한국정밀공학회지 제 22 권 제 9 호 (2005 년 9 월)
Journal of the Korean Society of Precision Engineering, Vol. 22, No. 9, September 2005

솔리드 STL 모델의 옵셋 방법

김수진*, 양민양#

Offset of STL Model Generated from Solid Model

Su Jin Kim*, Min Yang Yang#

ABSTRACT

This paper introduces and illustrates the results of a new method for offsetting triangular mesh by moving all vertices along the multiple normal vectors of a vertex. The multiple normal vectors of a vertex are set the same as the normal vectors of the faces surrounding the vertex, while the two vectors with the smallest difference are joined repeatedly until the difference is smaller than allowance. Offsetting with the multiple normal vectors of a vertex does not create a gap or overlap at the smooth edges, thereby making the mesh size uniform and the computation time short. In addition, this offsetting method is accurate at the sharp edges because the vertices are moved to the normal directions of faces and joined by the blend surface. The method is also useful for rapid prototyping and tool path generation if the triangular mesh is tessellated part of the solid models with curved surfaces and sharp edges. The suggested method and previous methods are implemented on a PC using C++ and illustrated using an OpenGL library.

Key Words : STL (삼각망), Offset (옵셋), Tool Path (공구경로), NC Machining (수치제어가공)

기호설명

$\mathbf{f}_i$ = normal vector of a face
$\mathbf{v}_i$ = multiple normal vectors of a vertex
w = weight factor
δ = offset allowance

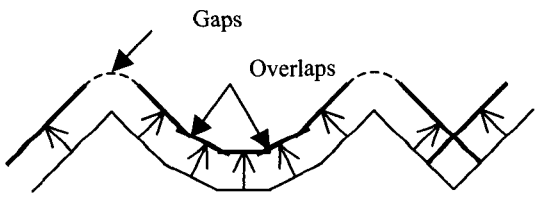
1. 서 론

3D 모델의 옵셋은 내부와 외부를 나누는 경계면을 법선 방향으로 일정한 거리만큼 이동하는 것으로 NC 공구경로 생성, 쾌속 조형, 내부가 빈 모델의 디자인 등에 이용된다. 특히 3D 모델의 NC 가공에서 모델을 공구반경만큼 옵셋 보정한 CL 면을 이용함으로써 간섭이 없는 공구 경로를 효율적으로 생성할 수 있다 [1].

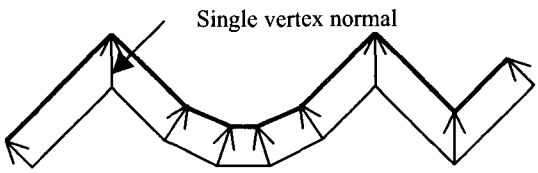
삼각형 면들로 솔리드의 경계를 표현하는 STL 모델은 FEM 해석, 역 공학, 쾌속조형, NC 가공에

접수일: 2004 년 12 월 31 일; 게재승인일: 2005 년 7 월 8 일
* 한국과학기술원 기계공학과
# 교신저자: 한국과학기술원 기계공학과
E-mail myyang@kaist.ac.kr Tel. (042) 869-3264

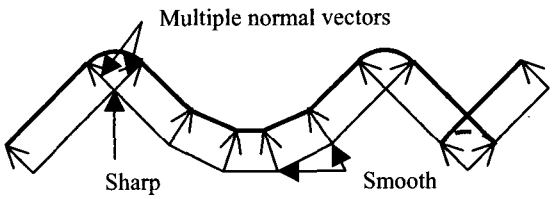
이용 되고 있으며 [2-5], 대부분의 CAD 시스템에서 지원하고 있다. 따라서 STL 모델을 제품의 제조 데이터 생성에 사용하거나 해석에 사용하기 위해서 효율적인 옵셋 방법이 있어야 하겠다. 정밀한 옵셋 모델이 필요한 NC 가공을 위해서는 Fig. 1 (a)와 같이 면을 법선 방향으로 옵셋 거리만큼 이동하는 면 옵셋 방법이 이용되고 있다 [4]. 면 옵셋 방법은 옵셋 오차가 없는 아주 정밀한 옵셋 방법 이지만 평면을 제외한 모든 꼭지점과 모서리에 연결해야 하는 공간과 잘라내야 하는 겹침이 발생해 데이터의 크기가 많이 증가하는 단점이 있다. 데이터의 크기가 작고 균일함이 중요한 FEM 해석이나 쾌속 조형에서는 Fig. 1 (b)와 같이 꼭지점을 법선 방향으로 이동하는 옵셋 방법이 이용되고 있다 [5,6].



(a) Offsetting surfaces along face normal directions



(b) Offsetting vertices along the single normaldirection of a vertex



(c) Offsetting vertices along the multiple normal directions of a vertex

Fig. 1 Offset methods of STL model

꼭지점 옵셋방법은 꼭지점과 모서리에서 간격이 발생하지 않으며 겹침이 발생하는 경우도 적으므로 데이터 크기가 증가하지 않고 삼각형의 크기도 원본과 비슷하지만 정밀도가 크게 떨어지는 문제가 있어서 두 방법을 보완할 수 있는 효율적인 옵셋 방법이 요구된다.

본 연구에서는 주위에 있는 면의 법선 벡터들을 이용해 계산되는 꼭지점의 다중 법선 벡터를 이용하는 새로운 STL 모델의 옵셋방법을 제안한다. 이 방법에서는 Fig. 1(c)와 같이 꼭지점들을 다중 법선 벡터 방향으로 이동시키고 날카로운 꼭지점과 모서리에서 생기는 공간들은 법선 벡터로부터 계산되는 삼각 망으로 연결해 옵셋 망을 만든다. 먼저 꼭지점의 다중 법선 벡터를 소개하고 주위에 있는 면들의 법선 벡터들로부터 꼭지점의 다중 법선 벡터를 계산하는 방법을 설명한다. 다음으로 꼭지점의 다중 법선 벡터를 이용해 STL 모델을 옵셋 함으로써 CL 면을 생성하는 과정을 자세히 설명한다. 마지막으로 기존의 두 방법과 새로운 방법을 비교하고 다양한 STL 모델에 새로이 제안된 옵셋방법을 적용해 봄으로써 알고리즘의 솔리드 모델에서 생성된 STL 모델을 옵셋하는데 효율적임을 보인다.

2. 꼭지점의 다중 법선 벡터

STL 파일은 삼각형 면을 이루는 세 꼭지점과 면의 법선 벡터로 구성되므로 꼭지점의 법선 벡터는 주위 면의 법선 벡터들로부터 예측하게 되는데 주로 한 꼭지점이 하나의 법선 벡터를 가지도록 하고 있다. 본 논문에서는 한 꼭지점이 여러 개의 법선 벡터를 가지도록 하는 꼭지점의 다중 법선 벡터를 제안하고 꼭지점을 둘러싸는 면들의 법선 벡터들로부터 꼭지점의 다중 법선 벡터를 계산하는 방법에 대해 설명한다.

솔리드 모델은 여러 NURBS 곡면들이 연결되어 내부와 외부 경계를 이루는 것으로 하나의 곡면은 부드럽지만 곡면과 곡면이 만나는 모서리와 모서리들이 만나는 꼭지점은 날카롭다. 솔리드 모델에서 생성된 STL 파일은 평면과 모서리 꼭지점으로 이루어 지지만 하나의 곡면에서 생성된 모서리와 꼭지점은 각이 아주 작고 두 곡면이 만나는 곳에서 생긴 모서리와 꼭지점은 날카롭다. 이렇게 STL 모델의 모서리와 꼭지점이 부드러운 경우와

날카로운 경우로 나뉘는 것을 표현하기 위한 방법으로 논문에서는 꼭지점의 다중법선 벡터를 이용하였다. 부드러운 곡면에서 생성된 꼭지점은 둘러싸인 면들의 법선 벡터들로부터 추정되는 하나의 법선 벡터를 가지고, 날카로운 모서리에서 생성된 꼭지점은 연결된 면의 법선 벡터와 비슷한 방향을 가지는 두 개 이상의 법선 벡터들을 가진다.

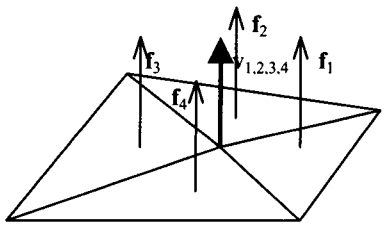
꼭지점의 다중 법선 벡터를 계산하는 순서는 다음과 같다. 먼저 한 꼭지점을 둘러 쌓는 모든 면들의 법선 벡터들을 그 꼭지점의 법선 벡터들로 초기화한다. 다음으로 각 꼭지점에 대해 차이가 가장 작은 두 법선 벡터를 하나의 법선 벡터로 치환하는 작업을 모든 벡터들의 차이가 기준 보다 클 때까지 반복한다.

Fig. 2 (a)과 같이 한 꼭지점을 둘러싸는 면들의 법선 벡터 $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$, $\mathbf{f}_4$ 가 모두 같은 비슷한 방향을 향하는 경우 그 꼭지점은 하나의 법선 벡터 $\mathbf{v}_{1,2,3,4}$ 를 가지게 된다. 또한 Fig. 2 (b)와 같이 두 곡면이 만나는 모서리에서 생성된 꼭지점은 두 개 이상의 법선벡터를 가지는데, 이 때 꼭지점의 한 벡터 $\mathbf{v}_{1,2}$ 는 두 면의 법선 벡터 $\mathbf{f}_1$, $\mathbf{f}_2$로부터 계산 되었고, 다른 벡터 $\mathbf{v}_{3,4}$ 는 다른 두 면의 법선 벡터 $\mathbf{f}_3$, $\mathbf{f}_4$ 로부터 계산 되었다.

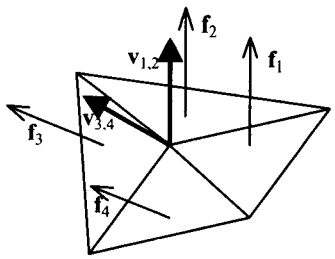
꼭지점의 옵셋 벡터를 계산할 때 꼭지점과 만나는 면의 법선 벡터들의 평균을 꼭지점의 옵셋 벡터로 취하는 방법이 있는데 이것은 꼭지점의 옵셋 거리는 정확하지만 면의 옵셋 거리는 부정확하다. 반면 면의 옵셋 거리가 정확하게 옵셋 되도록 꼭지점의 법선 벡터를 계산하고자 할 경우 식(1)과 같이 면의 법선 벡터에 가중치 w_i 를 곱하고 더해서 꼭지점의 법선 벡터를 계산한다 [6]. 이 방법에서 한 꼭지점과 만나는 면이 세 개이면 정확하게 옵셋 되지만 네 개 이상이면 특수한 경우를 제외하고는 만족하는 가중치가 존재하지 않는다. 따라서 네 개 이상의 면이 한 꼭지점과 만날 경우 면의 옵셋 거리가 정확하게 옵셋 되는 꼭지점의 옵셋 벡터가 존재하지 않으므로 오차가 비교적 작은 옵셋 벡터를 찾아야 한다.

$$\mathbf{v}_{1,\ldots,n} = \sum_{i=1}^{n} w_i \mathbf{f}_i \qquad (1)$$

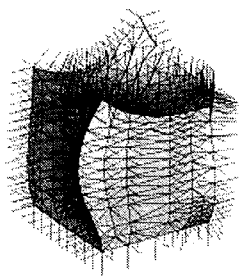
한 꼭지점이 두 개 이상의 옵셋 벡터를 가지는 꼭지점의 다중 법선 벡터를 이용할 경우에는 이론적으로 3N 개 이하의 면이 한 꼭지점에서 만나는 경우에 N 개의 법선 벡터를 계산하면 면의 옵셋 거리가 정확한 옵셋이 가능하다. 하지만 면을 정확히 옵셋 하기 위한 방법은 꼭지점의 옵셋 거리에 오차를 발생 시키며, 꼭지점의 옵셋 거리를 정확하게 하는 경우에는 면의 옵셋 거리에 오차가 생긴다. 따라서 다중 법선 벡터를 이용한 옵셋 방법에서는 정해진 허용오차 이내로 STL 모델을 옵셋하였다.



(a) Smooth vertex has one normal vector



(b) Sharp vertex has more than two normal vectors



(c) Multiple normal vectors of a STL model

Fig. 2 Multiple normal vectors of a vertex

꼭지점의 옵셋 거리가 r 일 때의 옵셋 오차 δ_i 는 꼭지점의 법선 벡터와 면의 법선 벡터간의 각도에 의해 식(2)와 같이 계산된다.

$$\delta_i = r\left(1 - \frac{\mathbf{v}_{1,\ldots,n} \cdot \mathbf{f}_i}{|\mathbf{v}_{1,\ldots,n}||\mathbf{f}_i|}\right) \quad (2)$$

여기서 옵셋 반경은 일정하므로 옵셋 오차를 제어하기 위해서는 하나의 옵셋 벡터로 치환되는 다중법선 벡터와 면의 법선 벡터 사이의 각도를 제한해야 한다. 즉 한 꼭지점에서 차이가 가장 작은 두 법선 벡터를 선택해 하나의 벡터로 치환하는 과정을 식 (2)가 만족되는 동안 반복해 다중 법선 벡터를 계산한다.

Fig. 2 (c)는 STL 모델에서 위 알고리즘으로 계산된 다중 법선 벡터를 보여주고 있는데 하나의 면에서 생성된 꼭지점에는 하나의 법선 벡터가 존재하고 모서리와 꼭지점에서 생성된 꼭지점에는 두 개 이상의 법선 벡터가 존재함을 볼 수 있다.

3. STL 모델의 옵셋

3.1 면의 옵셋

STL 모델에서 꼭지점의 다중 법선 벡터를 계산한 뒤에 모든 꼭지점을 다중법선 벡터 방향으로 옵셋반경만큼 이동해 면을 옵셋한다. 한 곡면에서 생성된 꼭지점들은 단일 법선 벡터를 가지게 되며 이러한 꼭지점으로 구성된 모서리와 꼭지점에서는 옵셋 후에 공간이나 겹침이 발생하지 않는다. 이렇게 면을 옵셋하는데 소요되는 시간은 꼭지점의 개수 n 에 선형으로 비례하므로 시간 복잡도는 O(n)이다. 곡면과 곡면의 각진 경계에서 생성되는 꼭지점은 정밀한 옵셋을 위해 두 개 이상의 법선 벡터를 가지며 이러한 꼭지점으로 구성된 모서리와 꼭지점에서는 옵셋 후에 공간이나 겹침이 발생한다. 공간은 다음에 소개되는 모서리 옵셋과 꼭지점 옵셋 방법으로 채우게 되고 겹침은 가공 데이터 생성과정에서 제거하게 된다.

3.2 모서리의 옵셋

두 면을 연결하는 한 모서리는 양 끝의 두 꼭지점을 옵셋 반경만큼 이동해 옵셋한다. Fig. 3 (a)와 같이 각각 단일 법선 벡터 $\mathbf{v}_{1,2}$ 와 $\mathbf{v}_{2,3}$ 을 가지는 두 꼭지점으로 이루어진 부드러운 모서리는 두 꼭지점을 각각의 단일 법선 방향으로 이동해 옵셋한다. 두 면을 연결하는 모서리가 같은 방향으로 이동되므로 옵셋 후에도 두 면이 연결되어 있어 별도의 처리가 필요하지 않다. Fig. 3 (b)와 같이 각각 다중 법선벡터 $\mathbf{v}_1$, $\mathbf{v}_2$ 와 단일 법선 벡터 $\mathbf{v}_{3,4}$ 를 가지는 두 꼭지점으로 이루어진 모서리는 옵셋 이후에 잘려진 삼각 뿔 모양의 공간이 발생한다. 이 공간은 삼각 뿔 모양의 삼각망으로 연결할 필요가 있다. Fig. 3 (c), (d)와 같이 각각 다중 법선 벡터 $\mathbf{v}_1$, $\mathbf{v}_2$ 와 $\mathbf{v}_3$, $\mathbf{v}_4$ 를 가지는 두 꼭지점으로 이루어진 모서리는 두개의 분리된 모서리로 옵셋되어 한 모서리에서 만나는 두 면이 만나지 않게 된다. Fig. 3 (c)와 같이 볼록한 모서리는 공간이 생김으로 삼각망을 추가해 연결하며, Fig. 3 (d)와 같이 오목한 모서리는 겹침이 발생하므로 잘라내야할 필요가 있다. 3 차원에서 겹침을 제거하는 정규화는 긴 연산시간이 소요되므로 2 차원 공구경로 생성과정에서 제거하였다.

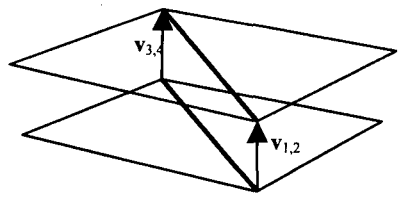
볼록하고 각진 모서리에서 발생하는 공간을 연결하는 삼각형 망은 곳 바로 결정할 수도 있고 Fig. 3 (e)와 같이 다중 법선벡터를 반복해서 나누어 계산할 수도 있다. 나눔 법선벡터 $\mathbf{v}_{12}$ 는 두 다중 법선벡터 $\mathbf{v}_1$ 와 $\mathbf{v}_2$ 를 반복해서 나누어 계산하며 평면들로 연결할 때 오차가 하용오차보다 작을 경우 나누는 과정을 정지하고 삼각형 면들로 연결하게 된다.

Blend sharp edge ($\mathbf{v}_{s1}$, $\mathbf{v}_{s2}$, $\mathbf{v}_{e1}$, $\mathbf{v}_{e2}$) {
  if (Error<Allowance)
    Insert face ($\mathbf{v}_{s1}$, $\mathbf{v}_{s2}$, $\mathbf{v}_{e1}$, $\mathbf{v}_{e2}$, edge $\mathbf{V}_s$, $\mathbf{V}_e$);
  else {

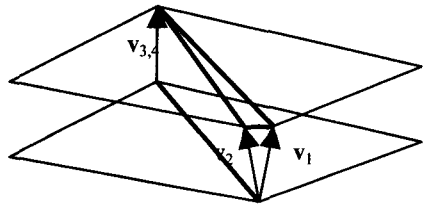
$$\mathbf{v}_{s12} = \frac{\mathbf{v}_{s1} + \mathbf{v}_{s2}}{|\mathbf{v}_{s1} + \mathbf{v}_{s2}|}$$

$$\mathbf{v}_{e12} = \frac{\mathbf{v}_{e1} + \mathbf{v}_{e2}}{|\mathbf{v}_{e1} + \mathbf{v}_{e2}|}$$

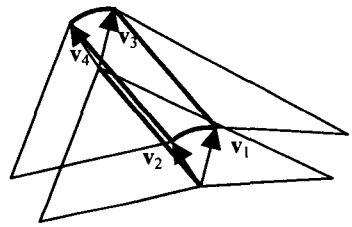
    Blend sharp edge ($\mathbf{v}_{s1}$, $\mathbf{v}_{s12}$, $\mathbf{v}_{e1}$, $\mathbf{v}_{e12}$);
    Blend sharp edge ($\mathbf{v}_{s12}$, $\mathbf{v}_{s2}$, $\mathbf{v}_{e12}$, $\mathbf{v}_{e2}$);
  }
}
$\mathbf{v}_s$ is the normal vector of a vertex at the start point
$\mathbf{v}_e$, the normal vector of a vertex at the end point



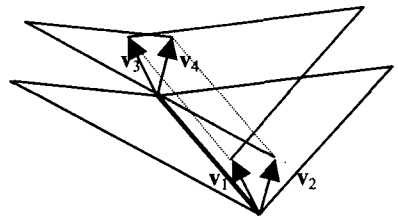
(a) Smooth edge



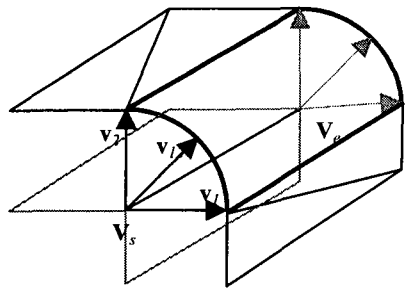
(b) The blend surface of three normal vectors



(c) The blend surface fills the gap at the convex edge



(d) The overlap at the sharp concave edge



(e) Blending mesh at sharp convex edge

Fig. 3 Offset edge using the multiple normal vectors of a vertex

3.3 꼭지점의 옵셋

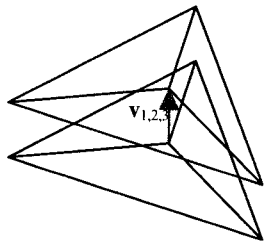
부드러운 면 위에서 생성된 꼭지점은 Fig. 4 (a)와 같이 단일 법선 벡터 $\mathbf{v}_{123}$를 가짐으로 옵셋될 때 같은 방향으로 이동되어 주위의 면들 사이에 공간이나 겹침이 생기지 않는다. 모서리에서 생성된 꼭지점은 Fig. 4 (b)와 같이 두개의 법선 벡터 $\mathbf{v}_{1,2}$와 $\mathbf{v}_{3,4}$를 가지며 옵셋될 때 모서리를 연결하는 삼각 망에 의해서 공간이 채워짐으로 따로 공간을 연결할 필요가 없다. 솔리드 모델의 꼭지점에서 생성된 꼭지점은 Fig. 4 (c) 와 같이 새 개 이상의 법선 벡터 $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$를 가지게 되며 옵셋 될 때 한 꼭지점이 세개 이상의 점으로 나뉘어 각각의 법선 벡터 방향으로 이동되므로 주위의 옵셋된 면들 사이에 공간이나 겹침이 발생한다.

꼭지점을 공유하는 모서리들 중에 세 개 이상이 볼록한 모서리 이면 옵셋 후에 공간이 생김으로 삼각형 망으로 연결해 주여야한다. 각진 꼭지점의 옵셋으로 발생하는 공간을 연결하기 위해서 Fig. 4 (d)와 같이 다중 법선 벡터를 반복해서 나누어 나눔 법선 벡터들을 만들고, 오차가 허용 값이내 일 때 나눔 법선 벡터들을 이용해 공간을 연결하는 삼각형 망을 생성한다.

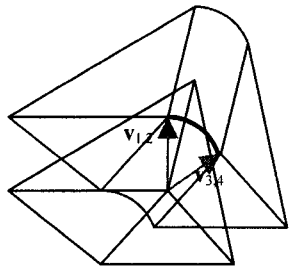
Blend sharp vertex (vector $\mathbf{v}_1$, vector $\mathbf{v}_2$, vector $\mathbf{v}_3$) {
  if (Error<Allowance)
    Insert face ($\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{V}$);
  else {
    $$\mathbf{v}_{12} = \frac{\mathbf{v}_1 + \mathbf{v}_2}{|\mathbf{v}_1 + \mathbf{v}_2|}$$
    $$\mathbf{v}_{23} = \frac{\mathbf{v}_2 + \mathbf{v}_3}{|\mathbf{v}_2 + \mathbf{v}_3|}$$
    $$\mathbf{v}_{31} = \frac{\mathbf{v}_3 + \mathbf{v}_1}{|\mathbf{v}_3 + \mathbf{v}_1|}$$
    Blend sharp vertex ($\mathbf{v}_1$, $\mathbf{v}_{12}$, $\mathbf{v}_{31}$);
    Blend sharp vertex ($\mathbf{v}_2$, $\mathbf{v}_{23}$, $\mathbf{v}_{12}$);
    Blend sharp vertex ($\mathbf{v}_3$, $\mathbf{v}_{31}$, $\mathbf{v}_{23}$);
    Blend sharp vertex ($\mathbf{v}_{12}$, $\mathbf{v}_{23}$, $\mathbf{v}_{31}$);
  }
}

$\mathbf{v}_i$ is the multiple normal vector of a vertex

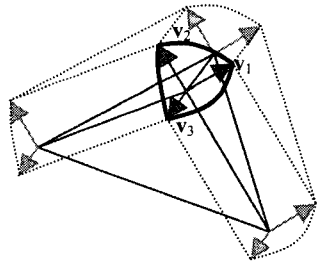
$\mathbf{v}_{ij}$, the dividing vector of the normal vectors, $\mathbf{v}_i$, $\mathbf{v}_j$



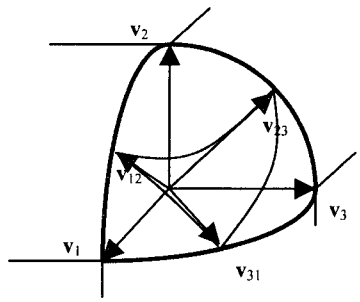
(a) A smooth vertex with a single normal vector



(b) A vertex with two normal vectors



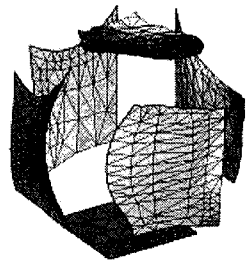
(c) The blend surface fills a gap at the convex vertex



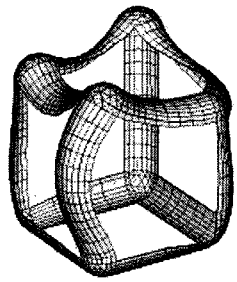
(d) Blending mesh at sharp convex vertex

Fig. 4 Offset vertex using the multiple normal vectors of a vertex

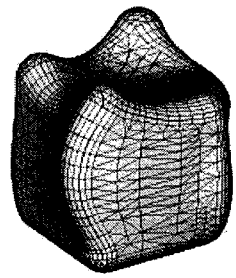
Fig. 5 는 다중 법선 벡터를 이용해 STL 파일을 옵셋하는 과정과 결과를 보여준다. Fig. 5 (a)는 모든 꼭지점들을 다중 법선 벡터 방향으로 옵셋 반경 만큼 이동하는 방법으로 면들을 옵셋한 결과이다. 부드러운 면상의 꼭지점들은 같은 방향으로 이동되어 공간이나 겹침이 발생하지 않았으며 각이 큰 모서리와 꼭지점에서는 큰 공간이 발생한다. 각진 모서리와 꼭지점에서 발생하는 공간은 Fig. 5 (b)와 같이 연결하는 삼각형 망을 추가함으로써 Fig. 5 (c)와 같은 옵셋 결과를 얻는다.



(a) Face offset



(b) Sharp edge and vertex offset



(c) STL model offset result

Fig. 5 STL model offset using the multiple normal vectors of a vertex

3.4 경로 생성과 겹침제거

옵셋된 모델은 가공 단계에서 공구의 기준 점이 지나는 CL 면에 해당한다. 따라서 공구경로는 CL 면을 연속되는 기준 평면으로 나누어 생성하였다 [4]. 모델의 오목하고 각진 모서리와 꼭지점에서는 옵셋 후에 겹침이 발생하므로 이를 제거해야 한다. 하지만 삼차원 공간상에서 겹친 면들을 찾고 잘라내는 정규화 작업은 복잡하고 시간이 많이 소요되는 작업이다. 다른 방법으로 모델에서 공구경로에 해당하는 이차원 직선들을 만들고 여기서 불필요한 선들을 제거하는 방법이 있다. 쾌속 조형이나 NC 가공을 위해서는 이 방법이 간편하고 빠르기 때문에 공구경로 생성 단계에서 불필요한 경로를 제거하였다.

4. 비교 실험

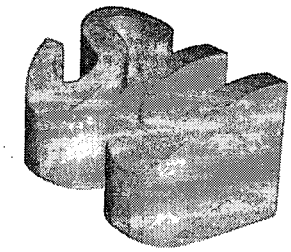
기존의 옵셋 방법인 면 옵셋 방법과 꼭지점 옵셋 방법 그리고 본 논문에서 새로 제안된 꼭지점의 다중 법선 벡터를 이용한 옵셋 방법을 OpenGL 라이브러리와 C++언어로 구현하였다. Fig. 6 (a)는 솔리드 모델에서 생성된 STL 파일을 개발된 시스템에서 읽은 것 이다.

Fig. 6 (b)는 기존의 면 옵셋 방법으로 STL 파일을 옵셋하는 과정으로 면의 법선벡터들, 옵셋된 면들 그리고 모서리와 꼭지점에서 생긴 공간을 삼각망으로 연결한 결과를 보여주고 있다. 첫 번째 그림에서 삼각형 면들을 음영처리 해 보여주고 있고 꼭지점에 위치한 면의 법선 벡터들을 검은 직선으로 보여주고 있다. 두 번째 그림에서는 모든 면들을 면의 법선 방향으로 옵셋 반경만큼 이동한 결과 이다. 이 옵셋 방법은 면과 면 사이의 거리가 정확하게 옵셋 거리만큼 이동되므로 정밀도가 아주 우수하다. 그러나 평면을 제외한 모든 모서리와 꼭지점에서 작은 공간들과 겹침이 발생하는 문제점이 있다. 세번째 그림은 모서리에서 발생한 공간은 잘려진 실린더 모양의 삼각망으로 꼭지점에서 발생한 공간은 잘려진 구 모양의 삼각망으로 연결한 결과로 아주 작은 삼각형 면들이 모서리와 꼭지점에 추가되기 때문에 삼각형 면의 크기가 균일하지 못하고 데이터 크기가 많이 증가하게 된다.

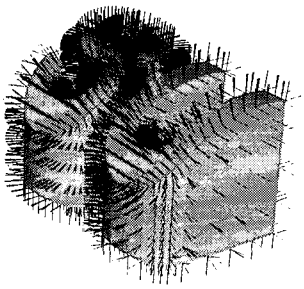
Fig. 6 (c)는 기존의 꼭지점 옵셋 방법으로 STL 파일을 옵셋하는 과정으로 꼭지점의 법선 벡터들과 옵셋된 결과를 보여준다. 첫 번째 그림의 꼭지점들에 검은 직선으로 표현된 법선 벡터 들은 꼭지점을 둘러 싸는 면들의 법선벡터들을 평균해 계산한 것이다. 두 번째 그림은 모든 꼭지점들을 꼭지점의 법선 벡터 방향으로 이동해 옵셋한 결과이다. 공간과 겹침은 발생하지 않았지만 각진 모서리와 꼭지점에서 옵셋 오차가 아주 커서 가공 데이터를 생성하는데 이용하기에는 부적합하다.

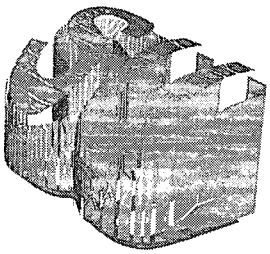
Fig. 6 (d)는 논문에서 새로 제안된 꼭지점의 다중 법선벡터를 이용한 옵셋 과정과 결과를 보여준다. 첫번째 그림은 꼭지점을 둘러싸는 면들의 법선 벡터들로부터 계산된 다중 법선벡터를 검은색 직선으로 보여주고 있다. 솔리드 모델의 곡면이었던 부분에서 생성된 꼭지점은 하나의 법선 벡터를 가지며 각진 모서리와 꼭지점에서 생성된 꼭지점은 여러 개의 법선 벡터를 가지게 된다. 두번째 그림은 모든 꼭지점을 다중 법선벡터 방향으로 이동해 옵셋한 결과를 보여준다. 부드러운 모서리와 꼭지점에서는 공간이나 겹침이 발생하지 않았으며 삼각형의 모양과 크기가 균일함을 볼 수 있다. 꼭지점의 법선 벡터를 계산함에 있어 면과 면사이의 각이 작은 경우만 하나의 벡터로 치환했으므로 옵셋 오차는 허용오차 이하로 제어된다. 각진 모서리와 꼭지점은 연결된 면의 법선과 비슷한 방향의 다중 법선벡터 방향으로 분리되어 옵셋되므로 삼각형 면이 찌그러지지 않으며 옵셋 오차가 허용오차 이내로 제어된다. 이 곳에서 발생되는 공간은 세번째 그림과 같이 추가적인 삼각형 망으로 부드럽게 연결되므로 결과적으로 생성될 가공 데이터가 정밀하고 부드럽게 된다. 오목하고 각진 모서리와 꼭지점의 겹침을 경로 생성 단계에서 제거된다. 마지막 그림은 옵셋된 면을 연속적인 수평 면으로 나누어 생성된 가공데이터 이다.

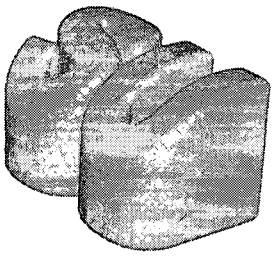
세가지 옵셋 방법의 비교 결과를 Table 1 에서 보여주고 있다. 꼭지점의 다중 법선 벡터를 이용한 옵셋 방법은 기존의 면 옵셋 방법과 꼭지점 옵셋 방법의 단점을 보완하고 각각의 장점을 살린 새로운 방법으로 옵셋 오차를 지정하는 값 이내로 제한할 수 있으며 삼각형 면의 크기가 균일하고 데이터의 크기가 작게 증가하는 장점이 있다. 따라서 솔리드에서 생성된 STL 모델에서 가공데이터를 생성 할 때 적합한 옵셋 방법으로 사료된다.



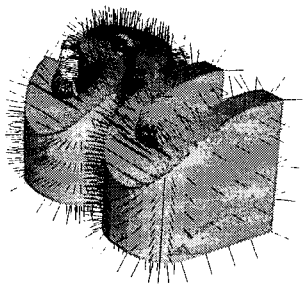
(a) STL file of a solid model

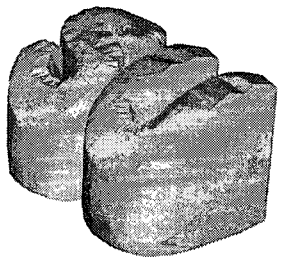




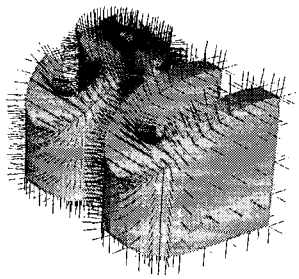


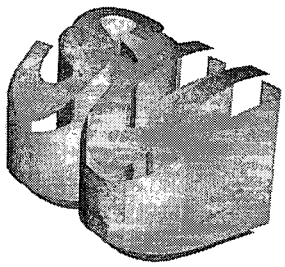
(b) Offsetting surfaces along face normal vectors, 12,600 gaps

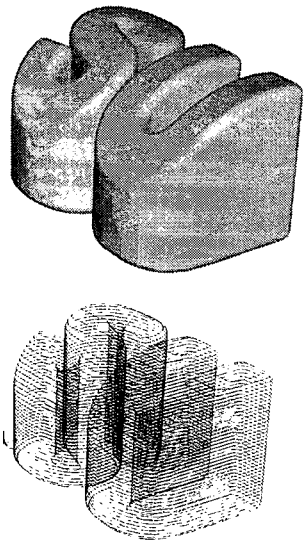




(c) Offsetting vertices along the single normal vectors of a vertex, 3.6mm error







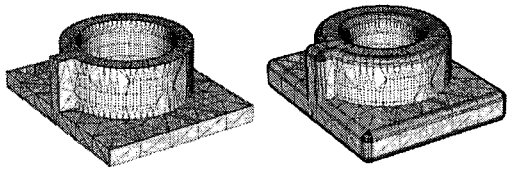
(d) Offsetting vertices along the multiple normal vectors of a vertex, 1,676 gaps

Fig. 6 Comparative examples of offset methods

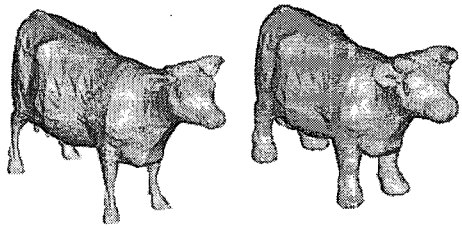
Table 1 The comparison of offset methods

	Face normal	Vertex normal	Multiple normal
Complexity	O(n)+bO(n) [0.375s]	O(n) [0.031s]	O(n)+bO(s) [0.094s]
Number of faces	Increase [48,842]	Same [4,200]	Increase [13,923]
Number of gaps	All edges [12,600]	No [0]	Sharp E. [1,676]
Uniformity	Poor	Good	Good
Precision	Exact [0mm]	Poor [3.5mm]	Allowable [0.1mm]
Tool path	Good	Poor	Better

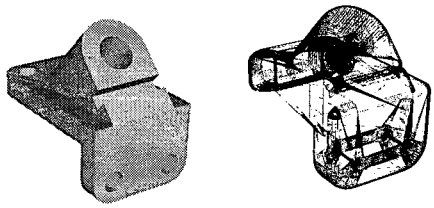
STL model with 4,200 faces, 0.1mm tolerance
n is the count of faces
b is the time complexity of blending
s is the count of sharp edges & vertices (s<<n)
[ ] is the result of the example, shown in Fig. 6



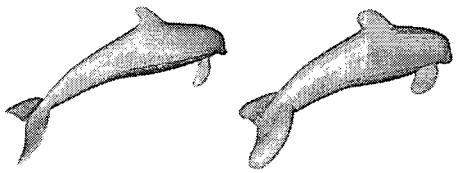
(a) STL model and offset mesh of cylinder



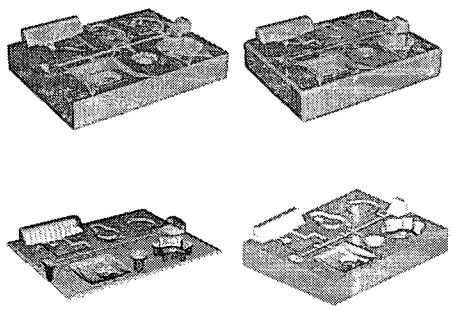
(b) Cow



(c) Anchor



(d) Dolphin



(e) STL model, offset mesh, tool path and simulation of injection mold

Fig. 7 The offset results of various triangular meshes

솔리드를 이루는 일반적인 STL 모델이 제안된 방법으로 옵셋 되는지 확인하기 위해서 다양한 CAD 시스템에서 생성된 STL 모델과 인터넷에서 받은 STL 파일을 Fig. 7 과 같이 옵셋해 보았다. 또한 Fig. 7 (e)는 사출 금형 업체에서 디자인한 금형을 STL 파일로 출력한 것을 본 시스템에서 옵셋하고 NC 코드를 생성해 검증한 것으로 원래의 STL 모델과 모의 가공된 결과가 잘 일치함을 볼 수 있다.

5. 결론

논문에서는 꼭지점의 다중 법선벡터를 이용해 STL 파일을 옵셋하는 방법을 제안하고 구현하였다. 꼭지점을 둘러싸는 면들의 법선 벡터를 그 꼭지점의 법선 벡터들로 초기화하고 가장 비슷한 두 벡터를 하나로 치환하는 작업을 허용오차 이내에서 반복해 옵셋을 위한 꼭지점의 다중 법선 벡터를 얻는다. 다음으로 모든 꼭지점을 다중 법선 벡터 방향으로 옵셋 반경만큼 이동하고 각진 모서리와 꼭지점에서 발생하는 공간은 삼각형 망을 추가해 연결한다. 제안된 옵셋방법은 부드러운 모서리와 꼭지점에 공간과 겹침이 발생하지 않아 삼각형의 크기가 균일하며 데이터 크기와 연산시간이 작다. 또한 각진 모서리와 꼭지점은 면의 법선 방향으로 옵셋되고 여기서 생기는 공간은 새로운 삼각형 망으로 연결됨으로 정밀한 옵셋이 가능하다. 제안된 방법은 부드러운 곡면과 각진 모서리를 동시에 가지는 솔리드 모델에서 생성된 STL 파일에서 쾌속조형과 NC 가공을 위한 데이터를 생성에 효과적으로 이용될 수 있다. 하지만 제안된 옵셋 방법은 꼭지점의 다중 법선 벡터와 공간을 채우는 삼각망의 계산에서 면, 모서리, 꼭지점의 위상관계를 이용함으로 완전한 솔리드를 이루는 STL 모델에만 적용이 가능하기 때문에 향후에는 곡면 모델에서 생성된 STL 모델의 옵셋 방법에 대한 연구가 요구된다.

참고문헌

1. Choi, B. K., Kim, D. H. and Jerad, R. B., "C-Space approach to tool path generation for die and mold machining," Computer-Aided Design, Vol. 29, No. 3, pp. 657-669, 1997.
2. Cho, S. H., Cho M. W. and Kim, J. D., "Reverse Engineering of 3D Compound Shapes using Delaunay Triangulation," Journal of the Korean Society of Precision Engineering, Vol. 17, No. 7, pp. 181-188, 2000.
3. Lee, S. H., Ahn, D. G. and Yang, D. Y., "Surface Recosntruction for Cutting Path Generation on VLM-Slicer," Journal of the Korean Society of Precision Engineering, Vol. 19, No. 7, pp. 71-79, 2002.
4. Jun, C. S., Kim, D. S. and Park, S. H., "A new curve-based approach to triangle machining," Computer-Aided Design, Vol. 34, No. 5, pp. 379-389, 2002.
5. Koc, B. and Lee, Y. S., "Non-uniform offsetting and hollowing objects by using biarcs fitting for rapid prototyping process," Computers in Industry, Vol. 47, pp. 1-23, 2002.
6. Qu, X. and Stucker, B., "A 3D surface offset method for STL-format models," Rapid prototyping journal, Vol. 9, No. 3, pp. 133-141, 2003.