

Bounding Box의 Inside Test를 간단화시킨 格子形空間分割을 이용한 立體圖形의 表現

(The Representation of 3-D Objects Using the Lattice-Structured Space Subdivision for the Simplification of the Inside Test in the Bounding Box)

金 英 一*, 趙 東 旭**, 崔 炳 旭**

(Young Il Kim, Dong Uk Cho and Byung Uk Choi)

要 約

本 論文은 Computer Graphics 의 목적인 사실적인 묘사를 하는데 있어서 가장 훌륭한 Rendering 方法 이라고 알려진 광선추적법의 가장 큰 단점인 방대한 교점계산량을 줄이기 위하여 Bounding Volume 을 이용한 格子形空間分割方法을 제안한다.

本 方法의 有用性을 실험을 통하여 기존방법과 비교해 볼때 計算量을 50%~70% 정도 줄일수 있음을 보인다.

Abstract

This paper proposes the lattice-structured space subdivision method using bounding volume to reduce a great number of ray-surface intersection calculations in ray tracing algorithm for the computer graphics.

We show that this method reduced 50%-70% calculations compared to pre-exist method by experiments.

I. 序 論

초기에 Bar 사 chart, Pie chart, Histogram 같은 2 次元의인 表現에 한정되었던 Computer Graphic은

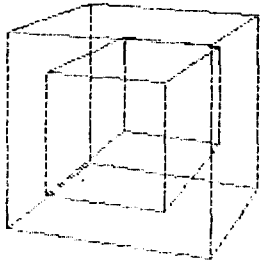
1970년대 말부터의 마이크로프로세서, 메모리의 급 격한 가격인하와 더불어 급격히 發展하기 시작했으 며, 요즘 거의 성숙단계에 접어 들었다. Computer Graphics 가 짧은 歷史에도 불구하고 급격히 발전하 게 된 것은 自動化, 情報化 社會의 대두로 인하여 그 必要性이 커진데 이유가 있다고 할수 있으며 그 應 用分野도 자동차나 항공기의 설계, 건축설계의 CAD, SF영화나 상업광고에 쓰이는 Animation, Flight Si- mulator 같은 Simulation 등 여러 分野로 확대되고 있 다.^{1,2,3,4)}

*正會員, 韓國電氣通信公社 事業支援團 (Research Center, KTA)

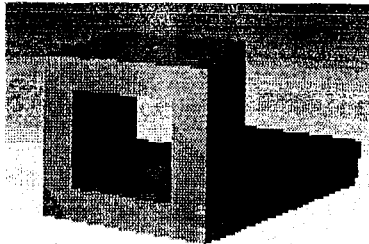
**正會員, 漢陽大學校 電子通信工學科 (Dept. of Elec. Communications, Eng., Hanyang Univ.)

接受日字: 1988年 2月 9日

3次元 物體를 2次元의 畫像에 나타낼때 그림1(a)와 같이 가려져서 보이지 않아야 되는 부분까지 나타나게되어 어떤 物體인지 識別이 어려워진다. 따라서 物體를 사실적으로 나타내려면 가려진 部分을 除去해야 하는 은선/은면제거(hidden line/surface removal)^[15]는 물론이고 명암, 그림자, 빛의 굴절, 반사까지도 포함시켜야만 物體의 사실적인 表現이 가능하다.



(a)



(b)

그림 1. (a) Wireframe 모델
 (b) 사실적 표현
 Fig. 1. (a) Wireframe model.
 (b) Realistic representation.

이러한 사실적인 表現을 위해서 광선추적법이 가장 훌륭한 方法이라고 알려졌으며 이 方法은 Global Illumination 범주에 속하는 Rendering技法으로써 1968年 Appel에 의해서 提案 되었으나 방대한 교점 계산량의 문제로 인하여 실용화되지 못하다가 1980年 Whitted에 의해서 改善되면서 활발한 研究가 진행중 이다.^[1,5]

本 論文에서는 광선추적법의 최대 결점인 방대한 交點 計算을 줄이기 위해서 Bounding Box의 Inside test를 간단화시킨 格子形空間分割을 提案한다. 分割된 各 空間에는 최소개의 物體 만을 포함하게 하

여 그 空間 外部에 있는 物體와의 불필요한 交點 計算을 줄임으로써 計算 時間을 단축 시킬 수 있음을 보인다

II. 물체 및 빛의 반사 모델

物體는 크게 두가지로 分類해서 꼭지점을 잇는 경계선(edge)으로 물체를 定義하는 Wireframe 모델과 어떤 면을 경계로 해서 한쪽 空間을 Half-Space로 定義하고 그 Half-Space들의 Intersection으로 물체를 정의하는 Solid 모델로 나눌수 있다.^[11] 本 論文에서의 物體는 Solid 모델로 정의하였으며 구(sphere), 원통(cylinder), 원뿔(cone), 육면체(cube) 등 4개의 기본물체(primitive objects)를 使用하였고 기본물체만으로 구성되지 않는 복잡한 물체는 기본물체의 크기와 위치를 정한 후 그들의 組合으로 定義하였다. 빛의 반사 모델에는 交點에서 物體의 Surface normal과 光源에서 직접 交點에 들어오는 빛만을 고려하는 Local Illumination 모델과 그림2와 같이 光源에서부터 직접 오는 빛뿐만 아니라 다른 物體에 의해서 반사, 굴절되어 오는 빛도 고려하는 Global Illumination 모델이 있다.^[1,2] 광선 추적법의 장점은 그림2와 같이 Global Illumination 모델을 利用하여 보다 사실적으로 物體를 表現할수 있다는 점이며 이때의 Global Illumination 모델은 (1)式과 같다.

$$I = I_a + I_p K_d \max(0, \vec{L} \cdot \vec{N}) + K_s S + K_t T \quad (1)$$

여기서 I_p 는 입사광의 밝기, I_a 는 포위광 상수, S 는 굴절방향의 밝기 T 는 반사 방향의 밝기, K_d, K_s, K_t 는 각각 난반사, 경면반사, 투과계수이다.

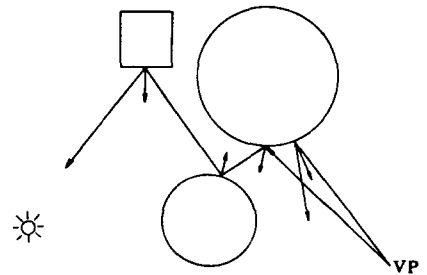


그림 2. Global illumination
 Fig. 2. Global illumination.

III. 格子形 空間分割

1. 광선추적법의 概要

광선추적법의 기본 원리는 光源에서 나온 빛이 어떤 경로를 거쳐서 視點에 도달하는가를 추적하는 것

이다. 이 경로에는 반사·굴절도 포함해서 추적해야 한다. 그러나 光源으로부터 광선을 추적하는 경우는 視點에 도달하는 광선이 극히 적으므로 視點에서부터 逆으로 추적한다.^[11] 그림 3과 같이 Window 内の 한 點을 지나는 直線(光線)을 (2)式과 같이 생성하여 이 直線과 物體와의 交點을 구한 다음 그 交點에서의 物體의 表面정보와 조명정보에 따라 物體의 명암, 색등을 決定하여 Window에 대응되는 Screen 상의 한 Pixel에 밝기를 부여하는 것이다.

$$\begin{aligned} X &= Ax't + V_x \\ Y &= Ay't + V_y \\ Z &= Az't + V_z \end{aligned} \quad (2)$$

(여기서 V_x, V_y, V_z 는 시점의 좌표, A'_x, A'_y, A'_z 는 시선 방향벡터)

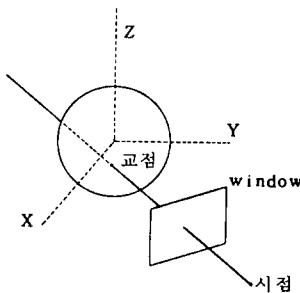


그림 3. 역추적
Fig. 3. Back tracing.

2. Bounding Box

Bounding Box는 그림 4(a)와 같이 物體의 최대 좌표점과 최소 좌표점이 box에 接하게 空間을 分割한 것이다. Bounding Box로 物體가 存在하는 영역을 나누었을 경우에는 Homogeneous Transform을 利用하여 光線이 z축이 되도록 變換하고 같은 Transform Matrix로 최대, 최소 좌표점을 變換하여 變換된 좌표점의 최대 x값과 최소 x값의 부호가 反對이면서 y값의 부호도 反對일 경우에 한하여 空間 内部의 物體와 交點 計算을 行한다. Bounding Box를 利用함으로써 物體와 직접 不必要한 交點計算은 減일수 있으나 모든 Bounding Box에 대하여 Inside Test를 遂行하여 光線이 지나가는 Box를 選擇하여야 한다.

3. 格子形空間分割

視點에서 Window의 한 點을 지나는 光線이 어떤 物

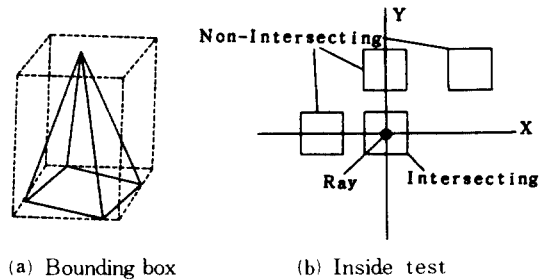


그림 4. Bounding box와 inside test
Fig. 4. Bounding box & inside test.

體를 향하는지 미리 알고 있지 못할 경우 定義한 모든 物體와 交點을 計算하는 과정과 交點이 여러 개 일때 視點에서 가장 가까운 交點을 찾는 과정이 필요하다.^[6,8] 광선추적법은 사실적인 物體 묘사를 위해서 가장 뛰어난 方法으로 알려져 있지만 이와 같이 計算量이 大대하다는 단점이 있다. 이러한 불필요한 交點 計算을 줄이기 위해서 Bounding Volume을 利用한 方法이 있으나^[14] Bounding Box로 空間을 分割했을 때 光線과 物體가 속한 bounding box^[11]와의 交點을 찾는 Inside Test 과정에서 많은 계산을 필요로 한다. 本 論文에서는 이러한 Inside Test를 간단히 하기 위해서 格子形空間分割 方法을 제안한다.

格子形空間分割 方法에서 空間分割 方法은 bounding Box와 같으나, 視點과 物體가 위치할 최대 범위로 world를 정하고 그 内部에서 그림 5(a)와 같이 空間을 나누어 格子 空間으로 할당한다. 이것을 X-Y 평면상에서 2次元으로 나타내면 그림 5(b)와 같다. 各 格子空間은 격자번호로 區別하고 격자번호는 다음과 같이 할당한다.^[13]

$$\begin{aligned} & \text{격자번호}(a, b, c) \\ X(a) & \leq X \leq X(a+1) \\ Y(b) & \leq Y \leq Y(b+1) \\ Z(c) & \leq Z \leq Z(c+1) \end{aligned}$$

(여기서 $X(), Y(), Z()$ 는 격자 좌표 list이며 격자로 나누어 지는 좌표값들의 array이다.)

여러 primitive로 構成된 복잡한 物體는 物體의 최대, 최소 좌표로 格子를 할당하고 그 격자 내에서 다시 分割하여 부격자를 만들어 各 空間에는 最小個의 物體만 포함하도록 하여 계층적인 자료구조를 갖는다. 다음은 광선 추적시 格子사이를 Traversing 하는 알고리즘이다.

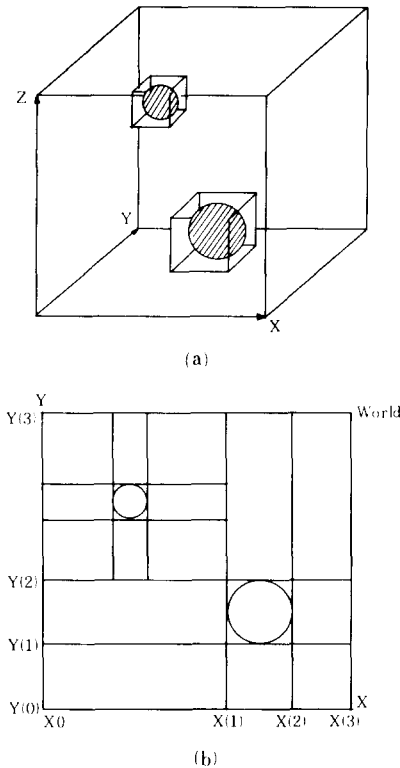


그림 5. (a) 격자형 공간분할(3 차원)
(b) 격자형 공간분할(2 차원)

Fig. 5. Lattice-structured space subdivision.
(a) Lattice-structured space subdivision (3Dimension).
(b) Lattice-structured space subdivision (2Dimension).

Step 1 : 시점의 lattice # 계산
 Step 2 : Source Ray 생성
 Step 3 : Stack 초기화
 Step 4 : current lattice # ← 시점의 lattice #
 Step 5 : IF object pointer(current lattice #) < nil,
 THEN GOTO Step 8
 Step 6 : current lattice # ← next lattice #
 Step 7 : IF next lattice # = world 의 외부,
 THEN GOTO Step13
 ELSE GOTO Step 5
 Step 8 : IF sub-lattice pointer(current lattice #)
 = nil,
 THEN IF 교점 존재,
 THEN GOTO Step13
 ELSE GOTO Step 6
 ELSE PUSH current lattice #

Step 9 : current lattice # ← sub-lattice #
 Step10 : IF object pointer(current lattice #) < nil
 THEN GOTO Step 8
 Step11 : current lattice # ← next sub-lattice #
 Step12 : IF next sub-lattice # = lattice 외부,
 THEN POP current lattice #
 GOTO Step 6
 ELSE GOTO Step10
 Step13 : 밝기 계산
 Step14 : GOTO Step 2

Step6,에서 다음 격자를 구하는 방법은 예를 들어 현재 격자번호가(a, b, c)이고 光線이 +x面을 通過해 나갈때 다음 격자번호는 (a+1, b, c)가 되므로 복잡한 Transform 없이 다음 격자의 판단(inside test)이 간단하게 된다. 따라서 Bounding Volume에서와 같이 모든 物體에 대해서 Inside Test를 할 필요가 없다. 위의 알고리즘은 Source Ray에 대한 광선추적으로, Global Illumination 모델을 適用하기 위해서는 step13의 밝기와 交點을 Root Node로 한 후 交點을 視點으로 대치하여 각 Step을 반복해서 Intersection Tree를 구성해야 한다. 최종적으로 한 Pixel에 Mapping되는 밝기는 Intersection Tree에서 각 Node 들의 밝기를 합하여 구한다.

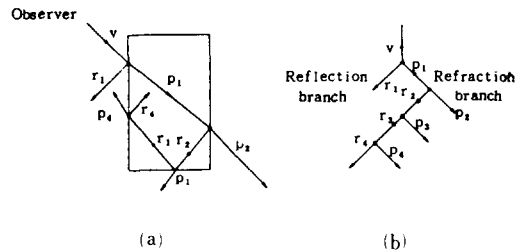


그림 6. Intersection tree의 예
 Fig. 6. Examples of intersection tree.

IV. 실험 및 고찰

本 實驗은 光源을 무한 거리에 있는 평면광으로 하고 물체 표면의 반사계수는 빛의 주파수에 無關하다는 가정하에서 IBM PC/AT 상에서 Turbo-pascal을 利用하여 구현하였다. 視點에서 Window에 대한 視角은 X축방향 40° Y축방향 25°로 고정시켰으며 실험 결과의 그림은 160×100의 해상도에서 8×8의 Bit Map Patterning⁹⁾을 利用하여 65개의 Gray Level

로 나타낸 것이다. 物體가 存在하는 格子 內에서만 그 物體와 交點 計算을 함으로 불 필요한 交點 計算을 피할 수 있었다. 그러나 光線이 物體가 存在하는 格子空間의 빈 부분을 통하여 다음 格子로 나가는 경우는 Bounding Volume에서와 같이 불 필요한 交點 計算 過程이 수행된다. 또한 under-sampling에 의하여 物體의 境界 部分에 심한 aliasing이 나타났다. 표 1 은 그림7, 8, 9, 10에 대하여 Bounding Box를 利用했을 때와 格子形空間分割을 利用했을 때의 處理時間을 비교한 것이다.

표 1에 나타난 바와 같이 格子形空間分割을 利用했을 때 既存의 Bounding Box에 비해서 處理時間이 約 1/2~1/3정도로 줄어 들었음을 알 수 있다.

표 1. 處理時間
Table 1. Processing time.

	그림 7	그림 8	그림 9	그림 10
Primitive갯수	7	17	13	2
* Time (min : sec)				
Bounding Box	48 : 32	63 : 17	55 : 24	34 : 49
격자형공간분할	19 : 30	23 : 54	20 : 32	17 : 20

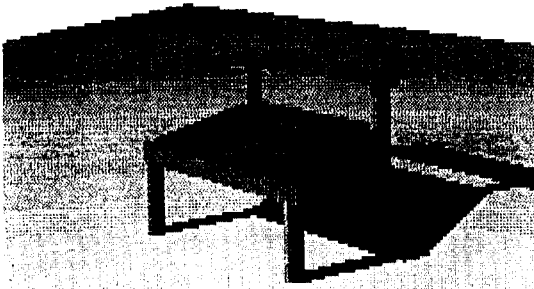


그림 7. 의자의 예
Fig. 7. Examples of chair.

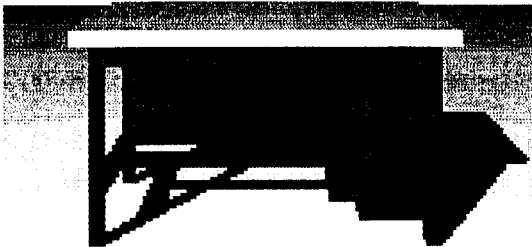


그림 8. 책상의 예
Fig. 8. Examples of desk.

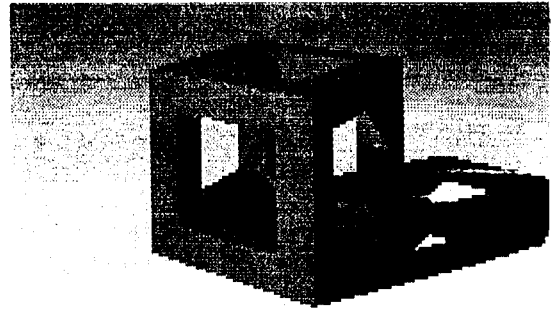


그림 9. 다각형의 예
Fig. 9. Examples of polyhedron.

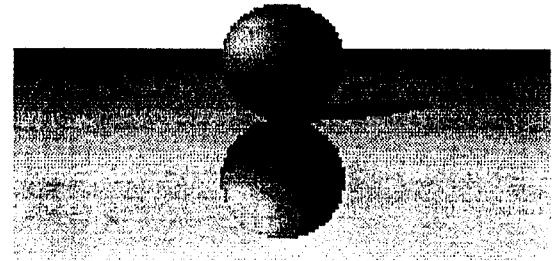


그림 10. 球의 예
Fig. 10. Examples of sphere.

V. 結 論

本 論文에서는 光선추적법을 利用하여 사실적으로 物體를 表現하고자 했으며 格子形空間分割을 제안하여 불 필요한 交點 計算을 줄이도록 하였다. 그러나 Bounding Box를 利用하게 되므로 格子內의 物體 이외에 빈 空間이 있게 된다. 따라서 格子內의 빈 부분에서 불 필요한 交點 計算을 줄이기 위해 더 세밀하게 空間을 分割하는 方法에 대한 研究가 필요하며, 실험결과와 같이 Bounding Box의 전반적인 處理過程에서 나타난 Aliasing을 除去하기 위해서 해상도를 높이는 것이 한 가지 方法이나 이 경우 Pixel수의 증가로 인해서 處理時間이 또 증가하게 되므로 저 해상도에서도 alising을 除去하기 위해서 Simple Area Anti-Aliasing技法을 도입해야겠다. 그러나 Aliasing이 생기는 부분은 주로 境界線 部分이므로 모든 Pixel에 이 技法을 적용하는 것은 무의미하며 따라서 境界線 部分에서만 이 技法을 使用하는 方法에 대한 研究와 Primitive 物體 이외에도 자유곡면을 表現하기 위해서

Bezier Curve^[12]등을 사용한 모델링 方法에 대한 研究가 必要하겠다. 끝으로 本 論文 作成에 정서등을 도와 준 本 研究室 김찬근께 감사하는 바이다.

參 考 文 獻

[1] David F. Rogers "Procedural elements for computer graphics," McGraw-Hill, pp. 296-305, pp. 363-379, 1985.

[2] Foley & Van Dam, "Fundamentals of interactive computer graphics," Addison Wesley, pp. 1-14, pp. 545-572, 1984.

[3] Newman & Sproul, "Principles of interactive computer graphics," McGraw-Hill, pp. 3-15, 1984.

[4] Van Dam, "Computer graphics comes of age," Comm. of the ACM, vol. 127, no. 7, pp. 638-648, 1984.

[5] Turner Whitted, "An improved illumination model for shaded display," Comm. of the ACM, vol. 23, no. 6, pp. 343-349, 1980.

[6] A.S. Glassner, "Space subdivision for fast ray tracing," *IEEE CG & A*, vol. 4, no. 10, 1984.

[7] Roy A. Hall & D.P. Greenberg, "A tested for realistic image synthesis," *IEEE CG & A*, vol. 3, no. 8, 1983.

[8] Saul Youssef, "A new algorithm for object oriented ray tracing," *CVGIP*, vol. 34, 1986.

[9] Jarvis & Judice & Ninke, "A survey of techniques for the continuous tone pictures on bilevel displays," *CGIP*, vol. 5, 1976.

[10] J. Amanatides, "Ray tracing with cones," *SIGGRAPH*, vol. 18, no. 3, pp. 129-135, 1984.

[11] Michael E. Mortenson, "Geometric modeling," John Wiley & Sons, pp. 366-369, 1985.

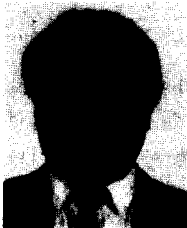
[12] Hea:n & Baker, "Computer graphics," Prentice-Hall, pp. 193-205, 1986.

[13] 김영일, 조동욱, 최병욱, "광선추적법과 공간 분할을 이용한 입체도형의 표현," 전기·전자공학 하계학술대회 논문집, 1987.

[14] 김찬, 김영일, 조동욱, 최병욱, "격자형 공간 분할을 이용한 입체도형의 표현," 전자공학 추계종합학술대회 논문집, 1987.

[15] 남국진, 최병욱, "입체도형의 표현을 위한 은 면세기," 전자공학회논문지, 1986. *

著 者 紹 介



金 英 一 (正會員)
 1962年 3月 1日生. 1985年 2月 한양대학교 전자공학과 졸업. 공학사학위 취득. 1988年 2月 한양대학교 대학원 전자공학과 졸업. 공학석사학위 취득. 1988年 3月 ~ 현재 한국전기통신공사 사업지원단 전임연구원. 주관심분야는 컴퓨터그래픽, 화상처리 등임.



趙 東 旭 (正會員)
 1959年 8月 3日生. 1983年 2月 한양대학교 전자공학과 졸업 공학사학위 취득. 1982年~1983年 (주)신도리코 기술연구소 연구원. 1983年 9月~1985年 8月 한양대학교 대학원 전자공학과 졸업. 공학석사학위 취득. 1985年 9月~1989年 2月 한양대학교 대학원 전자통신공학과 졸업. 공학박사학위 취득. 1987年~현재 한양대학교 전자통신공학과 강사. 주관심분야는 화상처리 및 인식, 컴퓨터그래픽, 전문가시스템, 자연언어처리 등임.

崔 炳 旭 (正會員)
 1949年 10月 2日生. 1981年 일본 경응의숙대학교 전기공학과 공학박사. 1981年~현재 한양대학교 공과대학 전자통신공학과 부교수. 1986年 9月~1987年 8月 University of Maryland 교환교수. 주관심분야는 Computer Vision 자연언어처리, 컴퓨터 그래픽, 전문가시스템 등임.