

# 거리장을 이용한 삼각망의 오프셋팅

유동진<sup>#</sup>

## Offsetting of Triangular Net using Distance Fields

Dong-Jin Yoo<sup>#</sup>

### ABSTRACT

A new method which uses distance fields scheme and marching cube algorithm is proposed in order to get an accurate offset model of arbitrary shapes composed of triangular net. In the method, the space bounding the triangular net is divided into smaller cells. For the efficient calculation of distance fields, valid cells which will generate a portion of offset model are selected previously by the suggested detection algorithm. These valid cells are divided again into much smaller voxels which assure required accuracy. At each voxel distance fields are created by calculating the minimum distances between corner points of voxels and triangular net. After generating the whole distance fields, the offset surface were constructed by using the conventional marching cube algorithm together with mesh smoothing scheme. The effectiveness and validity of this new offset method was demonstrated by performing numerical experiments for the various types of triangular net.

**Key Words :** Triangular net (삼각망), Offsetting (오프셋팅), Distance fields (거리장)

### 기호설명

$r_1$  = Position vector of a point to be projected  
 $r$  = Position vector of the projected point  
 $P_1, P_2, P_3$  = Position vectors of three vertexes  
 $U$  = Unit vector perpendicular to a triangle  
 $P$  = Perpendicular distance between the origin and a triangle  
 $a$  = Line vector from  $P_1$  to  $P_2$   
 $b$  = Line vector from  $P_2$  to  $P_3$   
 $c$  = Line vector from  $P_3$  to  $P_1$   
 $r_a, r_b$  = Start and end point of a line  
 $distance_{max}$  = Maximum distance in a cell

$distance_{min}$  = Minimum distance in a cell  
 $d_i$  = Distance value at i-th corner point of a voxel  
 $\xi_i, \eta_i, \zeta_i$  = Coordinates of i-th corner point of a voxel in the  $(\xi, \eta, \zeta)$  coordinate system  
 $n_x$  = X component of normal vector of a offset triangle  
 $n_y$  = Y component of normal vector of a offset triangle  
 $n_z$  = Z component of normal vector of a offset triangle  
 $d(x_r, y, z)$  = Distance value at right face of a voxel  
 $d(x_l, y, z)$  = Distance value at left face of a voxel  
 $d(x, y_u, z)$  = Distance value at upper face of a voxel  
 $d(x, y_d, z)$  = Distance value at lower face of a voxel  
 $d(x, y, z_f)$  = Distance value at front face of a voxel  
 $d(x, y, z_b)$  = Distance value at back face of a voxel

☞ 접수일: 2007 년 4 월 12 일; 게재승인일: 2007 년 7 월 1 일

# 교신저자: 대전대학교 컴퓨터응용 기계설계공학과

E-mail: djyoo@daejin.ac.kr Tel. (031) 539-2031

## 1. 서론

삼각망으로부터 육셋 곡면을 생성하는 문제는 컴퓨터원용가공(computer aided manufacturing), 컴퓨터원용설계(computer aided design), 폐속조형(rapid prototyping) 등의 분야에 있어서 주요한 연구대상이 되어 왔다. 정밀하고 강건한 삼각망의 육셋 알고리즘은 삼각망 모델기반의 CAM 시스템에서 간섭이 없는 공구경로를 생성하기 위하여 원래의 모델을 공구반경만큼 육셋 보정한 CL(cutter location) 면을 생성하거나 STL 모델기반의 폐속조형시스템에서 내부가 비어있고 일정한 두께를 가진 셸(shell)형태의 모델이나 후처리를 감안한 육셋모델 등을 생성하는데 적용될 수 있다. 최근에 STL 파일과 같은 삼각망의 정확한 육셋을 위하여 각 꼭지점에서 만나는 삼각형들의 법선벡터에 가중치를 부여하여 꼭지점의 육셋벡터를 계산하는 방법<sup>1</sup>이 제안되었다. 이 방법에서는 한 꼭지점과 만나는 삼각형이 세 개이면 정확하게 육셋되지만 네 개 이상이면 특수한 경우를 제외하고는 각 삼각형이 정확하게 육셋되는 꼭지점의 육셋벡터가 존재하지 않게 되는 단점이 있다. 또한 NC 공구경로 생성시 필요한 날카로운 모서리 부분에서의 라운딩(rounding)처리가 고려되어 있지 않기 때문에 CAM을 위한 육셋모델을 계산하는데는 문제가 있다. 이를 개선하기 위하여 각 꼭지점이 여러 개의 법선벡터를 갖도록 하는 꼭지점의 다중 법선벡터에 의한 솔리드 STL 모델의 육셋방법<sup>2</sup>이 국내 연구가들에 의하여 제안되어 다양한 형상에 성공적으로 적용된 바 있다. 그러나 이 방법에서는 육셋후에 발생하는 공간은 모서리육셋과 꼭지점육셋방법으로 적절히 처리하였으나 삼각형들간의 겹침은 가공데이터 생성과정에서 제거하는 방법을 사용했기 때문에 전용 후처리 프로그램이 없는 일반적인 사용자들이 접근하는 데는 한계가 있는 단점이 있다. 육셋 후에 발생하는 삼각망들의 겹침을 적극적으로 제거하기 위한 방법<sup>3</sup>이 다른 국내 연구가들에 의해 연구 보고 되었는데 이 방법에서는 육셋된 모든 삼각형들간의 교차 테스트를 통해서 모든 교차선을 구한 다음 교차된 삼각형들을 절단하고 새롭게 삼각망을 구성한 후 정규 삼각망을 제안한 유효영역탐색방법에 의하여 추출하였다. 이 방법에서는 겹침을 효율적으로 제거하기 위한 여러 알고리즘들이 제안되었으나 육셋후의 공간발생

문제 등에 대해서는 고려하지 않았고 특히 초기 모델의 삼각망의 질이 좋지 않으면 육셋후 교차부분에 날카로운 형상의 삼각형들이 많이 발생하기 때문에 사용하고자 하는 용도에 따라 적합한 후처리가 수반되어야 한다. 또한 점, 면이 접하는 경우 등의 특이 상황을 고려한 교차선 계산 및 강건한 유효영역탐색방법의 구현이 쉽지 않다는 단점이 있다. 저자의 선행연구결과를 통하여 보고된 음함수곡면기법에 의한 삼각망의 육셋방법<sup>8</sup>도 비교적 정밀한 육셋곡면을 얻을 수 있는 좋은 방법이기도 하나 날카로운 모서리로 인한 공간발생문제나 삼각망들의 겹침문제를 해결하기에는 문제점이 있으며 또한 정확한 육셋거리를 조절하기 위한 구속조건 처리가 쉽지 않다는 단점이 있다. 본 연구에서는 지금까지의 연구과정에서 언급 되었던 육셋후의 공간 발생 및 겹침 현상을 근본적으로 해소하고 다양한 형상의 3 차원 삼각망모델에 일률적으로 적용될 수 있는 거리장(distance fields)에 의한 새로운 삼각망 육셋방법을 제안하고자 한다. 거리장<sup>4,7</sup>은 CAD 및 CG 분야에서 3 차원 형상의 복셀화(voxelization), 솔리드모델의 불리언연산, 형상간의 모핑(morphing)등에 사용되던 개념으로서 형상을 둘러싸고 있는 공간상의 모든 점에서 형상과의 최소거리를 구해 스칼라장(scalar field) 형태로 정의해 놓은 것이다. 본 논문에서는 삼각망을 둘러싸고 있는 3 차원 공간상에 거리장을 정의하고 생성된 거리장으로부터 마칭큐브알고리즘(marching cube algorithm)을 이용하여 정확하고 공간 및 간섭이 완벽하게 처리된 육셋곡면을 추출해 내는 새로운 삼각망 육셋방법을 제안하고 다양한 모델에 대한 실험을 통하여 실제 응용 가능성을 검증하고자 한다.

## 2. 삼각망으로부터 거리장의 생성

삼각망으로부터 거리장을 생성한다는 것은 모델을 에워싸고 있는 3 차원 공간을 그리드(grid)형태로 분할한 후 그리드의 각 절점에서 형상모델까지의 최소거리를 계산한다는 것을 의미한다. 설명을 단순화하기 위하여 거리장의 2 차원적 개념을 Fig. 1 에 나타내었다. Fig. 1 에서 보는 바와 같이 형상의 바깥 절점에서의 거리는 음의 값을 내부 절점에서의 거리는 양의 값을 갖게 함으로써 모델의 외부 및 내부를 구별할 수 있다. 아울러 각 절

점에서의 스칼라 값은 그 점으로부터 형상까지의 최소거리를 의미하므로 원하는 읍셋거리가 지정되면 적절한 방법에 의하여 읍셋곡선을 추출할 수 있다. 형상의 복잡성에 따라 그리드의 간격을 조절함으로써 읍셋곡선의 정확도를 변화시킬 수도 있다. 본 연구에서는 읍셋하고자 하는 모델이 삼각망으로 구성되어 있을 경우를 생각하는 것이므로 각 절점에서 한 삼각형까지의 최소 거리를 구하기 위하여 Fig. 2 와 같은 알고리즘을 사용하였다.

Fig. 2 에서 보는 바와 같이 그리드의 각 절점

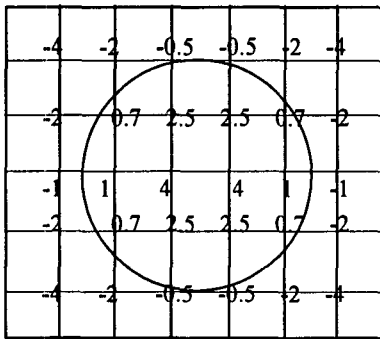


Fig. 1 Schematic diagram illustrating the concept of distance fields

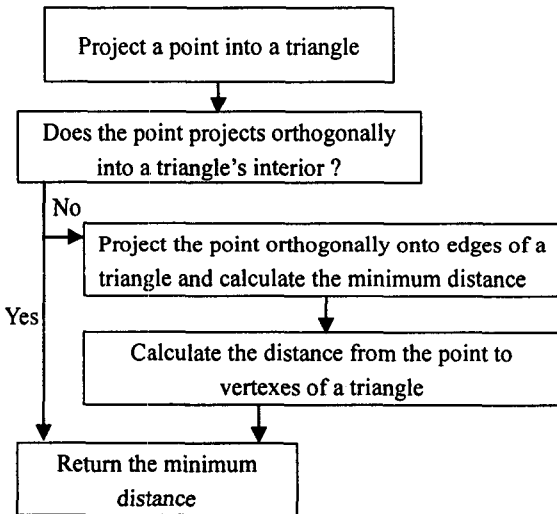


Fig. 2 Flow chart showing the procedures for the calculation of minimum distance from a point to a triangle

에서의 수직투영점이 삼각형의 내부에 존재하는 경우에는 절점과 투영점과의 거리가 최소거리가 되고 수직 투영점이 삼각형의 외부에 있는 경우에는 삼각형의 각 모서리와와의 수직거리 및 삼각형의 세 꼭지점과의 거리를 비교하여 최소가 되는 거리를 구하게 된다. 즉 그리드의 각 절점과 삼각망과의 최소거리를 구하기 위해서는 Fig. 3 에서 보는 바와 같은 세가지 경우를 함께 고려해야 한다.

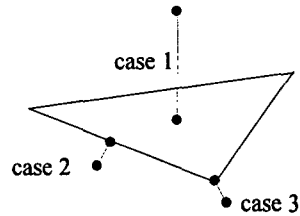


Fig. 3 Three cases for the calculation of minimum distance from a point to a triangle

한 점  $r_1$  과 삼각형과의 수직투영점  $r$  은 식 (1)과 같은 벡터방정식으로부터 구할 수 있다.

$$r = r_1 + (p - r_1 \cdot u)u \quad (1)$$

식 (1)에서  $u$  는 삼각형에 수직인 단위벡터를  $p$  는 삼각형과 좌표계 원점과의 수직거리를 의미한다. 수직투영점이 현재의 삼각형 내부에 포함되는지의 여부는 Fig. 4 에 나타나 있는 벡터들 간의 다음과 같은 판별식을 이용하여 판단할 수 있다.

$$(a \times d) \cdot u \geq 0 \quad (2)$$

$$(b \times e) \cdot u \geq 0 \quad (3)$$

$$(c \times f) \cdot u \geq 0 \quad (4)$$

모든 판별식이 만족되면 수직투영점  $r$  은 현재의 삼각형 안에 있다는 것을 뜻한다. 한 점과 직선과의 수직투영점은 식 (5)와 식 (6)과 같은 벡터방정식으로부터 구할 수 있다.

$$\lambda = n \cdot \frac{r_b - r_a}{|r_b - r_a|} \quad (5)$$

$$r = r_a + \lambda \frac{r_b - r_a}{|r_b - r_a|} \quad (6)$$

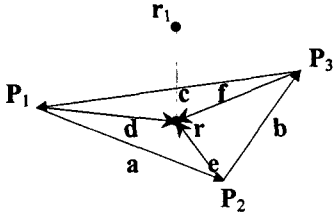


Fig. 4 Perpendicular projection of a point onto a triangle

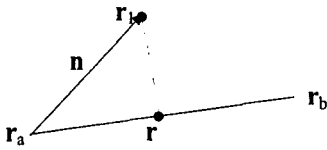


Fig. 5 Perpendicular projection of a point onto an edge

$\lambda$  가 0 과  $|r_b - r_a|$  사이에 있으면 투영점이 직 선상에 존재한다는 것을 의미한다. 한 점과 삼각 형들의 각 꼭지점들과의 최소거리는 점들과의 거 리를 구해 최소가 되는 거리를 선택하면 된다. 사 실 식 (1)에서 식 (6)까지의 투영점 계산과정을 프 로그램화 하는 것은 어렵지 않으나 거리장을 구성 하기 위한 수많은 점들과 수 십만, 수 백만 개의 삼각형들과의 수직 투영점을 일일이 계산한다는 것은 현재의 컴퓨터 성능을 감안해 보더라도 그리 간단치 않은 일이다. 최소거리계산 알고리즘이 효 율적이지 못할 경우에는 수 시간 내지는 수 십 시 간이 소요될 지도 모르는 막대한 계산량을 요하는 문제가 될 수도 있다. 본 연구에서는 그리드 상의 각 절점에서 절점과 형상까지의 최소거리를 구하 기 위하여 Fig. 6 에서 보는 바와 같이 삼각망 모 델을 포함하는 3 차원 공간을 특정한 크기의 셀 (cell)로 분할하여 처리하는 방법을 사용하였다. 본 연구의 주요 목적이 원래의 모델에 대한 읍셋 모 델을 구하는 것이므로 읍셋거리와 각 셀의 크기를 반영한 식 (7)과 식 (8)과 같은 판별식을 이용하여 사전에 읍셋곡면의 일부분이 될 가능성이 없는 셀 은 고려대상에서 배제 시킴으로써 계산시간을 단 축시켰다.

$$l_A \geq (\text{offset\_distance} + \frac{\sqrt{3}}{2} \text{cell\_size}) : \text{cell A is skipped} \quad (7)$$

$$l_B \leq (\text{offset\_distance} + \frac{\sqrt{3}}{2} \text{cell\_size}) : \text{cell B is selected} \quad (8)$$

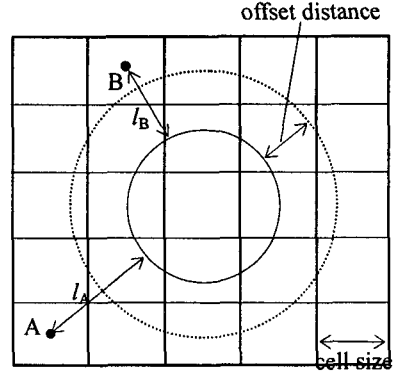


Fig. 6 Cell strategy for the efficient calculation of distance fields

각 셀의 중심과 삼각망과의 최소거리를 구해 이를 식 (7) 및 식 (8)과 비교해 보면 특정 셀이 읍 셋곡면의 일부분이 될 수 있을 지의 여부를 판단 할 수 있다. 일단 읍셋곡면이 통과할 것으로 예상 되는 셀들이 선택된 후에는 Fig. 7 에서 보는 바와 같이 각 셀을 요구되는 허용오차를 만족할 수 있 을 정도의 크기로 복셀화 한 후 복셀들의 각 절점 에서 삼각망과의 최소거리를 구하게 된다.

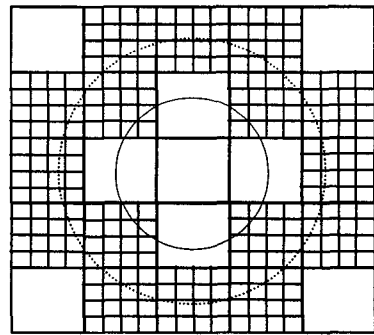


Fig. 7 Voxelization of selected cells

그러나 식 (7)과 식 (8)로 표현된 판별식은 읍 셋곡면이 통과할 셀을 정확히 판단해 주기는 하나 그렇지 않은 셀들도 상당수 가능한 것으로 오판할 여지가 있다. 이는 셀 중심과의 최소 수직거리만 생각했기 때문인데 본 연구에서는 일차적으로 선 택된 셀들에 대해서 Fig. 8 에서 보는 바와 같이 각 셀의 27 개의 절점에 대한 형상과의 최소거리 를 구하고 식 (9)와 식 (10)과 같은 판별식을 이용

하여 판단의 정확도를 향상시켰다.

$$distance_{max} \leq offset\_distance : \text{current cell is skipped} \quad (9)$$

$$distance_{min} \geq offset\_distance : \text{current cell is skipped} \quad (10)$$

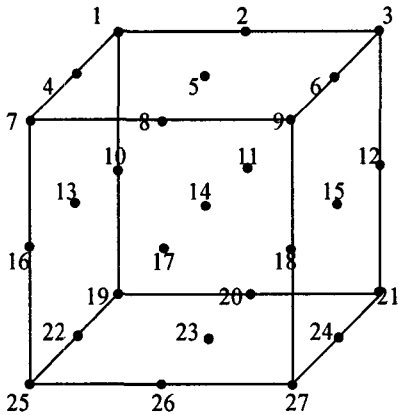


Fig. 8 Distance calculation at 27 points of a cell

최종적으로 읍셋곡면이 될 셀이 선택되면 Fig. 7 에서 보는 바와 같이 각 셀을 복셀화 한 후 복셀들의 각 절점에서 형상과의 최소거리를 구하게 된다. 한 셀에 있어서도 수 만, 수 십만 개의 절점에 대한 최소거리를 구할 수도 있으므로 원래의 모델을 구성하고 있는 모든 삼각형에 대하여 거리장을 계산할 경우 계산량이 크게 증가할 소지가 있다. 따라서 본 연구에서는 각 셀에 삼각망 모델 중 최소거리가 될 가능성이 있는 삼각형들만 클러스터링(clustering)하여 사전에 분배함으로써 거리장 계산의 효율성 향상을 도모하였다. 아울러 현재의 절점에 대하여 클러스터링된 삼각형들을 거리가 작은 순서대로 sorting 한 후 실제 최소거리를 발생시킬 최소한의 삼각형들에 대해서만 최소거리 계산을 수행함으로써 전체 거리장 계산에 소요되는 시간을 실제적인 적용이 가능한 수준으로 단축시켰다.

### 3. 거리장을 이용한 읍셋곡면의 추출

앞 절에서와 같이 셀들에 분할된 각 복셀들의 절점에 거리장이 정의되면 원하는 읍셋거리에 대한 읍셋곡면을 추출할 수 있다. 본 연구에서는 읍

함수 곡면의 가시화방법으로 널리 사용되고 있는 마칭큐브알고리즘(marching cube algorithm)과 스무딩(smoothing)을 통하여 양질의 삼각망을 재생성함으로써 여러 용도로 사용될 수 있도록 하였다. 마칭큐브알고리즘은 Fig. 9 에서 보는 바와 같이 거리장이 정의되어 있는 공간을 작은 정육면체로 나눈 후 각 구석점에서의 거리값의 최소치와 최대치를 구하여 읍셋거리가 이 범위를 벗어나면 그냥 지나가고 이 사이에 존재하는 경우에는 적절한 방법에 의하여 삼각망을 구성하는 방법이다.<sup>9</sup> Fig. 9 는 설명을 단순화 하기 위하여 간략히 나타낸 것으로 실제 모델에 대한 읍셋곡면 생성시에는 수 천, 수 만개의 셀과 각 셀에 대해서도 수 천, 수 만개의 복셀들을 생성하게 되므로 매우 정밀한 읍셋곡면을 추출할 수 있다.

offset surface(offset distance=1.5)

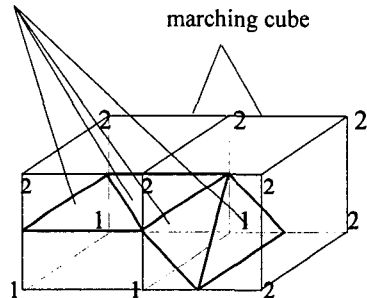


Fig. 9 Schematic diagram illustrating the concept of marching cube algorithm

정의되어 있는 거리장과 다른 크기의 마칭큐브로 가시화 하는 경우에는 식 (11) 및 식 (12)와 같은 FEM(Finite Element Method) 에서 많이 사용되는 육면체 요소의 보간함수를 이용하여 복셀 내부의 임의의 점에서의 거리값을 구하였다. 임의의 점의 절대좌표 값은  $\xi, \eta, \zeta$  좌표로 표시될 수 있고 각 구석점의  $\xi_i, \eta_i, \zeta_i$  는 Fig. 10 과 같이 정의되므로 보간함수  $N_i$  를 식 (12)로부터 구할 수 있다. 임의의 점에서의 거리값  $d$  는 식 (11)과 같이 8 개의 구석점들에서의 거리값  $d_i$  를 보간 하면 된다.

$$d = N_1 d_1 + N_2 d_2 + \dots + N_8 d_8 \quad (11)$$

$$N_i = \frac{1}{8} (1 + \xi_i \xi) (1 + \eta_i \eta) (1 + \zeta_i \zeta) \quad (12)$$

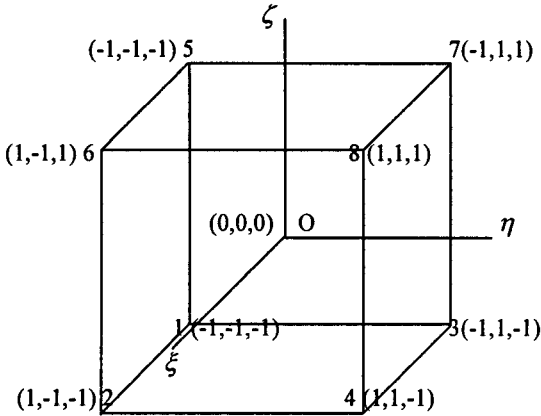


Fig. 10 Coordinate system for the tri-linear interpolation

이와 같이 생성된 복셀내의 삼각형들은 치수의 정확도는 물론이고 올바른 법선방향을 갖고 있어야 한다. 원래의 삼각망모델로부터 현재의 옹셋 삼각형의 법선방향을 계산할 수는 있으나 이 방법은 옹셋거리가 클 경우 부정확한 방향을 결과로 줄 수도 있고 또한 계산시간도 많이 걸리는 단점이 있다. 본 논문에서는 보다 정확하고 빠르게 복셀내의 임의의 점에서의 법선벡터를 구하기 위하여 현재의 점을 복셀의 전, 후, 좌, 우, 상, 하, 여섯 면에 투영해서 각 점에서의 거리값으로부터 현재 점에서의 법선벡터를 추정하는 방법을 사용하였다.

$$n_x = \frac{d(x_i, y, z) - d(x_j, y, z)}{|d(x_i, y, z) - d(x_j, y, z)|} \quad (13)$$

$$n_y = \frac{d(x, y_u, z) - d(x, y_d, z)}{|d(x, y_u, z) - d(x, y_d, z)|} \quad (14)$$

$$n_z = \frac{d(x, y, z_f) - d(x, y, z_b)}{|d(x, y, z_f) - d(x, y, z_b)|} \quad (15)$$

#### 4. 적용 및 결과

Figs. 11~13 은 초기 삼각망모델과 본 논문에서 제안한 거리장에 의한 방법을 이용하여 옹셋 곡면들을 생성한 예들을 보여 주고 있다. 결과에서 볼 수 있는 바와 같이 종래의 공간발생문제나 삼각형

들간의 간섭문제가 거리장으로부터 직접 옹셋 곡면을 추출해 내는 과정에서 자연스럽게 해결되고 있음을 확인할 수 있었다. 즉 공간을 적절히 채우기 위한 별도의 메쉬(mesh)생성 프로그램이나 옹셋 삼각형들간의 간섭을 제거하기 위한 교차선 계산 및 삼각형 재생성 프로그램 등이 전혀 필요 없음을 알 수 있다.

Fig. 14 는 사람의 전신모델에 대한 적용 예로서 61 만여 개의 삼각형으로 구성된 대형 모델이고 형상도 복잡하여 STL 상용 편집기를 이용해도 옹셋 곡면을 구하기가 매우 어려운 모델이다. 결과에서 보는 바와 같이 옹셋거리가 양 또는 음일 경우 모두 성공적으로 옹셋 곡면이 생성되었으며 특정부분의 확대된 그림에서 볼 수 있는 바와 같이 마칭큐브알고리즘에 의하여 삼각망을 구성하고 유연화 과정을 통해서 삼각망의 질을 개선함으로써 CAM, RP, CAE, CAD 등 많은 응용분야에 적용될 수 있는 양질의 삼각망을 생성할 수 있었다.

본 연구에서 제안한 방법은 Fig. 15 에서 보는 바와 같이 사용자가 용도에 따라 옹셋 곡면의 정밀도를 조절할 수 있다. 거리장으로부터 옹셋 곡면을 추출하는 과정에서 복셀의 크기를 조절함으로써 최종 옹셋 곡면의 질을 제어할 수 있다. Fig. 16 에서 보는 바와 같이 복셀의 크기가 감소하면 계산시간이 기하급수적으로 증가하게 되므로 용도에 따라 적절한 복셀의 크기를 선택해야 한다. 정밀한 제품의 NC 가공경로생성을 위한 옹셋 곡면이라면 복셀 크기를 작게 하고 그다지 정밀하지 않아도 무방한 모델의 옹셋 곡면이라면 다소 복셀 크기를 크게 하여 생성되는 곡면의 질과 계산시간을 적절히 제어할 수 있다.

Fig. 17 과 Fig. 18 은 법선벡터가 불연속적인 날카로운 모서리를 가진 형상을 옹셋한 후의 결과를 보여주고 있는데 날카로운 모서리부분에서의 라운딩처리가 자연스럽게 이루어지고 있음을 확인할 수 있었다.

Table 1 은 본 연구에서 다루었던 모든 적용 예제에 대한 계산 수행시간을 정리해 놓은 것이다. Table 1 에서 보는 바와 같이 수 십만 개의 삼각형으로 구성된 대형모델에 대해서도 비교적 빠른 시간 내에 정밀하고 후 처리가 필요 없는 옹셋 곡면의 생성이 가능하므로 삼각망 모델을 다루는 여러 산업분야에 실제적 적용이 이루어질 수 있을 것으로 기대된다.

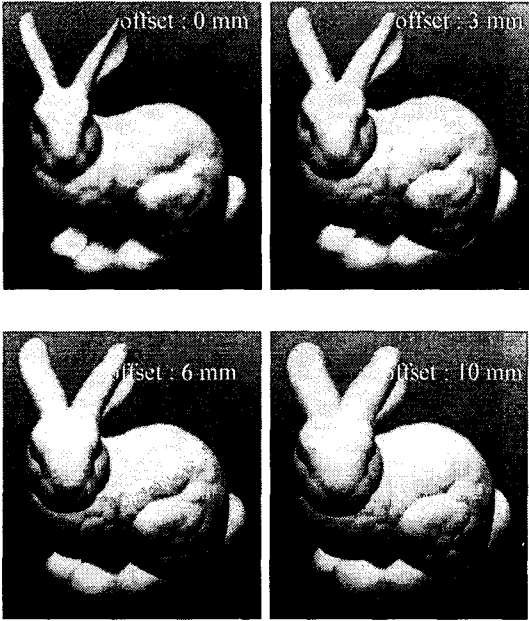


Fig. 11 The offset results of bunny model

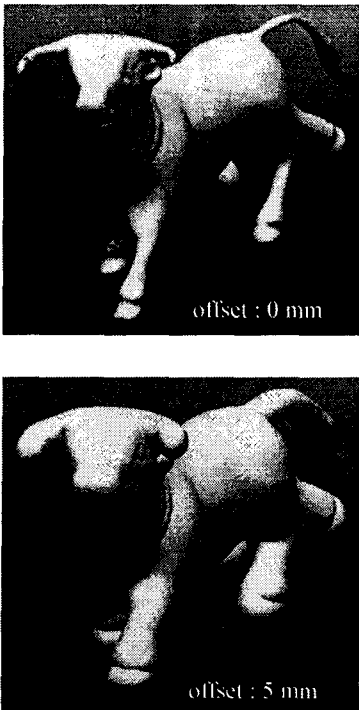


Fig. 12 The offset results of cow model



Fig. 13 The offset results of Armadillo model

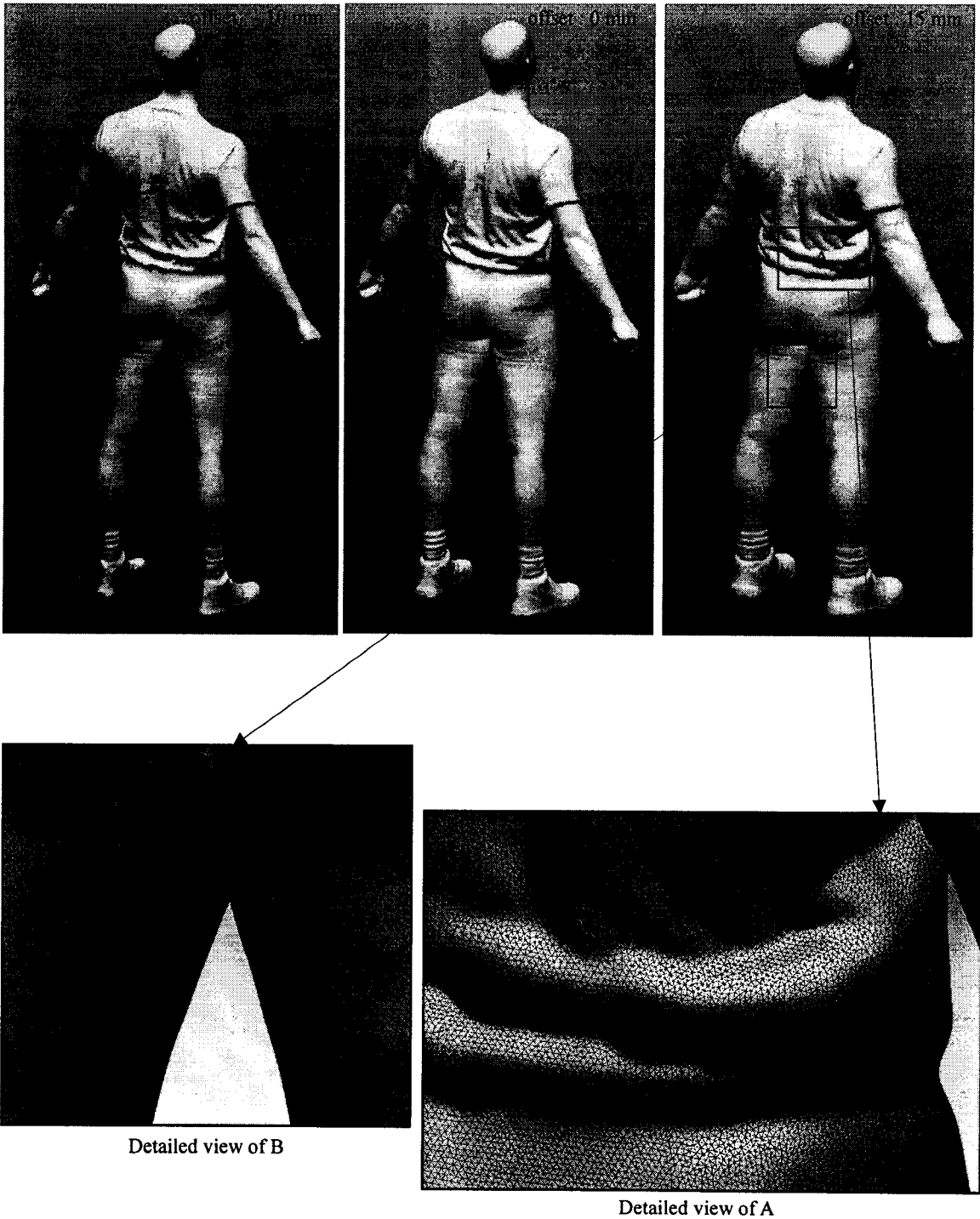


Fig. 14 The offset results of human model





(a) Original model



(b) Cube size : 3 mm



(c) Cube size : 2 mm



(d) Cube size : 1mm



(e) Cube size : 0.5 mm

Fig. 15 The effect of the marching cube size on the quality of offset surface of Igea model

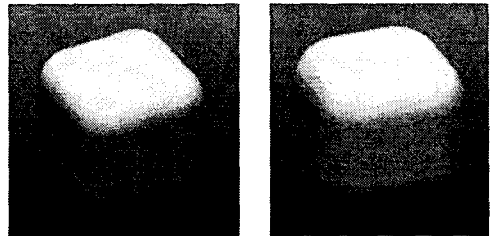
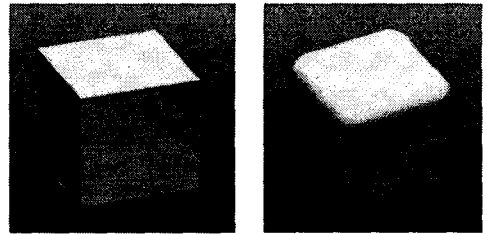


Fig. 17 The offset results of cube

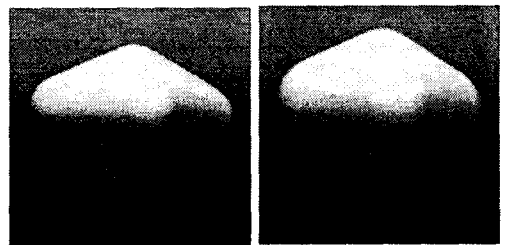
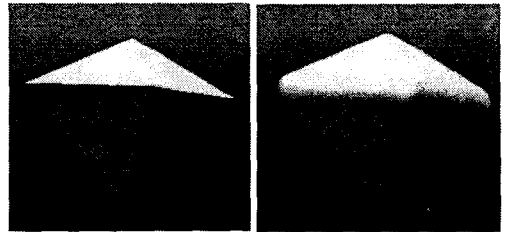


Fig. 18 The offset results of pyramid shape

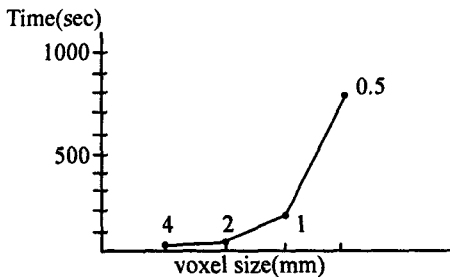


Fig. 16 The variation of computing time according to the voxel size of Igea model

### 5. 결론

본 논문에서는 삼각망 모델에 대한 읍셋곡면을 거리장을 이용하여 구하는 새로운 방법을 제안하였다. 정의된 거리장으로부터 읍셋 곡면을 직접 추출함으로써 종래의 공간발생문제나 삼각형들간의 간섭 문제를 원천적으로 해소할 수 있었으며 읍셋 곡면이 될 유효 셀만을 효율적으로 선별하는 알고리즘을 제안하여 거리장의 계산시간을 대형

Table 1 Computational results

Model	Number of triangles of model	Number of triangles of offset model		Time(sec)	
				Calculation of distance fields	Extraction of offset surface
Bunny	70,368	3mm	179,020	22	4
		6mm	198,750	41	7
		10mm	215,120	62	12
Cow	353,970	5mm	555,276	221	35
Armadillo	441,060	2mm	517,422	133	24
		5mm	627,422	245	43
Human	610,848	-10mm	538,086	630	112
		15mm	827,882	720	119
Igea	426,846	3mm	17,192	31	5
		2mm	68,848	43	7
		1mm	304,362	165	28
		0.5mm	828,614	680	115

( H/W : Pentium IV, 3 GHz CPU, 512MB RAM)

모델에 대한 실제 응용이 가능할 정도의 수준으로 단축시켰다. 향후에는 거리장 계산을 더욱 효율적으로 수행할 수 있는 새로운 알고리즘의 개발 및 형상 부위의 복잡성에 따라 적용으로 거리장을 계산하는 방법을 연구할 계획이다.

### 참고문헌

1. 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.
2. Kim, S. J. and Yang, M. Y., "Offset of STL Model Generated from Solid Model," Journal of the Korean Society for Precision Engineering, Vol. 22, No. 9, pp. 202-211, 2005.
3. Jung, W. H., Jeong, C. S., Shin, H. Y. and Choi, B. K., "Extracting a Regular Triangular Net for Offsetting," Journal of the Korean Society of CAD/CAM Engineers, Vol. 9, No. 3, pp. 203-211, 2004.
4. Breen, D., Mauch, S. and Whitaker, R., "3D scan conversion of CSG models into distance volumes," Proc. Of IEEE Symposium on Volume Visualization, pp. 7-14, 1998.
5. Gibson, S., "Using Distance Maps for smooth surface representation in sampled volumes," Proc. of IEEE Symposium on Volume Visualization, pp. 23-30, 1998.
6. Cohen-Or, D., Levin, D. and Solomovici, A., "Three - dimensional distance field metamorphosis," ACM Transactions on Graphics, Vol. 17, No. 2, pp. 116-141, 1998.
7. Parker, S., Parker, M., Livnat, Y., Sloan, P., Hansen, C. and Shirley, P., "Interactive ray tracing for volume visualization," IEEE Transactions On Visualization and Computer Graphics, Vol. 5, No. 3, pp. 238-250, 1999.
8. Yoo, D. J., "Shape Reconstruction from Unorganized Cloud of Point using Adaptive Domain Decomposition Method," Journal of the Korean Society for Precision Engineering, Vol. 23, No. 8, pp. 89-99, 2006.
9. Yoo, D. J., "A Study on Filling Holes of Large Polygon Model using Implicit Surface Scheme and Domain Decomposition Method," Journal of the Korean Society for Precision Engineering, Vol. 23, No. 1, pp. 174-184, 2006.