

# 오차조건을 만족하는 STL 모델의 불균일 변형 방법

## Non-Uniform Deformation Method of a STL Model Satisfying Error Criteria

\*김봉관<sup>1</sup>, #김대웅<sup>1</sup>, 이강수<sup>2</sup>, 김성환<sup>1</sup>  
 \*B. K. Kim<sup>1</sup>, #D. W. Kim(xxxx3303@naver.com)<sup>1</sup>, K. S. Lee<sup>2</sup>, S. H. Kim<sup>1</sup>  
<sup>1</sup> 서울산업대학교 기계설계자동화공학부, <sup>2</sup> 한밭대학교 기계공학부

Key words : STL model, Grading, Facet Subdivision, Error Criteria

### 1. 서론

신발, 의류 분야는 대상물을 CAD 시스템을 이용하여 3 차원 모델링하는데 특성상 하나의 디자인에 여러 개의 사이즈가 존재하므로 기준이 되는 사이즈를 하나 모델링하고 이로부터 여러 개의 치수를 자동으로 모델링 해주는 시스템이 개발되었는데 이를 그레이딩(grading)이라 한다. 그레이딩은 일반적으로 불균일 변형을 동반하는데 곡면모델에 대한 그레이딩 연구는 김성환<sup>[1]</sup> 등이 수행한 바 있다. 한편, RP(Rapid Prototype)를 제작하는 관점에서 보면 STL 데이터가 필요하게 되는데 이는 서페이스 모델이 아닌 솔리드 모델로부터 얻어지게 된다. 하지만 기준이 되는 사이즈에서 그레이딩되어진 모델은 솔리드로 sewing 되어지지 않을 수도 있다.

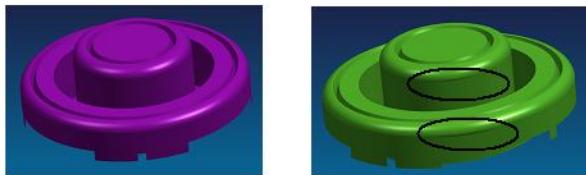


Fig. 1 Example of surface grading

Fig. 1의 좌측은 기준이 되는 사이즈를 모델링 한 것이고 우측은 이를 특정 사이즈로 그레이딩 한 것이다. 우측 그림을 보면 미세하게 틈이 벌어진 것을 볼 수 있다. 이론적으로는 이러한 틈새가 생기지 않아야 하지만, 맞닿는 두 개의 곡면 경계에서 서로 다른 포인트 샘플링이 이루어지기 때문에 중간부분의 보간(Interpolation)이 상이하게 이루어져 생기는 문제이다. 현재 이를 없애는 것은 매우 어려우며 기준 사이즈의 모델을 STL로 만들어 이를 그레이딩 하는 방법을 모색할 필요가 있다.

STL 모델을 그레이딩 하려면 우선 주어진 구속조건을 만족하는 부드러운 3 차원 벡터장을 생성하는 것이 필요하고, 벡터장을 통해 모델을 변형하면 된다. 즉 STL 모델은 삼각형으로 이루어져 있기 때문에 STL 모델의 그레이딩은 벡터장에 의해 정의되는 변형량만큼 꼭지점을 움직여주면 쉽게 얻어질 수 있다.

그러나 삼각형의 크기가 큰 부분에서는 큰 오차가 발생할 수 있다.

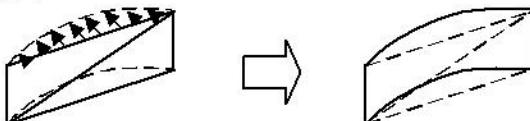


Fig. 2 Example of a geometric error in STL grading

Fig. 2는 이와 같은 문제가 발생하는 변형의 예를 보여준다. Fig. 2 좌측의 실선이 변형 전의 삼각형을 의미하고 우측의 실선이 원하는 형상이며 STL 모델이 그림과 같이 2개의 삼각형으로만 이루어져 있다면 존재하는 꼭지점만 이동해서는 아무런 변화를 얻을 수 없게 된다. 이러한 문제는 특히 모델 곡면의 곡률이 작은 형상이 곡률이 커지도록 변형되는 경우에 원래 형상에서 STL 모델 삼각형의 크기가 클 것이기 때문에 심각해 진다.

본 논문에서는 주어진 오차조건을 만족하는 변형된 STL

모델을 얻는 방법을 기술한다. 이를 위해 오차를 모서리와 면에서 검사하여 필요에 따라 분할(subdivision)하는 방법을 제안한다.

### 2. 모서리의 최대 오차점 찾기

먼저 모서리의 네 점( $u=0, 0.33, 0.67, 1$ )의 위치( $P_{10}, P_{30}, P_{40}, P_{20}$ )를 찾고 이를 벡터장을 이용해 변형된 위치( $P_{11}, P_{31}, P_{41}, P_{21}$ )를 찾는다. 이때 생성된 벡터는 모서리에 수직하지 않을 수 있으며 변형된 위치를 두 꼭지점을 연결하는 직선에 투영하여 오차 벡터를 찾는다. Fig. 3에서  $P_{10}, P_{20}$ 은 출발 및 끝 꼭지점이고  $P_{30}$ 은  $u=0.33$ 인 위치이다. 이 점은 벡터장을 거쳐  $P_{11}, P_{21}, P_{31}$ 로 옮겨가며,  $P_{32}$ 는  $P_{11} P_{21}$ 에서  $u=0.33$ 인 위치이다.  $P_{31}$ 에서 투영한 점, 즉, 최단 거리를 이루는 점  $P_{33}$ 은 Fig. 3에 나타난 수식을 통해 구한다. 또한  $u=0.67$ 인  $P_{43}$ 도 같은 방법으로 구한다.

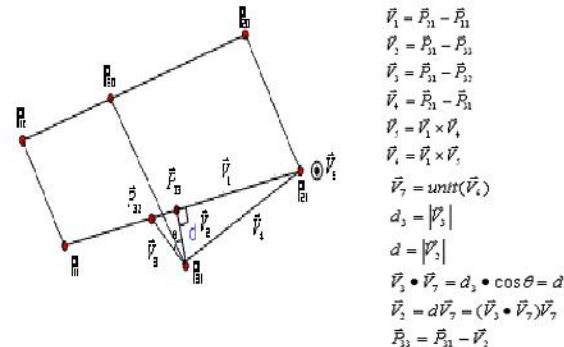


Fig. 3 Finding the point  $P_{33}$  and vector  $V_2$

Fig. 3과 같은 방법으로 벡터장으로 계산한 변형 된  $P_{11}, P_{31}, P_{41}, P_{21}$  및 그레이딩된 위치인  $P_{13}, P_{33}, P_{43}, P_{23}$ 을 구할 수 있다.  $V_{11}, V_{22}, V_{33}, V_{44}$ 를 각 위치에서 그 차이를 나타내는 벡터로 정의하면 통상  $V_{11}$ 과  $V_{44}$ 는 0이다. 네 점  $P_1, P_2, P_3, P_4$ 를 알 때 3 차 곡선식은 Mortenson<sup>[2]</sup>의 수식을 사용하였고  $P_1, P_2, P_3, P_4$  대신  $V_{11}, V_{22}, V_{33}, V_{44}$ 를 대입하였다.

$$\vec{p} = (13\vec{p}_2 - 13\vec{p}_3)\vec{u}^3 + (-22\vec{p}_2 + 18\vec{p}_3)\vec{u}^2 + (9\vec{p}_2 - 4.5\vec{p}_3)\vec{u} \quad (\text{식 } 1)$$

$$= \vec{A}_1\vec{u}^3 + \vec{A}_2\vec{u}^2 + \vec{A}_3\vec{u} \quad (\text{식 } 1)$$

(식 1)를 거리의 공식을 사용하여 표현하면

$$\vec{p} = x\hat{i} + y\hat{j} + z\hat{k} \rightarrow l = \|\vec{p}\| = \sqrt{x^2 + y^2 + z^2} \quad (\text{식 } 2)$$

(식 2)와 같이 기술 할 수 있고 이를 미분하면

$$\frac{dl}{du} = \frac{d\sqrt{x^2 + y^2 + z^2}}{du} = \frac{2x\dot{u} + 2y\dot{u} + 2z\dot{u}}{2\sqrt{x^2 + y^2 + z^2}} = \frac{xx\dot{u} + yy\dot{u} + zz\dot{u}}{\sqrt{x^2 + y^2 + z^2}} = xx\dot{u} + yy\dot{u} + zz\dot{u} = 0 \quad (\text{식 } 3)$$

여기서,

$$xx' + yy' + zz' = u[3(A_{1x}^2 + A_{1y}^2 + A_{1z}^2)u^4 + 5(A_{1x}A_{2x} + A_{1y}A_{2y} + A_{1z}A_{2z})u^2 + ((4A_{1x}A_{3x} + 2A_{2x}^2) + (4A_{1y}A_{3y} + 2A_{2y}^2) + (4A_{1z}A_{3z} + 2A_{2z}^2))u^2 + 3(A_{2x}A_{3x} + A_{2y}A_{3y} + A_{2z}A_{3z})u + (A_{3x}^2 + A_{3y}^2 + A_{3z}^2)] = 0 \quad (\text{식 } 4)$$

이다. 위의 식에서  $u=0$ 과 괄호 안의 4 차 방정식 [0,1]의 범위에서 최대 4 개의 해  $u_1, u_2, u_3, u_4$ 가 얻어질 수 있고 해당 점  $m_1, m_2, m_3, m_4$ 가 변곡점이며, 최대 오차점은 이 변곡점에서 나타난다. 두 꼭지점 사이에 변곡점이 없는 경우, 변형 후에도 직선인 경우로 오차가 발생하지 않는다.  $u$ 가 [0, 1] 사이의 범위에서 최대 변곡점이 허용 오차보다 작으면 이 모서리의 최대 변형 오차는 공차 안에 들며 최대 변

곡점의 위치  $M_i$ (변형 전 위치)와 오차  $e_i$ 를 기억해 이후 삼각형의 최대 오차점을 찾는데 사용한다. 두 꼭지점 사이의 변곡점에서 허용 공차보다 오차가 크면 최대 변곡점에서 분할하고 이웃 삼각형도 분할하여 다시 오차 검사를 수행한다.

### 3. 면에서의 최대 오차점 찾기

세 모서리에서의 오차측정에 의한 분할이 끝나면 다음으로 삼각형 면에서 최대 오차점을 찾는다. 이때 면에서의 최대 오차점  $M_F$ 는 (식 5)와 같이 표현한다. 여기서 초기 가중치 값은 각 점에서의 최대 오차의 비로 한다. 다음 검사 위치는 Fig. 4 와 같이 네 방향(+u, -u, +v, -v)으로 설정된 비율만큼 탐색하여 검사한다.

$$M_F = \sum_{i=1}^3 w_i M_i + \sum_{i=1}^3 w_{3+i} V_i, \text{ 단, } \sum_{i=1}^6 w_i = 1 \quad (\text{식 } 5)$$

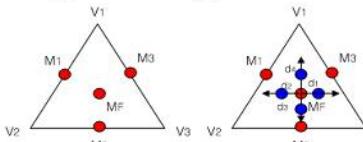


Fig. 4 Finding a maximum error point  $M_F$  in surface

검사의 과정을 반복하여 6 개의  $d_p$ 에서 검사한 값 중 오차가 가장 큰 점이  $M_F$  보다 오차가 클 경우  $M_F$ 를 갱신하고 찾은 최종 값이 최대 오차점  $M_F$ 가 된다.

### 4. 삼각형 분할 알고리즘

**단계 1.** 모서리의 최대 오차점을 조사하여 공차보다 큰 경우, Fig.5 (a)(b)(c)에 따라 삼각형 면을 분할한다. 여기서 (b)의 경우는 두 점에서 오차가 큰 점 먼저 분할한다. 이 후 분할된 삼각형 면은 모두 검사 대상에 다시 포함시켜 검사한다. 이 때 이웃의 삼각형 면이 분할되는 경우는 기준보다 오차가 작게 되므로, 분할 이전 면이 합격한 면이라면 이로부터 분할된 면들도 합격으로 분류하고 아직 검사하지 않은 면이라면 검사 대상으로 분류한다.

**단계 2.** 모서리에 최대 오차점이 공차보다 작은 경우, 면에서의 최대 오차점을 찾는다.

**단계 3.** 면에서 최대 오차점이 공차보다 큰 경우, 그림 Fig.5 의 (d)에 따라 삼각형 면을 분할하고, 분할된 삼각형 면은 모두 검사 대상에 포함시켜 다시 검사한다.

**단계 4.** 면에서 최대 오차점이 공차보다 작은 경우, 이 삼각형 면은 통과시키고 다음 면에 대해 검사한다.

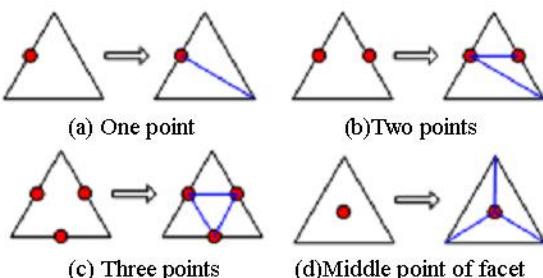


Fig. 5 Method of dividing facet

### 5. 자료 구조

STL 파일은 삼각형의 나열일 뿐 연결 혹은 인접 정보가 없다.<sup>[3]</sup> 그러나 본 논문의 방법은 삼각형의 모서리에서 관련된 삼각형을 탐색할 수 있어야 하므로 연결정보를 손쉽게 알 수 있도록 솔리드의 반모서리<sup>[4]</sup> 자료 구조를 변형한 Fig. 7 과 같은 자료 구조를 사용한다. 대상이 되는 STL 파일이 읽혀질 때 시스템은 자료구조를 채워 완성한다.

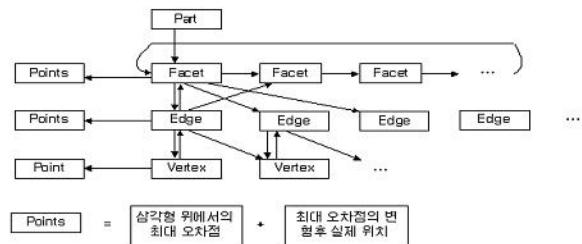


Fig. 7 STL part data structure

Fig. 7에서 Facet 과 Edge 는 검사 여부를 저장하며 Edge 는 Parent Facet 을 2 개씩 가리키고 Parent Facet 이 이 Edge 를 사용할 때 방향성을 나타내는 정보도 저장한다. 삼각형 위에서의 최대 오차점은 양 Vertex 를 선형 보간하여 얻고, 최대 오차점의 변형후 실제 위치는 위의 점을 벡터장을 통해 얻는 것이다.

### 6. 결론

본 논문에서는 주어진 오차조건을 만족하는 변형된 STL 모델을 얻는 위해 오차를 모서리와 면에서 검사하여 필요에 따라 분할(subdivision)하는 방법을 제안했다. 이 방법은 UG/Open 을 이용하여 상용 시스템인 Unigraffic 에 구현되었고 Fig. 8 과 같은 모델에 대해 테스트되었다. 이 방법을 이용하여 원하는 오차 범위 내의 STL 그레이딩 모델을 효과적으로 얻을 수 있게 되었다.

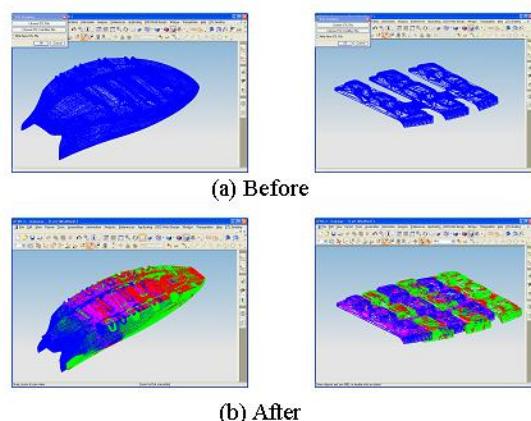


Fig.8 before and after the STL grading

### 참고문헌

1. 김성환, 가이드 곡면을 이용한 곡면의 롤팅, 2007 한국 CAD/CAM 학회 학술발표회 논문집
2. Michael E. Mortenson, Geometric Modeling , pp61~63
3. StereoLithography Interface Specification, 3D Systems, Inc., October, 1989
4. Eastman, Halfedge Data Structure, from <http://groups.csail.mit.edu/graphics/classes/6.838/S98/meetings/m4/T4.HalfEdge.html>, 1982
5. Les A Piegl, Wayne Tiller, Geometry-based triangulation of trimmed NURBS surfaces, Computer-Aided Design Vol. 30, No. 1, pp. 11-18, 1998
6. Chang Shu and Pierre Boulanger, Triangulating Trimmed NURBS Surfaces, Curve and Surface Design: Saint-Malo 1999, pp. 1~8.