

점 데이터를 이용한 3차원 형상의 구현을 위한 소프트웨어 개발 채희창

Development of a 3D Shape Construction Software Using Unorganized Point Data

Hee-Chang Chae

Abstract

Reverse engineering is an emerging technology to obtain CAD models from existing physical parts in the case that CAD models are not available or parts are changed and modified so that new CAD models for final parts are necessary. Reverse engineering helps designers to quickly generate computer interpretable data from existing physical objects. So it is applying for field of Rapid Prototyping, NC Processing, CAE, Inspection and so on.

The objective of this study is to develop the software that deals with unorganized point data and quickly obtains CAD model. In this paper, several models such as human's bone, car, are experimented by the proposed methods.

Key Words : Reverse Engineering(역공학), Point Data(점 데이터), STL

1. 서론

3차원 측정장치의 개발로 물체 표면의 정확한 측정이 가능하게 됨으로써, 측정 데이터로부터 곡면을 재구성하는 방법에 대한 연구가 Reverse Engineering, CAD/CAM, Geographic Data Processing, Biomedical Engineering, Medical Science 등의 분야에서 활발히 진행되고 있다. 3차원 측정 장치로부터 얻어진 클라우드 데이터는 설계로부터 생산, 해석, 혹은 신속 시작의 과정을 일관되게 하는데, 이는 클라우드가 수치제어 밀링이나 다각형 격자 생성을 위

한 프로그램으로 상대적으로 쉽게 변환해 주기 때문이다.

자유곡면 형상의 모델은 충분한 정의를 위해 수천에서 수 만개의 점 데이터가 요구되고, 더욱이 자유곡면은 섬세하고 미묘하며 유연하다. 따라서, 접촉에 의한 측정보다는 비접촉에 의한 빠른 측정이 요구되어진다.

역설계란 이미 기존에 있는 제품으로부터 3차원 측정을 통하여 특정 데이터를 얻은 후 이로부터 신속하게 CAD 모델을 생성하는 기술이다. 제품 개발 시 미적 감각이 요구되는 제품 디자인은 대부분 전문 디자이너가 클레이나(Clay)나 기타 다루기 쉬운 소재를 수공으로 제작하여 이루어지게

된다. 시작품 제작이 완료된 후 디자인 검증이 뒤따르며 제품이 양산 단계로 연계되기 위하여 CAD 데이터가 필요하다. 그러나 현재 많은 제품 형상이 자유곡면의 복합체로 구성되어 있기 때문에 3차원 CAD 모델링 시 상당한 어려움이 뒤따른다. 기존에 3차원 CAD 데이터가 없이 제작된 제품을 개조하여 새로운 제품을 개발하고자 할 때 이와 비슷한 문제가 발생한다. 이에 대응하여 최근 자동차, 가전, 전자, 항공 등 거의 모든 제조관련 분야에 걸쳐서 역설계 기술이 널리 이용되고 있다. 역설계 기술은 신제품의 개발 이외에도 기존에 사용되는 금형이나 제품의 검사 또는 복제, 의료 분야에서는 인공관절 및 치아 제작, 수술 준비에 필요한 모형이나 3차원 CAD 모델 생산에도 점차 널리 활용되고 있다⁽¹⁾.

3차원 측정기를 이용하여 제품에 대한 점 데이터를 얻고 그 얻어진 점 데이터를 STL 파일 형식으로 변환하여 CAD 데이터를 얻을 수 있게 된다. 이 얻어진 CAD 데이터를 이용하여 제품에 대한 수정이나, 복제, 신제품 개발에의 응용에 사용할 수 있게 된다. 3차원 측정기를 통하여 얻어진 점 데이터를 STL 파일로 만드는 데 있어서, 기존의 소프트웨어는 비싼 가격과 능숙한 숙련자라 할지라도 CAD 데이터를 얻기 위한 많은 시간이 소요된다. 본 연구는 점 데이터를 STL 파일로 변환하여 CAD 데이터를 얻고자 하는데 그 목적이 있다. STL 파일은 Rapid Prototyping, NC 가공 등에 이용할 수 있고, 다른 Reverse Engineering 소프트웨어의 입력 파일 형식으로도 쓰일 수 있기 때문에 활용도가 높은 파일 포맷이다. Fig. 1은 점 데이터를 CAD 데이터로 변환하기 위한 기존의 소프트웨어 프로세싱 과정과 본 연구에서

제시하는 프로세싱 과정을 보여주고 있다.

2. 입력 데이터의 조건과 속도 향상을 위한 알고리즘

2.1 입력 데이터의 조건

본 연구에서 실물형상에 대한 CAD 모델을 구축하는데 입력되는 정보는 배열되지 않은 3차원 특징점의 좌표값뿐이고 측정점들 사이의 관련정보 및 실물 형상에 대한 추가적인 정보가 전혀 없다. 이와 같은 경우에 실물 형상과 최종 구축된 CAD 모델의 위상학적, 기하학적 오차를 줄이기 위한 간단하고 효과적인 방법은 측정밀도를 높이는 방법, 즉 측정점의 수를 늘리는 것이다. 측정점의 수가 늘어남에 따라 구축된 CAD 모델은 위상학적인 오류 없이 실제 형상에 수렴하게 된다. 그러나 계산에 소요되는 시간과 전산자원의 부담이 늘어나게 되므로 전체적으로 동일한 밀도로 측정하는 것보다는 측정 대상물의 국소적인 형상이 복잡한 정도에 따라 측정밀도의 조정이 가능한 것이 바람직하다. 즉, 형상이 급격히 변화하는 작은 곡률(curvature)을 가진 부위는 측정간격을 작게 부여하고 곡률이 큰 부위는 위상학적인 오류가 발생하지 않는 범위 내에서 최대한 측정간격을 크게 부여하는 것이 good sampling이라고 할 수 있다. 특히 얇은 두께의 실물을 측정할 경우에는 반드시 반대방향 표면상에 위치한 특징점까지의 거리보다 작은 간격으로 측정해야 오류 발생을 방지할 수 있다. 또한 구축된 CAD 모델이 만족스럽지 못할 경우에는 오류가 발생한 부위의 특징점을 추가하거나 삭제함으로써 실물에 근접할 수 있도록 입력 데이터의 수정이 용이해야 한다.

2.2 데이터 구조

2.2.1 일반적인 데이터구조의 종류

일반적인 데이터구조에는 배열(array)을 정의하고 배열의 원소를 지시하는 번호(index)를 사용하여 랜덤 액세스(random access)하는 방법과 링크 리스트(linked list)에 해당 원소의 자료와 전후의 원소를 지시하는 지시자(pointer)를 붙여서 연결하고, 리스트의 머리(head) 지시자를 통해 순차적(sequential)으로 액세스하는 방식이 있다. 그 외 데이터베이스를 이용하는 방법도 있다.

(1) 배열방식(array)

각 원소의 주소가 규칙적으로 증가하는 방식으로서 임의 번째 원소의 주소를 즉각 계산할 수 있기 때문에 액세스 속

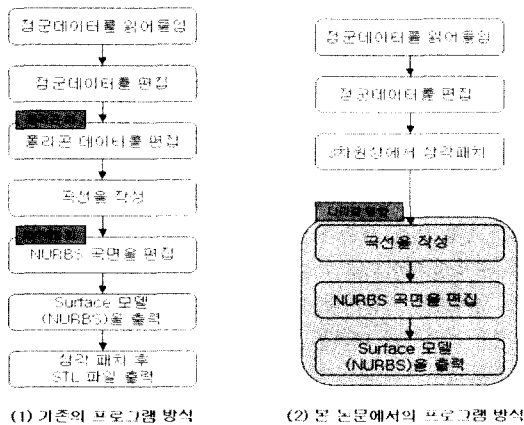


Fig. 1 점 데이터를 CAD 데이터로 변환하기 위한 프로세싱 과정

도가 빠르다. 원소의 최대 개수를 가정하여 정해진 개수의 메모리를 확보한다. 따라서 데이터의 개수가 작은 경우 메모리의 낭비를 가져올 우려가 있으며, 배열번호가 최대 개수를 초과하는 것을 주의해야 한다. 액세스 속도가 빠른 장점이 있지만, 원소들 사이에 새로운 원소를 삽입하거나 특정 원소를 삭제하려면, 삽입 또는 삭제할 위치 이후의 모든 원소를 뒤로 밀거나 앞으로 당겨야 한다. 주로 공정 메모리를 사용하며, 프로그램 실행시 배열의 크기만큼 연속적으로 비어있는 동적 메모리를 할당받아 사용할 수도 있다.

(2) 링크 리스트방식(linked list)

배열방식과 달리 각 원소의 위치가 불규칙적이다. 따라서 각 원소를 연결해 주는 지시자가 필요하다. 순차적인 액세스 방식이며 원소의 개수를 자유롭게 조절할 수 있어 데이터 개수의 변화폭이 큰 경우에 사용된다. 각 원소 옆에 전후를 연결하는 지시자를 가지고 있기 때문에 개수가 많은 데이터의 경우 지시자에 해당하는 메모리의 비율만큼 낭비가 된다. 그러나 동적인 구조이기 때문에 새로운 원소의 삽입이나 삭제는 매우 쉬우며, 지시자들을 변경하여 연결만 시키면 된다. 주로 프로그램 실행 도중에 동적 메모리로부터 한 개씩 할당받아 지시자들을 연결해 놓는다. 연결방법에 따라서 원소의 앞과 뒤의 원소를 모두 연결하는 이중연결(doubly linked)과 뒤에 있는 원소만 연결하는 단순연결(singly linked)로 구분된다.

2.2.2 데이터의 저장

데이터 구조를 설계함에 있어 고려되어야 할 사항으로는 메모리의 효율적 사용, 실행시간의 단축, 알고리즘의 간편성을 들 수 있다. 본 연구에서는 검색의 범위를 최소화하고 검색 속도를 빠르게 함과 동시에 메모리의 최저 사용을 위해 다음과 같은 방식으로 데이터 구조를 설계하였다[Fig. 2].

(1) 꼭지점의 저장[Fig. 2.(1)]

꼭지점의 검색, 추가, 삭제의 편의를 위하여 이중연결 링크 리스트를 사용하였다. 각 꼭지점 노드(node)에는 실수형 X, Y, Z값이 들어가고 vertex에 연결되어 있는 모든 Edge의 정보를 vertex가 가지기 위해 edge의 포인터 노드를 가리키는 head를 가지고 있다.

(2) Edge 저장[Fig. 2.(2)]

Edge 노드에는 두 개의 vertex형 포인터와 하나의 triangle형 포인터가 저장된다. Edge는 삼각형 생성시 세 개의 half edge가 생성이 되고 각 half edge는 자신을 포함하는 삼각형의 주소를 가지게 된다. Edge의 생성은 삼각형이 형

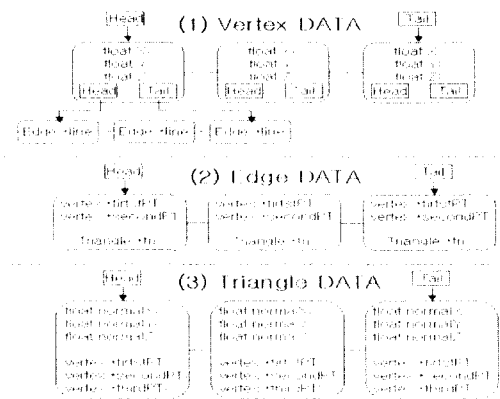


Fig. 2 Data structure

성되어 저장 될 때 삼각형을 구성하는 세 개의 모서리가 edge로 저장된다.

(3)삼각형 저장[Fig. 2.(3)]

Triangle 노드에는 삼각형을 표현하는 세 개의 vertex형 포인터와 Normal 벡터를 저장하기 위한 실수형 변수 세 개가 저장된다. 삼각형의 저장시 삼각형을 이루는 모서리가 이미 edge로써 존재하면 삼각형을 만들지 않는다.

2.3 속도 향상과 메모리 절약을 위한 알고리즘

3차원 상의 점 데이터를 비교 검색하여 중복점과 근접점을 제거하거나 삼각망을 형성하는 과정에서 3차원 균일 격자(3D Uniform Grid)를 적용하였다. 즉 전체 vertex를 포함하는 box를 만들고 이 box를 일정크기의 cell로 분할하여 vertex를 포함하는 위치의 Cell에 vertex를 저장한다.

격자를 생성하기 위한 첫 단계로 데이터 집합의 최소-최대 박스(Min-Max Box)를 얻는다. 최대-최소값은 점 데이

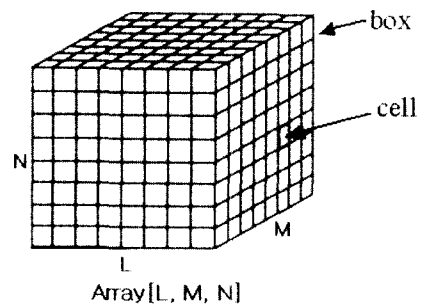


Fig. 3 Shape of vertex cell array

터를 불러오는 과정에서 얻을 수 있다. 다음 단계로 격자의 크기를 결정하는데 중복점과 근접점을 제거하는 방식에서와 삼각망 형성시 만들어지는 격자의 크기를 구하는 방식에서는 약간의 차이가 있다. 다음에서 각각의 격자의 크기를 구하는 방식을 서술하였다.

(1) 중복점과 근접점 제거시 격자의 생성

중복점과 근접점 제거시 격자의 크기를 구할 때는, 격자의 크기를 *size*라 할 때 다음과 같이 구한다. 먼저 *size*값을 0.01로 초기화시킨다. 이 값은 3차원 측정기의 정밀도를 고려하여 주어지는 값이다. 더 작은 값으로 주어도 좋다. 이때 격자의 크기가 제거하고자 하는 근접점 거리의 1.5배보다 작을 때는 다음 공식에 의해 구한다. 근접점 거리의 1.5배를 고려하는 것은 *point*를 제거할 때 비교하는 대상이 하나의 *cell*에서만 비교하기 때문에 *cell*과 *cell*사이에 위치한 근접점의 거리를 비교하여 근접점 거리 이내의 점을 제거하기 위함이다.

$$size = distanceTolerance * 1.5 \tag{1}$$

이때, *distanceTolerance*는 제거하고자 하는 근접점의 거리. 격자의 크기를 정해 주면 다음과 같이 X, Y, Z 방향으로의 격자의 수가 산출된다.

$$x_{no.} = \text{int}\left(\frac{x_{max} - x_{min}}{size}\right) + 1 \tag{2}$$

$$y_{no.} = \text{int}\left(\frac{y_{max} - y_{min}}{size}\right) + 1 \tag{3}$$

$$z_{no.} = \text{int}\left(\frac{z_{max} - z_{min}}{size}\right) + 1 \tag{4}$$

x_{no.}, *y_{no.}*, *z_{no.}*는 각각 X, Y, Z방향으로의 격자의 수를 의미한다. *int()*는 실수를 정수로 변환하는 *cast* 연산자(*cast operator*)이다.

위에서와 같이 격자를 생성하고 각각의 *cell*에서 중복점과 근접점을 제거한 후, 다시 *cell*과 *cell* 사이에 존재하는 근접점을 제거하기 위해 *box*의 위치를 X, Y, Z 방향으로 *size* × 0.5 만큼 각각 이동시키고 다시한번 중복점과 근접점을 찾아 불필요한 점들을 제거해 나간다.

(2)삼각망 형성시 격자의 생성

삼각망 형성시 점 데이터의 비교가 많아지게 된다. 이 경우 걸리는 시간은 최악의 경우 $O(n^4)$ 이 되는 경우도 있다.

그러나 격자를 생성하고 주위의 *cell*만을 비교 검색하는 경우, 검색하는데 걸리는 시간은 현저히 줄어들게 된다. 삼각망 형성시 격자의 크기를 구하는 방식은 다음의 식에 의해 구한다.

$$size = \sqrt[3]{\frac{(x_{max} - x_{min}) * (y_{max} - y_{min}) * (z_{max} - z_{min})}{PointNumber}} \tag{5}$$

이때, *PointerNumber*는 입력받은 *point data*의 개수.

삼각망 생성시 격자의 크기는 위의 공식에 의해 자동으로 생성되기도 하고 3차원 측정기의 특성에 따라 사용자가 임의로 주기도 한다. 격자의 크기에 따라 검색 및 연산 속도의 차가 커지거나 작아지게 된다. 식 (5)에 의해 X, Y, Z 방향에서의 격자의 수를 구한다. 위에서 구한 격자의 수에 의해 동적 메모리를 할당하고, 각각의 *cell*에 *vertex*를 저장하게 된다.

3. 3차원 삼각망 구성법

3.1 Point Data로부터 3차원 삼각망 구성법

Delaunay Triangulation 방식은 3차원 점 데이터에 적용하기에는 부적합한 면이 있다. 일반적으로 *Delaunay* 방식을 적용하기 위해서는 점들을 2차원 평면에 투영을 하고 삼각망을 형성해야 하는데 *Reverse Engineering* 분야에서 적용하고자하는 점 데이터는 3차원 폐곡면에 가까운 데이터들이 주류를 이룬다⁽²⁾⁻⁽³⁾. 또한 Fig. 4에서와 같이 3차원 점 데이터를 하나의 평면에 투영하여 얻은 삼각망과 Fig. 5의 본 연구에서 만들어낸 삼각망을 비교해 보면 3차원 상태에서 3각망을 만드는 것이 훨씬 양질의 형상을 만들어 내는 것을 확인할 수 있다. *Delaunay* 삼각법은 점들을 2차원에 투영할 때 형상이 수직에 가까우면 그 부분은 형상이 고르지 못한 경향을 보인다. 본 연구에서는 *Max-Min Length* 평가법과 삼각형 사이의 *Angle* 평가법이라는 새로운 기법을 도입하여 3차원에서의 삼각망을 구성하고자 한다⁽⁴⁾⁻⁽⁶⁾.

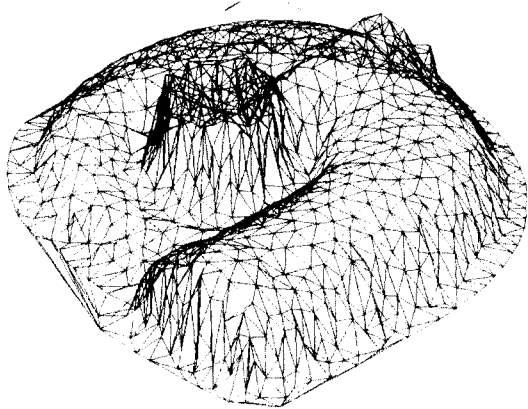


Fig. 4 점 데이터를 2차원에 투영하여 Dealunay를 적용한 Sample

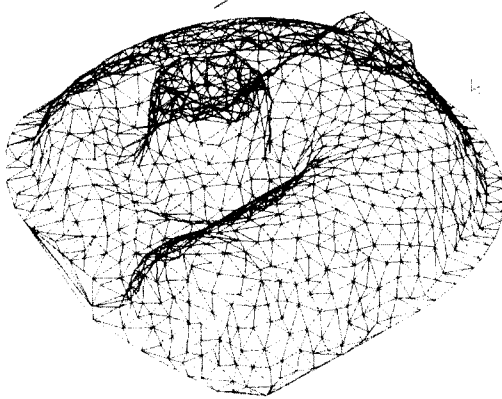


Fig. 5 본 연구에서 제안한 방식에 의해 생성된 삼각망

3.1.1 Max-Min Length 평가

효과적인 삼각망 구축에 있어서 가급적 정삼각형의 형태를 이루도록 해야한다. 본 연구에서 제시된 방법은 최초 선택된 삼각형의 각 edge에 대하여 실물 형상의 표면을 따라 인접한 삼각형들을 평가하여 가장 만족스러운 삼각형을 최종 삼각망에 추가시키게 된다. Fig. 6에서와 같이 stack에서 꺼낸 edge에 대하여 주위의 점들을 검색하고 최적의 삼각형을 이루는 점을 찾아 triangle list에 추가하게 된다. 이때 최적의 삼각형을 찾는 방식은 Max-Min Length 평가에 의해 이루어진다. Max-Min Length 평가법은 Fig. 6에서와 같이 정삼각형에 가장 가까운 점을 찾아서 삼각형을 형성하도록 한다⁽⁷⁾.

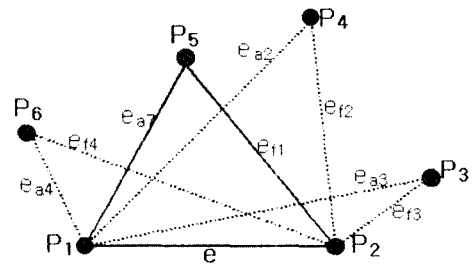


Fig. 6 Max-Min length 평가법

Max-Min Length 평가법의 방식은 다음과 같다.

- ① stack에 저장된 하나의 edge를 꺼낸다.
- ② 꺼낸 edge와 주위의 점들 중 하나의 점과 edge의 양 끝점으로부터 그 점까지의 거리를 계산한다. 두 개의 거리값 중에서 더 큰 값을 저장한다.
- ③ 다시 주위의 다른 점을 가지고 ②과정을 수행한다.
- ④ 주위의 점들을 모두 검색하고 나면 ②과정에서 저장된 길이값 중에서 가장 작은 값을 가지는 점을 가지고 삼각형을 형성하게 된다. 이때 만들어지는 삼각형의 세 개의 edge를 검색하여 그 edge가 이미 존재하면 그 삼각형은 triangle list에 추가하지 않는다.

3.1.2 삼각형 사이의 Angle 평가

삼각형 사이의 Angle을 평가하는 것은 이미 만들어진 삼각형(기준 삼각형)에서 하나의 edge를 가지고 다음 삼각형을 만들기 위한 평가기준이다. Fig. 7에서와 같이 공유하는 edge의 중심에서 기준 삼각형과 평가할 삼각형의 나머지 꼭지점에 대한 내각이 어느 각도 이내에 존재하면 삼각형 형성의 조건을 만족한다. 이들 삼각형 중에서 Max-Min Length 평가를 하여 최적을 이루는 점을 찾아 triangle list에 추가하게 된다. 본 연구에서는 삼각형 사이의 각도 평가

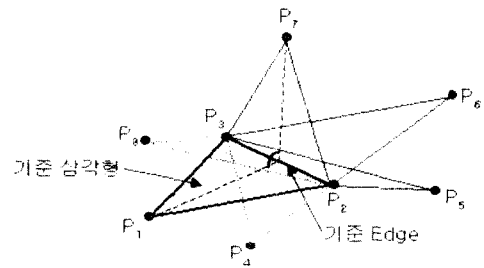


Fig. 7 삼각형 사이의 Angle 평가

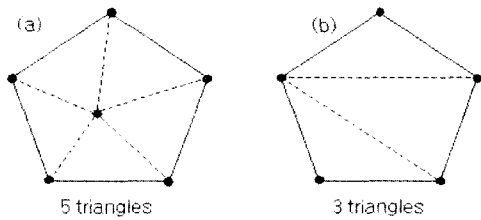


Fig. 8 Triangulation after deleting inner vertex

를 위해 초기 값으로 $\pm 135^\circ$ 값을 주고 있으나 사용자 정의에 의해 이 값을 변경시킬 수 있게 하였다. 형상에 따라 적절한 값을 주면 된다.

3.2 Diet Method

꼭선이나 꼭면 같은 경우 많은 점 데이터를 가진 경우 STL의 형태가 정확해 지기는 하지만 그렇지 않은 경우, 즉, 직선이나 평면과 같은 경우에 많은 점은 불필요하다. Diet method는 이러한 경우에 불필요한 데이터를 삭제시키는 기능으로서 꼭지점을 제거하게 된다. 꼭지점을 제거하는 경우는 두가지로 나눌 수가 있는데 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 제한 값을 설정해 준다.

(1) 꼭지점의 주위에 보조모서리만 있는 경우 :

Fig. 8과 같이 동일 평면상의 다각형 내부에 불필요한 꼭지점이 삽입된 경우이다. 그림에서 (a)는 다각형의 내부에 꼭지점이 한 개 포함되어 있고 따라서, 삼각형 5개로 분할되어 있다. (b)의 경우에는 내부의 꼭지점이 삭제되고 재분할 한 경우이다. 이 경우 내부에 포함된 불필요한 꼭지점을 삭제하고 관련 다각형을 삼각형으로 재분할해야 한다.

(2) 꼭지점 주위에 일직선의 실 모서리가 있는 경우 :

Fig. 9와 같이 STL로의 변환 시 일직선 위에 꼭지점이 다수 삽입되어 데이터의 크기가 늘어난 경우이다. 이 경우에도 일직선으로 연결된 모서리를 합쳐서 삼각형의 개수를 줄일 필요가 있다. 이러한 경우에도 각각의 loop에 따라 삼각형 재분할을 해야한다.

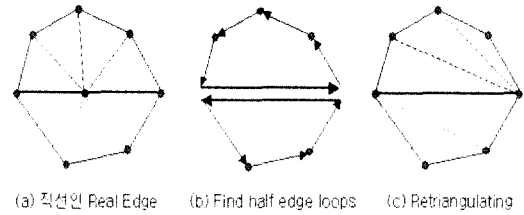


Fig. 9 Delete vertex on collinear real edge and retriangulate

4. 프로그램 시뮬레이션

프로그램 개발환경 및 특징을 간단히 소개하고 몇 개의 점 데이터를 STL 파일로 만드는 과정을 소프트웨어 개발에 응용하여 유효성을 검증하였다.

4.1 프로그램 개발환경 및 특징

플랫폼(platform)으로서 Microsoft사의 WindowNT 4.0을 사용하였고, 개발 도구는 Borland사의 Borland C++ Builder 3.0과 Microsoft의 그래픽 라이브러리인 OpenGL과 Gary의 OpenGL 컴포넌트(component)(8)인 TOpenGL을 사용하였다.

GUI(Graphic User Interface)환경에 적합하도록 모든 명령을 버튼메뉴로 구성하였고, 마우스와 키보드를 이용, 상용 CAD소프트웨어에서와 같이 zooming, panning, rotation 작업을 할 수 있도록 하였다. Clipping 기능을 추가하여 가상적인 slice와 입체의 내부, 오류의 검사 및 복잡한 부위의 일부분만을 가지고 작업을 수행할 수 있도록 하였다.

본 프로그램은 Diet Method를 적용할 때 기준이 되는 정밀도를 임의로 조절할 수 있도록 하여, 정밀성과 빠른 데이터 처리의 양면성을 고려할 수 있도록 하였다.

4.2 소프트웨어 화면 구성

Fig. 10에서와 같이 화면은 5개의 View로 나누어진다. Fig. 10은 사람의 뼈를 측정된 데이터를 삼각망을 형성한 전체 화면을 보여주고 있다.

- ① Tool bar : 사용 가능한 메뉴들이 나타난다.
- ② Window 창 : 위상정보(topology information)의 그래픽을 처리하여 보여준다.
- ③ STL Process 창 : 점 데이터로부터 STL 파일을 출력하기 위한 과정을 page control의 순서에 따라 처리하게 된다. Page control은 Adjust Point, Make Cell,

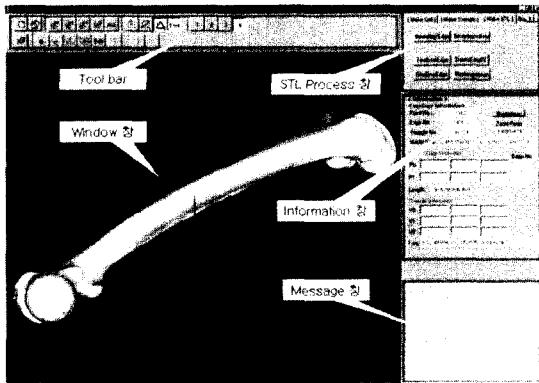


Fig. 10 본 연구에서 구현한 소프트웨어의 화면구성

Make Triangle, Error Modify, Diet 5개의 page로 구성되어 있다.

- ④ Information 창 : Point, Edge, Triangle에 대한 정보를 보여준다.
- ⑤ Message 창 : 어떤 기능이 수행되면 그 결과를 보여준다.

4.3 소프트웨어 시뮬레이션

다음에서 검증할 점 데이터는 3차원 레이저 측정기(Surveyor 계열)로 측정한 데이터이다. 사람의 뼈(Bone), 자동차 외관(Car)을 측정한 점 데이터를 STL 데이터로 변화하는 과정을 검증한다.

4.3.1 사람의 뼈(Bone)

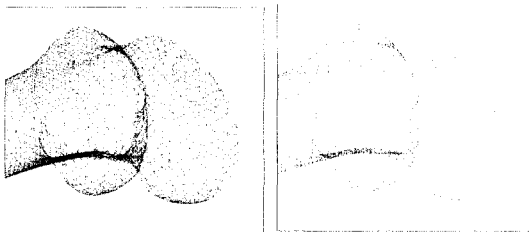


Fig. 11 Scanner로 읽어들이는 Fig. 12 중복점, 근접점을 제거한 후

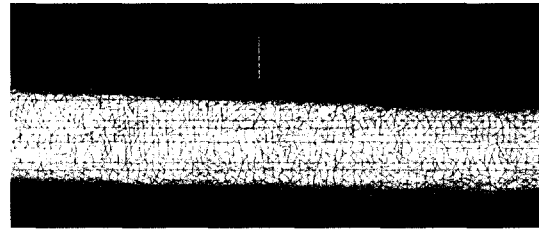


Fig. 13 점데이터로부터 형성된 삼각망

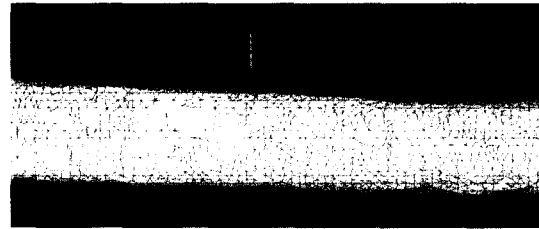


Fig. 14 Diet Method를 사용하여 데이터의 양을 줄인 후

[Bone]	입력 데이터	Triangulation	Diet
Vertex	82931	23422	22200
Edge	0	140178	133188
Triangle	0	46726	44396

Model Size : 455 × 96 × 89.5
 근접점 제거를 위한 Tolerance : 1.0
 Cell Size : 2.5
 Cell 개수 : 183 × 39 × 36
 Triangulation 처리시간 : 3초

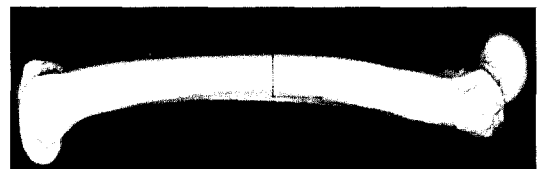


Fig. 15 점 데이터로부터 삼각망 형성이 완성된 상태

4.3.2 자동차(Car)

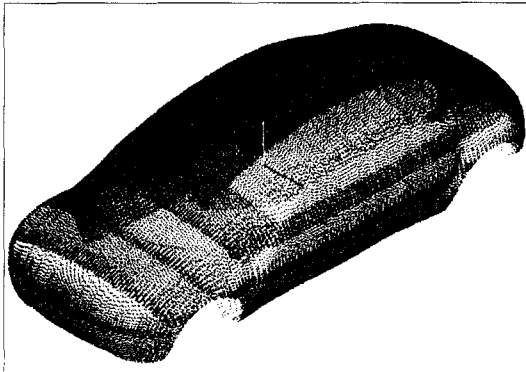


Fig. 16 읽어들이는 점 데이터(Car)

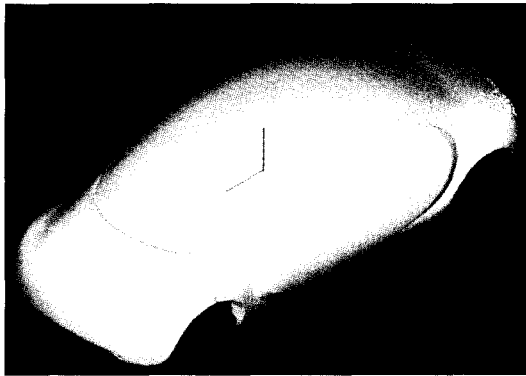


Fig. 17 점 데이터를 삼각형 패치를 한 후 Diet 과정을 거쳐 완성된 형상

[Car]	입력 데이터	Triangulation	Diet
Vertex	63961	39241	32703
Edge	0	232695	195309
Triangle	0	77565	65103
Model Size : 77.49×33.72×22.88 근접점 제거를 위한 Tolerance : 0.2 Cell Size : 0.416 Cell 개수 : 187×82×55 Triangulation 처리시간 : 4초			

5. 결론

본 연구에서는 점 데이터를 STL 파일로 만드는 과정에서 3차원 삼각형 분할시 검색속도, 연산 속도의 저하와 오류 발생의 원인이 되는 불필요한 중복점과 근접점을 제거하였다. 그리고 Max-Min Length 평가법과 Cell 방식을 이용하여 속도를 개선하였고, Diet Method 적용시 전제가 되는 정밀도를 각 모델에 맞게 임의로 조절할 수 있게 하여 정밀 성과 빠른 데이터 처리의 두 가지 성질을 사용자가 선택할 수 있게 하였다.

본 연구에 의한 소프트웨어에 의해 만들어진 STL 파일은 Rapid Prototyping, NC 가공 등에 이용되어질 수 있고, 다른 Reverse Engineering 소프트웨어의 입력 포맷 형식으로도 이용될 수 있다.

본 연구에서 개발한 소프트웨어는 직관적이고 빠르게 Reverse Engineering 결과를 획득하여 궁극적인 리드-타임 단축을 실현시킬 수 있는 소프트웨어가 될 것이다.

참고 문헌

- (1) 채희창, 명태식, 김태규, “역설계를 이용한 자유곡면 모델링 시스템 개발”, 한국공작기계학회지, 제9권 제3호, pp. 111~122.
- (2) 채희창, 김동우, “삼각형기반 형상모델러를 이용한 STL의 검증”, 한국공작기계학회지, 제6권 제1호, pp. 51~58.
- (3) 명태식, 채희창, 김옥현, “Delaunay 삼각형 분할법의 RP에의 응용”, 한국공작기계학회지, 제8권 제3호, pp. 35~41.
- (4) Hans Hagen, Surface design using triangular patches, Computer Aided Geometric Design, pp. 895~904, 1996.
- (5) O Volpin, Mesh simplification with smooth surface reconstruction, Computer Aided Design, Volume 30, pp. 875~882, 1998.
- (6) B. K. Choi, H. Y. Shin, Y. I. Yoon and J. W. Lee, “Triangulation of scattered data in 3D space”, Computer-Aided Design, Vol. 20, No. 5, pp. 239~248, 1988.
- (7) Y. H. Chen and Y. Z. Wang, “Genetic algorithms for optimized re-triangulation in the context of

- reverse engineering”, Computer-Aided Design, Vol. 31, No. 4, pp. 261 ~ 271, 1999.
- (8) J. D. Biossonnat, “Geometric structures for three-dimensional shape representation”, ACM Transactions on Graphics, Vol. 3, No. 4, pp. 266 ~ 286, 1984.
- (9) <http://www.hellix.com/Products/TOpenGL.asp>.