

투시도로부터 3차원 입체 형상 재구성 (Reconstructing 3D Solid Shape from a Single Perspective Drawing)

조 유정, 김 광수
포항공과대학교 산업공학과 CAD/CAM 연구실

요약

2차원 도면으로부터 3차원 입체 형상으로의 재구성은 CAD/CAM에서 많이 연구되어 온 분야이다. 이 논문에서는 건축에서 쓰이는 3점 소실법을 이용하여, 자유곡선을 포함하는 하나의 투시도로부터 입체형상을 재구성한다. 하나의 3점소실 투시도면으로부터 3차원 정보를 추출하고, 그 위상학적 관계와 loop detection 알고리듬을 통해서 면정보를 추출한다. 추출된 면정보를 입체 구조로 변환하고 연속성을 부여한다. 그리고 쉬운 사용자 인터페이스를 제공하여 사용자의 의도에 맞는 적절한 입체 형상을 만들어 내고자 하였다.

1. 서론

1.1 동기

현재 많은 산업 디자인은, 산업디자이너들에 의해 그 아이디어 스케치가 만들어지고, 모형을 간단하게 만들어 본 다음, 각종 3차원 모델러(CAD system)를 이용하여 실제 기하학적 정보(CAD data)로 설계해 낸다. 공학적 디자인은 정확한 수치와 위치를 요구하고, 많은 경우, 3차원 모델러에 의존한다.

그러나 기존의 3차원 모델러가 가진 수많은 기능들은 정확하기는 해도 습득하는데 더 많은 시간을 요구했고, 하나의 그림을 그리는 데에도 수백 가지 기능을 선택해야만 했다. 이런 기능은 산업 디자이너에게 사용하기가 너무 어려워서, 아이디어 스케치와 CAD data는 각각 따로 만들어 졌다. 제품을 수정할 때에는 또 다시 아이디어 스케치부터 모든 작업을 되풀이한다. 이런 반복되는 수작업을 줄이고 자유스케치를 직관적인 형태로 CAD data에 연결시켜서 빠른 모형(prototype)을 구하는 것을 필요로 하게 된 것이다.

1.2 기존 연구의 고찰

2D형상에서 3D형상으로의 재구성은 많이 연구되어 왔다. 그 중에서도 삼면도를 이용한 입체의 생성은 건축에서 쓰이는 것으로 가장 많이 연구되었으나, 곡면을 표현할 수 없다는 약점을 가지고 있었다. Lequette는 wire frame view에서 silhouette edge를 정의하여 몇 가지 cylindrical face를 포함하여 표현하였다. 하나의 input에서 3D 구조를 찾아내는 방법도 많이 연구되었다. 그러나 B-rep 표현방법과의 마찰과 internal face의 처리에서 문제를 안고 있다. Agarwal and Bagali는 wire frame을 하나의 graph로 보고, simple loop를 찾아내는 방법으로 solid의 face를 만들어 내었다. 이 방법은 만약 internal face가 여러 개이면 계산량은 기하급수적으로 불어나게되고, 면과 면 사이의 각도를 이용한다는 방법은 자유곡면일 경우 소용이 없게된다.

대부분의 평면 face 또는 cylindrical face에서는 face간의 각도를 이용하여 face간의 관계를 정의한다. 그러나 자유곡면이 face일 때에는 face간의 각도를 정의 할 수 없으므로 완전히 새로운 algorithm을 필요로 한다.

1.3 목적

정확한 치수를 필요로 하지 않는 상품 디자인의 프로토타입(모형)을 3차원으로 나타낼 수가 있게 한다. 즉, 입체 모델 형상(rapid prototype)을 컴퓨터상에서 빠리 가상으로 구현하는 것이다. 그러기 위해서 직관적인 입력(topology) 형태를 줌으로써 산업 디자이너 같은 비전문가도 쓸 수 있는 사용자 인터페이스를 제공하고자 하였다.

또, 하나의 그림을 그리는 데에도 수백 가지 옵션을 주어 사용자가 그 옵션들을 다 지정해야만 동작을 하는 형식을 배제하고, 지금까지의 경험에 의해 많이 쓰이는 형태를 자동적으로 처리해서 입체를 생성한 뒤에 나중에 고치는 형식의 사용자 인터페이스를 취했다.

사용자의 의도에 맞게 면과 면 사이를 부드럽게 잇기 위해 그런 의도가 있었다고 생각되는 부분에 G¹ 연속성을 주었다.

2. 기본 연구 방향

2.1 Coloring

단지 2차원의 그림 하나만으로는 복수개의 3차원 형상이 생긴다. 따라서, 다른 정보를 주거나 복수개의 2차원 도면을 더해 주어야 한다. 이 논문에서는 3차원 정보를 주는 방법으로 색깔 구분법을 사용한다. x축에 수직인 평면상의 곡선, 즉 x축을 법선으로 하는 평면상의 곡선은 파랑, y축에 수직인 평면상의 곡선은 빨강, z축에 수직인 평면상의 곡선은 노랑으로 색을 지정한다. 모든 곡선은 이런 세 가지의 평면상에 존재하고, 각각 선분 또는 자유곡선이다.

2.2 3차원 좌표 찾기

건축에서 쓰이는 3점소실법을 이용, 아이디어 스케치에서 3개의 소실점을 이용하여 원근을 따져서 투시도(projection plane)상의 선분과 곡선, 점 등을 3 차원상의 선분과 곡선 및 점 정보로 구해 내고 그 정보를 3차원 입체 (solid)로 재구성한다.

1) 3차원 변환에서의 성질보존

투시도 상에서의 선분은 그 두 개의 끝점의 최단 거리이다. 끝점을 3차원 공간으로 변환시켜도 그 두 점을 이은 선분은 선분자체를 변환시킨 것과 완전히 일치한다. Bernstein Polynimial에서는 그 내분점들이 그대로 적용이 된다. 즉, 4 개의 3차원 조종점으로 만든 Bezier curve 를 직접 투영한 것과 조종점들을 투영한 후 만든 Bezier curve는 완전히 동일한 것이 된다. 따라서, 3점소실법의 투시도 상의 어떠한 선분과 Bezier curve도 3차원으로 복원했을 때 그 기하학적 성질을 잃지 않는다. 또, 3차 함수(cubic polynimial)나 hermit curve, uniform b_spline curve는 언제나 3차 Bezier curve 로 바꿀 수 있다.

2) 투시도면 찾기

3차원 상에서 모든 x축에 평행한 직선은, 3점 소실도면에서는 한 점에서만난다. x, y, z방향 소실점 (V_x, V_y, V_z)은 반드시 눈에서 같은 거리에 존재하므로 눈의 위치는 세 소실점에서 같은 거리에 있게 된다. 세 개의 소실점은 하나의 평면을 이루게 되는데 이것이 3차원상의 하나의 평면 즉, 투시도면이 된다.

3) 3차원좌표 구하기

시점과 투시도면상의 점(p)을 잇는 직선은 투시도 상에서는 하나의 점(p')으로 나타난다. 투시도면 식을 구하면 모든 3차원상의 점(P)의 그 평면에 투영한 점

(P')들을 찾아 낼 수 있다. 이어진 모든 선분이나 곡선의 다른 끝점의 3차원좌표를 알 수 있다. 이렇게 모든 선분을 따라 가면서 3차원 좌표를 구해낸다.

2.3 3차원 입체 생성

1) loop detection

보조선이 아닌 하나의 선분이나 곡선을 하나의 edge로 보고, edge를 이어가며 찾아진 하나의 simple loop(자신속에 cycle이 없는 loop)를 하나의 걸면(face)으로 본다. loop의 edge들의 반대방향으로 다른 face를 찾을 수 있다. 이 논문에서는 평면 뿐만이 아닌 곡면의 face를 찾아야 하기 때문에 3.2와 같이 간단하고 빠른 알고리듬을 제공한다. End condition, infeasible solid을 check하는 방법과 internal face를 찾는 방법, loop detection을 재개하는 방법 등을 이논문에서 제시하였다.

2) 면(face) 구성

하나의 simple loop는 하나의 face가 된다. 모든 face는 그 edge를 boundary로 하는 Gregory patch 형태로 나타내었다. 3개또는 5개 이상의 edge들로 이루어진 경우에는 [Chyokura]의 방법을 따라 face를 3개 또는 5개 이상의 4-edge patch들로 다시 재구성하였다.

2.4 G¹ continuity 부여

Face의 부드러운 연결을 위해서는 face를 이루는 edge간의 부드러운 연결이 필수적이다. 디자이너들이 자유 스케치를 그릴 때에는 edge들을 정확히 부드럽게 연결 할 수 없다. 따라서 system내에서 이런 오차를 자동적으로 수정하여 edge들을 부드럽게 연결시켜야 했다. 어떤 부분은 현저히 다른 기울기를 가질 수 있는데 이런 경우는 깍어진 부분으로 간주하였다.

3. Algorithm 및 구현 방법

이 시스템은 크게 세 부분으로 나누어 진다. 2차원형태의 입력을 3차원으로 변환하는 부분, 변환된 데이터에서 면정보를 추출 및 재구성하는 부분, 그리고 입체형상에 연속성을 주는 부분이다. 먼저 2차원 데이터를 3차원으로 변환한 다음 기하학적 정보와 위상학적 정보를 이용하여 면을 추출해낸다. 그리고 마지막으로 면과 면사이에 연속성을 준다. Edge 간 연속은 어느 시기에 해도 상관이 없다.

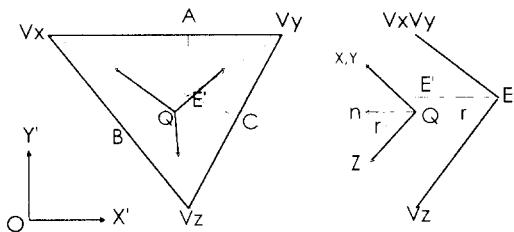
3.1 2차원 좌표의 3차원 좌표로의 변환

1) 보조선(Vanishing Line)과 소실점(Vanishing Point)

group 이 0, 1, 2 인 edge는 직선으로 각각 x, y, z축에 평행한 직선인다. 이들은 3점소실법에서 보조선이다. group 이 같고 일치하지 않는 두 개 쪽의 보조선을 끌라 각각의 교점을 구하면 그것이 소실점이 된다. 보조선들의 입력형태도 선분이므로 연장(extend)한 무한직선으로 계산한다.

2) 시점(Eyepoint)을 결정

[그림 1]에서처럼 시점(Eye, Eyepoint)의 투시도(Projection plane)상의 점(Eye')은 삼각형 VxVyVz 의 수심이 된다. 이 중명은 [Xiuwei]에 따랐다. 원점 Q(origin)과 시점의 투시도상의 점(Eye')이 결정되면 실제 시점좌표를 구할 수 있다.



[그림 1] 3점 소실도의 좌표계를 앞에서 본 그림과 옆에서 본 그림

3) 최초의 known point 결정

대개 한 보조선의 끝점을 원점으로 설정을 한다. 또는 한 점의 좌표를 알고 있다면 그것으로 설정한다.

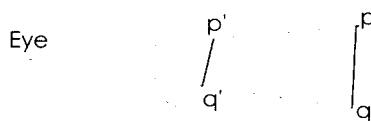
4) 보조선들의 식을 구한다.

하나의 보조선을 3차원 좌표로 변환할 때는 x,y,z 중 하나를 알고 있으면 구할 수 있다. 이런 원리를 이용하여 보조선들의 식을 이미 아는 점으로부터 구한다.

5) 모든 점들의 3차원 좌표를 구한다.

한 끝점의 3차원 좌표가 알려지면 그 점과 edge로 연결된 모든 점들의 좌표를 구할 수 있다. 예를 들어 [그림 2]에서처럼 한 점 p의 좌표가 (150,0,200)로 알려졌다 고 할 때, 이 점을 끝점으로 하는 group 의 0 edge는 x=150 의 평면상에 존재한다. 따라서 나머지 점 q(edge는 네 점 중의 하나)는 다음과 같이 구한다. Eye, q'을 알면 직선 Eye q'을 알 수 있다. 이 직선과 x=150과의 교점을 구하면 그것이 q의 좌표가 된다.

Projection plane plane x=150



[그림 2] 3차원 좌표 생성하기

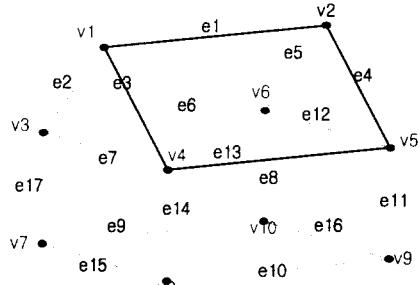
3.2 자동 면 생성

1) 시점(Eyepoint)에서 가장 가까운 거리의 edge

여기에서 거리는 edge상에 존재하는 가장 가까운 점(edge상에 존재하는 모든 실수의 합집합)과의 거리로 정의한다. 곡선의 경우에는 전부 Bezier curve로 전환시키다. 선분의 경우 시점에서의 선분까지의 가장 가까운 거리는 수선의 밭이 되지만 수선의 밭이 그 사이에 없을 때에는 두 끝점 중 하나가 된다. 그러나 곡선의 경우 3차함수 이상이 되므로 계산과정이 복잡하고 느려서 4개의 Bezier control point를 가지고 계산하였다. 이렇게 각각의 edge에 대해서 계산한 다음, 가장 거리가 짧은 edge를 선택한다.

2) edge의 status

Edge의 status=0은 아직 어느 face에도 소속되지 않았다는 뜻이다. Status=1은 출발점 도착점과 같은 방향의 edge가 face에 소속되었다는 뜻이고 100은 두 face에 소속되었다는 뜻이다. 이 시스템에서는 이것이 작업 완료시기(end condition)로 작용하기도 하고 half edge를 만들어가는 데 쓰인다. [그림 3]에서 만약 face1이 구성되고 face2가 구성되면 e1은 두 개의 face에 구성되므로 100 상태가 된다. 만약 이렇게 모든 edge가 두 개 쪽의 face를 가지게 되면 하나의 solid 가 된다.



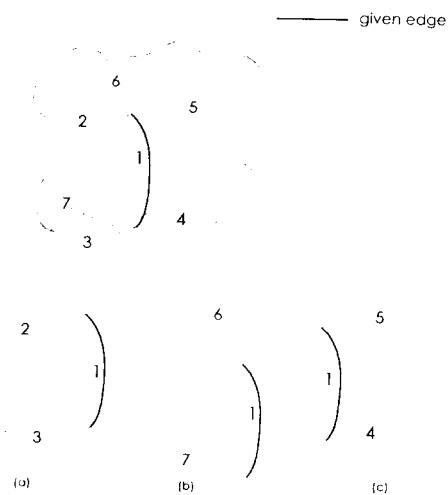
$$\text{face1} = e1-e3-e13-e4$$

$$\text{face2} = e1-e5-e6-e2$$

[그림 3] 면(face)과 edge의 관계

3) 최초의 face

Projection plane에서 눈에 보이는 edge(not hide)는 그대로 실제3차원에서도 눈에 보이는 edge이다. 최초의 face 후보들 중 겉면이 되는 것 중 하나를 최초의 face로 결정하고 나면 그 face를 따라 edge의 방향을 정해야 하는데 대부분의 경우 face의 바깥면에서 edge는 반시계 방향(CCW counter-clockwise)으로 이어진다. [그림 4]에서 처음 loop가 최초의 face로 결정되었으면 이 face는 반드시 눈에 보이는 face이다. 모든 edge는 두개씩의 방향을 가지고 하나의 face에서 edge의 방향이 결정되면 나머지 모든 face의 방향은 [그림 5]과 같이 자동적으로 결정된다.



[그림 4]

시점에서 가장 가까운 edge와 그를 포함하는 면(face)들

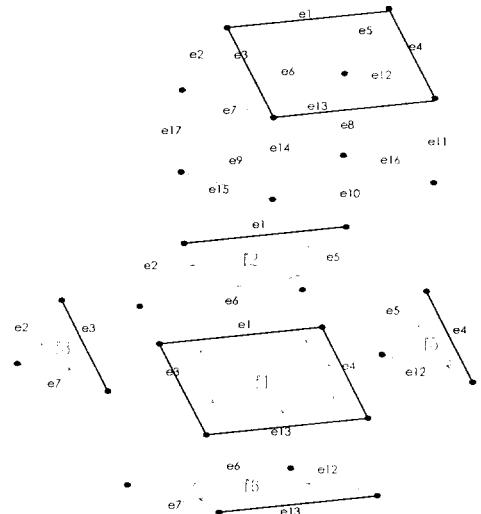
4) 연결된 face 찾기

loop를 이루는 edge는 방향성을 가진다. 이런 방향을 가지는 edge를 half edge라고 하고 같은 점을 가지고 방향이 반대인 half edge를 twin이라고 한다. 하나의 f1을 찾으면 f1을 둘러싸는 half-edge의 twin으로 나서 tree를 구성하여 face를 찾아낸다. 이렇게 face를 찾아 나가며 모든 face는 half-edge의 원쪽에 존재하게 된다. 만약 어떤 face를 찾은 즉시 그 face의 모든 edge가 complete가 되어 더이상 tree를 만들 edge가 없으면 stop한다. 만약 지금까지의 모든 edge가 두번씩 face를 형성했다면 이것은 완전한 solid가 된다.

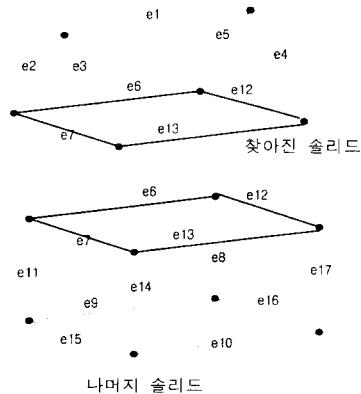
5) internal face 제거하기

완전한 solid가 아닌 경우에도 stop하는 경우가 있다. 이것은 internal face가 존재하는 경우인데, 이런 경우 모든 face의 edge는 두 개씩의 face를 갖게 되지만,

모든 edge가 두개씩의 face를 가지는 것은 아니다. [그림 5]을 보면 f1을 기준으로 face를 찾으면 f6이라는 internal face를 포함하게 된다. loop detection이 stop했을 때에는 하나의 solid가 생성된 것이다. 이런 solid가 infeasible solid라면 반드시 internal face가 겉면을 이루게 되고 이면을 사이에 두고 [그림 6]처럼 두 개(두 개 이상이 될 수도 있다.)의 부분으로 나뉘게 된다. solid을 이룬 쪽의 face 중에서 internal face는 반드시 이 두 부분의 경계에 있게 된다. Internal face가 발견되면 이 face를 이루고 있는 loop와 이 loop와 연결되면서 나중에 만들어진 모든 face를 같이 풀어준다. Internal face를 역방향으로 가상의 face를 만든다. 이렇게 만들어진 face를 최초의 face로 다시 loop detection을 하면 internal face는 제거되고 새로운 face가 찾아진다. 이 작업을 계속하면 결국에는 feasible solid을 구할 수 있다.



[그림 5] f1면을 기준으로 이웃하는 면들을 찾은 경우



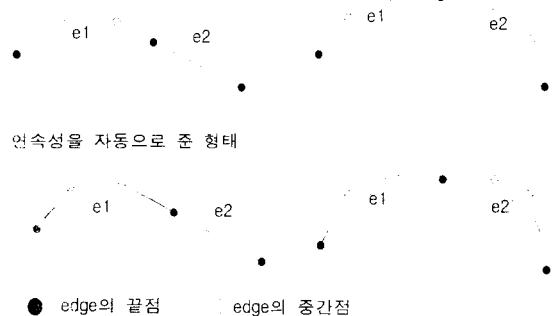
[그림 6] internal face로 인해 불완전한 입체가 된 모습

3.3 연속성(G¹ continuity) 부여

1) Edge간 연속성 부여

사용자는 연속성을 부여해야 하는 곳에 정확한 좌표를 주지 못하는 경우가 있다. 이런 경우, 끝점은 변하지 않고 형태를 사용자의 의도에 맞게 유지하면서 G¹ continuity를 주게 하였다. 이렇게 조정하고 나면 edge 구조의 타입은 Bezier가 되고, 중간점 두 개는 Bezier control point의 중간점이 된다. 곡선직선이 만나는 곳은 직선의 기울기를 곡선의 끝 기울기(Bezier의 tangent)로 하였다. [그림 7]에서 그 예를 보여준다.

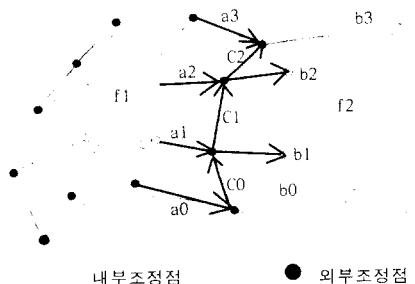
입력형태



[그림 7] 입력된 edge에 연속성을 부여한 모습

2) Patch간의 연속성 부여

입력형태 그 자체로는 patch를 만들 수 없다. 따라서 각 patch간에 연속인 점을 이용해서 내부 조종점 8개를 구해 내었다. Patch간의 연속성을 주는 방법은 [Chiyokura]를 따랐다.



[그림 8] 이웃하는 Patch간에 연속성을 준 모습

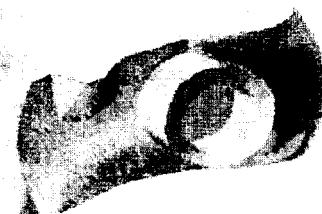
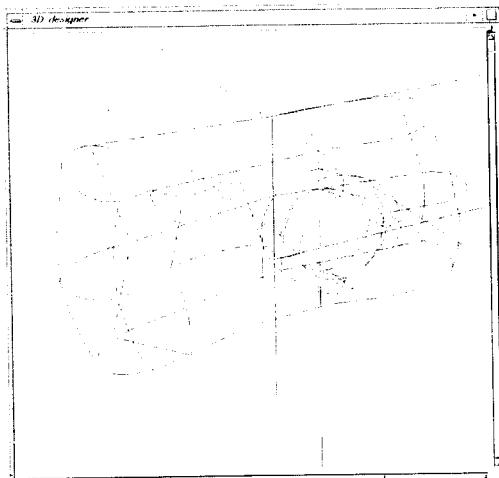
이렇게 각 patch의 내부 조종점을 연속성을 이용하여 주게되면 20개의 Gregory patch의 모든 점들이 구해진다.

3) 부정형 면 (Irregular Patch)의 재구성

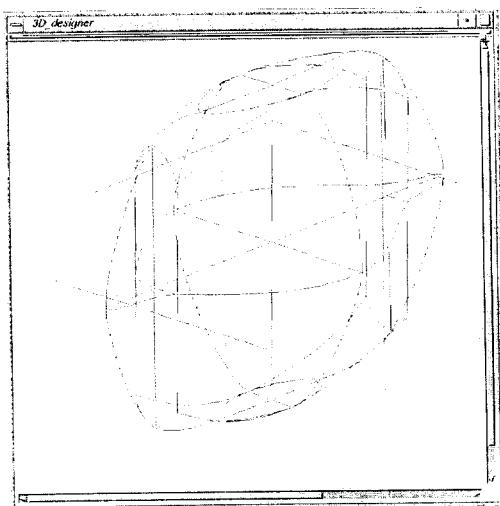
Gregory Patch는 그 연속성이나 그 모양이 4 개의 edge들에 의해서 정의된다. 그러나 3.2절에서 구한 면을

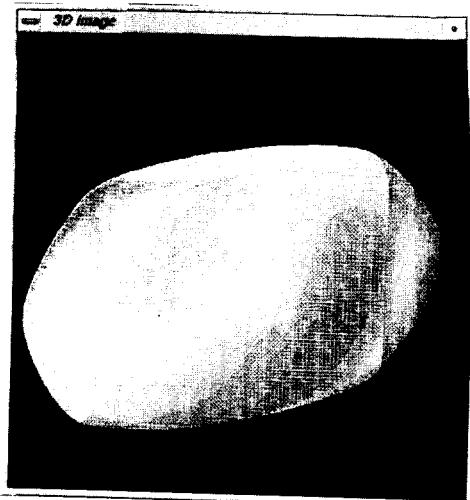
은 3개 또는 5개 이상의 edge들로 구성되어 있다. 따라서 이를 면을 Gregory Patch로 바꾸는 작업을 하여야 한다. 이 방법은 [Chiyokura]를 따랐다.

4. 결과



[그림 9] 카메라 형상의 입력과 결과





[그림 10] 향수병 형상의 입력과 그 결과

5. 결론

기존의 평면이나 cylindrical face 를 위주로 하던 방법에서 더 나아가 자유곡면까지도 충분히 표현할 수 있었다. 단지 몇개의 자유곡선의 집합으로 자유곡면을 가진 입체를 표현함으로써 다양한 형태를 쉽게 표현할 수 있게 되었다.

쉬운 사용자 인터페이스를 제공한다. 직관으로 알아보기 쉬운 평면상에 자유곡선을 표현하게 하는 방법이 사용자에게 많은 편의와 자유를 줄 수 있었다. 또, 가능한 한 사용자의 선택을 필요로하는 많은 기능을 없애고 거의 자동화하는 형식을 취했다. 역시 수정에 있어서 빠르고 쉬운 인터페이스를 제공한다.

모든 면(face)을 찾기위해 많은 search를 하던 것(언제나 complexity O(n²))을 loop detection이라는 알고리듬을 통해 빠른 시간내에 결과를 볼 수 있었다.

빠른 수행시간과 쉬운 입력형태는 산업디자이너로 하여금 같은 시간내에 많은 모델을 만들 수 있게 한다. 결과로 나온 data 는 B-rep구조이므로 번역기를 설치해 컴퓨터 내에서 바로 원하는 data type으로 변화시킬 수 있기 때문이다.

추후 연구의 대상으로는 사용자의 일반적 상식을 시스템 내에서 좀더 강화시키는 것과 사용자의 틀린 그림에 대한 자동 수정이 있다. 현재, 정확한 좌표를 구하기 위한 연구도 되어지고 있으며 실제 종이에서 바로 입력을 받을 수 있는 방법도 연구되고 있다.

참고 문헌

- [Agarwal] S. C. Agarwal and W. N. Waggenapack Jr., "Decomposition method for extracting face topologies from wireframe models ", Computer_Aided Design Vol. 24, No. 3, pp. 123-140, March 1992
- [Akeo] Makoto Akeo, Horishi Hashimoto, Taisuke Kobayashi, and Tetsuo Shibusawa, "Computer Graphics System for Reproducing Three-Dimensional Shape from Idea Sketch", EUROGRAPHICS, Vol. 13, No. 3, pp. C477 - C488, 1994
- [Bagali] Siddarameshwar Bagali, and Warren N. Waggenapack Jr., " A Shortest Path Approach To Wireframe To Solid Model Conversion", Proceedings Third Symposium on Solid Modeling and Applications, pp.339-349, 1995
- [Chiokura] Hiroki Chiokura, " Solid Modeling with DESIGNBASE", Addison-Wesley Publishing Co., 1988
- [Gujar] Uday G. Gujar and I. V. Nagendra, " Construction of 3D solid objects from orthographic views", Comput. & Graphics, Vol 13, No. 4, pp. 505-521, 1989
- [Lequette] Remi Lequette, " Automatic construction of curvilinear solids from wireframe views", Computer-Aided Design Vol. 20, No. 4, pp. 699-717, May 1988
- [Wang] Weing Wang, D. Sc., Georges G., and Grinstein, " A Survey of 3D Solid Reconstruction from 2D Projection Line Drawings", COMPUTER GRAPHICS forum, Vol. 12, No. 2, pp. 137-158, 1993
- [Yan] Qing-Wen Yan, C. L. Phillip Chen, and Zesheng Tang, "Efficient algrothm for the reconstruction of 3D objects from orthographic projections", Computer-Aided Design Vol. 26, No. 9, pp. 699-717, September 1994
- [Zhao] Xiuwei Zhao, and Jiaguang Sun, " Reconstruction of a Symmetrical Object from its Perspective Image", Comput. & Graphics, Vol 18, No. 4, pp. 463-467, 1994