

N색 분산 포톤매핑을 이용한 빛의 스펙트럼 표현

곽용식⁰

류승택

한신대학교 컴퓨터공학부

gyohall@naver.com, stryoo@hs.ac.kr

Representation of Light Spectrum using N-color Dispersion Photon Mapping

YoungSik Gwak⁰

SeungTaek Ryou

School of Computer Engineering, HanShin University

요 약

물체의 색은 모양과 더불어 사람이 바깥세상의 사물을 알아보는데 중요한 역할을 한다. 이런 색상을 인식하는 것은 태양과 같은 빛이 존재하기 때문이다. 빛은 여러 파장들로 구성되어 있으며, 눈에 보이지 않는 적외선, 자외선의 영역과 색을 인식시켜 줄 수 있는 눈에 보이는 가시광선 영역 등으로 이루어져 있다. 이러한 모든 파장의 빛이 균등하게 혼합되면 그 빛은 흰 색을 띄게 되며, 이를 백색광이라고 한다. 백색광이 굴절이 이루어지면 그 구성되어 있는 가시광선 영역이 여러 가지 색으로 갈라지는데 이것을 분산현상이라 한다. 본 논문에서는, 백색광을 무지개의 대표적인 일곱 가지 색상으로 구성하고 자연의 분산현상을 시뮬레이션하기 위해 N색 분산 포톤 매핑 방법을 제안하였다.

Abstract

The color of object is a main role that people recognize outdoor entity with its shape. We can perceive the object due to the existence of light such as direct sunlight. Light is classified by wavelength into radio, microwave, infrared, the visible region we perceive as light, ultraviolet, X-rays and gamma rays. White light is all of the colors of light combined within the visible light spectrum. When white light is separated through a prism, we see the visible light spectrum. The various wavelengths of visible light are separated into colors. In this paper, we construct white light as the seven colors of rainbow and suggest the method of N-way color dispersion photon mapping to simulate the natural dispersion phenomenon.

키워드: 분산, 포톤 매핑, 프리즘, 코스틱, 백색광

Keywords: Dispersion, Photon Mapping, Caustic, Prism, White Light

1. 서론

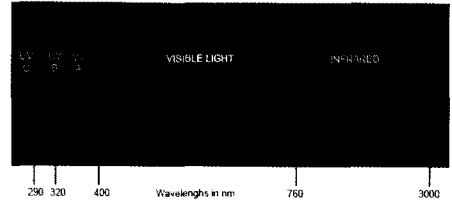
물체의 색은 모양과 더불어 사람이 사물을 알아보는데 중요한 역할을 한다. 물체의 색상을 인식하는 것은 태양과 같은 빛이 존재하기 때문이다. 빛은 여러 파장들로 구성되어 있으며, 눈에 보이지 않는 적외선, 자외선의 영역과 색을 인식시켜 줄 수 있는 눈에 보이는 가시광선 영역 등으로 이루어져 있다는 것을 알 수가 있다. 이러한 모든 파장의 빛이 균등하게 혼합되면 그 빛은 흰 색을 띄게 되며, 이를 백색광이라고 한다. 빛은 공기 속이나 어떤 매질을 통과할 때 여러 가지 현상이 나타나는데, 대표적으로 반사, 굴절, 산란, 분산 등이 있다.

기존의 분산 방법은 실제로 빛이 분산되어지는 원리를 계산해내기에 필요한 많은 시간으로 인해서, 미리 정의해 놓은 값에 의해서 표현을 하거나 텍스처 매핑을 이용하여 표현을 하였다. 이는 간이적인 표현만이 가능하기에 요즘 추세인 현실에 가까운 렌더링을 하기에 부족한 결과를 보여준다. 그러나, 최근에는 하드웨어의 발전과 함께 좋은 성능을 보여주는데 반해 많은 계산량으로 인해 생기는 속도문제로 실용화되지 못했던 레이 트레이싱(Ray-Tracing), 변위매핑(Displacement mapping), 모션블러(Motion Blur) 등의 기술들이 실제 게임에서 적용되어 가고 있다. 이는 Nvidia사의 범용 병렬 컴퓨팅 아키텍처인 CUDA를 통하여 이전보다 실사용의 폭이 넓어져 현실에 가깝게 표현하는 것이 가능해졌다.

본 논문에서는 빛이 굴절에 의해 여러 가지 색으로 갈라지는 분산현상에 대한 표현 방법을 제안한다. 조명모델의 백색광을 이루고 있는 색상을 기존의 단색광이 아닌 실제의 백색광과 같이 여러가지 색상으로 구성되어져 있다고 정의하고 빛을 프리즘을 통과시켰을 때 나타나는 코스틱(Caustic)현상과 빛의 굴절 원리를 적용했을 시에 나타나는 분산 현상을 표현하고자 한다.

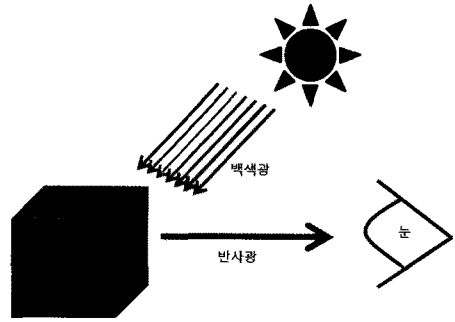
2. 관련연구

우리 생활 주변에서는 대단히 많은 종류의 색들이 나타나는 현상을 볼 수 있다. 이런 현상들은 빛에 의해서 발생하게 되는데, 빛은 에너지의 한 형태이고, 그림 1과 같이 눈으로 볼 수 있는 $8 \times 10^{-7}m \sim 4 \times 10^{-7}m$ 의 파장을 가지고 있는 가시광선 영역과 적외선, 자외선 등의 눈에 보이지 않은 영역들을 모두 가지고 있다. 이러한 가시광선 영역의 모든 진동수를 대체적으로 같은 양으로 포함하는 빛이 흰색으로 인식되며 이러한 빛을 백색광이라고 부른다. 그래서 백색광은 모든 방향으로 모든 가시광 진동수를 갖는 빛을 방출하는 것으로 생각할 수 있다[1].



[그림 1] 빛의 파장의 종류

물은 투명하지만 우유빛 유리와 비슷하게 수증기는 흰색으로 나타나게 되는데, 그 이유는 다음과 같다. 만약 입자의 크기가 빛의 파장보다 훨씬 크다면, 빛은 투명한 입자에 들어가서 여러번 반사와 굴절을 한 후에 나올 것이다. 이때 모든 파장에 대한 반사율이 같기 때문에 그림 2와 같이 관찰자에게 도달하는 반사광은 흰색일 것이다.



[그림 2] 색의 인식방법

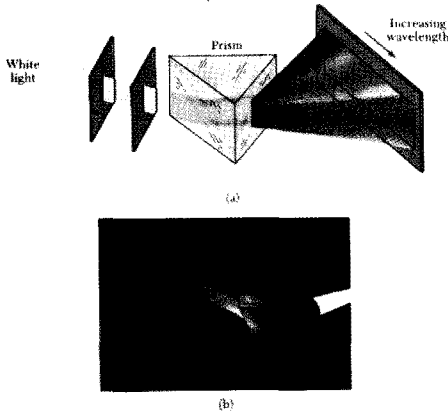
2.1 프리즘

빛의 분산현상을 확인하는 방법으로는 빛을 프리즘(Prism)에 통과시켜서 나타나는 연속스펙트럼을 관찰하는 것으로 확인할 수 있다. 프리즘은 빛을 굴절, 분산시키는 광학 도구이며, 일반적인 프리즘은 단면의 모양이 정삼각형이고 표면은 평탄하며 유리와 같은 투명한 재질로 이루어져 있다.

빛이 프리즘을 통과하게 되면, 빛을 구성하고 있는 파장들은 각각의 굴절률에 의하여 스넬의 법칙에 의해 굴절을 이루게 된다. 이러한 파장들은 각각의 색을 띄게 되는데, 파장이 길어 굴절률이 작은 빨간색에서부터 짧은 파장의 굴절률이 큰 보라색까지 연속된 스펙트럼 현상을 확인할 수가 있다. 이는 공기에서 출발한 빛이 유리로 이루어진 프리즘을 통과하는 서로 다른 매질의 경계면에서 굴절이 이루어진다는 스넬의 법칙에 의해서 발생하게 된다. 프리즘 실험에서는 그림 3의 (a)와 같이 양면에서 두 번의 굴절이 이루어지게 되는데, 각각의 굴절률이 두 번의 굴절에 의해서 프리즘에 첫 번째 입사하여 이루어진 굴절보다 더 많은 굴절이 이루어져, 각 파장에 따른 색상들이 눈으로 확인할 수 있을 만큼 연속적으로 보여지게 된다. 이 실험은 분산현상 외에도 백색광이 파장으로 이루어졌다는 것을 확인할 수 있

다[2].

한번 분산된 빛은 그림 3의 (b)와 같이 반대 방향의 프리즘을 다시 통과하게 되면, 나누어졌던 파장들은 다시 합쳐져 원래 흰색의 백색광으로 돌아간다는 것이다. 이것은 하나였던 파장을 나눈 상태에서 다시 또 하나로 합쳤을 때 백색광이 된다는 것을 확인할 수가 있게 되는데, 이는 분산된 빛은 각각의 파장이 합쳐지면 다시 백색광으로 돌아간다는 빛의 특성을 확인할 수 있다. 이 실험은 빛은 굴절률이 이루어졌을 시에 파장의 길이에 따라서 각각의 굴절률로 인하여 분산이 이루어진다는 것을 알 수 있다.



[그림 3] 프리즘 실험 (a)빛의 분산, (b) 빛의 합산[3]

3. 분산표현법(이전연구)

빛의 분산 현상을 표현하는 연구들은 광선추적법을 기반으로 발전하였다[4]. 그러한 연구들로는 분산광선추적법[5], 패스트레이싱[6], 포톤 매핑[7-10]등 있다. 본 연구의 목적은 빛의 분산현상을 실시간으로 표현하기 위한 포톤 매핑 기반 연구를 제안한다. 이를 위해 실시간 표현기술의 하나인 3색 분산기법을 알아보고 그 해결책을 제시하고자 한다.

3.1 3색 분산

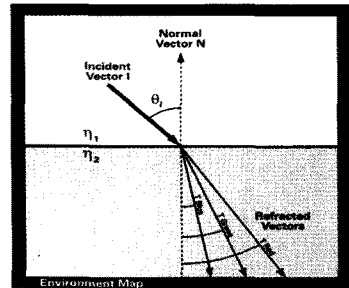


[그림 4] 프리즘 실험 (a)빛의 분산, (b) 빛의 합산[11]

빛을 프리즘에 통과 시켰을 때와 같이 시선이 면을 통과할 때 분산이 이루어진 것과 같이 보여진다. 일반적으로 굴절체를 눈으로 봤을 때는 시야각이 굴절되어 굴절된 환경을

눈으로 볼 수 있게 되는데, 빛을 통과할 때 이외에도 눈으로 바라보았을 때도 분산이 이루어지게 된다. 이는 색을 인식하는 과정에서 이유를 찾을 수가 있다. 눈이 색을 인식하는 방법은 빛을 받은 물체가 빛 안의 파장을 흡수하고 남은 파장을 반사하여 이를 눈에서 색으로 인식하는 것이다. 이때 그림 4와 같이 반사되어 눈에 들어오는 파장이 프리즘을 통과하게 될 때 굴절이 이루어져, 물체의 테두리에 여러 개의 테두리가 생기는 것으로 분산을 확인할 수 있다. 실제로 시선이 면을 통해 이루어지는 분산은 빛을 받은 물체가 반사하는 파장이 프리즘을 통과했을 때 파장의 길이에 따라 틀리게 발생하는 굴절에 의하여 눈에서 합산된 색상을 보게 되는 것이다.

이는 그림 5에서처럼 입사하는 벡터 I 가 눈에서 출발한 시선이 되고 이것이 스넬의 법칙에 의하여 굴절이 이루어지게 된다. 이때 물체의 경계면에서 나누어질 색상으로는 Red, Green, Blue의 3가지 색상으로 이루어지며 이때 3색상은 서로 다른 굴절률을 가지고 있다. 눈에서 출발한 시선은 매질의 경계면에서 3색상(R, G, B)의 각각의 굴절률을 가진 굴절벡터로 생성이 된다. 생성된 굴절벡터를 통해서 환경맵의 색상을 가져와 조합을 하는 것으로, 그림 5와 같이 면을 통한 분산표현을 할 수 있다.



[그림 5] 3-Color dispersion 원리[11]

이 방법은 실시간 렌더링을 위해서 반사되어 눈으로 들어오는 광선을 RGB 색상으로 제한을 두어 해당하는 색상의 비율을 혼합하는 것으로 분산을 표현한다. 하지만, 3색을 통한 분산이기에 실제의 스펙트럼을 표현하기에는 부자연스러운 표현을 하게 되며, 실제 프리즘 실험과 같이 빛이 프리즘을 통과했을 때 벽면에 맺히게 되는 표현이 불가능하다.

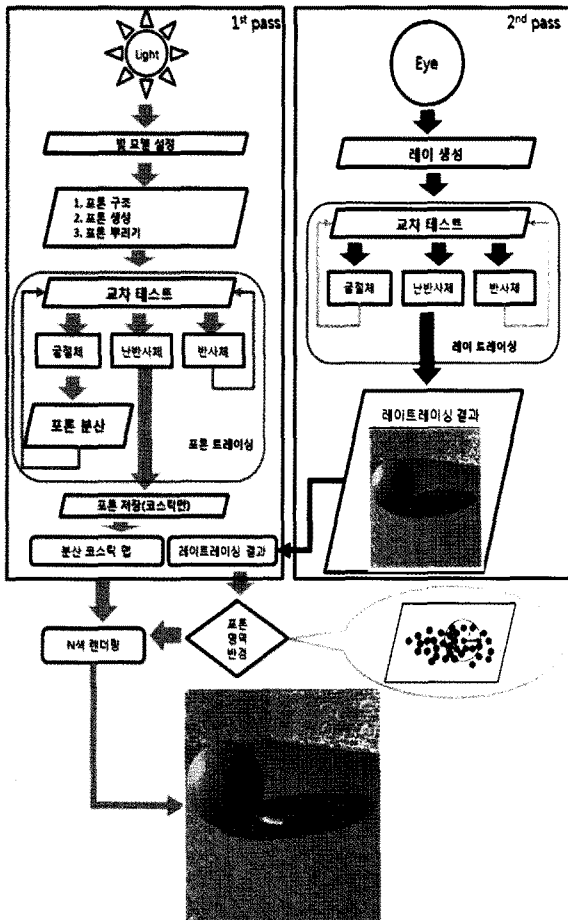
4. N색 분산 포톤매핑

본 논문에서 구현한 분산 포톤 매핑 시스템은 그림 6과 같다. 기본적인 레이트레이싱과 기존의 포톤매핑의 방법을 변형한 이 시스템은 프리즘실험과 같은 환경을 구성하기 위해서, 빛의 모델 중 스팟 라이트(Spot light)를 사용하였다. 포톤의 구조는 분산 포톤의 매핑에서 필요한 구조로 변경을

하였고, 스팟 라이트의 특징에 맞게 포톤들을 러시아 룰렛 (Russian roulette) 방법처럼 랜덤하게 뿌렸다. 뿌려진 포톤들을 추적하여 교차 테스트 시에 굴절체를 통과할 경우에만 분산이 이루어짐으로 굴절체를 통과할 시에만 분산이 이루어진다. 백색 광원의 구성은 3색과 7색에 따라 분산되어지는 정도가 틀러지며, 이를 통한 코스틱 현상인 포톤들만을 저장해 분산 코스틱 맵(Dispersion caustic map)으로 저장해 둔다.

레이트레이싱된 결과와 분산 코스틱 맵을 통해 바닥에 뿌려진 분산 포톤들의 밀도를 계산하여 분산 포톤 맵을 생성하게 된다. 생성된 분산 포톤맵과 레이트레이싱의 결과를 혼합하면 빛의 분산현상을 표현 할 수 있다.

로 정의하였다. 백색광이라 칭해지는 포톤들은 굴절체를 통과하게 되면 분산이 이루어지게 된다. 최초의 백색광 형태인 포톤은 분산이 이루어져 분산 포톤 맵에 저장되어 되어진다. 분산 포톤 맵은 분산된 포톤들만을 표현하기에, 포톤 맵에는 분산된 3색상이 저장되어 있다. 기본적으로 사용되는 레이트레이싱의 결과에서 포톤의 밀도를 계산하게 된다. 평면에 분산되어 존재하는 포톤들은 3색상으로 전체 포톤의 밀도와 3색상의 밀도를 표 2의 방법과 같이 수식 1을 이용하여 분산된 포톤들을 표현하였다.



[그림 6] 분산 포톤 맵 시스템 구성도

[표 1] 색상별 파장과 굴절률

명칭	Color	굴절률
R		1.515
G		1.550
B		1.564

[표 2] 3색 분산 포톤 렌더링 방법

설명
P : 레이트레이싱의 히트 포인트
r : 포톤밀도 영역의 거리
Total : 영역내 전체 포톤의 수
R_Photon : 영역내의 Red 포톤의 수
G_Photon : 영역내의 Green 포톤의 수
B_Photon : 영역내의 Blue 포톤의 수

진행 순서

1. P의 포인트를 기준으로 하여 r의 거리 만큼의 포톤의 수를 가져온다.
2. 전체 포톤의 수는 P에 해당하는 포톤의 밀도이다.
3. 영역내의 포톤의 색상들의 비율을 (수식 1) 처럼 계산을 한다.
4. 0.0~1.0 사이의 값으로 정규화하여 P의 해당하는 포톤 색상을 정의한다.

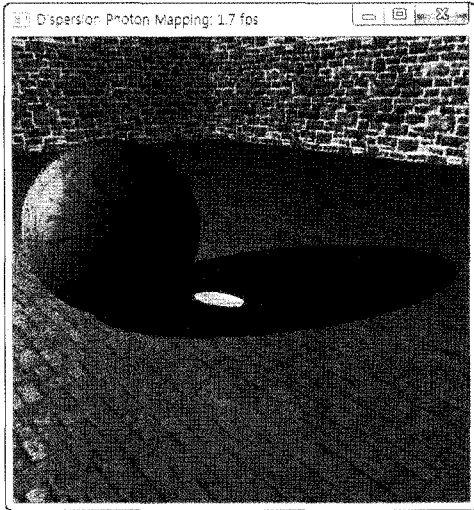
$$\begin{aligned}
 R &= R_Photon / Total \\
 G &= G_Photon / Total \\
 B &= B_Photon / Total
 \end{aligned}
 \tag{수식 1}$$

4.1 3색 분산 포톤 맵

분산시의 색상과 굴절률은 표 1의 빨강, 초록, 파랑의 3색으

로 3색을 통한 포톤 분산 맵핑으로 렌더링한 결과는 그림 7과 같다. 결과를 확인해 보면 굴절체인 구를 통과한 포톤들은 3색을 나누어져 굴절이 이루어지고 바닥에 뿌려주었다. 결과 화면을 보면 테두리에 빨간색과 파란색의 띠가 둘러진

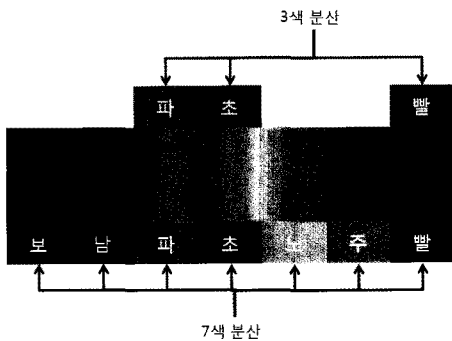
것을 볼 수가 있다. 3색을 통한 분산이 이루어졌지만, 3가지 색상의 분산으로 인해서 빨강색과 파랑색 중간의 색상들이 보여지지 않고 있다. 실제로 분산이 이루어졌다 하면 가시광선 영역처럼 다른 색상들이 순차적으로 보여져야 할 것이다. 이에 문제를 해결하고자 백색광을 이루고 있는 색상을 3가지 색상에서 무지개 색상에 근거한 7가지 색상으로 구성하여 렌더링하는 방법을 이용했다.



[그림 7] 3색 분산 포톤 매핑 결과

4.2 7색 분산 포톤 매핑

백색광의 구성을 R, G, B의 3색으로 구성된 3색 분산 방법은 그림 8과 같이 빛의 가시광선 영역을 대입해 보면 색상이 고르지 못하게 분산되어 있다는 것을 알 수가 있다. 이는 빛을 분산시켰을 때 나타나는 연속스펙트럼 현상에 대해서 표현을 하는 데에 부족하다는 것을 뜻한다. 이를 표현하기 위해서 가시광선 영역을 한정된 색상을 가지고 그림 8과 같이 나눌 수가 있다.



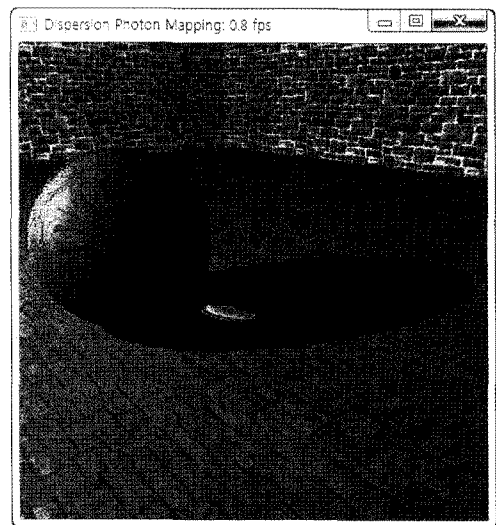
[그림 8] 3색과 7색 구성의 가시광선영역 표현

가시광선 영역을 무지개의 7가지 색상으로 나누어, 그 색

상들이 백색광을 구성하고 있다고 정의하여 7색 분산을 하였다. 색상을 표현하는 RGB 모델은 각 색상의 최대값을 255로 하여 3가지 색을 혼합하여 색을 나타내게 된다. 하얀색의 경우 가산혼합에 의해서 3색상이 모두 혼합했을 때 (255, 255, 255)로 나타내게 된다. 실제로 흰색은 모든 파장을 반사해 우리 눈이 흰색으로 인식하는 것이다. 그리하여 가중치가 (1,1,1)이 되었을 때 흰색이 된다고 가정하여 표 3과 같이 7색을 선정하여 그 가중치를 정하였다.

[표 3] 7색 분산의 굴절률과 가중치[11]

	Color	굴절률 (1.515~ 1.582)	RGB COLOR	가중치		
				C_{rr}	C_{rg}	C_{rb}
C_1		1.515	(255, 0, 0)	0.263 +0.002	0	0
C_2		1.536	(255, 165, 0)	0.263	0.244	0
C_3		1.547	(255, 255, 0)	0.263	0.377	0
C_4		1.550	(0, 255, 0)	0	0.377 +0.002	0
C_5		1.564	(0, 0, 255)	0	0	0.497 +0.001
C_6		1.578	(75, 0, 130)	0.077	0	0.253
C_7		1.582	(128, 0, 128)	0.132	0	0.249
		합계	(968,675,513)	0.998	0.998	0.999
				1	1	1



[그림 9] 7색 분산 포톤 매핑 결과

그림 9와 같이 색상을 표현 하는 방법은 표 4와 같다. 포

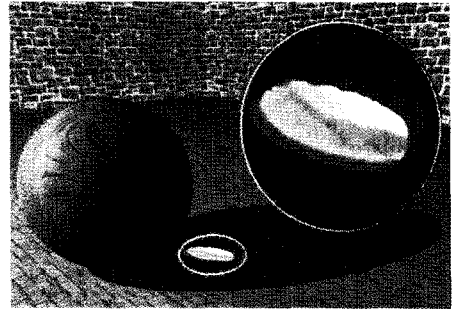
톤은 RGB 컬러모델로 이루어지며 표 3의 색상별 가중치를 사용한다. 먼저 포톤의 개수는 영역반경내의 전체 포톤의 수($Total$), 각 영역반경 내의 색상별 포톤의 수(T_i)를 가져온다. 기준점(P)의 영역반경(r)안의 포톤들의 색상에 대한 가중치(C_{ir}, C_{ig}, C_{ib})의 각 성분끼리 합산한다. 포톤들의 색상에 대한 합계 가중치인 $W(W_r, W_g, W_b)$ 는 $Total$ 로 나누는 것으로 포톤 밀도에 의한 P의 색상을 정하게 된다.

[표 4] 7색 분산 포톤 렌더링 방법

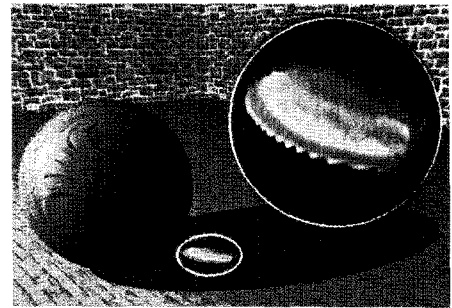
$$\begin{aligned}
 &R, G, B = \text{분산 포톤 밀도에 따른 색상} \\
 &R = W_r / Total \\
 &G = W_g / Total \\
 &B = W_b / Total \\
 &W_r = \sum_{i=1}^7 (C_{ir} \times T_i) \\
 &W_g = \sum_{i=1}^7 (C_{ig} \times T_i) \\
 &W_b = \sum_{i=1}^7 (C_{ib} \times T_i) \\
 &T_i = \text{영역반경내에 존재하는 } C_i \text{의 포톤 개수.} \\
 &Total = \sum_{i=1}^7 T_i
 \end{aligned}$$

추적할 광선의 수가 많아져 제안된 방법의 렌더링 시간 비용이 증가함을 알 수 있다. 제안된 7색분산 방법은 아직까지 실시간 렌더링에 적용하기 어렵다. 이를 해결하기 위해, 향후 최적화 방법 및 GPU 프로그래밍의 개선을 통해 실시간 처리를 연구할 예정이다.

그림 10은 포톤의 개수 20,000개, 영역반경 값을 1.0으로 하여 포톤 매핑한 결과를 보여주고 있다. 3색 분산에 비해서 7색 분산이 스펙트럼의 연속적인 색상을 자세히 표현한다.



3색 분산



7색 분산

[그림 10] 3색 분산 과 7색 분산 비교

5. 구현결과

본 논문에서는 분산현상을 통한 빛의 스펙트럼을 표현하기 위해 OpenGL과 CUDA를 이용하여 구현한 레이트레이싱을 기본으로 하였다. 구현 환경으로는 Intel Quad core 2.33GHz, Nvidia GTX 260에서 400 x 400의 스크린 사이즈로 하였다.

[표 5] 포톤수 및 렌더링 방법별 렌더링 시간 비교

실험1	포톤매핑(굴절), 영역반경(1.0), 텍스춰미적용			
포톤수	5,000	10,000	15,000	20,000
FPS	11.1	5.7	4.3	3.3
실험2	포톤수(20,000개), 영역반경(1.0), 텍스춰적용			
렌더링 방법	광선추적법	포톤매핑(굴절)	포톤매핑(3색분산)	포톤매핑(7색분산)
FPS	5.0	4.6	1.7	0.8

표 5-실험1은 포톤의 수에 따른 렌더링 시간을 비교하고 있다. 포톤의 수는 많을수록 렌더링 비용은 증가하나 넓고 고르게 퍼져, 포톤이 뿌러지지 않은 구멍 난 영역이 생기지 않는다. 표 5-실험2는 렌더링 방법별 시간을 비교하였다.

6. 결론 및 향후연구

본 논문은 빛의 특성 중 굴절을 통한 분산현상으로 인해 나타나는 스펙트럼을 표현하는 것을 목표로 하였다. 실제 백색광을 이루고 있는 파장의 굴절을 통해서 분산현상이 나타나는 원리를 이용하기 위해 백색광을 RGB 컬러모델의 3색상과 무지개의 대표적인 색상인 7색상으로 구성 하였다. 연속 스펙트럼이라 하는 가시광선 영역의 파장을 연속적으로 표현하는 것으로, 현재의 단색광으로 구성되어진 조명모델을 여러 색상으로 구성되어진 백색광으로 정의하여, 빛에 대한 현실적인 표현에 한발 더 나아갔다.

본 논문에서 구현 결과를 통해 설정한 적정 영역 반경(r)은 0.5, 포톤의 개수는 20,000개 이상일 때에 좋은 결과를 보여주었다. 포톤의 개수와 함께 백색광을 구성하고 있는

색상의 수와 포톤의 수를 늘릴수록 스펙트럼의 표현을 더욱 자세히 할 수 있을 것으로 보인다. 하지만 이것은 속도의 저하로 인해 실용성이 떨어진다는 문제점을 제기할 수 있다. 이를 해결하기 위해 Kd-tree를 통한 공간분할과 속도의 최적화로 렌더링 시간을 줄이고자 한다. 이를 통해 대표적인 분산 현상인 무지개를 표현하는데도 적용할 수 있을 것으로 보인다. 무지개에서의 분산 원리는 프리즘 실험에서의 분산 원리와 기본적으로 같기에 실내에서의 포톤 분산 맵핑이 아닌 실외에서의 무지개 표현을 하여, 빛의 분산 표현에 있어서 범용적인 시스템을 구축하여, 사실적인 빛에 대한 표현을 할 수 있을 것으로 보인다.

감사의 글

이 논문은 2010년도 한신대학교 학술연구비 지원에 의하여 연구되었음.

참고 문헌

- [1] 한국광학회 대구경북지회, "광학의 기초", *두양사*, 2003.
- [2] I. Newton, "Opticks", *Royal Society*, London, 1704
- [3] The university of toledo, "The discrete colors of light", <http://astro1.panet.utoledo.edu/~ljc/light3a.html>.
- [4] Alexander Wilkie, Robert F. Tobler, Werner Purgathofer, "Raytracing of Dispersion Effects in Transparent Materials", *WSCG 2000*, 2000.
- [5] Veach, E., and Guibas, L. J., "Metropolis light transport", *SIGGRAPH'97*, pp. 65-76, August 1997
- [6] E. LaFortune, Y. Willems, Bi-directional path tracing, *Proceedings of Compugraphics '93, Alvor, Portugal (December '93)*, pp. 145-153
- [7] Henrik Wann Jensen, "Realistic image synthesis using photon mapping", *A, K, Peters, Ltd*, Natick, MA, 2001.
- [8] B. Fabianowski and J. Dingliana, "Interactive Global Photon Mapping", *Eurographics Symposium on Rendering*. Vol.28. No.4. 2009.
- [9] Johannes Günther, Ingo Waldy, and Philipp Slusallek. "Realtime Caustics Using Distributed Photon Mapping". *Eurographics Symposium on Rendering*, 2004.
- [10] Tin-Tin Yu, John Lowther, Ching-Kuang Shene. "Photon mapping made easy", *ACM SIGCSE Bulletin*. Vol.37, No.1, 2005.
- [11] Matthias Wloka, "Fresnel Reflection", *NVIDIA Corporation*, 2002.
- [12] A. Kwan, J. Dudley and E. Lantz, "Who really discovered Snell's law?", *Physics World 15 (4): 64*, 2002.
- [13] H. D. Young, "University Physics 8e", *Addison-Wesley*,

Chapter 35, 1992

- [14] Philip Laven, "Refractive index", <http://www.philip-laven.com/p20.html>.
- [15] Tomas Möller, Ben Trumbore. "Fast, minimum storage ray-triangle intersection". *Journal of Graphics Tools*. Vol.2. No.1. p.21-28. 1997.
- [16] hypertextbook, "Index of Refraction of Glass, Crown", http://hypertextbook.com/facts/2005/Justin_Che.shtml, 2005
- [17] Molecular Expressions, "Light and Color", <http://micro-magnet.fsu.edu/optics/lightandcolor/sources.html>.
- [18] Raymond L. Lee Jr. "Mie theory, Airy theory, and the natural rainbow", *Applied Optics*. Vol.37. No.9. p.1506-1520. 1998.
- [19] Computer Graphics-Alexandra Institute, "Triers CUDA ray tracing tutorial", <http://cg.alexandra.dk/> 2009/08/10/triers-cuda -ray-tracing-tutorial, 2009

〈저자 소개〉



곽용식

- 2007년 한신대학교 소프트웨어학과 (이학사)
- 2010년 한신대학교 컴퓨터공학과 컴퓨터그래픽스 전공 (이학석사)
- 관심분야 : 실시간렌더링, 비사실적렌더링, 지형렌더링



류승택

- 1996년 중앙대학교 전자계산학과(공학사)
- 1998년 중앙대학교 컴퓨터공학과(공학석사)
- 2002년 중앙대학교 영상공학과 컴퓨터 그래픽스전공 (공학박사)
- 현재 한신대학교 컴퓨터공학부 부교수
- 관심분야 : 비사실적 렌더링, 실시간렌더링, 영상기반 렌더링