 공간(image space) 알고리즘, 오브젝트 공간(object space) 알고리즘과 혼합(hybrid) 알고리즘으로 나눌 수 있다[3]. 이미지 공간 알고리즘은 깊이 버퍼(depth buffer)나 법선 맵(normal map) 등에 저장된 영상의 불연속성을

발견하여 물체의 윤곽선을 얻는 방법이며, 오브젝트 공간 알고리즘은 윤곽선의 특징으로부터 해석적(analytic)인 방법으로 정확한 윤곽선을 표현하는 방법이다. 혼합 알고리즘은 물체의 면을 축소하거나 확대하는 등의 수정을 가한 후 z-버퍼 등에 렌더링한 결과로부터 윤곽선을 추출하는 방법이다. 오브젝트 공간 알고리즘은 정확한 윤곽선으로부터 다양한 스타일을 적용할 수 있는 장점이 있지만, 가시성(visibility) 문제를 해결하기 위한 부담이 있는 단점을 동시에 가지고 있다. 이미지 공간 알고리즘과 혼합 알고리즘은 가시성 문제가 자동으로 해결되고, 하드웨어 가속장치의 도움을 받을 수 있는 장점이 있지만, 다양한 표현의 윤곽선을 얻기 위한 추가적인 처리가 어려운 단점을 가지고 있다.

윤곽선을 얻기 위한 가장 간단한 방법은 캐릭터의 윤곽선이 가지는 시점 종속적인 특징을 이용하는 방법이다. 윤곽선은 3차원 모델에서 선분을 공유하는 두 면이 각각 시점을 향하는 방향(front face)과 시점에서 멀어지는 방향(back face)인 경우를 특징으로 한다(그림 3 참조).

모든 선분에 대하여 간단하고 동일한 연산을 반복하면 되므로 하드웨어적인 가속성을 이용하여 실시간 응용에 적용하기 손쉬운 장점을 가지고 있다. 다만 시점 혹은 캐릭터의 모델이 시간에 따라서 변화하는 경우 두 연속된 장면의 윤곽선 사이에서 연관성을 이용하지 못하기 때문에 매번 같은 연산을 반복해야 한다는 단점을 가지고 있다.

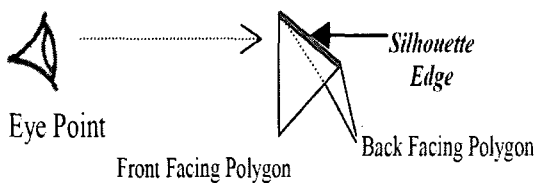


그림 3. 윤곽선의 시점 종속적 특징(1)

3. 만화적 반사효과의 표현

최근 만화적인 표현을 사용하는 게임은 실시간성의 제약사항으로 인해서 기본적으로 국소 반사 모델(local reflection model)에 기반한 만화적 셰이딩(toon shading) 기법을 주로 사용하고 있다. 그러나, 앞서 살펴본 대로 기존의 만화적 셰이딩 기법은 산란(diffuse reflection) 효과에 기반하여 모델에서 색상의 변화를 단순화하는 방법을 사용하고 있기 때문에 금속 혹은 유리재질의 매끄러운 표면에서 관찰되는 반사(specular reflection) 효과의 표현에 취약하다. 반사광은 물체의 질감 표현과 모델의 국소적 모양, 환경 변수의 영향 등을 반영하는 주요한 표현의 수단이다. 따라서 반사 재질의 물체를 보다 효과적으로 표현하는 것은 전체적으로 보다 풍부한 표현을 얻는 길이 될 것이다.

만화적 반사 효과에서도 다른 만화적 표현과 마찬가지로, 작가의 의도가 반영되는 경우가 일반적이다. 그림 4에서 유리창에 비추는 반사효과는 유리창이 평평하다는 느낌을 주고 있으며, 칼에 비치는 반사효과는 칼의 날카로움을 표현하기 위한 수단이 된다.

2.1절에서 설명한 Lake의 만화적 셰이딩 기법에서도 조명도에 따라서 더 많은 단계로 구분된 텍스처를 사용하면 부분적으로 반사광을 표현하는 방법으로 사용할 수 있다. 혹은 폰그 셰이딩(Phong shading) 모델을 부분적으로 사용하여 반

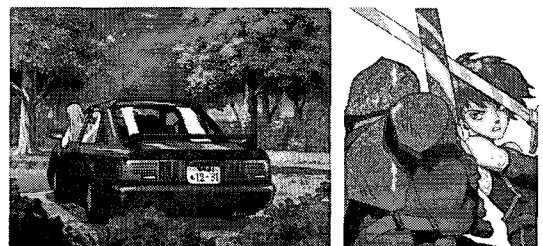


그림 4. 만화적 반사효과의 예(5)

사광을 표현할 수도 있다. 그러나 이러한 조명 모델은 근본적으로 부드러운 형태의 반사 효과만을 표현할 수 있으며, 작가의 의도가 반영된 만화적인 형태의 반사광을 표현하는 데에는 분명한 한계가 있다. 여기서는 만화적인 반사효과의 표현을 위한 최근의 연구 결과를 중점적으로 살펴본다.

3.1 중간 벡터를 이용한 표현 방법

Anjyo와 Hiramitsu는 Blinn의 반사모델[4]에 의하여 계산된 반사영역에서 중간 벡터(halfway vector)를 수정하여 만화적 반사효과를 얻는 방법을 제시하였다[5]. Blinn의 반사 모델에서 반사 효과는 다음과 같은 식(2)로 표현될 수 있다.

$$I_s = k_s(N \cdot H)^n, \tag{2}$$

이때 반사 효과를 위한 중간 벡터 H 는 $H = (L + V)/2$ 와 같이 정의되며, L 은 광원을 향하는 단위 벡터이며, V 는 카메라(시점)를 향하는 단위 벡터이다. 캐릭터 위의 점 p 에서 중간 벡터 H 를 변경하면, 이에 따라서 광원의 위치는 $L = 2(H \cdot V)H - V$ 로 변화하게 된다. 즉, 점 p 에서 반사효과를 일으키는 가상의 광원 L 을 새로이 정의할 수 있게 된다. 따라서 사용자가 원하는 위치에서 의도적으로 중간 벡터를 법선 벡터와 일치시켜 사용자가 원하는 위치에서 반사광이 발생할 수 있도록 하는 것이다.

중간 벡터를 변화시켜 반사효과를 표현하는 방법은 반사영역을 원하는 위치에, 원하는 모양으로 다양하게 변화시킬 수 있는 장점을 가지고 있다. 먼저 반사영역을 임의의 위치로 이동, 회전, 축소 및 확대 하는 연산은 다음과 같이 정의된다.

반사 영역의 이동

모델 위의 점 p 에서의 접평면(tangent plane)을 확장(span)하는 두개의 선형독립(linearly in-

dependent)인 단위 벡터 du, dv 에 대하여 $H' = H + \alpha du + \beta dv$ 로 정의할 수 있다. 이때 H' 를 정규화(normalize) 하여 모델의 렌더링 시에 적용하면 각각 du, dv 방향으로 α 와 β 만큼 이동한 새로운 반사영역 표현을 얻을 수 있게 된다.

반사 영역의 회전

모델 위의 점 p 에서의 접평면을 구성하는 두 벡터 du, dv 에 의하여 얻어지는 (u, v) 좌표계를 점 p 에서의 법선 벡터를 축으로 하여 회전하는 행렬식을 모델의 중간 벡터 필드에 적용함으로써 회전된 반사영역을 얻을 수 있다.

반사 영역의 축소 및 확대

모델 위의 점 p 에서 접평면을 구성하는 du 방향으로 축소 및 확대하는 경우를 가정하여 보자. 먼저 점 p 에서의 H 벡터를 임의의 양수 δ 에 대하여 $H' = H + \delta(N - H)$ 로 치환하게 되면, δ 가 1.0에 근접할수록 H' 은 N 에 근접하여 점 p 가 반사영역에 포함되게 된다. 이를 du 방향으로 적용하면, 식(3)의 새로운 중간 벡터 필드를 얻을 수 있다.

$$H'' = H + \delta((N - H) \cdot du)du = H - \delta(H \cdot du)du \tag{3}$$

점 p 에서의 H 가 처음부터 N 에 근접해 있는 경우에는 위의 식(3)에서 H'' 은 H 와 유사하여 거의 변화가 없을 것이지만, H 가 N 와 처음부터 충분히 다른 경우에는 H'' 은 N 에 접근하게 된다. 이에 따라서 점 p 가 새로이 반사영역에 포함되어 반사영역이 du 방향으로 커지게 된다. 만일 δ 가 음수가 되면 반사영역은 du 방향으로 축소된다. 같은 방법으로 dv 에 대하여 축소 및 확대 하는 연산을 구성할 수 있다.

Anjyo와 Hiramitsu의 연구는 반사영역의 모양을 사용자가 원하는 모양으로 변형할 수 있는 기능을 포함하고 있는데 반사영역의 분할(split)과 사각화(squaring)를 예로 들 수 있다.

반사 영역의 분할

음이 아닌 상수 γ_1 과 γ_2 에 대하여 다음과 같은 연산을 적용한다.

$$H' = H - \gamma_1 \text{sgn}(H \cdot du)du - \gamma_2 \text{sgn}(H \cdot dv)dv \quad (4)$$

이때 $\text{sgn}(x)$ 함수는 x 가 음수이면 -1을, 그 외의 경우에는 1을 전달한다. 이 연산은 확대 연산자가 H 를 N 에 근접시키는 것과는 달리, $\text{sgn}(x)$ 함수의 특성에 따라서 H 가 N 으로부터 일정한 차이를 벗어나게 되면 H 가 N 으로부터 멀어지는 효과를 가지게 된다. 따라서 일정한 거리 이상에서는 원래의 반사영역을 반사영역이 아닌 것으로 표현할 수 있게 하는 효과가 나타나 반사영역이 분할되게 된다. γ_1 혹은 γ_2 의 값을 0과 0이 아닌 양수의 값으로 조절하여 2분할 혹은 4분할의 효과를 거둘 수 있다.

반사 영역의 사각화

Blinn의 반사 모델은 물체가 부드러운 형상을 가진 경우 반사 영역이 부드러운 모양으로만 나타나는 문제를 가지고 있다. 사각화 연산자는 접평면의 du 및 dv 방향으로 반사영역을 사각형에 근접한 형상으로 만들기 위한 것으로 다음과 같이 정의된다.

$$\begin{aligned} \theta &= \min(\cos^{-1}(H \cdot du), \cos^{-1}(H \cdot dv)), \\ \text{sqrnorm} &= \sin(2\theta)^n, \\ H' &= H - \sigma \times \text{sqrnorm}((H \cdot du)du + (H \cdot dv)dv). \end{aligned} \quad (5)$$

이상과 같은 연산자를 이용하여 사용자는 원하는 반사영역의 모양을, 원하는 위치에, 원하는 크기와 방향으로 만들 수 있게 된다(그림 5 참조).

위와 같은 연산자 이외에 후처리를 위한 연산자를 추가로 사용하여 다양한 효과의 반사영역 표현을 얻을 수 있다(그림 6 참조).

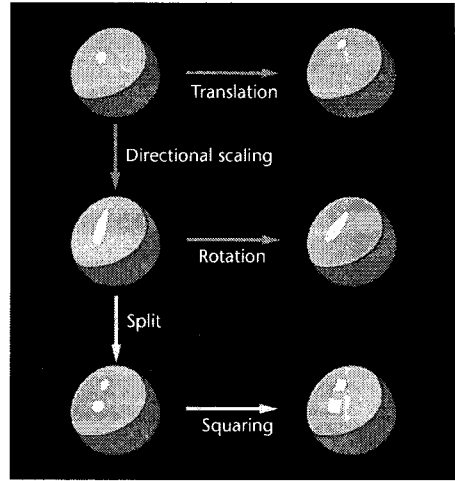


그림 5. 만화적 반사효과 표현을 위한 중간 벡터 변환 연산자(5)

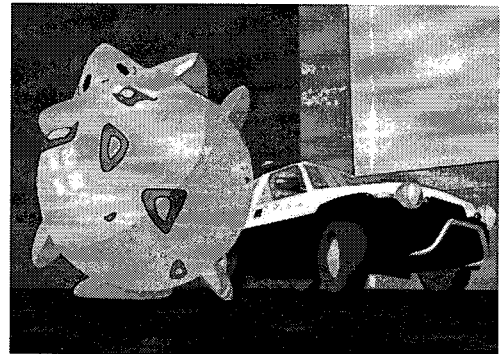


그림 6. 중간 벡터를 이용한 만화적 반사효과 표현의 예(5)

3.2. 투영 텍스처를 이용한 표현 방법

Anjyo와 Hiramitsu가 제시한 중간 벡터를 수정하여 반사영역의 모양을 변화시키는 방법은 만화 및 애니메이션 제작자의 의도를 반영하는 반사 효과 표현을 위한 만화적 렌더링 기법의 새로운 전기를 마련하였지만, 몇 가지 단점을 가지고 있다. 우선 반사영역의 모양을 변화시키기 위해 중간 벡터를 수정하는 연산이 직관적이지 않아 사용자가 미세한 수정을 하기에 적합하지 않으며, 아울러 게임과 같은 실시간 응용 분야에 적용하기

어려운 단점을 가지고 있다. 본 절에서는 실시간 응용에 적용할 수 있는, 텍스처 투영기법(texture projection)을 이용하는 만화적 반사효과의 표현 방법을 설명한다. 기본적인 아이디어는 원하는 반사영역의 모양을 2차원 텍스처로 준비하고, 이를 원하는 위치에, 원하는 크기와 방향으로, 텍스처 투영기를 사용하여 투영하는 것이다. 따라서 투영 텍스처를 이용한 만화적 반사효과의 표현은 텍스처 투영기를 설정하는 문제로 귀결되게 된다.

3.2.1 텍스처 투영기의 설정

텍스처 투영은 기본적으로 3차원 그래픽스에서 카메라를 이용한 시점 변환(view transform) 및 투영 변환(projection transform)과 원리적으로 동일하다. 설명을 간단하게 하기 위하여 텍스처 투영기는 평행 투영(parallel projection)을 한다고 가정하도록 한다. 따라서 투영기의 위치(position)는 투영기의 설정에 영향을 미치지 않으며 단지 투영기의 방위(orientation)가 투영기 설정을 위한 유일한 매개변수가 된다. 이때, 투영기의 방위는 (Right, Up, Look) 벡터로 정의하도록 한다.

투영의 중심(center of projection)

텍스처 투영기는 캐릭터 모델과 광원, 시점에 따라서 결정되는 최대 조명점(maximal intensity point)을 향하게 된다. Blinn의 조명 모델에 의하면 임의의 점 p 에서의 조명도는 법선 벡터 N 과 중점 벡터 H 의 함수로, 최대 조명점에서는 $H=N$ 의 조건을 만족한다. 실시간 응용에 사용하는 캐릭터 모델은 비교적 적은 수의 정점(vertex)과 면(face)으로 구성되며, 비록 충분히 많은 수의 정점(혹은 면)을 가진 모델이라도 $H=N$ 의 조건을 만족하는 정점을 항상 구할 수 있는 것은 아니다. 이는 대개 최대 조명점은 정점이 아닌 면의 내부에 존재할 가능성이 훨씬 크다는 사실에 기인한

다. 정점에서의 N 과 H 로부터 최대 조명점의 위치를 찾는 방법은 다음과 같다.

캐릭터 모델을 구성하는 임의의 삼각형 F 에서 각 정점의 위치를 p_0, p_1, p_2 라 하고, 각 정점의 법선 벡터를 N_0, N_1, N_2 라 하며, 중간 벡터를 H_0, H_1, H_2 라 하자. 삼각형 F 내부에 최대 조명점이 존재한다면, 최대 조명점 p_{max} 는 중심좌표계(barycentric coordinate)로 다음과 같이 표현된다.

$$p_{max} = \alpha(p_1 - p_0) + \beta(p_2 - p_0) + p_0, \quad (6)$$

여기서 $0 \leq \alpha, \beta, \alpha + \beta \leq 1$ 이다. 같은 방법으로 최대 조명점에서의 법선 벡터 N_{max} 와 중간 벡터 H_{max} 는 다음과 같이 표현된다.

$$\begin{aligned} N_{max} &= \alpha(N_1 - N_0) + \beta(N_2 - N_0) + N_0, \\ H_{max} &= \alpha(H_1 - H_0) + \beta(H_2 - H_0) + H_0. \end{aligned} \quad (7)$$

최대 조명점의 조건 $N_{max} = H_{max}$ 로부터 위의 식은 다음과 같은 행렬식으로 표현된다.

$$[\Delta N_1 - \Delta H_1 \quad \Delta N_2 - \Delta H_2] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = [H_0 - N_0], \quad (8)$$

$$\Delta N_i = N_i - N_0, \quad \Delta H_i = H_i - H_0, \quad i = 1, 2.$$

위의 행렬식의 해로부터 α 와 β 가 구해지면, 이를 식(6)에 대입하여 최대 조명점의 위치를 구할 수 있다. 텍스처 투영기는 최대 조명점에서의 법선 벡터(N_{max})를 따라서 임의의 거리에 위치시킬 수 있고, 이에 따라서 투영기의 방위 중에서 시선 벡터(Look vector)는 $-N_{max}$ 로 결정된다.

투영기의 방위

투영기의 나머지 방위를 결정하기 위하여 최대 조명점에서의 주극률 방향을 이용할 수 있다. 최대 조명점에서의 주극률 방향은 다면체 모델에 대하여 Alliez가 제시한 해석 모델[6]과 중심좌표계를 사용하여 간단히 구할 수 있다.

최대 조명점에서의 주곡률 방향을 e_{max} , e_{min} 이라고 하면 (e_{min} , e_{max} , $-N_{max}$)은 서로 수직인 단위 벡터로서 3차원 공간을 확장하므로 텍스처 투영기의 방위로 사용할 수 있다(그림 7 참조). 왼손 좌표계 혹은 오른손 좌표계에 따른 e_{min} , e_{max} 의 부호 문제를 무시한다면, 투영기의 방위를 결정하기 위하여 상향 벡터(Up vector)를 e_{min} , e_{max} 중 어느 것으로 선택하는가의 문제가 남게 된다. 일반적으로 투영기의 상향 벡터의 결정은 텍스처의 방향성 특징에 의존적이며, 실험적인 고찰에 의하면 e_{max} 의 방향을 상향 벡터로 사용할 때 일반적으로 좋은 결과를 보이고 있다(그림 8 참조).

투영기의 방위가 유일하게 결정되면, 투영기의 시점 변환 행렬을 다음과 같이 결정할 수 있다.

$$V = [e_{min} \ e_{max} \ -N]^T \tag{9}$$

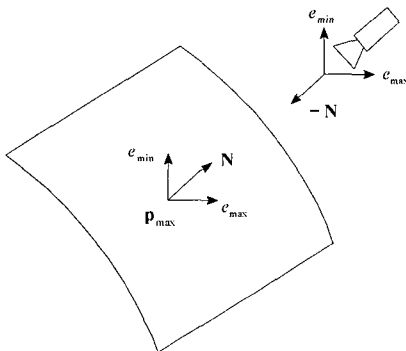


그림 7. 텍스처 투영기의 방위

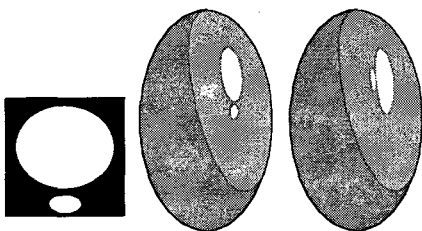


그림 8. 텍스처 투영기 상향 벡터의 결정 방법

투영 변환 행렬

텍스처 투영기의 설정을 위하여 남은 문제는 투영기의 투영 변환 행렬(projection transform matrix)를 구하는 문제이다. 문제를 간단하게 하기 위하여 투영의 중심으로부터 상하, 좌우로 대칭적인 크기를 갖는 투영을 가정한다. 투영기의 투영 변환 행렬은 투영 피라미드의 상하, 좌우의 크기에 영향을 받는다. 투영의 중심에서 주곡률 방향이 투영기의 상향 및 우향 벡터로 설정되었으므로, 투영의 중심에서의 곡률 반지름(radius of curvature)를 사용하여 투영 피라미드의 크기를 근사적으로 계산할 수 있다. 투영기의 상향 벡터(혹은 우향 벡터) 방향으로의 고유값(eigen value)을 κ 라 한다면, 투영 피라미드의 상하 방향의 크기는 다음과 같은 식으로 표현된다.

$$s = \min\left(2r, \frac{2}{|\kappa|} \sqrt{1-\sigma^2}\right), \tag{10}$$

여기서 σ 는 1.0에 가까운 상수이며, r 은 Alliez가 제시한 해석 모델[6]에서 곡률텐서를 구하기 위해 설정한 영역의 반지름으로서, κ 가 0에 가까워져 투영 피라미드의 크기가 지나치게 커지는 것을 방지하기 위하여 사용된다.

일단 시점변환행렬과 투영변환행렬이 구해지면, 투영 텍스처가 적용된 모델의 정점별 텍스처 좌표가 유일하게 결정되고, 프로그램 가능한 하드웨어 셰이더를 이용하여 빠르게 렌더링을 수행할 수 있다(그림 9 참조).

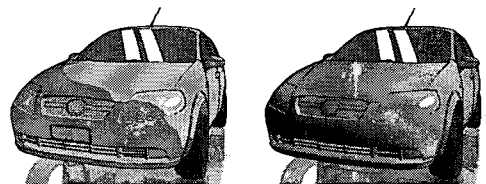


그림 9. 투영 텍스처를 이용한 만화적 반사 효과의 렌더링 예

반사 영역 변환

투영 텍스처를 이용한 반사 효과의 표현 방법을 이용하면, 기존에 Anjyo와 Hiramitsu의 연구에서 제시한 다양한 변환을 직관적으로 달성할 수 있는 장점을 가지고 있다. 반사 영역의 이동은 텍스처 투영기의 우향 및 상향 벡터 방향으로의 이동으로 구현될 수 있으며, 회전은 투영기의 시선 벡터(Look vector) 방향을 축으로 하는 회전에 의해 직관적으로 구현된다. 또한 반사영역의 축소 및 확대는 투영 피라미드의 크기 조절을 통해서 구현된다. 반사영역의 분할(split) 및 사각화(squaring)은 간단히 투영 텍스처의 변화로 손쉽게 구현된다(그림 10 참조).

3.2.2. 실시간 응용을 위한 고려사항

텍스처 투영기를 사용한 반사효과의 만화적인 표현 방법은 기본적으로 프로그램 가능한 하드웨어 셰이더의 성능을 최대로 사용할 수 있도록 구현할 수 있는 장점을 가지고 있다. 게임과 같은 실시간 응용에 사용하기 위하여 앞서 설명한 방법의 성능을 개선할 수 있는 몇 가지 고려사항 및 구현전략을 간략히 살펴본다.

주곡률 해석

Alliez에 의하여 제시된 다면체 모델의 주곡률 해석 모델은 재메쉬화(remeshing)를 목적으로 하고 있기 때문에 실시간 렌더링 응용을 위하여 개선할 여지를 가지고 있다. 먼저 Alliez의 곡률해석

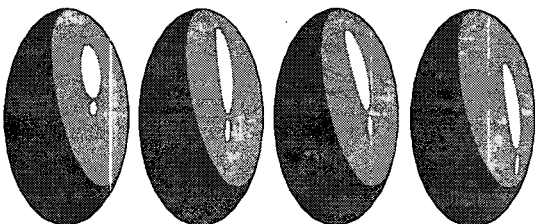


그림 10. 투영 텍스처를 이용한 반사 영역의 이동, 회전, 축소 및 확대의 예

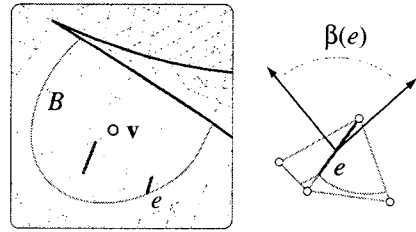


그림 11. 점점 v에서의 곡률텐서의 계산(6)

모델은 다음과 같은 식(11)으로 표현된다(그림 11 참조).

$$J(v) = \frac{1}{|B|} \sum_{edges\ e} \beta(e) |e| \bar{n} \bar{e} \bar{e}^t \tag{11}$$

여기서, B는 점점 v를 둘러싼 영역, |B|는 영역 B의 넓이, e는 점점 v를 공유하는 선분(edge), |e|는 선분 e의 길이, $\beta(e)$ 는 선분 e를 공유하는 인접한 두 면이 이루는 부호 있는 각(signed angle), \bar{e} 는 선분 e의 단위벡터이다.

이때, 곡률 텐서의 정확한 값 보다는 속도 향상을 위해서 영역 B를 점점 v를 중심으로 반지름 r인 구에 포함되는 모든 삼각형들로 근사하면 식(11)은 다음과 같이 간략하게 표현될 수 있다.

$$J(v) = \frac{1}{|B|} \sum_{edges\ e} \beta(e) |e| \bar{e} \bar{e}^t \tag{12}$$

이때, $\beta(e)$ 로서 부호 없는 각(unsigned angle)을 사용하면 반사효과의 표현에서 평균적으로 더 좋은 결과를 얻을 수 있다. 실시간 응용을 위해서는 수의 다면체로 이루어진 모델이 W 형태의 주름 모양을 갖는 경우를 가정하여 보자(그림 12 참조). 이때 Alliez의 식을 그대로 적용하면 전체 모양이 평면에 가까운 것으로 근사되어 상세성을 살리기 위한 목적을 갖는 반사 영역의 렌더링에 적합하지 않은 결과를 가지고 올 수 있다. 부호 없는 각을 사용할 경우 주름 모양이 그대로 반영되는 결과를 얻을 수 있다(그림 13 참조).

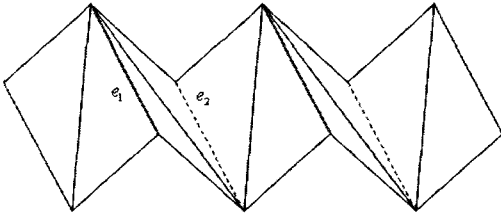


그림 12. 주름 모양을 이루는 다면체 구조



그림 13. 치마 부분의 주름이 표현된 만화적 반사효과

최대 조명점

3.2.1절에서 최대 조명점을 찾기 위하여 삼각형의 내부에 최대 조명점이 존재하는 경우를 가정하였다. 만일 삼각형의 내부에 최대 조명점이 존재하지 않는 경우에는 최대 조명점의 중심 좌표계를 이루는 값 (α, β) 가 조건 $0 \leq \alpha, \beta, \alpha + \beta \leq 1$ 을 만족하지 않게 된다. 비록 최대 조명점을 구하기 위한 행렬식(8)은 의사역행렬(pseudo inverse matrix)을 사용하여 비교적 빠르게 구할 수 있지만, 행렬식을 풀어 조건을 만족하는지 여부를 검사하기 전에는 삼각형 내부에 최대 조명점의 존재 여부를 미리 결정할 수 없고, 따라서 모든 삼각형에 대해서 매번 행렬식을 풀어야 하는 단점을 가지고 있다. 만일 임의의 삼각형 내에 최대 조명점이 존재하지 않을 조건, 혹은 존재할 가능성이 있는 조건을 구할 수 있다면, 위 행렬식의 계산 횟수를 줄여 수행성능을 개선할 수 있다.

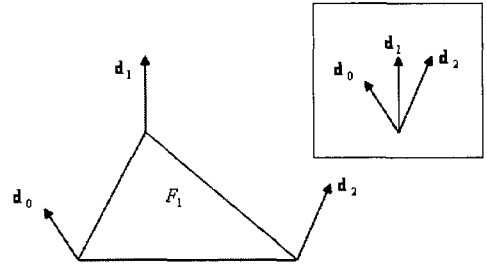


그림 14. 삼각형의 내부에 최대 조명점이 존재하지 않는 경우

삼각형 F 내부에서 N, H 가 선형적으로 변화하는 특성으로부터 다음과 조건을 만족하는 경우 삼각형 F 내부에는 최대 조명점이 존재하지 않는다.

- 삼각형 F 의 법선(face normal) 벡터와 중간 벡터의 내적이 음수인 경우.
- 삼각형 F 의 세 정점 (v_0, v_1, v_2) 에서 $d_0 \cdot d_1$ 와 $d_0 \cdot d_2$ 값이 모두 양수인 경우(그림 14 참조). 이때, $d_i = N_i - H_i, i = 0, 1, 2$.

위의 조건을 사용하는 실험적인 결과에 의하면 캐릭터 모델의 면의 개수에 비하여 1/10~1/100개 정도의 면만이 행렬식 계산을 위해서 사용되는 결과를 얻을 수 있다.

4. 결론 및 향후 연구방향

본 기고에서는 3차원 캐릭터 모델에 대한 만화적인 렌더링을 위한 기법을 간략히 살펴보고, 최근의 연구주제 중 하나인 만화적 반사효과 표현 방법을 중점적으로 살펴보았다. 중간 벡터를 수정하여 Blinn의 조명 모델을 일반화하는 기법과, 투영 텍스처를 사용하는 기법 외에도, 조명맵(light map)을 사용하는 방법 혹은 가상 광원(virtual light source)로부터 만화적인 반사효과를 얻는 다양한 방법들을 고려할 수 있을 것이다.

투영 텍스처를 이용하는 방법은 실시간 응용에 적합한 구현 가능성을 가지고 있고, 제어가 단순

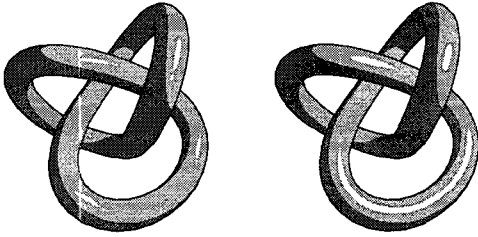


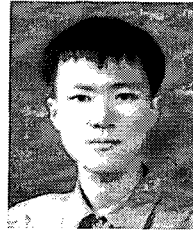
그림 15. 반사 영역이 큰 경우

한 특성이 있지만 몇 가지 단점을 동시에 가지고 있다. 먼저 텍스처 투영기를 사용하는 방법은 키 프레임 애니메이션 제작에 사용하기 어렵다는 단점을 가지고 있다. 즉, 캐릭터 모델, 광원, 시점의 움직임이 키 프레임 사이에서 선형적인 경우라도, 텍스처 투영기의 방위가 선형적이라는 보장이 없기 때문에 텍스처 투영기를 키(key)로 저장하기 위한 매개변수를 추출하기 어렵다. 또 다른 단점으로는 최대 조명점과 그 부근에서는 비교적 좋은 결과를 보이지만, 반사 영역이 일정 크기 이상이 되면 좋지 못한 결과를 보인다(그림 15 참조). 실험적인 관찰에 의하면 반사영역의 모양은 최대 조명점을 지나는 곡률선(line of curvature)을 따라서 생성되는 특징을 가지고 있다. 이는 투영 피라미드가 최대 조명점에서의 곡률만을 반영할 수밖에 없기 때문에 발생하는 결과로서, 추후 고품질 렌더링을 위하여 해결되어야 할 문제로 남아 있다.

참 고 문 헌

- [1] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3D animation. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 13-20, 2000.
- [2] JC Entertainment. FreeStyle. <http://fs.joycity.com/>, 2004.

- [3] Tobias Isenberg, Bert Freudenberg, Nick Halper, Stefan Schlechtweg, and Thomas Strothotte. A developer's guide to silhouette algorithms for polygonal models, *IEEE Computer Graphics and Applications*, pages 28-37, July/August 2003.
- [4] James F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on the Computer Graphics and interactive techniques*, pages 192-198, 1977.
- [5] Ken-Ichi Anjyo and Katsuaki Hiramitsu. Stylized highlights for cartoon rendering and animation. *IEEE Comput. Graph. Appl.*, 23(4):54-61, 2003.
- [6] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485-493, 2003.



이 환 직

- 2004년 아주대학교 미디어학부(학사)
- 2006년 아주대학교 미디어학과(석사)
- 2006-현재 아주대학교 정보통신공학과 박사 과정
- 관심분야 : 컴퓨터그래픽스, 비사실적 렌더링



최 정 주

- 1990년 한국과학기술대학 전산학과(학사)
 - 1992년 포항공과대학교 컴퓨터공학과(석사)
 - 1997년 포항공과대학교 컴퓨터공학과(박사)
 - 1997~2000년 한국전자통신연구원 선임연구원
 - 2000~2002년 (주)디지털아리아 선임연구원
 - 2002~현재 아주대학교 미디어학부 조교수
 - 관심분야 : 컴퓨터그래픽스, 비사실적 애니메이션 및 렌더링
-
-