실시간 영상기반 라이팅을 위한 고속 노말맵 추출방법 및 라이팅 기술 연구

A Study of Normal Map Extraction and Lighting Technology for Real-time Image Based Lighting

유세운, 방찬영, *이상화, **이상엽, **안상철, 박종일,
한양대학교, *서울대학교, **한국과학기술연구원

요약; 최근 가상현실 기술의 주요 연구 동향으로 몰입감을 증가시키는 실감공간 구현구술이 주목 받고 있다. 실감공간 기술이란 서로 다른 공간에 떨어져 있는 사용자가 같은 공간에 있는 효과를 구현하는 기술이다. 본 논문에서는 특히 상호간의 주변 환경을 일치시키는 기술에 중점을 두고, 실시간으로 두 공간의 조명정보를 일치시키는 기술로서 2 가지 핵심 내용을 소개한다. 첫째는 비주얼 헐 데이터를 기반으로 고속으로 노말벡터를 추출하는 방법이고, 둘째는 사용자 주변 조명 환경 정보를 반영하는 라이팅 방법이다. 본 논문에서 수행한 첫번째 방법은 비주얼 헐 데이터의 depth 존재영역에서 노말맵을 계산하도록 하고, 노말맵을 계산할 때 주변 폴리곤들 기하학적 변화가 심할수록 노말맵 계산에 사용하는 주변 벡터의 선태을 늘리거나 줄이는 방식으로, 불필요한 계산량을 감소시켰다. 본 논문에서 수행한 두번째 방법에서는 주변 조명 정보에서 빛의 세기와 라이팅을 반영할 객체의 반사율의 특성을 고려하여 라이팅에 사용할 광원을 선택적으로 반영하여 불필요한 연산량을 감소시켰다. 종래의 영상기반 라이팅 기술이 사전에 촬영된 영상을 사용하거나 정지영상에 적용되는 연구를 한 반면에 본 논문은 실시간에서 라이팅을 구현하기 위한 시도로서 고속 라이팅 연산 기법을 제시하고 있다. 본 연구의 결과를 이용하면 영상기반 라이팅 연구의 실제적이고도 폭넓은 적용이 가능할 것으로 사료되며 고화질의 콘텐츠 양산에도 기여할 것으로 사료된다.

핵심어: CG, IBL, Normal Map, Visual Hull

1. 서론

실감공간 구현 기술의 목적은 서로 다른 공간에 있는 사용자가 마치 같은 공간에 함께 있는듯한 현실감을 느끼게 하는데 있다. 사용자가 같은 공간에 있다고 지각하기 위해서는 시각, 청각, 후각, 미각, 촉각의 5감각 요소들을 일치시키는 방법이 필요하다. 본 논문은 5감각 중에서 시각적인 요소를 실시간으로 통합하는 기술 방법을 제안한다.

서로 다른 공간의 시각적인 요소를 통합하기 위해서는 두 가지 정보가 필요하다. 첫째는 조명 환경 정보이고, 둘째는 조명정보가 다른 공간의 물체에 대한 기하학적 정보와 텍스처 정보이다. 이 두 가지 요소를 이용하여 조명정보가 다른 물체의 기하학 정보와 텍스처 정보에 통합할 조명환경 정보를 랜더링하여 시각적인 실감공간이 구현되는 것이다.[1][2]

조명 환경 정보를 도출하는 방법으로는 미러볼을 사용하는 방법이 보편화되어 있다. 미러볼은 구 모양으로 생긴 전반사하는 물체를 가르킨다. 미러볼을 카메라로 촬영하여 얻어진 데이터를 이용하여 공간속의 광원의 방향과 색정보를 도출하는 것을 목적으로 한다. 이때 주의할 점이 두 가지 있다. 첫째는 광원의 방향을 연산할 때 미러볼의 중심을 기준으로 하고 미러볼 표면에서 반사되어 카메라로 입사하는 광경로를 계산하여야 한다. 따라서 미러볼의 반지름과 미러볼과 카메라간의 위치정보를 정확하게 고려해야 비교적 정확한 조명 환경 정보를 도출할 수 있다. 둘째는 촬영하는 카메라는 센서의 고유한 색특성을 갖고 있는데 이것을 보정하여 HDRI 영상을 활용해야 보다 사실적인 조명정보를 구할 수 있다.[3][4]

물체의 기하학적 정보는 그래픽 툴을 활용하여 제작한 폴리곤 모델을 사용하거나 실사물체로부터 깊이정보 맵을 도출하여 얻어진 모델 데이터를 사용한다. 본 논문에서는 실사물체로부터 얻어진 깊이정보 맵과 텍스처 영상을 사용하여 재조명 방법을 제안한다.

본 논문에서 조명환경을 일치시키는 랜더링을 재조명(relighting)이라 부를 것이며, 재조명을 수행하기 위해 물체의 깊이정보 맵을 각 정점(vertex)에서 법선벡터를 추출한 법선벡터맵(normal map)으로 변환하여 사용할 것이다. 이는 광선이 물체 표면에서 반사되는 경로를 계산할 때 편의를 위한 것이다. 재조명은 조명정보를 추출하고, 고속으로 물체의 법선벡터맵을 추출하여, '퐁'의 광원모델(Phong shading model)에 적용하여 재조명 영상을 생성한다.[5]

2. 실감공간 기술 개요

서로 다른 두 공간상의 조명환경 정보를 일치시키기 위해 그림 1과 같은 시스템을 구성하였다. 두 공간의 명칭을 Site A와 Site B로 구분하였다. Site A 공간에서 사용자 User A와 로봇 User B'이 존재한다. 이때 로봇 User B'는 Site B의 사용자 User B를 대신하여 Site A에 존재하는 아바타이다. 이 아바타는 User B의 움직임과 동일한 행동을 한다. Site B 공간은 사방이 빔 프로젝트 스크린으로 구성되며 Site A의 조명정보를 투영한다.

그림 1의 시스템은 Site A 공간의 조명환경을 기준으로 Site B의 사용자 User B의 영상을 실시간으로 재조명 영상을 합성한다.

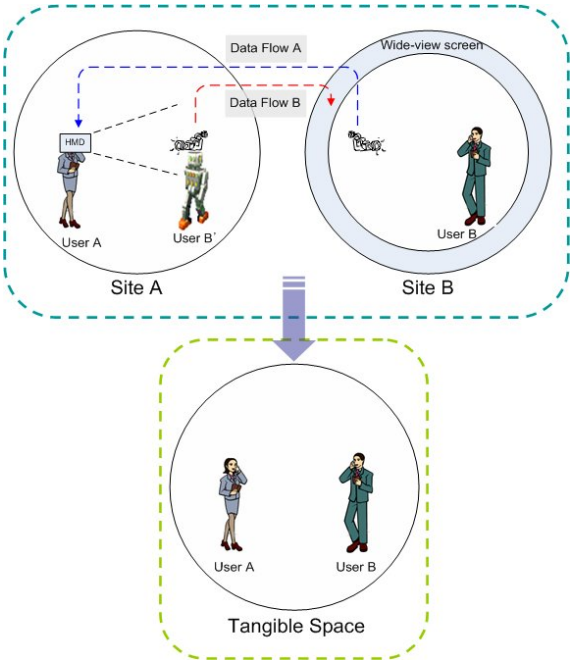


그림 1 실감공간 구성

본 논문에서 수행하는 재조명은 Site B의 User B의 3차원 기하학적 정보와 텍스쳐 영상을 이용하여 Site A의 조명환경 정보에 맞게 라이팅 랜더링을 수행하는 것이다. 그림 2는 본 논문에서 수행하는 실시간 재조명 시스템의 구성을 나타낸다.

그림 2는 실시간 재조명 시스템의 중요한 데이터 흐름을 3가지 요소를 보여준다. 첫째는 Site B의 User B의 모델정보이고, 둘째는 Site A의 조명환경 정보이고, 셋째는 재조명 영상을 합성하는 연산장치로부터 영상을 출력하는 것이다.

실시간으로 모델정보를 도출하기 위해서는 고속으로 User B의 깊이정보맵과 텍스처 영상을 입력받아 법선벡터 맵을 생성하는 것이 중요하다. 일반적으로 많이 사용하는 폴리곤 모델데이터는 법선벡터를 추출할 때 인접한 3개의 정점(vertex)로부터 벡터의 외적을 구함으로써 법선벡터를 정의한다. 반면에 본 논문에서 사용하는 모델은 깊이정보 맵을 가로, 세로 512*512 개의 정점으로 변환하여 사용하는 모델이다. 따라서 한 정점의 법선 벡터를 구할 때는 인접한 8개의 정점의 위치관계를 고려하여 총 8개의 삼각형의 외적을 계산해야 하는 특성을 갖고 있다. 이를 고속 처리하는 방법은 제 4장 "객체정보 취득 및 고속처리 방법"에서 소개한다.

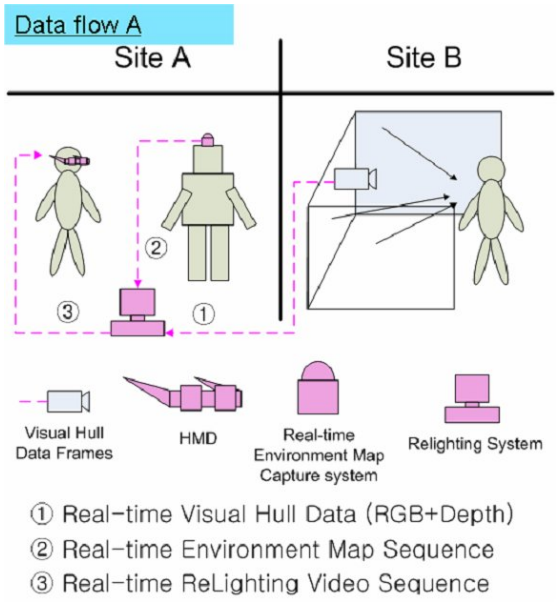


그림 2 실시간 재조명 시스템 구성

실시간으로 재조명 영상을 합성하기 위해서는 실시간으로 변화하는 조명환경 정보와 모델정보의 고속 처리가 필요하다. 일반적으로 실시간이라 함은 초당 30프레임의 영상을 출력하는 것을 목표로 한다. 실시간으로 조명환경정보를 도출하기 위해 미러볼과 미러볼을 촬영하는 카메라를 고정시킨 후 초당 30프레임으로 영상을 촬영한다. 촬영된 영상을 그림2의 두번째 라인과 같이 연산장치로 보내고, 연산장치에서는 3장 '조명정보 고속 추출 방법'에서 설명하는 방법에 따라 조명환경 정보를 임시 메모리에 저장한다.

일반적으로 카메라의 센서는 고유한 색 특성을 갖는다. 칼라정보를 빨강, 초록, 파랑의 3 채널로 구분할 때, 각 채널별 밝기 대비 입력 특성 곡선이 제각각 다르다. 이 3채널별 특성 곡선을 고려하여 카메라 센서의 색 특성을 정규화해야 광원의 고유 특성을 정상적으로 반영할 수 있다.

그림 2에서 첫번째 라인과 두번째 라인이 모두 하나의 연산장치에 의해 수행되는 구조를 갖는다. 재조명 수행 연산은 모델의 표면에서 광원이 반사되는 경로를 추적하는 과정을 처리하는데 퐁 라이팅 모델을 활용한다. 광원의 특성에 따라 반사광은 주변광(ambient reflection), 전반사(specular reflection) 부분과, 난반사(diffusion refraction) 부분으로 구분하고 적용한다.

영상합성 시스템은 고속 연산 처리를 위하여 CPU와 GPU(graphic processing unit)를 이용하여 재조명 영상 합성을 수행하였다. GPU는 일반 컴퓨터의 VGA에서 영상처리를 수행하는 프로세서로서 근래에는 프로그래밍 가능한 GPU가 보급되면서 사용자가 GPU를 사용하여 효과적인 랜더링이 가능해졌다.[6]

GPU를 사용하여 랜더링을 수행하는 것은 CPU만을 사용하여 랜더링을 수행하는 것에 비하여 고속 렌더링이 가능하다. GPU에서는 CPU에서 기본 연산 명령어들로 조합한 복잡한 연산을 하나의 연산 명령어로 처리할 수 있는 장점이 있다. 그러나 재조명 연산은 매우 많은 연산량을 수행하므로 하드웨어의 우수한 성능을 최적화하기 위해 최적화된 알고리즘이 무엇보다 중요하다.

3. 조명정보 고속 추출 방법

3.1 조명 촬영 장치 색특성 교정

영상을 촬영할 때 카메라 센서의 고유한 색특성을 고려하여야 한다. 일반적으로 카메라의 센서는 빌치 들어오는 광량의 세기와 카메라가 인식하는 밝기 세기에 비선형적인 특성을 나타낸다. 특히 칼라정보를 빨강, 초록, 파랑의 3 채널로 구분할 때, 각 채널별 밝기 대비 입력 특성 곡선이 제각각 다를 경우가 있다. 이 3채널별 특성 곡선을 고려하여 카메라 센서의 색 특성을 정규화해야 광원의 고유 특성을 정상적으로 반영할 수 있다. 그림 3은 카메라의 색 특성을 교정하고 비선형적인 특성을 나타낸 그림이다. 그림 3의 (a)는 광량차가 서로 다른 영상을 촬영하고, (b)는 촬영한 카메라의 색특성 곡선을 나타내고 있다.[7][8] 본 논문은 고속 재조명을 중점을 두므로 “RASCAL Software”를 사용하여 카메라 색특성을 쉽게 도출하였다.

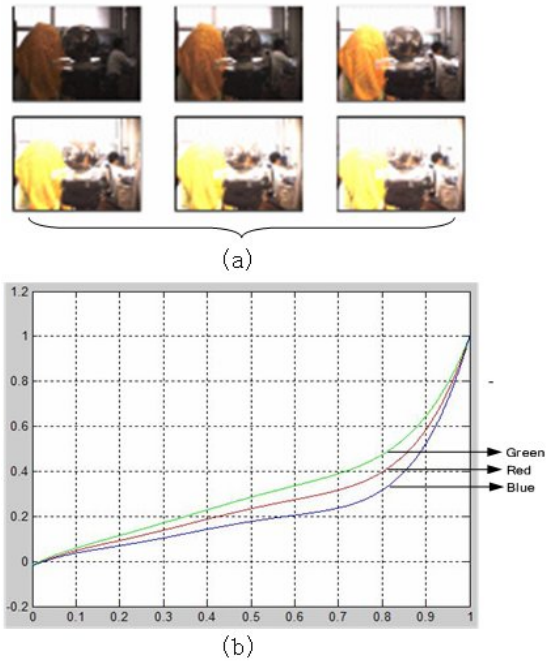


그림 3 카메라 색 특성 교정

도출한 카메라 색특성을 고려하여 칼라영상의 각 채널별 16비트 해상도를 갖는 48비트 HDRI(high Dynamic Range Image) 영상을 생성할 수 있다.

HDRI 영상을 조명정보 영상으로 사용하면 광원의 밝기 차이에대해 보다 섬세한 표현이 가능하다. 일반적으로 영상처리에 사용하는 빨강, 초록 파랑의 3채널별로 8비트를 사용하는 24비트 칼라 영상은 광량을 256간격으로 표현한다. 특히 카메라 센서는 비선형적인 색특성을 가지므로 셈플링 간격 사이에 존재하는 조명 밝기 정보 특성을 상실하게 되는 경우가 발생하기 쉽다. 채널별 밝기 샘플링 간격을 더욱 좁히고 선명한 영상을 얻기 위해 채널별 16비트의 고정밀 영상을 조명 환경 정보 맵을 작성하는데 사용한다.

3.2 영상으로부터 조명 정보 도출

미러볼을 대상으로 영상을 촬영하고, 입력된 영상으로부터 미러볼 부분만을 추출한다. 그리고 2차원의 미러볼 영상으로부터 3차원 그리드 구 모델을 투영하여 미러볼의 형태정보를 표현한다.. 미러볼의 영상은 픽셀정보가 곧 광원을 의미한다. 하지만, 인접한 픽셀과의 유사도를 고려하여 미러볼을 위도와 경도를 나누어 셈블링하고 셈플링 된 구간의 평균값을 취해 광원의 칼라 정보로 정의하면 적은 수의 광원으로 재조명 효과를 구현 할 수 있다. 그림 4는 미러볼 영상으로부터 3차원 구를 투영한 조명환경 맵을 표현하는 과정을 나타낸다

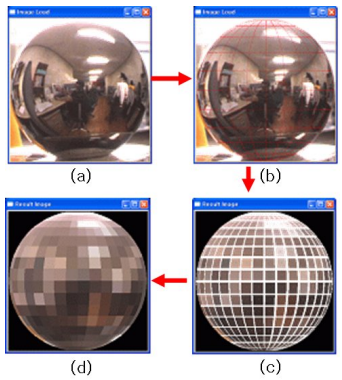


그림 4 미러볼 조명환경 맵 생성과정

그림 4의 과정을 거쳐서 3차원으로 투영된 미러볼 영상으로부터 미러볼 중심을 기준으로 미러볼로 입사한 광원의 위치정보를 도출한다. 그림 5는 광원의 방향 벡터를 추출하는 과정을 나타낸다. 그림 6은 2가지 셈플링 방법을 소개한다. (a)는 원본 영상이고, (b)는 위도와 경도를 기준으로 셈플하는 방법으로 정면에서는 균등한 광원의 분포를 보이지만 가장자리 부분에서 불균등한 점광원 분포를 갖는다. (c)는 화면의 중심을 축으로 동심원으로 셈플링하는 방법으로 가장자리 부분의 점광원을 균등하게 분포하지만, 정면에는 셈플링 간격이 커서 광원 정보가 줄어드는 특징이 있다.

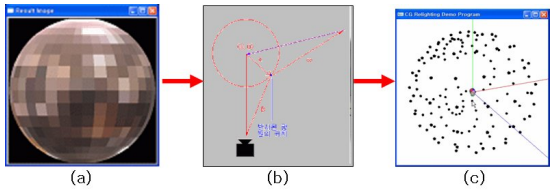


그림 5 조명환경맵의 광원 벡터계산 과정

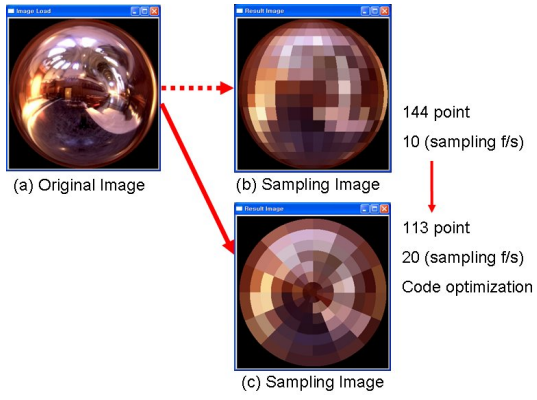


그림 6 셈플링 좌표계 변환

4. 객체정보 취득 및 고속 처리방법

4.1 모델정보 취득

그림 1에서 Site B의 User B에 대한 정보는 그림 7과 같은 깊이정보 맵과 텍스처 영상이다. 그림 7의 (a)는 가로 세로 512*512 사이즈의 깊이정보맵을 3차원 기하학적 형태로 표현한 것이고, (b)는 이 모델의 텍스처 영상이다. 택스처 영상과 깊이정보 맵의 사이즈는 서로 같다.

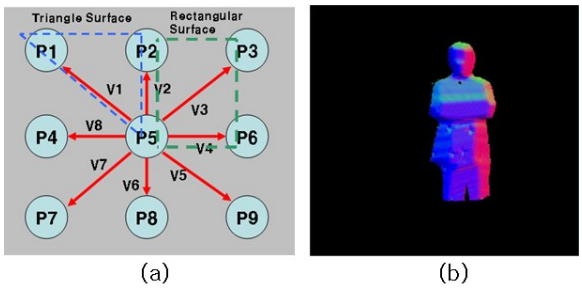
(a) (b)

그림 7 모델정보와 텍스처 정보

재조명은 '퐁'의 라이팅 모델을 사용하여 영상을 합성한다. 이를 위해 깊이정보 맵으로부터 각 정점(vertex)로부터 법선벡터를 구해야 한다. 일반적으로 폴리곤 모델로부터 법선을 추출할 때에는 3개의 정점으로 구성된 삼각형 평면의 외적을 구하여 계산한다. 그림 8의 (a)는 인접한 정점으로부터 법선벡터를 구할 때 선택하는 정점의 경우를 표현한다. (b)는 각 정점에서 법선벡터를 계산한뒤 RGB 3원색으로 표현한 결과 영상이다.

원칙적으로 P5의 정점에서 법선벡터를 구하기 위해서는 P1에서 P9 까지의 총 8개의 정점을 선택하여 8개의 삼각형 평면으로부터 법선벡터를 추출하고 이것의 평균값을 구해야 할 것이다. 하지만 깊이정보맵의 외형이 완만한 부분의 정점의 법선벡터를 계산할때에는 8개의 정점을 모두 사용하지 않고 그 절반인 4개의 정점만을 택해서 법선벡터를 계산해도 8개의 정점으로부터 법선벡터를 계산한 결과 비교하여 거의 동일한 결과값을 구할 수 있었다.

모델의 외형의 형태에 따라서 법선벡터를 계산할 때 선택하는 정점의 개수를 줄임으로써 연산속도를 높일 수 있다.



(a) (b)

그림 8 깊이정보 맵으로부터 법선벡터 맵을 도출

4.2 라이팅 모델

앞서 구한 법선벡터 맵을 이용하여 3장에서 샘플링한 점광원들의 경로를 그림 9와 같은 광원 모델을 사용하여 재조명 되는 영상을 합성한다. [5]

그림 9에서 L은 광선으로부터 입사하는 광선 벡터, N은 물체 표면의 법선벡터, Θ는 광선의 물체표면에 대한 입사각, R은 입사각과 동일하게 반사되는 반사광, V는 물체의 표면에서 산란되는 광선이며, Φ는 매질특성에 따라 산란되는 빔 폭을 나타내는 사잇각이다.

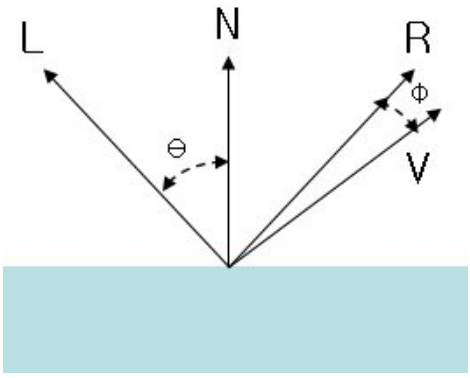


그림 9 라이팅 모델

광원 모델에 따라 3가지로 구분할 수 있다. 첫째는 주변광 둘째는 전반사 부분, 셋째는 난반사 부분이다.

주변광(ambient reflection)은 수식 1과 같이 나타내며, Ia는 일정한 세기의 주변광, Ka는 모델 표면에서 반사되는 주변광 상수를 나타낸다.

$$I = k_a I_a \tag{1}$$

난반사(diffusion refraction)는 수식 2와 같이 나타내며, Ii는 입사광선의 세기, Θ는 입사각, Kd는 물체 매질의 난반사 계수를 나타낸다. Id는 난반사 부분의 세기를 나타낸다.

$$I_d = I_i k_d \cos\theta \qquad 0 \le \theta \le \frac{\pi}{2}$$
$$I_d = I_i k_d (\hat{L} \cdot \hat{N}) \tag{2}$$

전반사(specular reflection)는 수식 3과 같이 나타내며, 주 반사광원과 Φ각도의 빔 폭을 갖는 V벡터 영역을 커버한다.

$$\cos^n \Phi \tag{3}$$

위 세가지 반사광원들을 모두 합성하여 수식 4와 같은 재조명 영상을 합성할 수 있다. Ks는 매질의 전반사 계수를 나타낸다.

$$I = I_a k_a + \frac{I_i (k_d (\hat{L} \cdot \hat{N}) + k_s (\hat{R} \cdot \hat{V})^n)}{(r + k)} \tag{4}$$

4.3 고속 라이팅 연산방법

3장에서 소개한 실시간으로 셈플링한 점광원과 4장에서 소개한 모델의 기하학적 형태 변화에 따라 법선벡터 연산에 선택하는 정점의 개수 변화 방법을 사용하여 고속으로 재조명 영상을 생성 할 수 있다.

조명 환경맵의 해상도는 가로 세로 360*360을 할당하였다. 이때 모든 픽셀들을 재조명 광원으로 사용할 경우 약 10만개의 점광원을 도출 할 수 있다. 반면에 3장에서 제안한 셈플링 방법을 사용하여 도출한 113~144 개의 점광원을 사용하여 생성한 결과 영상은 그 이상의 점광원을 사용한 재조명의 결과 영상과 거의 일치하는 효과를 거두었다. 제안하는 조명 환경의 셈플링을 방법을 통해 효과적인 재조명이 가능하면서 초당 30프레임의 실시간에 준하는 재조명 영상을 도출하였다.

또한 모델 데이터의 법선벡터 맵은 인접한 정점과의 형태가 복잡도를 고려하여 선택하는 정점의 개수를 8개로 할지 4개로 할지를 선택적으로 변화하면서 연산한다. 이를 통해 연산량을 최대로 절반에 가깝게 줄일 수 있다.

5. 실험결과

그림 10은 재조명 영상 합성 결과 영상을 나타낸다. (a)는 재조명을 수행하지 않는 영상이고, (b)는 재조명을 수행하여 전반사, 난반사 영상이 혼합된 결과영상이며, (c)는 전반사 영상 성분을, (d)는 난반사 영상 성분을 나타낸다. 특히 그림 (c)와 (d)의 하얀색 박스 안 부분이 전반사와 난반사 재조명 결과 차이를 잘 보여준다.

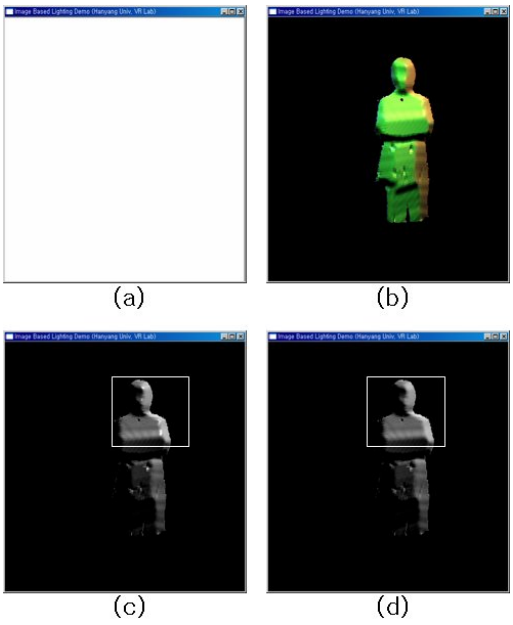


그림 10 전반사와 난반사 재조명 합성 결과

그림 11은 텍스처 맵을 적용하고, 조명 환경맵을 적용한 재조명 영상합성 결과를 나타낸다. (a)는 법선벡터 맵, (b)는 텍스처 영상, (c)는 조명 환경맵, (d)는 (c)의 조명에 따른 재조명 결과 영상을 나타낸다. 데모효과를 극대화시키기 위해 무지개 색깔을 조명환경 맵으로 사용하였다.

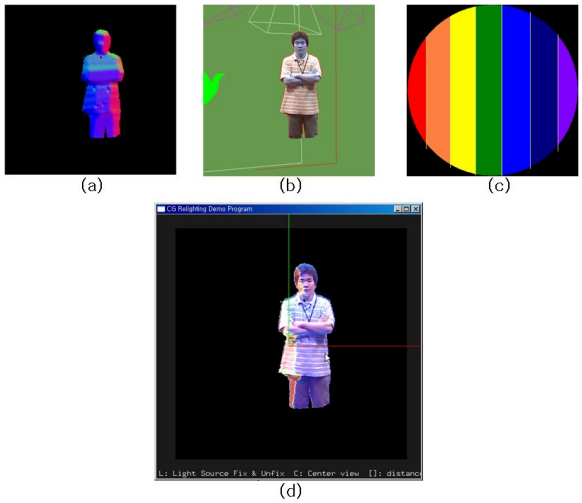


그림 11 텍스쳐 맵과 조명환경 맵을 재조명 합성 결과

6. 결론 및 향후계획

본 논문은 실시간으로 조명환경 정보를 도출하여 이를 실사 물체 모델에 재조명 하는 방법을 제시하였다. 실험 결과를 통하여 비교적 사실적인 재조명 영상을 실시간으로 합성할 수 있었다.

현재 실험에 사용한 모델은 객체의 매질 특성이 동일하다는 가정을 하고, 매질 변화에 따른 영향을 고려하지 않았다. 향후 더욱 고정밀 재조명을 수행 하기 위해서는 하나의 객체 안에서 다양한 매질 특성을 고려한 재조명 기술이 필요하게 될 전망이다. 하지만 고정밀 재조명을 수행하기 위해서는 더 많은 연산량을 필요로 하게 될 것이다.

좋은 품질의 랜더링 결과를 얻기 위해서는 그에 따르는 연산량 부담이 증가하게 된다. 이를 얼마나 합리적으로 최적화 하는 관점에서 본 논문에서 제기한 셈플링 방법과 모델의 기하학적 형태요소에 능동적으로 연산량을 가감하는 기법이 활용될 수 있을 것이다.

참고문헌

[1] P. Debevec, “Image-based lighting” Computer Graphics and Applications, IEEE Vol. 22, Issue 2, pp.26~34 March-April. 2002.

[2] P. Debevec et al., “Acquiring the Reflectance Field of a Human Face,” Computer Graphics (Proc. Siggraph 2000), ACM Press, New York, pp. 145-156. 2000,

[3] P. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," In SIGGRAPH 97, pp.369-378, Aug, 1997.

[4] P. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," In SIGGRAPH98 pp. 189~198, 1998.

[5] R. L. Cook and K. E. Torrance, "A Reflectance Model for Computer Graphics", Proceedings of SIGGRAPH 81, pp. 307~316. 1981,

[6] NVIDIA. "NVIDIA GPU Programming Guide". http://developer.nvidia.com/object/gpu_programming_guide.html.

[7] T. Mitsunaga and S.K. Nayar. "Radiometric Self Calibration,". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol.1, pp.374~380, Jun, 1999.

[8] CV Lab. Columbia Univ. "RASCAL Software". http://www1.cs.columbia.edu/CAVE/software/rascal/rrslrr.php.