

Unity3D기반 피부 투과광의 사실적 색표현을 위한 개선된 사전정의 BRDF*

김성훈*, 문윤영*, 최진우*, 양영규*, 한기태 *

*가천대학교 전자계산학과

e-mail:limiteddaily@gmail.com

Advanced Pre-Integrated BRDF for Realistic Transmission Light Color in Skin Rendering based on Unity3D

Seong-Hoon Kim*, Yoon-Young Moon*, Jin-Woo Choi*,

Young-Kyu Yang*, Gi-Tae Han*

*Dept of Computer Science, Gachon University

요 약

사실적 피부 렌더링은 피부 표면에서 일어나는 확산반사(Diffusion) 및 경면반사(Specular) 뿐 만 아니라 피부층 내에서 산란되어 나오는 산란광과 얇은 피부층을 통과하는 투과광 등을 고려하여 렌더링 되어야 한다. 이를 물리적인 개념들을 사용하여 실시간으로 계산하여 표현하는 것은 많은 계산량과 시간을 필요로 하므로 확산 반사 및 경면 반사 등을 미리 계산하여 텍스처로 저장하고 재사용하는 사전 정의 BRDF 방법으로 근사화하여 표현할 수 있다. 하지만 사전정의 BRDF를 통해 생성된 피부 투과광 색상 텍스처 맵은 그 색상이 고정되어있어 조명의 색상이 바뀌어도 피부를 투과하는 빛의 색상이 변하지 않아 부자연스러움을 보인다. 본 논문에서는 이러한 문제를 해결하기 위해 물체와 조명간의 거리를 이용하여 빛의 감쇠비율을 구하고 조명의 색상 값과 감쇠비율을 이용하여 피부 투과광 색상 텍스처 맵의 RGB채널 수정을 통해 피부 렌더링에서의 자연스러운 투과광 표현이 가능함을 보였다.

1. 서론

하드웨어와 렌더링 기술의 발전에 따라 3D모델의 구현 기술 또한 사실적인 표현이 가능하게 되었다. 이러한 기술은 산업제품뿐만 아니라 모든 분야에 적용되고 있으며 특히 영화, 게임에서의 재질별로 사실적인 표현을 위한 다양한 연구들이 활발하게 진행되고 있다.

재질을 사실적으로 표현하기 위한 렌더링은 물리적인 개념에 기반하여 계산되어 화면상에 표시되며, 재질의 특성에 따라 그 계산방법이 다양하고 복잡한 계산들이 요구된다. 그 중에서도 피부의 표현은 사람마다 각각 다른 피부 특성을 가져 천, 목재 등의 표현보다 그 복잡도가 높고 반투명 재질 특성을 가지고 있어 이를 사실적으로 표현하는 방법들이 지속적으로 연구되고 있다[1].

현재 피부의 표면을 표현하는 가장 일반적인 렌더링 방법에는 Bidirectional Reflectance Distribution Function(BRDF)이다. 이는 피부 표면에서 발생하는 빛의 물리적 현상을 정의한 것으로 표면의 한 점에 대해 빛의 입사 방향과 광량, 반사 방향과 광량을 나타내는 함수들로 표현된다[2, 3].

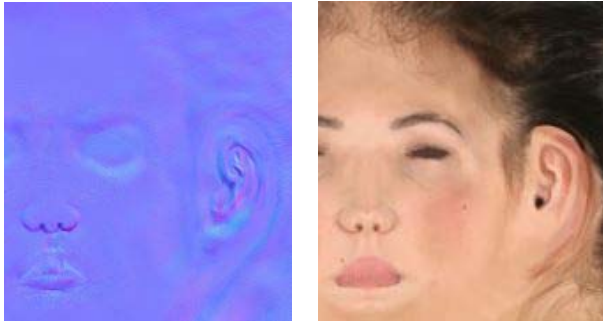
피부와 같은 반투명 재질의 내부 산란을 표현하기 위해서는 Bidirectional Surface Scattering Reflectance Distribution Function(BSSRDF)와 같이 피부 내부에서의 빛의 산란, 흡수, 투과를 정의하는 함수를 사용하여 렌더링 할 수 있으나 계산량이 많고 복잡하여 실시간으로 물체를 표현하는 게임 등에 적용하는 것은 어려움이 있다[4, 5]. 따라서 최근에는 기존의 BRDF를 개선하여 표면 렌더링 뿐 만 아니라 내부 산란 표현이 가능한 Pre-Integrated BRDF를 이용하여 피부 렌더링에 사용되고 있다.

Pre-Integrated BRDF는 사전에 계산된 다양한 맵을 반복 이용하여 렌더링하므로 계산량을 크게 줄일 수 있으나 조명 또는 재질의 변화에 따라 시각적으로 변화를 나타낼 수 없다는 단점이 있다[6]. 특히 귀와 같은 얇은 피부에서 발생하는 피부 산란 및 투과는 조명의 색상에 따라 그 투과광의 색상이 변할 수 있어야 자연스러운 피부 렌더링 결과가 나타난다. 하지만 Pre-Integrated BRDF는 피부의 투과광에 대한 색상을 고정된 색상의 Lookup image를 사용함으로써 조명환경에 따른 투과광 색상 표현이 자연스럽지 못하다. 따라서 본 논문에서는 얇은 두께의 피부에서 나타나는 투과광의 색상을 조명환경에 따라 자연스럽게 표현하기 위해 Lookup image의 RGB색상 수정을 통한 자연스러운 피부 렌더링 방법을 제안한다.

† This work was supported by the Technology Innovation Program (or Industrial Strategic technology development program, 10043453, 시네마틱 게임플레이에 적합한 물리기반 재질 렌더링 기술 개발) funded By the Ministry of Trade, industry & Energy(MI, Korea)''

2. 제안하는 방법

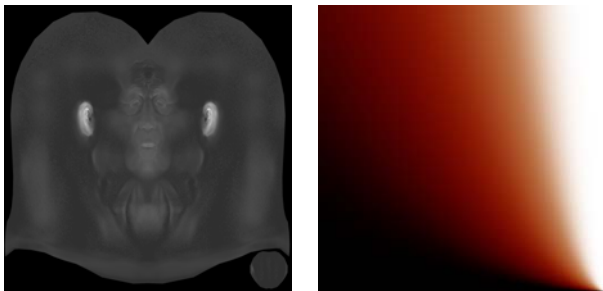
기존의 Pre-Integrated BRDF는 재질 렌더링 시 BRDF 계산에 필요 데이터들을 아래 (그림 1)과 같이 여러 개의 맵으로 미리 계산한 값을 저장하여 Lookup으로 참조함으로써 계산량을 줄이는 방법이다.



(a) Normal map (b) Diffuse map

(그림 1) Normal map과 Diffuse map

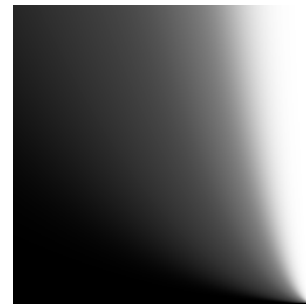
또한 피부의 투과광은 아래 (그림 2)와 같이 얼굴에서의 피부 두께 값을 나타내는 Depth map을 이용하여 값에 따라 Lookup image를 참조함으로써 얼굴 표면의 한 점에서의 투과광의 강도와 색상을 결정한다[6].



(a) Depth map (b) Lookup image

(그림 2) Depth map과 Lookup image

고정된 Lookup Image는 조명의 색상과 상관없이 항상 같은 색의 투과광을 나타내므로 이러한 단점을 보완하기 위해 제안하는 방법은 아래 (그림 3)과 같이 밝기 값만 갖는 Lookup image를 Base lookup image로 지정한다. 이는 조명마다 색상이 각기 다를 수 있고, 조명마다 구해지는 RGB채널 가중치를 적용하기 용이하도록 밝기 값만 가지는 Lookup image를 사용한다.



(그림 3) Base lookup image

밝기 값만을 갖고 있는 Base lookup image에 색상을 부여하기 위해 Base lookup image의 RGB채널 색상 값 BLL_{RGB} 에서 두개의 가중치 값을 사용하여 색상 값을 부여한다. B_i 는 혈액에 의한 기본 색상 가중치로 사람의 피부 속 혈액에 의해 투과광이 붉은색으로 나타나므로 이를 표현하기 위한 가중치이다. L_i 는 조명들의 색상과 감쇠비율을 이용하여 계산된 조명 색상 가중치로 두개의 색상 B_i 와 L_i 를 더하여 Base lookup image BLL_i 에 곱함으로써 최종적으로 수정된 Lookup Image BLL_i' 의 RGB 색상값을 계산한다.

$$BLL_i' = BLL_i * (B_i + L_i), \quad i = \{R, G, B\} \quad (1)$$

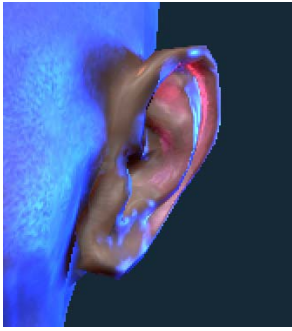
조명 색상 가중치는 조명들의 색상값과 감쇠비율을 이용하여 계산되며 감쇠비율 att_n 는 물체와 조명간의 거리 d 를 구하고 다음 식 2를 사용하여 n개의 조명마다 가중치 감쇠비율 att_n 을 구한다[7].

$$att_n = \min\left(\frac{1}{k_c + k_l d_n + k_q d_n^2}, 1\right) \quad (2)$$

k_c, k_l, k_q 는 빛의 감쇠 계수로 k_c 는 선형 감쇠 계수, k_l 는 2차 감쇠 계수, k_q 는 지수 감쇠 계수이다. n개에 조명에 대해 계산된 감쇠비율 att_n 은 아래 식과 같이 k번째 조명 l_k 의 각 RGB채널에 해당되는 컬러값을 곱하고, 해당 채널에서의 조명 색상 가중치를 전부 합산한다.

$$L_i = \sum_{k=0}^n l_{k_i} * att_k, \quad i = \{R, G, B\} \quad (3)$$

따라서 기본 색상 가중치와 조명 색상 가중치에 의해 Base Lookup image의 색상이 결정되고 이에 따라 피부의 투과광 색상이 (그림 4)와 같이 자연스럽게 나타난다.



(그림 4) 제안하는 방법의 투과광 색상

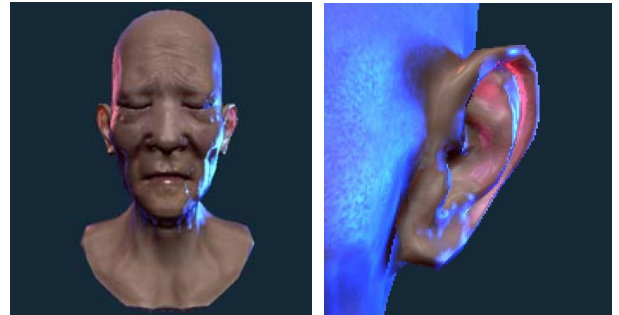
명환경은 아래 <표 1>과 같이 구성하였다.

<표 1> 실험 조명환경

구 분	R	G	B	Distance
Light1	255	112	69	0.92357
Light2	15	64	255	0.85231
Light3	255	255	255	1.01292

3. 실험결과

제안한 방법의 실험은 Unity3D 기반의 환경에서 Pre-Integrated BRDF로 구현된 피부 셰이더를 사용하였다. 기존의 방법을 사용하여 피부 투과광을 표현하였을 때 아래 (그림 5)과 같이 나타나는 것을 볼 수 있으며 투과광 맵의 색상이 고정되어 있어 (그림 6)과 같이 조명의 색상이 변하여도 귀 부분의 얇은 피부에서 나타나는 투과광의 색상이 변하지 않는 것을 볼 수 있다.

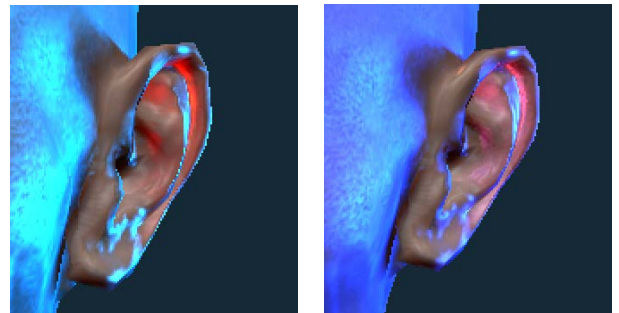


(그림 7) 제안하는 방법의 투과광 색상

기존 방법과 비교하였을 때, 제안하는 방법은 조명 색상에 따라 투과광이 변하는 것을 현실감 있게 나타낼 수 있었고, 아래 (그림 8)이 그 결과를 나타내고 있다.

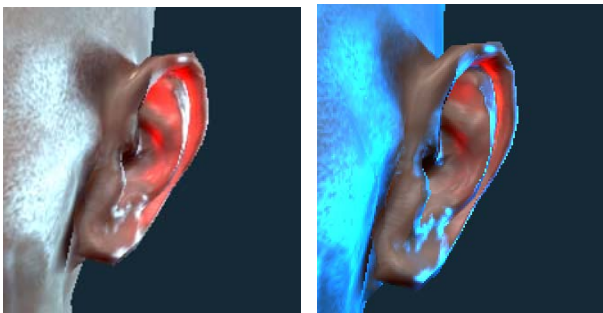


(그림 5) Unity3D기반 Pre-Integrated BRDF 피부 셰이더



(a) 기존 방법 (b) 제안하는 방법

(그림 8) 기존 방법과 제안하는 방법의 비교



(a) 조명 색상 변경 전 (b) 조명 색상 변경 후

(그림 6) 기존 방법에서의 조명 환경 변화에 따른 투과광 색상 변화

제안한 방법은 조명의 색상값과 조명 가중치를 사용하여 Lookup image의 RGB채널 색상을 변경하므로 기존 방법에서의 부자연스러운 투과광 색상을 (그림 7)과 같이 자연스럽게 표현할 수 있다. 기본 가중치 B_{RGB} 는 R채널에서 1.8, G채널에서 0.5, B채널에서 0.4를 사용하였으며 조

4. 결론 및 향후 연구방향

본 논문에서는 사전정의 BRDF에서 조명의 색상과 거리에 따라 얇은 피부를 투과하는 빛의 색상이 변하도록 하여 기존보다 사실적인 표현이 가능한 방법을 제안하였다. 기존의 사전정의 BRDF방법은 투과광 색상 텍스처 맵의 색이 고정되어 있어 조명의 색상이 변하여도 투과광의 색이 변하지 않아 부자연스러움을 나타내었으나 제안하는 방법은 조명의 색상값과 물체와 조명간의 거리를 이용하여 색상 가중치 값을 구하고, 이를 투과광 색상 텍스처 맵의 RGB값에 적용하여 조명의 색상과 거리에 따라 투과광의 색상을 자연스럽게 하여 피부재질의 표현이 자연스러워졌음을 알 수 있었다. 향후 연구에서는 조명의 세기, 방향 등을 고려한 물리기반의 가중치를 Lookup image에 적

용하여 사실적인 표현이 가능한 렌더링 방법을 연구할 예정이다.

참고문헌

- [1] Clemens Brandorff, "Rendering Human Skin", Seminar mit Bakkelaureatsarbeit, 2006.
- [2] NVIDIA Corporation, Chris Wynn, "An Introduction to BRDF-Based Lighting", 2000.
- [3] Ling Li, Carmen So-ling Ng, "A Physically-based Human Skin Reflection Model", WSEAS International Conference on AUTOMATION & INFORMATION. 2009.
- [4] NVIDIA Corporation, Eugene d'Eon, David Luebke and Eric Enderton, "Efficient Rendering of Human Skin", Eurographics Symposium on Rendering, 2007.
- [5] Erik Rune, "Efficient Rendering of Human Skin", Eurographics Symposium on Rendering, 2007.
- [6] Jorge Jimenez Garcia, "Practical Real-Time Strategies for Photorealistic Skin Rendering and Antialiasing", Zaragoza University, 2012.
- [7] Feng Qian, "Tutorial Project of Computer Graphics 3D Renderer(Ray Tracing Engine)", 2006.