

형상인식 규칙의 지식 베이스 운용에 관한 연구

박재홍, 반갑수, 이석희
부산대학교 공과대학 생산기계공학과

A Basic Research for Knowledge-Based Management of Feature Recognition Rules

Jae-Hong Park, Kap-Soo Bhan, Seok-Hee Lee
Dept. of Mechanical and Production Engineering
Pusan National University

ABSTRACT

In manufacturing process, usually 2-dimensional part drawing is used as a basic data. If a designer wants to recognize 2-dimensional drawing and formulate 3-dimensional shape, a proper feature recognition rule is required as a prerequisite step. These rules are converted into knowledge base, should be managed separately in the recognition program and can be referenced in similar way of database application. In this paper, basic feature recognition rules are addressed in structure type knowledge base, and the application system is formulated which can be operated separately with existing data driven program.

1. 서론

생산에 있어서 공정계획 작성, 절삭순서 결정, 절삭공구 선정, 절삭조건 선정, 절삭경로 시뮬레이션, NC 프로그램 자동생성 등의 기능들을 얻기 위해서 CAD/CAPP/CAM 집적 시스템(Integration System)의 구성이 필요한데, 여기에서 작업의 시작단계가 되는 것은 CAD 데이터(Data)의 생성이다. CAD 작업에 의하여 생성된 도면들은 기하학적 요소인 '점', '선', '원', '원호' 등으로 가공형상이 표현되어 있으므로 실제 생산에 필요한 기초 형상들의 속성의 획득이 중요한 문제가 된다.

형상 인식에 있어서 지식과 데이터의 혼합된 방식으로 인해서 생기는 프로그램 크기, 프로그램의 이해 및 운용, 지식 관리, 업데이트(up-date) 등의 어려움이 존재한다.

이번 연구에서는 실제 작업 단계에서 주로 사용되는 범용 CAD 시스템에서 구성된 3차원 입체형상의 2차원 도면에서의 Data를 기본 자료로 하며, 이를 3차원 정보로 변환하여 추출하고, 이를 바탕으로 3차원 형상을 정의하기 위한 기

본 형상들을 검색하여 추출하는데 사용되는 형상인식 규칙을 별도로 관리하여 지식베이스화 하여 실제 프로그램에서 사용될 수 있게 하였다.

지식 베이스 시스템은 정확하게 지정하기 어렵거나 전문가들의 지식을 필요로 하는 문제들의 해결에 도움을 줄 수 있다(1).

2. 형상 인식 요소의 분류 및 시스템 흐름도

2차원 도면에서 인식되어 3차원 형상의 모델링에서 기본 형상(primitive)이 되는 가공 형상들은 그림 1과 같이 계층적으로 분류될 수 있다(2).

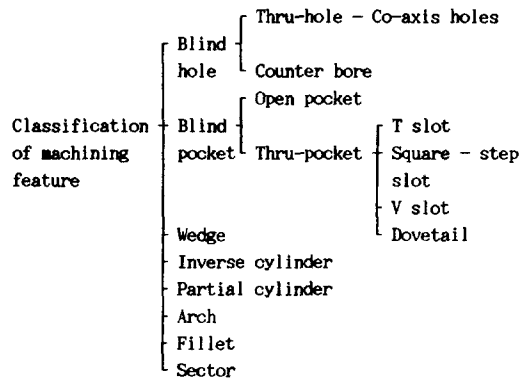


그림 1 가공 형상의 계층적 분류
Fig. 1 Hierarchical classification of machining features

그림 1에 나열되어진 기본적인 형상들 중에서 이번 연구에서는 기존 축에 수직인 구멍(Hole)과 카운터 보어(Counter bore)에 대한 인식작업에 관한 규칙을 우선적으로 연구하였다. 이들은 하나 또는 두개의 원과 직선으로 이

투어진 기본적인 회전형상으로, 3차원 위치 산출을 위한 인식작업에서의 규칙이 상대적으로 간단하여, 인식규칙의 적용에 적합하다고 생각된다.

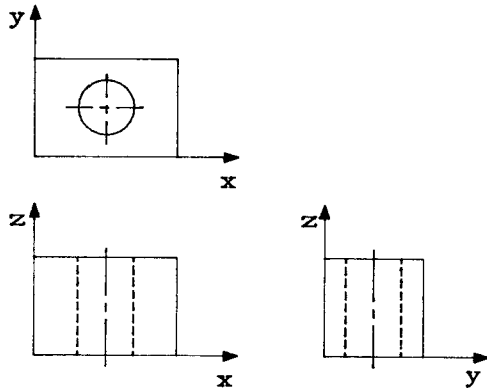


그림 2 구멍이 표현된 도면
Fig. 2 Example of drawing with hole

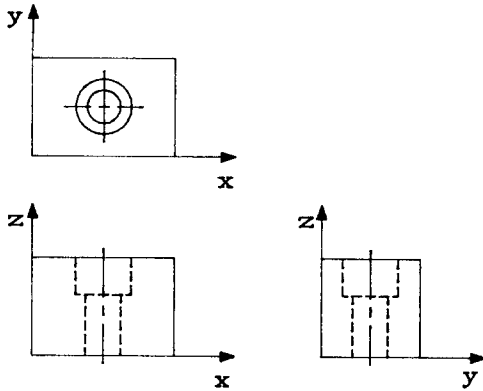


그림 3 카운터 보어가 표현된 도면
Fig. 3 Example of drawing with counter bore

실제의 인식 프로그램에서는 먼저 투상도별로 상대 좌표계가 설정되고, 설정된 상대 좌표계로 기본 요소(entity)의 정보를 변환하는 과정 및 변환된 좌표계에 의한 기본 요소 데이터의 추출과정이 AutoCAD 상에서 행해진다. 이 결과 생성된 인식 데이터 파일은 C 언어로 작성된 인식 프로그램에서 검색되고, 프로그램 내에서 미리 구성된 형상 인식 지식베이스를 참조하여 요구되는 형상인식 작업을 수행하며, 인식 결과 데이터 파일을 생성한다. 이러한 과정을 그림으로 나타내면 그림 4와 같다.

3. 지식 획득 및 운용.

지식은 그 분야의 전문가 또는 전문 서적을 통해서 획득한다. 규칙의 획득에서 활용까지는 다음의 단계를 거친다.

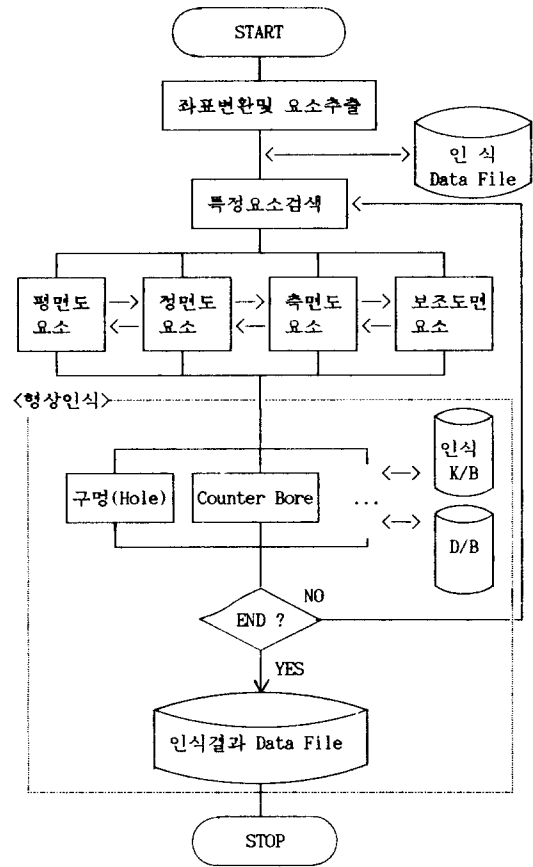


그림 4 시스템의 순서도
Fig. 4 System flow chart

- I. 전문가의 인식 규칙 획득.
- II. 지식의 표현.
- III. 인식단계의 흐름도.
- IV. 프로그램에의 활용.

3.1 축에 수직인 원형 구멍 및 카운터 보어의 인식 규칙(Rule)

먼저 2차원 도면에서 3각법으로 설계되어진, 축에 수직인 구멍(Hole)과 카운터 보어(Counter bore)에 대한 지식은 다음과 같이 표현될 수 있다.

(1) 구속조건 :

- 1) 투상도는 실척(1:1)으로 그려져야 한다.
- 2) 생략된 요소가 없어야 한다.
- 3) 점과 점 사이의 연결이 특정 오차 범위내에 들어야 한다.(전처리 과정)

(2) 인식 규칙 :

A. 각 투상도에 관한 좌표 변환

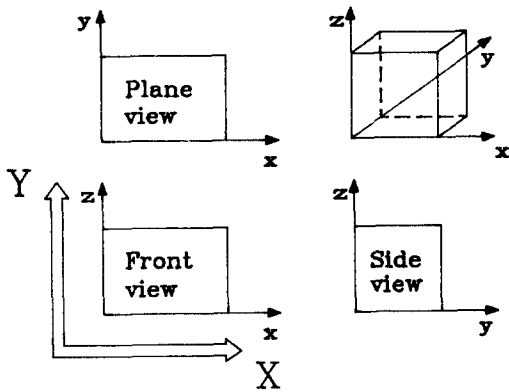


그림 5 좌표 변환
Fig. 5 Coordinate transformation

B. 전문가의 형상 인식 규칙

- 1) 구멍과 카운터 보어의 대표적인 형상인 원을 투상도에서 검색한다.
- 2) 동심원이 하나이면 구멍이, 둘이면 카운터 보어가 존재할 수 있다.
- 3) 원의 중심점과 반지름을 추출한다.
- 4) 원의 중심점에서 반지름을 더하거나 빼주어서 원의 가장자리의 위치를 구한다.
- 5) 위치에 해당되는 선(line)을 다른 투상도에서 검색한다. 두 선이 검색될 때 시작점과 끝점을 추출한다.
- 6) 두 선의 시작점과 끝점이 일치하는 지를 조사한다.
- 7) 두선의 끝점을 연결하는 선이 있는지 조사한다. - 만족하면 막힌 구멍이 된다.
- 8) 원의 선의 종류(Line type)를 검색하여 가공 방향을 조사한다.
- 9) 카운터 보어의 경우에는 두원에 대하여 위의 3) - 7) 까지의 순서를 반복하여 검색한다.
- 10) 위의 단계들을 모두 만족하면 결과의 자료(Data)를 출력한다.
그리고 나서 다음 원을 검색하여 위의 단계를 반복한다.
- 11) 모든 도면 요소가 검색되었으면 작업을 종료한다.

3.2 지식의 표현

전문가에 의해서 제시된 형상 인식 규칙이 인식 프로그램에서 사용되기 위해서는 그림 6과 같이 체계적인 형태로 재구성되어야 한다. 형상 인식 지식에서 판단의 조건이 되는 규칙들은 독립된 룰(Rule)로 표현되어 지식 베이스의 구성을 위한 흐름도(그림 7)작성의 기초가 된다.

```

** Global_Entity_CHECK
IF(CIRCLE == EXIST)                * Rule 1
  AND (CIRCLE.Num == ALONE)
  THEN (GO TO Other_View)
  AND Hole_CHECK;

IF(CIRCLE == EXIST)                * Rule 2
  AND (CIRCLE.Num == COCENTRIC)
  THEN (GO TO Other_View)
  AND Counter_bore_CHECK;

** Hole_CHECK
IF(Parallel_Line == EXIST)         * Rule 3
  AND (Line_Type == DASHED)
  THEN {
    IF(Length_of_Line == THICK)    * Rule 4
      THEN True_Hole;
    ELSE IF(Length_of_Line < THICK) * Rule 5
      THEN Blind_Hole;
  }

IF(CIRCLE.Line_Type == CONTINUOUS) * Rule 6
  THEN (Direction -> POSITIVE);
ELSE IF(CIRCLE.Line_Type == DASHED) * Rule 7
  THEN (Direction -> NEGATIVE);

** Counter_bore_CHECK
IF LOOP_for(i) (i=1,2)             * Rule 8
  {
    (Parallel_Line_i == EXIST)
    AND (Line_Type_i == DASHED)
  }
  THEN Counter_bore;
IF(CIRCLE.Line_Type == CONTINUOUS) * Rule 9
  THEN (Direction -> POSITIVE);
ELSE IF(CIRCLE.Line_Type == DASHED) * Rule 10
  THEN (Direction -> NEGATIVE);
  
```

그림 6 인식 지식의 표현
Fig. 6 Recognition Knowledge representation

3.3 인식 규칙의 흐름도.

그림 6에서 표현된 해당 형상에 대한 인식규칙의 적용에 관한 순서도는 다음과 같다.

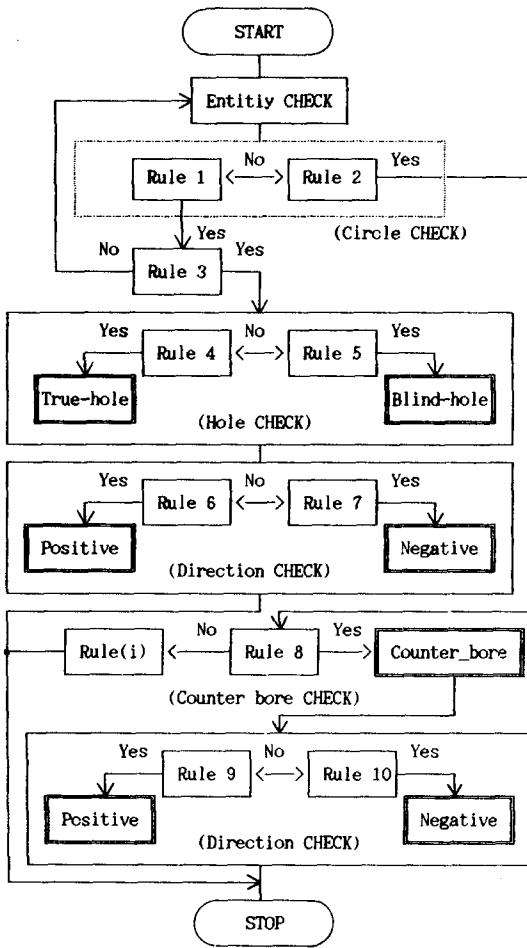


그림 7 지식활용의 흐름도

Fig. 7 Knowledge application flow chart

위와 같이 표현된 지식은 실제 프로그램 속에서 사용되어지며, 이때는 함수화 되어 프로그램과 값(Value)을 주고 받게 된다.

4. 입력 데이터(Data) 구성 상태.

- 형상 인식 프로그램에의 입력 데이터는 AutoCAD 내에서의 작업의 결과로 생성되며 ASCII 화일형태로 존재한다. 기존의 데이터 교환 화일(DXF, IGES)들과는 달리 전처리 작업을 통하여 도면내의 좌표 변환을 행한, 인식에 필요한 기본 요소들(원, 선)의 데이터만 포함하게 하여 간단하게 구성하였다.

- 도면 투상도 구분 : ^PLANE, ^FRONT, ^SIDE
- 화일 끝 표시 : ^END

- 원
 - 일련번호(해당 도면내에서의 순서)
 - 선의 종류(Line type)
 - 0 : CONTINUOUS (실선, 외형선)
 - 1 : DASHED (점선, 은선)
 - 2 : CENTER (일점쇄선, 중심선)
 - 중심점 (x)
 - 중심점 (y)
 - 반지름

- 선
 - 일련번호
 - 선의 종류(Line type) : 원과 같음
 - 시작점(x1, y1)
 - 끝점(x2, y2)

```

(예)
^PLANE
CIRCLE
0
0
0.5000
0.4000
0.2000
LINE
0
1
2.5000
2.5000
3.5000
2.5000
  
```

그림 8 입력 데이터

Fig. 8 Input data

5. 출력 Data 구성 상태.

- 인식 작업의 결과가 되며, 인식된 형상을 표현하기 위한 기본적인 데이터를 ASCII 화일 형태의 결과 화일에 출력한다. 인식되는 형상 데이터는 프로그램 속에서는 구조적인 형태를 취하여 구성된다.

- Hole :
 - 일련번호
 - 중심점 좌표(x, y, z)
 - 반지름
 - 방향(중심점에서의 구멍의 방향)
 - 깊이

- Counter :
 - 일련번호
 - bore
 - 중심점 좌표(x, y, z)
 - 반지름1(wide)
 - 반지름2(narrow)
 - 방향(중심점에서의 방향)
 - 깊이1(wide)
 - 깊이2(narrow)

```

(예)
# Hole data

Hole Number(1) :
center point x = 1.500000
               y = 1.500000
               z = 0.000000

radius        r = 0.500000
direction     = -Z
depth        = 2.000000

Hole Number(2) :
center point x = 3.000000
               y = 0.000000
               z = 1.000000

radius        r = 0.500000
direction     = -Y
depth        = 2.500000

# Counter Bore data

Counter Bore Number(1) :
center point  x = 0.500000
               y = 0.400000
               z = 0.000000

radius1(wide) r1 = 0.200000
radius2(narrow) r2 = 0.100000
direction     = -Z
depth1(up)   = 0.900000
depth2(down) = 0.400000
  
```

그림 9 형상 인식 결과 데이터
Fig. 9 Feature recognition output data

6. 결론

CAD 에서 생성된 2차원 도면의 데이터를 기초자료로 하여 3차원의 입체형상으로 모델링하는데 중요한 역할을 하는 인식 규칙들을 프로그램 내에서 별도로 지식 베이스화 하여 사용되게 하였다. 결국 지식을 체계적으로 관리하게 되기 때문에 인식 작업의 효율이 증가된다.

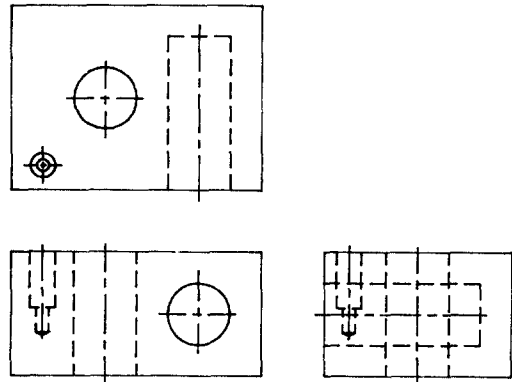


그림 10 형상 인식 예제 도면
Fig. 10 Feature recognition example drawing

앞으로의 연구에서는 보다 많고 광범위한 기본 가공 형상들에 대한 인식 규칙들을 획득하여, 3차원 모델링에 적용할 수 있도록 하기 위하여 전문가가 인식하는 방식을 최대한 활용하는 지식관리 체계를 구축하여야 한다.

7. 참고문헌

- (1) R.K.Li and T.Z.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.
- (2) 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.
- (3) H.Schildt, " Artificial Intelligence Using C ", Osborne McGraw-Hill Berkeley, California. 1987.
- (4) B.V.D.Pluym, " Knowledge-based decision-making or job shop scheduling ", Int.J.Computer Integrated Manufacturing, Vol.3, No.6, pp. 354 - 368, 1990.
- (5) J.Harhen, " Prospects for AI in PMS ", Knowledge Based Production Management Systems, Elsevier Science publishers B.V. (North-Holland), pp. 3 - 13, 1989.
- (6) AutoLISP Programmer's Reference, AUTODESK, INC. 1989.