韓國産業應用學會誌, 제5권, 제2호, 141-146, 2002.5
J. Korean Society of Industrial Application, Vol.5, No.2, 141-146, May, 2002

STL 포맷의 단면정보를 이용한 형상분할에 관한 연구
A Study on Feature Division using Sliced Information of STL Format

반갑수*
Gab-Su Ban*

<Abstract>

Stereolithography is the best known as rapid prototyping system. It uses the STL format data which is generated from CAD system. In this study, One of the main function of this developed CAM system deals with shape modification which divide a shape into two parts or more. The cross section of a STL part by a z-level is composed with nested or single polygonal closed loop. In order to make RP product, closed loops must fill with triangular facets from SSET and recover sliced triangular facets which is located normal direction to the cross sectional plane. The system is development by using Visuall C++ compiler in the environment of pentium PC. Operating system is Windows NT workstaion from Micro-Soft.

Key words : STereoLithography(STL), StereoLithography Appatatus(SLA), Rapid Prototyping(RP), Triangular facet, Selection set(SSET)

1. 서 론

급속조형을 이용한 시제품제작 기술(Rapid Prototyping Technology : 이하 RP)은 다양화 되는 고객의 욕구에 보다 빨리 대처하고 금형의 단납기를 위해서 연구의 필요성이 있다. CAD시스템 상에서 Modeling 된 3차원 형상을 표면정보를 이용하여 허용오차 내에서 삼각형면[1]으로 정보변환을 하고 이를 이용하여 원하는 모형을 제작하는데 그 목적이 있는 것이다. 한 예를 소개하면 여러 가지의 RP방식 중에서 SLA방식은 형상모델링의 내.외부 표면의 정보를 triangular facet으로 파일을 생성하고 이를 다시 단면으로 slicing하여 각 slice별로 조형정보를 생성한다. slice별로 해야할 작업은 단면형상의 채워지는 부분에만 Laser beam이 주사되도록 NC궤적을 생성하고 이 slice들을 단면과 수직 방향으로 적층하면 원하는 형상이 생성된다. 이 경우 제작하고자 하는 형상이 H/W의 가용 작업범위 보다 커서 한번에 가공하기 불가

* 정회원, 상주대학교 기계공학부 조교수, 工博
E-mail : gsban@sangju.ac.kr.
경북 상주시 가장동 386

* Assistant Prof., School of Mech. Eng.,
Sangju National University
E-mail : gsban@sangju.ac.kr

능하거나 특정부분에 대해서만 신속히 형상을 제작해 보기 위해서 형상분할을 할 필요가 있으며, 여러 형상을 동시에 나열하여 가공할 필요가 있다. 경우에 따라서는 가공위치를 이동할 필요도 있으며 회전이나 Mirror를 할 필요도 있다.

여기서 STL파일에 의한 기하학적 형상이 제조를 위한 CAM시스템의 작업공간을 초과하면 STL파일 생성 이전의 CAD모델에서 적절한 크기로 다시 생성할 수 있겠지만 여기에 대한 정보가 없을 때는 STL모델에서 형상분리 공정을 거쳐야 한다.

형상을 분리하는 것은 STL 오류수정에 해당되는 문제로 볼 수 있다. 형상수정에 대한 연구는 F. Tanaka[2] 등은 STL파일에서 얻어진 점군 데이터의 삼각형 패치 사이의 facet데이터가 빠져 있는 경우와 하나의 모서리에 2개 이상의 facet이 중복되게 겹쳐있는 경우의 점군데이터를 Delaunay삼각형 분할을 이용하여 삼각패치 재구성법을 제안하였고, P. Vuyyuru[3] 등은 STL파일은 3차원 삼각형 facet으로 근사하기 때문에 실제 CAD모델과의 오차가 생기므로 facet수를 많게 하면 이런 오차를 줄일 수 있지만 파일크기가 커져서 이를 해결하는 방안으로 NURBS (Non -Uniform Rational B-Spline)곡선을 SLA가 수용할 수 있는 2차원 직선 세그먼트로 분할하는 방법을 제시하였다. 채희창[4]은 입력된 STL파일에 대하여 다양한 위상정보를 제공하는데 데이터베이스에 입력하는 과정에서 위상정보를 부여하여 솔리드 또는 삼각형 패치에 대한 위상정보를 화면상에서 확인 할 수 있게 하고 수정된 영역에 대해서는 Delaunay분할법을 적용하여 STL포맷의 질을 향상시켰다. 손영지[5] 등은 구멍오류의 영역을 삼각형 facets으로 분할할 때 구멍오류의 형태를 분류하여 평면, 모서리, 꼭지점 및 급격한 곡면의 경우로 나누어 오차의 양을 줄이는 연구를 하였고, 최홍태[6] 등은 DXF파일을 이용하여 STL파일을 생성하는 연구를 수행하였다. 기존의 연구에서는 슬라이싱된 단면정보가 단일 loop인 경우에 STL누락 및 중첩 등의 오류수정에 대한 연구 위주로 이루어졌다.

본 논문에서는 단면정보가 중첩된 loop의 형상복구와 주로 형상의 측면에 위치하는 슬라이싱 된 삼각형 facet를 동시에 자동으로 복구하는 알고리즘 연구와 이를 독자적으로 개발한 CAM시스템에 적용해 보았다.

2. STL format의 기하학적 구조

2-1. 측벽에서의 구조

절단면의 z-level에 인접하는 trianglar facet은 끝이 없이 연결된 형태로 존재해 있다.

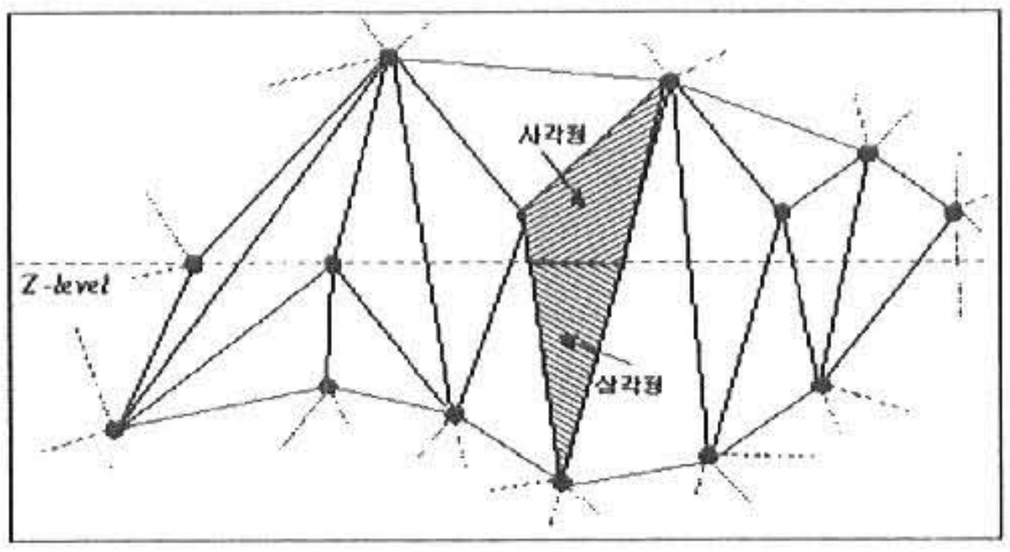

Fig.1 An example of spread flat form of STL data near the slice z-level.

Fig. 1은 STL data를 CAD시스템에서 한쪽 면을 절단하여 관심있는 z-level 부근에서 펼친 형상을 나타낸 것이다. trianglar facet으로 구성되어 있는 데이터를 z-level에서 절단하면 삼각형 두 개로 나누어지거나 한 개의 삼각형과 한 개의 사각형으로 분리된다. 이때 기존의 삼각형은 데이터베이스에서 삭제하고 새로운 삼각형과 사각형에 대해서, 삼각형은 3개의 vertex정보를 이용하고 사각형은 두 개의 삼각형으로 분리하여 STL facet로 신규 생성하여 STL DB에 저장한다.

2-2. Z-level에서의 구조

Error Check가 끝난 New_STL file을 Final_STL file 이라고 정의하고 이 절단면에서 분리된 trianglar facet들은 기하학적으로 원래 형상의 표면정보를 나타내고 있으므로 polygon형태의 closed loop형상을 가지게 되며 CAD형상모델에 따라서 복합적으로 중첩된 loop들의 집합(SSET)으로 볼 수 있다. 형상모델의 오차가 있는지 점검할 필요가 있는데 Fig. 2는 STL format에 대해 완전한 closed loop가 생성되는가를 점검하고 에러로 판명되면 색으로 표시를 하고 해당 z값을 저장하는 알고리즘의 flow

chart를 나타내었다. Slice방법은 Z 값별로 이루어지고 Z값에 교차되는 trianglar facet을 SSET에 저장한다. 설정된 Z값으로 교차점을 구하여 linked list로 연결된 점 군을 이용하여 closed loop를 생성하면 Slice는 완성된다.

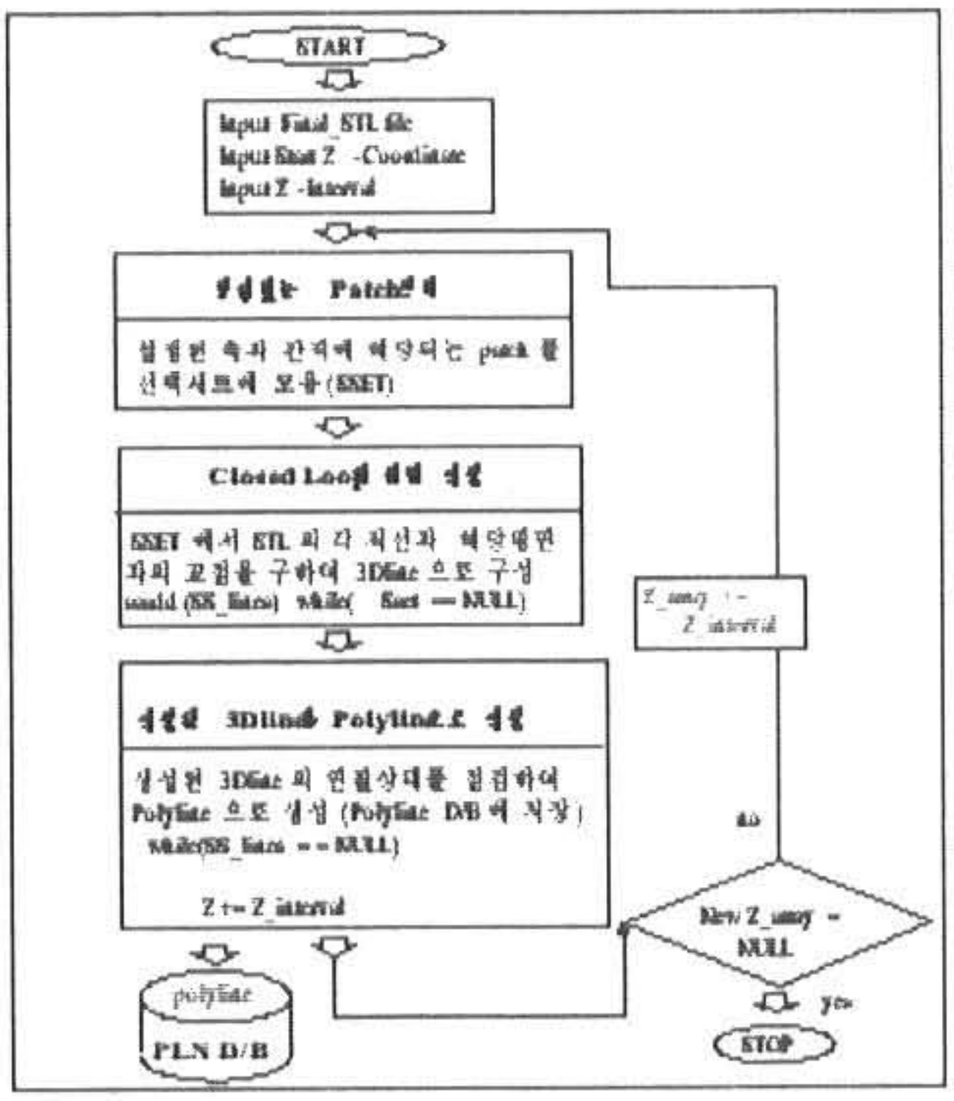

Fig. 2 Flowchart for slice to the STL format at arbitrary z-level.

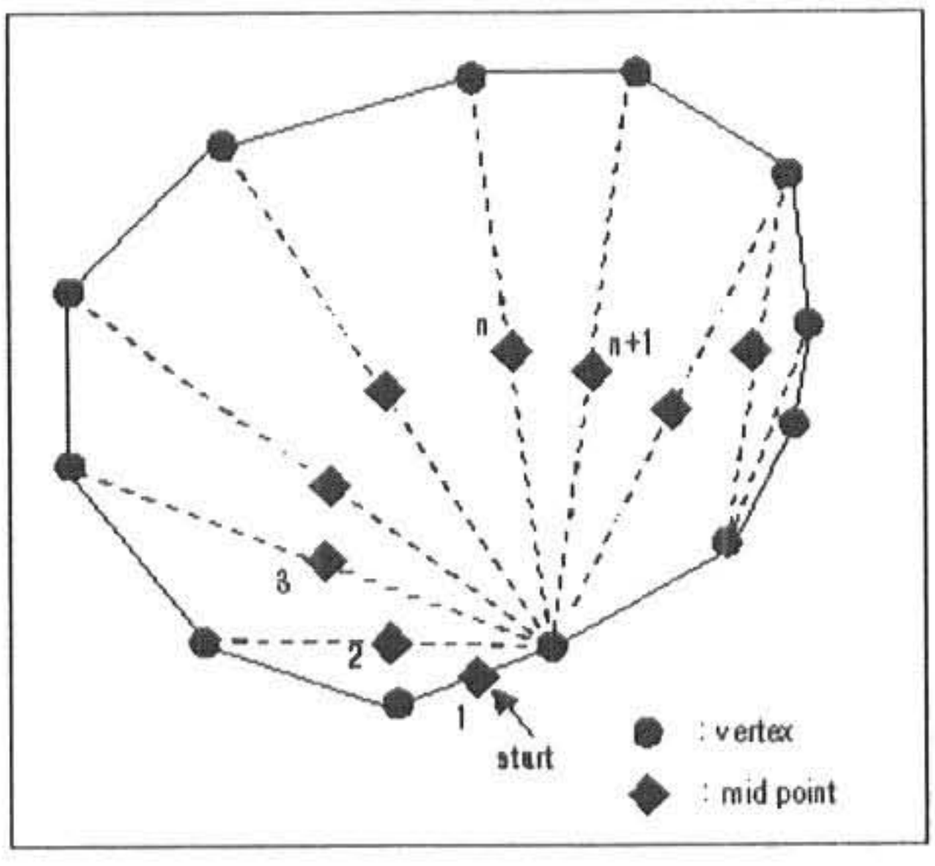

Fig. 3 An example of single loop modification

3. z-level에서의 trianglar facet 복구방법

3-1. 단일 loop의 복구방법

Fig. 3과 같이 중첩되지 않은 single loop의 trianglar facet 채우기는 linked-list된 loop의 시작 entity에 해당하는 연속되는 2개의 vertex로 구성되는 가상 3dline의 중점에서 시작한다. 다음 vertex를 선택하는 방법은 대상이 되는 나머지의 vertex 중에서 가장 가까운 점을 찾는 nearest-neighbour 방식을 취한다. 즉 loop의 linked-list는 vertex의 연속적인 pointer를 참조하며 기준이 되는 2점을 제외한 나머지의 모든 점이 고려 대상이 된다. 이런 방법을 반복적으로 수행하여 작업 loop상의 두 변을 동시에 만나면 작업이 완료된다.

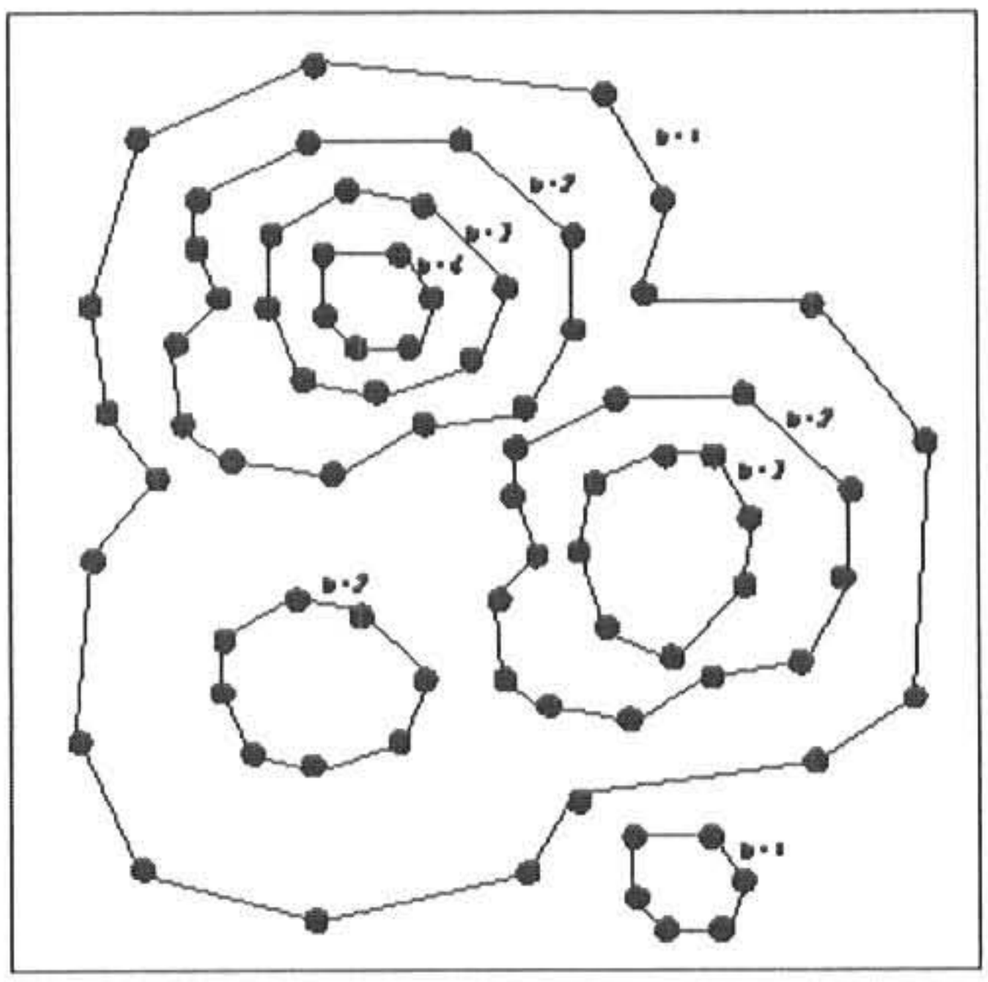

Fig. 4 An example of nested loops modification.

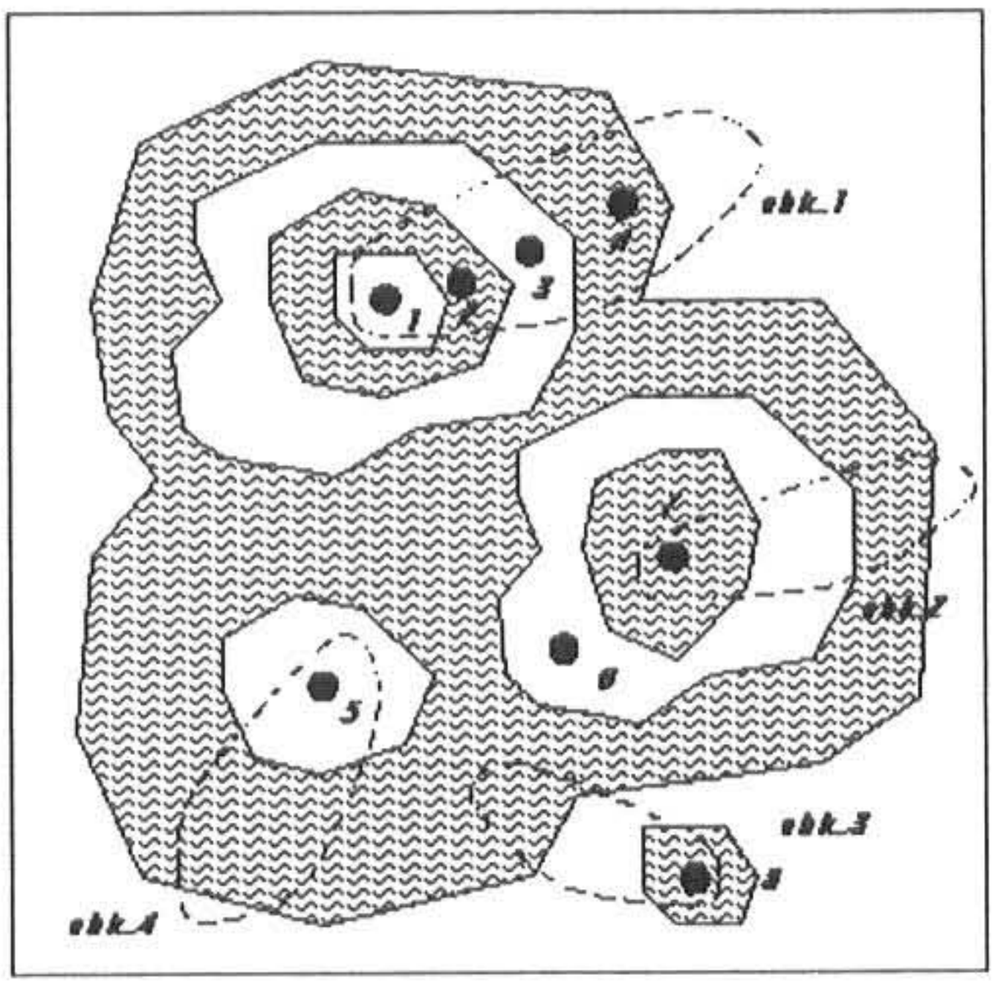

Fig. 5. An example of nested loops topology.

3-2. 복합 loop들의 복구방법

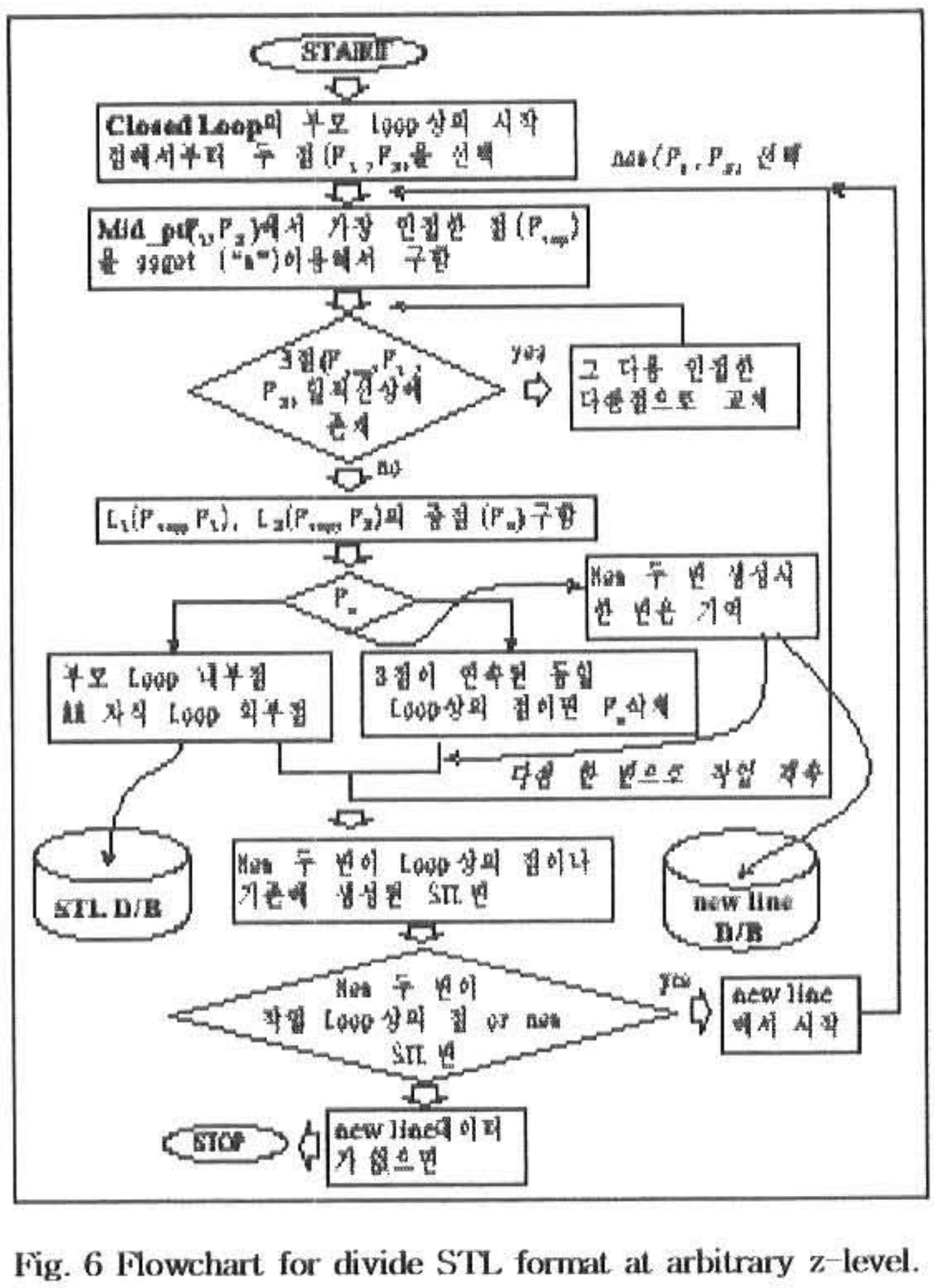

Fig. 6 Flowchart for divide STL format at arbitrary z-level.

Fig. 4와 같이 복합적인 nested loop를 채우는 것은 규칙을 정하기가 매우 까다롭다. b+1 loop를 최외곽 loop라 정하고 중첩된 순서에 따라 b+1, b+2,...b+n으로 구분하였다. 모든 b+1은 같은 조건을 갖고 있으며 loop의 기준이 되면서 parent loop가 된다. b+2,...,b+n에 대해서는 홀수 loop가 parent, 짝수 loop가 child의 특징을 갖는다. 프로그램의 용이성을 위해 Fig. 5와 같이 채워질 영역인 laser 가공영역과 비가공영역으로 구분을 하여 nested loop의 규칙을 정리하였으며 그림에서 빗금친 부분이 laser 가공영역에 해당된다. 또한 임의의 점1~8에서 그 점을 둘러싸고 있는 nested loop의 수를 SSET에 첨가하는 방법을 이용했다. 예를들면 점 1에서 nested loop의 수와 해당되는 loop를 찾아서 다른 SSET_1에 집합으로 첨가하기 위하여 chk_1과 같은 개념으로 loop 내부점을 점검하게 된다. 만약 임의 loop속에 존재하는 점이면 loop상의 모든 vertex와 내부점이 이루는 사이각의 합이 $\pm 2\pi$ 의 값을 가지고 loop의 외부점이면 합이 zero가 된다. 같은 방법으로 모든 SSET내의 loop를 모두 점검하면 SSET_1에 저장된 loop의 개수가 nested loop의 개수가 된다.

중첩된 closed loop는 최외곽의 parent loop와 내부의 child loop로 구분하는데 parent와 인접한 child loop가 짝이된다. STL 파일 복구방법은 부모 Loop의 상의 시작점에서 부터 두점을

```
void divide_stl(ssname, ax_val)
        stl_name ssname;
        stl_real ax_val;
{
int LENGTH;
stl_name ent1, work_sset, tmp_sset;
stl_point bp;
divide_2_part(ax_val);
stl_sslength(ssname, &LENGTH);
                do {
                tmp_sset[0] = tmp_sset[1] = 0;
                work_sset[0] = work_sset[1] = 0;
                get_laser_on_point(ssname, bp, ent1);
                get_working_loops(ssname,bp,work_sset);
                stl_namecpy(tmp_sset, work_sset);
                mak_section_stl(work_sset);
                erase_sub_in_all_sset(ssname, tmp_sset);
                erase_named_pline(tmp_sset);
                del_all_points(ax_val);
                stl_ssfree(work_sset);
                stl_ssfree(tmp_sset);
                stl_sslength(ssname, &LENGTH);
                } while (LENGTH != 0);
}
```

취한 다음, 선택된 두점의 중점을 취해 가장 인접한 점을 window개념으로 선택하고 3점이 일직선인지 검색해서 일직선이면 그 다음 인접한 다른 점으로 교체한다. 또한 2점으로 이루는 직선의 중점을 구해 부모 Loop 내부점인 동시에 자식 Loop 외부점이면 STL DB에 같은 trianglar facet이 존재하는지 여부, Laser_on영역에 존재하는지 여부, loop의 boundary와 교차하는지 여부, 기존의 trianglar facet에 교차하는지의 여부 등을 점검하여 STL D/B에 새로운 trianglar facet을 등록하며 작업이 완료되면 짝이 되는 loop를 DB에서 삭제한다. 만약 3점이 연속된 동일 Loop상의 점이면 가운데 점 D/B에서 삭제시킨다. 만약 새로운 두 변이 생성되면 한 변은 미 작업변에 기억하고 다른 한 변으로 작업을 계속한다. 만약 생성된 두변이 Loop상의 점이나 기존에 생성된 STL변으로 판명되면 그 조건에서의 생성작업 종료를 뜻한다.

그러나 새로운 작업요소가 저장된 데이터베이스에서 작업 변을 선택하여 다시 시작하고 같은 작업을 반복한다. 만약 새로운 .작업 집합의 원소가 없으면 작업을 종료한다. Fig. 6은 divide에 대한 알고리즘을 나타낸 것이다. 여기서 STL D/B와 New line D/B 등이 생성된다. 또한 관련 프로그램인 divide_stl()함수를 위에 나타내었다.

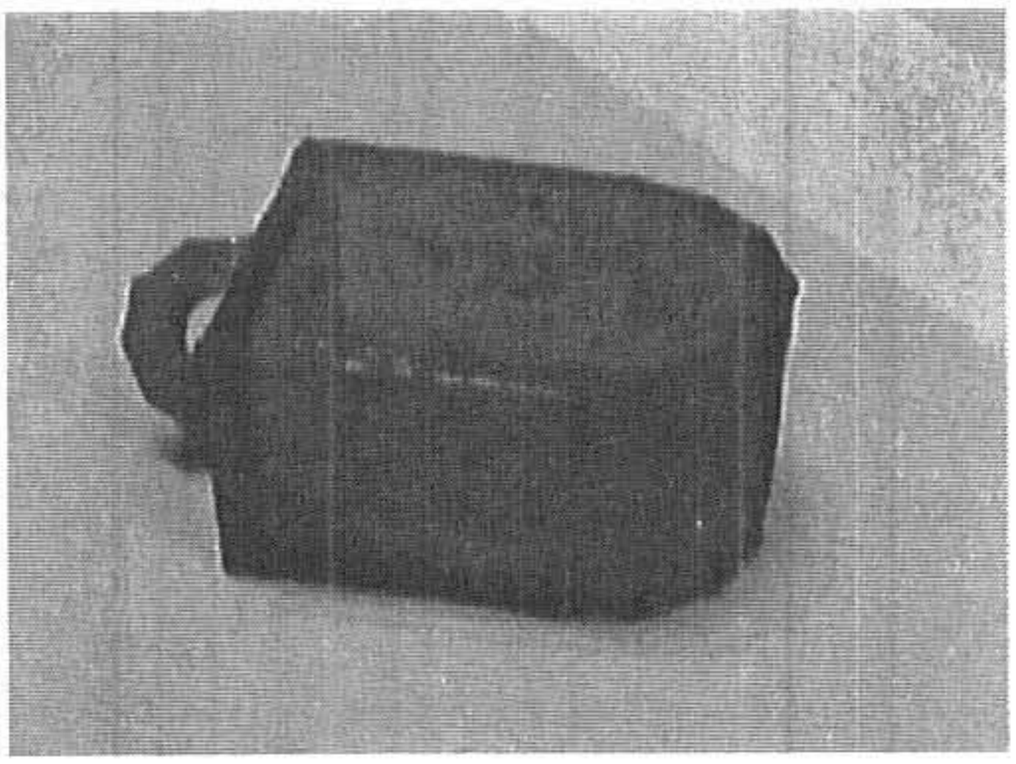

Fig. 7 Test model image.

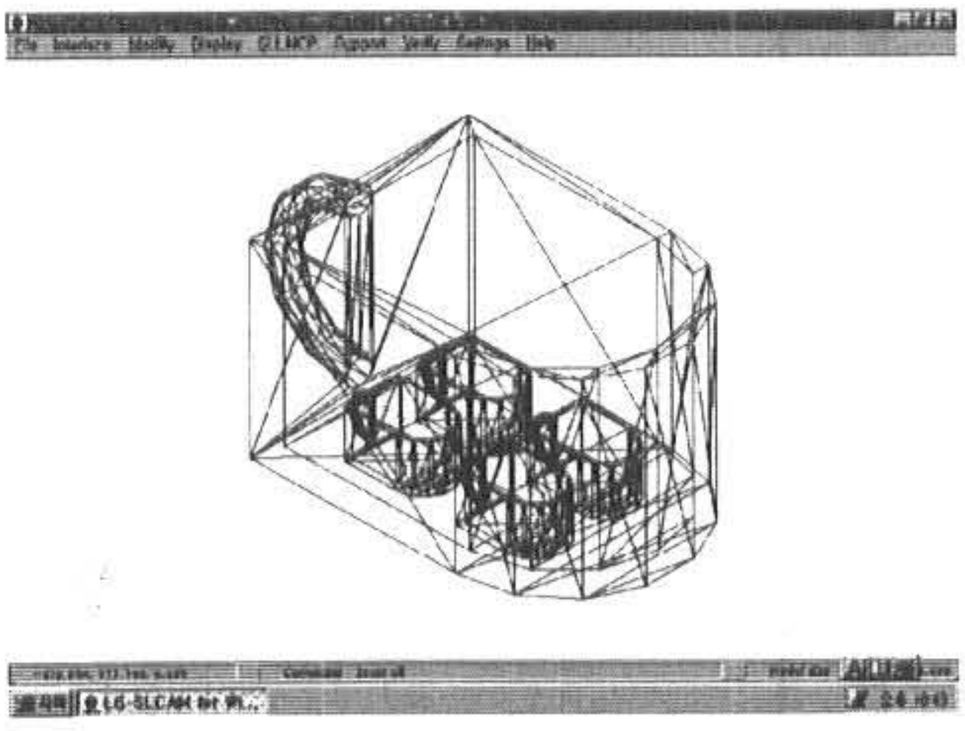

Fig. 8 Test model image for divide.

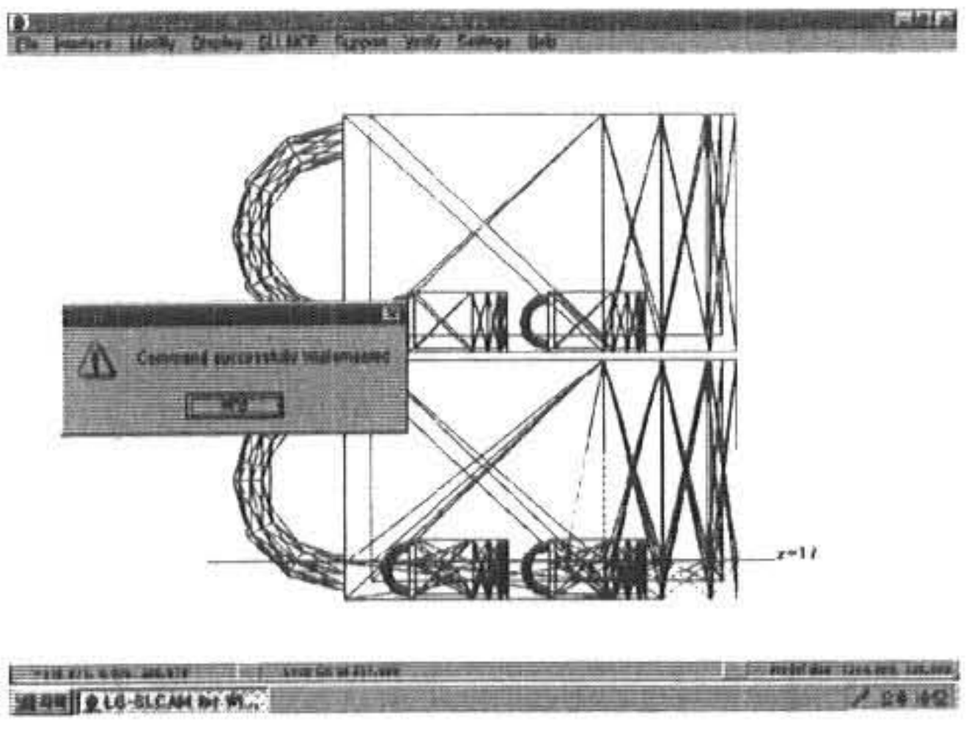

Fig. 9 Test model image divided at z=17.

4. 적용 및 고찰

본 시스템의 편집기능을 이용하여 Fig. 7의 컵을 복합 loop모델을 만들었다. Fig. 8은 테스트 모델의 이미지를 나타낸 것으로 최외곽 컵 안에 4개의 작은 컵이 존재하는 모양이며 view vector는 (1, -1, 1)의 관점에서 본 것이다. Fig. 9는 z=17 위치에서 divide가 완료된 것을 나타내고 있다. 측벽의 절단된 trianglar facet이 절단면 주위로 많이 복구되어 있음을 볼 수 있다.

Fig. 10은 divide된 형상에서 윗부분을 제거한 이미지인데 내부에 있는 4개의 작은 컵도 Fig. 9에서 보는 바와 같이 절단되어 있다. Fig. 11은 내부에 직사각형이 3개 포함된 nest loop의 triangulation 결과이다. Fig. 12는 divide된 면이 최대 4개의 nest loop로 구성되어 있는데 에러없이 좋은 결과를 얻었다. Fig. 12의 triangulation은 PC clock이 100Mhz정도에서 테스트를 수행하였는데 약 25초가 소요되었다.

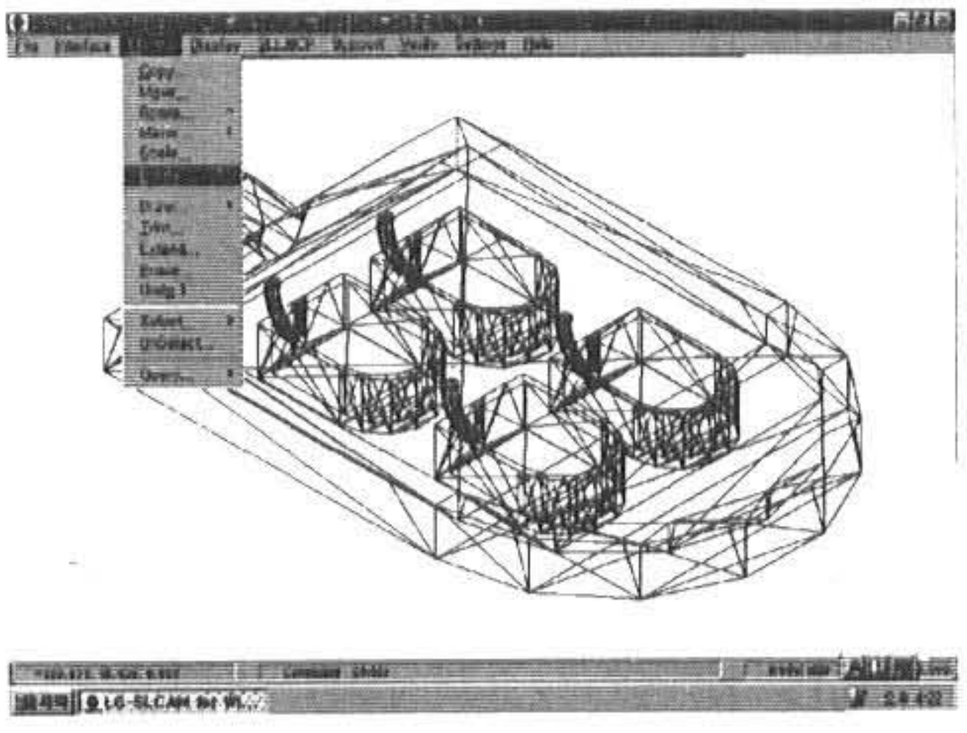

Fig. 10 Erased the upper part image.

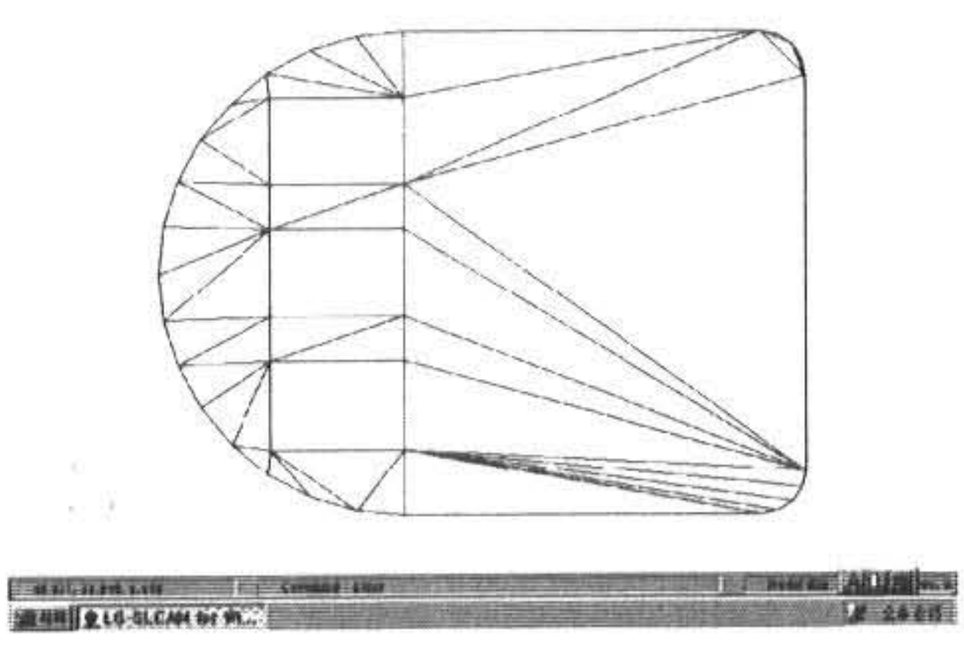

Fig. 11 Fill with trianglar facets in the single polygonal closed loop.

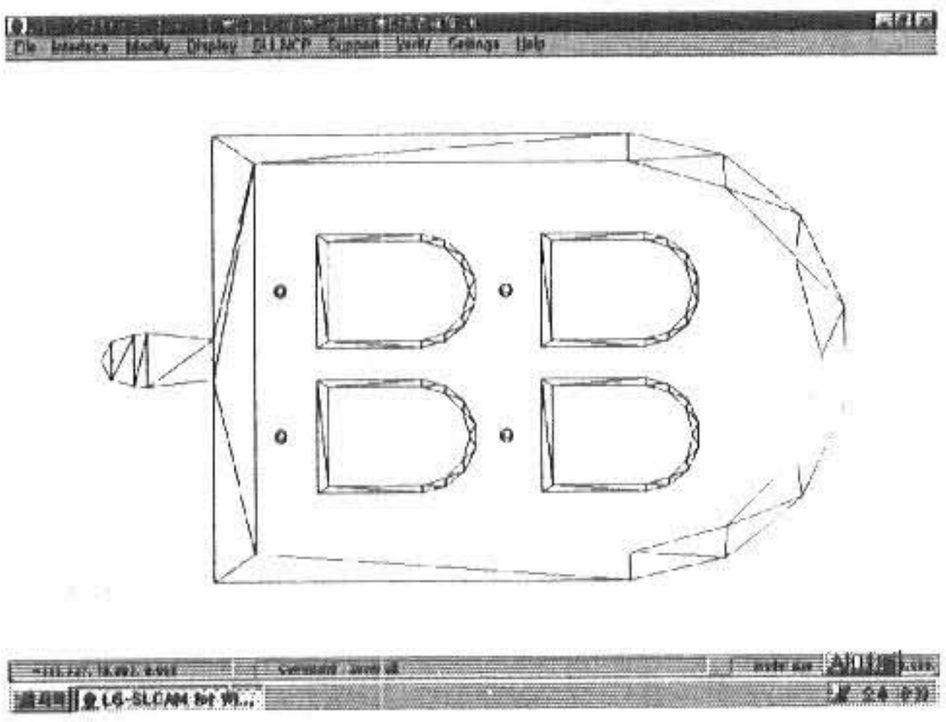

Fig. 12 Fill with trianglar facets in the nested polygonal closed loop.

5. 결 론

현재까지 광조형 장치는 미국, 일본을 중심으로 활발히 개발되어 전 세계적으로 보급되고 있는 실정이며, 국내에서도 활발한 연구가 이루어지고 있다. 본 연구에서는 RP모델 제작시, 제작모델이 제작기의 사양을 초과하는 경우에 STL format을 분리하여 제작하고 다시 결합하는 방법으로 원래 모델을 제작하는 기법을 위주로 다루었다. 이 방법은 모체가 되는 CAD시스템에서 분리하여 운용할 수도 있지만 CAD 모델이 없을 때 유용하게 사용될 수 있다.

또한 단면정보가 중첩된 loop의 형상복구와 주로 형상의 측면에 위치하는 슬라이싱 된 삼각형 facet를 동시에 자동으로 복구하는 알고리즘 연구와 이를 독자적으로 개발한 CAM시스템에 적용해 보았다.

참고문헌

1) Rapid Prototyping Report, CAD/CAM Publishing, Inc., pp.4~6, January (1992)
2) 丸谷洋二, 大川和夫, 中野誠治, 齊藤直一郎, 中井 孝 : 光造形法レーザによる3次元プロッタ, 日刊工業新聞社 (1990)
3) P. Vuyyuru, and C. F. Kirschman :A NURBS-Based Approach for Rapid Product Real ization, Preceeding of the Fifth international Conference on Rapid Prototyping, Dayton, Ohio, pp.229-238 (1994)
4) 최희창 : STL 오류수정 및 형상수정 시스템의 개발, 한국정밀공학회지, 제16권, 제3호, pp.53-60 (1999)
5) 손영지, 박정보, 김순경, 김중완, 전언찬 : STL 포맷의 오류수정을 위한 DB구축과 형상복구에 의한 효율성 연구, 한국정밀공학회지, 제15권, 제12호, pp. 21-27 (1998)
6) 최홍태, 이석희 : 급속조형시스템을 위한 STL포맷 오류 검증에 관한 연구, 한국정밀공학회지, 제13권, 제10호, pp. 46-55 (1995)
7) 반갑수 외3 : 광조형용 CAD/CAM 시스템 개발, 한국정밀공학회, 96년도 춘계학술대회 논문집, pp. 592-596 (1996)

(2002년 2월 6일 접수, 2002년 5월 25일 채택)