

효율적인 데이터 저장을 기초로한 광선 추적의 가속화 방안

(Ray Tracing Acceleration Schemes Based on Efficient Data Storage)

崔勳奎\*, 慶宗旻\*

(Hun Kyu Choi and Chong Min Kyung)

要約

본 논문에서는 광선 추적중 광선과 물체와의 교차계산 및 그림자 계산시간을 단축시키기 위한 3차원 물체를 저장하는 두가지 새로운 방안을 제시한다. 다각형 표면으로부터 출발하는 모든 1차 반사(굴절) 광선과 교차할 가능성이 있는 3차원 물체들을 '반사(굴절) 절두체'에 등록시켜둠으로써 같은 다각형의 표면으로부터 출발하는 2차 광선과 교차할 물체를 절두체 내에 등록된 물체의 목록을 찾아봄으로써 빠르게 찾을 수 있다. 다른 하나는 '음영 피라미드'라고 하는 것인데 해당 광원으로부터 다각형내의 모든 점으로 향하는 경로상에 있는 물체들의 목록을 갖고 있다. 어떤 한 점의 그림자 여부는 음영 광선과 해당 음영 피라미드 내에있는 물체들과의 교차 여부로써 결정 된다. 절두체 또는 음영 피라미드 내부로 등록하는데 드는 비용에도 불구하고, 광선과 교차 계산을 해야할 물체를 빨리 찾고 교차 계산 횟수의 감소로 인하여 광선 추적의 총 묘화 시간은 균일 분할된 공간상을 광선이 진행하면서 그 분할 공간의 내부에 존재하는 물체와의 교차 계산을 수행하는 기존의 방법에 비교했을때 대략 50% 감소되었다. (광선과 가장 가깝게 만나는 물체를 찾는 묘화 계산에서 10%, 그리고 그림자 계산에서 70% 감소)

Abstract

This paper presents two new storage schemes of 3-D objects to accelerate the finding of the first intersecting object and the shadow computation steps in ray tracing. The 3-D objects which are potentially intersected with the first reflection (refraction) ray are enrolled within a so called 'reflection (refraction) frustum' for a polygon object. Only those objects registered in the corresponding frustum are immediately checked for intersection with the secondary rays emanating from the same polygon. The other is called a 'shadow pyramid' which contains the candidate objects possibly blocking the path from the relevant light source to any point on the relevant polygon. The shadow testing of a point is performed only against the objects contained in the associated shadow pyramid. Despite the cost needed for registration of objects within frusta or shadow pyramids, the total rendering time of ray tracing using the proposed approaches was reduced by approximately 50% (10% in color rendering time and 70% in shadow testing time) from the conventional cell traversal scheme under the 3-D uniform subdivision environment due to the fast finding of candidate objects for intersection and the reduced number of intersection calculations.

\*正會員, 韓國科學技術院 電氣 및 電子工學科  
(Dept. of Electrical Eng., KAIST)

接受日字: 1989年 4月 27日

I. 서론

광선 추적 기법은 모든 교차점에서 광선 추적 과정

을 반복적으로 수행시킴으로써 반사와 굴절 및 그림자 효과를 줄 수 있는 3차원 이미지의 사실적 묘화를 위한 유력한 한 방법이다. 눈으로부터 출발한 근원 광선은 주사선 순서대로 발사되어 스크린상의 각 화소를 통과하여 광선과 가장 먼저 만나는 물체와의 교차점을 찾는다. 각 교차점에서는 최대 3가지의 광선(반사, 굴절, 그림자 광선)이 만들어진다. 물체의 표면이 반사성(굴절성)이면 반사(굴절) 광선이 만들어지고 이 광선이 새로이 공간상의 물체와의 교차점을 반복적으로 찾게 된다. 광선 추적은 광선이 물체 공간의 MBB (minimum bounding box)를 벗어나는 경우와 해당 화소에 대한 색깔의 기어도가 떨어져 더 이상의 추적이 불필요한 경우에 끝나게 된다.

광선 추적 방식에서 소요되는 대부분의 시간은 광선이 기본 물체(다각형, 구, 실린더 등)와 교차점이 있는지 또 그 교차점은 어디인가를 알아내는데 쓰인다<sup>11)</sup>. '내포 체적법'<sup>12,3)</sup>, '공간 분할법'<sup>4,5)</sup>, '연필 추적법'<sup>6)</sup>과 같은 기법들이 광선과 물체와의 교차 계산의 수를 줄이기 위해 개발되었다. 내포 체적법은 공간상의 물체들을 포함하는 구 또는 입방체와 같은 교차 계산을 쉽게 할 수 있는 체적으로 둘러싸고 이들을 포함하는 더 큰 체적으로 다시 둘러싸는 계층적인 구조를 만든다. 만일 어떤 광선이 내포 체적과 만나지 않을 경우 그 내부에 있는 모든 물체와는 만나지 않게 되므로 불필요한 많은 교차 계산을 피할 수 있게 된다. 공간 분할법은 물체들이 놓여있는 공간을 'octree 분할 방식'<sup>4)</sup>이나 '균일 분할 방식'<sup>5)</sup>으로 나누고 각 분할된 공간내에 들어있는 물체들을 등록시켜 놓고 광선이 만나게 되는 분할 공간내에 있는 물체와의 교차 계산을 수행하게 함으로써 교차계산의 수를 줄인다. 연필 추적법은 광선의 광학적 성질을 이용한 것으로 시스템 행렬과 광선 보간법을 써서 일관성이 있는 인접광선의 색깔을 결정해주는 방식이다.

광선 추적 방식에서의 이미지의 사실감에 크게 기여하는 그림자의 계산은 또 다른 많은 시간이 필요한 부분으로서 전체 묘화 시간의 절반 이상을 차지하고 있다. 눈의 위치에서는 보이지만 광원의 위치에서는 보이지 않는 부분이 그림자 영역이다. 따라서 물체 표면의 한 점에 있어서의 그림자 검사는 그 점에서 어떤 광원이 보이지 않는가를 알아내는 것이 된다. 기존의 방식에서는 광선과의 교차점에서 각 광원 방향으로 향하는 '음영 광선'을 출발시켜 광원에 도달하기 전까지 적어도 하나의 물체가 음영 광선과 교차하면 그 점은 해당 광원에 대해 그림자 영역에 있다고 판정한다. 만일 음영광선과 만난 물체가 없으면

그 점은 그림자가 지지 않는 것으로 판정된다.

그림자 계산을 가속화 시키기 위해 '그림자 체적법'<sup>7)</sup>, '광용기법'<sup>8)</sup> 등이 개발되었다. 그림자 체적법은 주로 주사선 알고리즘을 위해 사용된 것으로 그림자 체적을 음영 다각형으로 둘러싸인 체적으로 정의하는 바, 음영 다각형은 각 광원에 대해 물체의 실루엣 능선을 찾아서 해당 광원으로부터 투영된 것으로 정의된다. 광선 추적중 광선이 음영 다각형과 교차하여 그림자 체적내로 들어오면 현재 진행중인 광선은 교차한 음영 다각형을 만든 해당 광원에 대해 그림자 영역에 있게 된다. 즉, 광선과 물체와의 교차가 그림자 체적내에서 일어나면 그 교차점은 그림자 영역내에 있고 그림자 체적밖에서 교차가 일어나면 그림자 영역밖에 있다고 판정한다. 이 방법은 많은 음영다각형이 만들어지고 이들을 기본 물체와 똑같이 취급하므로 실제로 물체 갯수가 많은 이미지에 대해 교차 계산을 수행시키게 되는 결과가 되며 음영다각형이 많이 분포되어있는 부분에서는 많은 음영 광선과 음영다각형의 교차 계산이 필요하므로 기존의 방법보다 느려진다.

광용기 방법은 각 면이 2차원 배열을 가진 화소로 된 입방체의 중심에 광원을 두고 광원을 꼭지점으로 하고 화소의 네 모서리를 지나는 무한 사각뿔을 만들어 두고 무한 사각뿔 내에 있는 모든 물체들을 해당 화소에 등록시켜둔다. 어떤 교차점에서의 그림자 검사는 음영 광선이 광용기의 어떤 화소와 만나는가를 살펴보고 해당 화소에 등록된 물체들과의 교차 계산으로 간단히 수행된다. 그러나 광원이 물체들로부터 멀리 떨어져 있는 경우, 물체들이 광용기의 일정한 부분에 집중적으로 몰려 등록될 수도 있어서 이득을 보지 못하는 경우도 있다.

본 논문에서는 광선 추적에 소요되는 묘화 시간을 줄일 수 있는 두 가지 새로운 물체 저장 방안을 제안한다. 첫째로 눈으로부터 출발한 근원 광선과 만난 물체의 표면에서 출발하는 1차 반사(굴절)광선과 만나는 물체를 찾는 과정을 가속화 시키는 방안으로 반사(굴절)절두체를 제안한다. 눈으로부터 출발한 근원 광선과 가장 가까운 거리에서 만나는 물체는 3-D DDA(digital differential analyser)방식으로 분할 공간을 하나씩 방문하면서 찾는데, 만일 교차한 물체가 반사체(굴절체)라면 반사(굴절)절두체를 만든다. 1차 반사(굴절)광선과 만나는 가장 가까운 물체를 찾는 것은 절두체내에 등록된 물체들만 조사하면 된다.

둘째로 그림자 계산을 빨리 수행시키기 위해 '음영 피라미드'를 제안한다. 광선과 만난 다각형 물체

에 대해 각 광원에 대한 음영 피라미드를 만드는데 광원의 위치가 꼭지점이 되고 다각형 자체가 밑면이 된다. 피라미드 내부에 있는 광원을 차단할 수 있는 물체들을 등록시키고 음영 광선과 등록된 물체와의 교차 계산으로 그림자 검사를 수행한다. 절두체 또는 음영 피라미드 내부에 있는 물체들을 빨리 등록시키기 위해 균일 분할 방식으로 분할된 공간을 octree 방식으로 재구성해 놓았다.

II 장에서는 공간 분할 방식에 대해 설명하고, III 장과 IV 장에서는 2 차 광선 추적을 가속화 시키는 방안과 그림자 계산을 가속화 시키는 알고리즘을 기술하였고, 끝으로 인접 광선에 대해 쓸 수 있는 구간 일관성에 대해 설명한다.

II. 공간 분할 방법

공간 분할 방법은 교차 계산의 횟수를 줄이기 위한 것으로서 물체 공간을 작은 입방체로 분할하고 분할된 각 공간내에 부분적으로 걸쳐있거나 포함되어 있는 물체들을 등록시킨다. 광선이 어떤 분할 공간을 지나갈때 그 공간내에 등록된 물체들과 교차 여부를 검사함으로써 광선과 교차 여부를 검사할 물체의 갯수를 크게 줄일수 있다. 만일 교차하는 물체가 없으면 광선이 만나게 되는 그 다음 분할 공간으로 광선을 진행시킨다.

공간 분할 방식에는 octree 분할방식과 균일 분할 방식이 있다. octree 방식은 3 차원 물체 공간을 8 개의 분할 공간으로 나누고, 나누어진 분할 공간을 각 공간내에 등록되는 물체의 갯수가 어떤 문턱 값보다 작을때까지 반복적으로 나누는 방법이다. 그 결과 많은 물체가 몰려있는 영역은 작게 분할되고 물체의 갯수가 적은 영역은 커다랗게 분할되어 저장 효율을 개선할 수 있다. 그러나 octree 구조에서 광선 방향으로 인접 분할 공간을 찾는 비용은 분할 공간 사이의 크기가 균일하지 않기 때문에 균일 분할 방식에 비해 크다.

본 논문에서는 분할 공간사이의 광선의 진행과 해당 분할 공간내로의 물체의 등록을 빠르게 하기 위하여 물체 공간을 균일 분할 방식으로 분할하였고, 절두체 또는 음영 피라미드 내부에 있는 물체들을 빠르게 찾기위해 그림 1에서 보는 바와 같이 균일하게 분할된 공간을 semi-octree 방식으로 재구성해 두었다.

이것은 절두체 또는 음영 피라미드에 물체를 등록시킬때, 모든 물체를 하나씩 검사하는 것은 비 효율적이기 때문이다. 전체 분할 공간의 갯수는 기본 물체의 갯수에 50배한 것이 적당하므로<sup>[9]</sup> 분할 공

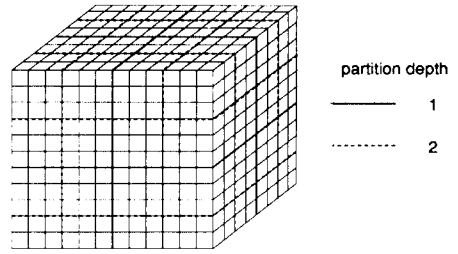


그림 1. 균일분할된 분할 공간과 semi-octree로 재구성된 물체 공간

Fig. 1. Uniformly divided object space and its reconstructed semi-octree structure.

간을 하나씩 검사하면서 그 속에 들어있는 물체들을 검사하는 것은 더욱 비효율적이다. 제안된 semi-octree 계층 구조의 바닥 node는 인접한 여러 개의 분할 공간내의 물체 목록의 합이 되는 물체 목록을 가지고 있고, 중간 node는 그 아래의 node들을 포함하는 것으로서 하나의 내포체적으로 사용된다.

III. 절두체 내에서의 2 차 광선 추적

1. 반사 및 굴절 절두체의 정의

눈으로부터 출발한 근원 광선이 만난 물체가 반사체이고 근원 광선과 처음으로 만나는 다각형인 경우 해당 다각형에 대해 반사 절두체를 만든다. 반사 절두체는 그림 2 (a)에서 보는 바와 같이, 그 물체의 표면으로부터 1 차 반사되는 모든 광선들을 포함하는 체적으로 정의된다.

그림 2 (a)에서 'E'는 눈의 위치를 표시하며 'P'는 반사 절두체의 가상 꼭지점을 나타내는데, 해당 다각형의 평면에 대해 'E'의 거울대칭점에 해당한다. 'P'의 위치는

$$P = E - 2dN \tag{1}$$

으로 주어지며, 식 (1)에서 'd'는 눈의 위치 'E'에서 다각형을 포함하는 평면까지의 최단거리를 의미하고, 'N'은 다각형의 단위 법선 벡터이다.

근원 광선과 만난 투과성 다각형인 경우 그림 2(b)와 같이 다각형 표면으로부터 출발하는 1 차 굴절 광선들을 포함하는 굴절 절두체를 만든다. 그림 2 (b)에서 'E'는 눈의 위치를, 'P'는 굴절 절두체의 가상 꼭지점을 나타내는데, 가상꼭지점 P를 계산하는 방법은 다음과 같다.

(A) 다각형의 한 모서리의 두 끝점에서의 굴절 광선

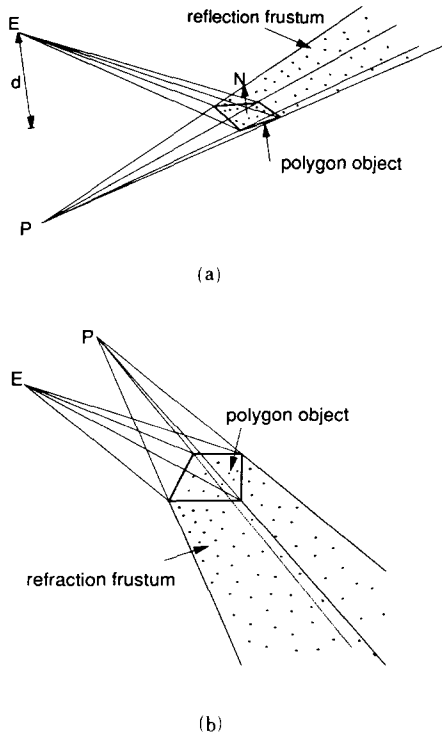


그림 2. (a) 반사 절두체의 정의  
(b) 굴절 절두체의 정의  
Fig. 2. (a) Definition of a reflection frustum.  
(b) Definition of a refraction frustum.

의 방향을 구한다<sup>[10]</sup>. Snell의 법칙으로부터  $\eta$ 는

$$\eta = \frac{\eta_1}{\eta_2} = \frac{\sin \theta_2}{\sin \theta_1} = \frac{1}{n} \quad (2)$$

이다.

여기서  $\theta_1, \theta_2$ 는 각각 입사각과 투사각을 나타내고,  $\eta_1, \eta_2$ 는 입사측과 반사측의 투과 계수를 나타내고,  $n$ 은 매질 1에 대한 매질 2의 상대 굴절율을 나타낸다.

굴절 방향  $T$ 는  $I$ 를 입사 광선의 단위 벡터로,  $N$ 을 다각형의 단위 법선 벡터라고 할 때

$$T = \eta I + (\eta c_1 - c_2) N \quad (3)$$

으로 주어진다. 여기서  $c_1 = \cos \theta_1 = -I \cdot N$ 이고

$$c_2 = \cos \theta_2 = \sqrt{1 - \sin^2 \theta_2} = \sqrt{1 - \eta^2 \sin^2 \theta_1} = \sqrt{1 - \eta^2 (1 - c_1^2)}$$

이다.

이를 정리하면

$$\begin{aligned} T &= \eta I + (\eta c_1 - \sqrt{1 - \eta^2 (1 - c_1^2)}) N \\ &= \frac{I}{n} + \frac{(c_1 - n \sqrt{1 - (1 - c_1^2) / n^2})}{n} N \\ &= \frac{I + (c_1 - \sqrt{n^2 - 1 + c_1^2}) N}{n} \end{aligned} \quad (4)$$

이 된다.

(B) 굴절 방향을 계산한 점을 지나고 식(4)에서 구한 방향을 갖는 두 직선의 식을 구한 다음 두 직선의 교차점을 계산하면, 이 점이 바로 굴절 절두체의 가상 꼭지점이 된다.

이의 자세한 계산 과정은 부록에 나타내었다.

물체 공간의 MBB 밖에는 물체가 존재하지 않으므로 실제적인 절두체의 크기는 정의된 체적과 물체 공간의 MBB와의 공통 부분이 된다.

## 2. 절두체 내로의 물체의 등록

절두체가 정의되면 절두체 내에 있는 물체를 찾아 등록해 주어야 한다. 절두체안으로의 물체의 등록은 분할 공간 포함 검사(이때 분할 공간이란 semi-octree 구조의 분할 공간을 의미)와 물체 포함 검사의 두 과정으로 되어있다. 하나의 다각형 물체의 절두체 내부에 있는 물체를 찾는 알고리즘의 pseudo코드는 다음과 같다.

object-registration (cell)

```

{
  if null object cell then return;
  if cell-rejection with polygon then return;
  if cell-rejection with side or tangent planes
  then return;
  if current cell is a leaf node then
    {while object list in the node is not empty do
      {if the object's normal is visible from the
        virtual point then
          {if the object lies below the polygon then
            goto label; for each side and tangent
            plane do
              if the object lies outside then goto label;
              register this object within the frustum;
            }
          label:next object;
        }
      }/* leaf node*/
    else/* internal node*/
      {for all its children cell do
        object-registration (child-cell);
      }
    }
}
    
```

Semi-octree의 계층 구조를 root에서부터 내려오면서 분할 공간과의 포함 검사를 하게 되는데, 절두체와 만나지 않는 분할 공간은 고려 대상에서 제외시키고 절두체와 만나는 분할 공간내의 물체와 포함 검사를 수행한다. 절두체와의 포함검사는 다음의 세 과정으로 구성되어있다.

(A) 다각형 평면으로의 제거

반사 절두체인 경우는 그림 3 (a)와 같이 완전히 다각형 물체를 포함하는 무한 다각형 평면의 아래쪽 공간에 놓여있는 분할공간 또는 물체는 제거된다. 굴절 절두체인 경우는 반대로 완전히 위쪽 공간에 있는 분할 공간 또는 물체가 제거된다.

(B) 절두체 옆면으로의 제거

다각형의 각 모서리에 대해 모서리의 두 끝점과 가상 꼭지점의 세점을 포함하는 경계 옆평면을 그림 3 (b)와 같이 만들고 완전히 이들의 바깥쪽에 있는 분할 공간 또는 물체를 제거한다.

(C) 절두체 접평면으로의 제거

인접하는 두 경계 옆평면에 의해 만들어지는 모서리를 지나고 두 옆평면의 법선의 평균을 법선으로하

는 접평면을 그림 3 (c)와 같이 추가적으로 만들고 이들 접평면의 바깥쪽에 존재하는 분할 공간 또는 물체를 제거한다.

이 세 과정에서 제거되지 않은 물체는 절두체 내부에 속해있다고 보아 절두체에 등록시킨다. 실제로 등록 대상 물체가 다각형인 경우 절두체 방향과 다각형의 법선 벡터의 내적의 부호가 '음(-)'인 경우에만 포함 검사를 하므로 등록되는 다각형 물체의 수는 실제 존재하는 다각형 물체 숫자의 절반 가량 된다.

3. 절두체를 이용한 2차 광선 추적 알고리즘

근원 광선과 가장 가깝게 만난 물체가 반사체(굴절체)라고 한다면, 그 교차점에서 반사(굴절) 방향으로 2차 광선 추적을 수행한다. 2차 광선이 분할 공간을 방문하면서 교차할 물체를 찾는 것이 아니라, 2차 광선이 만날 가능성이 있는 물체들을 근원 광선과 다각형 물체가 처음 만났을때 절두체를 이용하여 목록을 만들어두었으므로 이를 이용한다. 인접 근원 광선이 같은 다각형 물체를 만나면 같은 절두체내에 등록된 목록을 사용함으로써 분할 공간을 방문하는데 걸리는 시간을 제거할 수 있다. 절두체 내의 물체 목록을 이용하는 2차 광선 추적을 가속화하는 방법의 흐름도는 그림 4와 같다.

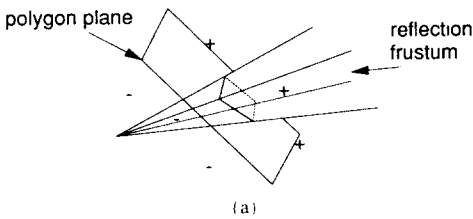
(A) 초기 처리 과정에서 물체 공간내의 모든 물체들에 대해 6개의 극단 좌표값( $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ )을 계산해 둔다.

(B) 다각형 표면으로부터 출발하는 모든 2차 광선의 방향은 다각형의 무게 중심을 지나는 2차 광선을 중심으로 분포하고 있으므로, 가상 꼭지점을 출발하여 다각형의 무게 중심을 지나는 광선의 방향을 '절두체 방향'이라고 정의한다. 그리고 절두체 방향 벡터가 갖는 최대 성분 방향을 '절두체의 주축 방향'으로 선택한다.

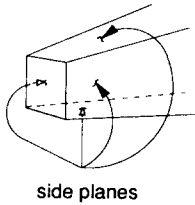
(C) 절두체내에 등록된 물체들을 절두체의 주축 방향에 대해 정돈시킨다. 예를 들어 주축방향이 '+x'이면 등록 물체들의  $x_{min}$  값이 커지는 순서로 정돈시키고, 주축방향이 '-y'이면  $y_{max}$ 가 감소하는 순서로 정돈한다.

2차 광선과의 교차 여부는 정돈된 목록의 물체를 순서대로 꺼내어 검사한다. 각 물체가 6개의 극단 좌표값을 갖고있게 하고 정돈된 목록을 만든 이유는 목록에 있는 물체들중에서 그들의 광선과의 교차점이 2차 광선의 출발점으로부터 지금까지 알려진 가장 가까운 교차점보다 먼곳에서 만나게 되는 물체를 미

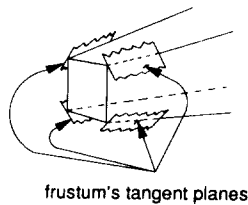
L3



(a)



(b)



(c)

그림 3. 반사절두체 내로의 물체의 등록 과정

- (a) 다각형 평면으로의 제거
- (b) 절두체 옆면으로의 제거
- (c) 절두체 접평면으로의 제거

Fig. 3. Object registration within a reflection frustum.

- (a) culling with a polygon plane.
- (b) culling with side planes.
- (c) culling with tangent planes.

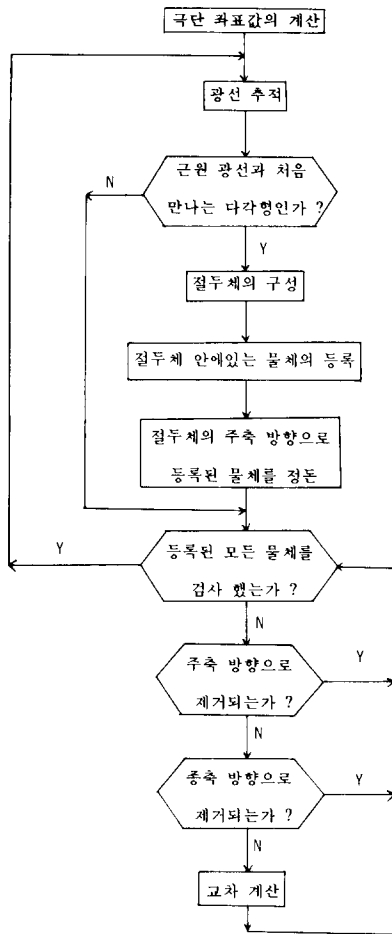


그림 4. 2차 광선의 가속화 방안에 대한 흐름도  
 Fig. 4. Flow chart of the accelerating scheme for secondary ray.

리 알아내어 불필요한 교차 계산을 피하기 위함이다. 하나의 주축과 두 개의 종축에 대하여 목록내의 물체중 교차 계산이 불필요한 것을 고려대상에서 제외시킨다. 그림 5는 2차원 상에서 주축과 종축의 방향이 '+' (+x, +y)일 때 각 축방향으로의 불필요한 교차 계산을 피하는 것을 보이고 있다.

(D) 주축 방향으로의 제거

지금까지 계산된 가장 가까운 교차점의 주축 성분보다 주축 방향 성분의 '최소값'이 '더 큰' 물체가 발견되면 그 물체와 목록의 나머지에 있는 물체를 무시할 수 있다. 그림 5에서 'obj-D'와 'obj-E'는 이들의 주축 방향 성분의 최소값 'd'와 'e'가 알려진 교차점 'P'의 주축방향성분 'P'보다 더 큰 곳에 있기 때문에 교차 계산을 하지 않는다.

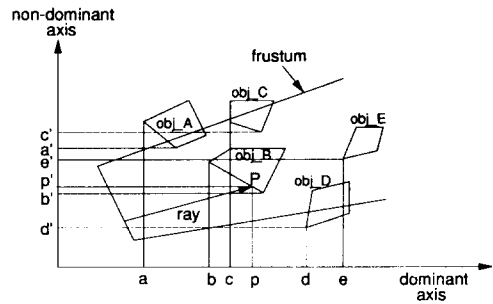


그림 5. 주축 방향으로의 정돈된 목록의 이용과 종축 방향으로의 최대 및 최소 좌표값의 이용  
 Fig. 5. Usage of sorted list along the dominant axis and extreme values along the non-dominant axes.

(E) 종축 방향로의 제거

2개의 종축 방향으로의 제거는 다음과 같다. 어떤 물체의 종축 성분의 '최소값'이 지금까지 계산된 가장 가까운 교차점의 같은 종축 성분보다 '더 큰' 경우 그 물체를 무시할 수 있다. 그림 5에서 'obj-C'는 종축 방향 성분의 최소값 'c'가 교차점 'P'의 같은 종축 방향 성분인 'p'보다 더 큰 곳에 있기 때문에 교차 계산을 하지 않는다. 따라서 그림 5에서 처음 두 물체 'obj-A'와 'obj-B'만 교차 계산을 하게 된다. 주축과 종축 성분이 '음(-)'인 경우(-x, -y, -z)에는 위의 설명에서 '최소값'과 '더 큰'이 '최대값'과 '더 작은'으로 바뀐다.

IV. 그림자 계산의 가속 방안

1. 음영 피라미드의 정의

음영 피라미드는 광선이 만난 다각형 물체 표면으로부터 광원으로 향하는 모든 음영 광선을 포함하는 체적으로 정의 되는데, 다각형을 포함하는 평면의 윗쪽에 있는 광원에 대해서만 하나씩 만들어진다. 광원이 다각형 평면의 아래쪽에 있으면 다각형 표면의 모든 점들은 해당 광원에 대해 항상 그림자 영역에 있기 때문이다. 음영 피라미드의 꼭지점은 광원의 위치가되며 음영 피라미드의 밑면은 해당 다각형 자체가 된다. 그림 6에서 'L<sub>1</sub>'과 'L<sub>2</sub>'는 다각형 평면의 위쪽에 위치한 광원을 나타내고 점찍힌 부분이 그들의 음영 피라미드를 나타낸다. 다각형 평면의 아래쪽에 있는 광원 'L<sub>3</sub>'에 대해서는 음영 피라미드를 만들지 않는다.

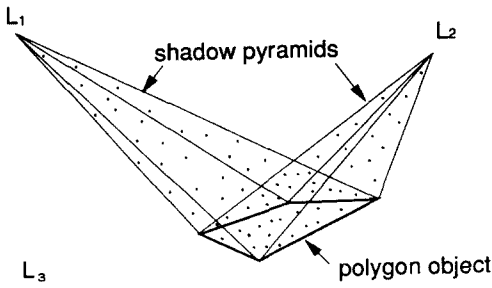


그림 6. 음영 피라미드의 정의  
Fig. 6. Definition of shadow pyramids.

2. 피라미드 내에있는 물체의 등록

피라미드 안으로의 물체의 등록은 절두체 내로의 등록과 비슷하다. Semi-octree로 나누어진 분할 공간의 계층구조를 따라 내려오면서, 분할공간(여기서의 분할 공간도 모두 semi-octree에서의 분할 공간을 의미)의 포함 여부를 먼저 수행하고 피라미드와 만나는 분할 공간내에 있는 물체와의 포함 여부를 수행한다. 분할 공간(또는 물체)과의 포함검사도 다각형 평면으로의 제거, 피라미드 옆면으로의 제거, 그리고 인접 옆면의 접평면으로의 제거를 차례로 수행하여 남아있는 물체를 등록한다. 등록되려는 물체가 다각형인 경우에는 그의 법선 방향과 음영 피라미드를 만드는 다각형의 법선 방향과의 내적의 부호가 '음(-)'인 경우(서로 마주보는 경우)에만 등록되므로 실제 등록되는 다각형 물체의 갯수는 절두체에서와 마찬가지로 실제 존재하는 물체 갯수의 절반 가량 되며, 음영 피라미드의 실제적인 체적도 정의된 체적과 물체 공간의 MBB와의 공통 부분이다.

3. 피라미드를 이용한 그림자 계산

광선 추적 과정에서 광선과 만난 다각형 물체와의 교차점에서 음영 피라미드내의 물체를 바로 찾아 교차 여부를 검사함으로써 그 점이 해당 광원에 대해 그림자 영역에 있는가를 알아낸다. 이는 어떤 다각형 물체의 음영 피라미드내에 등록되지 않은 물체들은 해당 광원에 대하여 그 다각형 물체에 그림자를 만들지 않기 때문이다. 대부분의 경우 음영 광선이 3-D DDA 방식으로 방문한 분할 공간들의 체적의 합이 음영 피라미드의 실제적인 체적보다 클때가 많고 이런 경우 방문한 분할 공간내에 등록된 물체의 갯수도 음영 피라미드내의 물체의 갯수보다 많게 되므로, 음영 피라미드를 사용한 경우 그림자 계산을 위한 교차 계산의 수는 줄어든다. 또한 2차 광선 추적에서와 같이 분할 공간을 지나갈 때 드는 비용이 전혀 없으므로 계산시간을 크게 줄일 수 있다. 그림

자 계산 알고리즘의 성능은 광원의 위치에 따라 크게 달라지는데, 대부분의 경우 광원이 물체 공간의 MBB에 있는 경우 그림자 계산이 어려워진다.<sup>[11,12]</sup> 반면에 광원의 위치가 멀리 떨어져 있는 경우 사용하기 곤란한 알고리즘도 있다.<sup>[8]</sup> 음영 피라미드를 썼을 경우에는 광원이 물체 공간의 MBB 내부에 있으면 음영피라미드의 체적이 작아져 등록되는 물체의 갯수가 적어지므로 유리하고, 광원이 멀리 있는 경우에는 음영 피라미드의 체적이 커지기는 하지만 물체 공간 MBB와의 공통 부분인 그의 실제적인 체적은 거의 늘지 않으므로 등록되는 물체의 갯수는 크게 변하지 않는다.

V. 구간 일관성의 이용

주사선의 어느 구간내의 인접 광선들과 어떤 물체와의 교차 검사의 결과가 같을 확률이 높을때 '구간 일관성'이 있다고 하는데, 교차 계산의 갯수를 줄이기 위한 구간 일관성을 절두체와 피라미드에서 찾을 수 있다. 같은 주사선상의 어느 구간내에서 근원 광선이 같은 다각형 물체를 만난다고 하면, 그 구간내의 2차광선은 같은 절두체내의 물체들과 교차 계산을 하게된다. 모든 물체가 불투과하다고 할 때 구간내의 한 화소를 통과한 광선의 2차 광선과 만난 물체가 다음번의 화소를 통과한 광선의 2차 광선에 대해서는 교차가 이루어지지 않는 경우, 이 물체는 구간내의 마지막 화소를 통하는 광선의 2차 광선까지에 대해서 절대로 교차가 일어나지 않는다. 따라서 현재의 화소를 지난 광선의 2차 광선이 만난 물체가 다음의 인접 화소를 지난 광선의 2차 광선과 만나지 않으면 이 물체를 '제거 목록'에 등록시켜두고 나머지 구간내의 2차 광선과의 교차 계산을 하지 않도록 한다. 즉, 교차 계산을 하려는 물체가 제거 목록에 있으면 교차 계산을 하지 않고 절두체내에 등록된 다음 물체를 검사한다. 제거 목록은 근원 광선이 만난 물체가 달라지게 되는 경우에 지운다. 그림 7은 반사 절두체 내에서의 구간 일관성을 이용한 것을 보인 것인데 ray<sub>1</sub>에서 ray<sub>6</sub>까지의 광선들은 눈으로부터 출발한 근원 광선이 동일한 반사 다각형을 만나서 생긴 반사 광선들을 나타내고 있다.

이전의 광선(ray<sub>1</sub>과 ray<sub>2</sub>)이 만났던 물체 obj<sub>1</sub>을 다음번의 인접 반사광선(ray<sub>3</sub>)이 만나지 않게 되면 '이 물체는 나머지 반사 광선들(ray<sub>4</sub>에서 ray<sub>6</sub>까지)과는 더 이상 교차하지 않을 것이 분명하므로 이 물체(obj<sub>1</sub>)를 제거 목록에 등록시켜두고 같은 주사선의 나머지 인접 반사 광선에 대해서 더 이상 교차 계산

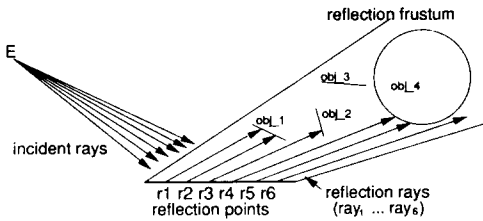


그림 7. 반사 절두체 내에서의 구간 일관성  
 Fig. 7. Span coherency within a reflection frustum.

을 하지 않게 한다. 마찬가지로 obj\_2는 ray<sub>i</sub>가 검사된 후 제거 목록에 추가로 등록되어 ray<sub>s</sub>, ray<sub>e</sub>에 대해서 교차 계산을 하지 않는다.

음영 피라미드에서도 절두체에서와 비슷한 방법으로 제거 목록을 이용할 수 있으며, 음영 광선에 대해서는 현재 그림자를 지게하는 물체가 인접 광선에 대해서도 그림자를 드리울 확률이 높기 때문에 각 광원에 대해 그림자를 드리웠던 물체를 저장해 두고 그림자 검사시 각 광원에 대해 이들을 먼저 조사해봄으로써 그림자 계산을 가속화 시킬 수 있다.

VI. 결 과

본 논문에서는 C언어를 SUN3/75 시스템의 UNIX 환경하에서 사용하여 이미지를 만들었다. 측정된 시간 데이터는 512×480의 화면에 대한 것이다. 첨부한 그림들은 1024×768의 화면에 anti-aliasing이 처리된 그림을 사진기로 찍은 것이다. 제안한 절두체와 피라미드를 사용한 광선 추적 방법과 균일 분할된 공간을 광선이 지나가면서 교차 계산을 수행하는 기존의 광선 추적 방법을 비교하였다. 그림 8의 'star' 이미지는 512개의 다각형 물체 (400개의 반사체 포함)를 3개 광원의 대해 묘화시킨 것이고, 그림 9의 'card' 이미지는 556개의 다각형 물체 (440개의 반사체 포함)를 5개의 광원에 대해 묘화 시킨 것이다. 그림 10의 'kitchen' 이미지는 686개의 다각형 물체를 2개의 광원에 대해 묘화한 것이고 그림 11의 'kaist' 이미지는 981개의 다각형 물체를 4개의 광원에 대해 묘화시킨 것이다. 표 1은 각 그림에 대한 반사 절두체와 음영 피라미드의 갯수와 그들에 등록된 물체의 평균 갯수를 나타내고 있다. 표 2는 광선이 균일 분할된 공간을 진행하는 방식을 사용할 때의 데이터를 보이고 있고 표 3은 제안된 방법을 사용했을 경우의 데이터를 보이고 있다. 두 방법 모두 그림자를 드

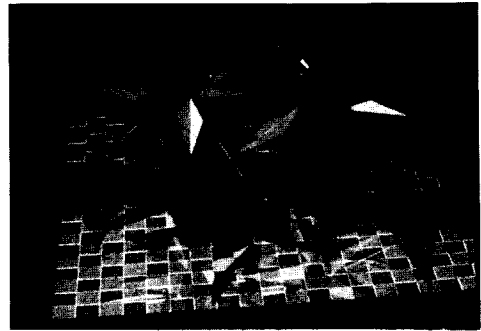


그림 8. 'star' 이미지  
 Fig. 8. 'star' image.

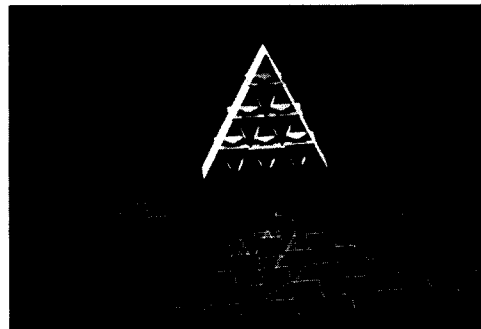


그림 9. 'card' 이미지  
 Fig. 9. 'card' image.

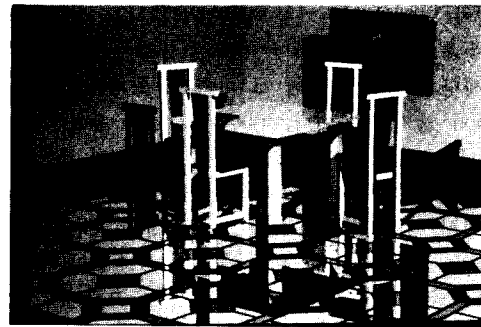


그림 10. 'kitchen' 이미지  
 Fig. 10. 'kitchen' image.



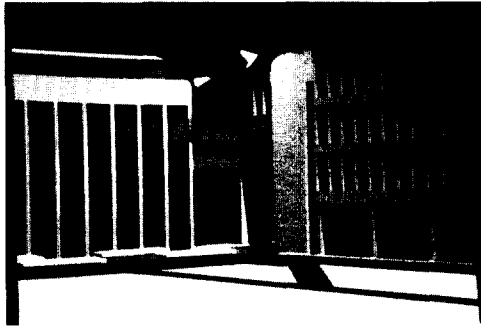


그림 11. 'kaist' 이미지  
Fig. 11. 'kaist' image.

리웠던 물체를 먼저 검사 하는 기법을 사용했을 때의 광선과의 교차 계산 횟수, 음영 광선과의 교차 계산 횟수, DDA 초기화 횟수, DDA로써 분할 공간을 방문한 횟수, 그리고 색 묘화 시간, 그림자 계산시간 및 총 묘화시간을 나타낸 시간 데이터를 보이고 있다. 교차 계산의 감소 효과는 음영 피라미드의 경우

두드러지게 나타났는데 퍼져나가는 모양의 절두체내에 등록된 물체의 갯수가 광원을 향해 좁아지는 모양의 음영 피라미드에 등록된 물체의 갯수보다 많기 때문이다. 절두체를 사용하였을때 교차 계산 횟수가 크게 줄지 않은 또 다른 이유는 절두체는 2차 광선 추적에만 사용되기 때문이다. 그러나 물체의 등록에 필요한 시간이 있음에도 불구하고 DDA의 초기화 및 다음 분할 공간으로의 진행에 드는 시간을 제거한 결과 묘화 시간이 다소 단축되었다. 음영 피라미드는 모든 교차점에서의 음영 검사에 사용되고 음영 피라미드에 등록된 물체의 갯수가 적기 때문에 그림자 계산이 크게 단축되었다. 그림 12는 네 이미지에 대한 두 방법간의 CPU 시간 비교를 막대 그래프로 표시한 것이며 표 4는 SUN 3/75로써 DDA초기화, DDA의 진행, 광선과 물체와의 교차 계산을 각각 100,000번 수행하는데 드는 CPU시간을 나타내고 있다.

Ⅶ. 결 론

2 차 광선 추적 과정과 그림자 계산을 가속화 시킨 광선 추적 프로그램을 개발하였다. 어떤 다각형 물체에 대하여 한 번 만들어진 절두체내의 물체 목록

표 1. 각 이미지에 대한 절두체와 음영 피라미드에 관한 데이터  
Table 1. Data of scenes including frusta and pyramids.

scenes	objects	lights	frusta	avrg. # of objects registered in frustum	pyramids	avrg. # of objects registered in pyramid
star	512	3	344	2.1	1933	2.5
card	556	5	325	3.9	1894	2.0
kitchen	686	2	220	8.9	768	4.3
kaist	981	4	38	20.3	743	15.5

표 2. 균일 분할된 공간을 광선이 진행하면서 교차 계산을 수행하는 기존의 방법을 사용했을 때의 데이터

Table 2. Data from the conventional cell traversal algorithm under the uniform subdivision environment.

scenes	# of color intersections	# of shadow intersections	# of DDA initializations	# of DDA propagations	rendering time	shadow time	total time
star	1,289,884	2,403,638	1,228,840	21,152,091	38: 41	58: 20	1: 37: 01
card	1,424,019	2,962,314	1,312,714	28,587,689	39: 59	1: 16: 52	1: 56: 51
kitchen	2,368,311	2,191,550	980,604	27,279,896	46: 34	1: 09: 53	1: 56: 27
kaist	1,981,756	6,931,243	1,166,316	23,799,582	54: 46	1: 39: 05	2: 33: 51

표 3. 제안한 방법을 썼을 때의 데이터

Table 3. Data from the proposed algorithm.

scenes	# of color intersections	# of shadow intersections	# of DDA initializations	# of DDA propagations	rendering time	shadow time	total time
star	926,631	1,734,266	350,893	3,762,851	33: 23	17: 51	51: 14
crad	1,296,963	670,112	329,112	3,651,930	30: 09	11: 51	42: 00
kitchen	2,266,901	932,315	319,846	6,167,740	42: 37	10: 20	52: 57
kaist	2,142,667	6,837,388	275,870	2,146,565	54: 29	47: 48	1: 42: 17

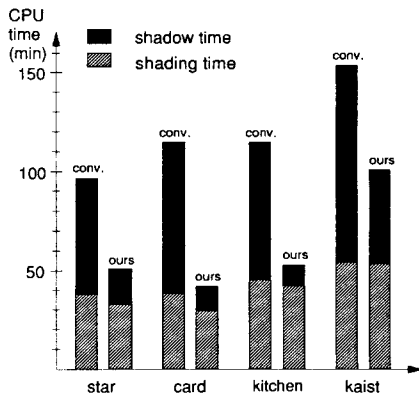


그림 12 균일 분할된 공간을 광선이 진행하면서 교차 계산을 수행하는 기존의 방법과 제안된 방법과의 네 이미지에 대한 CPU 시간의 비교

Fig. 12. Comparison of CPU times between the conventional cell traversal scheme under the uniform subdivision environment and proposed secondary for four example images.

표 4. SUN 3/75에서 100,000번 수행 시켰을 때 드는 CPU 시간

Table 4. CPU time of SUN 3/75 needed for 100,000 operations [sec].

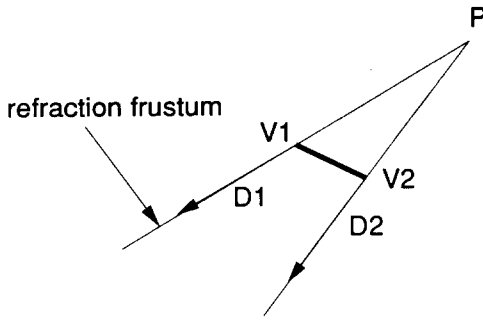
DDA initialization	38
DDA activation	7
Intersection calculation	51

록을 그 물체를 만나는 다른 근원 광선의 2차 광선들에 대해서도 이용하게함으로써 분할 공간을 진행하는데 드는 비용을 제거하였다. 등록된 물체들중에서 질두체의 주축 방향과 종축 방향으로 보았을 때 그의 예상되는 교차점이 지금까지 발견된 가장 가까운 교차점보다 멀리 있을 것이 분명한 물체에 대해서는 불필요한 교차 계산을 피하게 하고 같은 주사선상의 어느 구간내의 인접광선에 대하여 더 이상 교차 계산이 일어나지 않을 물체들을 제거 목록에 등록시킴으로써 교차 계산 횟수를 줄일 수 있었다. 음영 피라미드 방법도 어떤 물체에 대해 한 번 만들어둔 그림자를 지게할 가능성이 있는 물체 목록은 그 물체 표면상의 모든 교차점에 대해서도 음영 검사시 이용하게 함으로써 교차 계산을 수행할 물체를 빨리 찾아 분할 공간을 찾는데 드는 비용을 제거하였고 음영 피라미드 모양이 좁고 단축이므로 등록된 물체의 갯수도 작기때문에 교차 계산 횟수도 크게 감소되어 기존의 방법에 비해 여러 배 빠른 음영 검사를 수행할 수 있었다. 음영 피라미드는 광원이 멀고 가까움에 구애 받지 않고 눈의 위치에 무관하게 만들어지는 것이 특징이며, 눈이 움직이는 애니메이션의 경우 음영 피라미드 방법은 효율적으로 사용될 수 있는데 각 장면 마다 추가적으로 만들어져야 할 음영 피라미드의 수가 많지않고 한번 만들어진 음영 피라미드에 등록된 물체 목록은 계속적으로 이용될 수 있기 때문이다.

부 록

\*굴절 질두체의 가상 꼭지점의 계산

그림 A.1과 같이  $V_1(a, b, c)$ 과  $V_2(d, e, f)$ 를 다각형의 한 면의 두 끝점이라고 하고  $D_1 = l_1i + m_1j + n_1k$ 과  $D_2 = l_2i + m_2j + n_2k$ 를 해당 끝점의 방향 벡터라고 할때 가



그림A. 1. 지나는 점의 위치와 방향 벡터를 알고 있는 두 직선의 가상 꼭지점을 찾는 계산

Fig. A. 1. Computation of the apex point from the two line equations. Each line passes through a point and has a directional vector.

상 꼭지점  $P(p_x, p_y, p_z)$ 를 찾고자 한다. 끝점을 지나면서 방향 벡터를 갖는 두 직선의 방정식을 표준형을 표현하면 :

$$\frac{x-a}{l_1} = \frac{y-b}{m_1} = \frac{z-c}{n_1} \quad (\text{A. 1})$$

$$\frac{x-d}{l_2} = \frac{y-e}{m_2} = \frac{z-f}{n_2} \quad (\text{A. 2})$$

그런데, 가상 꼭지점은 두 직선상에 존재하므로 꼭지점의 좌표를 (A. 1), (A. 2)에 대입하면

$$\frac{p_x-a}{l_1} = \frac{p_y-b}{m_1} = \frac{p_z-c}{n_1} \quad (\text{A. 3})$$

$$\frac{p_x-d}{l_2} = \frac{p_y-e}{m_2} = \frac{p_z-f}{n_2} \quad (\text{A. 4})$$

(A. 3), (A. 4)으로부터 다음의 네 방정식을 유도할 수 있다.

$$p_x-a = \frac{l_1}{m_1}(p_y-b) \quad (\text{A. 5})$$

$$p_x-a = \frac{l_1}{n_1}(p_z-c) \quad (\text{A. 6})$$

$$p_x-d = \frac{l_2}{m_2}(p_y-e) \quad (\text{A. 7})$$

$$p_x-d = \frac{l_2}{n_2}(p_z-f) \quad (\text{A. 8})$$

(A. 5), (A. 6)의 우측과 (A. 7), (A. 8)의 우측으로부터

$$\frac{l_1}{m_1}(p_y-b) = \frac{l_1}{n_1}(p_z-c) \quad (\text{A. 9})$$

$$\frac{l_2}{m_2}(p_y-e) = \frac{l_2}{n_2}(p_z-f) \quad (\text{A. 10})$$

(A. 9), (A. 10)으로부터  $p_z$ 를 유도하면

$$p_z = \frac{n_1 n_2 (b-e) + n_1 m_2 f - n_2 m_1 c}{n_1 m_2 - n_2 m_1} \quad (\text{A. 11})$$

(A. 1)으로부터  $p_y$ 와  $p_x$ 를 다음과 같이 유도할 수 있다.

$$p_y = \frac{m_1}{n_1}(p_z-c) + b \quad (\text{A. 12})$$

$$p_x = \frac{l_1}{n_1}(p_z-c) + a \quad (\text{A. 13})$$

## 參 考 文 獻

- [1] T. Whitted, "An improved illumination model for shaded display," *Comm, ACM* 23, 6, pp. 343-349, June 1980.
- [2] J. Goldsmith and J. Salmon, "Automatic creation of object hierarchies for ray tracing," *IEEE CG&A*, pp. 14-20, May 1987.
- [3] T.L. Kay and J.T. Kajiya, "Ray tracing complex scenes," *Computer Graphics (Proc. SIGGRAPH 86)*, pp. 269-278.
- [4] A.S. Glassner, "Space subdivision for fast ray tracing," *IEEE CG&A*, pp. 15-22, Oct. 1984.
- [5] A. Fujimoto, T. Tanaka and K. Iwata, "ARTS: accelerated ray tracing system," *IEEE CG&A*, pp. 16-26, Apr. 1986.
- [6] M. Shinya, T. Takahashi, "Principles and applications of pencil tracing," *Computer Graphics (Proc. SIGGRAPH 87)*, pp. 45-52.
- [7] P. Berger, "A general version of crow's shadow volumes," *IEEE CG&A*, pp. 17-28, Sep. 1986.
- [8] E.A. Haines and D.P. Greenberg, "The light buffer: a shadow testing accelerator," *IEEE CG&A*, pp. 6-16, Sep. 1986.
- [9] D. March, "Ug Ray: an efficient ray-tracing renderer for unigrafix," Report No. UCB/CSD 87/360, Computer Science Division University of California Berkeley, May 1987.
- [10] P.S. Heckbert, "Derivation of refraction

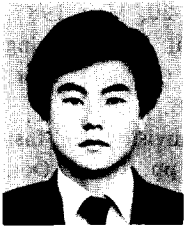
formulas," SIGGRAPH 87 Course notes #13, Introduction to Ray Tracing.

- [11] L. Williams, "Casting curved shadow on curved surfaces," *Computer Graphics (Proc. SIGGRAPH 78)*, pp. 270-274.
- [12] P. Atherton, K. Weiler and D.P. Greenberg, "Polygon shadow generation," *Computer Graphics (Proc. SIGGRAPH 78)*, pp. 275-281. \*

---

著 者 紹 介

---



崔 勳 奎(正會員)

1960年 8月 23日生. 1984年 2月 고려대학교 전자공학과 졸업. 1986年 2月 한국과학기술원 전기 및 전자공학과 석사학위 취득. 1986年 3月 ~ 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중. 주

관심분야는 Computer Graphics의 Software 가속화 Algorithm 등임.

慶 宗 旻(正會員)

1953年 6月 21日生. 1975年 서울대학교 전자공학과 공학사학위 취득. 1977年 한국과학기술원 전기 및 전자공학과 석사학위 취득. 1981年 한국과학기술원 전자공학과 박사학위 취득. 현재 한국과학기술원 전기 및 전자공학과 부교수. 주 관심분야는 CAD, Computer Graphics, Neural Net 등임.