

# STL 포맷의 구멍 오류 수정을 위한 삼각형 분할법 적용에 관한 연구

## A Study on Application of Triangulation Method for Hole Error Verification of STL Format

손영지\*, 조연상, 이승수 (동아대 대학원), 전언찬 (동아대)

Y. J. Son\*, Y. S. Cho, S. S. Lee (Graduate School, Dong-A Univ.),

E. C. Jeon (Dong-A Univ.)

Key Words : STL (Sterolithography), RP (Rapid Prototyping:급속조형), Triangulation Method (삼각형 분할법), Hole Error (구멍 오류)

### ABSTRACT

This paper verified error of STL file and presented a new application of reformed division method for reducing to appeared error by using Delaunay triangulation method that used to modify error of verified data. First, it is analyzed each vertices in hole error and classified follow case that is 1. case of plan or slope, 2. case of edge, 3. case of apex and 4. case of rapid curve, and reduced volume tolerance between original model and converted model after convert STL file.

### 1. 서론

시제품 생산에 있어서 제품을 생산하기 전에 문제점을 미리 확인해 보려는 시도가 대두되면서 최근 RP(Rapid Prototyping : 급속조형) System에 대한 연구가 활발하게 진행되고 있다. 특히 형상을 한 층씩 Slicing을 하여 레이저 빔을 발사한 뒤 광경화성 수지를 옹고 시키고 단계로 쌓아 조형을 하는 방식이 가장 많이 사용되고 있으며 인터넷 망을 통한 VR(Virtual Reality : 가상현실)에서 입체적으로 시뮬레이션해 보려는 시도도 연구되고 있다.

이 RP공정에 필요한 입력 데이터로서 STL (STeroLithography) 포맷을 사용하고 있는데 최초로 3D System사에서 RP 시스템으로 광조형법 성형 시스템을 개발하여 이 포맷을 사용한 것으로서 된 3차원 형상 정보 포맷이다. 이 STL 파일을 만들기 위해 CAD에서 3차원 형상의 모든 면을 Triangular facets로 변환해 주어야 하는데 그 변환 과정에서 정점과 면의 오류를 내포하는 등 단점이 없는 것은 아니나 이렇다 할 후속 표준 포맷이 나오지 않아서 계속 쓰이고 있으며 점차 보완하고 있는 실정이다.

법선 벡터와 삼각형 facets의 정보를 가진

STL 파일은 ASCII로 되어 있어서 직접 값을 확인 할 수 있으며 STL 파일은 CAD 시스템에서 모델링만 되어지면 쉽게 변환할 수 있다.

현재 시판되고 있는 CAD 시스템에는 대부분 STL 방식을 지원하고 있으며 기존 포맷 방식 즉 IGES, STEP, PDES 등의 포맷을 STL로 변환하는 변환기가 개발되어 사용되고 있다.

Block Rooney가 개발한 STL변환기가 있고 C-TAD 시스템사에서는 PDGS와 IGES 파일을 STL로 변환하는 변환기, Autodesk사의 AutoCAD, Pro/Engineer와 I-DEAS 시스템에서도 각각 STL변환기가 사용되고 있다.

이런 STL 파일을 이용하여 오류를 검증하고 가시화 시키는 국외의 연구로서 Morvan<sup>1)</sup>등은 가상 환경에 있어서 STL파일을 이용하여 구현하는 방법을 제안 했으며 Tanaka<sup>2)</sup>등은 facets의 오류를 구멍 오류와 모서리 오류로 분류하여 2차원 평면에 투영한 다음 빠져있는 구멍을 분할하고 재구성하는 것을 제안하였다. 또 M.J. Wozny<sup>3)</sup>등은 기존의 STL 포맷의 오류를 지적하고 중복된 정점을 줄이기 위하여 STL 파일 위에 정점, 모서리, 면의 색인 리스트를 가진 새로운 RPI를 제안을 하였다. A. Dolenc<sup>4)</sup>등은 IGES를 VDAFS방식으로 변환한 다음 facetted

Representation 방식의 STL 포맷으로의 변환 절차를 제시 하였다.

국내에서 활발히 연구중인 결과를 살펴보면 각각의 정점과 삼각형의 존재 조건의 관계에서 오류를 검증하여 viewer를 개발하였고<sup>5)</sup> 삼각형 기반 비다양체 형상을 중점적으로 다루어 오류를 수정하였다.<sup>6)</sup> 이와 같은 연구활동을 통하여 오류가 점차 극복되어지고 있다.

CAD 시스템에서 모델링한 3차원 곡면 형상을 RP System에 사용하기 위하여 STL 파일로 변환을 하면서 삼각형의 facets로 분할한다. 이 변환과정에서 곡면과 곡면이 만나는 부분에서 치명적인 구멍 오류가 생기기도 한다. 오류에도 종류가 많고 그 오류를 수정하는데 있어서 많은 연구들이 활발하였기에 별도의 설명은 하지 않겠다.

본 논문에서는 STL 파일의 오류중 구멍 오류의 수정에 대하여 사용되어지는 Delaunay의 삼각형 분할법을 적용하는 데 있어서 신뢰성을 향상 시키기 위하여 급속 곡면을 고려하여 분할법을 고찰하였다.

## 2. 기존의 삼각형 분할 적용

Delaunay의 삼각형 분할법은 평면상에 불규칙한 다각형을 삼각형으로 분할하는데 있어서 임의의 세점을 선택하여 내접원의 반지름을 구하고 다음 임의의 세점을 동일한 방식으로 구하여 가장 내접원의 반지름이 작은 세점부터 면을 형성하여 만들어가는 방식이다.

RP System의 STL 포맷에서 오류가 가장 많이 발생하는 것은 급속 조형의 맞물림 부분인데 기존 Delaunay의 삼각형 분할법을 사용할 경우 Fig. 1과 같이 정점과 정점을 직접 연결함으로써 전체 체적상의 오차를 발생시킬 수도 있는 것이다.

Fig. 2의 (a)와 같이 평면이나 완만한 곡면의 구멍오류와 달리 (b), (c), (d)와 같은 부분은 각각 형태의 종류에 따라 아래와 같은 방법

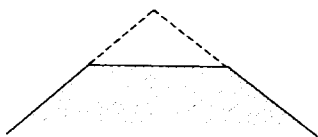


Fig. 1 Volume tolerance

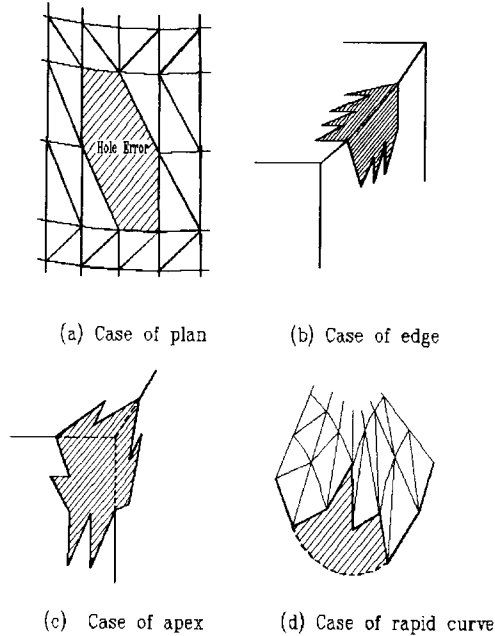


Fig. 2 Type of hole error

으로 Delaunay의 삼각형 분할법 적용을 고찰해 보기로 한다.

## 3. 삼각형 분할법 적용의 새로운 방법

먼저 발생되어진 구멍 오류의 경계를 구하여 좌표값을 기억한다.

다음은 구멍의 오류가 어떤 형상부분인지 판단한다. 즉, 직선으로 찍인 모서리 부분인지 체적의 꼭지점 부분인지 완만한 곡형을 이루는 부분인지를 우선 판단한다.

후에 구멍의 오류가 발생한 장소가 평면에 유사한 경우와 완만한 곡면을 이루는 경우(Fig. 2의 (a))는 기존의 Delaunay의 삼각형 분할법을 적용한다. 완만한 곡면의 경우란 본 논문에서는 법선 벡터의 각도차가  $30^\circ$  이내일 경우를 적용했다. 다음에 직선으로 찍인 모서리 부분인 경우(Fig. 2의 (b))는 주변의 꼭지점을 참조하여 구멍 부분에 임시의 직선을 그어서 면을 분할 한 다음 각각의 면에 Delaunay의 삼각형 분할법을 적용하여 삼각형을 만든다. 꼭지점 부분의 오류인 경우(Fig. 2의 (c))는 정점의 연관성을 고려하여 꼭지점을 찾아낸다. 역시 꼭지점으로부터 가상의 선을 그은 다음 분할되어진 면에서 Delaunay의 삼각형 분할법을 적용하여 분할을 시작한다. Fig. 2에서의 (d)처럼 급격한 곡면의 형상으로 면의 오류가 생긴 경우는

Delaunay의 삼각형 분할법을 적용하는 것이 아니라 곡면에 대한 유추해석을 한다.

STL 포맷의 data를 분석하면 알 수 있듯이 각각의 좌표들은 일정한 방향성을 가지고 있으며 급격한 곡면 형상의 오류는 facets가 여러개 임을 알 수 있다.

이런 급속 곡면의 구멍오류에 면을 만들어 내기 위해서 그 오류의 경계를 이루는 좌표들 중에 경계의 중심부로 향하고 마주보는 임의의 두 정점을 구하고 그 정점에서의 단위 접선 벡터를 구한다.(Fig. 3)

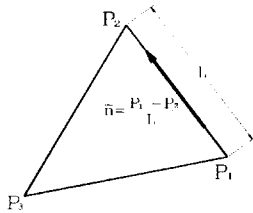


Fig. 3 Unit vector

구해진 이 두 정점과 두 벡터를 가지고 가상의 3차 스플라인 곡선을 만들어 낸다.

Fig. 4은 3차 스플라인 곡선을 도식화 하였다.

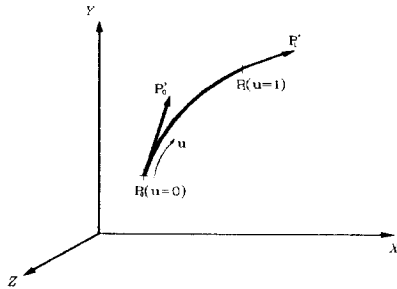


Fig. 4 Spline curve of 3D

그리고 여기에 관련된 식은 다음과 같다.<sup>7)</sup>

$$P(u) = (2u^3 - 3u^2 + 1)P_0 + (-2u^3 + 3u^2)P_1 + (u^3 - 2u^2 + u)P'_0 + (u^3 - u^2)P'_1, \quad 0 \leq u \leq 1 \quad (1)$$

$$P'(u) = (6u^2 - 6u)P_0 + (-6u^2 + 6u)P_1 + (3u^2 - 4u + 1)P'_0 + (3u^2 - 2u)P'_1, \quad 0 \leq u \leq 1 \quad (2)$$

여기서,  $u$ 는 매개변수이며,  $P_0, P_1$ 은 곡선의 끝 정점이고,  $P'_0$ 과  $P'_1$ 은 그 정점에서의 접선 벡터이다.

(1)식과 (2)식을 행렬 형식으로 간략화 하면 (3)식과 (4)식이 된다.

$$P(u) = U^T[M_H]V, \quad 0 \leq u \leq 1 \quad (3)$$

$$P'(u) = U^T[M_H]uV, \quad (4)$$

여기서,  $U = [u^3 \ u^2 \ u \ 1]^T$ 이며,  $[M_H]$ 는 Hermite 행렬이고,  $V$ 는 형상 벡터이다.

$u$ 의 함수를 이용하여 Fig. 2의 (d)처럼 가상의 곡선을 만들어 주변 삼각형 정점의 길이를 고려하여 정점을 찍는다.

프로그램을 위한 흐름도는 Fig. 5와 같다.

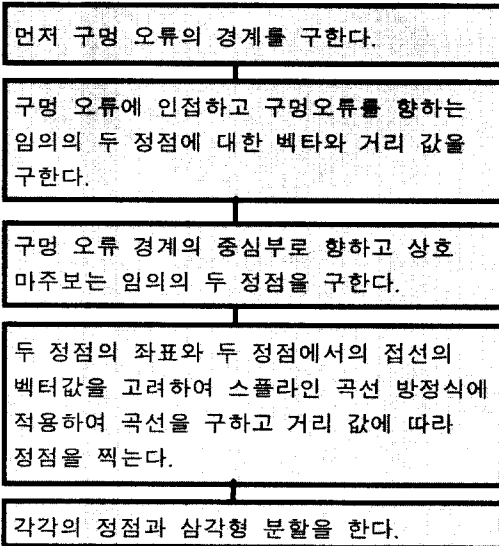
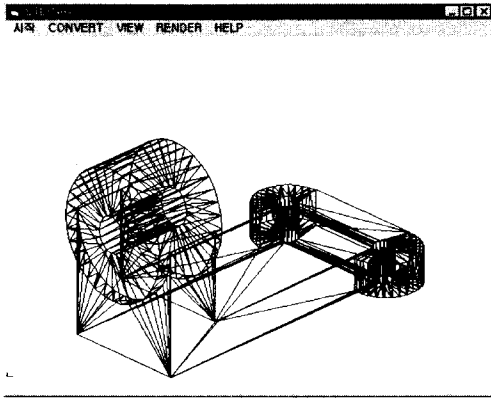


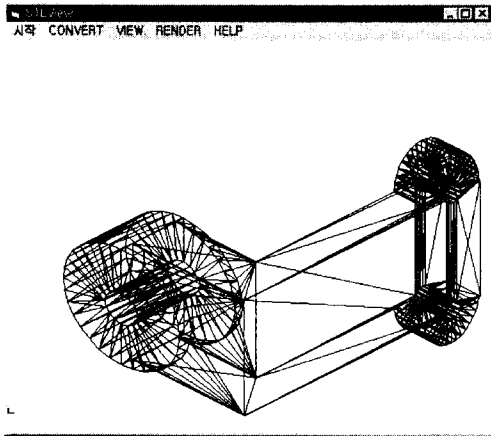
Fig. 5 Flowchart for program

본 연구과정에서 테스트한 것처럼 오류의 형상에 따라 구분을 하여 분할 방식을 달리하였을 경우에 있어서 체적 오차가 상대적으로 줄어들었음을 보였다.

먼저 급격한 형상으로 인해 구멍의 오류가 많이 발생할 만한 형상을 모델링하였다.



(a) Curve surface of braket



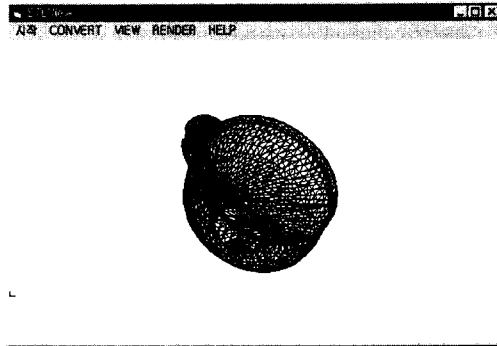
(b) 90° rotation

Fig. 6 Model of rotational braket

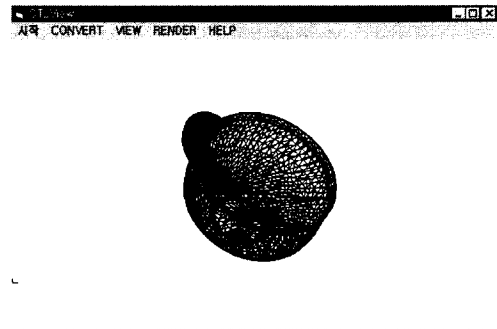
Table 1 Comparisons of volume for non application and application

|         | Original volume | Conventional method | New method |
|---------|-----------------|---------------------|------------|
| Model 1 | 27326.332       | 27319.548           | 27323.311  |
| Model 2 | 3923.575        | 3923.090            | 3923.382   |
| Model 3 | 71265.496       | 71264.856           | 71265.192  |
| Model 4 | 5626.149        | 5622.619            | 5624.934   |
| Model 5 | 34873.634       | 34873.110           | 34873.489  |

같은 모델링에서 삼각형 분할 방식을 적용하였을 경우와 적용하지 않았을 경우를 구분하여



(a) Cup model including hole error



(b) Cup model after conversion

Fig. 7 Comparison for model including hole error and after conversion

Table 1과 같이 체적의 오차로 비교하였다.

#### 4. 개발 도구 및 프로그램의 특징

지금까지 개발되어졌던 프로그램이 대부분 UNIX 환경하의 것이어서 PC에서의 개발이 필요하게 되었다. 본 논문에서는 windows 95 환경에서 Visual Basic 5.0과 Visual C++ 4.0으로 만들었고 WinApi를 이용하였다.

모델링은 AutoCAD R13의 AME를 가지고 실시하였고 STL로의 변환은 AutoCAD자체 변환기를 사용하였다.

본 연구로 만들어진 프로그램은 애니메이션 기법을 추가하여 낮은 시스템에서도 효과를 올리기 위하여 30° 방향마다 화면을 display하였고 상하좌우 전체 28개의 장면을 연출할 수 있도록 하였다.

#### 5. 결론

본 연구에서 다음과 같은 결과를 얻었다.

(1) 구멍오류의 형태를 분류하여 적용단계를 다르게 하였다.

- (2) 급속 곡면에 발생하는 오류에는 그 곡선을 유추하여 형상을 유지하도록 하였다.
- (3) 오류수정시 발생할 수 있는 체적오차를 줄였다.
- (4) 수정된 STL 데이터를 중심으로 가시화할 수 있는 STL-viewer를 개발하였다.

**참고문헌**

- 1) Morvan S.M., Fadel G.M., "Virtual Prototyping Using STL Files", International Body Engineering Conference/IBEC , Cobo Exposition Center, October 1-3, 1996, Detroit, Mi.
- 2) 田中文基, 岸浪建史, "光造形法における問題点とその解決法", 第6回光造形システムシンポジウム, pp. 39~45, 1994
- 3) M.J. Wozny, "Data Driven Solid Freeform Fabrication", IFIP Transactions B-3: Human Aspects in Computer Integrated Manufacturing, pp. 71~82, 1992
- 4) A. Dolenc, I.Makela, R.Hovtun, "Better Software for Rapid Prototyping with INSTANTCAM", IFIP Transactions B-3: Human Aspects in Computer Integrated Manufacturing, pp. 449~456, 1992
- 5) 최홍태, 이석희, "급속조형시스템을 위한 STL 포맷의 오류 검증에 관한 연구", 한국정밀공학회 96년도 춘계학술대회논문집, pp. 597~601, 1996
- 6) 황동기, 채희창, "삼각형기반 형상모델러를 이용한 STL의 검증", 한국정밀공학회 96년도 추계학술대회논문집, pp. 578~582, 1996
- 7) Ibrahim Zaid, "Original CAD/CAM Theory and Practice", McGraw-Hill Book Co., -Singapore, pp. 229~234, 1995