

RP를 위한 절단 알고리즘의 개발

Development of Slicing Algorithm for Rapid Prototyping

채 희 창 (전북대학교 기계공학부)

유 상 훈 (전북대학교 기계공학부)

Abstract

Nowadays, as the development period of new products becomes shorter and consumer's requirement is more various, the importance of Rapid Prototyping Technology has been rapidly increased. Rapid Prototyping makes prototypes or functional parts directly using the 3D CAD data. But RP machines can make prototype in limit size. For making large size prototype, we slice solid, which is made of STL file, and then glue sliced solid. This paper is concerned with slicing solid on STL file.

1. 서 론

CAD로부터 데이터를 받아 실물을 만든다는 점과 정교한 수치제어가 필요하다는 점에서는 RP와 NC가공에는 차이점이 없다. 그러나 NC가공에서는 공구간섭에 의하여 가공하기 어려운 형상이 많은 반면에 RP는 cutter를 사용하지 않기 때문에 어떤 임의의 형상이라도 만들 수 있다. RP의 대부분의 방법은 기계 가공과 같은 기존의 방법과 달리 3차원 모델을 2차원으로 단면화 시켜 한 층씩 적층시켜 성형한다. 이를 위해 3차원 CAD 소프트웨어 상의 모델을 RP장비의 표준 입력 파일인 STL 파일로 변환해야 한다. 원래 STL 파일 포맷은 SLA 광조형 장치를 위해 개발한 파일 형식이었는데 현재 RP의 표준으로 사용되고 있다.

현재 국내외로 많이 사용되고 있는 RP 장비를 크게 분류하면 Laser를 광경화성 수지에 조사

하여 입체를 제작하는 광조형법, 고체 분말수지에 Laser를 조사하여 소결시켜 단면을 생성하는 SLS(Selective Laser Sintering), 열가소성 수지에 열을 가해 압출하여 형상을 제작하는 FDM(Fused Deposition Modeling), 종이나 수지등을 절단하여 가열, 압축시켜 만드는 LOM(Laminated Object Manufacturing), 액상의 왁스나 고분자를 직접 뿌려 입체형상을 제작하는 3DP(Three Dimensional Printing) 등이 있다.⁽¹⁾ 또한 패속 금형 제작(Rapid Tooling)용으로는 미국 DTM사와 독일 EOS사가 96년 각각 SLS 장비를 상용화 시켰으며 98년에 미국 Optomec사에서 LENS방법을 이용한 장비를 개발하여 상용화 시켰다. 미국 Extrude Hone 사는 MIT에서 개발한 3D Printing기술을 이용하여 PROMETAL장비를 상용화 시켰다.⁽²⁾

그러나 위에 열거된 장비들이 시작품을 제작할 수 있는 크기에는 제한이 있어 보다 큰 제품을 성형하기에는 불가능하다. 탁상용으로 나오는 제품의 경우에는 성형크기가 가로, 세로, 높이가 각각 30cm를 넘지 않으며 보다 큰 크기의 제품을 성형할 수 있는 장비라도 가로, 세로, 높이가 각각 70cm이상의 제품을 성형할 수는 없다. 큰 크기의 제품을 성형하기 위해서는 부분을 나누어 따로 성형한 후에 접착제로 붙여야 한다. 따라서 본 논문은 큰 크기의 입체를 성형이 가능하도록 하기 위해 입체의 절단 알고리즘을 개발하고자 하는데 있다.

2. 면과 삼각형의 기하학

(1) 임의점 P(X,Y,Z)와 무한평면과의 위치판별
임의점의 무한평면에 대한 위치판별은 삼각

형과 무한평면과의 관계를 판별하는데 있어서 기본이 되는 과정이다. Kripac⁽³⁾은 임의점의 좌표를 평면의 방정식에 대입하여 얻은 평면과 임의 점 사이의 수직거리를 이용하여 임의점과 평면사이의 수직거리의 부호를 판별함으로써 임의점의 평면에 대한 위치를 판별하였다. 무한평면을 기준으로 3차원 유클리드 공간을 2 개의 반공간(half-space)으로 분리할 수 있으며, 평면으로부터 법선벡터 방향에 있는 반공간을 상부공간, 반대방향에 있는 반공간을 하부 공간이라 한다. 점 P의 위치를 평면위(on), 상부공간(upper), 하부공간(lower)로 분류하면 평면의 방정식에 임의점 P(X,Y,Z)를 대입하여 얻는 값에 의해 P의 평면에 대한 위치를 판별할 수 있으며, 이를 다음과 같이 반공간 파라미터로 정의하였다.

$$K = N_x X + N_y Y + N_z Z + D$$

이때 K의 절대값은 평면과의 수직거리를 의미하고 그 부호는 하부 반공간의 내부, 외부, 평면위에 있는지 여부를 나타낸다.

(2) 선분과 평면과의 관계

선분의 양단의 좌표를 A(X_a,Y_a,Z_a), B(X_b,Y_b,Z_b)라 하고 이 두점의 반공간 파라미터를 각각 K_a, K_b라 하면,

$$K_a = N_x X_a + N_y Y_a + N_z Z_a + D$$

$$K_b = N_x X_b + N_y Y_b + N_z Z_b + D$$

- ㉠ K_aK_b > 0 (교차하지 않음)
- ㉡ K_a = 0 이고 K_b > 0, K_b < 0 (A점에서 교차)
- ㉢ K_b = 0 이고 K_a > 0, K_a < 0 (B점에서 교차)
- ㉣ K_a = 0, K_b = 0 (평면위에 선분이 존재)
- ㉤ K_aK_b < 0 (1점(X_p,Y_p,Z_p)에서 교차)

$$X_p = (K_a X_b - K_b X_a) / (K_a - K_b)$$

$$Y_p = (K_a Y_b - K_b Y_a) / (K_a - K_b)$$

$$Z_p = (K_a Z_b - K_b Z_a) / (K_a - K_b)$$

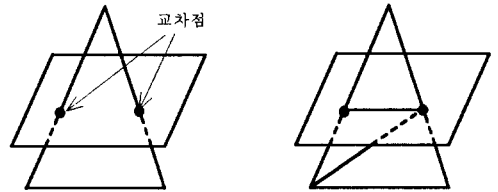
(3) 삼각형과 평면과의 관계

선분과 평면과의 관계를 이용하여 삼각형과 평면과의 관계를 판별할 수 있다. 왜냐하면 삼각형은 3개의 선분으로 이루어져 있기 때문이다. 삼각형으로 구성된 Solid를 임의의 평면으로 자르기 위해서는 먼저 모든 삼각형과 평면과의 관계

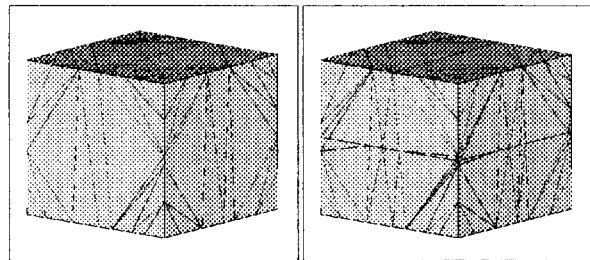
를 판단하여야 한다. 고려해야 하는 경우의 수는 평면과 삼각형이 교차하지 않는 경우(2가지), 평면과 삼각형의 한 꼭지점이 교차하는 경우(6가지), 평면과 삼각형의 한 변이 교차하는 경우(6가지), 평면 위에 삼각형이 놓여있는 경우(1가지), 삼각형의 두 변이 평면과 교차하는 경우(12가지), 로 총 27개의 경우의 수가 있다. 이때 삼각형을 구성한 데이터 구조에서 임의의 변수 지시자를 만들어 삼각형과 평면의 위치관계를 저장한다. 이렇게 함으로써 계산할 때마다 위치관계를 판별할 필요 없이 단 한번만 위치판별을 하면 된다.

(4) 삼각형의 재분할

위에서 고려한 삼각형의 경우에서 삼각형의 두 변이 평면과 교차하는 경우에는 평면을 중심으로 삼각형이 나뉘어져야 하기 때문에 삼각형을 재분할 해야한다. 이것은 삼각형의 두 변과 평면과의 교차점을 구함으로써 간단히 해결할 수 있다. 아래 그림 1과 그림 2는 이것을 보이고 있다.



<그림 1> 절단평면에 대한 삼각형의 재분할

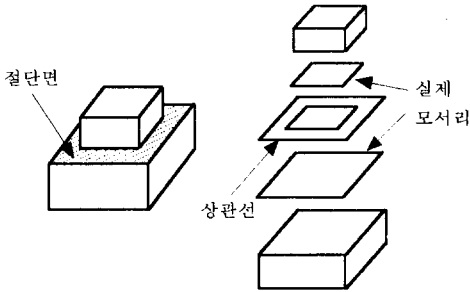


<그림 2> Y=40인 평면에 대해 입체를 절단하기 위한 삼각형의 재분할

3. 입체의 절단과정에 관한 고찰

입체를 절단하는 일은 절단평면이 무한평면이기 때문에 절단평면에 의한 면의 절단은 쉬운

일이나, 상관선으로 부터 두 개의 단면을 만드는 일은 단순하지 않다. 이는 1차원적인 상관선으로부터 2차원적인 면을 복원하는 일이기 때문이다. 또한 절단평면에 생기는 상관선은 단면의 모서리가 되는 후보가 될 수는 있지만, 단면의 모서리 자체는 아니다. 앞에서 고찰한 삼각형과 평면과의 관계에서 삼각형의 한 변이 평면 위에 있는 경우에는 실제 모서리가 되는지의 여부를 정확히 판단해야 한다.



<그림 3> 단면의 모서리에 대한 고찰

위의 그림3에서 상관선과 단면의 실제 모서리와 차이를 알 수 있다. 차원의 감소에 의해 부족하게 된 정보를 입체의 기하학적 정보로부터 얻어내어 이를 보충해야 한다. 이 경우는 절단할 입체의 면과 모서리가 절단평면에 존재할 경우이며 위치판별법을 사용할 필요가 있다.

(1) 3차원 모서리 근방해석

입체의 한 모서리에서 상관하는 삼각형의 입체에 대한 위치판별은 입체의 형성조건 중의 인접조건으로부터 쉽게 얻을 수 있다. 이는 3차원 모서리 근방해석으로 볼 수 있으며 다면인접 입체의 경우 여러 개의 섹터로 이루어진다.⁽⁴⁾ 입체의 삼각형 중에서 공통 모서리에 인접하는 모든 삼각형을 찾아 공통모서리를 중심으로 회전각을 계산한 다음 이를 올림차순으로 저장한다. 최초로 찾은 삼각형을 t_1 이라고 하고 공통모서리와 인접하는 모서리의 방향벡터를 e_1 , 삼각형의 법선벡터를 N_1 이라면, 이 삼각형의 접선벡터 T_1 은 다음과 같다.

$$T_1 = N_1 \times e_1$$

회전각 순서로 정리된 회전각을 계산하고자하는 i 번째 삼각형을 t_i 라고 하고 공통 모서리와 인접하는

삼각형 모서리의 방향 단위벡터를 e_i , 법선벡터를 N_i 라 하면 t_i 의 접선벡터 T_i 는 다음과 같다.

$$e_i = (-1)^{i+1} e_1, \quad T_i = N_i + e_i$$

t_i 의 회전각 θ_i 는 T_1 과 T_i 사이의 각이며, 다음과 같이 계산한다.

$$\cos(\theta_i) = T_1 \cdot T_i$$

$$\sin(\theta_i) = (T_1 \times T_i) \cdot e_1$$

(2) 위치판별

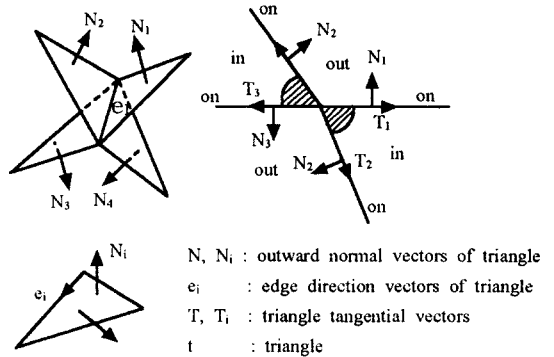
삼각형 리스트에 있는 삼각형들과 위치판별할 삼각형의 공통 모서리를 중심으로 한 각도와 법선벡터를 비교하여 다음과 같은 4가지 위치를 결정한다. on/on 모호성을 처리하기 위해 on의 경우를 on^+ , on^- 로 분리하였다.

out : 입체 외부에 있는 삼각형

in : 입체 내부에 있는 삼각형

on^+ : 입체에 있는 삼각형과 겹치며 면의 방향이 동일한 삼각형

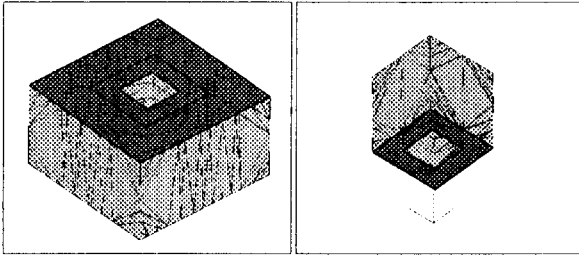
on^- : 입체에 있는 삼각형과 겹치며 면의 방향이 정반대인 삼각형



<그림 4> 3차원 모서리 근방해석

단면의 모서리 후보가 되는 절단평면에 생기는 상관선들 중에 in 과 on^+ 가 되면 단면의 모서리가 된다. in 의 경우에도 단면의 모서리가 되도록 하면 그림 3에서 절단된 아래쪽의 입체에서 단면의 모서리가 실제 모서리가 아닌 상관선으로 구해지는 단점이 있으나 이는 입체를 절단하는 과정에서 전혀 문제되지 않는다. 아래 그림 5는 실제 프로그램 상에서 3차원 모서리근방해석을 고려하여 잘려진 입체의 모서리를 구한 그림이다. 그림

에서 굽은선이 입체의 모서리이다.



<그림 5> 3차원 모서리 근방해석을 통한 절단된 입체의 모서리 구하기

위의 그림 5에서 보여지는 것처럼 절단평면에 의해 삼각형으로 구성된 입체를 절단한 후에 채워지지 않은 면을 생성해 주어야 한다. 이를 위해 다음 장에서 Delaunay 삼각형분할법을 고려한다.

4. Delaunay Triangulation

유한 요소 해석에 대해, 큰 너비를 가진 삼각형(fat triangle)이 가장 좋다. 즉, 작은 각도를 가진 삼각형을 생산시키지 않는 것이다. 따라서, 분할 시에 가장 큰 최소각(the largest smallest angle)을 가진 분할을 해야 한다. 다시 말해서, 생성될 수 있는 모든 삼각형 분할의 경우에 대해 최소각을 최대화하는 것이다. Delaunay 삼각형 분할은 이러한 개념을 만족하는 삼각형 분할이다.

입체의 점 집합 S의 삼각형 분할을 T로 가정하고, 각이 작은 순으로 배열된 T의 각 배열(angle sequence)을 $(\alpha_1, \alpha_2, \dots, \alpha_n)$ 로 놓는다. α_1 은 최소 각, α_n 은 최대각이며, t 는 T에서 삼각형의 개수이다. t 는 각 S에 대해 상수이다. T의 각 배열의 크기가 T'의 각 배열이 큰 경우($T \geq T'$) 즉, $\alpha_1 > \alpha'_1$ 이거나 $\alpha_1 = \alpha'_1$, 그리고, $\alpha_2 > \alpha'_2$ 이거나 $\alpha_3 > \alpha'_3 \dots$ 인 경우에 해를 얻게 된다. 이 알고리즘은 삼각형 분할을 일정한 수만큼 계속하여 최소각이 최대인 분할을 찾게 된다. 이에 Delaunay 분할 $T = D(P)$ 는 위의 개념을 만족하게 된다. Delaunay 분할법은 일반적으로 평면상에서 볼록한 외부 경계를 가진 점군 데이터에 대해 지역을 분할하는 알고리즘이다. 이 분할법의 알고리즘을 구현하는 것은 그리 복잡하지는 않으나, 내부에 구멍(hole)을 포함하고 있거나, 오

목한 경계(concave boundary)가 있는 경우에는 이들을 고려하여 구현해야 하므로 알고리즘이 훨씬 복잡해지게 된다.

(1) 정의에 의한 알고리즘

다음은 Delaunay Triangulation을 위한 알고리즘이다.

step 1. 시작점을 선택한다.

step 2. 시작점을 선택한 후, 가장 가까운 점(nearest point)을 찾고 두 점에 의한 모서리를 첫 번째 모서리로 결정한다.

step 3. 삼각형의 방향은 반시계 방향으로 결정하기 때문에, 첫 번째 모서리에 대해 왼쪽에 있는 점을 찾아 stack에 저장한다.

step 4. 저장된 순서에 따라 점을 선택하면서 세 점을 이용해 삼각형을 구성한다. 삼각형이 결정되면 삼각형의 외심을 계산한다.

step 5. 외심과 다른 점과의 거리(distance)를 계산한다.

step 6. 외접원의 반지름(radius)을 계산한다.

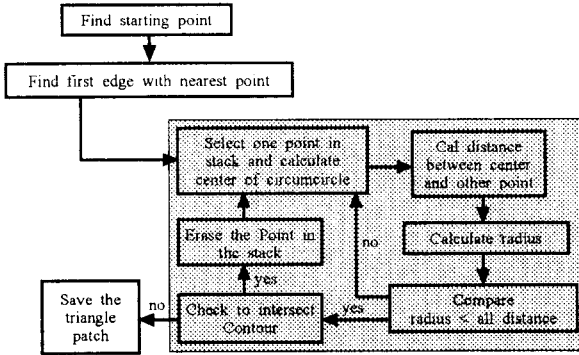
step 7. 외접원의 반지름이 모든 distance보다 작은 경우 삼각형으로 결정하고, 그렇지 않은 경우 step 4로 돌아간다.

위의 step 7에서 얻어지는 삼각형이 Delaunay 삼각형이다. 그러나 이 방법으로는 경계가 있는 점들을 가지고 삼각형분할을 할 수 없다. 왜냐하면 Delaunay Triangulation 방법으로 삼각형을 생성하는 것은 삼각형의 외접원만 가지고 판단하기 때문에 경계 자체는 고려되지 않는다. 때문에 step 7에서 얻어지는 삼각형이 완성된 삼각형이라 할 수 없고 이 삼각형이 경계와 상관하는지의 여부를 판단해야 한다. 만약 경계와 상관하게 된다면 step 3에서 stack에 저장된 점들 중에서 step 7에서 삼각형으로 결정된 점을 제거시키고 다시 step 4로 되돌아간다. 그림 6은 위의 알고리즘을 순서도로 표시한 그림이며 그림 7은 Contour를 고려한 경우와 고려하지 않은 경우를 나타낸 그림이다.

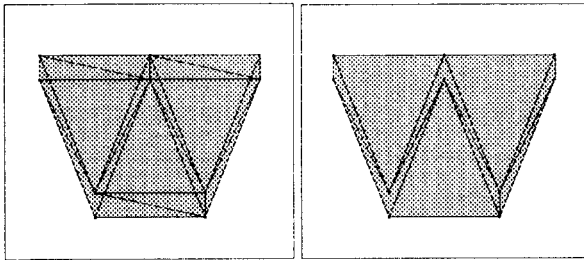
(2) 불필요한 삼각형의 제거

내부에 구멍을 포함하고 있거나 오목한 경계가 있는 경우에는 앞에서 설명한 방법으로 삼각

형을 먼저 생성한 다음에 불필요한 삼각형을 제거시킨다. 불필요한 삼각형의 판단여부는 탐색선 알고리즘을 사용하였다.⁽⁵⁾ 이 방법은 먼저 삼각형



<그림 6> Delaunay triangulation algorithm



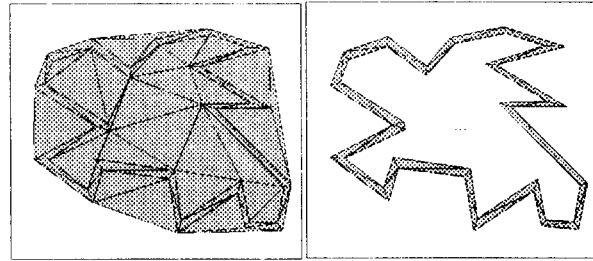
<그림 7> 경계를 고려하지 않은 경우와 고려한 경우의 Delaunay Triangulation

의 무게 중심을 구한 후 이점을 가지고 경계의 내, 외부 판단하는 방법인데 x 방향으로 음의 무한대에 이르는 탐색선을 생성하고 이 탐색선과 경계와의 교점의 수를 세어 삼각형이 내부에 포함되어 있는지를 판단하는 것이다. 교점의 개수가 홀수라면 경계의 내부에 있는 것으로 판단되며, 교점의 개수가 짝수라면 경계의 외부에 있는 것으로 판단되어 이 삼각형을 제거 시키면 된다. 그림 8은 탐색선 알고리즘을 이용하여 불필요한 삼각형을 제거시킨 후의 그림이다.

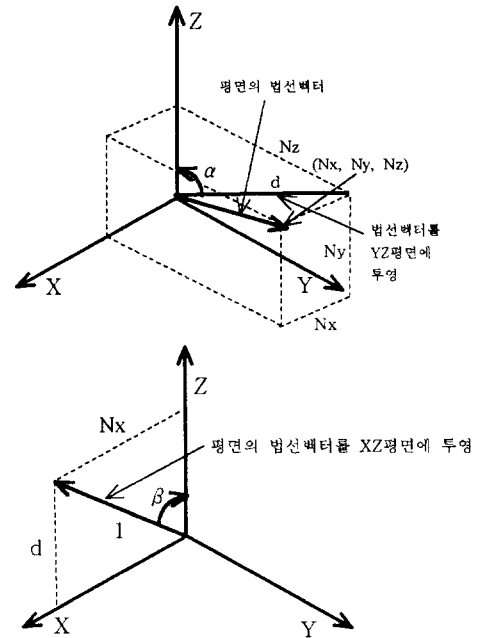
(3) 임의의 평면을 XY평면과 일치시키기

입체를 임의의 평면으로 자르기 위해서는 앞에서 구한 상관선과 교차점들을 XY 평면으로 회전과 이동을 이용하여 좌표값을 변환시킨다. 그 후 Delaunay Triangulation 방법으로 삼각형을 생성하고 이 삼각형들을 다시 원상태로 복구시킨다. 임의의 평면을 XY평면과 일치시키는 것은

이 평면의 Normal Vector의 방향을 Z축과 일치시키는 것과 동일하다. Normal Vector의 방향을 Z축과 일치시키기 위해서는 X축으로 회전을 한 후에 Y축으로의 회전이 필요하다. 먼저 평면의 법선벡터를 YZ평면에 투영시켜 이 직선과 Z축과의 각을 아래 그림 9처럼 α 라하면 이 각은 X축을 중심으로 회전한 각과 같다. 또 평면의 법선벡터를



<그림 8> 탐색알고리즘을 이용한 불필요한 삼각형의 제거



<그림 9> 임의의 평면을 XY평면과 일치시키기

XZ평면에 투영시켜 이 직선과 Z축과의 각을 β 라하면 이 각은 Y축을 중심으로 회전한 각과 같다. 여기서 α 와 β 를 구하면

$$d = \sqrt{N_y^2 + N_z^2}$$

$$\cos \alpha = \frac{N_z}{d}, \quad \sin \alpha = \frac{N_y}{d}$$

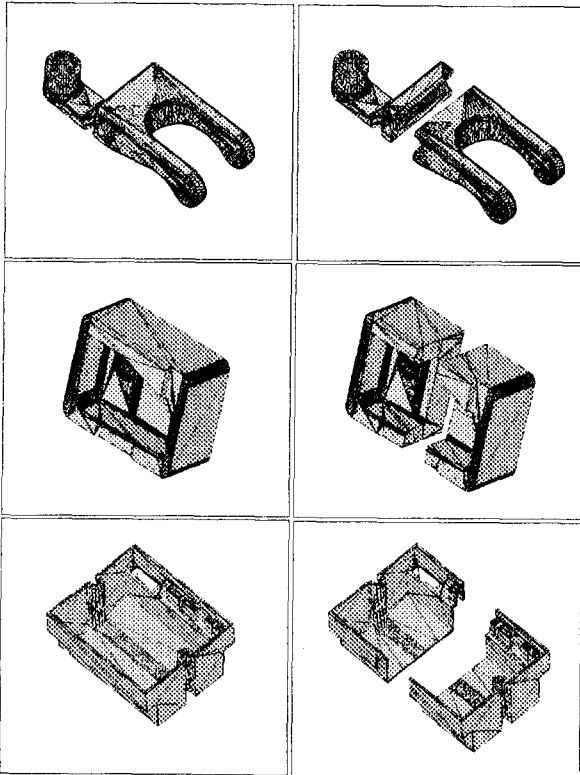
$$\cos \beta = \frac{N_x}{d}, \quad \sin \beta = \frac{N_z}{d}$$

이다. 임의의 평면을 XY평면과 일치시키는 변환 행렬은

$$[M] = [T] [R_x] [R_y]$$

이다.

위에서 구한 변환행렬을 이용하여 상관선들을 XY평면으로 이동 후 삼각형분할을 한 후에 다시 역으로 분할된 삼각형들을 이동시키면 입체의 절단이 완성된다. 아래 그림은 실제 프로그램에서 입체가 절단된 모습을 보여주고 있다.



<그림 10> 입체의 절단

5. 결론

지금까지 큰 크기의 입체를 RP 기술로 제작하기 위해 STL파일로 이루어진 입체를 절단하는 알고리즘을 설명했다. 맨 처음 절단평면에 대해 삼각형의 관계를 판단한 후 상관하는 삼각형들을 재분할하고 잘려진 입체의 경계를 구하기 위해 3

차원 모서리 근방해석을 고려한다. 상관선들을 XY평면으로 좌표변환을 하고 잘려진 부분을 삼각형으로 채우기 위해 경계를 고려한 Delaunay Triangulation 방법을 썼다. 마지막으로 잘려진 단면이 작은 경우에 양 단면을 계단화하여 접촉면적을 넓히고 성형시간을 단축하기 위해 잘려진 입체의 부분을 따로 성형하지 않고 RP 기계가 허용하는 범위라면 동시에 성형하게 함으로써 제작시간을 단축시키는 방법에 대한 연구가 남아있다.

참고문헌

- (1) J.P. Kruth, "Material Increase Manufacturing by Rapid Prototyping Techniques", Annals of the CIRP Vol. 40, pp. 603~614, 1991
- (2) 송용익 외 6인, "용접과 절삭가공을 이용한 쾌속 금속 조형기술 개발", 쾌속시작기술 연구회 98년도 춘계학술대회 논문집, pp. 24~25
- (3) Kripac, J. : "Algorithm for Splitting Planar Faces", Computer-Aided Design, Vol .19, No .6, pp. 293-298 (1987).
- (4) Requicha, A.A.G, Voelcker, H.B : "Boolean Operation in Solid Modeling : Boundary Evaluation and Merging Algorithms", Proceedings of the IEEE, Vol. 73, No. 1, pp. 30-44 (1985)
- (5) Minyang Yang and Eungki Lee : "NC verification for wire-EDM using an R-map", Computer-Aided Design Vol.28, No. 9 pp.733-740(1996)