

인크리멘탈 컨벡스 헐 알고리즘을 이용한 래디오시티 렌더링 시스템 구현

김종용*, 최성희, 송주환, 최임석, 권오봉
전북대학교 컴퓨터과학과

Radiosity Rendering System Implementation using Incremental Convex Hull Algorithm

J. Y. Kim*, S. H. Choi, J. W. Song, Y. S. Choe, O. B. Gwon
Dept. of Computer Science, Chonbuk National University

요 약

래디오시티 렌더링 시스템에 적합한 모델링을 하는데 있어서 수작업을 비롯한 기존의 여러 방법을 적용해 본 결과, 제작하고자 하는 환경 설정에 많은 어려움을 겪었다. 본 논문은 이러한 문제를 해결하기 위해, 인크리멘탈 컨벡스 헐 모델링 기법을 응용하여 시스템을 개발하였다. 이 시스템은 기본 데이터 정보를 입력하면 자동적으로 패치와 알리먼트가 생성되는 프로그램으로, 물체가 많은 환경이나 곡면을 비교적 쉽고, 고속으로 표현할 수 있었다. 그리고 인크리멘탈 컨벡스 헐 알고리즘으로 생성된 환경에, 폼 팩터의 대표적인 세 가지 알고리즘들을 비교·평가 해본 결과 Cubic-Tetrahedron 알고리즘이 가장 효율적이었음을 알 수 있었다.

1. 서 론

래디오시티 알고리즘은 직사광선이 미치지 않는 부분도 간접광에 의해 빛이 나게 하고, 반 그림자를 생성하고, 반사면의 색이 인접하는 면에 영향을 주는 칼라 브리딩 효과를 연출할 수 있어 실내 공간의 표현에 적합하다. 복잡한 실내 공간을 시뮬레이션 할 때, 기존의 방법으로는 3차원 데이터를 생성하는 것이 어려웠으나, 인크리멘탈 컨벡스 헐 알고리즘(Incremental Convex Hull Algorithm)을 응용해 점들의 정보만 알고 있으면 패치와 알리먼트를 자동 생성해 주는 프로그램을 개발했다. 이 프로그램을 이용해 고속으로 가상 공간 환경을 구현할 수 있었다.

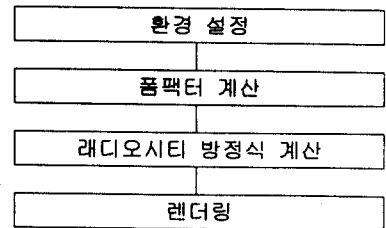
본 논문에서는 인크리멘탈 컨벡스 헐 알고리즘을 응용한 래디오시티 렌더링 시스템의 구현에 관하여 논한다. 논문의 구성은 2 장에서 고속의 데이터 생성을 위한 인크리멘탈 컨벡스 헐 알고리즘과 시스템 구축에 기반이 되는 래디오시티 알고리즘에 대하여 고찰한다. 3 장에서는 구현 과정과 평가 데이터에 대하여 실험한 결과를 보이고, 4 장에서 결론을 내린다.

2. 래디오시티(Radiosity) 알고리즘

래디오시티 알고리즘은 열역학의 열전도 이론을 이용하여 장면 속의 각 물체 표면의 빛의 세기(Light intensity)를 구하는 것이 핵심이다. 장면을 구성하는 물체는 여러 개의 구성요소(패치)로 나누어지고 이 패치는 서로 에너지를 주고받는다. 광원에서 나온 빛은 몇 번인가 반사된 후 각 패치의 밝기인 빛의

세기로 안정된다. 이 패치의 빛의 세기(래디오시티)는 래디오시티 방정식을 풀어서 구한다. 래디오시티 방정식을 세우는데 가장 중요한 것은 폼팩터(Form Factor)인데, 폼팩터 F_{ij} 는 패치 A_j 의 한 면에서 나온 에너지 중 패치 A_i 로 들어가는 에너지의 비율을 표시한다.

본 시스템의 처리 과정은 [그림 2-1]과 같다. 환경 설정에서는 물체, 시점, 광원을 설정하고 장면을 구성한다. 폼팩터 계산에서는 각 물체의 폼팩터를 계산하고, 래디오시티 방정식 계산에서는 위의 폼팩터를 이용하여 각 패치의 래디오시티를 계산한다. 렌더링에서는 이미 구한 Z 버퍼 알고리즘과 위에서 구한 래디오시티 값을 이용하여 각 물체의 음영을 계산한다.



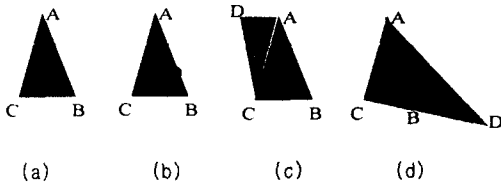
[그림 2-1] 시스템의 처리과정

2.1 환경설정

시스템 처리과정의 첫 단계로, 다음 단계에서 사용할 데이터 구조, 광원, 물체를 정의하고 밀폐된 공간에 배치한다. 이러한 처리를 위해서 필요한 기하변환, 관측변환, 폴리곤 쿨리핑연산, 은연소거 등의 연산을 한다. 물체는 인크리멘탈 컨벡스 힐 알고리즘을 응용하였고, 각 물체는 랑베르트 난반사 법칙 (Lambert's Law)을 따른다.

2.2 인크리멘탈 컨벡스 힐 알고리즘

Joseph O'Rourke에 의해 개발된 인크리멘탈 컨벡스 힐 알고리즘(Incremental Convex Hull Algorithm)은 순서대로 입력된 점들의 정보를 가지고 삼각형 기법을 이용해 물체의 표면을 구성한다. [그림 2-2]에서 이 알고리즘에 대해 설명한다. (a)에서와 같이 세 점으로 삼각형(컨벡스 힐)을 구성하고, 다음 들어온 점 D가 이 삼각형에 안에 있는지, 밖에 있는지 체크를 해서 (b)와 같이 안에 있다면 점 D는 무시되고, (c)와 같이 밖에 있다면 컨벡스 힐에 추가가 되며, (d)에서는 점 B가 무시된다. 체크되는 점들은 데크(Deque)에 의해 입·출력된다.



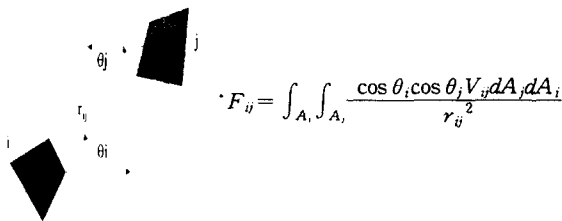
[그림 2-2] 컨벡스 힐 디텍션(Convex hull detection)

2차원 알고리즘을 이용하여 3차원으로 확장을 시켰을 때, 기분이 되는 사면체(Tetrahedron)를 이용하여 새로 들어온 점이 또다른 사면체를 구성하는지, 못 하는지에 따라서 그 점을 체크한다.

총 수행시간은 Faces = $O(n)$ 과 Edges = $O(n)$ 의 반복 루프에 의해 $O(n^2)$ 이 된다.

2.3 폼팩터 계산

폼팩터 F_{ij} 는 Patch i 에서 나온 에너지중 Patch j 에 도달한 에너지의 비로 [그림 3-2]의 식과 같이 이중 적분으로 정의된다. 그러나 이러한 적분은 계산이 어렵기 때문에 근사적인 방법을 이용한다. 폼팩터는 원래 패치간에 정의되나 근사해에서는 A 패치 내의 한 점과 B 패치와의 점 폼팩터를 구하여 A 패치의 면적을 곱하여 구한다.



[그림 3-2] Form Factor

폼팩터를 구하는 방법은 Hemicube, Cubic Tetrahedron, Ray Casting의 3가지 방법이 있다. Hemicube 방법은 반구를 육면체로 근사하는 방법인데 패치의 중심에 육면체를 놓고 육면체의 각 면을 픽셀로 분할해 미리 픽셀의 폼팩터를 계산하여 투영된 면적에 픽셀의 폼팩터를 곱해 구한다. Cubic Tetrahedron 방법은 육면체 대신에 사면체를 사용하는 방법이다. Ray Casting 방법은 몬테카르로 방법의 일종으로 A 패치에 반구를 덮고 반구의 밑면에서 위로 광선을 방사시켜 반구에서 법선 방향으로 향하도록 한다. A 패치에서 방사된 광선중 일부가 패치 B에 도달하였다면 이의 비를 폼팩터로 한다. 본 논문에서는 위의 3가지 방법을 서로 비교한다.

2.4 래디오시티 방정식 계산

Form Factor 계산 후 이를 다음의 식에 넣어 모든 패치의 래디오시티를 가우스-자이델 법을 이용하여 계산한다.

$$B_j = E_j + \rho_j \sum_{i=0}^N B_i F_{ij}$$

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & \dots & \dots & \dots \\ \vdots & \vdots & \dots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n1} & \dots & -\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

위에서 B_j 는 j 면에서의 래디오시티

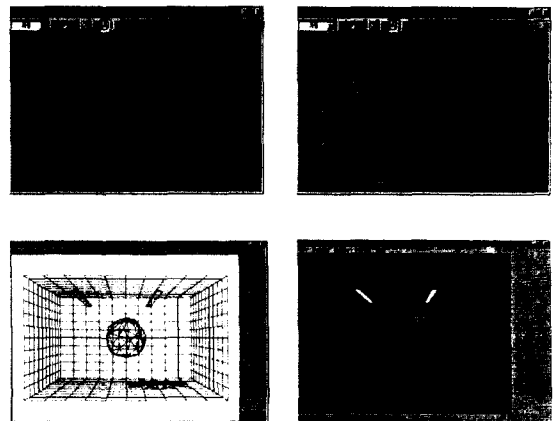
E_j 는 j 면에서의 방사 에너지

ρ_j 는 j 면에서의 반사율

F_{ij} 는 i 에서 j 면으로의 빛 에너지줄 의미한다.

3. 구현 및 실험 결과

본 시스템을 IBM PC Pentium II 333MHz의 플랫폼에서 Visual C++ 이용하여 구현하였다. 각각의 데이터는 컨벡스 힐 알고리즘을 사용하여 점들의 정보를 읽어서 자동 생성되도록 하였다. [그림 3-1]은 이 알고리즘을 이용하여 구 모양의 물체를 직접 생성해 곡면 생성시에도 고속의 데이터를 얻을 수 있었다.



[그림 3-1] 인크리멘탈 컨벡스 힐 알고리즘을 이용한 구 생성

가상 공간을 구성하는 데이터(침대, 책상, 의자, 형광등 등)는 인크리멘탈 컨텍스 할 알고리즘을 이용하여 모델링 하였고, 계층은 Instance → Surface → Patch → Element → Vertex 순으로 정의되어서 폴리팩터 계산에 이용된다. 평가 데이터는 [표 3-1]과 같다.

구분	개수	Vertex (개수)	Patch (개수)	Element (개수)	
침대	1	99	26	42	
책상	1	100	28	50	
의자	1	126	34	50	
액자	1	16	16	32	
벽, 천장	6	121	50	100	
전체 조명	2	36	28	56	
부분 조명	침대	1	52	18	34
	책상	1	45	20	40

[표 3-1] 평가 데이터

이 데이터를 이용하여 Cubic tetrahedron, Hemicube, Ray casting의 3가지 폴리팩터 구하는 알고리즘을 평가하였다. 평가를 위해서 와이어프레임, 푼 세이딩, 렌더링 3가지의 영상을 생성하였다.

폴리팩터 별 실행 시간 결과는 [표 3-2]와 같다. 이것으로 Cubic Tetrahedron 알고리즘이 가장 빠르다는 것을 알 수 있었다.

Form-Factor 알고리즘	실행시간	비율
Cubic Tetrahedron	15초	0.75
Hemicube	20초	1.00
Ray Casting	12분 20초	0.37

[표 3-2] 알고리즘별 실행시간 비교

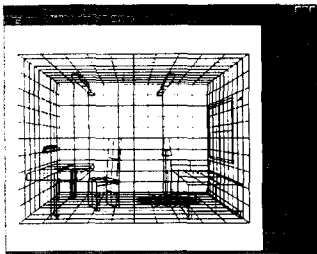
[그림 3-2]은 데이터를 읽어들이어서 와이어프레임 형식으로 표시한 것이고, [그림 3-3]은 [그림 3-1]을 푼 세이딩한 결과이며, [그림 3-4], [그림 3-5], [그림 3-6]은 각각 Cubic Tetrahedron 알고리즘, Hemicube 알고리즘과 Ray casting 알고리즘으로 렌더링하고, 전체 조명을 준 결과이다. [그림 3-7]은 부분 조명이 침대에 주는 영향을 조사하기 위해 책상 위의 스탠드에만 조명을 켜줄 경우이다. 이것은 부분 조명이 전체 조명보다 안정감을 주는 것을 보이고 있다.

4. 결론 및 향후 연구과제

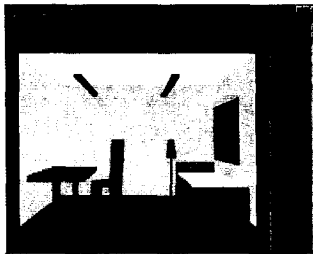
본 논문은 환경설정을 위한 데이터 모델링 기법으로 인크리멘탈 컨텍스 할 알고리즘을 응용하였으며, 많은 데이터와 곡면을 고속으로 쉽게 모델링 할 수 있었다. 래디오시티 렌더링 계산시 Cubic Tetrahedron 알고리즘을 이용한 폴리팩터 계산이 가장 빨랐으며, 자연스러운 분위기를 연출하였다. 향후 과제로는 다양한 재질을 표현할 수 있는 텍스처 매핑에 관하여 연구할 예정이다.

5. 참고 문헌

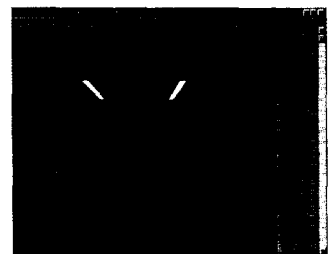
- [1] "Modeling The Interaction of Light Between Diffuse Surface", Computer Graphics Vol. 18, No. 3, pp. 213-222, July 1984
- [2] "SIGGRAPH 1993 Education Slide Set", by Stephen Spenceer.
- [3] "3D Computer Graphics", Alan Watt, 1993, pp. 331-350.
- [4] "조명공학", 지철근, 문운당, 1998.01



[그림 3-2] 와이어 프레임



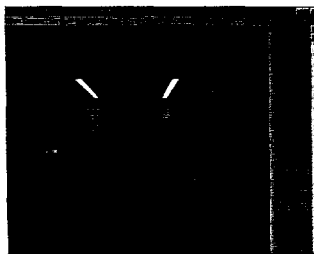
[그림 3-3] 세이딩



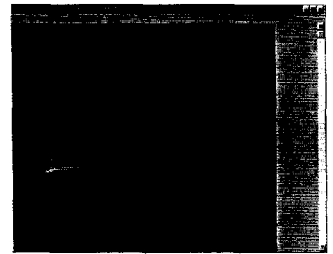
[그림 3-4] 렌더링
(Cubic tetrahedron)



[그림 3-5] 렌더링
(Hemicube)



[그림 3-6] 렌더링
(Ray-Casting)



[그림 3-7] 부분조명