

## STL 포맷의 오류수정을 위한 DB 구축과 형상복구에 의한 효율성 연구

손영지\*, 박정보\*\*, 김순경\*\*\*, 김중완\*\*\*\*, 전언찬\*\*\*\*\*

### A Study on DB Construction for Error Modification of STL Format and Efficiency by Shape Restoration

Young-Ji Son\*, Jeong-Bo Park\*\*, Soon-Kyung Kim\*\*\*, Joong-Wan Kim\*\*\*\*, Eon-Chan Jeon\*\*\*\*\*

#### ABSTRACT

This paper recommends the database construction for STL data and the new method of Delaunay's triangulation method in order to decrease volume errors by verifying STL file errors and modifying hole errors. Each vertex for the boundary coordinates of the verified hole errors was analyzed according to the following cases: planes or slow curves, edges, apexes, and rapid curves. In each case, a different modification process was adopted.

With this method, most errors resulting from the conversion into the STL data as well as lots of volume errors arising from error modification were decreased.

**Key Words :** RP(급속조형), STL(Stereolithography), Triangulation Method(삼각형 분할법), CAD(전산용설계),  
Hole error(구멍오류)

#### 1. 서 론

시제품 생산에 있어서 제품을 생산하기 전에 문제점을 미리 확인해 보려는 시도가 대두되면서 최근에 급속조형(RP : rapid prototyping) 시스템에 대한 연구가 활발하게 진행되고 있다. 특히 형상을 한 층씩 슬라이싱(slicing)을 하여 레이저 범을 발사한 뒤 광경화성 수지를 응고시키고 단계로 쌓아 조형을 하는 방식이 가장 많이 사용되고 있으며 인터넷 망을 통한 가상현실(VR : virtual reality)에서 입체적으로 시뮬레이션 해보려는 것도 연구

되고 있다.<sup>(1-2)</sup>

STL(stereolithography) 포맷은 RP공정에 필요한 입력 데이터로 사용하고 있는데 최초로 3D System 사에서 RP 시스템을 위한 광조형법 성형 시스템을 개발하여 이 포맷을 사용한 3차원 형상 정보 포맷이다. 이 STL 파일을 만들기 위해서는 CAD에서 3차원 형상을 모델링한 후 그 모델링의 모든 면을 삼각형 면으로 변환해 주어야 하는데 그 변환 과정에서 정점과 면의 오류를 내포하는 등의 단점이 있다. 이런 단점을 보완하기 위한 각처의 연구가 속속 발표되고 있다.

\* 동아대학교 기계공학과 대학원  
\*\* 삼척산업대학교 기계공학과  
\*\*\* 동의공업전문대학 자동차과  
\*\*\*\* 동아대학교 기계공학과

변환된 STL 파일을 이용하여 오류를 검증하고 가시화 시키는 국외의 연구로서 Morvan<sup>(3)</sup>등은 가상환경에 있어서 STL파일을 이용하여 구현하는 방법을 제안 했으며 Tanaka<sup>(4)</sup>등은 facets의 오류를 구멍오류와 모서리 오류로 분류하여 2차원 평면에 투영한 다음 빠져있는 구멍을 분할하고 재구성하는 것을 제안하였다. 또 M.J. Wozny<sup>(5)</sup>등은 기존의 STL 포맷의 오류를 지적하고 중복된 정점을 줄이기 위하여 STL 파일 위에 정점, 모서리, 면의 색인 리스트를 가진 새로운 RPI를 제안하였다. 그리고 A. Dolenc<sup>(6)</sup>등은 IGES를 VDAFS방식으로 변환한 다음 faceted representation 방식의 STL 포맷으로의 변환 절차를 제시하였다.

국내에서도 여기에 관하여 연구가 진행중에 있으며 지금까지의 연구 결과를 살펴보면 각각의 정점과 삼각형의 존재 조건의 관계에서 오류를 검증하여 viewer를 개발하였고<sup>(7)</sup>. 삼각형 기반 비다양체 형상을 중첩적으로 다루어 오류를 수정하였다.<sup>(8)</sup> 또 구멍오류의 수정에 대한 삼각형 분할을 상태별로 다르게 적용하여 형상 및 체적의 오차를 줄인 연구도 있었다.<sup>(9-12)</sup> 이와 같은 연구를 통하여 오류가 점차 줄어들고 있다.

본 논문에서는 STL 파일의 오류를 검증하기 위해 DB를 구축하고 이를 이용하여 오류검증과 형상복구를 하게 되는데 오류가 발생하는 것을 종류별, 형태별로 각각 구분하여 오류를 수정하였다. 이들을 본래의 체적과 비교하여 각 모델에 대한 오류복구의 효율성을 검토하므로써 데이터에 대한 신뢰성을 향상시키고자 하였다.

## 2. STL data를 위한 시스템 구성

### 2.1 시스템 구성을 위한 개발환경

본 논문의 모델들은 AutoCAD에서 모델링하여 stl 파일로 변환해 준 것이다. 이 변환된 stl 파일을 구현할 수 있도록 프로그램을 개발하였는데 windows 95환경에서 visual C++ 4.0과 visual Basic 5.0을 사용하였다.

### 2.2 STL 포맷의 구조 및 오류

이 STL 파일은 CAD 시스템에서 모델링을 하여 데이터를 변환하게 된다. 삼각형 면(facet)에 대한 법선벡터와 세 개의 정점(vertex) 정보로 이루어진 STL 파일은 ASCII로 되어 있어서 Fig. 1에서와 같이 직접 눈으로 값을 확인할 수 있다. vertex라는 tag를 가지고 있는 첫줄의 좌표를 P1, 다음 줄을 P2, 그 다음 줄을 P3로 선정하

였다. 이는 정점이 P1->P2->P3로 순서대로 흐르고 있으며 이 흐름에 맞추어 법선벡터(facet normal)의 방향은 오른손 법칙이 적용되고 있다.

이 STL 파일의 데이터는 모델형상의 접합부분과 급속 곡면을 이루는 부분에서 오류가 발생된다고 보고되고 있으며<sup>(3-12)</sup> 연구된 바에 의한 오류를 대별하여 살펴보면 facet이 빠져버리는 구멍오류, 하나의 모서리(edge)를 두 개의 facet이 공유를 하는 중복오류, 세 개의 정점이 삼각형을 이루지 못하는 정점오류 등이 있으며 완전한 facets는 모여 있지만 체적을 이루지 못하는 비다양체오류로 나눌 수 있다.

```
solid AutoCAD
facet normal 0.000000e+000 0.000000e+000 1.000000e+000
outer loop
vertex 1.1859983e+001 4.3492240e-000 3.8241639e+000
vertex 1.1859983e+001 8.6209809e+000 3.8241639e+000
vertex 5.6365373e-000 8.6209809e+000 3.8241639e+000
endloop
endfacet
.....
endsolid AutoCAD
```

Fig. 1 STL format

### 2.3 오류수정을 위한 DB 구축 및 검증

한 개의 facet 정보를 살펴보면 x, y, z의 각 축에 대한 법선벡터 그리고 세 개의 정점이 있다. 이들은 정보의 연속으로 이루어져 있어서 검증을 할 경우은 각각의 좌표들간의 위치와 방향의 비교로 이루어 진다.

이들 정보의 오류수정을 위한 DB 구축은 Fig. 2와 같다. 전체의 database는 각각이 하나의 facet정보로 이루어 졌다. 이 facet은 각 정점의 위치와 이웃하는 facet과의 관계를 비교하기 위해 Edge정보를 구축한다. 카운터(C)는 하나의 Edge가 이웃하는 facet을 가지고 있을 때 숫자를 증가시키기 위해 만들었다.

Fig. 2의 정보 구축에서 STL파일이 정상적인 데이터가 되기 위한 조건은 다음과 같다.

첫째, 한 개의 facet은 법선 벡터와 세 개의 정점이 존재해야 한다. 둘째, 세 개의 정점은 일직선이 되거나 두 정점 이상의 위치가 중복되지 말아야 한다. 셋째, 삼각형의 세 모서리는 각 하나씩 이웃하는 facet이 한 개씩만 존재해야 한다. 넷째, 이웃하는 facet의 edge는 현재 facet의 edge와 정점의 방향이 반대이며 위치는 같아야 한다. 다섯째, 법선벡터와 세 정점의 방향은 오른손 법칙에 따라야 한다.

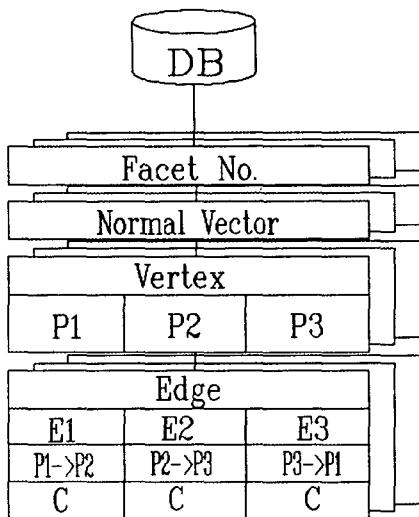


Fig. 2 Database for STL information

## 2.4 오류발생에 따른 데이터의 신뢰성

이들의 조건이 만족하지 않는 한 정상적인 데이터라 할 수 없다. 오류가 가장 많이 발생되는 부분은 조형이 맞물리는 부분과 급속 곡면인 경우인데 Delaunay의 삼각형 분할법을 사용할 경우 Fig. 3과 같이 정점과 정점을 직접 연결함으로써 전체 체적상의 오차(volume error)를 발생시킬 수도 있는 것이다.

따라서 데이터에 대한 신뢰성은 stl 파일로 변환하기 이전의 AutoCAD 모델을 계산한 체적값을 기준으로 하여 stl 파일로 변환한 후에 오류 수정을 한 다음 체적값의 비교에 의한 오차의 정도로 판단할 수 있다.

체적에 대한 계산에 있어서 기준이 되는 AutoCAD 모델의 체적은 자체 계산 명령으로 산출한 체적값을 근거로 하였다. 변환 및 오류 수정후의 체적 계산은 모델을 Z축을 단위 길이로 충분활(slicing)하여 각각의 교점과 마주치는 경계의 영역을 면적으로 계산한 후 각 층의 면적의 합을 계산하였다.<sup>(12)</sup> 여기서 단위길이는 가능한 정확한 계산값을 위하여 0.01로 하였다.

## 3. 오류의 수정

### 3.1 종류별 오류의 수정

각각의 정점들이 구축하고 있는 DB를 검색해 각 오류의 종류별로 다음과 같이 수정을 한다.

(1) 정점오류가 되는 위의 첫째와 둘째 조건에 위배되

는 경우는 리스트에서 삭제한다.

(2) 중복오류가 되는 세번째 조건에 위배되는 경우는 한 모서리를 중복사용한 경우이다.

세 모서리가 각각 하나씩 이웃하는 Edge를 가지지 못할 경우는 리스트에서 삭제한다.

(3) 방향오류가 되는 넷째의 조건에 위배되는 경우는 이웃하는 법선벡터와의 방향이 바뀐 경우이므로 리스트에서 삭제한다. 다섯째 조건에 위배되는 경우는 세 정점의 순서(P1->P2->P3)를 (P2->P1->P3)로 바꾸어 다시 중복오류를 검사한다.

(4) 각 오류로 인해 facet이 삭제되면 구멍오류가 남게 된다. 구멍오류를 복구할 경우 Delaunay의 삼각형 분할법을 사용하게 되는데 이 때에는 오류부분에 있는 형상을 고려해야 한다.

Fig. 4는 구멍오류가 일어날 수 있는 여러 가지 종류를 나타낸 것이다. 여기서 (a)는 평면이나 완만한 곡면의 구멍오류인 경우에 반해 (b), (c) 및 (d)는 (a)와 달리 오류의 수정이 복잡하게 된다. 따라서 이러한 것에 Delaunay의 삼각형 분할법의 적용을 아래와 같이 형태별로 나누어 수정하여야 한다.

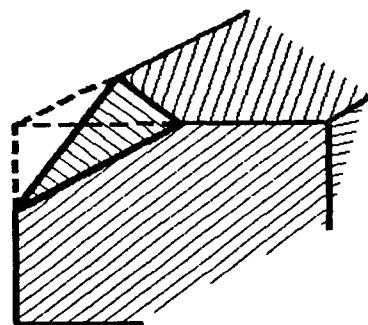


Fig. 3 Volume error

### 3.2 형태별 오류의 수정

Fig. 4의 (b)와 (c)처럼 모서리와 꼭지점 부분의 오류인 경우는 경계정점의 연관성을 고려하여 임의의 모서리와 꼭지점을 찾아낸다. 꼭지점으로부터 가상의 선을 각 모서리로 그은 다음 가상의 선에 의해 분할되어진 면에서 Delaunay의 삼각형 분할법을 적용하여 분할을 시작한다. Fig. 4의 (d)처럼 급격한 곡면의 형상으로 구분이 없는 면의 오류가 생긴 경우는 Delaunay의 삼각형 분할법을 직접 적용하는 것이 아니라 곡면에 대한 유추해석을 한다.

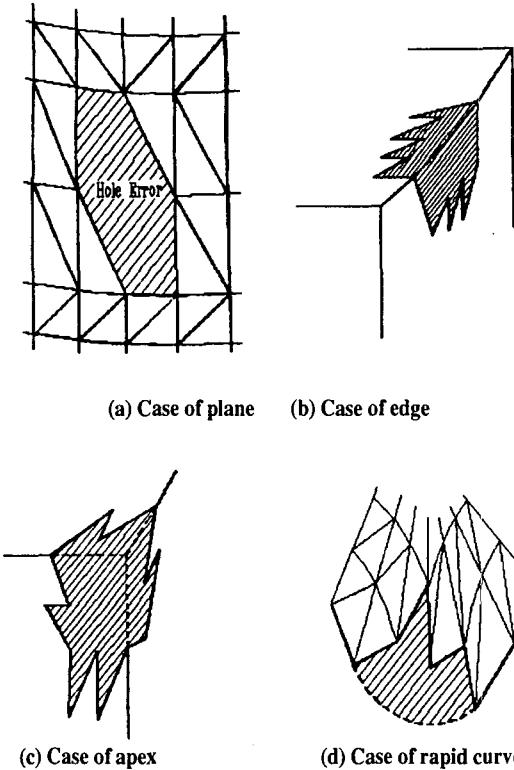


Fig. 4 Type of hole error

Fig. 5에서와 같은 삼각형에서 각각의 정점을  $P_1, P_2$  및  $P_3$ 라 하고  $P_1$ 과  $P_2$ 사이의 거리를  $L$ 이라 하면 단위접선벡터  $\bar{n}$ 는 식 (1)과 같다.

$$\bar{n} = \frac{(P_1 - P_2)}{L} \text{ or } \frac{(P_2 - P_3)}{L} \text{ or } \frac{(P_3 - P_1)}{L} \quad (1)$$

여기서

$P_1, P_2, P_3$  : 삼각형의 각 정점

$L$  : 두 정점간의 거리

$\bar{n}$  : 단위접선벡터

이다.

두 정점과 정점에서의 단위 접선벡터를 가지고 가상의 3차 스플라인 곡선을 만들어 냈다.

Fig. 6은 3차원 공간상에 스플라인 곡선을 나타낸 것이다. 임의의 두 정점을  $P_1$ 과  $P_2$ 라 하고 각 정점에서의 접선벡터를  $P'_1$ 과  $P'_2$ 라 하면 매개변수  $u$ 가 1에서 2사이의 변화되는 모습을 나타내고 있다.

그리고 여기에 관련된 식은 다음과 같다.<sup>(13)</sup>

$$P(u) = (2u^3 - 3u^2 + 1)P_1 + (-2u^3 + 3u^2)P_2 \quad (2)$$

$$+ (u^3 - 2u^2 + u)P'_1 + (u^3 - u^2)P'_2, \quad 1 \leq u \leq 2$$

$$P'(u) = (6u^2 - 6u)P_1 + (-6u^2 + 6u)P_2 \quad (3)$$

$$+ (3u^2 - 4u + 1)P'_1 + (3u^2 - 2u)P'_2, \quad 1 \leq u \leq 2$$

여기서,

$u$  : 매개변수

$P_1, P_2$  : 곡선의 끝 정점

$P'_1, P'_2$  : 정점에서의 접선벡터

이다.

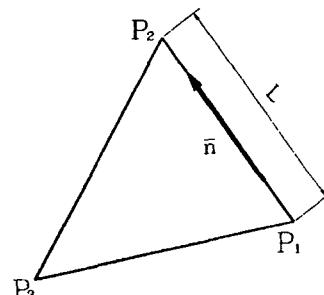


Fig. 5 Unit vector

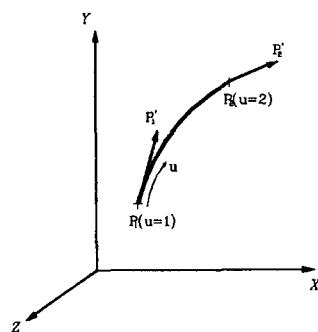


Fig. 6 Spline curve of 3D

급속곡면의 오류수정인 경우의 절차는 Fig. 7과 같다.

#### 4. 오류수정에 의한 결과 및 고찰

아래 그림들은 오류가 있는 경우의 모델들을 나타내고 있으며 이 오류들을 종류별, 형태별로 수정하여 본래의 체적과 수정후의 체적을 비교하였다.

Fig. 8에서 Fig. 12까지의 모습을 보는 바와 같이 이들 모델을 가시화시킬 수 있도록 각 정점을 선분으로 연결하여

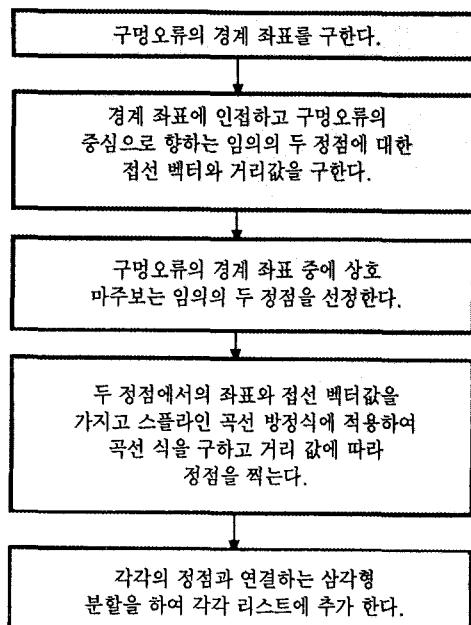


Fig. 7 Flowchart for modification of rapid curve

나타내는 프로그램을 만들었다. STL 파일을 불러올 수 있는 시작 메뉴와 오류가 발생하였을 경우 수정할 수 있는 CONVERT 메뉴가 있다. VIEW메뉴는 모델을 회전하여 배면의 모습을 볼 수 있도록 하였고 모델의 확대 축소를 할 수 있도록 하였다. 또 렌더와 도움말메뉴를 넣었다.

Fig. 8은 브라켓(bracket)의 형상을 모델링하여 나타낸 그림인데 형상이 간단해서인지 오류가 나타나지 않았다. 그래서 연속적인 임의의 10개 facets을 삭제하여 오류를 만든 다음 수정을 적용시켜 보았다(Model 1). Fig. 9는 컵형상을 모델링하였는데 손잡이 부분에서 구멍오류가 발생한 경우이다. 이는 급속 곡면이 이루는 부분으로 오류의 가능성성이 있는 부분으로 판단된다. 연속적인 곡면으로 모델링되어졌기 때문에 정점의 개수가 많아 data의 용량이 크다(Model 2). Fig. 10은 병의 입구 부분과 바닥부분인 두군데에 구멍오류가 발생하였음을 볼 수 있는데 역시 급속한 곡면을 이루고 있는 부분이다(Model 3).

Fig. 11은 플랜지(flange) 형상을 모델링하였고 명확한 경계선부위에 오류가 빠져있음을 볼 수 있다. 이런 경우는 오류를 수정하였을 때 오류의 수정율이 높음을 예측할 수 있다 (Model 4). Fig. 12는 플라스틱 커버 형상을 모델링하였다. 여기서 곡면을 이루고 있는 모서리부분에 구멍오류가 발생하여 오류수정율이 높을 것으로 예측한

다. 또 이 커버형상은 작은 크기로 모델링되었기에 오차의 양도 작게 나을 것으로 예측한다(Model 5).

위의 구멍오류의 경계영역에서 정점과 정점사이의 거리값  $L$ 을 크게 잡을 경우는 spline 곡선 위에 생성되는 정점의 개수가 증가하게 되므로 data의 용량이 증가하는 것은 물론 속도가 저하될 수 있으나 본래의 곡면에 가장 적절히 부합할 수 있다. 그러나 본 논문에서 거리값으로 사용되는 것은 경계좌표 주변에 있는 정점과 정점사이의 거리값을 참조로 적용을 하였다.

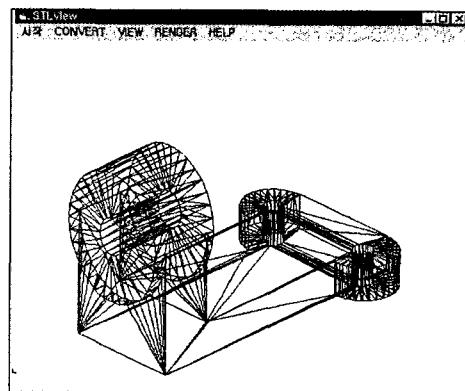


Fig. 8 Bracket model including hole error(Model 1)

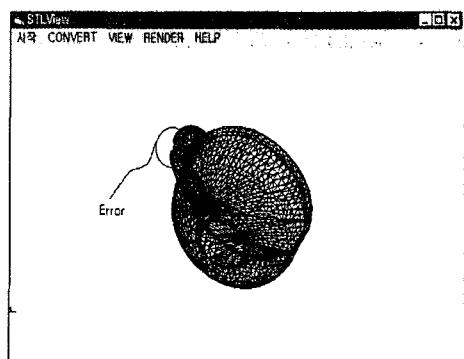


Fig. 9 Cup model including hole error(Model 2)

위에서 열거한 다섯가지의 모델을 본래의 체적에 대하여 오류를 수정한 후 체적을 비교하여 Fig. 13에 나타내었다. 여기서 첫 번째 막대는 모델을 할 당시 최초(Original)의 체적을 나타내었고 두 번째 막대인 체적이 가장 작은 Conventional method는 오류가 나타나는 부분을 형태별로 수정하지 않은 경우이다. 또 세 번째 막대인

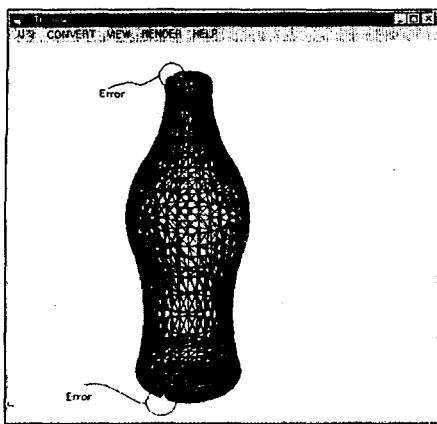


Fig. 10 Bottle model including hole error(Model 3)

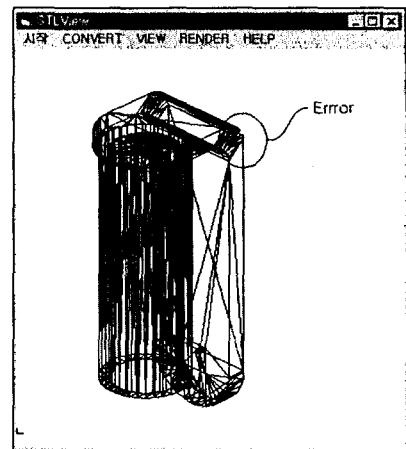


Fig. 12 Plastic cover model including hole error(Model 5)

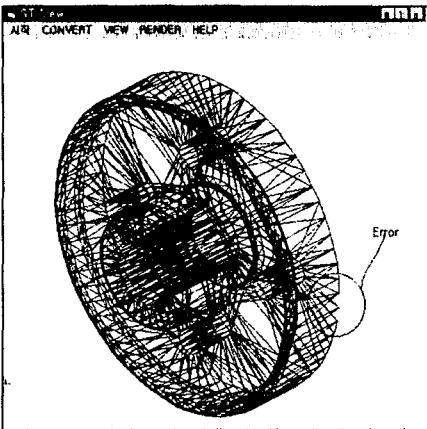


Fig. 11 Flange model including hole error(Model 4)

New method는 각각의 오류를 분류하여 형태별로 수정을 한 경우이다. 각 모델이 본래의 체적에 비해 근소한 차이로 오차를 가지고 있어서 체적의 오차가 거의 없지만 오차의 정도는 다섯가지 모델 모두공통적으로 형태별 오류수정(New method)이 적다는 것을 알 수 있다.

Fig. 14는 각 모델에 대하여 본래 체적과 오류를 수정하고 난 다음의 체적의 차이를 나타내고 있다. 각 모델의 첫 번째 막대인 Conventional method는 형태별 오류수정을 하지 않아 오차의 양이 많다. 두 번째 막대인 New method는 형태별 준류에 의한 오류수정의 경우이며 오차의 양이 상대적으로 적다는 것을 볼 수 있다. 각 모델의 ratio(%)는 Conventional method의 오차를 기준으로 New method가 오차를 얼마나 줄였는가의 효율을 나타낸다.

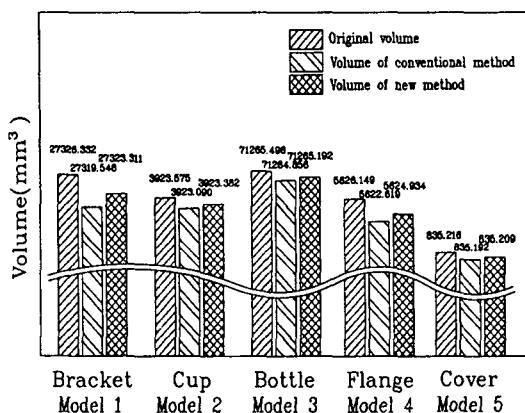


Fig. 13 Comparison of volume for conventional method and new method

내고 있다. 이중 네 번째와 다섯 번째 모델은 예측했던 것처럼 명확한 모서리의 경계부위에 오류가 수정이 되었기에 다른 모델들보다 오차감소율이 높음을 볼 수 있다.

전체적으로 형태별 분류에 의한 수정방식의 오차감소율이 기존의 형태별분류를 하지 않은 경우보다 60.92%만큼 향상되었음을 볼 수 있다. 이보다 더 좋은 효율을 위해서는 금속곡면의 오류수정에서 spline 곡면에 적용하는 정점의 개수를 증가시켜야한다.

## 5. 결 론

본 연구에서는 STL 데이터의 정보 관리를 위해 database를 구축하였다. 이로써 오류의 검증을 체계적으로

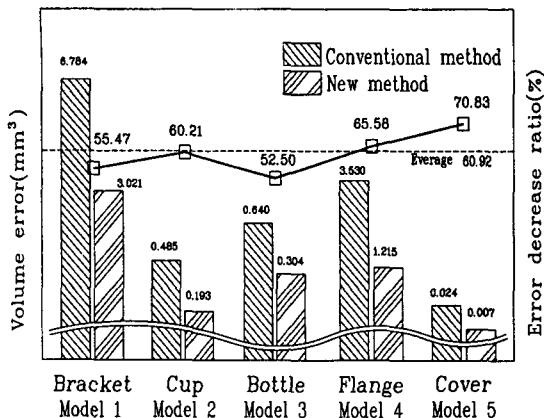


Fig. 14 Comparison of volume error for conventional method and new method

할 수 있게 하였고 형태별 오류의 분류를 할 수 있게 하였다. 종류별, 형태별 오류의 수정을 통해 다음과 같은 결과를 얻었다.

- 1) 구멍오류의 영역을 삼각형 facets로 분할할 때 직접적으로 삼각형을 분할하기 보다는 구멍오류의 형태를 분류하여 평면, 모서리, 꼭지점 및 급격한 곡면의 경우로 나누어서 각각 다르게 적용하는 것이 오차의 양을 더 많이 감소시킨다는 것을 알았다.
- 2) 급속 곡면에 발생하는 오류에는 정점과 접선의 벡터를 이용하여 곡선을 유추해서 형상을 유지하도록 하였으나 모서리나 꼭지점 부근의 오류보다 오차감소율이 적음을 알았다.
- 3) 본래 모델의 곡면에 완전히 부합하기 위해서는 구멍오류의 영역에 있는 정점과 정점사이의 거리를 적게하여 더 많은 개수로 정점을 삽입해야 한다.

## 후 기

본 연구는 한국 과학 재단 지정 동아대학교 지능형 통합 항만관리 연구센터의 지원에 의한 것입니다.

## 참 고 문 현

1. 홍삼열, 김준안, 반갑수, 백인환, “광조형용 CAD/CAM 시스템 개발,” 한국정밀공학회 96년도 추계학술대회논문집, pp. 592~596, 1996.
2. 정영태, 최홍태, 이석희, “급속조형법을 위한 실시간

형상 모델링과 VRML 응용에 관한 연구,” 한국정밀공학회 97년도 추계학술대회논문집, pp. 321~326, 1997.

3. S. M. Morvan, G. M. Fadel, “Virtual Prototyping Using STL Files,” International Body Engineering Conference/IBEC, Cobo Exposition Center, October 1-3, Detroit, Mi, 1996.
4. 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.
5. M. J. Wozny, “Data Driven Solid Freeform Fabrication,” IFIP Transactions B-3: Human Aspects in Computer Integrated Manufacturing, pp. 71~82, 1992.
6. A. Dolenc, I.M. kel, R. Hovtun, “Better Software for Rapid Prototyping with INSTANT-CAM,” IFIP Transactions B-3: Human Aspects in Computer Integrated Manufacturing, pp. 449~456, 1992.
7. 최홍태, 이석희, “급속조형시스템을 위한 STL 포맷의 오류 검증에 관한 연구,” 한국정밀공학회 96년도 추계학술대회논문집, pp. 597~601, 1996.
8. 황동기, 채희창, “삼각형기반 형상모델러를 이용한 STL의 검증,” 한국정밀공학회 96년도 추계학술대회논문집, pp. 578~582, 1996.
9. 손영지, 김민주, 이승수, 전언찬, “RP 시스템의 STL 포맷 오류 검증을 위한 삼각형 분할법의 고찰,” 대한기계학회 부산지부 97년도 추계학술대회논문집, pp. 134~138, 1997.
10. 손영지, 조연상, 이승수, 전언찬, “STL 포맷의 구멍 오류 수정을 위한 삼각형 분할법 적용에 관한 연구,” 한국정밀공학회 97년도 추계학술대회논문집(II), pp. 889~893, 1997.
11. 손영지, “STL 포맷의 구멍오류 수정을 위한 삼각형 분할법에 관한 연구,” 석사학위 논문, 동아대학교 대학원, pp. 8~29, 1997.
12. 김준안, “급속시작을 위한 광조형시스템 개발 및 성능향상에 관한 연구,” 박사학위논문, 부산대학교 대학원, pp. 73~77, 1997.
13. Ibrahim Zaid, “Original CAD/CAM Theory and Practice,” McGraw-Hill Book Co.,-Singapore, pp. 229~234, 479~516, 1995.