

형상인식을 이용한 정사영 도면의 3차원 모델링에 관한 연구

이 석 희*, 반 갑 수**, 이 형 국**

Formulating 3-dimensional modeling from the orthographic projection drawing using feature recognition technique.

Seok-Hee Lee*, Kab-Soo Bahn**, Hyoung-Kook Lee**

ABSTRACT

In CAD/CAM system, it is required to produce manufacturing information from the drawing output of design system. The most difficult task is to formulate 3-dimensional modeling information utilizing 2-dimensional data. This paper addresses the automatic converting steps of 2-dimensional drawing data to 3-dimensional solid modeling using feature recognition rules as an expert shell. With the standardization of design process and recognition rule as a fundamental steps, the developed system shows a good application tool which can interface the design and manufacturing stage in CAD/CAM system on PC level.

1. 서 론

CAD 시스템에서 작성된 부품도를 가공정보로 사용하기 위해서는 부품의 형상을 인식하여 가공데이터로 변환해 주어야 한다. 따라서 CAD/CAM 시스템에서 도면정보를 가공정보로 변환하는 일련의 공정을 자동화하기 위한 연구가 필요하다. 도면정보를 가공정보로의 자동변환을 위해서는 CAD 시스템에서 작성된 도면을 인식할 수 있어야 하며, 이 형상인식의 결과에 의해 생성된 형상인식 데이터를 이용하여 가공정보를 만든다. 따라서 형상인식 과정이 필요한데, 2차원으로 표현된 투영도에서 3차원으로 정의되는 기계부품의 형상으로 재구성하는 방법은 2가지의 주된 접근방법으로 분류할 수 있다. (1)

- (1) 와이어프레임 지향의 접근방법(wireframe-oriented approach).
- (2) 볼륨 지향의 접근방법(volume-oriented app-

roach).

첫번째 방법은 Idesawa (2), Preiss (3-4), Yoshiura (5), Bin (6), Gujar와 Nagendra (7) 등으로 이어지면서 연구되었고, 두번째 방법은 Aldefeld (8), Chen과 Perng (9) 등으로 이어지면서 계속적인 연구가 이루어졌다. 두가지 방법이 추구하는 목적은 3차원 와이어프레임이나 솔리드모델링을 구성하게 되는데 이들의 활용도에 대해서 보편타당한 방향과 실용적인 시스템으로 인정받고 있는 시스템은 없다. (10) 설계자가 설계한 사양을 보편타당한 방법으로 정확히 표현하기 어렵고, 표현된 도면을 인식해야하는 다음공정의 전문가는 설계자의 정확한 의도를 파악하는 것이 쉽지않기 때문에 도면상에 필요한 방법으로 설계의사를 표현한다. 이때 정확한 설계자의 의도를 파악하는 것은 매우 어려우므로 설계의사 표현의 표준화가 필요하다.

본 연구에서는 사출금형에서 성형부를 제외한 설치판,

* 부산대학교 생산기계공학과 기계기술연구소 연구원

** 부산대학교 대학원

형판등 각종 판류의 형상을 전문가의 인식방법에 근접한 인식방법으로 인식한다. 인식대상이 되는 형상을 특징형상으로 정의하여 지식베이스화 하고 특징형상과 일치하는 형상들 즉 형상인식 끝난 형상들을 하나씩 소개하는 방법으로 형상인식을 수행한다. 인식된 결과는 CSG 방식에 의해 3차원 모델링을 함으로써, 형상인식이 정확하게 되었는지 확인할 수 있고, CAD/CAM의 일관된 자동화를 추구하여 NC정보, CAD/CAPP의 통합화로 활용할 수 있도록 한다. 이 분야에서는 주로 CAD 시스템에서 다른 프로그램과 도면정보 교환을 위하여 제공하는 DXF(Drawing Interchange)나 IGES(Initial Graphics Exchange Specification) 데이터 파일을 이용하여 형상인식을 하고 있으나, (11-12) 여기서는 CAD 형상데이터를 사용하여 화면상에서 직접 형상인식을 수행한다. 화면상에서 직접 인식하는 방법은 인식전문가의 시각적인 기능에 해당되는데 인식규칙을 보다 효과적으로 구축할 수 있으며, 관심이 있는 단면도나 임의영역에 있는 형상요소를 용이하게 검색하여 정보를 얻을 수 있다. 만일 2차원으로 표현된 투상도가 제도규격(13)에 의해 정확하게 제도(drawing)되지 않았다면, 인식의 곤란함을 제거하기 위해 전처리(pre-process) 과정을 통하여 도면을 수정할 수 있다. DXF나 IGES방식의 데이터 파일을 이용하는 경우 형상인식 규칙을 적용하기 위해서는 데이터의 조작을 통한 중간파일로의 변환이 필요하나, 여기서는 CAD 형상데이터를 이용함으로써 데이터의 변환이 불필요하다. 형상인식을 통한 3차원 모델링은 육면체, 구멍, 슬롯, 포켓, 카운터보어(Counterbore), 카운터싱크(Countersink) 등 주로 2.5D에 가까운 3D에 대해 수행하였다.

2. 시스템의 구성 및 데이터

개발된 시스템은 AutoCAD Drawing Editor(이하 ADE)에서 수행된다. ADE에서 작성된 모든 엔티티(Entities)들에 대한 정보는 Auto CAD Drawing Database(이하 ADD)에 저장되어 있다. ADD는 텍스트(Text)와 수치로 기술된 레코드(record)들로 구성되어 있다. 또한 설계도면을 비 그래픽적으로 기술하기 위해 필요하며 필요한 정보를 사용자가 액세스(Access)할 수 있도록 구성되어 있다.

2-1. 시스템의 구성도 및 흐름도

CAD 시스템에서 내부 데이터베이스에 의해서 표현되는 엔티티들은 각각의 속성이 있는데 그것들은 선의 종류(Line type), 색(Color), 층(Layer) 등으로 분류할 수 있다. 내부 데이터베이스는 필요에 따라 외부 데이터베이스로 변환하여 활용하며, 타 CAD 시스템과의 연결을 위해서 사용되기도 한다. 3차원의 형상은 보는 관점에 따라 다르게 표현되며 경우에 따라서는 최소한의 View로서 표현이 가능하도록 2차원으로 표현되는 투영도를 사용하기도 한다. 또한 형상과 관련된 지식베이스(K/B)를 이용하여 형상을 인식하며 인식가능한 기공형상들의 위상기하학(Topolgy)적인 기법을 이용하여 복합적인 형상을 생성한다. 그림1은 이러한 일련의 과정을 도식적으로 나타낸 것이다.

2.2 도면 요소(Entities)의 분류

CAD상에서 형상인식에 가장 많이 사용되는 엔티티는 선, 원호, 원 등이 대부분이다. 문자, 속성(Attribute) 및 치수(Dimension)도 사용되는 도면 요소들이다. 각

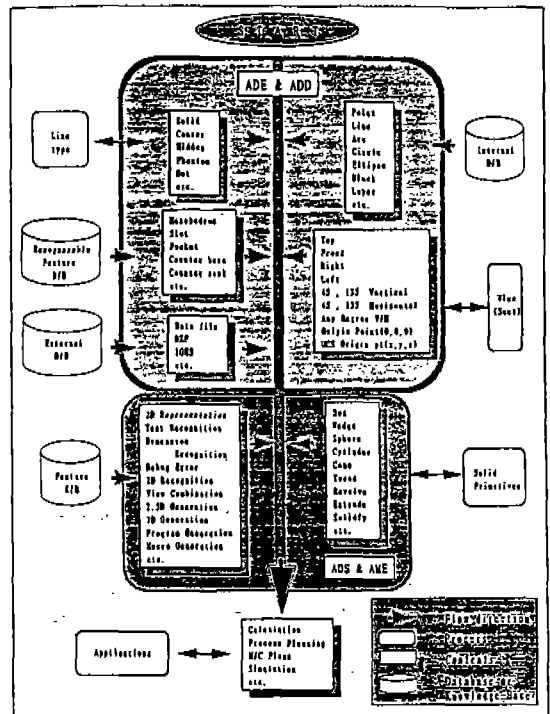


Fig.1 The process of modeling and system flow chart.

Table 1. Entity Classification of AutoCAD Database.

NO.	속 성 명	기 능	해당 Entity	변 수 타 입
1	e_type	각종 Entity의 형태	L, A, C, B	String
2	b_name	Block의 이름	B	String
3	L name	Layer 이름	L, A, C, B	String
4	s_point	각 Entity의 시작점	L, B	Point
5	e_Point	각 Entity의 끝점	L	point
6	radius	반경	A, C	Real
7	s_p	Arc의 시작점	A	Point
8	e_p	Arc의 끝점	A	Point
9	s_ang	Arc의 시작각	A	Real
10	e_ang	Arc의 끝각	A	Real

약식표기 : L(ine), A(rc), C(ircle), B(lock)

각의 도면 요소에 대하여 색상 및 선의 형태를 지정할 수 있다. 형상을 분류하여 정리한 것을 표1에 나타내었는데, 직선의 정보로는 시작점과 끝점만 알면 다른 정보를 알 수가 있으며 원에 대한 정보는 중점의 좌표와 지름 혹은 반지름, 원호에 대한 정보로는 시작각, 끝각, 중심, 지름 혹은 반지름이 필요하며 경우에 따라서는 시작점, 끝점도 알아야 될 필요가 있다. 시스템에 따라서는 이 점들에 대한 정보가 없기 때문에 주어진 데이터베이스(D/B)에서 필요한 정보를 생성해 내어야 한다.

2.3 ADD의 텍스트에 대한 정보

CAD화면상에서 직접적으로 형상인식을 할 경우에 엔티티들의 정보를 충분히 활용하지 않으면 관련 K/B를 효과적으로 운용하는 것이 용이하지 않다. 예를들어 그림5(c)와 같이 카운터보어의 단면정보 대신에 텍스트로 표현되어 있는 경우에 텍스트를 인식하여 단면정보를 추출해야 한다. 텍스트 기재 방식은 설계자에 따라 표현방식이 다를 수 있으므로 표현방식의 표준화가 필요하지만 인식은 가능하다. ADD에 기재된 텍스트의 내용을 인식하면 형상의 정보를 알 수 있고 텍스트의 시작점과, 방향벡터등도 알 수 있다.

3. 특징형상 인식시스템

3-1. 특징형상의 분류

2차원 도면에서 인식되어 3차원 형상의 모델링에서 기본형상(Primitive)으로 사용될 수 있는 가공형상

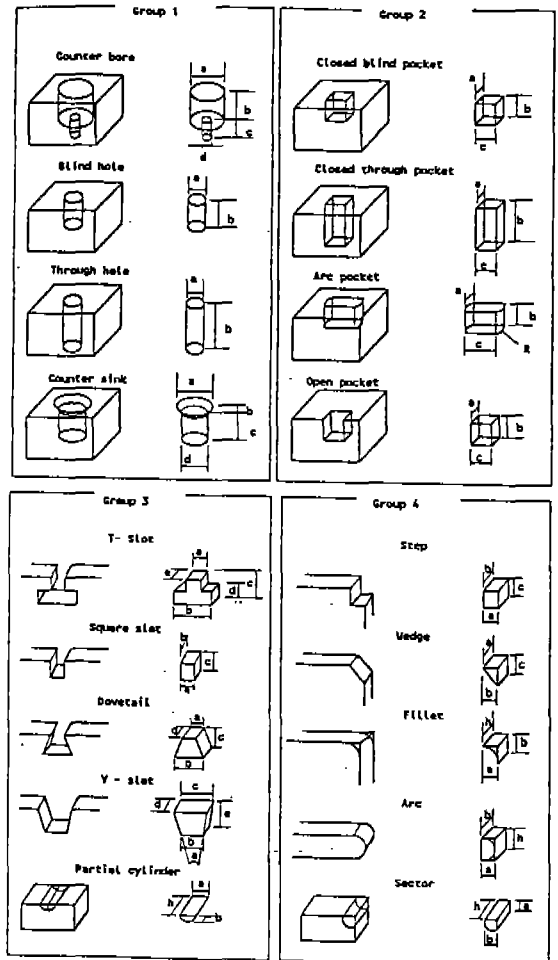


Fig. 2 Recognizable Types of Machining Features

(Machining Feature)들 중에서 인식이 가능한 형상들의 대표적인 예를 그림2에 나타내었다. (14) 그림2의 형상들은 직육면체의 원재료(Raw Material)에서 절삭되어지는 부분이며, 형상을 정의하기위해 필요한 수치데이터 및 위치데이터를 문자로 변수화 한 것이다. 금형에서 일반적으로 사용되는 형상들은 구멍류의 경우에 Group 1의 형상과 이들이 조합된 형태로 나타나는 것이 대부분이며, 밀링류의 경우 Group 2의 포켓류는 단면이 일정하게 변하는 형상이 대부분이며, 성형부 등에서는 계단형태나 경사형태로 나타나는 경우도 있는데 이들의 인식도 비슷한 방법으로 가능하다. Group 3의 슬롯에 대한 형상들은 단면의 형상정보로 인식이 가능하며, Group 4의 형상들은 Group 3의 형상들의 인식방법과 유사하다. 그림 3은 대상이 되는 특징형상들을 유사한 특징으로 그룹화하고 형상의 기하학적, 위상학적인 관계를 이용하여 형상을 분류하였다.

- Group 1 : Hole의 유사형상
- Group 2 : Pocket의 유사형상
- Group 3 : Slot의 유사형상
- Group 4 : Fillet의 유사형상

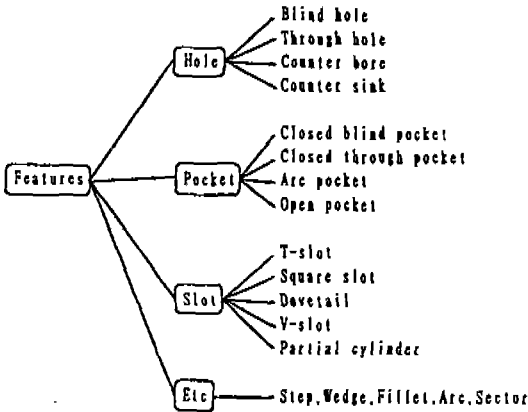


Fig. 3 Classification of features

3.2 좌표변환 및 방향벡터

ADE에서 X, Y 평면에 3각법으로 그려진 2차원 도면을 각 투상도별로 좌표변환을 수행한다. 그림 4에서와 같이 기준점(Reference point) P를 중심으로 X좌표값이 기준점보다 큰 도면 요소는 평면도이다. 그리고 기준점보다 X, Y좌표값이 작은 도면 요소는 정면도이다. 이와같은 방법에 의해서 분리된 각 투상도에 새로운 좌

표를 부여한다. 즉 정면도의 각 도면요소들은 x, z 좌표를 가지며 평면도는 x, y 좌표를 그리고 우측면도는 y, z 좌표를 가진다.

- p : reference point
- X, Y : ADE상의 절대좌표
- x, y, z : 투상도별로 변환한 좌표

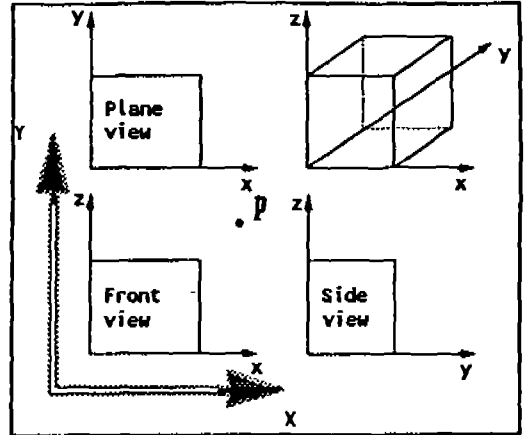


Fig. 4 Coordinate Transformation

형상인식의 과정에서 임의의 평면에 수직인 방향벡터를 알아야 될 필요가 있다. 이 벡터는 직선과 평면과의 관계를 이용하여 구할 수 있다.

L을 공간상에 존재하는 직선이고 주어진 임의의 점P1(x1, y1, z1)을 통과하며 주어진 공간상에 제로 아닌(nonzero) 벡터v에 평행한 직선이라고 가정하면 L은 식(1)과 같이 표현된다.

$$\frac{P - P_1}{v} = t \tag{1}$$

여기서, v=Ai+Bj+Ck이고 t는 Scalar량이다. 식(1)을 성분별로 구분하면

$$\begin{aligned} x - x_1 &= tA, y - y_1 \\ &= tB, z - z_1 = tC \end{aligned} \tag{2}$$

식(2)가 된다.

평면에 대한 방정식을 얻기 위하여 평면에 존재하는 하나의 점 P1(x1, y1, z1)을 가정하고 제로 아닌 벡터를 N이라하면 N은 식(3)과 같이 표현된다.

$$N = Ai + Bj + Ck \tag{3}$$

점 P가 같은 평면상에 존재하고 N이 p_{1p}에 수직일 때 다음식이 성립한다. $N \cdot p_{1p} = 0$ 혹은 $A(X-X_1) + B(y-y_1) + C(Z-Z_1) = 0$ 이 식은 다음과 같이 표현된다.

$$Ax + By + Cz = D \quad (4)$$

식(4)는 N 벡터에 수직인 평면을 표현하는 방정식이다. 그러므로 평면에 수직인 벡터를 알고 평면상에 임의의 한 점을 알면 평면의 방정식을 알 수 있다. 수직 벡터는 경사진 면에 존재하는 임의의 형상을 인식하여 모델링할 때 기준점에서 형상의 존재하는 방향을 수직벡터로 하여 형상을 구성한다. 금형에서 슬라이드 코어 등은 성형부의 형상에 따라서 위치나 각도가 달라지므로 인식이 보다 난해하다. 이런 경우에 식 (4)는 유용하게 활용된다.

3.3 특징형상의 데이터 구조

특징형상의 인식결과를 데이터의 항목으로 나타내어 대표하는 약자, 방향, 형상정보 데이터로 표현하면 다음과 같다.

- 특징형상의 구분 : 형상의 특징을 의미하는 영문자의 약자를 사용한다.

예) OR, BO, HO, CB, CS, AP...

- 방향 : 음수와 양수 1, 2, 3을 사용하여 표시한다.

+x방향 : +1, -x방향 : -1

+y방향 : +2, -y방향 : -2

+z방향 : +3, -z방향 : -3

- 특징형상의 데이터구조의 예

카운터보어의 형상을 특징적으로 분류하여 표시한 데이터는 다음과 같은 구조를 갖고 기억공간에 하나의 레코드(Record) 형식으로 저장된다.

```
{struct Counter-bore}
ads-point originp; // 원점
ads-real radius1; // 넓은 폭의 반경
ads-real radius2; // 좁은 폭의 반경
ads-real depth1; // 넓은 폭의 길이
ads-real depth2; // 좁은 폭의 길이
int direct; //방향
```

3.4 형상의 인식 알고리즘

형상인식을 수행하기 위해서는 먼저 기본적인 형상들에 대하여 형상특징을 정의해 놓고, 이 정의에 따라 형상인식규칙을 만든 후, 3각법에 의해서 2차원으로

로 그려진 투상도에서 형상을 인식하기 위해서는 각 투상도에 대한 좌표변환을 수행하여야 한다. 형상인식은 두가지의 특징으로 이루어지는데 하나는 도면의 전체적인 형상이 되는 인식이고 다른 하나는 특징형상에 대한 인식이다. 도면전체의 인식목적은 3면도로 구성된 도면은 그림 5(a)와 같이 평면도의 형상에 대해서 단면도가 정면도나 측면도의 어느쪽에 위치하는가를 알아야 한다. 특히 복잡하게 구성된 도면의 인식에서는 엔티티의 관리가 복잡하므로 같은 형상이 여러개 존재할 때 효과적인 관리를 할 수 있다.

3.4.1 전체도면의 인식 알고리즘

(1) 전체도면에서 3면도(Top view, Front view, Side view)의 영역을 분리한다. 이때 인식의 편의를 위해서 3면도의 분리점을 입력하는 방식과 입력하지 않으면 도면의 최대, 최소 영역과 경계점을 인식하여 분리한다. (View-Box[0] : Top, View-Box[1] : Front, View-Box[2] : Side)

(2) 평면도를 검색하여 서로 다른 부품의 Block name를 구분하여 배열에 넣어 관리한다. (Block_names[i], i=1, n)

(3) 단계(2)에서 같은 형상이 여러개 존재하면 Block_names[i]의 i와 삽입점(insert point)을 관련시켜 파일구조의 요소로 출력파일(insert_list_file)에 저장한다.

(4) 나머지 View에서도 같은 방법으로 단계(2)-(3)을 수행한다. 이 공정이 완료되면 View별 Block_names[i]와 삽입점의 저장 파일이 3개 존재한다.

(5) Block_names[i]에서 Block_names[0]부터 꺼내어 해당 Block을 찾는다. 선택된 Block을 해제시키면 본래의 기본적인 엔티티들을 찾을 수 있다. 이때 해제된 엔티티는 작업영역에서 관리한다. 만일 현재 Block이 작업을 종료하면 작업영역에 있는 엔티티는 제거하고 새로운 Block의 엔티티를 넣는다.

(6) 작업영역에 있는 엔티티는 특징형상의 인식규칙에 따라 개별형상인식을 수행한다. 이때 평면도에 있는 형상의 단면형상이 정면도나 측면도에 존재한다면 평면도의 삽입점과 관련된 정면도나 측면도에 있는 Block을 점검하여 특징형상의 측면도 인식규칙으로 인식한다. 특징형상과 일치하는 특징을 갖으면 인식된 형상정보를 출력파일(Fere_list_file)에 저장하고 해당되는 Block을 전부 삭제한 후 개별형상인식은 완료한

다.

(7) 단계 (5)-(6)을 현재 View의 Block_names(i)가 끝날때까지 반복해서 수행한다. 만일 평면도와 관련된 단면도의 정보가 없다면 해당 Block_names(i)를 New_Block_names(i)에 저장하고 마지막에 처리한다.

(8) 단면도의 정보가 그림 4의 (c)와 같이 텍스트(Text)나 개별 엔티티로 존재하면 Table 2의 텍스트를 이용하거나 평면도의 인식 범위를 설정하여 인식한다.

(9) 나머지 View에 대해서도 단계 (7)-(8)을 계속한다.

(10) 잔여 엔티티가 없을 때까지 View별로 단계 (7)-(9)를 반복한다.

3.4.2 특징형상 인식의 알고리즘

● Hole의 인식규칙

Hole 형상이 포함된 투상도를 그림 5의 (a)에 나타내었다. Hole 형상의 특징은 하나의 투상도에 원이 나타나고 이와 관련된 다른 투상도에 원의 좌표와

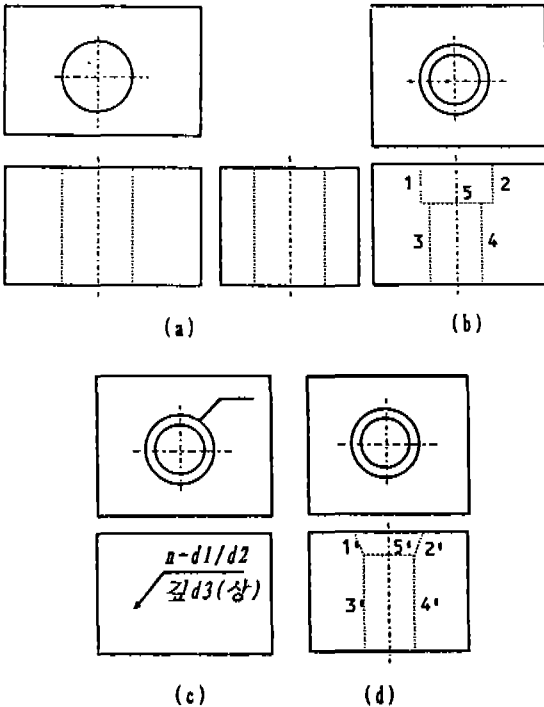


Fig.5 Projection drawing with a hole pattern

관련이 있는 좌표값을 가지는 선이 나타난다. 그림 5의 (a)에서는 평면도에 원이 있다. 따라서 이 투상도는 원이 포함되는 형상, 즉 관통된 구멍, 막힌 구멍, 카운터 보어, 카운터 싱크 등 가능한 여러가지 형상이 있을 수 있다. 이 원의 중심좌표와 지름을 ADD에서 추출한다. 이 원의 중심좌표와 같은 중심을 가지는 원이 있는가를 확인한 후 원이 검색된 다른 View에서 원의 중심좌표와 지름을 이용해서 계산된 위치에 선이 존재하는가를 확인한다.

구멍의 형상이 표시된 위치에 해당되는 엔티티가 단면도를 존재하여 조건을 만족시키면 구멍이 존재하게 된다. Fig. 6의 (a)는 Hole의 인식규칙을 나타내고 있다.

```

Circle 정보를 갖는 Entity 검색 //From internal D/B
if (존재하면 관련정보 추출) (
    해당 View 선정
    동일한 중심(동심원)을 갖는 다른 Circle 검색
if (존재하지 않으면) ( //Hole의 대상이 될
    Circle과 관련된 Line 존재여부 //다른 View에서 검색
if (원의 지름의 거리를 갖는 평행선 존재 &&
    다른 View에도 같은 조건) (
        Hole의 형상 존재
    )))
관련 D/B 저장
    
```

(a)

```

Circle 관련정보 추출
Circle 정보를 갖는 Entity 검색 //From internal D/B
if (존재하면 관련정보 추출) (
    해당 View 선정
    동일한 중심(동심원)을 갖는 다른 Circle 검색
if (존재하면) ( //Counter bore or
    //Counter sink의 대상
    Circle과 관련된 Line 존재여부 //다른 View에서 검색
if (작은 원의 지름의 거리를 갖는 평행선 존재 &&
    큰 원의 지름의 거리를 갖는 평행선이 존재 &&
    큰 원을 나타낸 연결선 상에 작은 원을 나타낸 연결선이 존재
    다른 View에도 같은 조건) (
        Counter sink 존재
    )
else (작은 원의 지름의 거리를 갖는 평행선 존재 &&
    큰 원의 지름에서 작은 원의 지름을 갖는 &&
    연결된 경사선 선이 존재 &&
    대칭된 교점이 서로 연결 &&
    다른 View에도 같은 조건) (
        Counter sink 존재
    ))
관련 D/B 저장
    
```

(b)

Fig.6 Rules for "hole", "Counter sink" and "Counterbore"

● 카운터보어(Counter bore) 및 카운터싱크(Counter sink)의 인식규칙

카운터 보어 및 카운터 싱크의 공통점은 하나의 투상도에 중심점의 좌표가 같은 두개의 동심원이 존재하느냐에 기준을 둔다. 그림 5의 (b), (d)는 카운터 보어 및 카운터 싱크가 포함된 도면을 나타낸다. 카운터 보어 및 카운터 싱크의 형상인식을 하기 위해서는 먼저 동심원이 두개 존재하는지의 여부를 검색한 후 두원의 가장자리의 좌표값을 찾은 후에 이 좌표값에 연관된 직선 1, 2, 1', 2'가 다른 투상도 상에 존재하는지 확인한다. 이 조건을 만족하면 직선 3, 4, 5 또는 3', 4', 5'를 조사한다. 카운터 보어 형상에서는 직선 1, 2의 양 끝점은 큰 원의 가장자리와 좌표가 일치하고, 그 한쪽 끝점을 직선 5가 연결하고 있으며, 직선 3, 4를 통과하게 된다. 카운터 싱크 형상에서는 직선 1', 2'의 한쪽 끝은 큰 원의 가장자리에 존재하고 다른 끝은 작은 원의 가장자리에 존재하며, 작은 원쪽의 두 가장자리를 직선 5'가 연결하고 있으며 직선 3', 4'의 각 시작점이 된다. 해당 투상도 내의 모든 Entity를 검색하여 직선 3, 4 그리고 5'의 조건을 만족하면 카운터 싱크의 형상이 존재함을 알 수 있다. 카운터 보어 및 카운터 싱크의 인식규칙은 그림 6(b)와 같다.

● 루프방식(loop) 방식에 의한 아크포켓(Arc pocket)의 인식규칙

그림 2에서 그룹 2의 포켓류 등은 평면도의 형상이 대부분 폐루프(Closed loop)를 이루고 있다. Hole종류의 인식방법과 달리 폐루프 인식방법은 루프의 형상에 따라 대부분의 특징형상이 정의되므로 단면도의 형상은 단순하다. 아크포켓을 전문가가 인식하는 방법은 서로 마주보는 대칭적인 아크(Arc)가 존재하고 양 끝점을 연결하는 선이 존재하여 그것이 폐루프를 이루면 아크포켓의 대상이 된다.

그림 8의 중앙에 존재하는 아크포켓은 검색대상이 원호와 직선과 연결된 기하학적 관계를 먼저 규명하고 이것과 관련된 단면도의 형상정보를 정면도나 측면도에서 추출하여 이미 정의된 아크포켓의 특징을 만족하면 인식을 완료한다.

● 기타 특징형상의 인식방법

그림 2의 형상 중에서 그룹 3과 그룹 4의 형상들은 단면이 일정한 2.5D의 형상에 가까우므로 임의의 View

에 존재하는 단면의 형상이 인식되면 그 형상을 단면으로 하는 기둥모양이라고 생각할 수 있으므로 인식이 비교적 용이하다. 다만 이들 형상의 존재위치와 방향을 정확히 인식해야 한다.

3.5 인식된 결과의 3차원 모델링

2차원 도면을 인식한 결과를 이용하여 3차원 모델링을 수행하기 위해서는 먼저 3차원 모델링에서 필요로 하는 데이터 형태로 데이터를 변환해야 한다. 다음으로 Advanced Modeling Extension(이하 AME)기능을 이용하여 부품형상을 3차원으로 표현한다. 2차원 도면에서 형상인식이 기본형상별로 수행되기 때문에 3차원 모델링에서도 기본형상요소의 조합에 의하여 표현되어야 한다. 기본형상의 연산에 의하여 복합 형상도 표현할 수 있다.

3.5.1 투영도에 의한 방법

앞에서 보인 예제형상처럼 각각의 View에서 나타난 형상을 형상인식 규칙에 맞는 형상특징이 존재하면, 그 형상의 속성을 데이터화하여 모델링과 관련된 K/B와 연결하여 3차원 모델링을 할 수 있다.

3.5.2 형상의 특징이름에 의한 방법

적용된 예제형상처럼 인식하고자 하는 형상의 필요조건을 모두 갖추고 있지는 않지만 같은 속성을 갖는 특징(예 형상을 그룹화한 Block)은 그 중에서 알고 있는 속성을 이용하여 나머지 모르는 형상들까지도 모델링할 수 있는 방법이다. 예를들어 그림 8에서와 같이 똑같은 부품이 여러개 존재할 때 중복 인식을 피하기 위해서 ADD상에서 특징적인 이름을 갖는 Block으로 부품을 정의하여 운용하면 같은 부품의 갯수를 알 수 있고 단면형상을 각각 따로 표현하지 않더라도 그 중에서 한개만 표현되면 나머지는 알 수 있는 잇점이 있다. 특히 금형과 같이 구조가 복잡한 경우 단면형상을 생략없이 나타내는 것은 인식의 혼돈과 비합리적인 결과를 초래하기 쉬우므로 예제도면과 같이 평면도 상에는 생략된 부품이 없이 모두 나타낸다. 측면도 상에는 도면화 되었을 때 인식의 혼돈을 피하기 위하여 적법한 공간의 배치를 고려하여 필요한 것만 표현해도 평면도의 정보로 측면도의 생략된 부품을 고려할 수 있다.

3.5.3 형상의 자동생성을 위한 AME 운용 프로그램

인식된 데이터로부터 3차원 기본형상들이 구성될 때, 3차원 모델링 프로그램에서는 AME의 API에서 제공되는 AME 기본 솔리드 생성 함수와 규칙화된 Boolean operators 함수를 사용하여 3차원 기본형상들을 생성한다. AutoCAD의 AME에서는 CSG 방식을 사용하여 모델링을 하는데 Boolean 연산은 합집합(Union), 차집합(Subtraction), 교집합을 이용하여 복합적인 형상을 생성한다. 그림 7은 본 연구에서 취급한 형상을 중심으로 Boolean 연산의 순서를 도식적으로 나타낸 것이다. 아래의 프로그램은 인식된 데이터 파일을 입력데이터로 사용하여 생성된 3차원 기본 형상들을 가공 전 원재료에 해당되는 직육면체 형상의 해당위치로 옮긴 후, 회전 및 병진을 수행하고 제거될 형상들을 제거함으로써 전체 조합형상에 대한 3차원 모델링이 완성된다. 이 공정은 자동으로 프로그램화 되는데 아래에 카운터보어 형상에 관련되는 예를 나타내었다.

3.5.4 인식된 데이터의 예

본 연구에서 취급한 각종 형상을 인식한 데이터를 예로서 나타낸 것이 아래와 같다. 여기서 점은 3차원 데이터를 갖는 것이 원칙인데 배열중에서 또다른 배열로 구분되어 있고 나머지 데이터는 점의 배열과 동등한 레벨을 갖는 형태로 파일에 저장된다. 여기서 OR은 원점을 나타내며, BO는 육면체, HO는 구멍, CB와 CS는 카운터 보어와 카운터 싱크, AP는 아크포켓을 나타낸다.

```
OR((2.0 2.0 0.0))
BO((0.0 0.0 0.0) (23.5 0.0 0.0) (0.0 12.5
0.0) (0.0 0.0 10.5))
HO((2.0 2.0 5.0) 1.0 1.0 -3)
CB((7.0 7.0 5.0) 2.0 1.0 2.0 1.0 -3)
CS((7.0 12.0 5.0) 2.0 1.0 3.0 0.5 -3)
AP((3.0 5.0 5.0) 3.0 1.0 1.5 -2)
```

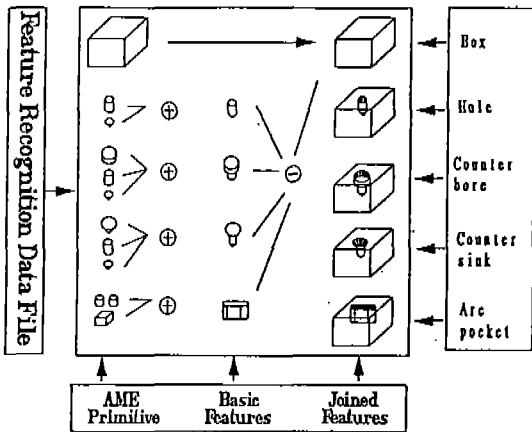


Fig.7 An input CSG tree.

```
ap_cylinder(CB_rw, CB_rw, CB_dpw, & cb(0)) :
ap_cylinder(CB_rn, CB_rn, CB_dpn, & cb(1)) :
ap_translate(0.0, 0.0, CB_dpn, cb_mat1) :
ap_assocmat(cb(0), cb_mat1) :
ap_cone(CB_rn, CB_rn, 0.57735 * CB_rn, & cb
(2)) :
ap_translate(0.0, 0.0, -0.57735 * CB_rn, cb
mat3) :
ap_assocmat(cb(2), cb_mat3) :
cb(3) = (ap_Solid) NULL :
ap_union(cb, & cb1) :
```

4. 적용사례

직육면체에 슬롯, 관통된 구멍, 카운터보어, 카운터싱크 등이 있는 금형 관류의 대표적인 형상을 ADE에서 그림 8과 같은 투상도를 그렸다. 형상인식 규칙을 적용하여 투상도에서 형상을 인식하여 형상인식 데이터를 출력하고, 형상인식된 데이터를 이용하여 ADS (AutoCAD Development System)에서 생성된 데이터에 의해 AME를 이용한 솔리드모델링을 수행한 것이

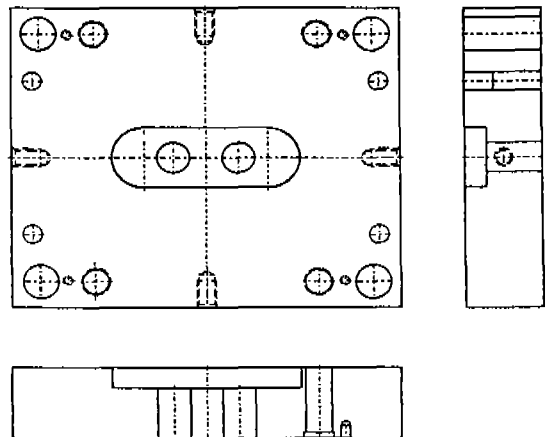
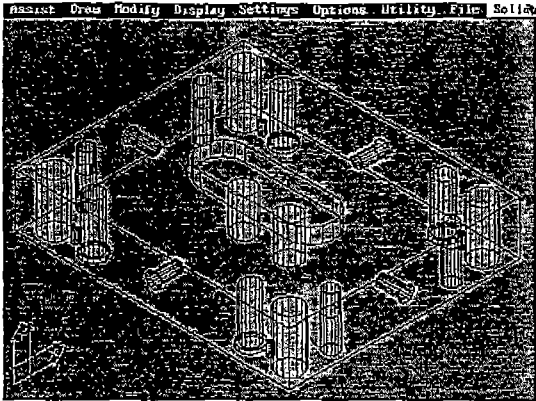
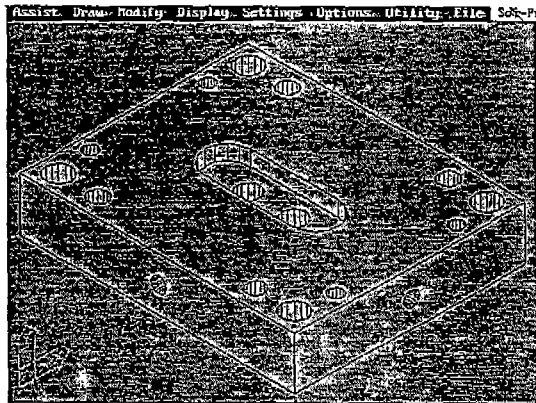


Fig.8 Example projection drawing

그림 9(a)와 같다. 그림 9(b)는 현재의 관점(View)에서 보이지 않는 부분을 숨긴모양(Hide)으로 나타내었다. 여기서 각종 해석을 위한 체적, 관성모멘트, 임의의 단면모양 등을 마음대로 계산할 수 있다. 중앙에 있는 복잡한 형상은 관통된 hole 형상과 slot 형상이 조합된 모양을 나타내고 있다.



(a)



(b)

Fig. 9 3-dimensional solid modeling

5. 결론 및 계산

본 연구에서 주로 취급한 형상은 구멍, 슬롯, 포켓, 카운터보어, 카운터싱크 등 금형의 전형적인 형상으로 비교적 인식이 용이한 형상들을 전문가의 인식규칙에 의하여 K/B화 하고 인식된 D/B와 연결, 2차원의 도면을 3차원적으로 모델링하였다. 그것에 의한 특징적인 결론은 다음과 같다.

(1) 대상이 되는 형상들은 형상특징이 명확하게 정의되어 있으므로 인식된 형상의 3차원적 표현이 정확하게 수행되었다.

(2) 기존의 연구들에 의해서 인식한 3차원 형상의 후보가 2개 이상일 때⁽¹⁾, 설계자가 원하는 정확한 형상을 자동적으로 찾아내는 것이 용이하지 않지만 본 연구의 인식방법은 전문가의 인식방법과 근사하게 적용되므로 원하는 형상을 찾는것이 용이하다.

(3) 금형의 판류와 같은 복잡한 형상을 완전하게 3면도로 제도화하는 것은 인식의 혼란을 초래하기 쉬우므로 최소한의 표현으로 도면화하는데 이런 생략된 도면에 대해서도 D/B검색 방법으로 인식하여 모델링이 가능하다.

(4) 조립도나 판류가 완성되면 복잡한 성형부를 제외한 나머지 부분을 3차원 모델링까지 자동적으로 연결시킴으로써 CAD/CAM자동 연결시스템 구축에 활용할 수 있다.

앞으로 곡면이 포함된 형상 등 다양한 형상을 인식할 수 있는 규칙을 개발함으로써 CAD/CAM 시스템의 연결을 일관성 있게 도모하고, 설계도면에서 3차원 모델링을 자동적으로 연결하여 NC가공정보, 해석, 시뮬레이션 등의 공정에 적용될 수 있도록 지속적인 연구가 필요하다.

참고문헌

1. ZEN CHEN, DER-BAAU PERNG, CHII-JANG CHEN and CHU-SONG WU. "Fast reconstruction of 3D mechanical parts from 2D orthographic views with rules", INT. J. COMPUTER INTEGRATED MANUFACTURING, Vol. 5, No. 1, pp.2~9, 1992.
2. IDESAWA, M., "A system to generate a solid figure from three-view" Bulletin of the JSME, : Japan Society of Mechanical Engineers, 16(92), pp.216~225, 1973.
3. K. Preiss, "Algorithms for Automatic Conversion of a 3-View Drawing of a Plane-Faced Part to the 3-D Representation", Computers in industry, Vol. 2, pp.133~139, 1981.

4. K. Preiss, "Constructing the solid representation from engineering projections", *Comput & Graphics*, Vol. 8, No. 4, pp.381~389, 1984.
5. Hiroshi Yoshiura, Kikuo Fujimura and Toshiyasu L. Kunii, "Top-Down Construction of 3-D Mechanical Object Shapes from Engineering Drawings", *Computers*, Vol. 12, pp.32~40, 1984.
6. Ho Bin, "Inputting constructive solid geometry representations directly from 2D orthographic engineering drawings", *Computer-Aided Design*, Vol. 18, No. 3, april, 1986.
7. 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.
8. ALDEFELD, B., "On automatic recognition of 3D structures from 2D representation" *Computer-Aided Design*, Vol. 15, No. 2, march, 1983.
9. Zen Chen and Der-Baau Perng, "Automatic reconstruction of 3D solid objects from 2D orthographic views", *Pattern Recognition*, Vol. 21, No. 5, pp.439~449, 1988.
10. 손영태, 박면용 "각주형 부품상의 가공 특징형상 인식" 대한기계학회 논문집, 제17권 제6호, pp.1412~1422, 1993.
11. Srinivaskumar S. Madural and Li Lin, "Rule-based automatic part feature extraction and recognition from CAD data", *Computers ind. Eng.*, Vol. 22, No. 1, pp.49~62, 1992.
12. RONG-KWEI LI and TAI-ZEN HWANG, "A part-feature recognition system for rotational parts-non-turning features", *INT. J. COMPUTER INTEGRATED MANUFACTURING*, Vol. 2, No. 5, pp.257~267, 1989.
13. 金淇煥, 金亨燮, 鄭善謨. 외 4명, 機械設計製圖, 淸文閣, 1986.
14. D. B. Pering, Z. Chen and R. K. Li, "Automatic 3D machining feature extraction from 3D CSG solid input", *Computer-Aided Design*, Vol. 22, No. 5, pp.285~294, June, 1990.