

### 3 차원 그래픽스 시뮬레이터 : SOFTGRA와 RACA

이 김수, 최 훈규, 경 종민  
한국 과학 기술원 전기 및 전자 공학과

### 3-D Graphics Simulators : SOFTGRA and RACA

Kil Su Eo, Hun Kyu Choi, Chong Min Kyung  
Dept. of Electrical Engineering, KAIST

#### Abstract

Two graphics simulators, SOFTGRA and RACA, have been developed. The SOFTGRA has been implemented by the scan line algorithm using 2 shading models with multi-light sources. The RACA is a ray casting simulator which produces shadowed images with realism. Above two simulators run on Sun micro system and MV10000 and produce image data which are transfered into IBM AT with Methcus' Omega graphics board and drawn there.

#### 1. 서 론

정보사회화되어 가면서 인간과 machine 간의 interface 는 가장 절실하게 해결되어야 할 문제로 등장하고 있다. 시각에 의한 인식이 인간에게 있어서 가장 효과적이라는 이유로 computer graphics 는 man-machine interface 의 해결책으로 여겨지고 있으며 이에 대한 많은 연구가 진행되고 있다.

Computer graphics 는 초기의 DVST(direct view storage tube) graphics terminal에 wire-frame 형태로 그리는 방식에서 반도체 메모리의 대용량화와 가격인하로 frame buffer 에 기반을 두는 raster-scan 방식의 graphics 으로 전환하였다. 이와 함께 리얼리즘을 지향하는 computer graphics art와 real time을 지향하는 graphics engine [1][2]의 두 가지 방향의 연구가 활발히 진행되고있다.

오늘날 computer graphics의 응용 분야는 CAD/CAM, Chemistry, Cartoon animation, Advertising, Flight simulation 등 다양하다. 본 논문에서는 물체의 표현을 작은

polygon 의 조합으로 이루어진 다면체(polyhedron)로 근접화 시킨 data를 scan line 방법[3]과 ray casting[4] 방법으로 image를 구현하였다. SOFTGRA는 scan line 방법을 쓴 것으로 두 가지의 shading 기법을 써서 real한 image를 얻어내었고, RACA는 ray tracing 방법을 사용한 simulator로서 shading은 물론 그림자까지 나타낼 수 있었다.

#### 2. Object 표현 및 입력 데이터

두 system의 호환성을 위해 두 system이 같은 형태의 input data file을 갖도록 하였다. input file에는 object 표현을 위한 data file과 simulation을 위한 user file이 있다. object의 표현을 위한 data 구조는 다음과 같다.

object = {vertices, polygons, call objects,  
          bounding sphere}

,where vertex = {name, coordinate}

          polygon = {color, vertex set, bounding sphere}

          bounding sphere = {radius, center coordinate}

이것의 주요 특징을 살펴보면, 한 object에서 정의된 vertex의 이름은 local하다. Polygon은 그 object 내에서 정의된 vertex만으로 이루어진다. Bounding sphere란 이름 그대로 그 물체 혹은 다각형을 둘러싸는 구이다. 이 구는 ray casting을 수행할 때, 물체 또는 polygon과 ray의 교차 가능성을 빨리 알아내는데 유용하게 쓰인다. 입력 data의 syntax format는 다음과 같다.

Syntax format:

```
define object
  Vxxx X Y Z;
  ....
  P color Vm .... Vn;
  ....
  call predefined_object orientation;
  ....
defend;
```

여기서 orientation은 translation, rotation, mirroring, scaling 의 네가지 요소가 가능하다.

user file에는 눈의 위치, 눈의 초점 방향, light source의 위치 및 밝기, 그리고 surface parameter로서 물체 표면의 윤택도를 나타내는 glossiness (이는 Phong의 illumination model에서의 cosine 함수의 역수가 된다.), 분산 계수, 윤택 반사 계수가 있으며 ambient light intensity가 들어 있다.

### 3. Illumination

물체 표면의 intensity 값은 표면의 orientation과 고유의 성질, 눈의 위치 및 보는 방향, light source의 위치등을 고려해야 한다. 물체 표면의 한 점에서의 intensity는 그 점에서의 light intensity의 Lambert term과 Phong term을 합한 것으로 계산되며 만일 그 점이 shadowed region에 속해 있으면 그림자 효과를 추가시키면 된다. Lambert term은 diffuse reflection term이라고도 하며 다음과 같이 계산된다.

$$I_d = I_p k_d (L \cdot N)$$

(L, N 은 normalize 되었다고 가정)

- where  $I_d$  : 분산 반사 광도  
 $I_p$  : light source 광도  
 $k_d$  : 분산 계수  
 L : direction to light source  
 N : surface normal

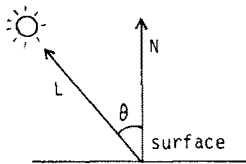


그림 1. L vector와 N vector의 정의

reflected light의 광도는 그 점의 surface normal vec-

tor와 그 점으로부터 광원으로 향하는 vector가 이루는 각의 cosine, 즉 두 vector의 dot product에 비례한다. 이때 광원으로부터 발산되는 빛은 모든 방향에 걸쳐 두루 퍼져나가므로 보는 사람의 위치에 무관하다. light energy는 집광원일 경우 거리의 제곱에 반비례하지만 본 시스템에서는 이를 무시하였다. Phong term은 흔히 specular reflection term이라 불리우는데 이는 윤택이 있는 물체에서 발견되는 현상이다. 즉 반질반질한 표면을 갖는 물체의 표면에는 light source의 color가 나타나는 highlight 부분이 있다. 이 부분은 보는 사람의 위치가 바뀔 때 달라진다. 이것은 매끈한 물체의 표면이 빛을 불균등하게 반사시키는데 기인하며 거울과 같은 완전 반사체인 경우 입사각과 반사각이 같은 방향으로만 물체를 반사시킨다. 사파와 같은 불완전 반사체에서는 reflected light의 광도가  $\alpha$ 가 증가함에 따라 급격히 감소한다. 이 비율은  $\cos^n \alpha$ 에 비례한다. 이때 n 값은 surface의 윤택도에 의존하는 factor이다. 본 시스템에서는 다음 식과 같이 계산했다.

$$I_s = I_p k_s (R \cdot V)^n$$

(R, V 는 normalize 되었다고 가정)

- where  $I_s$  : 윤택 반사 광도  
 $I_p$  : light source 광도  
 $k_s$  : 윤택 계수  
 R : direction of reflection  
 V : direction to viewpoint

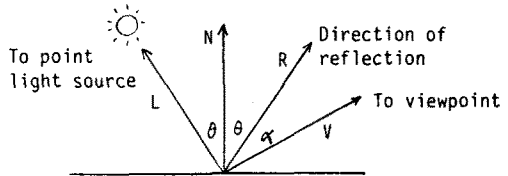


그림 2. Specular reflection의 정의

### 4. SOFTGRA

Hierarchical한 input data file을 global한 vertex list set와 polygon list set으로 planarize시킨다. Polygon list의 coherence를 이용하기 위해 각 polygon을 maximum y 값 순서로써 sorting해 둔다. 매 screen line 마다 sorting된 polygon list를 갖고서 active polygon list와 active edge list를 만든다. 두 active list만을 갖고 계산함으로써 계산량을 줄일 수 있다. 눈으로부터 가까운 polygon을 찾아 이를

display함으로써 보이지않는 부분을 제거시킨다. 보이는 부분의 polygon에 대하여 pixel의 intensity 값을 계산하는데 intensity 값은 Lambert term과 Phong term을 더한 것으로 계산한다.

SOFTGRA는 shading 방법에 따라 Gouraud shading 방법을 쓴 것과 Phong shading 방법을 쓴 것의 두 가지 version이 있다.

Gouraud shading 방법은 먼저 모든 polygon에 대하여 surface normal을 계산하고 모든 vertex에 대하여 vertex에 인접하는 polygon들의 surface normal들을 평균함으로써 vertex normal을 계산한 후 vertex intensity를 계산한다. scan line과 active edge의 교차점의 intensity는 active edge의 양 vertex의 intensity를 linear interpolation 함으로써 구하고 polygon 내부의 intensity는 교차점들의 intensity를 다시 linear interpolation 함으로써 구한다.

Phong shading 방법은 polygon의 surface normal들을 계산하여 vertex normal 들을 구하고 scan line과 active edge와의 교차점에서의 vertex normal을 교차점들의 edge의 두 vertex normal의 interpolation에 의해 구한다. polygon span에 대한 vertex normal을 교차점에서의 normal vector의 interpolation에 의해 구한 다음 최종적으로 intensity를 계산하는 방식이다. 이 방식은 계산량이 Gouraud 방법보다 많지만 보다 real한 image를 얻을 수 있다.

한편 스크린의 pixel의 갯수 (resolution)가 보통 1000 x 1000 정도밖에 되지않기 때문에 aliasing 현상이 나타난다. aliasing 현상은 신호를 discrete하게 sampling할 때 원래의 신호가 갖고있던 spectrum을 다 포함하지 못했을 때, 즉 Nyquist rate를 만족시키지 못했을 때 발생한다. 이것의 양상은 계단 모양의 모서리, 작은 물체의 소멸등으로 나타난다. 이것을 어느 정도 방지하기 위해서는 low pass filtering[5]이 필요하다. SOFTGRA에서는 이를 위하여 하나의 pixel을 9 개의 3 x 3의 sub-pixel로 나누어 각각의 sub-pixel에 대한 intensity 값을 구한 후 평균 값을 취하는데 이러한 방법을 super sampling이라고 한다.

## 5. RACA

Ray casting[6]은 ray tracing의 depth가 1인 경우에 해당한다. 따라서, ray tracing에서 취급할 수 있는 것들 중에서 refraction, mirror effect등을 ray casting에서는 묘사할 수가 없지만 shading, shadowing antialiasing등은 수행할 수가 있다. Ray의 방향은 눈에서부터 pixel의 모서리를 통과하는 방향이며 그 방향의 intensity는 ray 방향의 가장 가까운 물체의 표면의 한 점의 intensity 값이 된다. intensity값은 그 점에서의 light intensity의 Lambert

term과 Phong term을 구한 다음 light source를 추적하여 만일 해당 점으로부터 light 방향으로 ray를 쏘았을 때 물체와의 교차점이 있으면 그림자 영역에 포함되는 경우로서 그림자 효과를 추가한다. 그림자 효과는 교차점에서의 intensity 값에 1 보다 작은 shadow factor를 곱하여 나타내었다. 한 pixel의 light intensity를 계산하기 위하여 사용되는 ray는 그 pixel의 4 개의 모서리를 지나가는 것들로 intensity 값은 그것들의 평균으로 구해진다. 그래서  $m \times n$  개의 pixel을 갖는 스크린에 대해서는 적어도  $(m + 1) \times (n + 1)$  번의 ray casting이 필요하다.

한편, ray casting과정에서 가장 많은 CPU time을 소모하는 [7] polygon-line intersection을 빠르게 구하기 위하여 각각의 object와 polygon은 그것들을 둘러싸는 구(bounding sphere)를 가지도록 하였다. Minimum bounding sphere가 가장 이상적이겠으나 그것은 구하기가 힘들기 때문에 크기가 minimum bounding sphere보다는 크지만 쉽게 구할 수 있는 구로 하였다. 그리하여, 어떤 ray가 교차하는 물체를 찾는 문제를 직선과 구의 교차를 test하는 문제로 변환하여 교차하지 않는 물체들을 빨리 discard 시킨다. 즉 bounding sphere가 ray와 교차하는 물체에 대하여 내부의 polygon 구에 대한 ray와의 교차 test를 하고 교차하는 polygon에 대해서만 detail한 test를 거치게 한다. detail test에서는 polygon을 포함하는 평면과 ray와의 교차점을 구하고 교차점이 polygon의 내부에 있는가를 check한다.

RACA에서의 aliasing은 pixel의 4개의 모서리의 intensity값이 차이가 큰 경우에만 pixel 내부를 4개의 sub-pixel로 나누어 intensity값을 계산하고 해당 pixel의 intensity는 8개의 값의 평균으로 계산한다. 이는 aliasing 현상이 물체의 모서리와 같은 intensity값의 변화가 큰 부분에서 심각하게 나타나기 때문에 이러한 부분에서만 aliasing문제를 해결함으로써 전체 계산량을 줄이고자 함이었다.

## 6. 보조프로그램들

Object data를 생성하는 몇가지 program들을 소개한다. 아무리 훌륭한 rendering software를 가지고 있어도 object data를 쉽게 얻을 수 없으면 쓸모가 적다는데 이들 보조프로그램들의 중요성이 존재한다.

CHECK : check board 를 생성한다.

DONUT : Toroid 혹은 Doughnut 모양의 object를 생성 다.

CYLINDER: 실린더 모양의 물체를 생성한다.

SPHERE : 구 모양의 물체를 생성하는데 사용된다.

7. Discussion

Scan line 알고리즘의 SOFTGRA와 ray casting 방법에 의한 RACA라는 graphics simulator를 소개하였다. SOFTGRA는 shade, antialiased image를 나타낼 수 있으며 RACA는 shadowed image까지 얻을 수 있다. 이들은 color mode 와 흑백 mode 둘 다 적용될 수 있다. Image data는 Sun 이나 MV 10000 에서 만들어 IMB-AT (Omega graphics board 내장, 1024 x 768 x 8)에서 그림을 그린다.

차기의 계획은 ray tracing 방법에 의한 graphics simulator의 개발에 두고 있으며 물체를 parametric patch에 의한 표현을 시도하고자 한다.

References

- [1] John Poulton, Henry Fuchs, John D. Austin, John G. Eyles, Justin Heinecke et al : "PIXEL PLANES : Building a VLSI-Based Graphic System." 1985 Chapel Hill Conference on VLSI, proceeding, pp35 -- 60.
- [2] N. Gharachorloo and C. Pottle : "SUPER BUFFER : A Systolic VLSI Graphics Engine for Real Time Raster Image Generation", System." 1985 Chapel Hill Conference on VLSI, proceeding, pp285 -- 305.
- [3] J.D. Folly, A. Van Dam, "Fundamentals of Interactive Computer Graphics", Addison Wesley, 1984.
- [4] Robert A. Goldstein and Roger Nagel : "3-D Visual Simulation." *simulation*, Jan. 1977, pp25 -- 31.
- [5] Franklin C. Crow : "The Antialiasing Problem in Computer Generated shaded Images." *Communication of ACM*, Nov. 1977, Vol. 20, No. 11, pp 799 -- 805.
- [6] 이길수, 최훈규, 경종민, "RACA : Ray casting에 의한 그래픽 시뮬레이터", 전기재료, 반도체 및 CAD 학술대회 논문집, 1987, pp 152 -- 154.
- [7] Turner Whitted : "An Improved Illumination Model for Shaded Display". *Communication of ACM*, Jun. 1980, Vol.23, No.6, pp 343 -- 349.

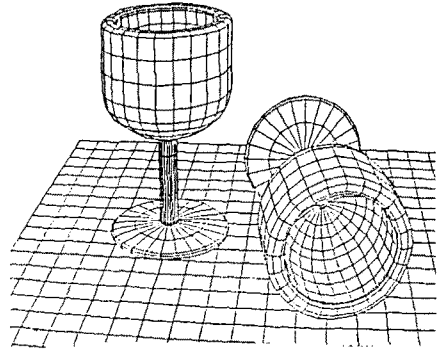


그림 3. wire frame drawing

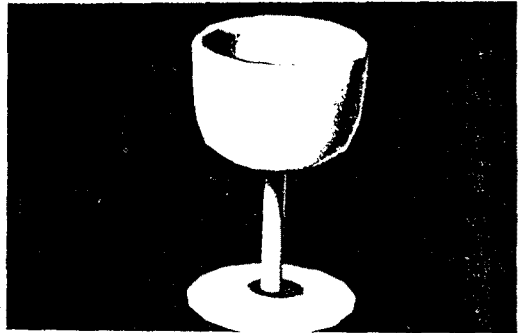


그림 4. glass 정물 (SOFTGRA)

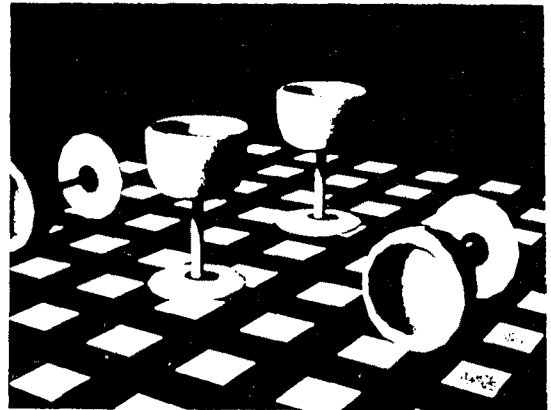


그림 5. glasses (SOFTGRA)

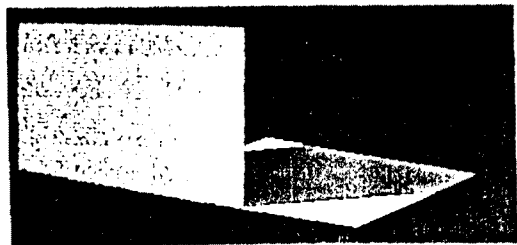


그림 6. RACA 출력 예