

Delaunay 삼각형 분할법의 RP에의 응용

김대원(전북대 대학원), 채희창(전북대 기계공학부)

The Application of Delaunay Triangulation on RP

D. W. Kim(Graduate school, Chonbuk University), H. C. Chae(Chonbuk University)

ABSTRACT

STL which is used in Rapid Prototyping is composed of a lot of triangular facets. The number of triangles and the shapes of these triangles determine the quality of STL. Therefore, proper algorithm is necessary to enhance the quality of triangular patch. In this paper we used the Delaunay triangulation method to apply to following processes. 1) On processing for reducing sharp triangles which cause errors on intersection. 2) On processing for connecting two or more collinear edges. 3) On processing for deleting unnecessarily inserted points in coplanar polygon.

Key words : Rapid Prototyping(RP, 급속 조형), STL, Delaunay Triangulation(Delaunay삼각형분할), Sharp triangle(날카로운 삼각형), Intersection(교차연산), Collinear edge(동일 직선상 모서리)

1. 서론

Rapid Prototyping System(급속 조형 시스템, 이하 RP)은 최초의 상용 광조형 장치가 미국의 3D Systems Inc.[Charles Hull]에 의해 개발되고, 1987년 미국의 디트로이트에서 개최된 AUTO-FACT show에서 첫선을 보인 이래, 약 10년여 동안 그 연구가 계속되어 오고 있다. 최근 국내에서도 연구기관, 대학, 기업체 등이 시스템을 도입해 제품 생산 단계에 적용하기 위한 연구를 활발히 진행하고 있다.

RP의 대표적인 방법인 광조형법은 절삭에 의해 생산하는 방법이 아니고 광 경화성 수지에 레이저 빔을 조사하여 층별로 경화시키는 성형방법이며 그 입력 수단으로 STL(StereoLithography) format을 사용한다. STL format은 3D Syst-

ems사가 개발한 SLA(StereoLithography Apparatus)광조형 장치를 위한 데이터 형식으로, 현재 RP의 표준으로 사용되고 있으며, 3차원으로 모델링되어진 데이터를 삼각형 단위의 facet으로 근사화시킨 것이다. AutoCAD, Pro/Engineer, CATIA, I-DEAS등의 각종 CAD 소프트웨어들은 3차원 모델을 STL로 변환시킬 수 있는 변환기들을 지원하고 있다.

변환기에 의해 STL로 변환된 데이터는 삼각형을 표현하기 위한 세 꼭지점과 법선 벡터를 가지고 있다. 그러나, 변환되는 과정에서 동일한 꼭지점의 중복 등으로 인해 자료의 양이 방대해지고, 면의 누락이나 중복 등과 같은 오류가 발생하기도 하며 극히 작은 각을 가진 예각삼각형(acute triangle)으로 인해 연산과정 중에 오류가 발생할 가능성도 높아진다. 따라서, RP시스템에

입력하기 전에 반드시 오류를 확인하고, 수정해야만 한다.

STL format 파일의 생성과 문제점을 지적하고 수정하려는 연구가 국내외에서 활발히 진행되고 있다. 채희창[1] 등은 위상정보를 가지고 있지 않은 STL에 위상정보를 부여한 후, 각종 기하학적인 처리를 하기 위한 삼각형기반 형상모델링의 데이터 구조를 제안하였다. 손영지[2] 등은 윈도우 프로그래밍 기법을 이용한 STL viewer를 개발하고 구멍 오류의 형태를 분류하여 수정할 수 있도록 하였다. 최홍태[3] 등은 STL의 삼각형 facet에 대해 edge오류, 중복, 구멍오류에 대하여 검증하였다. Fumiki Tanaka[4] 등은 STL파일에서 구멍 오류에 대하여 점데이터를 2차원 평면에 투영하고, 이에 Delaunay 삼각형 분할을 적용하는 삼각패치 재구성법을 제안하였다.

앞서 언급한 바와 같이 STL파일은 상용 CAD 소프트웨어에서 제공하는 변환기에 의해서 출력되어진다. 3차원의 형상모델이 삼각화 되어진 패치로 변환되면서 가지고 있던 형상정보가 손실되어질 수 있다. 점, 모서리, 면 등의 중복이나 누락, 예각 삼각형의 생성 등이 그것이며 STL의 문제점으로 대두되고 있다. 현재 상용 CAD시스템의 STL변환기의 성능은 생성된 STL의 질적인 면에서는 우수한 편이 못되며 불필요하게 삼각형 개수를 늘려놓은 경우가 많다.

본 연구는 Delaunay 분할법을 적용하여 삼각형 분할을 함으로써 편집 과정에서 신뢰성 있는 STL데이터를 추출해 질적인 향상을 꾀하는데 그 목적이 있다. 일반적으로 Delaunay 분할법은 볼록한 외부경계(convex hull)를 가진 많은 수의 점군 데이터에서 내부의 점들을 삼각형화 하는 알고리즘이다. 그러나, RP에서 사용되는 STL파일은 볼록한 형태만 가진 것이 아니고 복잡한 형상 즉, 오목한(concave) 형상이나 내부에 구멍(hole)이 존재하는 경우, 혹은 두 가지가 혼합된 경우가 많으므로 알고리즘을 적용하기가 쉽지 않다. 그러나, STL 수정시의 해당 부분은 많은 수의 점을 가지지 않는 단순한 형태이기 때문에 충분히 적용 가치가 있다.

2. STL 편집용 소프트웨어에서 삼각형 분할의 적용

2.1. STL파일 및 집합연산시 발생하는 문제점

(1) Diet 의 필요성

STL Editor에서 STL을 수정하는데 삼각형 수가 많은 경우에 화면도시 및 수정작업의 속도가 저하된다. 화면 디스플레이 속도는 삼각형의 개수를 n 이라 할 때 $O(n)$ 이나, 집합 연산등을 위하여 삼각형 사이의 intersection을 시킬경우 $O(n^2)$ 이 되므로 작업 효율이 더욱 저하된다.

Fig.1은 다각형 내부에 불필요한 꼭지점이 삽입되어 있는 경우이며 이 경우 불필요한 꼭지점을 삭제하고 관련 다각형을 삼각형으로 재분할하여야 한다.

Fig.2는 STL생성 시 모서리가 너무 작게 분할된 것으로서, 일직선으로 연결된 모서리를 합쳐서 삼각형의 개수를 줄일 필요가 있다. 이 경우에도 삼각형 분할이 필요하다.

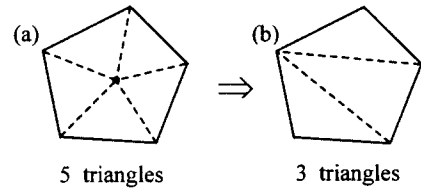


Fig. 1. Triangulation after deleting inner vertex

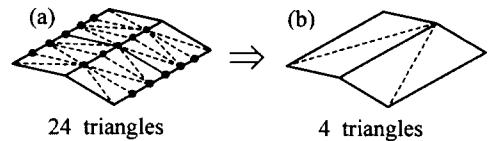


Fig. 2. deleting points on collinear real edges

(2) Intersection시의 문제점

STL 파일에서 예각삼각형, 즉 아주 작은 각을 가지고 있어 거의 일직선에 가까운 삼각형은 집합연산 시에 에러를 발생시킬 인자가 된다. 일부 STL파일은 집합연산이 수행되지 않은 상태로 변환되어져 편집과정에서 다시 집합연산을 시켜주어야 하는 경우가 생긴다. 집합연산을 하기 위해서는 교차연산(intersection)과정이 필수적이다. 그러나 삼각형들을 교차연산을 하게되면 연산도중에 예각 삼각형이 나타날 확률이 현저하게 높

아진다(Fig. 3). 예각 삼각형이 있는 부분에서 교차연산이 진행될 경우에는 점차적으로 더욱 작은 각을 가진 삼각형이 나타나게 된다. 따라서, 교차연산을 하기 전에 예각 삼각형을 제거해 주게 되면 연산과정 중 에러를 일으킬 확률을 줄일 수 있게 된다.

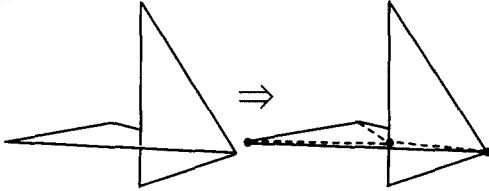


Fig. 3. An acute triangle after intersection.

또한 교차연산 도중에 상관선에 의해 모서리(edge)가 절단되면 불필요하게 삼각형의 개수가 늘어나 데이터의 양이 커지게 되고 소프트웨어의 속도저하를 유발하게 된다.(Fig.4)

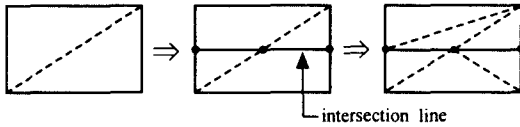


Fig. 4. Increased triangle number after intersection

2.2. 삼각형 분할의 적용 방법

STL파일 자체는 face단위로 저장된 것이 아니고 삼각형 패치(patch)로 구성되어 있어 face의 개념이 없다. STL을 검색하여 동일 평면에 속한 삼각형들을 조사한 후 삼각형 재분할을 할 수 있으나 이 경우 삼각형 분할은 내부의 hole등이 있는 경우가 있어 매우 복잡하게 된다. 따라서 면 단위로 적용하는 삼각형 재분할보다는 개별 보조 모서리에 인접한 삼각형 2개로 볼록한(convex) 사각형을 구성하고 Delaunay삼각형 분할 조건에 적합한지 여부를 판별하여 부적합한 경우 새로운 대각선(diagonal)으로 사각형을 다시 분할하는 방법이 바람직하다(local optimization). 또한, local optimization방법을 적용함으로써 intersection 도중에 예각 삼각형을 발생시킬 확률을 낮추기 위하여, Delaunay삼각형 분할을 부분적으로 적용할 수 있는 장점이 있다.

Delaunay 분할법은 최소 각을 최대로 하는

방법이다. Fig.5에서와 같이 주어진 세 점 (a,d,b)를 지나는 원에 대하여 다른 한 점 c와의 관계를 조사한다. 즉, 점 c가 외접원의 외부에 존재하게 되면 $\triangle adb$ 의 최소 각 θ_1 은 $\triangle bca$ 의 최소 각 θ_3 보다 크게 된다[5][6]. 따라서, $\square abcd$ 에서 대각선은 bd로 정해진다. 만일 대각선이 최초로 ac로 정해져 있었다면 최소 각 검사에 의해 bd로 결정되어진다.(Fig.5, Fig.6)

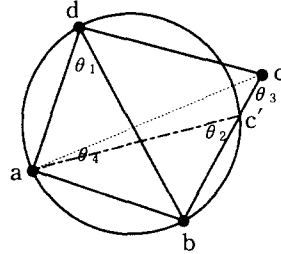


Fig. 5. Illustration for basic concept of Delaunay Triangulation

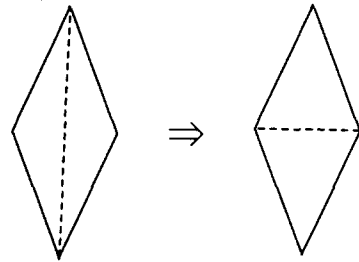


Fig. 6. Swap diagonal after comparing minimum angle of each triangle

삼각형 patch들을 사각형 단위로 나누어 두 삼각형 사이의 최소 각을 비교해 대각선을 결정하고 이웃 삼각형들에 대해 위의 과정을 계속적으로 수행하게 된다.

2.3. 삼각형 분할이 적용되는 단계

STL 파일을 편집하는 과정에서 삼각형 분할이 뒤따르게 된다. 이 때 Delaunay 분할법을 적용하여 분할하게 되며, 적용하는 시점은 다음과 같다.

(1) 데이터의 양을 줄이는 경우(Diet method)

1) STL 파일을 편집기로 불러오는 과정에서 1차적으로 데이터의 크기를 감소시킨다. 꼭지점

사이의 거리에 따른 허용공차(tolerance)를 적용해 공차 이내의 꼭지점과 해당 삼각형을 데이터 베이스에 추가하지 않고 기존의 꼭지점으로 대체함으로써 입력 도중에 데이터의 크기를 감소시킬 수 있다.(Fig.7)

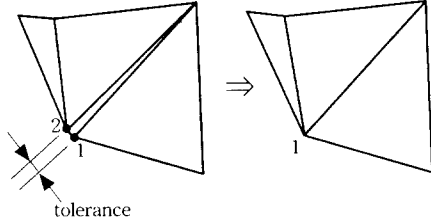


Fig. 7. Calculate the tolerance between points

2) STL의 변환 과정 중 불필요하게 삽입되어진(2.1 (1)) 꼭지점들을 제거한 후 삼각형을 재분할하는 경우에 분할법을 적용한다. 꼭지점의 삭제는 두 가지의 경우로 나뉘며 첫째는 꼭지점의 주위에 보조모서리(auxiliary edge)만 있는 경우이고, 둘째는 꼭지점의 주위에 일직선의 실 모서리(real edge)가 있는 경우이다. 이 때 불필요한 모든 꼭지점을 제거하게 되면 모서리의 길이가 길어질 수 있기 때문에 그 길이 값에 대한 제한을 설정해 준다.

- ① 꼭지점 주위에 보조모서리만 있는 경우(Fig.1)
- ② 꼭지점 주위에 일직선의 실 모서리가 있는 경우(Fig.8)

Fig.8에서는 꼭지점이 실 모서리를 절단하고 있으며 실 모서리의 위아래는 각각 다른 면(face)이므로 삼각형 분할 시에 분리해서 분할하게 된다.

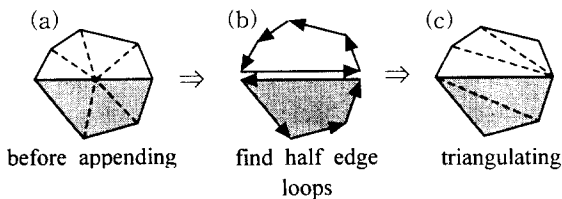


Fig. 8. Appending real edge

(2) 예각삼각형에 대한 국부 최적화(local optimization)의 경우

교차 연산을 수행할 경우 기존의 예각삼각형

이 있는 위치에서 연산이 수행되면 더욱 작은 각을 가진 삼각형이 발생하고 그들 사이에 서로 상충작용, 즉 진행이 되면 될수록 각이 작아지는 결과가 생겨 연산 작업에 에러를 발생시키게 된다. 따라서, 연산 이전에 미리 날카로운 삼각형을 없애기 위해 Delaunay 삼각형 분할법을 적용시킨다.(Fig.9)

먼저, 보조 모서리에 이웃하는 2개의 삼각형들로 사각형 단위를 구성한다. 불룩한 사각형의 경우에만 사각형 내의 대각선에 대해 서로의 최소 각을 비교하고, 이웃하는 보조모서리들에 대해 대각선(diagonal)이 될 수 있는지 없는지를 판별한다.(Fig.10)

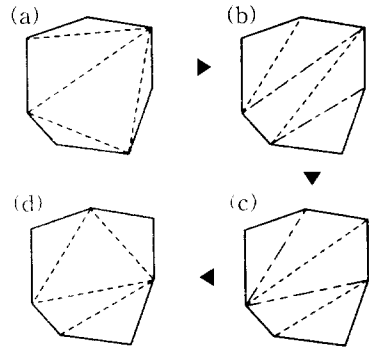


Fig. 9. Find diagonals for quadrangle before intersection process.

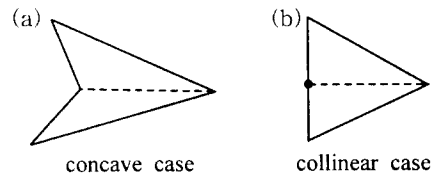


Fig. 10. Impossible case of swapping diagonal

교차연산을 수행한 후 생성된 모서리에 대해서 다시 한번 최적화를 하게 된다.

3. 소프트웨어 시뮬레이션

3.1. 소프트웨어 개발환경 및 특징

본 연구의 Delaunay 삼각형 분할법을 이용한 diet 및 STL 최적화를 STL편집용 소프트웨어 개

발에 응용하여 유효성을 검증하였다. 플랫폼으로는 WindowsNT를 사용하였고, 개발 도구는 Borland C++ Builder와 Microsoft의 그래픽 라이브러리인 OpenGL과 Gary의 OpenGL component인 TOpenGL을 사용하였다.

GUI(Graphic User Interface)환경에 적합하도록 모든 명령을 버튼메뉴로 구성하였고, 마우스와 키보드를 이용, 상용 CAD소프트웨어에서와 같이 zooming, panning, rotation작업을 할 수 있도록 하였다. 또한 STL에 대한 정보와 오류가 발생한 부분등을 신속히 알아볼 수 있도록 하였으며, Union, Intersection, Subtraction의 기본 연산 기능도 갖추고 있다.

3.2. 시뮬레이션

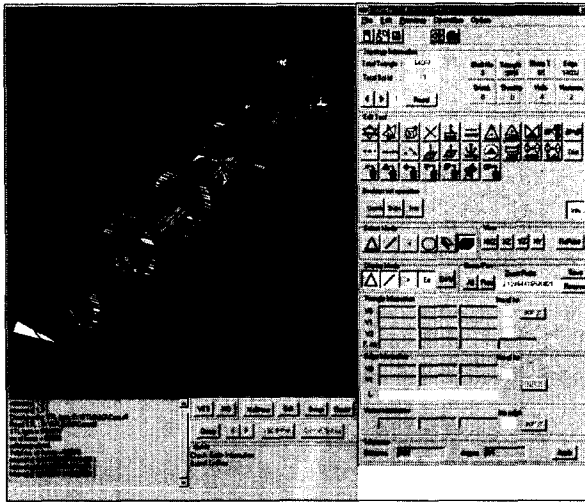


Fig. 11. Full screen of STL editor

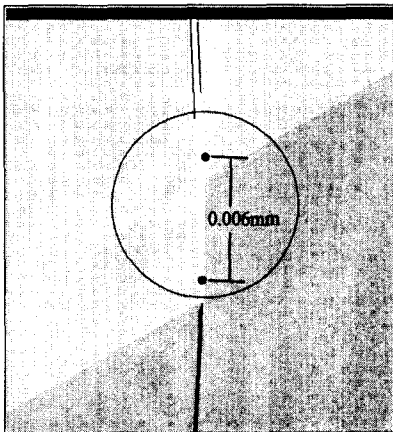


Fig. 12. Sharp triangle can be removed by applying tolerance.

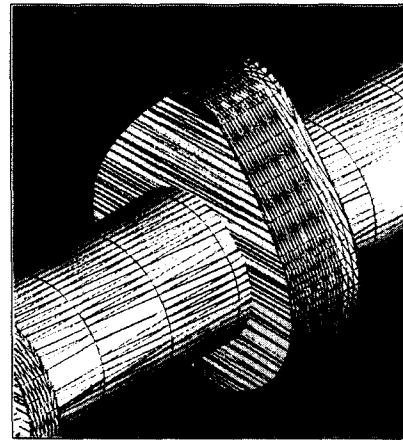


Fig. 13. Before diet

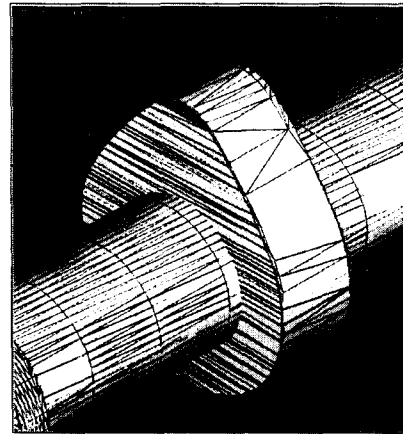


Fig. 14. After diet

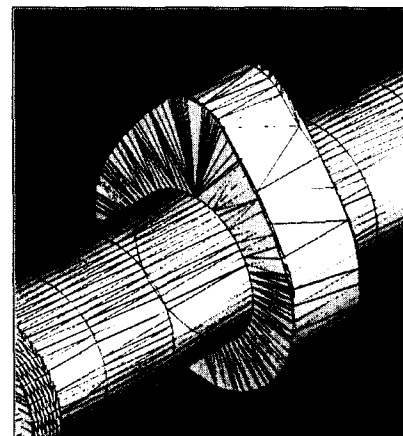


Fig. 15. After intersection and optimizing model

Fig.11은 소프트웨어의 전체 화면을 표시하고 있다. 좌측 위 부분은 그래픽 창이며, 아래쪽은 메시지 창이다. 우측은 메뉴 창이며 메뉴 창에는 파일의 입출력과 각종 수정 작업을 위한 버튼 메뉴, 선택모드 스위치, 표시모드 스위치 등으로 구성되어 있다.

Fig.12는 STL 파일에서 각각 다른 면에 위치한 두 개의 서로 다른 꼭지점간의 거리를 약 31006배로 확대해 나타낸 그림이다. 두 꼭지점 사이의 거리는 0.006mm이며 STL이 만들어진 당시 생성된 꼭지점이다.

Fig.13은 불필요한 포인트, 모서리, 삼각형을 삭제하지 않은 STL 입력후의 데이터이다. 그림에서 보면 알 수 있듯이 많은 수의 모서리와 삼각형이 있다.

Fig.14는 교차하고 있는 두 모델에서 Diet 과정을 적용한 모습이다. Fig. 13과 비교했을 때 삼각형의 수가 현저히 줄어든 것을 볼 수 있다.

Fig.15는 교차하고 있는 두 모델을 교차연산 후 최적화 시켜준 후의 모습이다. 앞쪽 면의 삼각형의 형태가 수정 전의 모습인 Fig.14에서 보다 많이 향상된 것을 알 수 있다.

Table 1은 입력전의 original 모델과 입력시 tolerance를 적용한 경우, Diet를 적용한 후의 삼각형의 수를 각각 비교한 것이다.

Table 1. Comparison of no. of triangle for each case

		No. of triangle
Original model		66976
Apply tolerance	0.001	66495
	0.005	66047
	0.01	64007
Apply Diet		17623

4. 결 론

STL은 변환되는 과정에서 불필요한 꼭지점들의 삽입으로 인해 그 데이터가 방대해지고, 날

카로운 삼각형을 많이 포함해 연산과정의 안정성을 떨어뜨린다.

본 연구에서는 STL의 질을 높이고 삼각형 수를 줄여 수정작업의 효율성을 높이기 위하여 Delaunay triangulation을 적용하였으며 다음과 같은 결론을 얻었다.

【1】 STL을 입력하는 과정에서 허용공차를 적용해 공차 이내의 꼭지점을 데이터베이스에 추가하지 않고 이전 꼭지점으로 대체함으로써 삼각형의 수를 줄일 수 있다.

【2】 Diet 기법을 적용함으로써 삼각형의 수를 대폭 줄일 수 있다.

【3】 교차연산시 예각삼각형에 의한 오류를 줄이기 위하여 local optimization을 적용하였고 그 과정에 Delaunay 삼각형 분할법을 사용함으로써 STL의 질을 향상시킬 수 있다.

참고 문헌

- [1] 채희창, "STL에 위상정보를 부여하기 위한 삼각형 기반 형상모델링", 한국정밀공학회지 제14권 제2호(통권71호).
- [2] 손영지, 조연상, 이승수, "STL 포맷의 구멍 오류 수정을 위한 삼각형 분할법 적용에 관한 연구", 한국정밀공학회 '97추계학술대회 논문집 pp.889~893.
- [3] 최홍태, 이석희, "급속조형시스템을 위한 STL포맷의 오류 검증에 관한 연구", 한국정밀공학회 '96추계학술대회논문집 pp.597~601.
- [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 日本光造形システムシンポジウム, pp.39~45, 1994.
- [5] D.T.Lee and A.K.Lin, "Generalized Delaunay Triangulation for Planar Graphs", Discrete Comput Geom 1:201-217(1987)
- [6] Joseph O'Rourke, "Computational Geometry in C".