

## 시뮬레이트된 방사선 사진에서 엑스선과 물체의 교차점 계산을 위한 효율적인 알고리즘의 개발

오 재 윤\*

### Development of an Efficient Algorithm for the Intersection Calculations in a Simulated Radiograph

Chae-youn Oh\*

#### ABSTRACT

This paper develops an algorithm for efficiently computing the intersection points between rays and an object in a simulated radiograph. This algorithm allows interactive calculation of simulated radiographs for very complex parts. It needs a geometric model of a part which is approximated by a bounding surface made up of flat triangular polygons. Since rays have a point source, a perspective transformation is applied to convert the point source problem to one that has parallel rays. This permits to use a scan-line algorithm which utilizes the coherence of the grid of rays for the intersection calculations. The efficiency of the algorithm is shown by comparing compute time of the intersection calculations to a commercial software that computes each ray intersection independently.

#### 1. 서 론

방사선 검사법(radiographic inspection)은 널리 사용되는 비파괴시험 방법이다[1,2]. 이것은 엑스선 방사원(source)과 필름 사이에 부품을 놓고서 엑스선을 투과시켜 부품을 검사하는 방법이다. 엑스선이 부품을 통과할때 에너지가 흡수되는데 이 에너지의 흡수량은 부품의 두께와 재질에 의존한다. 이같이 부품을 통과하는 엑스선은 필름 위에 방사선 사진(radiograph)이라고 알려진 상을 만든다. 이 방사선 사진은 부품의 원재료에 포함되어 있거나 혹은 부품을 만드는 과정에서 발생할 가능성이 있는 결함들을 확인하기 위해 사용된다. 이때 야기되는 어려움은 주어진 엑스선 방사원 및 필름의 위치와 엑스선의 노출양에 대해 어떤 종류의 결함과 어떤

위치에 있는 결함을 발견할 수 있는지를 아는 것이다.

최근에 많은 연구원들은 컴퓨터를 사용해서 방사선 사진을 시뮬레이션 하기 위한 방법들을 개발해 왔다. 삼차원 입체모델(solid model)를 사용해서 에너지의 흡수량을 계산하고 시뮬레이트된 방사선 사진을 만든다. 방사선 사진 시뮬레이션은 다음 두가지 목적을 가지고 행해진다. 하나의 목적은, 어느 부품에 존재할 수 있는 결함을 발견할 수 있는 최대의 가능성을 가지는 시험절차를 개발하는데 사용된다. 다른 목적은, 부품들을 엑스선에 의한 비파괴 검사가 가능하게끔 설계하는데 사용된다. 여기서, "검사할 수 있다"라는 것은 중요한 위치 혹은 발견하기 어려운 위치에 있을 수 있는 어떤 유형의 결함들도 확인 할 수 있다라는 것을 의미한다[3-6].

방사선 사진 시뮬레이션 과정에서 어려움은 주로 가

\* 전북대학교 공과대학 정밀기계공학과 (정희원)

상의 엑스선이 부품의 경계선을 통과할때 엑스선과 부품의 교차점을 찾는 것이다. 이같은 교차점을 알면 주어진 엑스선의 경로에 대한 에너지의 흡수량을 계산할 수 있고 엑스선이 필름에 교차하는 점에서 필름이나 방사선 사진에 나타나게 되는 엑스선의 강도값을 정할 수 있다. 위의 과정은 단지 한점에 관한 것이기 때문에 이 과정은 방사선 사진을 나타내기 위해 충분한 해상도(resolution)를 가지는 격자점(grid point)들에 대해서 반복해서 수행되어야 한다.

본 논문은 아주 복잡한 부품들의 시뮬레이트된 방사선 사진을 대화식(interactive)으로 계산할 수 있게 하는 엑스선과 부품 사이의 교차점들을 주사선 알고리즘(scan-line algorithm)과 원근 변환(perspective transformation)을 동시에 사용해서 효율적으로 계산하기 위한 방법을 제시한다. 이같은 교차점들을 계산하기 위한 첫번째 단계로 어느 부품의 기하학적 형상 모델이 평평한 삼각형들로 구성된 경계면(bounding surface)으로 근사하게 나타내진다. 이것은 단속적으로 선형(piecewise linear)인 기하학적 형상을 정의한다. 그다음, 컴퓨터 그래픽스 연구에서 개발된 주사선(scan-line) 방법을 기초로 하는 각기 엑스선의 교차점들을 순차적으로 계산함으로써 효율성을 가지는 새로운 알고리즘을 개발한다. 이 알고리즘은 현 엑스선이 어느 물체와 교차하는 점들을 계산하는데 도움을 주기 위해 지난번 엑스선으로 부터 계산한 교차점들을 사용한다.

주사선(scan-line) 알고리즘은 어느 물체의 보이지 않는 부분을 제거하기 위해 사용된 알고리즘들 중의 하나이다[7]. 이 알고리즘은 물체의 공간적 일관성(spatial coherence)을 사용하는 잇점을 가진다. 즉, 어느 물체의 보이는 부분을 나타내기 위한 계산을 절대적이지 아닌 증가적으로 수행하는 것이다[8]. 이같은 잇점을 가지는 주사선 알고리즘은 아주 효율적이어서 매개변수를 이용하거나 다각형을 이용해서 정의한 물체의 보이지 않는 부분을 제거하고 [7-10], 물체 표면 음영(shading)을 나타내기 위한 법선벡터를 계산 하는데 [11, 12] 널리 사용된다. Atherton[13]은 CSG(constructive solid geometry)를 이용해서 정의된 물체의 보이는 부분을 나타내기 위해 주사선 알고리즘을 응용했다.

## 2. 주사선(scan-line) 알고리즘의 개요

먼저 어느 물체가 경계면들을 사용해서 모델되어지고,

물체 경계면은 다각형을 이용해서 모델되어진다고 가정한다. Fig. 1은 다각형들(polygons)과 주사선들(scan-lines) 및 주사점들(scan points)의 예를 보여준다.

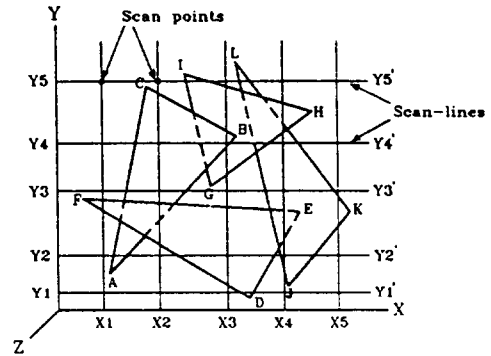


Fig. 1 Polygons, scan-lines and scan points

이 그림은 ABC, DEF, GHI, JKL의 네개의 다각형을 가진다. 각기의 다각형은 세개의 모서리(edge)로 구성된다. 예를 들어, 다각형 ABC는 AB, BC, CA의 세개의 모서리를 가진다. 일정한 Y값을 가지는 선이 Y주사선이다. Fig.1에서 Y1-Y1', Y2-Y2', Y3-Y3', Y4-Y4', Y5-Y5' 같은 선들이 Y주사선들이다.

주사선 알고리즘은 통상적인 TV 화면 주사(raster scanning)와 같이 어느 물체를 왼쪽에서 오른쪽으로, 아래에서 위로 주사(scanning) 함에 의해 그 물체의 보이는 부분을 나타낸다. 이 알고리즘을 응용하는 곳에 따라 주사하는 순서는 바뀔 수 있다. 그래서 이 알고리즘은 기본적으로 두개의 큰 고리(loop)로 구성되어 있다. 하나의 고리는 주사선을 아래에서부터 위로 움직이게 하고(Y 좌표 주사) 다른 고리는 각기의 주사선 위에서 투시선(sight-line)을 왼쪽으로부터 오른쪽으로 움직이게 하는(X 좌표 주사) 것이다. 여기서 투시선은 현재의 주사점(X, Y)에서 X-Y 평면에 수직인 선이다. X좌표 주사 과정에서, 투시선들(혹은 엑스선들)은 서로 평행이라고 가정한다. 즉 각기의 엑스선은 Z축에 평행하다고 가정한다.

이 알고리즘의 첫번째 단계는 모든 다각형의 꼭지점들의 Y값들을 작은 값에서 큰 값순으로 정리하는 것이다. 이때 Y주사선은 꼭지점들의 가장 작은 Y값으로부터 즉, Y값 목록(list)의 맨 아래 행으로부터 한번에 하나의 주사선씩 위로 움직인다. 현재의 Y주사선과 교차하는 모든 모서리를 포함하는 AET(active edge table)이 Y값 목록의 맨 아래 행에 대해 만들어진 다음 Y주사선이 위로 움직임에 따라 갱신된다. 예를 들어, Fig.

1에서 Y주사선 Y2-Y2'은 CA, AB, FD, DE, LJ, JK의 여섯개의 active edges들로 구성되는 AET를 가진다.

현재의 주사선과 active edges들과 교차점들에서 X값이 계산된 다음 이 X값들이 작은 값에서 큰 값순으로 정렬된다. X좌표 주사는 이같이 정렬된 X값 목록의 가장 작은 X값으로부터, 즉 주사선의 왼쪽 끝으로부터 시작해서 X값 목록의 가장 큰 X값에 달할 때까지 현재의 주사선을 따라 오른쪽으로 움직인다.

APT(active polygon table)은 맨처음 주사점에 대해 만들어진 다음 주사점이 오른쪽으로 움직임에 따라 갱신된다. APT는 현재의 주사점을 통과하는 투시선과 교차하는 모든 다각형들을 수록한다. 예를 들어, Fig.1에서 주사점 (X3, Y4)를 통과하는 투시선은 ABC, GHI의 두개의 active polygons들로 구성되는 APT를 가진다. 이 APT를 가지고 현재의 주사점에서 어느 다각형이 보이는지를 정하기 위해 깊이 비교(depth comparison)가 행해진다. APT의 갱신과 깊이 비교는 주사점이 X값 목록의 가장 큰 X값에 달할 때까지 한번에 하나의 주사점씩 오른쪽으로 옮겨감에 따라 되풀이된다.

만약에 주사점이 X값 목록의 가장 큰 X값을 지나면 주사선이 한단계 위로 옮겨가고 X좌표 주사와 깊이 비교가 되풀이된다. 이 주사선은 꼭지점의 Y값 목록에서 가장 큰 Y값을 주사선이 통과 할때 까지 계속해서 한번에 한단계씩 위로 옮겨간다.

### 3. 물체의 원근 변화 (Perspective transformation)

앞절에서 설명한 주사선(scan-line) 알고리즘은 모든 엑스선들이 서로 평행이라는 가정하에 수행되었다. 엑스선 방사법은 무한히 멀리 떨어진 점에 놓인 방사원(Source)을 가지는 것이 아니라 점 방사원(Point Source)을 가진다. 이 연구의 주된 공헌중의 하나는 원근 변환을 응용해서 점 방사원 문제를 평행한 엑스선들을 가지는 문제로 변환시켜서 앞절에서 기술한 주사선 알고리즘을 이용할 수 있게 하는데 있다. 원래의 물체에 점 방사원으로 부터 나오는 엑스선들을 가해서 얻어지는 방사선 사진과 같은 방사선 사진을 평행한 엑스선들을 가했을때 얻어지게끔 하기 위해서는 필수적으로 물체의 형태를 변화시켜야만 할 것이다. 이같이 원근 변환된

물체는 앞절에서 기술한 주사선 알고리즘을 사용해서 교차점을 계산할 수 있게 한다.

우리는 한개의 엑스선 방사원을 가지기 때문에 단일점 원근 변환을 수행한다. 이같은 변환을 수행하기 위하여 다음 세가지의 가정을 한다. 첫번째 가정은 엑스선 방사원이 양의 Z축 위에 놓인다는 것이다. 두번째 가정은 엑스선 필름이 X-Y평면 위에 놓이고 이필름은 컴퓨터 그래픽스 터미널의 소자(Pixel)와 유사한 격자로 구성된다는 것이다. 마지막 가정은 물체의 중심이 좌표계의 중심과 일치 한다는 것이다.

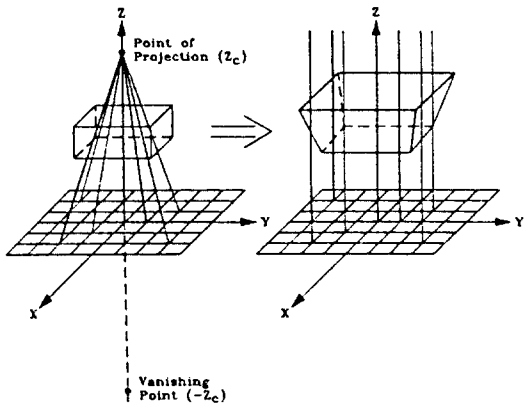


Fig. 2 A schematic diagram of the single point perspective transformation

단일점 원근 변환의 도식적인 선도가 Fig. 2에서 보여진다. Fig. 2a)는 원근 변환이 수행되기 전의 물체를 보여주고 Fig. 2b)는 원근 변환이 수행된 후의 물체를 보여준다. 엑스선 방사원이 놓이는 위치인 Z가 투영 중심(projection center)이 된다. 따라서 소멸점(vanishing point)은 Z축상에서 투영 중심과 거리가 같고 반대지점인 -Z에 놓인다. Z는 단일점 원근 계수 Zr과 다음과 같은 관계를 가진다.

$$Z_c = -\frac{1}{Z_r} \tag{1}$$

단일점 원근 변환은 다음의 행렬 연산에 의해 수행된다 [14].

$$\begin{aligned} \begin{bmatrix} X & Y & Z & H \end{bmatrix} &= \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z_r \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} \\ &= \begin{bmatrix} X & Y & Z & (Z_r \times Z + 1) \end{bmatrix} \tag{3} \end{aligned}$$

여기서  $X, Y, Z$ 는 원래 좌표계에서의 좌표이다. 변환된 좌표들은 식(3)에서  $X, Y, Z$ 를  $H(=Z \times Z + 1)$ 로 나누면 주어지고 다음과 같이 표현된다.

$$\begin{bmatrix} X^* & Y^* & Z^* & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \quad (4)$$

모든 다각형들의 꼭지점( $X, Y, Z$ ) 좌표값은 식(4)를 이용해서 변환된다. 그다음, 앞절에서 기술한 일반적인 주사선 알고리즘이 교차점들을 계산하기 위해 이용된다.

원근 변환된 물체를 가지고 엑스선이 물체를 통과할때의 교차점들을 구한 다음 이 교차점들의 원래 물체에 상응하는 좌표들을 구해야 한다. 변환된 물체에서 교차점들  $(X_i^*, Y_i^*, Z_i^*)$ 에서 가진다고 가정할 때, 이 교차점의 원래 물체에서 좌표는 다음 세개의 방정식을 동시에 풀어서 구한다.

$$X_i - X_i^* (Z_r \times Z_i + 1) = 0 \quad (5)$$

$$Y_i - Y_i^* (Z_r \times Z_i + 1) = 0 \quad (6)$$

$$Z_i - Z_i^* (Z_r \times Z_i + 1) = 0 \quad (7)$$

위의 세개의 방정식을 풀면 다음의 해석적인 해가 주어진다.

$$X_i = \frac{X_i^*}{(1 - Z_i^* \times Z_r)} \quad (8)$$

$$Y_i = \frac{Y_i^*}{(1 - Z_i^* \times Z_r)} \quad (9)$$

$$Z_i = \frac{Z_i^*}{(1 - Z_i^* \times Z_r)} \quad (10)$$

위의 식을 이용해서 구한  $(X_i, Y_i, Z_i)$ 는 원래 물체에서의 교차점 값이다. 식(8)~(10)을 이용해서 변환된 물체에서 교차점들의 원래 물체에 상응하는 좌표들을 구할 수 있다.

#### 4. 상세한 알고리즘 설명

어느 물체의 기하학적 형상 자료 파일(geometric data file)은 상업용 소프트웨어인 I-DEAS를 이용해서 만들어진다. 물체의 기하학적 형상 모델은 다각형을 이용해서 정의된다. 각기의 다각형은 세개의 꼭지점으로 구성된 삼각형으로 이는 다각형의 평평함을 보장하기 위한 것이다. 다음 두개의 제약이 다각형 모델에 가해진다. 첫째로, 각기의 다각형은 면적을 가져야 하는데

이는 세개의 꼭지점이 모두 동일 선상에 놓여서는 안된다는 것을 의미한다. 또한, 어느 물체이든간에 실질적으로 표면에 의해 싸여서 닫히므로, 모델 하고자 하는 물체는 다각형들에 의해 닫혀야 한다.

이같이 준비된 기하학적 모델에 식(2)~(4)를 이용해서 원근 변환을 수행한다. 즉, 기하학적 형상 자료 파일의 모든 다각형의 꼭지점 좌표값이 원근 좌표계에서의 값으로 변환된다. 원근 변환을 수행한 다음 각기 다각형의 법선벡터를 계산한다. 이 법선벡터는 엑스선이 물체와의 교차점에서 물체 안으로 들어가는가 혹은 물체 밖으로 나오는가를 정하기 위해 후에 사용된다.

그다음, 앞의 2절에서 기술한 일반적인 주사선(scan-line) 알고리즘을 이용해서 교차점들을 계산한다. 시뮬레이션된 방사선 사진의 해상도(resolution) 혹은 격자 크기는 POD(possibility of detection) 시뮬레이션 설계자에 의해 정해진다. 여기에는 분명히 계산시간과 격자점 수와의 사이에 상호 보상이 있다. 계산시간은 계산할 엑스선들의 수에 직접적인 관계가 있고 대략 격자점 수에 따라 증가한다.

주사선 알고리즘의 첫번째 단계로 모든 꼭지점의  $Y$  좌표가 작은 값에서 큰 값순으로 정리된다. 이같이 정리된 꼭지점들의  $Y$  좌표 목록의 가장 작은  $Y$ 값과 같거나 이보다 큰 크고 이에 가장 가까운 격자선에서부터  $Y$  좌표 주사는 시작된다. AET(active edge table)이 처음  $Y$ 좌표 주사 단계에서 만들어지고  $Y$  주사선이 위로 옮겨감에 따라 개정된다.

각기의  $Y$ 주사선에 대해  $X$ 좌표 주사를 실시한다.  $X$ 좌표 주사에서는 첫째로 AET에 수록된 모든 active edges와 현  $Y$ 주사선이 교차하는 점들에서  $X$ 값을 계산한다. 이  $X$ 값들을 작은 값에서 큰 값순으로 정리한다. 그리고 이같이 정리된  $X$ 값 목록의 가장 작은  $X$ 값과 같거나 이보다 큰 크고 이에 가장 가까운 격자점에서부터  $X$ 좌표 주사를 시작한다.

현 주사점  $(X, Y)$ 를 통과하는 엑스선과 교차하는 모든 다각형들을 포함하는 APT(active polygon table)이 첫번째 주사 단계에서 만들어지고  $X$ 좌표 주사가 오른쪽으로 진행됨에 따라 개정된다. APT에 수록된 모든 active polygons들이 현 엑스선과 교차하는 점들에서  $Z$ 값을 계산한다. 그리고서, 앞에서 계산된 각기 active polygons의 법선벡터가 엑스선과 같은 방향이면 엑스선은 교차점에서 물체 밖으로 나오는 것이고 만약 반대 방향이면 교차점에서 물체 안으로 들어 간다는 것을 나

타낸다.

이제 각기 엑스선이 물체를 들어가는 점과 나가는 점에서 좌표  $[X, Y, Z]$ 가 구해졌다.  $X$ 와  $Y$ 값은 현 주사점  $[X, Y]$ 좌표로부터 주어지고  $Z$ 좌표는 바로 위의 과정에서 계산된 교차점들에서  $Z$ 값으로부터 주어진다. 그러나 이 값들은 변환된 물체로부터 구해진 것이므로 다음 단계는 각 교차점들이 원래 물체에서 이들에 상응하는 값들을 찾는 것이다. 이 계산에는 식(8)-(10)이 사용된다. 원래 물체로 엑스선이 들어가고 나가는 점에서의 좌표를 가지고 주어진 격자점(주사점)에 대해 엑스선이 주어진 물체를 지나간 총 통과 거리를 계산할 수 있다.

APT은  $X$ 좌표 주사가 오른쪽으로 진행됨에 따라 개정된다. 그리고 교차점에서  $Z$ 값과 엑스선의 총 통과 거리가 주사점이  $X$ 값 목록의 가장 큰  $X$ 값을 지날때까지 각각의  $X$ 좌표 주사단계에 대해 반복해서 계산된다. 그리고서,  $Y$ 주사선이 한단계 위로 옮겨가고, AET이 개정되고,  $X$ 좌표 주사가 반복된다.  $Y$ 좌표 주사는 꼭지점들의 가장 큰  $Y$ 값을  $Y$ 주사선이 지날때까지 계속된다. 즉  $Y$ 주사선이 꼭지점의  $Y$ 값 목록의 맨위의 행을 통과하면 이 알고리즘의 모든 과정은 끝나게 된다.

Fig. 3은 위의 알고리즘의 구조를 도식적으로 보여준다. 이 알고리즘을 실행하기 위한 컴퓨터 프로그램은 컴퓨터 기종의 호환성을 위해 FORTRAN 77 표준언어를 사용해서 배정도(double precision)로 만들어졌다.

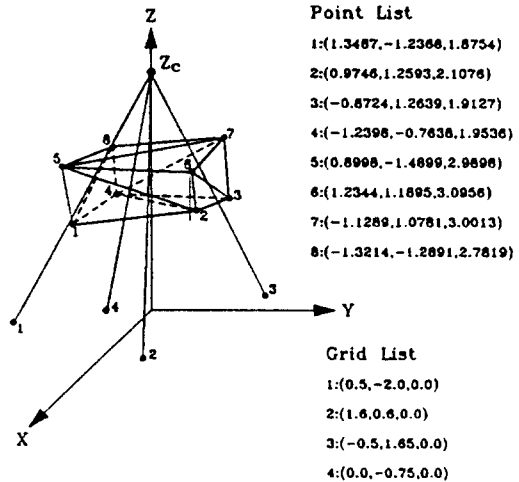


Fig. 4 An example for the algorithm validation

위에서 개발된 알고리즘의 정확도를 검증하기 위해 Fig. 4에서 보여지는 바와 같이 12개의 다각형으로 구성된 육면체 형상의 물체를 이용하였다.  $Z$ 축상의 10단위(unit) 지점에 위치하는 투영 중심(엑스선 방사원)으로부터  $Z$ 좌표가 0인  $X$ - $Y$  평면상에 있는 임의의 네곳에 있는 점들을 통과하는 엑스선들을 고려하였다. 이 육면체의 면들과 엑스선이 교차하는 점들을 각각의 경우에 대해 독립적으로 방정식을 풀어서 구한 교차점 값과 위의 알고리즘을 이용해서 구한 교차점 값을 비교한 결과가 Table 1에서 주어진다. Table 1에서 두번째열은 독립적인 방정식 계산을 수행해서 구한 결과이고 세번째열은 위의 알고리즘을 이용해서 구한 결과이다. Table 1에서 보여지는 바와 같이 본 논문에서 개발한 알고리즘이 아주 정확하다는 것을 알 수 있다.

### 5. 계산시간 비교

본 논문에서 개발된 알고리즘과 상업용 소프트웨어 I-DEAS와의 교차점 계산시간을 비교하기 위해서 구면체가 사용되었다. 이 구면체에 대해 두개의 기하학적 형상 자료 화일이 만들어진다. 하나의 자료 화일은 구

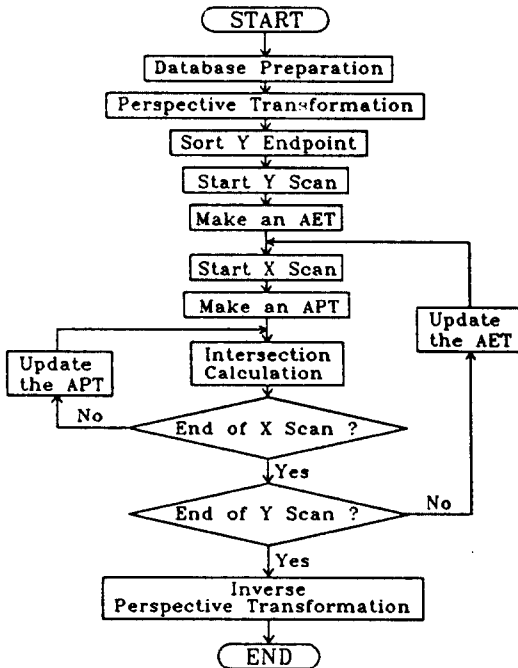


Fig. 3 Flow chart

Table 1. Validation of the algorithm

GRID #	Analytical Solution	Ray-Intersection
1	$X_1=0.35295$ $Y_1=-1.41179$ $Z_1=2.94104$	$X_1=0.35295$ $Y_1=-1.41179$ $Z_1=2.94104$
	$X_2=0.35620$ $Y_2=-1.42479$ $Z_2=2.87605$	$X_2=0.35620$ $Y_2=-1.42479$ $Z_2=2.87605$
2	$X_1=1.12024$ $Y_1=0.42009$ $Z_1=2.99851$	$X_1=1.12024$ $Y_1=0.42009$ $Z_1=2.99851$
	$X_2=1.10976$ $Y_2=0.41616$ $Z_2=3.06399$	$X_2=1.10976$ $Y_2=0.41616$ $Z_2=3.06339$
3	$X_1=-0.36785$ $Y_1=1.21390$ $Z_1=2.64304$	$X_1=-0.36785$ $Y_1=1.21390$ $Z_1=2.64304$
	$X_2=-0.34832$ $Y_2=1.14945$ $Z_2=3.03365$	$X_2=-0.34832$ $Y_2=1.14945$ $Z_2=3.03365$
4	$X_1=0.00000$ $Y_1=-0.52649$ $Z_1=2.98007$	$X_1=0.00000$ $Y_1=-0.52649$ $Z_1=2.98007$
	$X_2=0.00000$ $Y_2=-0.60365$ $Z_2=1.95133$	$X_2=0.00000$ $Y_2=-0.60365$ $Z_2=1.95133$

를 264개의 삼각형으로 정의하고, 다른 하나는 같은 구를 1200개의 삼각형으로 정의한다. 엑스선 방사원은 Z축상의 100단위 지점(0, 0, 100)에 놓인다.

교차점 계산시간을 비교하기 위해, I-DEAS로 매개변수를 사용해서 위의 구면체를 정의하고 교차점을 계산한다. I-DEAS는 교차점들을 각기의 격자점에서 정의되는 각기의 엑스선에 대한 식들을 풀어서 구한다. 중요한 사실은 이 교차점들을 기하학적 일관성(geometric coherence)을 이용하지 않고 각기의 엑스선에 대해서 독립적으로 계산을 수행해서 구한다는 것이다. 그 다음, 같은 교차점들이 본 논문에서 개발된 알고리즘을 이용해서 계산된다. 모든 계산은 Apollo workstation을 이용해서 수행된다.

이 두방법을 이용한 계산시간은 Table 2에서 나타나 있다. I-DEAS를 이용했을 때 계산시간이 격자점 수가 증가됨에 따라 비례적으로 증가된다는 것을 Table 2의 두번째 열로부터 알 수 있다. Table 2의 세번째와 네번째 열은 본 논문에서 개발된 알고리즘을 이용했을 때의 계산시간을 보여준다. 계산시간은 격자점들의 수가 증가됨에 따라 상당히 완만히 증가된다. 또한 계산시간이 다각형 수가 증가됨에 따라 증가한다. 이것을 계산시간의 대부분이 좌표값들을 정리(sorting)하는데 주로 사용된다는 것을 말해준다. 이것은 만약 좀더 효율적인 좌표값을 정리하는 프로그램(sorting program)이 사용되었을 때 계산시간을 더욱 줄일 수 있다는 것을 의미한다. 교차점을 계산하기 위해 본 논문에서 개발된 알고리즘을 이용한 경우와 독립적인 계산을 수행하는 경우와 계산시간을 비교했을 때, 방사선 사진 시뮬레이션을 위해 많은 격자 수가 필요한 경우 본 논문에서 개발한 알고리즘을 이용했을 때 효율성에 있어서 아주 많은 증가를 있음을 알 수 있다.

Table 2. Comparison of compute time of intersection calculations

GRID	I-DEAS	Ray-Intersection (264 Poly)	Ray-Intersection (1200 Poly)
11×11	73 sec.	10 sec.	155 sec.
21×21	257 sec.	11 sec.	157 sec.
41×41	994 sec.	12 sec.	158 sec.
81×81	3864 sec.	17 sec.	160 sec.
201×201		48 sec.	171 sec.

## 6. 결 론

본 연구의 목적은 매우 복잡한 부품들의 시뮬레이팅된 방사선 사진을 대화식(interactive)으로 계산을 할 수 있게 하는 엑스선과 물체와의 교차점을 효율적으로 계산하기 위한 알고리즘을 개발하는데 있었다. 이 알고리즘은 맨 먼저 부품의 형상을 평평한 다각형들로 구성된 경계면(bounding surface)을 사용해서 근사하게 나타내었다. 이같이 근사하게 부품의 형상을 나타냈을 때, 만약에 충분한 수의 다각형들이 표면의 모델을 위해 사용된다면 시뮬레이팅된 방사선 사진의 정확도에 영향을 미치지 않는다는 것을 알 수 있었다. (앞절의 구면체에

서 264개의 다각형으로 모델된 예보다는 1200개의 다각형으로 모델된 예가 좀 더 정확한 교차점을 가진다.) 두 번째로 원근변환이 점 방사선 문제를 서로 평행한 엑스선을 가지는 문제로 변화시키기 위해 사용되었다. 세 번째로 엑스선의 격자의 일관성을 이용하는 주사선 알고리즘이 실행된다. 시뮬레이트된 방사선 사진을 계산하는데 이같은 방법은 매우 효율적이고 정확하다는 것을 알 수 있었고, 시뮬레이트된 방사선 사진을 대화식으로 계산할 수 있게하는 능력을 부여한다.

### 참고문헌

1. Kosanetzky, J., Harding, G., Fisher, K. H. and Meyer, A., "Compton Back-Scatter Tomography of Low Atomic Number Materials with the Surpass System", In Non-Destructive Testing, pp. 2118~2132. Edited by Farley, J. M. and Nichols, R. W., Proceedings of the 14th European Conference, Vol. 3, London, 1987.
2. Goldspiel, S., "Status of Reference Radiographs", In Nondestructive Testing Standards-A Review, pp. 115~128. Edited by Berger, H. ASTM Technical Publication 624, 1976.
3. Berger, H. "Some Recent Developments in X-Ray and Neutron Inspection Methods", In Non-Destructive Testing, pp. 155~162. Edited by Farley, J. M. and Nichols, R. W., Proceedings of the 14th European Conference, Vol. 3, London, 1987.
4. Sullivan, P. G. and Davis, L. W., "Nondestructive Testing Methods for Graphite/Aluminum Composites", In Nondestructive Evaluation And Flaw Criticality For Composite Materials, pp. 339~354. Edited by Pipes, R. B., American Society for Testing and Materials, Philadelphia, 1978.
5. Halmshaw, R., "Radiograph-Flaw Sensitivity Considerations", In Non-Destructive Testing, pp. 2017~2024. Edited by Farley, J. M. and Nichols, R. W., Proceedings of the 14th European Conference, Vol. 3, London, 1987.
6. Heidt, H., Stade, J. and Schnitger, D., "Standardization and Measurement of Radiographic Film Properties", In Non-Destructive Testing, pp. 2035~2041. Edited by Farley, J. M. and Nichols, R. W., Proceedings of the 14th European Conference, Vol. 3, London, 1987.
7. Crocker, G. A., "Invisibility Coherence for Faster Scan-Line Hidden Surface Algorithms", Computer Graphics, 18, No. 3, pp. 95~102, July 1984.
8. Heckbert, P. S. and Hanrahan, P., "Beam Tracing Polygonal Objects", Computer Graphics, 18, No. 3, pp. 119~127, July 1984.
9. Hamilton Jr., G., "Raster-Scan Hidden Surface Algorithm Techniques", Computer Graphics, pp. 206~213, 1979.
10. Lane, J. M., Carpenter, L. C., Whitted, T. and Blinn, J. F. "Scan Line Methods for Displaying Parametrically Defined Surfaces", Communications of the ACM, 23, No. 1, pp. 23~34, January 1980.
11. Porter, T. K. "The Shaded Display of Large Molecules", Computer Graphics, pp. 234~236, 1979.
12. Robertson, P. K., "Spatial Transformation for Rapid Scan-Line Surface Shadowing", IEEE Computer Graphics & Applications, pp. 30~38, 1980.
13. Atherton, P. R., "A Scan-Line Hidden Surface Removal Procedure for Constructive Solid Geometry", Computer Graphics, 17, No. 3, pp. 73~82, July 1983.
14. Rogers, D. F. and Adams, J. A., "Mathematical Elements for Computer Graphics", McGraw-Hill Book Company, New York, 1989.