

## 단면도를 이용한 3차원 파라메트릭 설계

### 3D Parametric Design Using Orthographic Views

김병인\* · 김광수\*

Byungin Kim and Kwangsoo Kim

#### Abstract

Orthographic views are traditionally used for engineering drawings. This paper presents a methodology for 3D parametric design using orthographic views. The parametric design technique, which is used to design 2D orthographic views, is based on production rules. In the production rule-base, several view interrelation rules and over 50 geometric rules are included. An efficient algorithm is also developed to expedite the reasoning process. For 3D object construction from orthographic views, the approach known as bottom-up geometrical approach is used. The approach consists of 4 steps : 1) generation of wire-frame, 2) construction of face from wire frame, 3) formation of 3D subobjects from faces, and 4) construction of final 3D objects. Curvilinear solids as well as planar solids can be constructed. A method of converting existing 2D CAD data to parametric 3D CAD data is also presented.

#### 1. 서론

최근 CAD시스템의 동향은 와이어 프레임 모델링이 솔리드 모델링으로 대체되는 과정에 있다. 와이어 프레임 모델이 형상의 모서리와 꼭지점 정보만 갖는데 반하여 솔리드 모델은 면에 관한 정보도 포함하기 때문에

물체를 더욱 정확하게 표현할 수 있다. 그러나 이러한 장점에도 불구하고, 솔리드 모델링의 설계방법이 기존의 설계 방법과 달라 설계자들이 적용하기가 쉽지 않다. 일반적으로 기계부품의 설계시 사용하는 방법은 삼면도(정면도, 평면도, 측면도)를 이용한 방식이다. 따라서 솔리드 모델러에 삼면도를 이

\* 포항공과대학교 산업공학과

용하여 설계할 수 있는 기능을 제공할 필요가 있다. 그리고 이때 지능적인 설계 기법으로 인정되고 있는 파라메트릭 설계(parametric design)기능을 제공할 수 있으면 더욱 좋은 모델러가 될 수 있을 것이다.

또한, 솔리드 모델러가 개발되었지만, 이미 작성된 수많은 2차원 CAD도면들은 솔리드 모델링 기법으로 설계되지 않았기 때문에 솔리드 모델러가 도면을 있는 그대로 이용하기에는 정보가 부족하다. 따라서 기존의 2차원 도면을 이용하기 위해서는 이를 솔리드 모델로 바꾸어 줄 수 있는 시스템이 필요하다. 이때에도 시스템이 도면의 치수 등을 해석하여 파라메트릭 설계 기능을 부여 할 수 있다면 유사부품들의 설계를 쉽게 얻을 수 있을 것이다.

본 연구는 크게 2부분의 관련연구 분야로 나누어 지는데, 파라메트릭 설계 기법과 2차원 단면도로부터 솔리드를 생성하는 방법에 관한 것이다. 파라메트릭 설계 기법은 점, 선, 곡선 등의 형상 요소간의 관계를 치수에 관한 함수로 나타내어 형상을 정의하는 방법으로 형상의 수정을 용이하게 할 수 있는 기법이다. 이와 관련하여 Light등[1]은 치수와 형상간의 관계를 방정식으로 나타낸 후에 수치해석적 방법을 이용하여 형상을 정의하는 방안을 제안하였다. 그런데 이 방법은 수치해석적 방법을 사용하기 때문에 초기해에 따라 최종해가 달라질 수 있고, 불완전하게 정의된 형상은 처리할 수 없게 된다. 또한 치수를 변화시킬 때마다 연립방정식을 풀어야 하기 때문에 시간이 많이 소요된다.

Aldefeld[2,3], Roller[4,5,6,21], Sunde[7], Kimura[8] 등은 인공지능(AI)기법을 이용하

여 파라메트릭 설계 기법을 구현하였다. 이 방법은 rule-base, fact-base 와 inference engine 을 이용하는 전형적인 인공지능 기법(추론: reasoning)을 이용한다. 이 방법의 장점은 일단 형상이 추론에 의하여 정의되면 수정이 실시간에 이루어질 수 있다는 데에 있다. Roller[4,5,6,21]가 고안한 방법은 시스템이 주어진 형상에 대한 실행순서(sequence of execution)를 가지고 있어서 생성될 형상의 치수에 해당되는 변수값이 입력되었을 때 원하는 형상의 변이(variant)를 내는 방법이다. 그러나 이 방법은 다소 간단한 방법이라는 하지만 설계자가 형상을 설계할 때 형상이 만들어지는 절차에 따라서 설계를 해야하기 때문에 사전에 형상에 대한 정보를 알아야하므로 모델링에 많은 제한요소가 따르게 된다. 그리고 추론과정에서 많은 시간을 필요로 하는 단점이 있다[9]. 따라서 본 연구에서는 형상설계의 순서와 관련이 없는 설계방법을 제시하며 추론시간을 줄이기 위한 휴리스틱 방법을 제시한다. 상세한 내용은 2절에서 설명한다.

2차원 단면도로부터 솔리드를 생성하기 위한 연구는 크게 2가지 접근방법이 있다. Markowsky등[10], Gujar등[11], Lcquette[12] 등은 기하학적인 구조를 기초로 3차원 꼭지점, 모서리, 면, 솔리드를 찾는 단계적인 방법을 제안하였고, Courter[13], Ganter등[14]은 각 면의 경계를 이루는 모서리들의 연관관계에 중점을 둔 위상학적(topological) 접근방법을 제안하였다. 또한 사용되는 2차원 단면도의 수에 따라서도 연구가 진행되고 있다 [15,16]. 이 분야의 연구에 있어 가장 큰 문제는 단면도로부터 생성되는 솔리드가 한개

이상일때 어떻게 처리할 것인가와 취급할 수 있는 곡면의 범위에 관한 것이다. 본 연구에서는 솔리드가 한개 이상일때에도 처리할 수 있게 하고, 원통형, 구 등의 곡면을 포함한다.

본 연구에서는 기존에 각각 수행되어 온 파라메트릭 설계 기법과 2차원 단면도로부터 솔리드를 생성하는 방법을 동시에 적용하는, 삼면도를 이용한 3차원 물체의 파라메트릭 설계 방법을 제안한다. 제안된 방법을 솔리드 모델러에 적용하면, 설계자에게 익숙한 2차원 단면도를 이용한 3차원 물체의 설계가 가능하다. 또한 파라메트릭 설계 기능이 추가 되기 때문에 3차원 형상의 수정을 용이하게 할 수 있다. 그리고 이 방법은 기존의 2차원 도면을 솔리드모델로 바꾸어 주는 데에도 적용될 수 있다.

본 연구에서 개발한 시스템의 설계과정은 다음과 같다. 먼저 설계자가 삼면도를 그리면서 치수를 부여하면 시스템은 각 삼면도의 형상간의 관계 및 치수와 형상간의 관계를 규정한다. 이 과정에서 시스템은 필요한 보조선을 제공하여 설계자가 삼면도간의 관계를 고려하여 설계할 수 있도록 돕는다. 이렇게 2차원 파라메트릭 삼면도의 형상이 정의되면 삼면도로부터 3차원 형상을 인식하여 B-rep으로 형상을 표현한다. 다음에 3차원 형상을 수정할 필요가 생기면, 설계자가 해당되는 치수 파라미터를 선택하여 값을 바꾸어 준다. 이때 시스템은 자동적으로 그 치수에 연관되는 삼면도의 형상 및 3차원 형상을 수정한다.

본 논문의 구성은 다음과 같다. 2절에서는 파라메트릭 설계 기법을 이용한 삼면도의 설

계에 대하여 설명하고, 3절에서는 이로부터 솔리드를 생성하는 과정에 대하여 설명한다. 4절에서는 본 연구에서 제안한 방법을 기존의 2차원 CAD도면에 적용시키는 방법에 대하여 설명하고, 5절에서는 시스템 구현 및 결과를 보이고 6절에서는 결론 및 추후 연구과제에 대하여 언급한다.

## 2. 파라메트릭 설계 기법을 이용한 삼면도 설계

### 2.1 파라메트릭 설계 기법

파라메트릭 설계 기법은 설계를 보다 용이하게 할 수 있도록 하는 지능적인 CAD기법 중의 하나이다. 본 연구에서 사용한 파라메트릭 설계 기법은 앞에서 언급한대로 인공지능에서 사용하는 추론을 도입한 기법이다. 이 방법은 스케치에 의한 형상과 주어지는 치수를 이용하여 아직 정의되지 않은 형상요소들을 단계적으로 정의하는 방법이다. 이미 정의된 형상요소의 정보들을 가지고 아직 정의되지 않은 형상요소를 정의하기 위해서는 이미 알려진 정보의 형태를 판단하여 그 상황에 합당한 수학적인 규칙을 적용시킬 수 있어야 한다. 이때 상황에 합당한 규칙을 선택하기 위하여 추론(pattern matching)을 이용한다. 이를 위해서 형상의 구성요소(점,선,곡선 등)간의 관계와 형상-치수의 관계를 형상을 정의하는 구속조건으로 보고, 이를 기호(symbol)로 표기한다. 또한 수학적인 규칙도 기호로 표기한다.

구속조건으로는 도면에 부여되는 치수(거리, 각도 등)와 형상요소간의 내부적인 상관관계(대칭관계, 수직관계, 포함관계 등) 등이

포함된다. 예를 들어, '점 P1 이 직선 L1 위 에 위치한다' 는 형상요소간의 관계는 하나 의 구속조건이고 'ON P1 L1'과 같이 기호를 사용하여 표현할 수 있다. 또한 '직선 L1과 직선 L2가 수직관계이다' 관계 역시 구속조 건이 될 수 있고 'PERPEND L1 L2'와 같이 표현할 수 있다. 이렇게 기호로 표현된 구속 조건은 fact-base에 저장되어 추론에 사용된

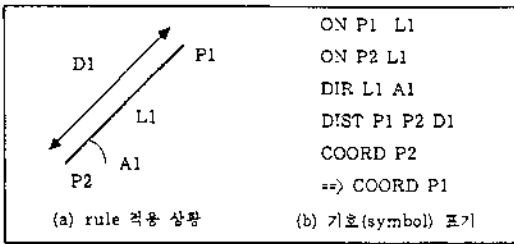


그림 1. 규칙(rule) 예

다.

추론을 하기 위해서는 rule-base가 필요한 데, 여기에 사용되는 rule은 간단한 수학적인 규칙들로서 역시 기호로 표현된다. 예를 들 어, '만약 점 P1과 P2가 직선 L1위에 있고, L1의 방향과 점 P1의 좌표를 알고, 점 P1과 P2 사이의 거리를 안다면, 점 P2의 좌표를 알 수 있다' 는 규칙을 그림 1과 같이 표현할 수 있다. 모든 rule은 조건-결과의 형태를 취한 다. 물론, 각 rule에 대한 실제적인 계산을 하 는 프로그램도 있어야 한다. 현재의 시스템 에서는 50여개의 rule을 사용하고 있다.

이와같이 일반적인 규칙들을 모아놓은 rule-base와 설계시 생성시키는 fact-base가 열 어지면 이로부터 추론과정을 행하여 형상을 단계적으로 정의할 수 있다. 이에 따른 실제 적인 계산은 추론과정이 완료된 이후에 실행

된다. 즉, 추론과정은 순전히 기호수준에서 이 루어 진다.

본 연구에서 구현한 파라메트릭 설계 시스 템은 그림 2와 같다. 시스템은 user interface

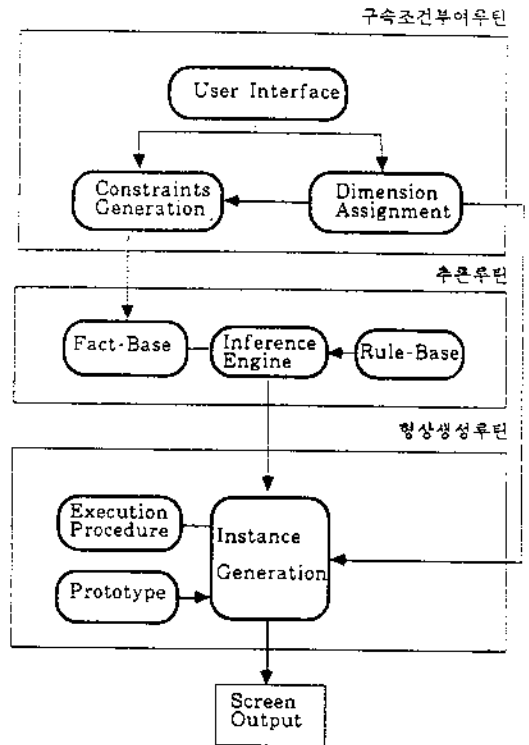


그림 2. 파라메트릭 설계 시스템

를 통하여 사용자로부터 형상의 스케치와 치 수를 입력받는다. 이로부터 구속조건을 이끌 어내고 기호로 표현하여 fact-base를 만들게 된다. 또한 여기에는 inference engine에 의하 여 추론된 결과도 새로운 fact로 첨가되어 다 음단계의 추론에 사용하게 된다. 이때 구속 조건이 부족(under constrained)하여 형상이 완전히 정의되지 않는 경우가 있는데, 이 문 제는 각 형상요소에 flag를 사용하여 해결하 였다. 추론과정이 끝난 후에 flag값을 비교하

여 정의되지 않은 형상요소를 사용자에게 알려 주면 사용자는 구속되지 않은 형상요소를 화면을 통해 추가로 구속시킬 수 있다.

Rule-base 에는 앞에서 언급한 바와 같이 기하학적인 관계를 표현하는 수학적 규칙들이 포함되어 있고, 이는 쉽게 확장이 가능하다. 물론, 확장시킬 때는 각 rule에 해당하는 수학적 procedure도 첨가시켜야 한다. Inference engine에서는 rule-base와 fact-base로부터 pattern matching을 이용하여 새로운 fact를 찾아낸다. 본 시스템에서는 forward chaining 과 depth first search를 택하였다. forward chaining을 사용한 이유는, 본 시스템에서의 추론은 새로운 fact를 생성하는 형태이기 때문이다[17].

추론내력( inference history)은 추론에 사용된 rule들과 그때 사용된 fact를 모아 저장한 것으로서 다음에 실제적으로 수행되는 수학적 계산에 사용된다. 치수가 바뀔때에 다시 추론할 필요없이 찾아진 추론내력에 수치만 대입하면 된다. 또한 사용자가 부여한 구속조건을 없앨때에 그 구속조건에 해당하는 fact와 그 구속조건으로부터 추론된 모든 fact를 없애 주어야 하는데 이때에도 추론내력을 참조하여 해당되는 fact를 지운다. 이렇게 하여야만 시스템이 일관성을 유지하게 된다.

여기까지의 과정은 치수의 직접 대입없이 단순히 기호들로만 수행된다. 이렇게 생성한 추론내력을 참조하여, instance generation 모듈에서는 입력된 치수 값을 대입하여 각 추론에 해당하는 수학적 계산을 한다. 이 과정을 수행하여야 실제적인 수치값들로 구성된 형상이 정의된다. Execution procedure 에는 rule-base에 포함된 각 rule에 해당하는 수

학적인 모델이 포함되어 있다. Prototype geometry는 각 rule 에 해당하는 수학적 계산을 할 때 두가지 이상의 해가 있을 수 있는 경우에 사용된다. 그림 1의 규칙도 양방향으로 P1의 값을 가질 수 있다. 이러한 경우에 본 시스템에서는 두가지 해를 모두 구한후 prototype과 비교하여 합당한 해를 찾게 된다. 이 prototype은 사용자가 입력한 스케치 형상이거나 치수가 변화하기 바로 전단계의 형상이다. 그림 1의 경우에도 바로 전단계의 형상에서 P1과 P2의 상대적인 위치를 고려하여 새로운 P1의 좌표값을 구하게 된다.

## 2.2 추론시간에 관련한 휴리스틱

서론에서도 언급한대로 parametric design 기능을 실현하기 위하여 인공지능기법을 적용할 경우 가장 문제가 되는 부분이 추론에 소요되는 시간이다. 본절에서는 추론시간을 줄이기 위한 3가지의 휴리스틱을 제시한다. 먼저 추론에 사용되는 규칙은 다음과 같이 표현된다.

$$R_1 E_{11} E_{12} E_{13} C_1 (Cond1)$$

$$R_2 E_{21} E_{22} E_{23} C_2 (Cond2)$$

.....

$$R_m E_{m1} E_{m2} E_{m3} C_m (Cond_m)$$

$$\Rightarrow R_k E_{k1} E_{k2} E_{k3} C_k (Conclusion)$$

여기서,  $R_1$  : geometric relationship( = fact type )

$E_{ij}$  : geometric element

$C_1$  : geometric constant

위와 같은 규칙과 fact-base를 이용하여 새로운 fact를 생성하게 되는데 fact-base에 M개의 fact가 있고 해당되는 규칙의 조건문이 m

개라면 이 규칙을 만족시키기 위한 complexity는  $O(M^m)$ 이 된다. 아래에  $m, M$ 을 줄이는 효과를 얻는 휴리스틱을 제시한다.

i)  $m$ 을 줄이는 휴리스틱

추론을 위해 forward chaining을 사용할 경우에는, 이미 알고 있는 사실을 다시 추론할 가능성이 있다[17]. 이를 방지하기 위해서 본 연구에서는 rule을 구성할 때 조건부분(condition part)의 첫번째 조건에 결론부분(conclusion part)의 정보를 가지는 조건을 위치시킨다. 예를 들면, 그림 1의 rule에서 결론 COORD P1의 geometric element인 P1을 포함하는 조건문인 ON P1 L1을 첫번째 조건문에 위치시켰다. 이렇게 rule을 구성시키면, 첫번째 조건문과 fact-base를 match시킨 후에는 결론부분이 어떤 fact임을 알게 되므로, 더 이상의 추론을 할 것인지의 여부를 결정할 수 있게 된다. 추론될 fact가 이미 알고 있는 사실이라면 더 이상의 match를 하지 않는다. 이미 존재하는 fact의 경우 complexity는  $O(M^m)$ 에서  $O(M^1)$ 으로 줄게 된다.

또한 일반적으로 자주 나타나지 않는 조건문을 조건부분의 위쪽에 위치시킴으로써 탐색시간을 줄이고자 하였다. 예를 들어 ‘어떤 점이 어떤 직선위에 있다(ON P1 L1)’ 라는 fact는 자주 나타나는 반면 ‘어떤 직선과 어떤 원이 접한다(TANGENT L1 C1)’ 와 같은 fact는 드물게 나타나는 fact 이다. 본 시스템에서는 후자를 조건문의 위쪽에 위치시킴으로써 탐색시간을 줄였다. 즉, 완전히 match되지 않을 rule의 경우에는 가능한한 조건문을 모두 match시켜 보지 않고 match를 중단하고자 하는 의도를 구현한 것이다. 이 경우의 complexity는 추정하기가 난해하지만

평균의 경우  $O(M^m)$ 에서  $O(M^m)$ 으로 줄게 된다( $m>m'$ ).

ii)  $M$ 을 줄이는 휴리스틱

복잡한 형상의 경우에는 fact-base에 포함된 fact의 수가 수천개가 된다. 따라서 형상의 복잡도가 증가할 경우 추론시간은 기하급수적으로 증가하게 된다. 본 연구에서는 탐색하는 fact-base의 크기를 줄이기 위하여 그림 3과 같이 fact-base를 여러개의 sub-fact-base로 나누어 구성하였다. 이렇게 할 경우 complexity는 각각의 sub-fact-base의 크기가 균일하다면  $O(M^m)$ 에서  $O(M_0^m)$ 으로 줄게 된다( $M_0 < M$ ). 여기서  $M_0$ 는 sub-fact-base크기의 평균이다.

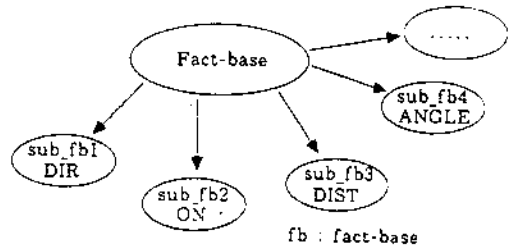


그림 3. Fact-base의 분할

2.3 파라메트릭 설계 기법을 이용한 삼면도 설계

삼면도의 설계에 파라메트릭 설계 기법을 적용하기 위해서는 각 단면도들간의 관계를 규정하여야 한다. 일반적으로 삼면도는 평면도를 XY좌표계, 정면도를 XZ좌표계, 측면도를 YZ좌표계에 표현하므로, 각 단면도들 간에 하나의 축을 공유한다. 따라서 3차원 물체상의 한 꼭지점은 평면도와 정면도에서 같은 X값을, 평면도와 측면도에서 같은 Y값을, 정면도와 측면도에서 같은 Z값을 갖는다. 본

연구에서는 이러한 관점에서 각 단면도들간에 보조선을 도입하여 설계자가 설계하는데 도움을 줄 수 있게 하였다. 아래에서 육면체를 단면으로 설계하는 과정을 통하여 삼면도의 설계에 대하여 설명한다.

먼저 설계자는 3개의 단면 중 하나에서 설계를 시작한다. 여기에서는 설명의 편의상 정면도에서 시작한다. 그림 4(a)와 같이 정면도에서 형상 스케치를 한 다음 구속조건을 부여한다. 다음에 설계자는 치수 등의 값을 변화시켜 원하는 형상을 생성하는데 그림 4(b)에서 변화된 형상을 보여 준다. 정면도 설계가 끝나면 다음으로 평면도 설계를 하는데, 이 때 설계자는 시스템에게 보조선을 요구할 수 있다. 시스템이 제시하는 보조선은 그림 4(c)와 같다. 설계자는 보조선을 참조하여 그림 4(d)와 같이 스케치 및 구속조건을 부여하고 추론을 거친 후에 형상을 수정하여 원하는 형상을 만든다. 마지막으로 측면도를 설계하는데, 앞에서와 같이 시스템이 보조선을 제공할 수 있다. 그림 4(e)에서 이를 보여준다. 그림 4(f)는 보조선을 참조한 측면도의 설계를 보여준다. 측면도의 설계후에 시스템은 추론과정을 통하여 형상을 정의하게 된다. 이렇게 삼면도의 파라메트릭 설계가 완성된 후에는, 하나의 도면치수를 변화시키면 다른 도면의 관련 요소들도 함께 수정 되고, 다음장에서 설명되는 입체도 수정되게 된다. 그림 4(g)에서는 하나의 도면치수를 수정하였을 때의 결과를 보여 준다. 그림 4(h)에서는 원호가 있을때의 보조선을 나타내는 방법을 보인다.

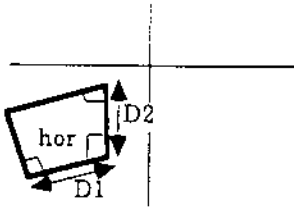
여기서 보조선은 자신을 생성시킨 점과 같이 움직이기 때문에 정면도→평면도, 정면도

→측면도, 평면도→측면도, 평면도→정면도, 측면도→정면도, 측면도→평면도의 여섯가지 종류가 있다. 이 성질 때문에 하나의 보조선에 대응하는 반대방향의 보조선을 생성시킬 수도 있다. 하지만 두 단면도에서 서로 관련된 형상요소들을 연결하여 주는 보조선은 하나로 충분하다. 따라서 동일한 값을 갖는 보조선을 중복되게 생성시킬 필요는 없다. 예를 들어 정면도→평면도, 평면도→정면도의 보조선일 경우는 X값이 보조선을 결정하기 때문에, 정면도나 평면도 위의 어떤 점에서부터 보조선을 생성할 때에는 기존의 정면도→평면도, 평면도→정면도 보조선의 X값들을 참조하여 공차(tolerance)안에 드는 보조선이 이미 존재하면, 그점으로부터는 다시 새로운 보조선을 생성하지 않는다.

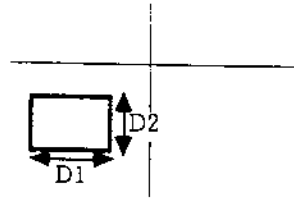
삼면도에 파라메트릭 설계 기법을 도입하기 위해서는 보조선과 연관되는 rule들이 필요하다. 본 시스템에서는 이와 관련하여 10여개의 rule을 만들었는데 그림 5에 한가지 예를 보인다. 이 rule이 의미하는 바는 '만약 점 P1과 P2가 같은 보조선 위에 있고, 두점간의 거리를 알고, 점 P2의 좌표를 알면, 점 P1의 좌표를 알 수 있다'는 것이다.

### 3. 삼면도로부터 입체 생성

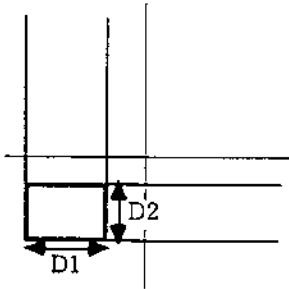
본 연구에서는 삼면도로부터 입체를 생성하는 방법으로 서론에서 언급한대로 기하학적인 방법을 사용하였다. 전체 알고리즘은 Gujar등[11]이 제시한 방법을 택하였는데 곡면을 포함할 수 있게 확장하였다. 그림 6은 알고리즘의 전체적인 흐름을 보여 준다. 먼저 3차원 점(vertex)과 모서리(edge)를 단면



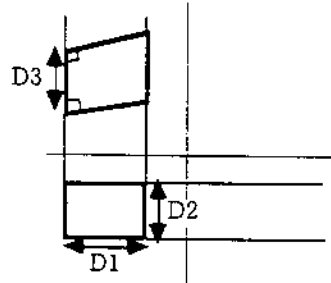
(a) 스케치 및 구속조건 부여



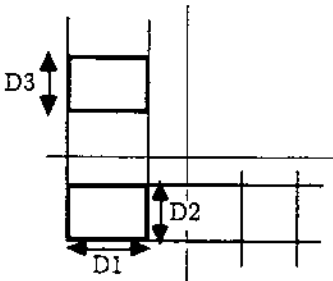
(b) 변화된 형상



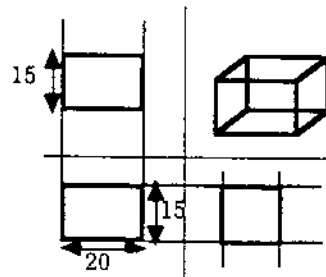
(c) 보조선 생성



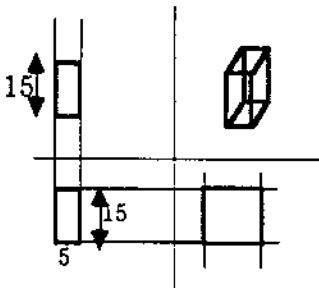
(d) 평면도 스케치 및 구속조건 부여



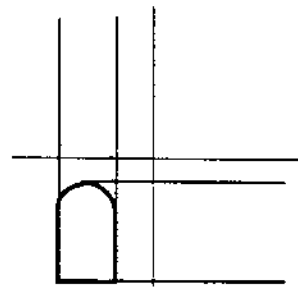
(e) 보조선 생성



(f) 측면도 설계



(g) 도면치수의 수정



(h) 원호에서의 보조선 생성

그림 4. 파라메트릭 삼면도 설계



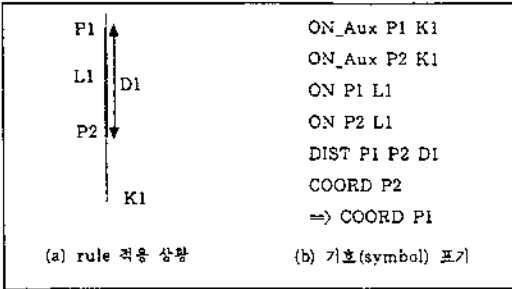


그림 5. 보조선에 관련된 rule 예

도면으로부터 찾아 와이어 프레임(wire-frame)을 생성하고, 면정보를 추출하고 입체를 생성하게 된다. 각각에 대하여 아래에 설명한다.

### 3.1 전처리과정

이 과정은 와이어 프레임을 찾기 위한 준비작업으로서, 각 단면도에서부터 2차원 점(node) 및 2차원 선(line, circular arc)성분을 추출하는 과정이다. 여기에서는 설계자가 직접 마우스로 입력한 점과 선성분 이외에도 선들이 교차할 경우 교차점(intersection point)을 찾아야 하고 이 때 새로운 선성분도 생성해야 한다. 두직선의 교차여부는 참고문헌 [18]에 제시된 알고리즘으로 쉽게 구할 수 있다. 또한 단면도에 곡선이 포함되어 있을 경우에는 그림 7과 같이 실루엣(silhouette) 점 및 선을 첨가시켜 주어야 한다. 실루엣 점은 원호성분의 0°, 90°, 180°, 270°인 부분에 생성시키고, 실루엣 점을 이어 선을 생성시킨다.

### 3.2 와이어 프레임 생성

이 과정은 단면도로부터 3차원 꼭지점 및 모서리를 추출하여 와이어 프레임을 생성하

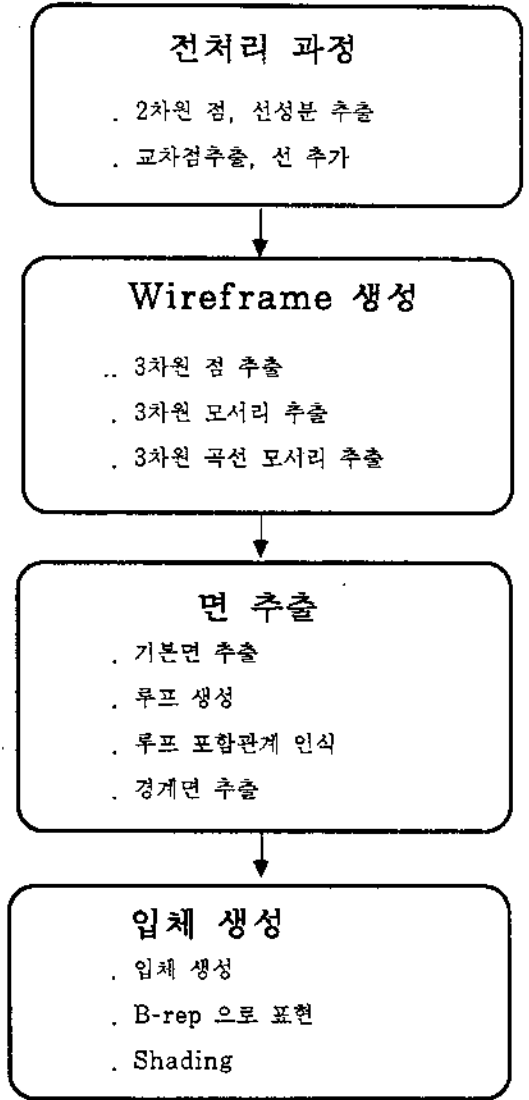
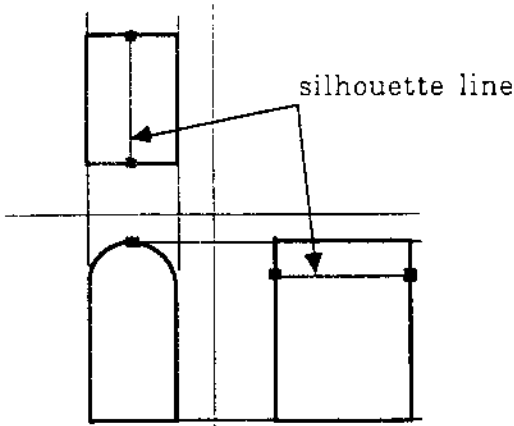


그림 6. 삼면도에서부터 입체 형성 과정

는 단계이다. 정면도와 평면도의 점을 비교하여 공통축(X)의 값이 같으면, 두점으로부터 (Y,Z)값을 찾아서 측면도의 점들의 좌표값과 비교하여 일치하는 점이 존재하면 여기에서 사용된 3개의 점으로부터의 (X,Y,Z) 값을 갖는 3차원 점(vertex)을 생성한다. 이때 3개의 점중에 실루엣점(silhouette vertex)이



■ silhouette point

그림 7. 실루엣 점, 선 첨가

포함되어 있으면 3차원 점을 실루엣점으로 분류한다.

다음으로 3차원 모서리(edge)를 찾는데 다음과 같은 과정을 따른다. 앞에서 생성된 점 중에서 임의의 2개를 선택하여 직선으로 잇고 각 단면에 투사(projection)시켜 본다. 이때 각 단면에 대응되는 2차원 성분(점 또는 선성분)이 모두 존재하면 3차원 직선(모서리)으로 생각할 수 있다. 이렇게 직선성분이 찾아진 후에는 3차원 곡선 성분을 찾는데 과정은 다음과 같다. 먼저 3차원 점(vertex)중에서 실루엣점으로 분류된 것을 각 도면에 투사시켜 대응되는 점들을 찾는다. 다음으로 대응되는 점들중에서 실루엣점과 그와 관련된 원호를 찾아 3차원 점으로 복사(copy)시킨다. 그림 8에서 이과정을 보인다. 이러한 방법은 삼면도로 표현되는 곡선성분이 보는 각도와 수직되어 위치하기 때문에 적용할 수 있는 방법이다.

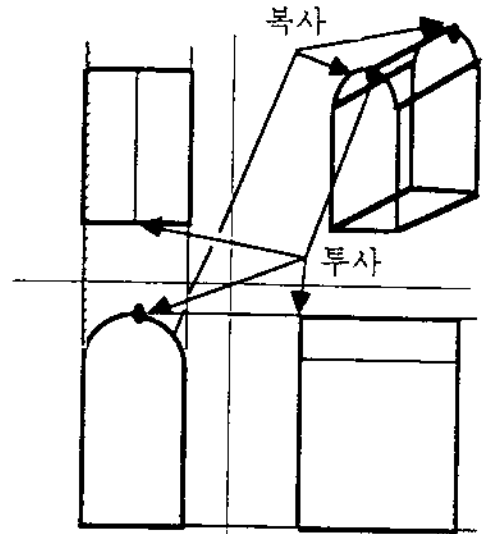


그림 8. 3차원 원호 성분 생성

3.3 면추출

찾은 와이어 프레임으로부터 먼저 기본면(surface)을 찾고 그 이후에 경계면(face) 즉, 3차원 물체의 경계로 사용하는 면을 찾는다. 기본면은 3차원 edge들간의 관계를 가지고 정의할 수 있는데 규칙은 아래와 같다.

- 두개의 직선이 한점에서 만나면 평면이다.
- 직선과 곡선이 한점에서 만나고 한 평면상에 위치할 수 있으면 평면이다.
- 직선과 곡선이 한점에서 만나고 한 평면상에 위치하지 않고 직교하면 원기둥(cylinder)이다.
- 직선과 곡선이 한점에서 만나고 한 평면상에 있지 않고 서로 직교하지 않으면 원뿔(cone)이다.
- 두개의 곡선이 한점에서 만나고 한 평면상에 있지 않으면 반지름과 위치에 따라 구(sphere)이거나 원환체(torus)이다.

위 규칙에 따라 기본면을 찾는데, 각 기본면의 식은 아래와 같이 구해질 수 있다. 먼저 평면의 경우 다음과 같은 음함수 형태로 나타난다.

$$ax + by + cz + d = 0 \quad (\text{식 1})$$

또한 3점( $x_J, y_J, z_J$ ) ( $x_K, y_K, z_K$ ) ( $x_L, y_L, z_L$ )을 지나는 평면은 다음과 같이 구할 수 있다 [18].

$$\begin{vmatrix} x-x_J & y-y_J & z-z_J \\ x_K-x_J & y_K-y_J & z_K-z_J \\ x_L-x_J & y_L-y_J & z_L-z_J \end{vmatrix} = 0$$

따라서 계수는 다음과 같이 구할 수 있다.

$$a = (y_K - y_J)(z_L - z_J) - (z_K - z_J)(y_L - y_J)$$

$$b = (z_K - z_J)(x_L - x_J) - (x_K - x_J)(z_L - z_J)$$

$$c = (x_K - x_J)(y_L - y_J) - (y_K - y_J)(x_L - x_J)$$

$$d = -(ax_J + by_J + cz_J)$$

본 연구에서는 기본면을 다음과 같이 정의하는 방식을 채택한다. 기본면이 원기둥인 경우는 기준이 되는 3차원 원의 중심과 반지름을 알고 축을 알면 정확하게 정의된다. 삼면도를 이용할 경우에는 원기둥의 축이 좌표계의 축과 평행하므로 3가지 형태만이 존재한다. 또한 기본면이 구인 경우에는 구의 중심 좌표와 반지름으로 표현되고, 원뿔인 경우는 원뿔의 꼭지점과 밑면을 구성하는 원의 중심 좌표와 반지름으로 표현할 수 있다. 삼면도를 통해 나타낼 수 있는 원뿔의 형태도 축의 방향으로 3가지로 나타난다. 그리고 원환체(torus)인 경우에는 큰 원의 중심 및 반지름과 작은 원의 반지름으로 표현할 수 있다.

기본면을 찾은 후에는 모든 모서리(edge)

를 각 기본면과 비교하여 포함관계를 확인하여 분류한다. 다음으로 각 기본면에 포함된 모서리들로 루프(loop)를 구성하게 되는데 방향을 일관성 있게 설정하기 위해서는 법선 벡터가 필요하다. 여기서는 평면의 경우 원점에서 출발하는 법선 벡터를 양의 방향(positive direction)으로 설정한다. 그리고 평면 이외의 기본면은 각점에서 법선벡터가 달라진다.

평면의 경우 위에서 구한 계수로부터 원점에서 출발하는 방향으로의 법선벡터는 아래와 같이 구할 수 있다. (식 1)의 계수 a,b,c,d를 이용하여 새로운 법선벡터를 구한다.

원점을 지나고 방향벡터가 (a, b, c)인 직선은 다음과 같다.

$$x = a * t$$

$$y = b * t$$

$$z = c * t$$

$$t = \text{매개변수}$$

이 직선이 평면과 만날때의 t는 (식 2)를 (식 1)에 대입하면 알 수 있다. 즉,

$$a^2t + b^2t + c^2t = -d$$

$$t = -d / (a^2 + b^2 + c^2) \quad (\text{식 3})$$

따라서 원점에서 출발하는 법선 벡터는 (식 3)을 (식 2)에 대입하여 얻을 수 있다. 즉,  $-d / (a^2 + b^2 + c^2) (a, b, c)$  이다. 그런데  $d = 0$  인 경우가 있을 수 있는데 이때는 이미 (a, b, c) 벡터가 원점에서 출발하는 방향의 법선벡터이므로 그대로 사용하면 된다.

이제 각 기본면들에 포함된 모서리들로부터 루프를 생성한다. 먼저 임의의 시작 모서리를 선택하고 모서리의 방향을 정한다. 여기서 모서리는 두 방향을 모두 가질 수 있는

데 임의의 방향으로 정하면 된다. 다음으로 이 모서리의 끝 부분과 만나는 모서리를 찾아 루프에 첨가시키며 전진한다. 계속 진행하다 처음 출발한 모서리로 다시 돌아오면 루프가 완성되게 된다. 이 과정에서 전진할 수 있는 갈래가 두가지 이상일 경우가 발생할 수 있는데, 이 때는 시계방향으로의 회전각(turning angle)이 가장 작은 모서리를 택한다. 회전각은 현재의 모서리로부터 다음 모서리로 진행할 때 회전되는 각을 말한다.

모서리가 원호 성분일 때에는 진행방향으로 연결하여 주고 회전각은 꼭지점들을 이어 직선으로 취급하여 구할 수 있다. 또한 해당 모서리가 완전한 원일 경우에는 독립되어 있기 때문에 그 모서리 자체가 하나의 루프를 생성한다. 그리고 하나의 루프가 완성되면 루프를 분류하는데, 이는 회전각의 합을 비교하여 분류한다. 회전각의 합이  $+360$ 도이면 내부영역을 경계짓는 in-loop로 분류하고  $-360$ 도이면 외부영역을 경계짓는 out-loop로 분류한다. 모든 루프들이 양방향으로 사용될 때까지 위의 과정을 반복한다. 그림 9에서 이 과정을 설명한다.

각 기본면에서 루프가 구성되면 루프들간의 포함관계를 조사하여 경계면을 찾는다. 먼저 어떤 in-loop가 out-loop에 포함되는가의 여부를 판단하게 되는데, 이는 해당되는 루프들이 모서리를 공유하는 여부에 따라서 분류한다. In-loop는 내부영역을 경계짓고 out-loop는 외부영역을 경계짓기 때문에 모서리를 공유하면 그때의 in-loop는 out-loop에 포함된다. 또한 어떤 out-loop가 어떤 in-loop에 대한 포함관계를 확인한다. 이는 어떤 점이 다각형의 내부점인지 외부점인지를 결정하는

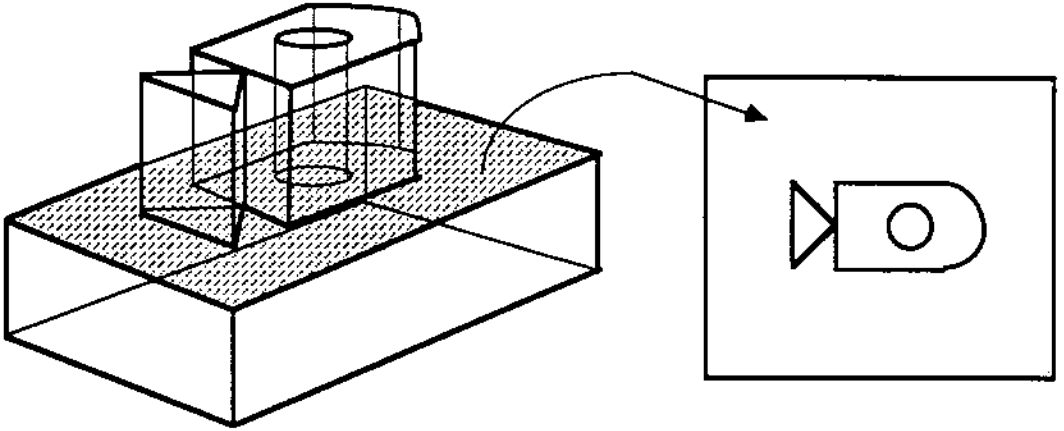
방법과 같이 결정한다. 대상이 되는 점으로부터 임의의 방향으로 직선을 발생시켜서 다각형과 만나는 점의 갯수를 비교하여, 홀수개이면 내부점이고 짝수개이면 외부점이다. 이 방법을 이용하여 out-loop의 모든 꼭지점들이 in-loop의 내부점으로 판명되면 포함관계가 있는 것으로 판정한다.

루프의 포함관계가 형성되면 이로부터 경계면을 추출한다. 이는 in-loop로 바깥경계를 설정하고 out-loop들로 안쪽경계를 설정함으로써 얻어진다. 그림 10에 예를 보인다. 여기서 구한 경계면이 3차원 형상의 후보면이 된다.

### 3.4 솔리드 생성

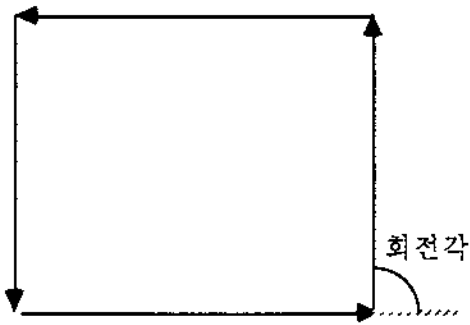
이제 위에서 찾아진 경계면들을 조합하여 sub-objects를 생성한다. 이는 기본면(surface)에서 루프를 형성하는 방법을 3차원으로 확장시킨 방법을 따른다. 먼저 임의의 면을 선택하고 법선벡터(2가지 중에서)를 선택하여 출발면으로 설정한다. 다음으로 이웃하는 면을 첨가시키는데, 이는 모서리를 공유하는 면들로부터 얻어진다. 모서리 하나에 이웃하는 면이 둘 이상일 때에는 루프생성시 회전각과 같은 표준이 필요한데 여기서는 Gujar등[11]이 제시한 표 1과 같이 얻어지는 connecting angle을 사용한다.

즉, 이웃면이 두개 이상일 경우에는 connecting angle이 작은 면을 택한다. 그런데 Gujar 등이 고려한 것은 평면과 평면이 이웃할 경우에만 해당한다. 본 연구에서는 곡면을 포함하기 때문에 아래와 같이 곡면을 처리하였다. 이웃하는 면이 원기둥, 구, 원환체, 원뿔의 일부이면 평면으로 근사하여 connecting

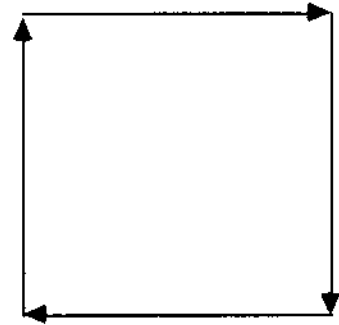


(a) 입체

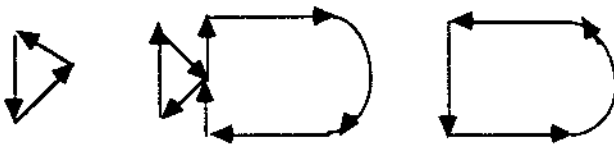
(b) 평면위의 모서리들



1. in-loop



2. out-loop



3. in-loop

4. out-loop

5. in-loop

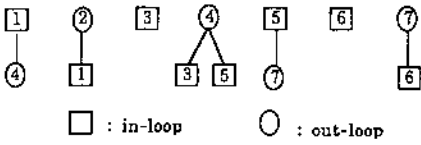


6. in-loop

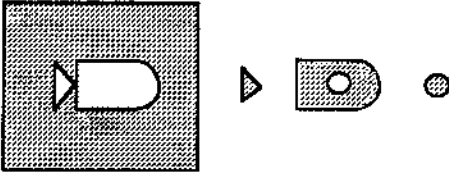
7. out-loop

(c) 루프 형성

그림 9. 루프생성



(a) 루프간의 포함 관계



(b) 경계면 추출

그림 10. 유한면 추출

angle을 표 1과 같이 구하고 완전한 원기둥 등의 경우에 대해서는 두개 이상이 동시에 이웃하지 않는다는 가정하에 그대로 이웃하는 면으로 선택한다. 이 과정을 수행하는 도

것이다. 그러면 다음으로 아직 선택되지 않은 면을 선택하여 새로운 sub-object를 생성한다. 이와같은 과정을 모든면이 sub-object들에 두번 사용될 때까지 반복한다. 그림 11에 단면도로부터 얻어지는 sub-objects의 예를 보인다. 여기에서 무한부피(infinite volume)를 가지는 하나의 sub-object가 첨가되는데 이는 나타내지 않았다. 무한부피를 가지는 sub-object는 궁극적으로 구하고자 하는 3차원 입체 형상에 고려할 필요가 없기 때문에 다음 단계로 접어들기 전에 sub-objects 집합에서 제거시킨다.

마지막으로 sub-objects를 조합하여 최종적인 입체를 추출하게 되는데 단면도와 비교하여 타당한 입체인지를 결정한다. sub-object의 수가 n개일 때의 가능한 조합의 수  $N = 2^n - 1$ 이다. 여기서 n이 1 증가하면 가능한 조합의

표 1. 면 연결표

기준면의 방향	o	o	o	o	s	s	s	s
두면의 법선벡터의 외적과 공유모서리와의 방향 비교	o	o	s	s	o	o	s	s
두면에서의 공통모서리의 방향	s	o	s	o	s	o	s	o
Connecting angle (a=두면이 이루는 최소각)	360-a	180-a	a	180+a	a	180+a	360-a	180-a
선택되는 면의 방향	s	o	s	o	o	s	o	s

\* o : 방향이 같음을 의미, 면의 방향일 경우는 기준법선벡터의 방향임  
 s : 방향이 반대임을 의미, 면의 방향일 경우는 기준법선벡터의 반대방향임

중에 sub-object에 포함된 모든 면의 모든 이웃면들이 모두 그 sub-object안에 포함되면 그 sub-object는 완성된다. 즉, 닫힌 입체가 되는

수는 2배가량 증가함을 알 수 있다. 따라서 가능한 조합의 수를 줄이기 위해서는 sub-object를 먼저 분류할 필요가 있다. 만약 2차

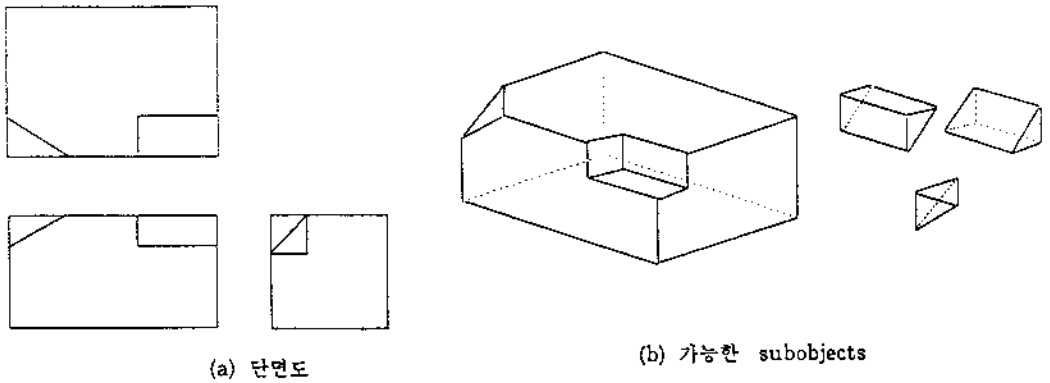


그림 11. sub-objects 생성

원 단면도의 어떤 선성분에 해당하는 3차원 모서리를 갖는 sub-object가 하나 뿐이라면 그 sub-object는 반드시 최종적인 solid에 포함되어야 한다. 이는 sub-object를 세가지 방향으로 투사시켜서 단면도와 비교하면 쉽게 판별할 수 있다. 이러한 sub-object는 모든 조합에 전부 포함되어야 하기 때문에 전체 조합수의 계산에서  $n$ 을 1 감소시킨 효과를 얻는다.

각 sub-object의 분류가 된 후에는 반드시 포함되어야 할 sub-object들을 하나의 솔리드로 결합시킨후에 세가지 방향으로 투사시켜서 삼면도와 비교한다. 모든 삼면도의 요소들이 구해진 솔리드로부터 전부 얻어질 수 있으면 이 솔리드는 최종해의 하나가 된다. 다음으로 반드시 포함되어야 할 sub-objects로 구성된 솔리드에 다른 sub-object들의 조합을 결합한 후에 삼면도와 비교하여 최종해의 여부를 결정하는 과정을 반복한다. 이렇게 하면 삼면도로부터 얻을 수 있는 가능한 모든 솔리드를 얻을 수 있다. 일단 시스템이 가능한 모든 솔리드를 제시하여 주면 설계자가 본래 의도한 솔리드를 선택할 수도 있을 것이다. 그런데 솔리드에 sub-object를 결합시

킬 때 공유하는 면이 있고 법선벡터의 방향이 반대일 때는 이 공유면은 솔리드 내부에 존재하는 가상의 면이 되기 때문에 소거시킨다.

본 연구에서 얻어지는 솔리드는 면의 종류(type)와 방향(direction) 그리고 모서리(edge), 꼭지점(vertex)들로 표현하는 B-rep (Boundary representation) 형태로 저장된다. 또한, 삼면도가 파라메트릭 설계 기법으로 설계되었기 때문에 여기에서 생성되는 솔리드는 삼면도의 형상이 수정되면 자동적으로 동시에 수정된다.

본 연구에서 사용되는 형상은 평면, 원기둥, 구, 원뿔, 원환체의 조합으로 이루어진 형상이다. 또한 원기둥, 원뿔, 원환체의 경우에는 X, Y, Z 축중에서 하나의 축과 나란하게 놓여야 한다는 제한이 있다.

#### 4. 기존도면에서의 적용

본 연구에서 개발한 시스템은 이미 존재하는 CAD도면 데이터를 입력으로 받을 수 있다. 즉, 본 연구에서 개발한 시스템에서 설계하지 않은 도면이라도 입력받아 파라메트릭

설계 기능을 부여하고 입체를 생성할 수 있다. 본 연구에서는 지금까지 도면의 호환에 표준으로 사용되고 있는 IGES 파일을 입력 받을 수 있게 하였다[19]. 기존도면의 적용을 위해서는 아래의 과정이 필요하다.

#### 4.1 전처리과정

기존의 2차원 도면 파일에는 정면도, 측면도, 평면도의 구분이 되어 있지 않다[20]. 따라서 도면을 각각의 단면도로 분류하고 2차원 점 및 선성분을 찾아야 한다. 이를 위해서 본 연구에서는 모든 점의 좌표를 비교하여 가장 큰 X좌표를 갖는 점을 측면도의 출발점으로 보고, 가장 큰 Y좌표를 갖는 점을 평면도의 출발점으로 하여 선성분을 추적하면서 평면도와 측면도를 분리한다. 이때 각 도면의 최대와 최소좌표를 함께 구한다. 다음으로 남아 있는 성분들은 분류하는데, 선성분의 좌표값이 앞에서 구한 각 도면의 최소와 최대값 사이에 있으면 그 도면에 포함시키고 그렇지 않으면 정면도로 분류한다. 각각의 도면으로 분리된 후에는 새로운 좌표계를 설정하여 2차원 점 및 선성분을 정의한다. 이때에도 각단면도들간에 공통축이 있기 때문에, 좌표계를 일관성있게 설정하여야 한다. 따라서 앞에서 구한 점의 최대, 최소값 이외에 곡선위의 최대, 최소값도 구하여 함께 고려하여야 한다.

#### 4.2 구속조건 부여

기존의 2차원 도면 파일에는 치수(dimension)도 포함되어 있는 것이 일반적이다. 본 시스템에서는 치수를 해석하여 형상과의 관계를 규정하는데 이는 IGES 파일의 group

codc를 참고하면 알 수 있다[19]. 이렇게 해석된 치수는 앞에서 설명한 방식처럼 구속조건을 형성하게 된다. 또한 도면간의 관계를 구속시키기 위하여 가상의 보조선을 만들게 되는데 본 연구에서는 단면도 중에서 원호 성분에서 보조선을 먼저 발생시키고 남은 형상요소를 위하여서는 정면도에서 출발하는 보조선과 평면도에서 출발하는 보조선 순서로 보조선을 만들고 그에 따르는 구속조건을 부여한다. 즉, 3장에서는 보조선을 이용하여 설계를 하였고 그 과정에서 구속조건을 생성하였는데, 여기에서는 이미 존재하는 설계위에 보조선을 삽입시켜 구속조건을 부여하는 것이다. 그런데 대부분의 도면에서는 주어진 치수만으로는 형상을 모두 구속하여 정의할 수 없다. 따라서 본 연구에서는 수평과 수직인 직선의 방향을 고정하여 구속조건을 늘려주는 방식을 택하였다. 보통의 경우 수직, 수평 직선의 방향은 형상의 변화에 따라 변하지 않는 성질이기에 이 방식이 과도한 제약 조건은 아니다. 그리고 원과 직선이 접할 경우처럼 접선관계가 있을 경우에는 접선조건을 확인하여 구속조건으로 부여한다. 이런 과정을 거쳐서 찾을 수 있는 구속조건들을 이용하여 형상요소를 추론한 후에도 형상이 완전히 정의되지 않으면 정의되지 않은 형상요소를 화면에 표시하여 설계자가 구속조건을 부여할 수 있게 한다.

#### 4.3 솔리드 생성 및 수정

삼면도로부터 입체를 생성하는 것은 3장에서 설명한 내용과 동일하다. 위 4.2절의 과정을 통하여 삼면도가 완전히 구속되어 정의되면 입체의 수정은 쉽게 이루어진다. 그렇지



만 파라메트릭 설계 기능이 완전히 주어지지 않는 삼면도라 하여도 입체 형상은 생성할 수 있다. 즉, 파라메트릭 설계 기능과는 별도로 입체 형상을 인식하는 작업은 수행될 수 있다. 하지만 이런 경우의 형상의 수정은 기존의 파라메트릭 설계 기능이 부여되지 않은 CAD시스템에서 수정하는 것과 같이 어렵게 된다.

### 5. 시스템 구현 및 결과

본 연구에서 개발한 시스템은 C언어로 프로그래밍 되었으며, Sun 10상의 X11윈도우 시스템 환경하에서 운용된다. 시스템은 user interface, 설계를 위한 그래픽 기능, 파라메트릭 설계 기능, 입체 생성 기능, IGES 파일 처리 기능을 위한 모듈로 구성되어 있다. 그림 12에서는 d14, d16, d18, d20, r1, r12, r13 등의 변화에 따른 형상의 수정을 보인다. 그림 13에는 CAM-I 자료교환의 benchmark로 사용되는 형상[12]의 생성 및 수정을 보인다. 본 연구에서 제시한 휴리스틱을 적용한 결과 추론시간은 그림 12는 112초, 그림 13은 359초가 소요되었다. 하지만 한번 추론된 형상의 수정은 실시간에 이루어진다.

### 6. 결론

본 연구에서는 기계부품의 설계에 보편적으로 사용하는 삼면도를 이용한 3차원 물체의 파라메트릭 설계 방법을 제시하였다. 파라메트릭 설계 기법을 구현하기 위하여 인공지능 기법을 이용하였고 이에 따르는 규칙들을 발견하여 rule-base를 구성하였다. 또한 인

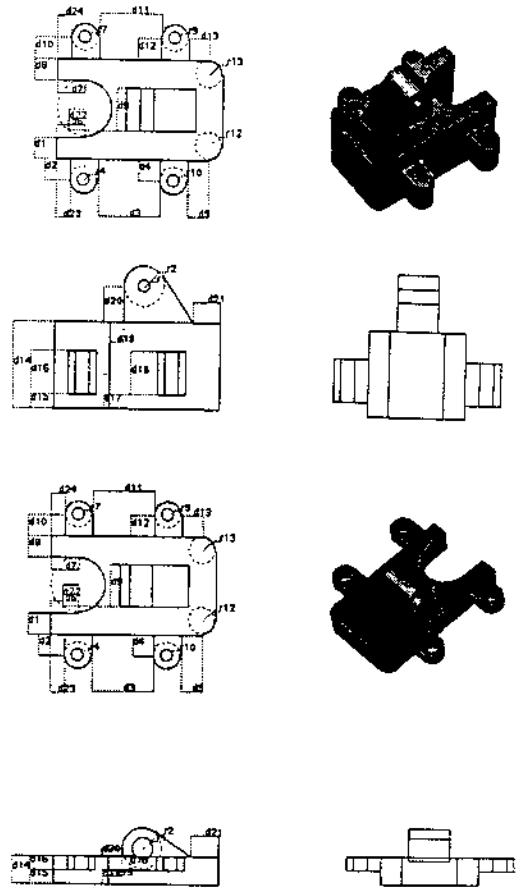


그림 12. 실행 예(1)

공지능 기법의 문제점인 추론시간을 줄이기 위한 2가지의 휴리스틱을 제시하였다. 그리고 파라메트릭 설계 기능을 가진 삼면도를 효과적으로 설계할 수 있도록 보조선을 도입하였고, 그와 관련된 규칙들을 발견하여 rule-base에 포함시켰다. 삼면도로부터 솔리드를 생성하기 위하여는 기하학적이고 단계적인 접근방법을 이용하였는데 실린더, 구 등의 곡면을 포함할 수 있게 하였다. 본 연구에서 제시한 방법은 기존의 설계방법을 그대로 사용

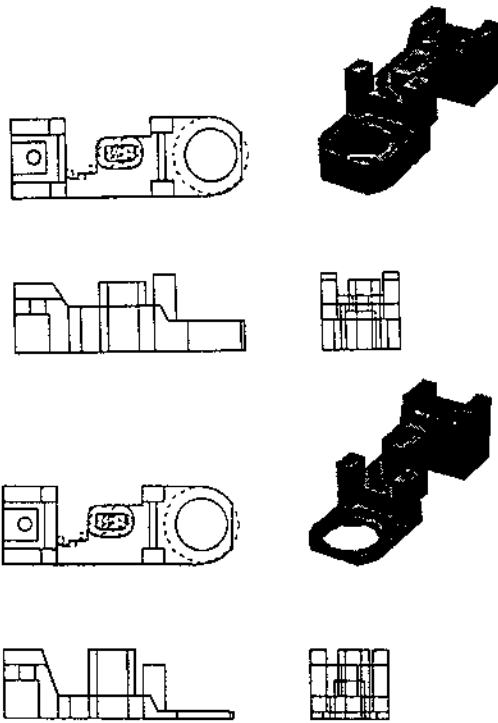


그림 13. 실행 예(2)

하기 때문에 입체 모델러의 user interface로 사용하기에 편리하다. 또한 파라메트릭 설계 기능이 있기 때문에 솔리드의 수정을 쉽게 할 수 있다. 이는 제품 설계시 잦은 수정이 필요하다는 것을 고려할 때 대단히 유용한 기능으로 판단할 수 있다. 또한 본 연구에서 개발한 시스템은 기존의 2차원 도면을 입체 모델로 바꾸어 줄 수 있을 뿐 아니라 파라메트릭 설계 기능을 첨가시킬 수 있다.

추후 연구 과제로는 1) 삼면도에 부여된 구속조건을 솔리드 생성 이후에는 솔리드에 부여하여 솔리드에서 직접 형상을 수정할 수 있도록 하는 방법에 관한 연구와, 2) 형상을 구속하기 위한 추론시간을 더욱 줄일 수 있

도록 backward chaining과의 접목등을 고려한 알고리즘 개발에 관한 연구, 3) 생산계획을 위한 특징형상인식(feature recognition), 4) 활용되는 도면에는 3면도 이외에 단면도들이 많이 포함되어 있는데 이에 관한 처리방안, 5) 기존도면의 적용부분에 있어 구속조건 부여시 치수와 수평/수직조건, 접선조건 외에도 평형조건, 대칭조건 등의 처리, 6) 모든 치수 (dimension)를 취급할 수 있도록 user interface 와 rule-base의 확장 등을 들 수 있다.

### 참 고 문 헌

- [1] Light R. and Gossard D., "Modification of geometric models through variational geometry", *Computer Aided Design*, Vol. 14, No. 4, pp. 209-214, 1982
- [2] Aldefeld B., "Variation of Geometries Based on a Geometric-Reasoning Method", *Computer -Aided Design*, Vol. 20, No. 3, pp. 117-126, 1988
- [3] Aldefeld B., Malberg, H., Richer, H. and Voss, K., "Rule-Based Variational Geometry in Computer-Aided Design", in *Artificial Intelligence in Design*, D.T.Pham (Ed.), 1991
- [4] Roller D., Schonek F. and Verroust A., "Dimension-Driven Geometry in CAD: a Survey", in *Theory and Practice of Geometric Modelling*, Springer-Verlag, 1989
- [5] Roller D., "Advanced Methods for Parametric Design", in *Geometric Modeling Methods and Applications*, Springer-Verlag, 1991

- [6] Verroust A., Schonek F. and Roller D., "Rule-oriented method for parametrized computer-aided design", *Computer-aided Design*, Vol. 24, No. 10, pp. 531-540, 1992
- [7] Sunde G., "Specification of Shape by Dimensions and Other Geometric Constraints", in *Geometric Modeling for CAD Applications*, Elsevier, 1988
- [8] Suzuki H., Ando H. & Kimura F., "Geometric Constraints and Reasoning for Geometrical CAD Systems", *Computers & Graphics*, Vol. 14, No. 2, pp. 211-224, 1990
- [9] 김진호, "Parametric CAD 시스템의 개발에 관한 연구", 포항공과대학 석사학위논문, 1993
- [10] Markowsky G. and Wesley M. A., "Fleshing Out Wire Frames", *IBM J. Res. Dev.*, Vol. 24, No. 6, pp. 934-954, 1981
- [11] Gujar U. G., Nagenda I. V., "Construction of 3D Solid Objects From Orthographic Views", *Computers & Graphics*, Vol. 13, No. 4, pp. 503-521, 1989
- [12] Lequette R., "Automatic Construction of Curvilinear Solids From Wireframe Views", *Computer-Aided Design*, Vol. 20, No. 4, pp. 171-180, 1988
- [13] Courter S. M., "Automated Conversion of Curvilinear Wire-Frame Models to Surface Boundary Models: A Topological Approach", *SIGGRAPH*, Vol. 20, No. 4, pp. 171-178, 1986
- [14] Ganter M. A. and Uicker, J. J., "From Wire-Frame to Solid Geometric: Automated Conversion of Data Representations", *Computer Mechanical Engineering*, Vol. 2, No. 2, pp. 40-45, 1983
- [15] Wangm W, Grinstein G. G., "A Survey of 3D Solid Reconstruction From 2D Projection Line Drawings", *Computer Graphics forum*, Vol. 12, No. 2, pp. 137-158, 1993
- [16] Nagendra I. V. and Gujar U. G., "2-D Objects From 2-D Orthographic Views - A Survey", *Computers & Graphics*, Vol. 12, No. 1, pp. 111-114, 1988
- [17] Rich E. and Kevin K., *Artificial intelligence*, McGraw Hill, Singapore, 1991
- [18] Bowyer A. and Woodwark J., *A Programmer's Geometry*, Butterworths, London, 1985
- [19] AutoCAD IGES Interface Specifications (Release 11), Autodesk Inc., 1990
- [20] Meeran S. and Pratt M. J., "Automated Feature Recognition From 2D Drawings", *Computer-Aided Design*, Vol. 25, No. 1, pp. 7-17, 1993
- [21] Roller, D. "An approach to computer-aided parametric design" *Computer-Aided Design*, Vol. 23, No. 5, pp. 385-391, 1991