

# 사실적인 피부 렌더링을 위해 표면하 산란 모델을 적용한 마야 셰이더 플러그인 개발

유태경<sup>\*</sup>, 이원형<sup>\*\*</sup>, 장성갑<sup>\*\*\*</sup>

## 요 약

인간의 피부를 사실적으로 렌더링하는 것은 그 동안 컴퓨터 그래픽스 분야에서 난제로 여겨져 왔으며 지금까지도 중요한 연구 과제로서 남아있다. 피부와 같은 반투명성 물질은 표면하 산란을 포함한 복잡한 광학 특성을 가진다. 본 논문에서는 사실적인 피부 렌더링을 위해 표면하 산란을 고려한 셰이더를 제안하고 3D 패키지, 마야의 플러그인 형태로 구현하였다. 구현된 피부 셰이더를 적용한 렌더링 이미지는 기존의 전통적인 셰이딩 기법을 이용한 이미지에 비해 보다 더 사실적인 피부의 표현이 가능하였고 피부의 피지, 표피, 진피 층을 각각 정반사, 다중 산란, 단일 산란으로 모델링할 수 있었다.

## Development of Maya Shader Plug in Based on Subsurface Scattering for Realistic Skin Rendering

Tae-Kyung Yoo<sup>\*</sup>, Won-Hyung Lee<sup>\*\*</sup>, Sung-Ghab Jahng<sup>\*\*\*</sup>

## ABSTRACT

In computer graphics, realistic skin rendering has been regarded as difficult tasks and remains as an important research subject. Translucent materials like skin have some complicated optical properties including subsurface scattering. In this paper, we proposes a skin shader based on subsurface scattering to render realistic skin and it has been implemented as a plug-in for Maya, 3D package. The rendered image using this proposed skin shader appears more realistic than the rendered image using classical shading techniques. Furthermore, we could model sebum, epidermis, dermis using specular reflection, multiple scattering, single scattering respectively.

**Key words:** Subsurface Scattering(표면하 산란), Local Illumination Model(국소 조명 모델), Shader(셰이더), Skin(피부)

## 1. 서 론

컴퓨터 그래픽스에서 피부를 렌더링하는 것은 여

※ 교신저자(Corresponding Author) : 유태경, 주소 : 서울시 동작구 흑석동(156-756), 전화 : 02)820-5412, FAX : 02)823-5417, E-mail : tk-you@daum.net

접수일 : 2004년 3월 10일, 완료일 : 2004년 8월 5일

<sup>\*</sup> 준회원, 중앙대학교 첨단영상대학원 영상공학과 박사과정

<sup>\*\*</sup> 중앙대학교 첨단영상대학원 교수

(E-mail : whlee@cau.ac.kr)

<sup>\*\*\*</sup> 중앙대학교 첨단영상대학원 교수

(E-mail : makefilm@cau.ac.kr)

※ 본 연구는 교육부의 BK21 사업의 지원으로 수행된 결과의 일부임.

러 가지 이유에서 매우 힘든 일이다. PDI/Dreamworks의 Jonathan Gibbs는 '슈렉'에서의 작업을 통해 피부가 렌더링하기 힘든 이유를 다음과 같이 세 가지로 설명하였다.

우선, 피부와 같은 반투명성 물체에서는 표면하 산란(subsurface scattering)이 일어난다. 또 피부는 균질하지 않은 물질로 이루어진 여러 층으로 구성되어 있다. 마지막으로 피부는 사람들이 일상에서 자주 접하는 재질 중에 하나이기 때문에 사람들에게 너무 나도 익숙하다는 것이다. 그래서 렌더링 된 피부가 조금만 시각적으로 낮설어도 그것은 관객들이 작품에 몰입하지 못하게 하는 장애로서 작용하게 된다[1].

기존에 피부 렌더링을 위해 사용하였던 전통적인 셰이더들은 이러한 피부의 시각적인 특성들을 반영하는 데에 한계를 가지고 있을 뿐만 아니라 셰이더의 매개변수들도 피부의 특성을 제어하기에는 비직관적이다.

본 논문에서는 사실적인 피부 렌더링에 적합한 셰이딩 모델을 3D 응용 프로그램인 마야의 플러그인 형태로 제작하고 물리적인 사실성보다는 시각적인 효과와 조작성의 편의성을 고려한 셰이더를 제안한다. 또한 컴퓨터 그래픽스에서 사실적인 재질을 표현하기 위해 적합한 표면 모델을 연구하는 공학적인 접근 뿐만 아니라 재질을 관찰하고 이해하여 사물에 생명력을 불어넣는 예술적인 접근도 함께 고려함으로써 문제 해결에 대한 다각적인 접근을 시도해 본다.

## 2. 관련연구

### 2.1 인간의 피부

사실적으로 재질을 표현하기 위해서는 표현하고자 하는 사물의 외형뿐만 아니라 사물의 구조와 특성, 기능, 주위 환경과의 상호 작용 등이 포괄적으로 고려되어야 한다. 그래서 텍스처 아티스트들은 재질을 표현하기 위해 앞서 예리한 관찰자가 되어야 함은 물론이고 사물을 이해하고 느낄 수 있어야 한다. 사물의 구조와 특성을 잘 이해하고 있는 것은 시각적인 관찰의 훌륭한 출발점이 될 것이다.

피부의 복잡한 구조만큼이나 피부색은 다채롭고 여러 가지 요인에 의해서 결정된다. 피부의 경우에도 피부의 해부학적 구조와 생리학적 특성을 잘 이해하는 것은 피부가 다양한 환경과 사람의 심리적인 상태에 대해 시각적으로 어떻게 반응할 것인가를 이해하는 데에 중요한 역할을 한다.

피부는 외부에서부터 표피(epidermis), 진피(dermis) 및 피하지방층(subcutaneous fat)의 독특한 세 개 층으로 구성되어 있다[2]. 각 층들의 특성들이 반영되어 전체적인 피부의 광학 특성이 결정된다.

피부의 표면은 광택이 없는 난반사 표면으로 근사할 수 있다. 왜냐하면 피부의 최상위 표면이 일반적인 반사가 없는 죽은 세포들로 덮여 있기 때문이다. 이들 세포들은 광학적으로 비활성이기 때문에 우리가 보는 피부의 광택은 땀, 혹은 스킨 오일이나 피부 표면을 덮고 있는 어떠한 미용 제품에 의한 것이라

볼 수 있다.

광택을 제외한 피부색의 양상은 피부의 세 가지 착색 요인에 의한 광학 필터링 정도에 영향을 받는다. 세 가지 착색 요인은 표피의 멜라닌, 진피와 피하지방층의 카로틴, 그리고 진피에 분포한 모세혈관의 헤모글로빈이다. 멜라닌은 갈색 색소이고 카로틴은 오렌지 색조를 띤다. 헤모글로빈은 산소와 결합하여 산화 헤모글로빈이 되었을 때는 선명한 적색이나 분홍색을, 산소와 분리되면 암적색이나 청색을 띤다.

피부의 스펙트럼은 피부와 빛의 상호작용에 의해 형성된다. 피부에 닿은 빛은 층들을 통해 투과, 흡수, 반사된다. 일반적으로 인간의 피부에 대한 스펙트럼은 멜라닌과 헤모글로빈에서 일어나는 흡수로 인한 특성으로 연속적이고 유사한 형태를 가진다[3]. 또한 단파장(청색, 녹색)에 비해서 장파장(주황색, 적색)에서 상대적으로 더 높은 반사 특성을 가진다. 피부에는 빛과 반응하여 산란과 착색을 하는 착색 입자들이 분포하기 때문에 광학적으로 비균질한 물질이다.

### 2.2 피부 렌더링

전통적인 반사 모델들은 일반적으로 람베르트 코사인 항(Lambert cosine term)과 엠비언트 항(ambient term)의 조합을 사용하여 물질의 표면하 산란 효과를 모델링하고 있다. 이러한 모델들을 가지고 사실적인 피부를 렌더링하는 것은 전적으로 애니메이션의 몫으로 남겨졌었다[4]. 그러나 최근에 들어 여러 가지 응용 분야에서의 피부 렌더링의 필요성과 렌더링 기술의 발전을 바탕으로 표면하 산란을 적절하게 표현할 수 있는 기법들이 다양하게 제시되고 있다.

Hanrahan과 Krueger는 일차원 선형 수송 이론(one-dimensional linear transport theory)을 이용하여 층상 표면에서의 표면하 산란에 대한 모델을 제안하였다[5]. 그들은 물질 내부에서의 광자 추적을 이용해 표면하 산란을 시뮬레이션(simulation)할 수 있는 방법을 제안하였지만 최종 모델에서 단일 산란만을 고려한 BRDF(Bidirectional Reflectance Distribution Function)[6]를 사용하였다. BRDF는 한 점에서 일어나는 산란만을 고려할 뿐, 표면의 다른 지점들 사이에서 일어나는 물질 내부에서의 빛의 이동을 시뮬레이션할 수 없다. 이것을 고려하려면 물질을 입자 성분으로서 취급해야 하는데 그러한 방법을 통

해서 Dorsey는 풍화된 암석에서의 표면하 산란을 광자 맵핑 방법으로 시뮬레이션하였다. Pharr와 Hanrahan은 산란 방정식(scattering equation) 개념을 소개하고 이 개념이 전통적인 몬테카를로 광선 추적 기법과 비교해 얼마나 효율적으로 사용될 수 있는가를 보였다[7].

최근 들어, 반투명 물질에서의 빛의 산란을 확산 과정으로서 모델링하는 방법이 제안되었다. 확산 근사(diffusion approximation)는 특히 전통적인 몬테카를로 방법을 사용할 때에는 계산량이 많았던 산란이 많이 일어나는 물질에 대해 효과적으로 사용될 수 있다. Jensen은 그의 논문에서 해석적인 쌍극자 확산 근사(dipole diffusion approximation) 모델을 제안하고 이 모델이 반투명 물질에서 산란된 빛의 측정치와 잘 부합됨을 보였다[8]. 그는 BSSRDF(Bidirectional Surface Scattering Reflectance Distribution Function)를 공식화하기 위해 이러한 근사 방법을 이용하였다.

Jensen이 제안한 BSSRDF 근사는 몬테카를로 광자 추적보다 계산이 빠르다. 하지만 표면의 입사 광속 분포(incident flux distribution)에 대한 샘플링(sampling)이 필요하기 때문에 여전히 전통적인 BRDF에 비해 계산 시간이 많이 소요된다. 그는 이후 논문[9]에서 반투명한 물질에 대한 빠르고 일반적인 이중 렌더링 기법(two-pass rendering technique)을 제안하였다.

피부는 광자가 들어와서 튀어 나가기까지 여러 차례 산란을 하는 다층적인 매질(medium)이다. 어떤 과정들은 두께에 따라 다르게 감쇠한다. 광자의 진로는 복잡하고 광자의 에너지뿐만 아니라 각 층의 성질의 영향을 받는다. 그러나 한 가지 확실한 것은 광자들이 들어온 위치를 통해 다시 나가는 일은 드물다. 그래서 BSSRDF가 필요한 것이다.

Jensen의 논문에서 제시된 이중 렌더링 기법은 두 부분으로 구성된다. 경로 추적을 통해서 빛이 피부 내부에서 오직 한번만 반사되는 경우에 대한 근사적인 해를 구할 수 있는 단일 산란과 통계적인 쌍극자 조명 확산 근사를 통한 다중 산란으로 이루어진다. 이 모델의 장점은 간단한 광선 추적기(ray-tracer)에서도 구현할 수 있고 몇 가지 절차를 거치면 z 버퍼(z-buffer) 렌더러와 함께 사용할 수도 있다는 것이다[10].

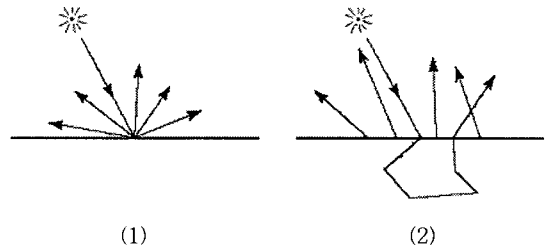


그림 1. (1) BRDF와 (2) BSSRDF에서의 빛의 산란

### 2.3 마야 셰이더 개발

대부분의 3D 패키지가 궁극적으로 수행하는 일들은 동일하지만 그 접근 방식에 있어서는 많은 차이들을 보인다. 특정한 작업을 완성하기 위해서 수행하여야 하는 과정은 주어진 패키지에 따라 정해지는 것이 보통이지만 마야의 경우 유연한 작업 흐름(workflow)을 가지고 있기 때문에 그러한 제약이 많지 않다. 대부분의 경우 특정한 작업에 접근하는 방식이 하나 이상 존재하기 때문에 마야 사용자들은 자신에게 적합한 것을 선택하면 된다. 그러기 위해서는 마야의 구조적 특성을 알아야 한다.

전체 마야 시스템을 주요한 구성 요소들로 나누어 보면 그림 2와 같다.

우리가 마야의 GUI와 상호작용을 할 때 실제로는 마야에 MEL 명령어들을 전달하는 것이 된다. 이러한 명령어들은 명령어 엔진으로 보내어져서 해석되고 실행되게 된다. 배치 모드인 경우 GUI를 통하지 않고 직접 전달하는 것 이외에는 동일한 방식이다.

대다수의 MEL 명령어들은 디펜던시 그래프(dependency graph, DG)의 조작을 위한 것이다. DG는 직관적으로 전체 scene이라고 생각할 수 있다. DG는 scene 내부의 모든 데이터를 포함하고 있으며 DG의 구조와 레이아웃은 데이터들이 어떤 식으로 처리될 것인지를 나타낸다. DG는 그야말로 마야의 두뇌이

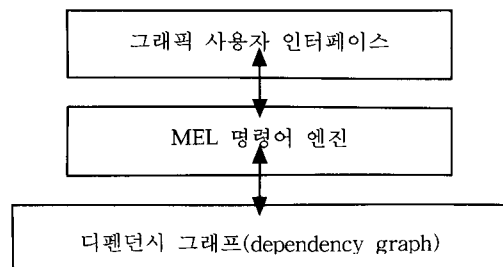


그림 2. 마야 시스템

자 심장이라고 말할 수 있다.

3D 응용 프로그램은 모델링, 애니메이션, 라이팅이나 렌더링과 같은 다양한 작업을 수행하는 수단을 제공한다. 이러한 개개의 작업들에서 생성되는 데이터와 정보는 차이가 있기 때문에 기존의 응용 프로그램에서는 이러한 요소들을 개별적으로 구현하였다. 이러한 구조에서는 모듈들 사이에서 데이터와 정보를 주고받는 것이 종속적이기 때문에 만일 하나의 모듈에 변화가 생기면 나머지 다른 모듈들도 그 변화에 맞추어 수정을 하여야 한다. 이러한 이유 때문에 더욱더 일반적이고 유연성 있는 구조가 필요하게 되었다.

3D 응용 프로그램들이 실질적으로 수행하고자 하는 것은 간단하게 말하자면 어떠한 데이터를 생성하고 그것을 일련의 과정들을 통해서 처리하는 것이다. 데이터 생성과 처리 과정의 결과는 폴리곤 메쉬나 이미지의 픽셀들이 될 수 있다. 이러한 전체 과정은 일련의 처리 과정의 한 쪽 끝으로 들어간 데이터가 처리를 거친 후 다른 쪽 끝으로 나오는 데이터 흐름 모델(data flow model)로 생각할 수 있다. 여기에서 개별적인 모듈들을 단순한 기능을 가진 작은 단위까지 분할함으로써 나뉜 모듈들은 더욱 일반화 된 연산자(operator)로서의 역할을 한다. 결과적으로 연산자들의 연결을 파이프라인이라고 생각할 때, 원하는 작업이면 무엇이든 정확하게 구성된 파이프라인으로 데이터를 흘려보내어 이루어질 수 있다는 것이다.

Alias는 위에서 언급한 데이터 흐름 패러다임을 사용하여 마야의 핵심부인 디펜던시 그래프를 구현하였다. DG에서는 데이터와 연산이 노드(node)들로 캡슐화되어 있다. 노드들이 특정한 단일 작업을 수행하도록 디자인함으로써 노드들은 간단하고 더 처리하기 쉬운 상태가 될 수 있다. DG의 유연성은 마야의 모든 데이터들(모델링, 다이내믹스, 애니메이션, 셰이딩 등)이 이러한 노드들을 통해서 생성되고 처리된다는 데에 있다. 마야에 새로운 기능을 추가하자면 단순히 새로운 노드 하나를 생성하면 된다.

일반적인 3D 패키지에서는 scene을 구성하는 모든 데이터들을 중앙의 데이터베이스에 저장한다. 반면 마야의 DG는 상호연결된 노드들의 집합이기 때문에 각 노드들은 내부적으로 자신의 데이터를 저장하고 있다. 그래서 일반적인 쿼리(query) 타입을 사용하기는 어렵다. 하지만 마야의 프로그래밍 인터페이스는 내부적인 구조를 정확히 모르더라도 데이터

에 접근할 수 있는 함수를 제공함으로써 중앙 데이터베이스에 저장된 것과 같이 사용할 수 있게 한다.

그림 3은 일반적인 노드를 간단하게 표현한 그림이다.

노드의 주된 역할은 데이터를 저장하고 그것을 특정 목적에 맞추어 변형하는 것이다. 그림 3에서처럼 내부적으로 데이터는 크게 input과 output으로 나누어지지만 데이터에 따라 절대적인 분류 기준이 있는 것은 아니다. 데이터의 용도에 따라 input attribute가 되기도 하고 output attribute가 되기도 한다.

노드의 구성에서 무엇보다도 핵심적인 구실을 하는 것은 compute 함수이다. compute 함수는 입력 데이터를 특정 목적에 맞추어 변형하여 출력 데이터를 만드는 역할을 한다. 노드의 구성을 프로그래밍 함수로서 표현하자면 다음과 같다.

$$\text{output} = \text{compute}(\text{input0}, \dots, \text{inputN})$$

정리하면, 마야는 DG라는 구조를 사용하여 scene을 구성하는데 DG의 기본 단위는 노드이고 노드의 핵심은 compute 함수이다. 결국 마야에서 이루어지는 일들은 모든 compute 함수의 조합이 만들어내는 결과라고 볼 수 있다.

제안하는 피부 셰이더 플러그인 역시 마야의 노드가 된다. 마야 셰이더 플러그인을 제작하는 것은 오브젝트(object)를 셰이딩 하기 위해서 입력 받아야 하는 데이터와 셰이딩 결과(output color, output transparency)의 형태를 결정하고 거기에 맞는 compute 함수를 작성하는 것이다.

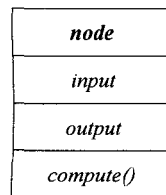


그림 3. 일반적인 노드

### 3. 제안하는 피부 셰이더

#### 3.1 단일 산란 (Single Scattering)

그림 4에서 보는 것처럼 단일 산란은 굴절된 입사 광선과 나가는 광선(outgoing ray)이 교차할 때, 즉 빛이 피부 내부에서 한번만 방향 전환을 할 경우만을

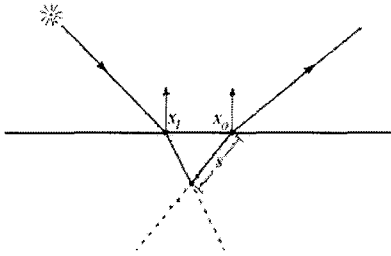


그림 4. 단일 산란 (8)

고려하는 1차 근사이다. 단일 산란을 해석적으로 계산하기 위한 몇 가지 방법이 있다. 제안된 피부 셰이더에서는 Matt Pharr의 피부 모델을 바탕으로 단일 산란을 구현하였다[11].

우선, 층(layer) 내부에서의 산란을 모델링할 때, 빛이 내부 입자와 상호 작용한 결과를 묘사하기 위해서 위상함수를 사용할 수 있다. 위상함수는 광선이 층 내부의 입자와 충돌한 후에 일어나는 빛의 산란 분포를 설명한다. 여기에서는 일반적으로 사용되는 매개변수화 위상함수인 Henyey-Greenstein 위상함수를 사용하였다.

Henyey-Greenstein 위상함수는 다음과 같다.

$$p(\cos\theta) = \frac{1-g^2}{(1+g^2-2g\cos\theta)^{3/2}}$$

여기에서 비대칭성 매개변수(asymmetry parameter)  $g$ 는  $-1$ 에서  $1$  사이의 범위를 가지는데, 그림 5에 보이는 것처럼 후방산란(backscattering)과 전방산란(forward scattering) 사이의 범위를 나타낸다.

Henyey-Greenstein 위상함수는 아래의 소스 코드와 같이 구현되었다.

```
float skin::phase(MFloatVector v1, MFloatVector v2, float g)
{
    float costheta = -v1 * v2;
    return float((1.0f - g * g) / pow(1.0f - g * g - 2.0f * g * costheta, 1.5f));
}
```

우리가 만일 각 입사 광선에 대해 물질의 입자들과의 단일 산란만을 고려한다면 산란을 설명하는 닫힌 형태의 수식을 유도할 수 있다. 이러한 근사는 층이 상대적으로 얇거나 산란에 비해 감쇄가 더 많이 일어나는 경우에 적당하다. 다음은 제안된 셰이더에

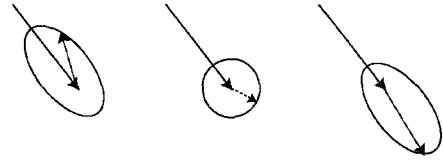


그림 5. 비대칭성 매개변수  $g$ 의 음수값은 후방산란 (왼쪽), 영은 등방성산란 (가운데), 양수값은 전방산란 (오른쪽)에 각각 대응한다.

서 사용한 단일 산란을 설명하는 BRDF이다.

$$f_r(w_i \rightarrow w_o) = \frac{a p(w_i, w_o) e^{-d(\frac{1}{w_i \cdot N} + \frac{1}{w_o \cdot N})}}{((w_i \cdot N) + (w_o \cdot N))}$$

여기에서  $a$ 는 산란 알베도(scattering albedo),  $\sigma_i/\sigma_o$ 이고  $d$ 는 층의 두께이다.

다음은 주어진 입사 방향  $w_i$ , 나가는 방향  $w_o$ , 표면 노멀(surface normal)  $N$ , 비대칭 값  $g$ , 산란 알베도 albedo, 체적의 두께 thickness으로 층 내부의 산란을 근사하기 위해 단일 산란 식을 사용하는 소스 코드이다.

```
float skin::singleScattering(MFloatVector wi, MFloatVector wo,
    MFloatVector n, float g, float albedo, float thickness)
{
    float win = fabsf(wi * n);
    float won = fabsf(wo * n);
    return float(albedo * phase(wo, wi, g) / (win + won)
        * (1.0f - exp(-(1.0f / win + 1.0f / won) * thickness)));
}
```

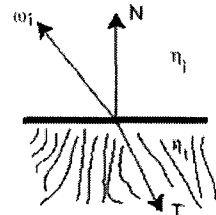


그림 6. 굴절률이 각각  $n_1, n_2$ 인 물질 사이의 경계에서 프레넬 반사와 투과를 계산하기 위한 설정

피부의 경우에서의 공기와 표피처럼 굴절률이 다른 물질이 맞닿아 있는 경계에서 굴절률과 빛의 입사각에 따라 입사광의 반사, 투과가 결정되는 프레넬 반사가 고려되어야 한다. 프레넬 식은 이러한 효과를 나타낸다. 부도체 사이의 경계에서 프레넬 반사는 다음과 같다.

$$F_r = \frac{1}{2}(r_1^2 + r_2^2)$$

여기에서,

$$r_1 = \frac{\eta_i(N \cdot w_i) + \eta_i(N \cdot T)}{\eta_i(N \cdot w_i) - \eta_i(N \cdot T)}$$

$$r_2 = \frac{\eta_t(N \cdot w_i) + \eta_t(N \cdot T)}{\eta_t(N \cdot w_i) - \eta_t(N \cdot T)}$$

정반사 투과 방향(specular transmission direction) T는 스넬의 법칙에 의해 구할 수 있다. 프레넬 투과  $F_t$ 는  $F_t = 1 - F_r$ 을 사용하여 계산하였다. 프레넬 함수는 다음과 같이 구현되었다.

```
void skin::fresnel(MFloatVector I, MFloatVector N, float eta, float& Kr,
                 float& Kt, MFloatVector& R, MFloatVector& T)
{
    R = reflect(I, N);
    T = refract(I, N, eta);
    float rParallel, rPerpendicular;
    rParallel = ((N * I) - eta * (N * T)) / ((N * I) - eta * (N * T));
    rPerpendicular = (eta * (N * I) + (N * T)) / (eta * (N * I) - (N * T));
    Kr = ((rParallel * rParallel) - (rPerpendicular * rPerpendicular)) / 2.0f;
    Kt = 1.0f - Kr;
}
```

피부는 공기와는 다른 굴절률(피부의 굴절률은 약 1.4)을 가지고 있기 때문에 피부 표면에서는 프레넬 효과가 고려되어야 한다. 여기에서는 Matt Pharr가 프레넬 효과를 적용하는 과정을 간단하게 하기 위해 사용한 efresnel 함수를 사용하였다. efresnel 함수는 다음과 같은 점에서 fresnel 함수와 차이가 있다.

- efresnel 함수는 투과 방향 T를 직접 리턴한다. 이것은 전체 코드를 좀 더 명확하게 한다.
- fresnel 함수에 의해 구해진 Kt 값을 오버라이딩 한다. 이는 제안된 셰이더에서는 의미가 없지만 특정 렌더러나 API에서 제공하는 fresnel 함수를 사용할 경우에서 Kt값이 틀리게 리턴될 때에도 의도한 Kt 값을 갖게 한다.
- smoothstep 함수를 사용하여 프레넬 반사 계수의 값을 증가시킨다. 이것은 물리적으로는 올바르게는 아니지만 결과 이미지에서는 모서리의 프레넬 효과가 강조되게 함으로써 시각적으로는 만족스럽다.

```
MFloatVector skin::efresnel(MFloatVector I, MFloatVector N, float eta,
                          float& Kr, float& Kt)
{
    MFloatVector R, T;
    fresnel(I, N, eta, Kr, Kt, R, T);
    Kr = smoothstep(0.0f, 0.5f, Kr);
    Kt = 1.0f - Kr;
    T.normalize();
    return T;
}
```

단일 산란 근사를 위해서 g값으로 Hanrahan과 Krueger의 논문에서 제시하였던 피부의 단일 산란 추정값인 0.8을 사용하였다. 단일 산란의 전체적인 강도를 조절하기 위한 변수로 singleScatteringCoeff를 사용하였다.

### 3.2 다중 산란 (multiple scattering)

Jensen은 다중 산란을 확산 산란(diffusion scattering) 근사로서 설명하였다. Jensen이 확산 산란 근사에 사용한 쌍극 점 조명 모델은 그림 7에서 보는 것처럼 표면 아래, 위 두 개의 점 조명원으로 구성된다. 쌍극 점 조명원은 피부에서의 빛 감쇠를 시뮬레이션하기 위한 가상 구조로서 의료 과학자들에 의해 처음 사용되었다. 각 샘플에서 합산한 값이 확산 반사(diffuse reflectance)가 된다.

인간의 피부를 렌더링할 경우, 확산 산란 항이 BSSRDF의 결과값에 지배적인 영향을 미친다. 값을 조절하고 블러(blur)를 주는 등의 과정을 통해 단일 산란에 의해서도 좋은 결과를 얻어 낼 수 있다. 하지만 단일 산란은 방향성을 띠기 때문에 조명의 전이(transition)가 부드럽지 못하다. 대리석과 같이 더욱 등방성 산란 성질을 많이 가지는 물질의 경우에는 이러한 효과가 더욱 치명적이다.

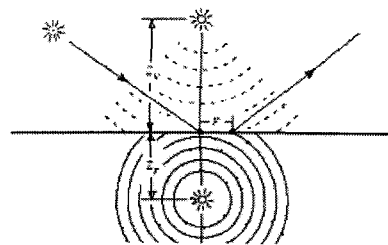


그림 7. 하나의 입사 광선은 확산 근사를 위해 쌍극 조명원으로 변환될 수 있다(8).

확산 산란은 피부의 반투명성을 표현하기 위해 고려되어야 한다. BSSRDF의 계산은 상당히 많은 시간이 걸리기 때문에 여러 가지 가속 기법들이 연구되고 있다. 근래 프로덕션의 애니메이션 제작에도 사실적인 피부 렌더링을 위해 Point cloud cache를 이용한 기법과 같은 가속 기법들이 사용되고 있다[10].

이러한 가속 기법들은 실질적으로 연구 개발 인력이 충분하지 않은 프로덕션이나 개인 작업자들이 사용하기에는 여러 가지 어려움이 따른다. 선처리(pre-computed) 된 산란 맵(scattering map)을 작성하여 계산 과정에서 읽어들이는 것과 같은 부가적인 처리가 필요하며 이러한 과정으로 인해 결과의 제어에 있어서도 비교적 직관성이 저해되는 것이 사실이다.

본 논문에서는 확산 산란 모델을 근거로 하여 피부의 표피에 의한 확산 산란을 엠비언트 향과 표면 법선벡터를 이용하여 페이킹함으로써 단일 산란에서 표현하지 못하는 반투명성을 표현하고자 하였다. 확산 산란을 페이킹하기 위한 조건으로 다음과 같은 사항이 고려되었다.

- 추가적인 조명이 없이도 제어가 가능하도록 설계
- 멀티 레이어 렌더링을 위해 다른 반사 성분들과 독립적으로 제어와 렌더링이 가능하도록 설계

그림 8은 Stam의 피부 셰이더로 표피에 의한 확산 산란 성분과 진피에서의 방향성 산란을 조합하여 피부 렌더링을 한 예이다. 왼쪽 이미지는 확산 산란 성분에서 프레넬 효과가 고려되었고 가운데 이미지에서는 단일 산란 성분이 표현되었다.

누구나 피부 표면에서 거울과 같은 반사가 일어나지 않을 것이라는 것을 알 수 있다. 하지만 피부와 같은 반투명성 유기체에서 표면하 산란과 같이 물리광학적 현상이 어떠한 메커니즘으로 일어나는가를

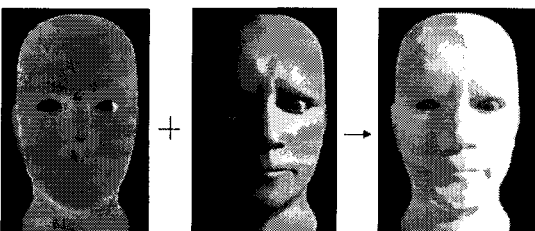


그림 8. 엠비언트 조명(왼쪽)과 스폿 조명(가운데)에 의한 렌더링 결과를 합성한 렌더링 이미지(오른쪽)(4)

정확히 이해하기란 쉬운 일이 아니다. 본 논문에서는 피부 셰이더를 구현하기 위하여 피부의 표면하 산란과 관련한 몇 가지 관찰을 통해 피부의 특성을 이해하고자 하였다.

실제 피부와 나뭇잎과 같은 유기체가 빛과의 상호작용에서 어떠한 형태로 보여 지는지를 알아보기 위해 레이저 포인터를 사용하여 각 물질에 빛을 투과시켜 관찰하였다. 앞서 언급한 Jensen의 연구도 레이저 포인터를 사용하여 우유가 든 유리잔에 광선을 투과하는 실험으로 시작되었다.

나뭇잎은 반투명성의 좋은 예다. 그림 9와 같이 비교적 얇은 나뭇잎에 카메라의 반대편에서 레이저 포인터로 광선을 투과시켜 나뭇잎의 섬유질 층들이 빛을 어떻게 산란시키는 지를 볼 수 있었다.

그림 9에서 중앙의 가장 밝은 부분이 단일 산란으로 인한 성분이고 그 주위로 부드럽게 번지는 빛이 다중 산란으로 인한 성분이다. 그림 10은 좀 더 두꺼운 나뭇잎에 레이저 포인터로 광선을 투과 시키고

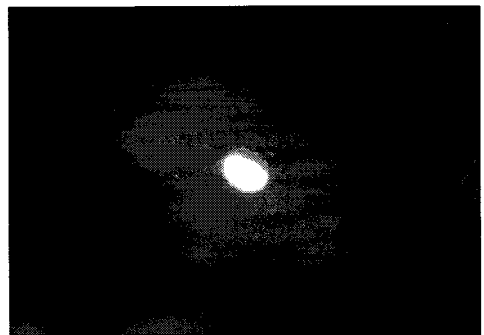


그림 9. 얇은 나뭇잎에 레이저 포인터로 카메라 반대 방향에서 빛을 투과하였을 경우

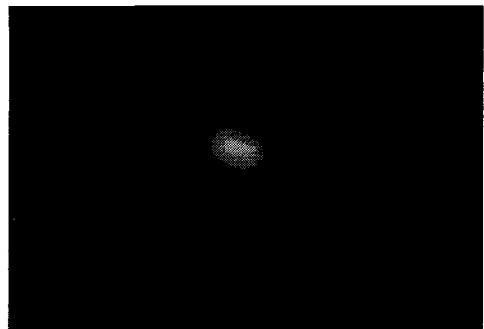


그림 10. 두꺼운 나뭇잎에 레이저 포인터로 카메라 반대 방향에서 빛을 투과하였을 경우

동일한 노출로 촬영한 것이다. 이 경우에는 그림 9에서 보았던 것과 같은 부드럽게 번지는 빛이 거의 없다. 이것은 단일 산란이 많이 일어나고 다중 산란이 적게 일어나는 물질의 예라고 할 수 있다. 또한 표면하 산란이 층의 두께와 관계가 있다는 것을 보여주는 예다.

피부는 앞서 말한 것과 같이 성분이 다른 여러 가지 층으로 되어 있기 때문에 좀더 복잡한 양상을 가진다. 그림 11, 그림 12는 컷볼에 레이저 포인터로 광선을 투과하여 촬영한 사진이다. 컷볼은 나뭇잎에 비해 두께가 두껍고 표면하 산란이 많이 일어나기 때문에 단일 산란의 결과를 관찰하기 힘들다. 우리가 여기에서 주의 깊게 보아야 할 것은 방향성이 강한 레이저 포인터의 광선인데도 불구하고 다중 산란은 등방성에 가깝게 일어난다는 것이다.

또 그림 13에서 볼 수 있는 것처럼 피부의 경우 그림자가 생긴 영역에도 산란으로 인한 붉은 색조가 생긴다. 그리고 그림자 영역과 빛을 받는 영역 사이의 경계가 부드러운 것을 볼 수 있다. 또 얼굴 표면에

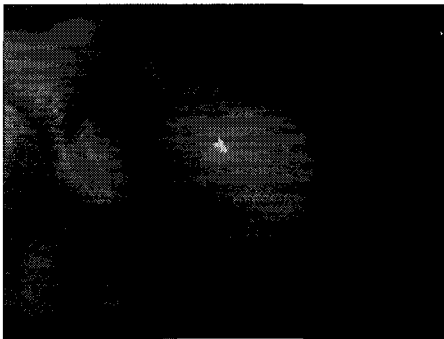


그림 11. 컷볼에 레이저 포인터로 빛을 투과하였을 경우

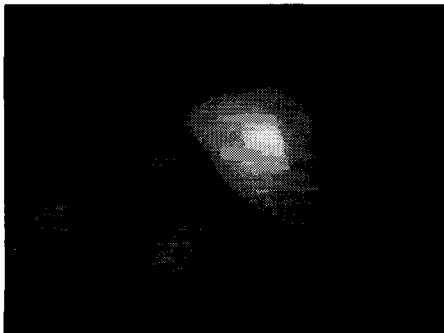


그림 12. 컷볼에 레이저 포인터로 빛을 투과하였을 경우 (노출을 증가시킴)

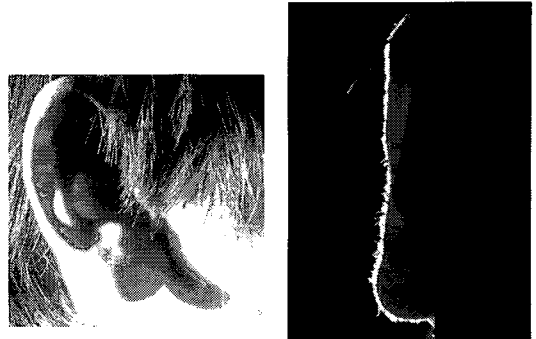


그림 13. 얼굴 뒤에서 조명을 비출 경우

서 일어나는 프레넬 반사는 얼굴 표면에 존재하는 모발에 의해 더 강조되어 보인다는 것을 알 수 있다.

제안하는 피부 셰이더의 다중 산란은 관찰한 결과를 바탕으로 엠비언트 성분으로 구현하였다.

그림 12에서 보았던 것과 같이 거의 등방성에 가까운 다중 산란과 그림 13에서처럼 뚜렷하게 나타나는 프레넬 효과를 재현하기 위해서 단순히 시선 벡터  $I$ 와 표면 법선 벡터  $N$ 의 내적에 클램핑 계수를 더하여 이것으로 확산 칼라 값을 나누어 주었다. 여기에 다중 산란의 강도를 조절할 수 있는 다중 산란 계수가 곱해진다.

$$C_{ms} = \frac{C_d \times I_{ms}}{|I \cdot N| + l_c}$$

여기에서,  $C_{ms}$ 는 다중 산란 칼라,  $C_d$ 는 확산 칼라 입력값,  $I_{ms}$ 는 다중 산란 계수,  $l_c$ 는 클램핑 계수를 각각 나타낸다.

다중 산란을 구현한 소스 코드는 다음과 같다.

```
multipleScatteringSkinColor[0] += diffusionColor[0] *
    multipleScatteringCoeff / (fabsf(cosin) + clampCoeff);
multipleScatteringSkinColor[1] += diffusionColor[1] *
    multipleScatteringCoeff / (fabsf(cosin) + clampCoeff);
multipleScatteringSkinColor[2] += diffusionColor[2] *
    multipleScatteringCoeff / (fabsf(cosin) + clampCoeff);
```

### 3.3 셰이딩 매개변수 (shading parameter)

셰이더의 셰이딩 결과를 제어하기 위한 셰이딩 매개변수는 다음과 같다.

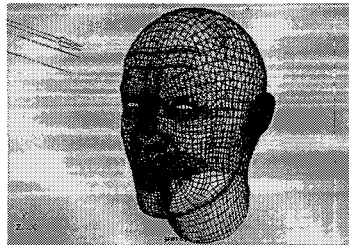
- Skin color : 단일 산란의 색상을 결정한다.
- Sheen color : 단일 산란에서의 광택 성분의 색상을 결정한다.



- Opacity : 전체 셰이더의 투명도를 결정한다.
- AmbientCoeff : 엠비언트 조명에 의한 영향을 결정한다.
- Thickness : 단일 산란이 일어나는 층의 표면 두께를 결정한다.
- Eta : 표면 외부와 표면과의 굴절률 차이를 나타낸다. 피부의 경우 공기와의 굴절률 비가 1.4이다.
- SingleScatteringCoeff: 단일 산란의 전체 강도를 결정한다.
- DiffusionColor : 다중 산란의 색상을 결정한다.
- MultipleScatteringCoeff : 다중 산란의 정도를 결정한다.
- ClampCoeff: 다중 산란 감쇠의 경계를 결정한다.
- Power : 표면의 정반사의 power 값을 결정한다.
- Specularity : 표면 정반사의 정도를 결정한다.

## 4. 구현 결과 및 평가

### 4.1 렌더링 이미지



wire-frame model (face)

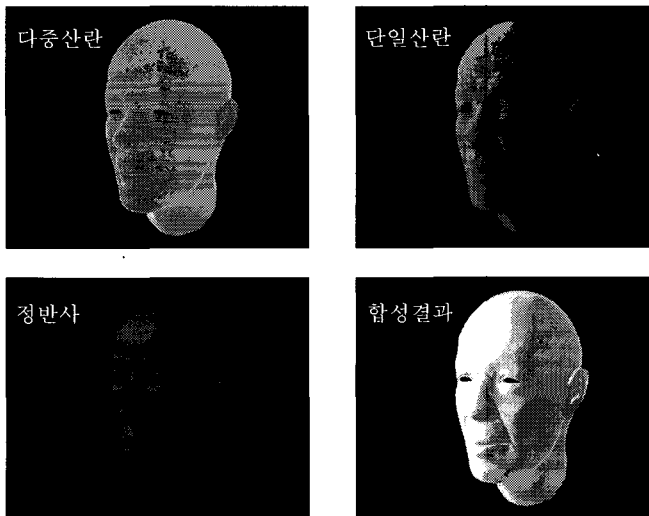


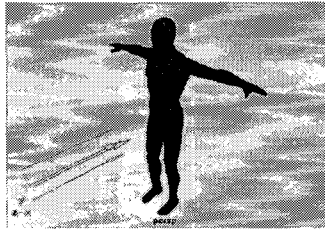
그림 14. 제안하는 피부 셰이더를 적용시킨 이미지 - 1

### 4.2 셰이딩 인터페이스

제안하는 피부 셰이더를 사용하기 위해서는 마야의 플러그인 매니저에서 skin.mll 파일을 로딩하거나 미리 skin.mll 파일을 마야의 플러그인 경로에 넣어 두어야 한다. 일단 플러그인이 로딩되면 하이퍼 셰이더의 create materials 메뉴에서 skin 노드를 찾을 수 있다.

기본적인 사용 방식은 마야의 다른 셰이더들과 동일하지만 기본적인 엠비언트 성분 이외에 다중 산란 성분이 엠비언트 성분처럼 사용된다는 차이가 있다. 그림 18의 셰이더 Attribute editor 창에서 볼 수 있는 것처럼 셰이더를 제어하는 각 매개변수들을 피지, 표피, 진피에 대응시켜 3개의 그룹으로 나누었다.

관련 연구에서 살펴보았던 것처럼 우리가 볼 수 있는 피부의 광택은 피지 성분, 땀, 그리고 스킨이나 로션과 같은 화학제품에 의한 것이다. 이러한 광택의 원인이 되는 성분을 피지 층으로 모델링하였다.



wire-frame model (male body)

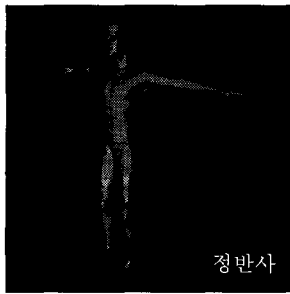
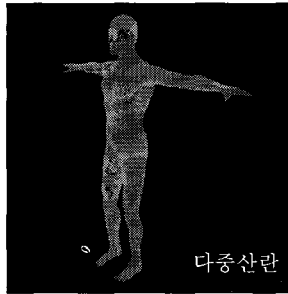


그림 15. 제안하는 피부 셰이더를 적용시킨 이미지 - 2



wire-frame model (female body)

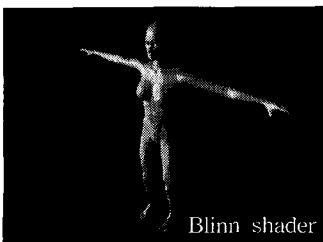
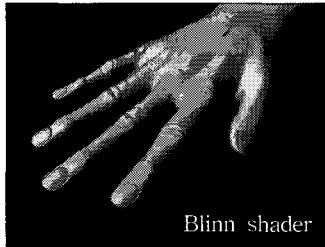


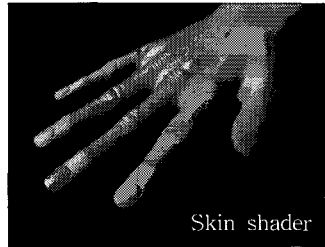
그림 16. 제안하는 피부 셰이더를 적용시킨 이미지 - 3



wire-frame model (hand)



Blinn shader



Skin shader

그림 17. 제안하는 피부 셰이더를 적용시킨 이미지 - 4

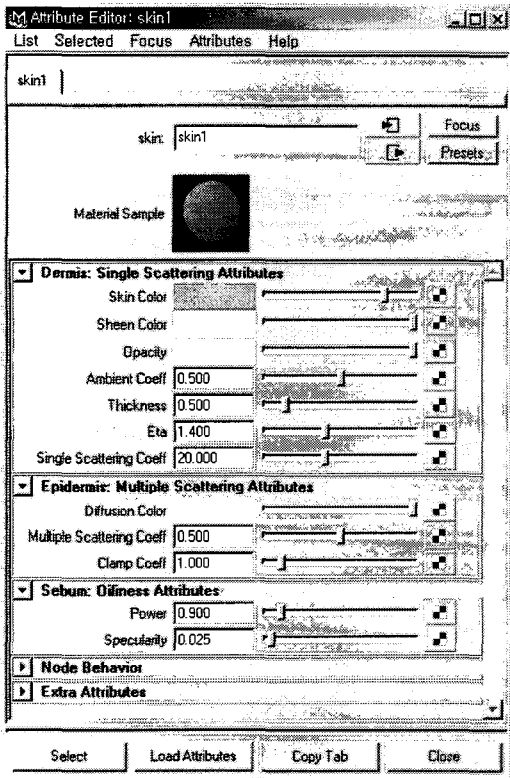


그림 18. 제안하는 피부 셰이더의 Attribute Editor

Phong 모델의 specular 성분을 사용하여 광택을 표현하였다.

표피와 진피는 각각 그 반사 특성을 고려하여 다

중 산란과 단일 산란으로 모델링하여 각각 멜라닌의 갈색 색조와 헤모글로빈의 붉은 색조를 표현 할 수 있도록 하였다.

- 피지(selbum): specular component
- 표피(epidermis): multiple scattering component
- 진피(dermis): single scattering component

그림 19. 제안하는 피부 셰이더의 인터페이스 구조

### 4.3 구현 결과 평가

4.1절에서 보는 것과 같이 제안하는 피부 셰이더를 적용하여 렌더링한 결과에서는 기존의 셰이더에서 엠비언트 성분을 이용하여 표현하여야 하였던 다중 산란 성분을 추가적인 라이트의 생성 없이 표현할 수 있었다. 또 서로 다른 칼라를 가진 표피와 진피층을 합성함으로써 그림 20에서 보는 것과 같이 실제 피부에서 헤모글로빈과 멜라닌 성분이 표현되는 것과 같은 색조 변화를 표현할 수 있었다.

표 1에서 볼 수 있는 것처럼 기존의 Blinn이나 Lambert 같은 일반적인 셰이더와 비교했을 때 렌더링 시간이 약간 더 소요되었다. 또한 Stam이 제안한 마야의 피부 셰이더와는 렌더링 시간에서는 거의 차이가 없었다. 물론 전역 조명 효과를 반영하는 렌더링 시간과는 직접적인 비교가 무의미하다.

구현된 셰이더에서는 추가적인 라이트의 생성 없



그림 20. 렌더링 결과 비교

표 1. 렌더링 속도 비교

Model Name	No. of triangle	Rendering Time		
		제안하는 피부 셰이더	Blinn, Lambert	Stam의 Skin Shader
Face	17244	00:00:02	00:00:01	00:00:03
Male body	103381	00:00:07	00:00:06	00:00:07
Female body	120518	00:00:07	00:00:06	00:00:07

이 다중 산란 성분을 엠비언트 성분과 같은 원리로 조절함으로써 간편한 조절이 가능하다는 이점도 있지만 기존의 라이팅 환경과 통합되는데 어려움이 따른다.

피지의 광택을 표현하기 위해 Phong 모델의 정반사 값을 사용하였는데 power 값과 specularity를 적절히 조절하면 피부의 광택 조절이 가능했다. 그러나 반투명성을 표현하기 위해 단일 산란의 수치를 크게 사용했기 때문에 전체적인 밝기가 밝아진다는 단점이 있다.

왼쪽부터 제안하는 피부, Lambert, Blinn, Stam의 피부 셰이더 순으로 적용시킨 결과이다.

### 5. 결론 및 향후 연구

컴퓨터 그래픽스 기술의 발전으로 컴퓨터에 의해 생성된 이미지들은 실제와 구분하기 힘든 수준에 이르렀다. 최근 영화들에 유행처럼 등장하는 디지털 배우들이 이러한 상황을 잘 보여주고 있다. 하지만 아직도 인간과 똑같은 CG 캐릭터를 만드는 일은 연구 과제로 남아 있으며 사실적인 디지털 배우를 만들기 위한 노력은 계속될 것이다.

본 논문에서는 사실적인 인간 캐릭터 생성에서 가장 중요한 요소로 작용하는 사실적인 피부 렌더링을 위한 피부 셰이더를 제안하였다. 기존의 전통적인 셰이더에서 표현하기 힘들었던 피부의 광학 특성을 표

현하기 위해 표면하 산란 모델을 적용하였다. 관찰을 통해서 단일 산란과 다중 산란 성분을 확인하고 단일 산란은 공학적인 근사를 통해 다중 산란은 페이킹 방법을 통해 구현하였다.

제안한 피부 셰이더를 적용한 렌더링 이미지는 기존의 전통적인 셰이더를 적용한 이미지에 비해 피부의 반투명성을 잘 표현하고 피부의 피지, 표피, 진피 층을 각각 정반사, 다중 산란, 단일 산란으로 모델링할 수 있었다. 기존의 반투명성 페이킹 방법과 비교했을 때 추가적인 라이트를 생성하지 않고도 확산 산란 성분을 제어할 수 있다.

하지만 라이팅 설정이 되어있는 환경에 적용할 때 기존의 설정을 그대로 반영할 수 없기 때문에 추가적인 조절이 필요하고 다른 물체들과 균형 있는 라이팅에 어려움이 있다. 앞으로 이러한 문제를 보완하고 실제적인 텍스처링에 적합하도록 조절하는 연구가 필요하겠다.

### 참고 문헌

- [1] Jonathan Gibbs, Rendering Skin and Hair, <http://silicon-valley.siggraph.org>, March 2001.
- [2] 대한피부과학회, 피부 과학, 대한피부과학회 교과서 편찬 위원회, 2001.
- [3] E.A. Edwards and S.Q. Duntley, The Pigments and Color of Living Human Skin, American Journal of Anatomy, July 1939.
- [4] Jos Stam, An Illumination Model for a Skin Layer Bounded by Rough Surfaces, Proceedings of the 12th Eurographics Workshop on Rendering Techniques, 2001.
- [5] P. Hanrahan and W. Krueger, Reflection from layered surfaces due to Subsurface Scattering, SIGGRAPH 93, August 1993.
- [6] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Gindberg, and T. Limperis, Geometric considerations and nomenclature for reflectance, National Bureau of Standard, 1977.
- [7] Matt Pharr and Pat Hanrahan, Monte Carlo Evaluation of Non Linear Scattering Equations for Subsurface Reflection, Proceedings of the 27th annual conference on Computer

graphics and interactive techniques, 2000.

- [8] H. W. Jensen et al, A Practical Model for Subsurface Light Transport, In Proceedings of SIGGRAPH 2001, August 2001.
- [9] H. W. Jensen et al., A Rapid Hierarchical Rendering Technique for Translucent Materials, In Proceedings of SIGGRAPH 2002, July 2002.
- [10] Christophe Hery, Implementing a Skin BSSRDF, SIGGRAPH 2003 Course 9: RenderMan, Theory and Practice, July 2003.
- [11] Matt Pharr, Layered Media for Surface Shaders, SIGGRAPH 2001 Course 48: Advanced RenderMan 3, August 2001.



**유 태 경**

2002년 중앙대학교 기계설계학과 졸업  
 2004년 중앙대학교 첨단영상대학원 영상공학과 졸업  
 2004년~현재 중앙대학교 첨단영상대학원 영상공학과 박사과정

관심분야: 디지털시각효과, 컴퓨터그래픽스, 예술공학



**이 원 형**

1981년 부경대학교 어업학과 졸업  
 1983년 부경대학교 대학원 수산물리학과 졸업  
 1987년 호주 University of New South Wales 박사과정 수료  
 1998년 국민대학교 대학원 법학과 졸업

2001년~현재 한국컴퓨터게임학회 부회장  
 2001년~현재 중앙대학교 첨단영상대학원 교수  
 관심분야: 디지털 콘텐츠 보호 기술, 컴퓨터 게임



**장 성 갑**

1988년 중앙대학교 전자공학과 졸업  
 1990년 중앙대학교 대학원 전자공학과 졸업  
 1992년~1999년 부천대학교 컴퓨터제어과 겸직교수  
 2000년 중앙대학교 대학원 전자공학과 졸업

2000년~2003년 중앙대학교 첨단영상대학원 연구교수  
 2003년~현재 중앙대학교 첨단영상대학원 교수  
 관심분야: 영상특수분야, 영상합성, 특수효과기술