

CAD/CAM 인터페이스를 위한 정사영도면의  
형상인식을 이용한 3차원 모델링  
3-Dimensional Modeling using Feature Recognition Rules  
from Orthographic Views for CAD/CAM Interface

\* 정구섭      \*\* 이형국      \*\*\* 이석희  
Ku-seob Jung    Hyoung-Kook Lee    Seok-Hee Lee

\* 부산대학교 대학원  
\*\* 대우조선 (주)  
\*\*\* 부산대학교 생산기계공학과

## ABSTRACT

As a basic step of interfacing CAD and CAM, it is required to convert the drawing data into manufacturing information automatically. In this study the developed system carries out feature recognition from the orthographic views of press mold containing various pockets. Using the recognized output data, 3 dimensional model is built using ADS and AME in order to check the result of recognition. The system consists of 4 main parts, such as, Preprocessing, Coordinate handling, Feature recognition and 3D-modeling. The system shows a good application example which can interface the design and manufacturing stage in CAD/CAM system on PC level.

## 1. 서론

최근의 제품개발 환경에서는 제품의 다양화, 고정도화, 신제품개발의 단축화 등을 위하여 기계가공에서 생산자동화가 급격히 진척되고 있다. 특히 금형제품의 설계 및 생산에 있어서는 디폴종소량체제이며 형상이 복잡, 다양하고 제품의 수명이 단축되는 등 복합적인 문제에 대처하기 위하여 자동화된 CAD/CAM 시스템이 절실히 요구되고 있다. CAD에서 생성된 도면정보를 가공정보로 자동으로 변환하기 위해서는 먼저 CAD시스템에서 작성된 도면을 인식할 수 있어야 하며, 이 인식된 결과에 의해 생성된 형상인식 데이터를 이용하여 가공정보를 만든다. 따라서 형상인식 과정이 무엇보다 중요하다.

본 연구에서는 금형에서 많이 사용되고 있는 특징형상들에 대해서 형상인식 알고리즘을 개발하였다. 인식대상이 되는 형상을 특징형상으로 정의하여 지식베이스화하고, 이 정의에 의해서 형상인식 규칙에 따라 2차원 도면에서 형상을 인식하여 인식데이터를 생성한다. 인식된 결과 데이터를 이용하여 3차원 모델링을 함으로써, 형상인식이 정확하게 되었

는지 확인할 수 있고 CAD/CAM의 일관된 자동화를 추구하여 NC정보, CAD/CAPP의 통합화로 활용할 수 있도록 한다.

CAD/CAM 인터페이스(Interface)는 CAD 정보가 생성되면 형상 데이터베이스와 인식에 관련된 지식베이스(Knowledge)를 이용하여 CAM 정보로 변환하기 위한 필요한 공정이다. 이때 도면정보가 대상이 되며 도면정보 교환을 용이하기 위하여 DXF(Drawing interchange files)나 IGES(Initial Graphics Exchange Specification) 데이터 파일을 이용하여 형상인식을 하고 있는 것이 일반적이다.[1-2] 이 방법은 인터페이스 환경을 교체할 수 있는 잇점이 있으나 복잡한 형상일수록 검색할 엔티티(Entity)수가 많아지게 되어 검색 효율이 떨어지고, 전환의 번거러움 등의 단점 때문에 본 연구에서는 DXF나 IGES 데이터 파일을 이용하지 않고 CAD 데이터 베이스내의 entity 속성 데이터를 이용하여 화면상에서 직접 형상인식을 수행한다. 화면상에서 직접 인식하는 방법은 인식 전문가의 시각적인 기능을 활용하는 것이라고 볼 수 있으므로 인식규칙을 보다 효과적으로 구축할 수 있으며, 관심이 있는 단면도나 임의영역에 있는 형상요소를 용이하게 검색하여 정보를 얻을 수 있다. 본 연구에서는 형상인식과 형상인식 결과 데이터를 이용한 3차원 모델링은 육면체, 구멍류(Through hole(관통구멍), Blind hole(막힌구멍), 키운터보이, 카운터싱크), 포켓류(아크포켓, open pocket, 사각포켓(Closed through pocket, Closed blind pocket))등에 친하여 수행하였다.

## 2. 시스템의 구성 및 흐름도

본 시스템의 형상인식 부분은 AutoCAD환경의 AutoLISP을 사용하였고, 여기서 나온 결과 데이터를 이용한 3차원 모델링 부분은 ADS(AutoCAD Development System)환경의 C언어로 작성되었다. 본 시스템 구성은 그림 1에서와 같이 삼면도로 그려진 도면을 입력하면, 일단 전처리 과정을 통하여 도면 수정 작업을 완료한다. 다음으로 각각의 투상도 별로 도면을 분류하고 각 투상도에 대하여 좌표변환을 수행한다. 다음은 이미 정의 되어 있는 형상인식 규칙을 사용하여 기본 형상별로 형상인식을 수행하고 인식 결과데이터를 화면에 출력, 데이터 화일에 저장된다. 생성된 형상인식 데이터를

사용하여 2차원으로 그려진 도면을 일련의 과정을 거쳐 3차원 형상으로 모델링하는 것이다. 개발된 시스템의 흐름은 그림 1에서와 같이 도식적으로 표현할 수 있다.

### 3. 특징형상 및 형상인식 알고리즘

#### 3.1 특징형상의 정의 및 분류

특징형성이란 제품의 부분요소로서 기능, 재조 또는 가공되는 방법 등과 같은 특정한 공학적인 작업과 직접적인 관련이 있는 공학적인 기본요소이며, 기하학적 및 위상학적 요소들의 조합으로 그들간의 상관관계와 특정 파라미터에 의해 정의될 수 있다[3-4]. 따라서 특징형상은 가공측면의 고려없이 설계측면으로만 모델링된 CAD 데이터로부터 가공작업에 필요한 정보를 도출하기 위해서는 가공작업에 의하여 생성되는 형상들을 일정한 형태로 정의할 필요가 있다. 따라서 본 연구에서는 금형부품의 절삭가공할 때 생성될 수 있는 형상을 고려하여 특징형상을 정의하고 이 특징형상들을 유사한 특징으로 그룹화하고 형상의 기하학적, 위상학적 관계를 이용하여 그림 2와 같이 특징형상을 분류하였다. 특징형상은 Hole 유사형상, Pocket 유사형상, Slot유사형상 그리고 기타 웨지(Wedge)나 필렛(Fillet), 섹터(sector) 등으로 구분한다. 표 1은 형상인식이 수행된 기본 특징형상들을 특정 파라미터에 의해 나타내고 있다.

#### 3.2 형상인식을 위한 좌표변환

형상인식을 수행하기 위해서는 AutoCAD 드로잉 에디터에서 X, Y 평면에 3각법으로 그려진 2차원 도면을 각 투상도별로 좌표변환할 필요가 있다. 그림 3에서와 같이 기준점 P보다 절대좌표 X, Y가 작은 도면요소를 정면도(Front view)로 두고, 기준점 P보다 Y좌표가 큰 도면요소를 평면도(Top view)로 둘, P보다 X좌표가 큰 도면요소를 우측면도(Side view)로 둔다. 이와 같은 방법에 의해서 분리된 각 각 투상도에 새로운 좌표계를 부여한다.

P : Reference point  
X, Y : 드로잉 에디터상의 절대좌표  
x, y, z : 투상도별로 변환한 좌표

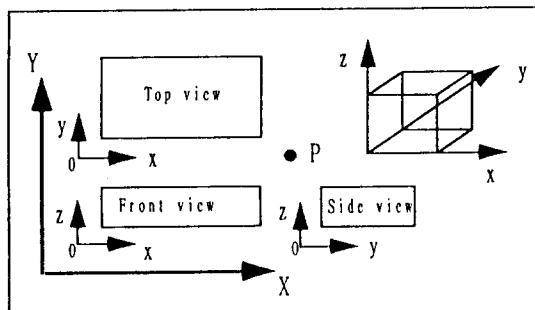


그림 3. 좌표 변환

#### 3.3 형상인식 알고리즘

형상인식은 크게 두 부분으로 나누어 수행되는데 전체도면

인식과 특징형상인식으로 나눌 수 있다. 전체도면인식은 삼면도로 구성된 정면도, 평면도, 측면도를 검색하여 각 단면도의 기준점을 찾고, 각 단면도들간에 기하학적 상관관계를 확인하여 직육면체 형상에 필요한 길이, 폭, 높이를 알아낸다. 그런 다음 각 단면도내에 있는 특징형상들을 인식규칙에 따라 형상인식을 수행한다.

#### 3.3.1 전체도면인식 알고리즘

[단계 1] 전체도면에서 좌표변환에 의해 삼면도로 분리된다. 삼면도 인식을 수행하기 위해서 삼면도 중앙에 기준점 P를 잡는다. 측면도(Side view)를 인식하기 위해서 S점을, 평면도(Top view)를 인식하기 위해서 T점을 잡는다.

[단계 2] 정면도에서 기준점 P보다 x,y점이 작은 모든 엔티티를 검색하여 그 중에서 가장 최소인 점(x,y)을 선택하여 정면도의 기준점인 b1를 찾는다.

[단계 3] 측면도에서 기준점 P의 x좌표값보다 큰 entity의 x좌표값만을 검색하여 entity들의 시작점과 끝점 중 가장 최소가 되는 점(x,y)를 찾아 측면도의 base point인 b2를 선정한다.

[단계 4] 평면도에서 기준점 P의 y값보다 큰 entity의 y좌표값을 검색하여 entity들의 가장 최소가 되는 점(x,y)를 찾아 평면도의 기준점인 b3를 선정한다.

[단계 5] 각 단면도의 기준점의 값들을 화면에 출력하고 각 단면도들간에 기하학적 상관관계를 고려하여 직육면체 형상에 필요한 길이, 폭, 높이를 알아낸다.

#### 3.3.2 특징형상인식 알고리즘

- Through hole과 Blind hole 인식규칙

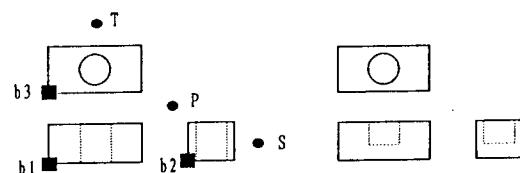


그림 4(a) Through hole이 포함되어 있는 투상도      (b) Blind hole이 포함되어 있는 투상도

Hole 형상의 특징은 평면도내에 원의 형상이 있고, 이와 관련하여 다른 투상도에 원의 좌표와 관련이 있는 선이 나타난다. 이 원의 중심좌표와 지름은 드로잉 데이터베이스에서 추출한다. 투상도에 원이 포함되어 있는 형상은 관통구멍, 막힌구멍, 카운터보어, 카운터싱크가 될 수 있는데, 동일한 중심을 갖는 다른 원이 있는지를 검색하여 없으면 관통구멍이나 막힌구멍으로 인식하고, 동일한 중심을 갖는 다른 원이 존재하면 카운터보어나 카운터싱크로 인식한다. 다음은 원이 검색된 이외의 투상도에서 원과 관련된 해당 line이 존재하는지를 확인하여 조건을 만족하면 Hole로 인식한다. 그림 4(a)를 보는 바와 같이 원과 관련된 해당 line entity의 y좌표값 차의 절대치와 기준점과 관련된

entity의 좌표값 차의 절대치를 비교하여 관통구멍과 막힌 구멍을 구별한다.

그림 5은 막힌구멍에 관한 형상인식 과정과 결과 데이터를 화면에 출력하고 사용자가 지시하는 파일에 저장하는 과정 예를 보여주고 있다.

```
*****
*          MAIN MENU
*****
* 1. FEATURE RECOGNITION
* 2. 3-D MODELING
* 3. RECOGNITION AND MODELING
* 4. QUIT
*****  
Choose the Number : 1  
blipmode ON/OFF <OFF> : OFF  
Command :  
Input the base point :  
Input the side view point :  
Input the Plane view point :  
(2.0 1.0)  
{2.0 5.0}  
(7.0 1.0)  
Enter the data file name : bhole1.dat  
  
HEXAHEDRON  
POINT-1 = (0 0 0)  
POINT-2 = (3.0 0 0 )  
POINT-3 = (0 2.0 0 )  
POINT-4 = (0 0 2.0 )  
LENGTH(X) = 3.0  
WIDTH(Y) = 2.0  
HEIGHT(Z) = 2.0  
  
HOLE-1 (Blind)  
Center point : (1.5 1.0 2.0)  
Direction : -3  
Diameter : 1.0  
Depth : 1.0
```

```
BHole1.dat  
OR  
x_or = 0.9  
y_or = 0.6  
z_or = 0.0  
BOX  
L = 3.0  
W = 2.0  
H = 2.0  
BHO  
x_c = 1.5  
y_c = 1.0  
z_c = 2.0  
r = 0.5  
dp = 1.0  
dir = -3  
END
```

그림 5. 막힌구멍에 관한 형상인식과정과 결과 data의 예

#### ● 카운터보어 및 카운터싱크의 인식규칙

그림 6에 나타난 것과 같이 카운터보어 및 카운터싱크의 형상을 인식하기 위해서는 먼저 중심이 같은 두개의 동심원이 존재하는지의 여부를 검색한다. 다음 두 원의 기저자리의 좌표값을 찾은 후에 이 좌표값에 연관된 직선 1, 2, 1', 2'가 다른 투상도상에 존재하는지를 확인하여 이 조건을 만족하면 직선 3, 4, 5 또는 3', 4', 5'를 조사한다. 카운터보어 형상은 직선 1, 2의 양 끝점을 큰 원의 가장자리 좌표와 일치한 반면에 카운터싱크는 직선 1', 2'의 한쪽 끝은 큰

원의 가장자리에 존재하고 다른 끝은 작은 원의 가장자리에 존재한다.

#### ● 아크포켓 (Arc pocket)의 인식규칙

아크포켓을 인식하기 위해서는 드로잉 데이터베이스에서 원호(Arc)에 대한 데이터를 추출한 다음 한 투상도에 서로 마주보는 대칭적인 원호가 존재하는지를 검색한다. 다음이 원호의 끝점을 연결하는 선이 존재하는지를 확인하고 다른 투상도에 아크포켓 형상과 관련된 line entity가 존재하면 아크포켓으로 인식한다.

#### ● Closed through pocket, Closed blind pocket, Open pocket의 인식규칙

Closed through pocket, Closed blind pocket, Open pocket을 인식하기 위해서는 각 투상도의 테두리 line entity를 제외한 line entity를 검색하여 그림 7(a)에서 보는 바와 같이 기준점 b3의 좌표값보다 큰 line 1, 2, 3, 4가 존재하는지를 검사한다. line 1, 2, 3, 4가 존재하면 다른 투상도에서 이 line과 관련된 line이 존재하면 Closed through Pocket이나 Closed blind pocket으로 인식한다. 두 포켓의 구별은 막힌구멍과 관통구멍의 인식 구별과 유사하다. Open pocket의 인식은 line 1', 2', 3'의 존재여부를 확인하고 다른 투상도에서 해당위치에 관련된 line이 존재하면 Open pocket으로 인식한다. 그림 8는 포켓류 인식 프로그램 일부를 보여주고 있다.

### 4. 인식된 결과의 3차원 모델링

형상인식된 결과 데이터를 이용한 3차원 모델링 작업은 ADS와 AME(AutoCAD Modeling Extension)의 API(Application Programming Interface)에서 제공되는 AME 기본 솔리드 생성함수와 규칙화된 Boolean Operators 함수를 사용하여 3차원 기본형상들을 생성한다. 3차원 모델링 작업은 CSG방식을 사용하여 모델링을 하는데 복합적인 형상들은 Boolean 연산인 합집합(Union), 차집합(Subtraction), 교집합(Intersection)을 이용하여 생성한다. 그림 9는 카운터보어 형상에 대한 인식결과로 생성된 데이터파일을 입력데이터로 사용하여 기본적인 3차원 primitive를 만든후, Boolean연산인 합집합을 사용하여 정의해 놓은 특징형상을 만드는 것을 보여준다. 이렇게 생성된 3차원 기본 형상들을 가공전 원재료에 해당되는 직육면체 형상의 해당 위치로 옮긴 후, 회전 및 명진을 수행하고 다시 Boolean 연산을 수행하면 전체적으로 조합된 형상에 대해 3차원 모델링이 완성된다. 이 공정은 C언어(ADS)로 구현하여 자동으로 구현하도록 프로그램화 되었는데 그림 9는 카운터 보어형상에 관련된 예를 나타낸 것이다.

```

ap_cylinder (CB_rw, CB_rw, CB_dpw, &cb[0];
ap_cylinder (CB_rn, CR_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);
cb[3]=(ap_Solid)NULL;
ap_union(cb,&cb1) ;

```

그림 9 Example of automated AME Programming in Counterbore

## 5. 적용 사례

본 연구를 통하여 개발된 특징형상 인식시스템은 구멍, 카운터보어, 카운터싱크, 아크포켓, 오픈포켓, 사각포켓등 금형부품의 가공에 많이 사용되는 특징형상들에 대해서 적용하였다. 그림 10은 프레스 금형용 도면을 보여주고 있는데 2차원도면으로 그려진 삼면도로부터 특징형상 인식규칙에 따라 형상인식하여 결과 데이터를 화일에 저장하는 예를 그림 11에 보여준다. 그림 12은 인식된 데이터를 이용하여 ADS와 AME를 사용하여 일련의 과정을 거쳐 삼차원 모델링한 것이다. 그림 12(a)는 와이어프레임 모델링(wireframe modeling)으로 나타내었고, 그림13(b)는 와이어 프레임 형태에서 Solmesh, Hide 명령을 사용하여 대상을 내부가 채워진 mesh block 형태로 변형시키 은선을 제거한 모양으로 나타내었다. 여기서 각종 해석을 위하여 재료, 밀도, 탄성계수, 체적, 무게중심, 관성농률등 계산할 수 있다.

## 6. 결 론

본 연구에서는 금형에서 많이 사용되고 있는 특징형상들을 기하학적, 위상학적 관계를 고려하여 유사한 형상으로 그룹화하고 각 특징형상들에 대하여 형상인식 규칙을 만들어 2 차원도면에서 형상을 인식하여 형상인식 데이터를 추출한다. 인식된 결과 데이터를 이용하여 3차원 모델링을 수행하였다.

하였다. 본 연구에 수행한 특징을 살펴보면 다음과 같다.

(1) 고가의 CAD시스템을 사용하지 않고 PC상에서 가장 많이 보급되어있는 AutoCAD를 사용하여 형상인식을 수행하였으며, 생성된 형상인식 데이터를 변환화일을 이용하지 않고 직접 3차원 입체형상으로 모델링할때 입력데이터로 사용하였고, 가공데이터로도 이용할수 있게 하였다.

(2) 대상부품, 인식할 수 있는 형상특징의 종류 등에서 한계성을 갖고 있지만 유사한 형상끼리 그룹화하여 형상인식을 용이하게 하였다.

(3) 본 프로그램은 각 기능별로 함수화 시켰기 때문에 서로 독립적이고 개별적으로 수행이 가능하도록 하여 차후 프로그램의 추가와 발전시키는데 용이하도록 하였다. 향후에는 곡면이 포함된 형상등 보다 다양한 종류의 형상들을 인식할 수 있는 알고리즘 개발이 필요하고 설계도면에서 3차원 모델링을 자동적으로 연결하여 NC 가공정보, 해석, 시뮬레이션 등의 공정에 적용될 수 있도록 지속적인 연구가 요구된다.

## 참고문헌

- [1] Srinivaskumar S. Madural and Li Lin, "Rule-based automatic part feature extraction and recognition from CAD data", Computers Ind. Engng, Vol. 22, No.1, pp.49-62, 1992.
- [2] RONG-KWEI and TAI-ZEN HWANG, "A part-feature recognition system for rotation parts-non-turning features", INT. J. COMPUTER-INTEGRATED-MANUFACTURIN, Vol.2, No.5, pp.257-267, 1989.
- [3] Arlington, "Functional Requirements for A Feature Based Modelling- System," CAM-I R-89-GM-01, 1989.
- [4] Shah, J.J., "Philosophical Development of form Feature Concept", in Proceeding of the CAM-I Features Symposium, P-90-FM-02, pp.55~77., 1990
- [5] 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
- [6] 이식희, 박재홍, 박길수, "형상인식 규칙의 지식 베이스 운용에 관한 연구", 한국 자동 제어 학술 회의 논문집, 1991
- [7] 손정태, 박면웅, "각주형 부품상의 가공 특징형상 인식" 대한기기 회의 논문집, 제 17권 제 6호, pp.1412-1422, 1993
- [8] 박기수, "금형 부품의 CAD 및 CAM정보 자동생성시스템", 박사학위논문, 부산대학교, 1994

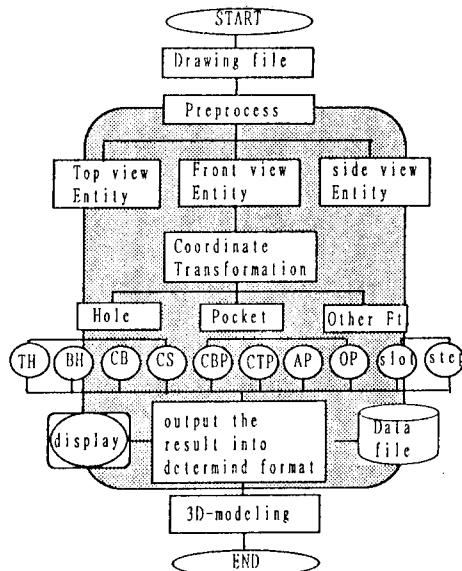


그림 1 형상인식수행부분

그림 1 시스템 구성도

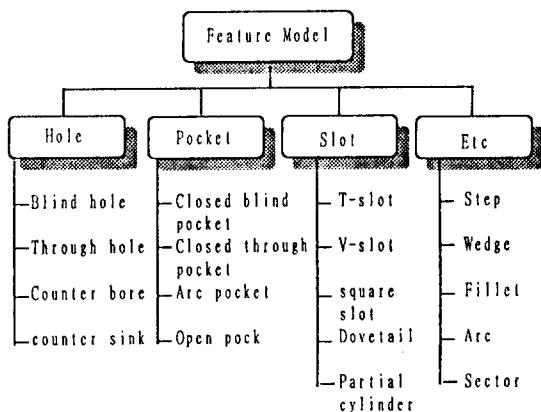
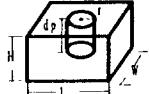
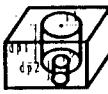
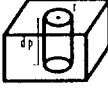
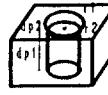
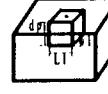
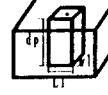
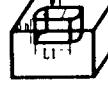
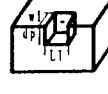
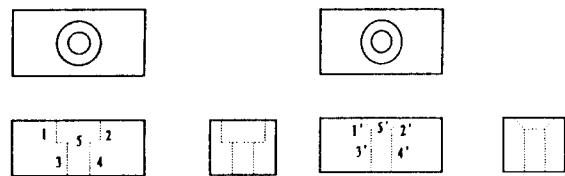


그림 2 특징형상 분류

표 1. 기본특징형상

특징형상 분류	표현 방법
	<p>OR (모델링의 기준점)  <math>x_{\text{or}}</math> = 기준점 x 좌표  <math>y_{\text{or}}</math> = 기준점 y 좌표  <math>z_{\text{or}}</math> = 기준점 z 좌표</p> <p>Box (직육면체형상)  <math>L</math> = 깊이  <math>W</math> = 폭  <math>H</math> = 높이</p> <p>Blind hole      BHO(특징형상 이름)  <math>x_{\text{c}}</math> = 중심점 x 좌표  <math>y_{\text{c}}</math> = 중심점 y 좌표  <math>z_{\text{c}}</math> = 중심점 z 좌표  <math>r</math> = 반지름  <math>dp</math> = 깊이  <math>dir</math> = 방향</p> 
	<p>Counter bore      CB(특징형상 이름)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>r1</math> = 반경 1  <math>r2</math> = 반경 2  <math>dp1</math> = 깊이 1  <math>dp2</math> = 깊이 2  <math>dir</math> = 방향</p> 
	<p>Through hole      THO(특징형상 이름)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>r</math> = 반지름  <math>dp</math> = 깊이  <math>dir</math> = 방향</p> 
	<p>Counter sink      CS(특징형상 이름)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>r1</math> = 반경 1  <math>r2</math> = 반경 2  <math>dp1</math> = 깊이 1  <math>dp2</math> = 깊이 2  <math>dir</math> = 방향</p> 
	<p>Closed blind pocket (CBP)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>ll</math> = 깊이  <math>wl</math> = 폭  <math>dp</math> = 깊이  <math>dir</math> = 방향</p> 
	<p>Closed through pocket (CTP)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>ll</math> = 깊이  <math>wl</math> = 폭  <math>dp</math> = 깊이  <math>dir</math> = 방향</p> 
	<p>Arc pocket (AP)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>ll</math> = 깊이  <math>wl</math> = 폭  <math>dp</math> = 깊이  <math>dir</math> = 방향</p> 
	<p>Open pocket (OPP)  <math>x_{\text{c}}, y_{\text{c}}, z_{\text{c}}</math> = 중심점좌표  <math>ll</math> = 깊이  <math>wl</math> = 폭  <math>dp</math> = 깊이  <math>dir</math> = 방향</p> 



(a) 카운터보어 (b) 카운터싱크

그림 6 카운터보이와 카운터싱크의 심면도

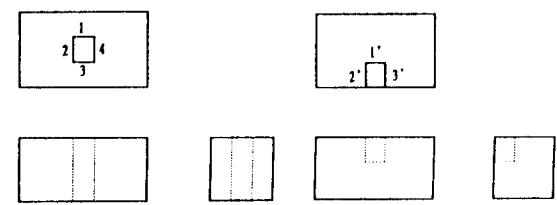


그림 7(a) Closed through pocket (b) Open pocket

```

ARC 정보를 갖는 entity를 검색 //From internal
D/B
if (존재하면) {
    arc에 대한 데이터를 추출
    해당 view를 설정
    두 개의 마주보는 arc가 존재하는지 여부 &&
    각 arc의 끝점을 연결하는 line의 존재 여부 &&
    if (존재하면) {
        다른 view를 설정
        아크포켓 형상과 관련된 line entity가 존재
        Arc pocket 형상존재
    }
    관련 D/B저장
}

```

그림 8 (a) Arc pocket 형상입식

```

line에 관련된 정보 추출 //From internal
D/B
삼면도 테두리 line entity를 제외한 line 검색
if(존재하면) {
    base point b3보다 큰 line 1,2,3,4 존재여부
    if (존재하면) {
        다른 view를 선정
        line 1,2,3,4와 관련된 해당 line이 존재&&
        base point와 관련된 line entity검색&&
        entity y좌표값 최대치와
        포켓과 관련된 해당 line entity의 y좌표값
        차이 철대치 비교
        if(같으면)
            Closed through Pocket 인식
        else (다르면)
            Closed blind pocket 인식
    }
}
관련 D/B저장

```

삼면도 테두리 line entity를 제외한 line 검색

```

if(존재하면) {
    base point b3보다 큰 line 1',2',3' 존재여부
    if (존재하면) {
        다른 view를 선정
        line 1',2',3'와 관련된 해당 line이 존재
        조건은 만족하면
        Open pocket 인식
    }
}
관련 D/B저장

```

(b) Closed through pocket, Closed open pock,  
Open pocket 형상인식

그림 8 포켓류 인식 알고리즘

OR	
x_of = 0.9	AP
y_of = 0.6	x_c = 2.25
z_of = 0.0	y_c = 1.15
BOX	z_c = 2.0
L = 8.0	ll = 3.0
W = 5.0	wl = 1.0
H = 2.0	dp = 2.0
	dir = -3
THO	
x_c = 0.5	
y_c = 1.0	CTP
z_c = 2.0	x_c = 1.25
t = 0.25	y_c = 0.5
dp = 2.0	z_c = 2.0
dir = -3	ll = 0.75
	wl = 0.75
	dp = 2.0
	dir = -3

그림 11. 형상인식 결과 데이터

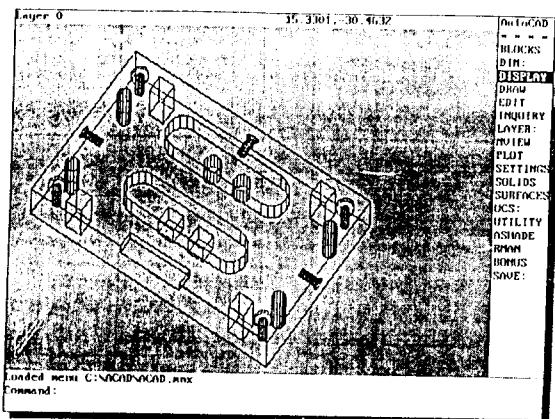


그림 12(a) WIRE FRAMING MODELING

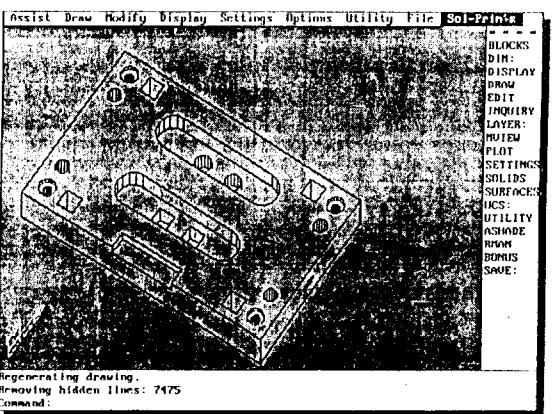


그림 12(b) SOLID MODELING

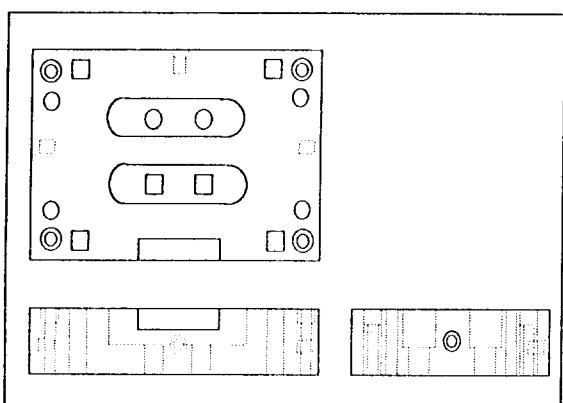


그림 10. 시스템 적용도면