

卒業論文

PC를 이용한 3次元 立体 그래픽스에
關한 研究

A STUDY ON THE THREE-DIMENSIONAL SOLID
GRAPHICS USING A PERSONAL COMPUTER

指導教授 정 재 현



1990年 2月

한국해양대학교 선박기계공학과

서 재복 전 복문 정 병우
정 병기 최 충배 박 경은

목 차

1 . 서 론

2 . 기본 명령문

- 2. 1 그래픽스 화면의 준비
- 2. 2 그래픽스 화면의 모드 선택

3 . 3 차원 그래픽스

- 3. 1 3 차원 도형의 2 차원 평면에서의 투영변환
- 3. 2 회전이동
- 3. 3 원점이동
- 3. 4 SCALING
- 3. 5 평행이동
- 3. 6 대칭변환
- 3. 7 투시도
- 3. 8 SUBROUTINE을 이용한 프로그램

4 . DATA 관리 프로그램

- 4. 1 DATA 관리 프로그램
- 4. 2 SEQUENTIAL FILE 을 사용한 입체도

5 . 은선처리 방법

- 5. 1 최대, 최소치에 의한 방법
- 5. 2 PAINTING에 의한 방법

6 . 결 론

참고문헌

1. 서 론

인간은 수천년전부터 도구와 기계를 사용하여 여러가지 계산을 해왔다. 그래서 계산을 위한 도구들은 엄청난 발전을 거듭해왔고 현대에 와서는 과거 상상으로만 생각해왔던 수준까지 오른 것이 사실이다. 이러한 계산기를 크게 애널로그(ANALOG)형, 디지털(DIGITAL)형, 이 두 가지의 혼합형인 하이브리드(HYBRID)형으로 구별하고 있는데 지금의 전자계산기라 하면 일반적으로 디지털 계산기만을 지칭하며 컴퓨터 혹은 전산기라 한다. 이 전자계산기는 인간이 지금까지 고안하고 이용하여온 수많은 기계들이 대부분 물질과 에너지를 생산하고 이용하기 위해서 만들어진 것임에 비해, 언어를 이해하고 정보를 처리하는 기계라는 점에서 타 기계와 현저하게 다르고, 인간의 두뇌에 해당하는 부분으로 비유된다. 따라서 이러한 전자계산기가 수학상의 문제는 물론 현대의 고도화된 정보사회에서 인간이 취급하는 거의 모든 정보를 원하는대로 신속하게 처리할 수 있을 뿐만 아니라 사용이 용이하고, 소비 전력이 적으며, 기계의 구성이 소형화되고, 용량과 기능이 증가하는 반면에 가격은 급격하게 저하되는 이유등으로 보급이 날로 증가하여 현 사회의 모든 분야에서 심지어 개인업무용으로 까지 사용되기에 이르렀다. 진공관으로 구성된 제1세대 컴퓨터로 시작하여 언어를 이해하고 스스로 사고하는 지능 컴퓨터(INTELLIGENT COMPUTER)의 제5세대 컴퓨터까지 컴퓨터는 많은 발전을 해왔고 컴퓨터를 조작하는 언어(LANGUAGE)도 따라서 발전해 왔다는 것은 다 주지하고 있는 사실이다. 이렇게 생각해 볼 때 현대를 살아가는 학생 뿐만 아니라 모든 사람들이 컴퓨터의 도움없이 급변하는 사회를 따라갈 수가 없게 되고있고, 컴퓨터의 조작기술 측면에서 볼 때 특히 공학도들에게는 필수적인 요건이 되고있다. 따라서, 본 연구에서는 이러한 제반 요소들을 감안하고 또 날로 그 중대성이 증가하는 CAD/CAM의 추세에 부응하여 2차원 평면 좌표 시스템의 이해를 기초로 3차원 입체물의 PC(PERSONAL COMPUTER)에 의한 GRAPHICS표현에 대해 서술해 보고자 한다. 참고로 본 연구에 사용된 PC는 NEC의 PC-9801이며 프로그래밍에 사용한 언어는 N-88 BASIC임을 밝혀 둔다.

2. 기본명령문

이 장에서는 BASIC 언어로 사용가능한 그래픽스의 기본적인 명령문의 종류를 표로 열거한다. 이들 명령문은 다음의 프로그램에서 간단하게 언급이 될것이다. 이러한 기본명령문은 이미 개인적으로 습득이 되었으리라 보고 자세한 해설은 생략하였으므로 이해가 어려운 부분에 대해서는 퍼스컴 매뉴얼이나 기타 전문서적을 참고하기 바란다. 아래 표 2-1 에 BASIC 일반의 기본적인 명령문을 보인다.

表 2-1

コマンド・ステートメント			
CLOSE	GOSUB~RETURN	LPRINT	SAVE
CONSOLE	GOTO	MERGE	READ
COPY	IF~THEN~ELSE	MOD	RENUM
DATA	INPUT	NAME	RESTORE
DEF FN	INPUT #	NEW	RUN
DELETE	KILL	ON~GOSUB	SAVE
END	LIST	OPEN	STOP
ERASE	LLIST	PRINT	WIDTH
FILES	LOAD	PRINT #	WRITE #
FOR~NEXT	LOCATE		
関 数			
ATN	INKEY\$	SPC	ASC
COS	INT	SQR	
EOF	SIN	TAN	

2.1 그래픽스 화면의 준비

화면에는 문자나 수식을 표시하는 것과 직선이나 원과 같은 도형을 표시하는 것의 두 종류가 있다. 전자를 텍스트 화면이라 부르는데 대해 후자를 그래픽스 화면이라 부른다. 이 두개의 화면은 각각 독립된 화면이고, 서로 영향을 미치는 것은 아니지만 실제 두 개의 화면은 겹쳐져서 표시되어지기 때문에 마치 한 개의 화면인것처럼 보인다.

텍스트 화면에서는 문자단위의 표시이기 때문에 상세한 그래프나 그림을 그릴 수가 없다. 이에 대해 그래픽스 화면에서는 도트 단위로 그림을 표시하기 때문에 선을 긋는다든지 원을 그리는 것이 가능하다.

그래픽스 화면은 PLANE 이라 불리는 12매의 화면으로 구성되어져 있다. 12매의 PLANE 은 赤(R) , 綠(G) , 靑(B) 의 삼원색으로 구별되어진 640*200 닷트의 화면이 각 색마다 4 매씩 있는 것이다. 각각의 PLANE 에 그려진 그림을 개별로 또는 조합해서 도형을 표시하기 때문에 640*200 닷트의 흑백 12화면, 640*400 닷트의 고해상 흑백 6 화면, 640*200 닷트의 컬러 4 화면, 640*400 닷트의 고해상 컬러 2 화면의 4 가지의 표시 모드를 선택할 수 있다. 이 그래픽스 화면의 모드를 선택하는 것이 SCREEN 명령문 이고, 텍스트 화면의 모드를 선택하는 것이 CONSOLE 명령문이다.

2.2 그래픽스 화면의 모드 선택

그래픽스 명령을 사용할 경우에는 그래픽스 화면을 어떻게 사용할 것인가를 우선 SCREEN 명령을 사용해서 선언하지 않으면 안된다.

* 기본형

SCREEN	화면모드	화면스위치	ACTIVE PAGE	DISPLAY PAGE
	0	0		
	1	1		
	2	2		
	3	3		
- 화면모드	0	:	640*200 닷트의 컬러 4 화면	
	1	:	640*200 닷트의 흑백 12화면	
	2	:	640*400 닷트의 고해상 흑백 6 화면	
	3	:	640*400 닷트의 고해상 컬러 2 화면	
- 화면스위치	0	:	NORMAL SPEED	
	1	:		
	2	:	특수용도	
	3	:		

*기능 : 그래픽스 화면의 모드 선택.

3. 3 차원 그래픽스

3.1 3 차원 도형의 2 차원 평면에서의 투영변환

투영의 기초이론은 컴퓨터로써 도형을 그릴 때에도 마찬가지로 적용되는 지식이다. 우선, 3 차원도형을 표현할 공간의 직교좌표계를 도입해보자. 공간에 임의의 점(원점 0)을 정하고, 이 0 를 통과해서 서로 직교하는 직선 OX, OY, OZ 를 그린다. 여기서, 평면 $O-XY, O-YZ, O-ZX$ 를 각각 XY 평면, YZ 평면, ZX 평면이라 부르고 이것들을 총칭해서 좌표평면이라 한다. 직선 OX, OY, OZ 를 각각 X 축, Y 축, Z 축이라 하고 총칭해서 좌표축이라 한다.

그래서, 공간에서 한점 P 를 생각할 경우, 그것에 대한 X, Y, Z 의 좌표축상에는, 순서를 가진 실수의 조 x, y, z 를 정한다. 이 때, x, y, z 를 각각 P 의 x 좌표, y 좌표, z 좌표라고 하는 (x, y, z) 를 P 의 좌표라고 부른다. 이것을, $P(x, y, z)$ 로 표시한다. (그림 3. 1)

거꾸로 순서를 가진 실수의 조 x, y, z 가 고려된다면, 이것에 관한 공간상의 점 P 가 구해진다.

좌표축에 눈금을 붙일 경우는, 정방향의 좌표축상에 양수를 배열하고, 그것과 역방향으로 음수눈금을 붙인다. (그림 3. 2)

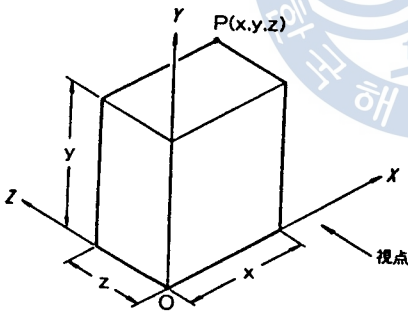


圖 3. 1

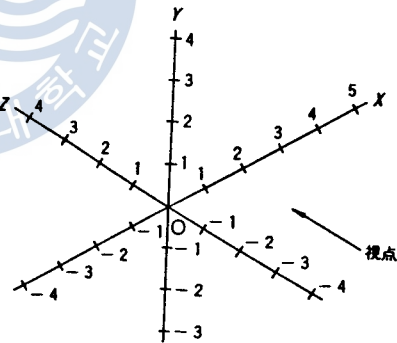


圖 3. 2

본 책에서는, 시점과 좌표계는, 그림 3. 1 과 같이 위치관계에 있는 것으로 된다. 이와같은 관계에 있을 때, 이 좌표계는 시점좌표계 혹은 원편좌표계라고 부른다.

원인은, 왼손을 들어서 엄지손가락을 세울 경우, 엄지손가락이 X 축의 정방향을, 다른 손가락은 Y 축의 정방향을 가르키고, 손바닥이 Z 축의 정방향으로 향해 있기 때문이다.

이 3 차원의 시점좌표계의 속에 위치한 물체를 2 차원으로 표현하는 처리를 투영 변환이라기한다. 그림으로 나타내면, 그림 3. 3 과 같이 된다.

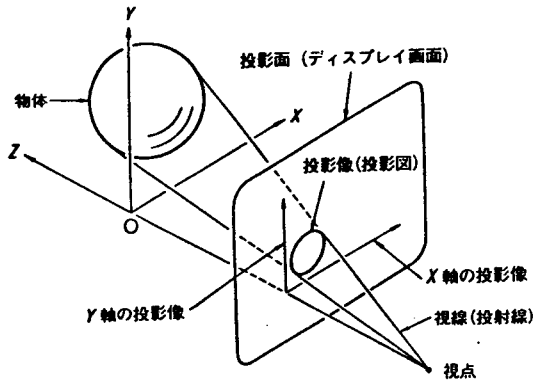


圖 3.3

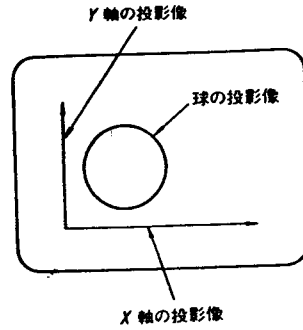


圖 3.4

그림 3. 3은, 시점좌표계의 안의 물체 (이경우 공) 를, 시점과 물체의 사이에 어떤 투영면을 통과시킨다고 본 그림이다. 시점이 인간의 눈, 투영면이 화면이라고 생각하면 좋을 것이다.

바꿔말하면, 화면에는 물체의 투영상 (투영도) 가 비춰진다고 생각한다. 투영면을 투명한 glass 라고 생각하고, 시점의 위치에 눈을 고정하고, 물체를 본 그대로를 glass 판상에 비끼기가공한 상을 투영상이라고 생각할 수도 있다. 이 때, 시점과 물체의 각 점을 연결한 선을 시선 (투사선) 이라고 부른다.

시점이 Z 축의 연장상에 있는 것이라고 한다면, 시점좌표계의 화면상으로서의 투영상은, 그림 3. 4와 같이 Z 축은 나타나지 않고, 직교하는 X 축, Y 축에 의해서 구성되는 2차원의 직교좌표로 된다.

이번에는, 물체에 입방체를 골라 보자. 입방체를 시점좌표계의 안에, 입방체의 기준 3면을 좌표계면에 평행하게끔 놓아보자. (그림 3. 5) . 이 때, 투영면에 투영된 도형은, 단순한 정방형에 지나지 않는다.

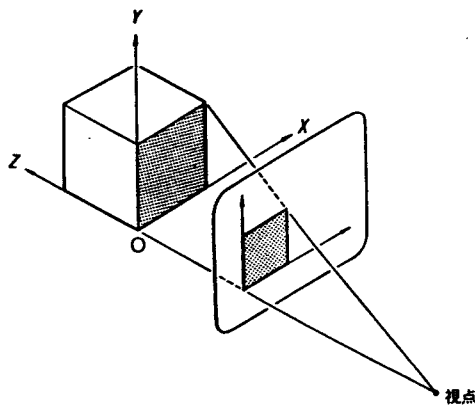
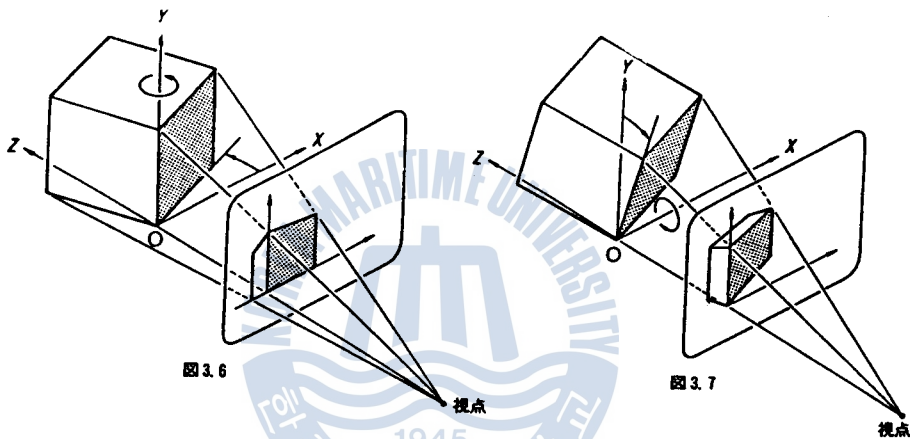


圖 3.5

다음으로, 이 입방체를 Y 축을 기준으로 해서 회전시켜 보자. 회전방향은, 원점에서 Y 축의 양의 방향으로 향하게 해서 시계방향을 양으로 한다.

회전후의 투영도는, 그림 3. 6으로 된다. 입방체의 양측면이 보여지지만, 아직 완전하게 입방적인 그림이 되지 않는다. 거기에서, 다시한번 이 입방체를 X 축방향으로 회전시켜 보자. 회전의 방향은, 원점에서 X 축의 양의 방향으로 향하는 시계방향을 양으로 한다.

회전후의 투영도는, 그림 3. 7과 같이 되고, 조금씩 입방적인 그림이 나타난다. 이와같이, 물체의 입방적인 투영상을 얻기 위해서는, 물체를 시점좌표계의 안으로 Y 축방향의 회전, X 축방향의 회전순서로 회전이동시킨 투영면(화면) 상의 투영상을 구해 보면 좋을 것으로 안다.



다시한번, Z 축방향으로 회전시킨 경우는, 그림 3. 8과 같이 원점에서 Z 축의 양의 방향으로 향하는 시계방향을 양으로 해서 회전시킨다.

그림 3. 7에서 얻어진 입방체의 투영상은 그림 3. 9와 같이 되지만, 입방체의 각 면을 구성하는 모서리의 투영상은 평행은 아니고, 그림과 같이 각각 1점에 모이고 있다. 이와같은 투영도를 투시도라고 부르고, 모아진 점을 소점이라고 부른다. 소점의 수에 의해서, 1소점투시도, 2소점투시도, 3소점투시도로 나누어진다.

여기에서, 시점의 위치를 물체에 대해서, 무한이 먼 곳까지 멀리한다고 한다. 그래서, 결국은 시점은 그림 3. 10에 나타난 것같이 서로 평행으로 되어서 투영면과 직교하고, 입방체의 투영상은 그림 3. 11과 같이 각 면을 구성하는 모서리의 투영상이 서로 평행한 그림으로 된다. 이와같이 할 수 있는 투영도를 축측투영도라고 한다.

말하자면, 시점이 투영면에서 유한한 위치에 있는 경우는 투시도로 되고, 무한이 먼 곳에 있다면 축측투영도로 된다.

투시도는 원근감이 있는 자연적인 그림인데 비해서, 축측투영도는 원근감이 없는 길고 훑직한 물체등은 비뚤어진 정도를 느끼는 그림으로 된다.

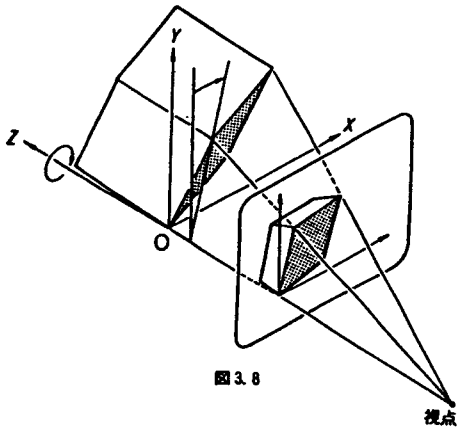


图 3.8

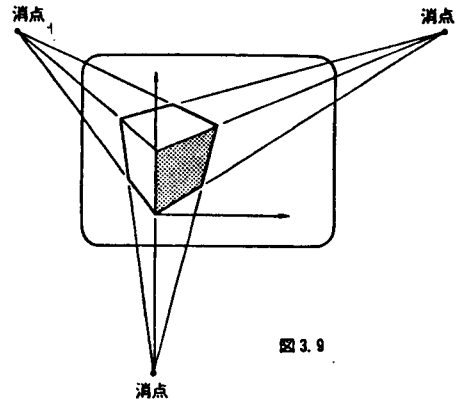


图 3.9

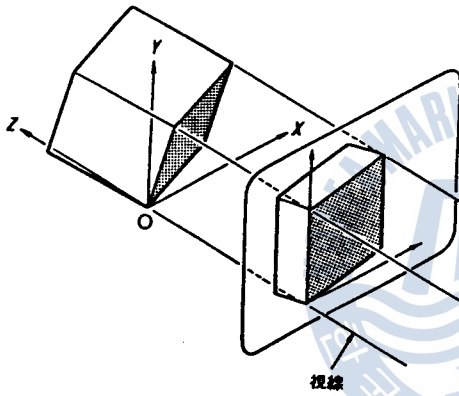


图 3.10

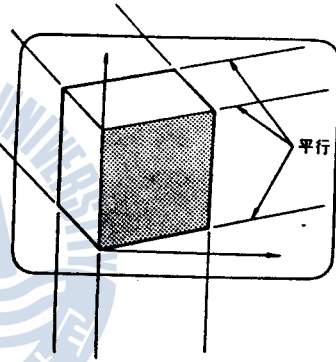


图 3.11

3.2 회전이동

일반적으로 2 차원평면에서 그림3.12에서 보는 바와 같이 $P(X,Y)$ 가 각도 θ 만큼 회전해서 $P'(X',Y')$ 가 되었다고 한다면

$$X' = X \cos \theta + Y \sin \theta$$

$$Y' = -X \sin \theta + Y \cos \theta \text{의 관계가 성립한다.}$$

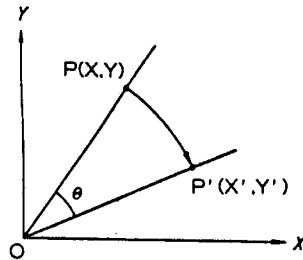


圖 3.12

따라서, 전술한 Y 축 주위의 회전의 변환식은 다음과 같이 해서 구할 수가 있다. Y 축 주위의 회전각을 Y_R , 회전 이전의 어떤 점의 좌표를 (X_1, Y_1, Z_1) , 회전후의 그 점의 좌표를 (X_2, Y_2, Z_2) 라 한다. Y 축 주위의 회전이기 때문에 Y 좌표값은 변하지 않고 $Y_2 = Y_1$ 이 된다. X 와 Z 의 좌표치의 변화는 2 차원의 경우와 같이 그림 3.13에서

$$X_2 = X_1 \cos(Y_R) - Z_1 \sin(Y_R)$$

$$Z_2 = Z_1 \sin(Y_R) + X_1 \cos(Y_R) \text{로 구해진다.}$$

같은 방법으로 X 축 주위의 회전 (회전각 X_R) 으로 얻는 좌표 (X_3, Y_3, Z_3) 는

$$X_3 = X_2$$

$$Y_3 = Z_2 \sin(X_R) + Y_2 \cos(X_R)$$

$$Z_3 = Z_2 \cos(X_R) - Y_2 \sin(X_R) \quad (\text{그림 3.14})$$

회전각 Z_R 로 Z 축 주위의 회전으로 얻는 좌표 (X_4, Y_4, Z_4) 는

$$X_4 = Y_3 \sin(Z_R) + X_3 \cos(Z_R)$$

$$Y_4 = Y_3 \cos(Z_R) - X_3 \sin(Z_R) \quad (\text{그림 3.15})$$

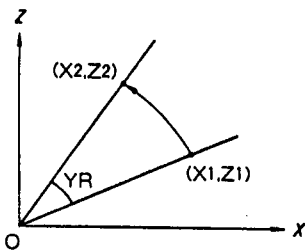


圖 3.13

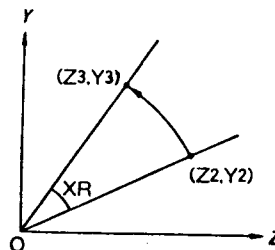


圖 3.14

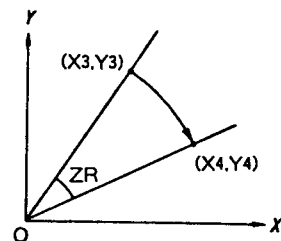


圖 3.15

정리하면, 우선 Y 축 주위의 ZX평면상에서 $P1(X1, Y1, Z1)$ 을 YR 회전해서 $P2(X2, Y2, Z2)$ 를 구하고, 다음은 X 축 주위의 YZ평면상에서 XR 회전해서 $P3(X3, Y3, Z3)$ 를 구한다. (그림 3.16, 3.17)

또, Z 축 주위의 XY평면상에서 ZR 회전해서 $P4(X4, Y4, Z4)$ 를 구할 수 있다. (그림 3.18)

이 $P4$ 를 투영면(화면)에 투영한 것이 그림 3.19이다. 여기에서 $Z4$ 의 값은 지워져있고 최종적으로 필요한 것은 $X4$ 와 $Y4$ 의 값이다.

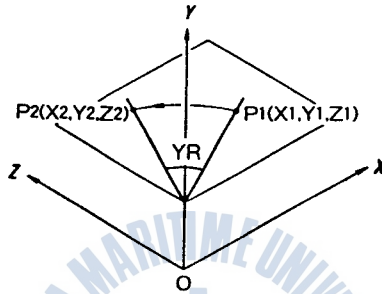


圖 3. 16

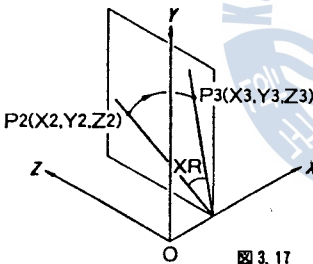


圖 3. 17

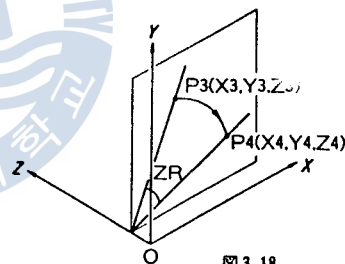


圖 3. 18

회전의 순서는 Y 축 주위, X 축 주위, Z 축 주위의 순으로 하는데 같은 각도 YR, XR, ZR 로 회전시켜도 회전순서가 바뀌면 투영상의 위치, 형이 달라지게 된다. 일반적으로는 Y 축 주위, X 축 주위의 회전에 의해 3 차원도형을 얻고, 특별한 경우 외에는 불안정한 형의 3 차원도형이 되는 Z 축 주위의 회전은 생략한다.

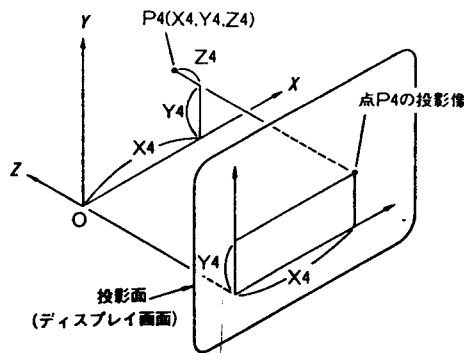


圖 3. 19

3.3 원점이동

지금까지의 GRID 표시는 원점을 화면의 중앙에 놓았었지만, 큰 그림등을 표시할 때 원점이 고정된 채로 있으면 불편할 것이다. 그래서, GRID를 표시하기 전에 원점의 이동량을 키-인해서 원점을 변경할 수가 있다.
(SUBROUTINE PROGRAM 2230-2620 참조)

3.4 SCALING

도형의 확대, 축소를 행하는 것을 SCALING 이라 한다. 아래 그림에 있어서 원점 O를 중심으로 해서 점 P(X1,Y1)의 X 성분과 Y 성분을 각각 S 배하면, 점 P는 P'(X2,Y2)로 이동한다. 결국 그림 전체가 S 배된 결과가 된다. X 성분과 Y 성분을 별도로 SCALING 하면 그림을 변경시킬 수가 있다.
(SUBROUTINE PROGRAM 3600-3740 참조)

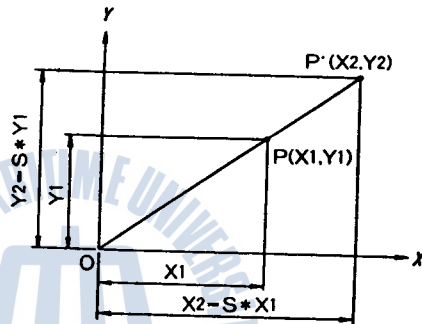


圖 3.43

3.5 평행이동

아래 그림의 XY 평면에 있어서 점 P(X1, Y1)를 점 P'(X2, Y2)로 평행이동하는 경우를 생각해 보자. X 방향의 이동량을 L, Y 방향의 이동량을 M 이라고 하면,

$$X2 = X1 + L$$

$$Y2 = Y1 + M \quad \text{로 계산된다.}$$

여기에서는 투시도를 작성할 때의 응용을 고려해서 이동량을 그림 3.46과 같이 X 축 방향에 XM, Y 축 방향에 YM, Z 축 방향에 ZM으로 지정할 수가 있을 것이다. 즉, 투영변환 후에 도형을 이동하는 것 뿐 아니라, X, Y, Z 의 각 축에 연해서 이동되어지기 때문에 SCALING 이나 투영변환을 행하는 것이다.

(SUBROUTINE PROGRAM 2660-2680, 3100-3170 참조)

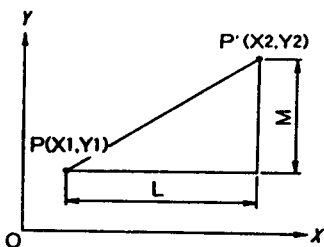


圖 3.45

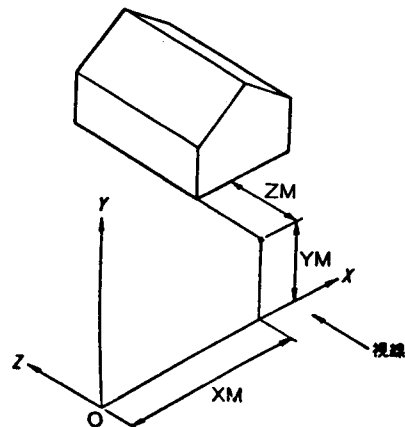


圖 3.46

3.6 대칭변환

임의의 선 또는 점에 관해서 대칭적으로 이동을 행하기 위한 변환을 대칭변환이라 한다. 원래의 좌표 (X_1, Y_1) 의 Y 축에 관해서 대칭인 위치의 좌표 (X_2, Y_2) 는

$$X_2 = -X_1, \quad Y_2 = Y_1 \quad \text{이 된다.}$$

따라서, SUBROUTINE PROGRAM 3920과 3960을 다음과 같이 바꾸면, Y 축에 대해 대칭인 도형이 얻어진다. 원래의 도형이 그림 3.48의 A 라고 하면, 도형 B 가 얻어지는 것이다.

```
3920 POINT (-X(I),-Y(I))
```

```
3960 LINE (-X(I),-Y(I))
```

원래의 좌표 (X_1, Y_1) 의 X 축에 대해 대칭인 위치의 좌표 (X_3, Y_3) 는

$$X_3 = X_1, \quad Y_3 = -Y_1 \quad \text{이 된다.}$$

그러므로, SUBROUTINE PROGRAM 3920과 3960을 다음과 같이 바꾸면, X 축에 대해 대칭인 도형이 얻어진다. 그림 3.48의 C 가 얻어진다.

```
3920 POINT (X(I),Y(I))
```

```
3960 LINE (-X(I),Y(I))
```

원래의 좌표 (X_1, Y_1) 의 원점에 대해 대칭인 위치의 좌표 (X_4, Y_4) 는,

$$X_4 = -X_1, \quad Y_4 = -Y_1 \quad \text{이 된다.}$$

프로그램은 다음과 같고, 그림 3.48에 보이는 도형 D 를 얻는다.

```
3920 POINT (-X(I),Y(I))
```

```
3960 LINE (-X(I),Y(I))
```

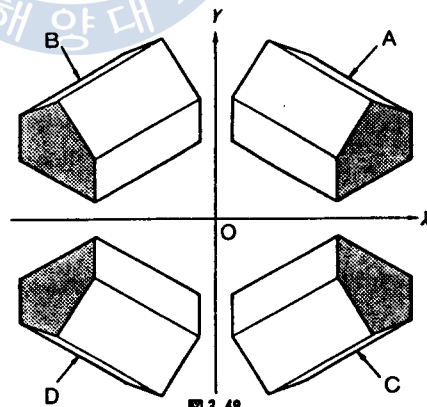


圖 3.48

3.7 투시도

지금까지 설명한 축척투영에서는 투영면에 대해서 시선(투사선)이 서로 평행했었지만 투시투영에서는 인간의 눈에 해당하는 시점과 물체의 거리가 유한해서, 시선은 1 점(즉, 시점)에 구속되는 것처럼 된다. (그림 3.49)

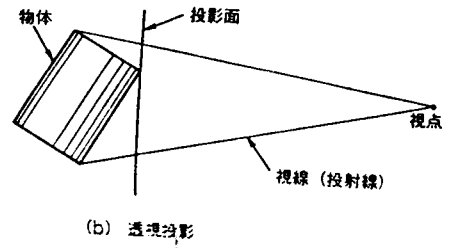
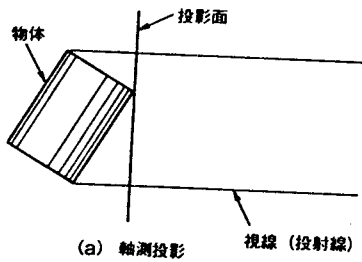


圖 3.49

축척투영에서는 물체가 좌표의 어느 위치에 있어도 좌표에 대한 물체의 회전각이 같으면 같은 형태의 투영도가 얻어진다. 그러나, 투시도에서는 물체가 좌표의 어느 위치에 있는가에 따라, 혹은 물체와 시점과의 거리에 의해 투영도형은 조금씩 변화한다.

3.7.1 투시도의 원리

그림 3.50(a)는 Y 축 주위, X 축 주위에 회전된 물체를 바로 위에서 바라본 평면도이고, (b)는 바로 옆에서 바라본 측면도이다. 이렇게 해서 얻어진 평면도, 측면도 각각에 있어서 물체 각 점의 칫수를 투영면상에 하나씩 대응시켜 나타내면 (c)와 같은 투시도가 구해진다.

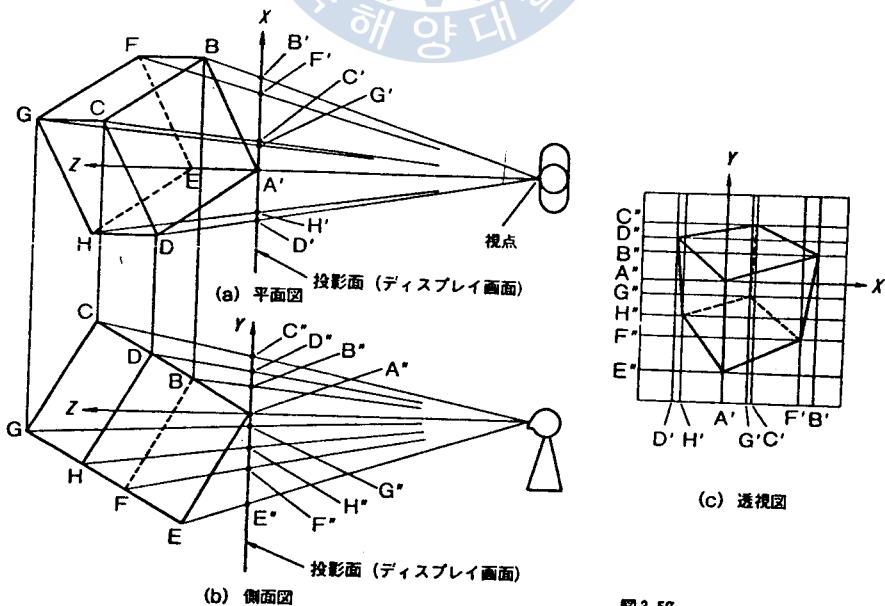


圖 3.50

이번에는, 이것을 작도 뿐만 아니라 계산으로 구하는 것을 생각해 보자. 그림 3.50(b) 에 있어서 직방체의 변 A"E"가 투영면에서는 A"E"의 길이로 투영되어지지만 이 길이를 계산으로 구해보자.

그림 3.51은 이 A"E", A"E"만을 취해낸 것이다. 점 E의 좌표치는 알고있는 것으로 한다. 즉, Y와 Z의 값은 이미 알고있다고 한다. 또, 투영면과 시점과의 거리는 SP라 한다. 하면,

$$\frac{Y}{Y'} = \frac{Z + SP}{SP}$$

의 관계가 성립하므로,

$$Y' = \frac{Y * SP}{Z + SP}$$

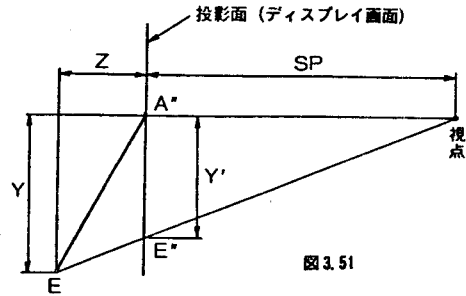


图 3.51

로 되고, 투영면상에서의 A"E"의 값을 구할 수가 있다.

이 관계는 물체의 다른 점에 대해서도 마찬가지로 성립한다. 따라서, 투시변환을 위해서는 프로그램에

$$X(I) = SP * X(I) / (SP + Z(I))$$

$$Y(I) = SP * Y(I) / (SP + Z(I))$$

$$Z(I) = SP * Z(I) / (SP + Z(I))$$

를 추가하면 좋다는 것을 알 수 있다.

3.7.2 1 消点透视图

물체의 기본면을 좌표면에 평행으로 놓았을 때 가능한 투시도를 1 초점투시도라 부른다. 즉, Y 축, X 축 주위의 회전을 생각하지 않았을 때 가능한 투시도. 그림 3.52와 같고, 초점은 1 개, 실내도등에 잘 사용된다.

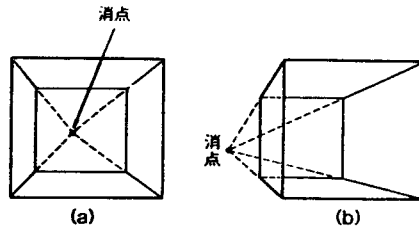


图 3.52

3.7.3 2 消点透视图

물체의 직교하는 세개의 기본축(직방체에서는 각변) 중에서 1 개를 투영면에 평행하게 놓았을 때 가능한 투시도. 바꿔 말하면, Y 축 주위의 회전만 생각하고, X 축 주위의 회전은 고려하지 않았을 때 가능한 투시도. 건물의 투시도에서는 3 초

점투시도보다도 보기 쉽기 때문에 자주 사용된다. 그림 3.53과 같고, 초점은 2 개이다.

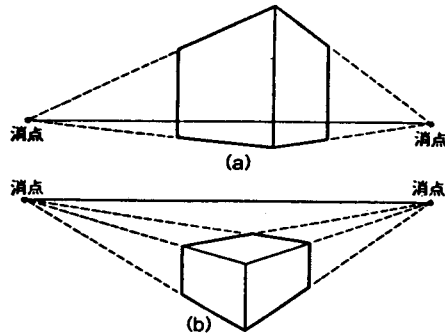


图 3.53

3.7.4 3 消失点透视图

물체의 직교하는 세개의 기본축을 모두 투영면에 대해서 경사지게 놓았을 때 가능한 투시도로서 Y 축 주위의 회전에 연속해서 X 축 주위의 회전을 고려할 때 가능한 투시도. 인간의 눈에 보이는 물체의 형상에 가장 가까이 접근, 자연적인 그림이 된다. 그림 3.54와 같이 초점은 3 개이다.

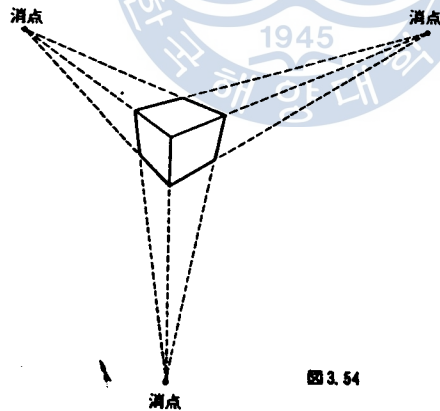


图 3.54


```

1 '*****
2 '*****
3 '   PROGRAM NAME : MAIN PROGRAM (FOR DRAWING AIR PLANE)
4 '*****
5 '*****
10 CHAIN MERGE "SUB1" ,ALL,DELETE 10
20 GOSUB 2100 : GOSUB 2900
30 FOR SKY=0 TO 2*PAI+PAI/8 STEP PAI/8
40   RESTORE 80 : GOSUB 8200 : GOSUB 8300 : GOSUB 3100
50   GOSUB 3200 : GOSUB 3600 : GOSUB 8400
60 NEXT SKY
70 DATA 14
80 DATA 0,0 , 1,1 , 6,22 , 7,30 , 7.5,40 , 7.5,66 , 7.5,75 , 7.5,88 , 7.5,115 ,
   7,123 , 6,129 , 4,135 , 1,139.5 , 0,140
110 ERASE X,Y,Z,DL$,XX,YY : RESTORE 200 : GOSUB 2900
120 FOR K=1 TO 2
130   FOR SKY=0 TO 2*PAI+PAI/5 STEP PAI/5
140     RESTORE 205 : GOSUB 8200 : GOSUB 8300
150     XMM=24 : YMM=75 : ZMM=6 : IF K=2 THEN XMM=-24
160     GOSUB 10000
170     GOSUB 3200 : GOSUB 3600 : GOSUB 8400
180   NEXT SKY
190 NEXT K
200 DATA 6
205 DATA 2,0 , 2.8,5 , 3,8 , 4,8 , 4.8,15 , 4,19
210 ERASE X,Y,Z,DL$,XX,YY : RESTORE 230 : GOSUB 2900
220 RESTORE 240:GOSUB 3000:GOSUB 3100:GOSUB 3200:GOSUB 3600:GOSUB 3800
230 DATA 29
240 DATA 6.5,62,4,P , 22,60.5,2,* , 68.5,43.5,-2.5,* , 68,49,-2.5,* ,
   67,51,-2.5,* , 9,92,4,* , 6.5,95,4,*
250 DATA -6.5,62,4,P , -22,60.5,2,* , -68.5,43.5,-2.5,* , -68,49,-2.5,* ,
   -67,51,-2.5,* , -9,92,4,* , -6.5,95,4,*
260 DATA 1.5,4,-2,P , 27.5,-2,-6,* , 27,2.5,-6,* , 7,19,-2,* , 5.5,21,-2,*
270 DATA -1.5,4,-2,P , -27.5,-2,-6,* , -27,2.5,-6,* , -7,19,-2,* , -5.5,21,-2,*
280 DATA 0,8,-3,P , 0,-3,-33,* , 0,4,-33,* , 0,27,-10,* , 0,33,-7,*
300 END

```

```

1000 '*****
1001 '*****
1002 '**                                PROGRAM NAME : SUBROUTINE                                **
1003 '*****
1004 '*****
1005 '
1009 '***** SPLINE - CURVE *****
1010 T(0)=0
1020 FOR I=1 TO N-1
1030   T(I)=T(I-1)+SQR(((XX(I)-XX(I-1))^2)+((YY(I)-YY(I-1))^2))
1040 NEXT I
1050 FOR FL=1 TO 2
1060   FOR P=0 TO N-1
1070     IF FL=1 THEN Y(P)=XX(P) ELSE Y(P)=YY(P)
1080     X(P)=T(P)
1090   NEXT
1100   AA=2 : P=1
1110   F=(Y(P+1)-Y(P))/(X(P+1)-X(P))
1120   G=(Y(P)-Y(P-1))/(X(P)-X(P-1))
1130   L=(X(P+1)-X(P))/(X(P+1)-X(P-1))
1140   BB=6*(F-G)/(X(P+1)-X(P-1))
1150   A(P)=AA : B(P)=BB
1160   FOR P=2 TO N-2
1170     E=(X(P)-X(P-1))/(X(P+1)-X(P-1))
1180     F=(Y(P+1)-Y(P))/(X(P+1)-X(P))
1190     G=(Y(P)-Y(P-1))/(X(P)-X(P-1))
1200     D=6*(F-G)/(X(P+1)-X(P-1))
1210     BB=D-E*BB/AA : AA=2-E*L/AA : A(P)=AA : B(P)=BB
1220   NEXT
1230   P=N-2 : M(P+1)=0 : Z=B(P)/A(P) : M(P)=Z
1240   FOR P=N-3 TO 1 STEP -1
1250     L=(X(P+1)-X(P))/(X(P+1)-X(P-1)) : Z=(B(P)-L*Z)/A(P)
1260     M(P)=Z
1270   NEXT
1280   M(0)=0
1290   FOR I=0 TO N-1
1300     IF FL=1 THEN M1(I)=MI ELSE M2(I)=MI
1310   NEXT
1320 NEXT
1330 T=T(0)
1340 J=0
1350 J=J+1 : IF J>N THEN GOTO 1450
1360 IF T>=T(J) THEN GOTO 1350
1370 H=T(J)-T(J-1) : U=T(J)-T : V=T-T(J-1)
1380 X=M1(J-1)*((U^3)/H-U*H)/6+M1(J)*((V^3)/H-V*H)/6+U*XX(J-1)/H+V*XX(J)/H
1390 Y=M2(J-1)*((U^3)/H-U*H)/6+M2(J)*((V^3)/H-V*H)/6+U*YY(J-1)/H+V*YY(J)/H
1400 X(I)=X : Y(I)=Y : Z(I)=0
1410 X=X(I) : X(I)=X(I)*COS(SKY) : Z(I)=X*SIN(SKY)
1420 GOSUB 3160 : GOSUB 3260 : GOSUB 3650
1430 LINE -(X(I),-Y(I))
1440 T=T+ST : GOTO 1340
1450 RETURN

```

```

2000 '
2010 '          SUBROUTINE PROGRAM
2020 '
2100 CONSOLE 0,25,0,1 : SCREEN 3,0 : WIDTH 80,25 : CLS 3 : COL=7
2110 POX=640 : POY=400 : PAI=3.14159
2120 DPX=2.678 : 'DOTS PER 1mm TO THE X-DIRECTION
2130 DPY=2.721 : 'DOTS PER 1mm TO THE Y-DIRECTION
2150 PRINT:PRINT:PRINT:PRINT:PRINT SPC(9) "A          1 - FOCUS PROJECTION"
2160 PRINT SPC(9) "B          2 - FOCUS PROJECTION"
2170 PRINT SPC(9) "C          3 - FOCUS PROJECTION"
2180 PRINT SPC(9) "D          SHAFT PROJECTION"
2190 PRINT : PRINT : PRINT
2200 INPUT "CHOOSE ONE AMONG A - D !" ; A$
2220 CLS 1
2230 PRINT : PRINT
2240 INPUT "DECIDE THE ORIGINAL-POINT (X0,Y0) (mm)!" ; X0,Y0
2320 X0=X0*DPX : Y0=Y0*DPY
2330 WINDOW (-POX/2-X0,-POY/2+Y0)-(POX/2-X0,POY/2+Y0)
2332 INPUT "DO YOU INTEND TO EXPRESS THE GRID (Y/N)" ; YN$
2334 IF YN$="Y" OR YN$="y" THEN 2340 ELSE 2650
2340 LINE (-POX,0)-(POX,0) : 'X-SHAFT
2350 LINE (0,-POY)-(0,POY) : 'Y-SHAFT
2360 'GRID MAKING-OUT
2370 '(1) X-DIRECTION
2380 NDX=INT(POX/(DPX*10))
2390 FOR I=-NDX TO NDX
2400     IF I=0 THEN 2450
2410     XF=I MOD 5
2420     IF XF<>0 THEN 2440
2430     LINE (DPX*10*I,-POY)-(DPX*10*I,POY),1
2440     LINE (DPX*10*I,-POY)-(DPX*10*I,POY),1,,&H9999
2450 NEXT I
2460 '
2470 '(2) Y-DIRECTION
2480 NDY=INT(POY/(DPY*10))
2490 FOR I=-NDY TO NDY
2500     IF I=0 THEN 2550
2510     YF=I MOD 5
2520     IF YF<>0 THEN 2540
2530     LINE (-POX,DPY*10*I)-(POX,DPY*10*I),1
2540     LINE (-POX,DPY*10*I)-(POX,DPY*10*I),1,,&H9999
2550 NEXT I
2560 COLOR 4
2570 YHJ=INT(Y0/(POY/25))
2580 IF YHJ>11 OR YHJ<-11 THEN 2600
2590 LOCATE 78,11-YHJ : PRINT "X"
2600 XHJ=INT(X0/(POX/80))
2610 IF XHJ>39 OR XHJ<-39 THEN 2630
2620 LOCATE 41+XHJ,0 : PRINT "Y" : COLOR 7

```

```

2630 '
2640 PRINT : PRINT : PRINT : PRINT
2650 INPUT "          MAGNIFICATION" ; RT
2660 INPUT "          MOVEMENT TO THE X-DIRECTION (mm)" ; XM : XXM=XM
2670 INPUT "          MOVEMENT TO THE Y-DIRECTION (mm)" ; YM : YYM=YM
2680 INPUT "          MOVEMENT TO THE Z-DIRECTION (mm)" ; ZM : ZZM=ZM
2690 IF A$="D" GOTO 2760
2700 INPUT "          DISTANCE OF PP-SP (mm)" ; SP
2710 IF A$="A" THEN 2770
2720 IF A$="B" GOTO 2740
2730 IF A$="C" GOTO 2750
2740 INPUT "          INPUT ROTATION-ANGLE ABOUT Y-SHAFT" ; YR : GOTO 2770
2750 INPUT "          INPUT ROTATION-ANGLE ABOUT Y-->X SHAFT" ; YR,XR : GOTO 2770
2760 INPUT "          INPUT ROTATION-ANGLE ABOUT Y-->X-->Z SHAFT" ; YR,XR,ZR
2770 CLS 1
2780 P=PAI/180 : YR=P*YR : XR=P*XR : ZR=P*ZR
2790 RETURN
2800 '
2900 '***** DIMENSION *****
2910 READ NN
2920 DIM X(NN),Y(NN),Z(NN),DL$(NN),XX(NN),YY(NN)
2930 RETURN
2940 '
3000 '***** READ DATA *****
3010 FOR I=1 TO NN
3020   READ X(I),Y(I),Z(I),DL$(I)
3030 NEXT I
3040 RETURN
3050 '
3100 '***** MOVEMENT *****
3110 FOR I=1 TO NN
3120   GOSUB 3160
3130 NEXT I
3140 RETURN
3150 '
3160 X(I)=X(I)+XM : Y(I)=Y(I)+YM : Z(I)=Z(I)+ZM
3170 RETURN
3180 '
3200 '**** ROTATION-TRANSFORMATION , PROJECTION-TRANSFORMATION SUBROUTINE ****
3210 FOR I=1 TO NN
3220   GOSUB 3250
3230 NEXT I
3240 RETURN
3250 IF A$="A" THEN 3450
3260 '
3270 '(1) ROTATION ABOUT Y-SHAFT
3280 X1=X(I) : Z1=Z(I)
3290 X(I)=X1*COS(YR)-Z1*SIN(YR)
3300 Z(I)=X1*SIN(YR)+Z1*COS(YR)
3310 IF A$="B" THEN 3450
3320 '
3330 '(2) ROTATION ABOUT X-SHAFT
3340 Z2=Z(I) : Y2=Y(I)
3350 Y(I)=Z2*SIN(XR)+Y2*COS(XR)
3360 Z(I)=Z2*COS(XR)-Y2*SIN(XR)
3370 IF A$="C" THEN 3450

```

```

3380 '
3390 '(3) ROTATION ABOUT Z-SHAFT
3400 Y3=Y(I) : X3=X(I)
3410 X(I)=Y3*SIN(ZR)+X3*COS(ZR)
3420 Y(I)=Y3*COS(ZR)-X3*SIN(ZR)
3430 IF A$="D" THEN RETURN
3440 '
3450 '(4) PROJECTION-TRANSFORMATION
3460 X(I)=SP*X(I)/(SP+Z(I))
3470 Y(I)=SP*Y(I)/(SP+Z(I))
3480 RETURN
3490 '
3600 '***** MEASUREMENT-CORRECTION , SCALING *****
3610 FOR I=1 TO NN
3620     GOSUB 3650
3630 NEXT I
3640 RETURN
3650 '
3670 '(1) MEASUREMENT-CORRECTION
3680 X(I)=X(I)*DPX : Y(I)=Y(I)*DPY
3690 '(2) SCALING
3700 X(I)=X(I)*RT : Y(I)=Y(I)*RT
3730 RETURN
3740 '
3800 '***** LP OR LINE *****
3810 FOR I=1 TO NN
3820     IF DL$(I)="P" THEN GOSUB 3900 : GOTO 3870
3850     IF DL$(I)="L" THEN GOSUB 3900 : GOTO 3870
3860     GOSUB 3940
3870 NEXT I
3880 RETURN
3890 '
3900 '(1) LP
3920 POINT (X(I),-Y(I)) : RETURN
3940 '(2) LINE
3960 LINE -(X(I),-Y(I)),COL : RETURN
3980 '
4000 '***** READ DATA (CIRCLES) *****
4010 READ R,XC,YC,ZC,AS,AE
4020 RETURN
4030 '
4100 '***** CIRCLE-DRAWING SUBROUTINE *****
4110 N=INT(R*RT/1.5+20) : BB=2*PAI/N
4120 AS=P*AS : AE=P*AE
4130 RETURN
4140 '
4200 '***** CIRCLES ON THE X-PLANE *****
4210 FOR I=INT(AS/BB) TO INT(AE/BB)+1
4220     AA=BB*I
4230     IF I=INT(AS/BB) THEN AA=AS
4240     IF I=INT(AE/BB)+1 THEN AA=AE
4250     YC2=R*SIN(AA) : ZC2=R*COS(AA)
4260     X(I)=XC : Y(I)=YC+YC2 : Z(I)=ZC+ZC2

```

```

4270     GOSUB 3160 : GOSUB 3250 : GOSUB 3650
4280     IF AA=AS THEN GOSUB 3900 : GOTO 4300
4290     GOSUB 3940
4300 NEXT I
4310 RETURN
4320 '
4400 '***** CIRCLES ON THE Y-PLANE *****
4410 FOR I=INT(AS/BB) TO INT(AE/BB)+1
4420     AA=BB*I
4430     IF I=INT(AS/BB) THEN AA=AS
4440     IF I=INT(AE/BB)+1 THEN AA=AE
4450     ZC2=R*SIN(AA) : XC2=R*COS(AA)
4460     X(I)=XC+XC2 : Y(I)=YC : Z(I)=ZC+ZC2
4470     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
4480     IF AA=AS THEN GOSUB 3900 : GOTO 4500
4490     GOSUB 3940
4500 NEXT I
4510 RETURN
4520 '
4600 '***** CIRCLES ON THE Z-PLANE *****
4610 FOR I=INT(AS/BB) TO INT(AE/BB)+1
4620     AA=BB*I
4630     IF I=INT(AS/BB) THEN AA=AS
4640     IF I=INT(AE/BB)+1 THEN AA=AE
4650     YC2=R*SIN(AA) : XC2=R*COS(AA)
4660     X(I)=XC+XC2 : Y(I)=YC+YC2 : Z(I)=ZC
4670     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
4680     IF AA=AS THEN GOSUB 3900 : GOTO 4700
4690     GOSUB 3940
4700 NEXT I
4710 RETURN
4720 '
4800 '***** SPHERE ABOUT X-SHAFT (1) *****
4810 FOR SXR=0 TO PAI STEP PAI/DVN
4820     FOR I=INT(AS/BB) TO INT(AE/BB)+1
4830         IF I=INT(AS/BB) THEN AA=AS : GOTO 4860
4840         IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 4860
4850         AA=BB*I
4860         ZC2=R*SIN(AA) : XC2=R*COS(AA)
4870         X(I)=XC2 : Y(I)=0 : Z(I)=ZC2
4880         Y(I)=Z(I)*SIN(SXR)+YC : Z(I)=Z(I)*COS(SXR)+ZC : X(I)=X(I)+XC
4890         GOSUB 3160 : GOSUB 3260 : GOSUB 3650
4900         IF AA=AS THEN GOSUB 3900 : GOTO 4920
4910         GOSUB 3940
4920     NEXT I
4930 NEXT SXR
4940 RETURN
4950 '
5000 '***** SPHERE ABOUT Y-SHAFT (1) *****
5010 FOR SZR=0 TO PAI STEP PAI/DVN
5020     FOR I=INT(AS/BB) TO INT(AE/BB)+1
5030         IF I=INT(AS/BB) THEN AA=AS : GOTO 5060
5040         IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 5060
5050         AA=BB*I

```

```

5060      YC2=R*SIN(AA) : XC2=R*COS(AA)
5070      X(I)=XC2 : Y(I)=YC2 : Z(I)=0
5080      Z(I)=X(I)*COS(SZR)+ZC : X(I)=X(I)*SIN(SZR)+XC : Y(I)=Y(I)+YC
5090      GOSUB 3160 : GOSUB 3260 : GOSUB 3650
5100      IF AA=AS THEN GOSUB 3900 : GOTO 5120
5110      GOSUB 3940
5120  NEXT I
5130  NEXT SZR
5140  RETURN
5150  '
5200  '***** SPHERE ABOUT Z-SHAFT (1) *****
5210  FOR SYR=0 TO PAI STEP PAI/DVN
5220      FOR I=INT(AS/BB) TO INT(AE/BB)+1
5230          IF I=INT(AS/BB) THEN AA=AS : GOTO 5260
5240          IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 5260
5250          AA=BB*I
5260          YC2=R*SIN(AA) : ZC2=R*COS(AA)
5270          X(I)=0 : Y(I)=YC2 : Z(I)=ZC2
5280          X(I)=Y(I)*SIN(SYR)+XC : Y(I)=Y(I)*COS(SYR)+YC : Z(I)=Z(I)+ZC
5290          GOSUB 3160 : GOSUB 3260 : GOSUB 3650
5300          IF AA=AS THEN GOSUB 3900 : GOTO 5320
5310          GOSUB 3940
5320      NEXT I
5330  NEXT SYR
5340  RETURN
5350  '
5351  '***** SPHERE ABOUT X-SHAFT (2) *****
5352  RR=R : XXC=XC : AS=P*AS : AE=P*AE
5354  FOR ANGX=-PAI/2+PAI/DVN TO PAI/2-PAI/DVN STEP PAI/DVN
5356      R=RR*COS(ANGX) : N=INT(R*RT/1.5+20) : BB=2*PAI/N
5358      XC=XXC+RR*SIN(ANGX) : GOSUB 4200
5360  NEXT ANGX : RETURN
5361  '
5370  '***** SPHERE ABOUT Y-SHAFT (2) *****
5372  RR=R : YYC=YC : AS=P*AS : AE=P*AE
5374  FOR ANGY=-PAI/2+PAI/DVN TO PAI/2-PAI/DVN STEP PAI/DVN
5376      R=RR*COS(ANGY) : N=INT(R*RT/1.5+20) : BB=2*PAI/N
5378      YC=YYC+RR*SIN(ANGY) : GOSUB 4400
5380  NEXT ANGY : RETURN
5382  '
5390  '***** SPHERE ABOUT Z-SHAFT (2) *****
5392  RR=R : ZZC=ZC : AS=P*AS : AE=P*AE
5394  FOR ANGZ=-PAI/2+PAI/DVN TO PAI/2-PAI/DVN STEP PAI/DVN
5396      R=RR*COS(ANGZ) : N=INT(R*RT/1.5+20) : BB=2*PAI/N
5398      ZC=ZZC+RR*SIN(ANGZ) : GOSUB 4600
5399  NEXT ANGZ : RETURN
5400  '
5405  FOR I=1 TO NN
5410      X(I)=X(I)+XXM : Y(I)=Y(I)+YYM : Z(I)=Z(I)+ZZM
5420  NEXT I
5430  RETURN

```

```

5500 '***** ROTATION ABOUT X-SHAFT (2nd) *****
5510 FOR I=1 TO NN
5520   Y=Y(I) : Z=Z(I)
5530   Y(I)=Z*SIN(SKX)+Y*COS(SKX)+YC : Z(I)=Z*COS(SKX)-Y*SIN(SKX)+ZC : X(I)=X(I)+X
5540 NEXT I
5550 RETURN
5600 '***** ROTATION ABOUT Y-SHAFT (2nd) *****
5610 FOR I=1 TO NN
5620   X=X(I) : Z=Z(I)
5630   X(I)=X*COS(SKY)-Z*SIN(SKY) : Z(I)=X*SIN(SKY)+Z*COS(SKY)
5640   X(I)=X(I)+XC : Y(I)=Y(I)+YC : Z(I)=Z(I)+ZC
5650 NEXT I
5660 RETURN
5700 '***** ROTATION ABOUT Z-SHAFT (2nd) *****
5710 FOR I=1 TO NN
5720   X=X(I) : Y=Y(I)
5730   X(I)=X*COS(SKZ)-Y*SIN(SKZ) : Y(I)=X*SIN(SKZ)+Y*COS(SKZ)
5740   X(I)=X(I)+XC : Y(I)=Y(I)+YC : Z(I)=Z(I)+ZC
5750 NEXT I
5760 RETURN
5770 '***** ROTATION ABOUT Y-SHAFT (2nd-V2) *****
5775 FOR I=1 TO NN
5780   X(I)=X(I)+XC : Y(I)=Y(I)+YC : Z(I)=Z(I)+ZC : X=X(I) : Z=Z(I)
5785   X(I)=X*COS(SKY)-Z*SIN(SKY) : Z(I)=X*SIN(SKY)+Z*COS(SKY)
5790 NEXT I
5795 RETURN
5800 '***** ROTATION ABOUT X-SHAFT (S) *****
5805 FOR I=1 TO NN
5810   SXR=40 : SXR=P*SXR
5815   ZZ=Z(I) : YY=Y(I)
5820   Y(I)=ZZ*SIN(SXR)+YY*COS(SXR) : Z(I)=ZZ*COS(SXR)-YY*SIN(SXR)
5825 NEXT I
5830 RETURN
5850 '***** ROTATION ABOUT Y-SHAFT (S) *****
5855 FOR I=1 TO NN
5860   XX=X(I) : ZZ=Z(I)
5865   X(I)=XX*COS(SKY)-ZZ*SIN(SKY) : Z(I)=XX*SIN(SKY)+ZZ*COS(SKY)
5870 NEXT I
5875 RETURN
5880 '***** ROTATION ABOUT ANY-SHAFT *****
5882 FOR I=1 TO NN
5884 XX=X(I) : YY=Y(I) : ZZ=Z(I)
5886 XXPX=XX-PX : YYPY=YY-PY : ZZPZ=ZZ-PZ : L=1-COS(T)
5888 X(I)=XXPX*(L*COS(A)*COS(A)+COS(T))+YYPY*(L*COS(A)*COS(B)-CC*SIN(T)+ZZPZ*(
COS(A)*CC+COS(B)*SIN(T))+PX
5890 Y(I)=XXPX*(L*COS(A)*COS(B)+CC*SIN(T))+YYPY*(L*COS(B)*COS(B)+COS(T))+ZZPZ*(
COS(B)*CC-COS(A)*SIN(T))+PY
5892 Z(I)=XXPX*(L*COS(A)*CC-COS(B)*SIN(T)+YYPY*(L*COS(B)*CC+COS(A)*SIN(T))+ZZPZ
(L*CC*CC+COS(T))+PZ
5894 NEXT
5896 RETURN

```



```

5900 '***** CIRCLE ABOUT X-SHAFT (1) *****
5910 FOR SXR=0 TO 2*PAI STEP PAI/DVN
5920   FOR I=INT(AS/BB) TO INT(AE/BB)+1
5930     IF I=INT(AS/BB) THEN AA=AS : GOTO 5960
5940     IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 5960
5950     AA=BB*I
5960     ZC2=R*SIN(AA)+ZC : XC2=R*COS(AA)
5970     X(I)=XC2 : Y(I)=0 : Z(I)=ZC2
5980     Y(I)=Z(I)*SIN(SXR)+YC : Z(I)=Z(I)*COS(SXR)+ZC : X(I)=X(I)+XC
5990     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6000     IF AA=AS THEN GOSUB 3900 : GOTO 6020
6010     GOSUB 3940
6020   NEXT I
6030 NEXT SXR
6040 RETURN
6100 '***** CIRCLE ABOUT Y-SHAFT (1) *****
6110 FOR SYR=0 TO 2*PAI STEP PAI/DVN
6120   FOR I=INT(AS/BB) TO INT(AE/BB)+1
6130     IF I=INT(AS/BB) THEN AA=AS : GOTO 6160
6140     IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 6160
6150     AA=BB*I
6160     YC2=R*SIN(AA) : XC2=R*COS(AA)+XC
6170     X(I)=XC2 : Y(I)=YC2 : Z(I)=0
6180     Z(I)=X(I)*COS(SYR)+ZC : X(I)=X(I)*SIN(SYR)+XC : Y(I)=Y(I)+YC
6190     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6200     IF AA=AS THEN GOSUB 3900 : GOTO 6220
6210     GOSUB 3940
6220   NEXT I
6230 NEXT SYR
6240 RETURN
6300 '***** CIRCLE ABOUT Z-SHAFT (1) *****
6310 FOR SZR=0 TO 2*PAI STEP PAI/DVN
6320   FOR I=INT(AS/BB) TO INT(AE/BB)+1
6330     IF I=INT(AS/BB) THEN AA=AS : GOTO 6360
6340     IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 6360
6350     AA=BB*I
6360     YC2=R*SIN(AA)+YC : ZC2=R*COS(AA)
6370     X(I)=0 : Y(I)=YC2 : Z(I)=ZC2
6380     X(I)=Y(I)*SIN(SZR)+XC : Y(I)=Y(I)*COS(SZR)+YC : Z(I)=Z(I)+ZC
6390     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6400     IF AA=AS THEN GOSUB 3900 : GOTO 6420
6410     GOSUB 3940
6420   NEXT I
6430 NEXT SZR
6440 RETURN
6500 '***** CIRCLE ABOUT X-SHAFT (2) *****
6510 FOR SXR=0 TO 2*PAI STEP PAI/DVN
6520   FOR I=INT(AS/BB) TO INT(AE/BB)+1
6530     IF I=INT(AS/BB) THEN AA=AS : GOTO 6560
6540     IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 6560
6550     AA=BB*I
6560     XC2=R*COS(AA) : YC2=R*SIN(AA)
6570     X(I)=XC2 : Y(I)=YC2 : Z(I)=ZC : YY=Y(I)
6580     Y(I)=Z(I)*SIN(SXR)+YY*COS(SXR)+YC : Z(I)=Z(I)*COS(SXR)-YY*SIN(SXR)+ZC
: X(I)=X(I)+XC

```

```

6590      GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6600      IF AA=AS THEN GOSUB 3900 : GOTO 6620
6610      GOSUB 3940
6620      NEXT I
6630 NEXT SXR
6640 RETURN
6700 '○○○○○○○ CIRCLE ABOUT Y-SHAFT (2) ○○○○○○○
6710 FOR SYR=0 TO 2*PAI STEP PAI/DVN
6720     FOR I=INT(AS/BB) TO INT(AE/BB)+1
6730         IF I=INT(AS/BB) THEN AA=AS : GOTO 6760
6740         IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 6760
6750         AA=BB*I
6760         XC2=R*COS(AA) : YC2=R*SIN(AA)
6770         X(I)=XC2 : Y(I)=YC2 : Z(I)=ZC : ZZ=Z(I)
6780         Z(I)=ZZ*COS(SYR)-X(I)*SIN(SYR)+ZC : X(I)=ZZ*SIN(SYR)+X(I)*COS(SYR)+XC
: Y(I)=Y(I)+YC
6790         GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6800         IF AA=AS THEN GOSUB 3900 : GOTO 6820
6810         GOSUB 3940
6820     NEXT I
6830 NEXT SYR
6840 RETURN
6900 '○○○○○○○ CIRCLE ABOUT Z-SHAFT (2) ○○○○○○○
6910 FOR SZR=0 TO 2*PAI STEP PAI/DVN
6920     FOR I=INT(AS/BB) TO INT(AE/BB)+1
6930         IF I=INT(AS/BB) THEN AA=AS : GOTO 6960
6940         IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 6960
6950         AA=BB*I
6960         YC2=R*SIN(AA) : ZC2=R*COS(AA)
6970         X(I)=XXC : Y(I)=YC2 : Z(I)=ZC2 : XX=X(I)
6980         X(I)=XX*COS(SZR)-Y(I)*SIN(SZR)+XC : Y(I)=XX*SIN(SZR)+Y(I)*COS(SZR)+YC
: Z(I)=Z(I)+ZC
6990         GOSUB 3160 : GOSUB 3260 : GOSUB 3650
7000         IF AA=AS THEN GOSUB 3900 : GOTO 7020
7010         GOSUB 3940
7020     NEXT I
7030 NEXT SZR
7040 RETURN
7100 '○○○○○○○ ROTATION OF CIRCLE ○○○○○○○
7110 FOR T=0 TO 2*PAI STEP PAI/DVN
7120     FOR I=INT(AS/BB) TO INT(AE/BB)+1
7130         IF I=INT(AS/BB) THEN AA=AS : GOTO 7160
7140         IF I=INT(AE/BB)+1 THEN AA=AE : GOTO 7160
7150         AA=BB*I
7160         YC2=R*SIN(AA) : XC2=R*COS(AA)
7170         X(I)=XC2 : Y(I)=YC2 : Z(I)=0
7180         Z(I)=X(I)*COS(SZR)+ZC : X(I)=X(I)*SIN(SZR)+XC : Y(I)=Y(I)+YC
7190         ZZ=Z(I) : Z(I)=Z(I)*COS(T)-Y(I)*SIN(T)
7200         Y(I)=ZZ*SIN(T)+Y(I)*COS(T)
7210         GOSUB 3160 : GOSUB 3260 : GOSUB 3650
7220         IF AA=AS THEN GOSUB 3900 : GOTO 7240
7230         GOSUB 3940
7240     NEXT I
7250 NEXT T
7260 RETURN

```

```

7300 '##### HELIX-FORM ON THE Y-PLANE #####
7310   FOR I=INT(AS/BB) TO INT(AE/BB)+N*MKS
7320     IF I=INT(AS/BB) THEN AA=AS : GOTO 7350
7330     IF I=INT(AE/BB)+1+N*MKS THEN AA=AE+2*PAI*MKS : GOTO 7350
7340     AA=BB*I
7350     ZC2=R*SIN(AA) : XC2=R*COS(AA)
7360     SX2=XC+XC2:SZ2=ZC+ZC2:X(I)=SX2:Y(I)=AA/(2*PAI)*PT+YC:Z(I)=SZ2
7370     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
7380     IF AA=AS THEN GOSUB 3900 : GOTO 7400
7390     GOSUB 3940
7400   NEXT I
7410 RETURN
7500 '##### ROTATION ABOUT Y-SHAFT (3rd) #####
7520 FOR I=1 TO NN
7530   X=X(I) : Z=Z(I)
7540   X(I)=X*COS(SKY)-Z*SIN(SKY) : Z(I)=X*SIN(SKY)+Z*COS(SKY)
7550   Y(I)=Y(I)+20*NN1
7560 NEXT I
7570 RETURN
7600 '##### CIRCLE ON THE Y-PLANE (2) #####
7610 FOR I=INT(AS/BB) TO INT(AE/BB)+1
7620   AA=BB*I
7630   IF I=INT(AS/BB) THEN AA=AS
7640   IF I=INT(AE/BB)+1 THEN AA=AE
7650   ZC2=R*SIN(AA) : XC2=R*COS(AA)
7660   SX2=XC+XC2 : SZ2=ZC+ZC2 : SSX2=SX2
7670   SX2=SX2*COS(J)-SZ2*SIN(J) : SZ2=SSX2*SIN(J)+SZ2*COS(J)
7680   X(I)=SX2 : Y(I)=YC : Z(I)=SZ2
7690   GOSUB 3160 : GOSUB 3260 : GOSUB 3650
7700   IF AA=AS THEN GOSUB 3900 : GOTO 7720
7710   GOSUB 3940
7720 NEXT I
7730 RETURN
7800 '##### CIRCLE ON THE X-PLANE (2) #####
7810 FOR I=INT(AS/BB) TO INT(AE/BB)+1
7820   AA=BB*I
7830   IF I=INT(AS/BB) THEN AA=AS
7840   IF I=INT(AE/BB)+1 THEN AA=AE
7850   YC2=R*SIN(AA) : ZC2=R*COS(AA)
7860   SZ2=ZC+ZC2 : SY2=YC+YC2 : SSZ2=SZ2
7870   SZ2=SZ2*COS(J)-SY2*SIN(J) : SY2=SSZ2*SIN(J)+SY2*COS(J)
7880   Z(I)=SZ2 : X(I)=XC : Y(I)=SY2
7890   GOSUB 3160 : GOSUB 3260 : GOSUB 3650
7900   IF AA=AS THEN GOSUB 3900 : GOTO 7920
7910   GOSUB 3940
7920 NEXT I
7930 RETURN

```


4 . D A T A 관리 프로그램

4.1 DATA 관리 프로그램

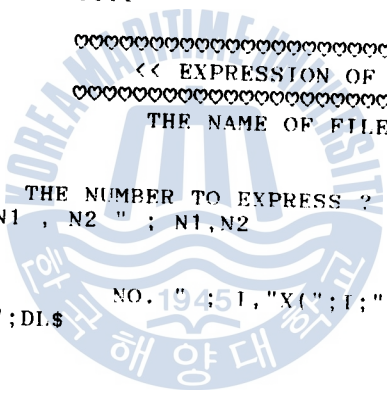
입체물의 각 정점, 즉 데이터의 총합적인 관리가 가능한 프로그램을 생각해 보자. 1 개의 프로그램에 화일의 신규작성 (NEW MAKING OUT) , 데이터 추가 (ADDITION OF DATA) , 삭제 (DELETION OF DATA) , 치환 (SUBSTITUTION OF DTA) , 삽입 (INSERTION OF DATA) , 표현 (EXPRESSION OF DATA) 이 가능할 수 있도록 한 프로그램을 다음 페이지에 보인다.




```

940 OPEN F$ FOR INPUT AS #1
950 OPEN "FNNEW" FOR OUTPUT AS #2
960 IF EOF(1) THEN 1050
970 INPUT #1,X,Y,Z,DL$
980 IF I<>N1 THEN 1020
990 PRINT : PRINT "DATA TO INSERT : X(";I;"), Y(";I;"), Z(";I;"), P/*(";I;") " ;
: INPUT XI,YI,ZI,DLI$
1000 WRITE #2,XI,YI,ZI,DLI$
1010 I = I+1
1020 WRITE #2,X,Y,Z,DL$
1030 I = I+1
1040 GOTO 960
1050 CLOSE
1060 KILL F$
1070 NAME "FNNEW" AS F$
1080 PRINT : INPUT " DO YO INTEND TO CONTINUE (Y/N) " ; YN$
1090 IF YN$="Y" OR YN$="y" THEN 920 ELSE RETURN
1100 '***** EXPRESSION OF DATA ! *****
1101 CLS 1 : PRINT
1105 PRINT "
1110 PRINT " *****
1111 PRINT " << EXPRESSION OF DATA >> "
1120 PRINT : PRINT : INPUT " *****
1130 OPEN F$ FOR INPUT AS #1 THE NAME OF FILE " ; F$
1140 I = 1
1150 PRINT : PRINT " THE NUMBER TO EXPRESS ? FROM N1 TO N2."
1160 INPUT " INPUT N1 , N2 " ; N1,N2
1170 IF EOF(1) THEN 1230
1180 INPUT #1,X,Y,Z,DL$
1190 IF I>=N1 THEN PRINT " NO. ";I,"X(";I;")=";X;,"Y(";I;")=";Y;,,,
,"Z(";I;")=";Z;,,, "P/*(";I;")=";DL$
1200 IF I=N2 THEN 1230
1210 I = I+1
1220 GOTO 1170
1230 CLOSE #1
1240 INPUT " DO YOU INTEND TO CONTINUE (Y/N) " ; YN$
1250 IF YN$="Y" OR YN$="y" THEN 1130 ELSE RETURN

```



4.2 SEQUENTIAL FILE 을 사용한 입체도

이렇게 해서 만든 SEQUENTIAL FILE 은 입체도의 데이터로서 사용되어질 수가 있다. 다음 프로그램은 SEQUENTIAL FILE 로 부터 데이터를 읽어내어 입체도를 그려내는 프로그램이다. SUBROUTINE 은 3 장에서 예시된 SUBROUTINE PROGRAM 을 사용한다. 프로그램을 실행시키면 투영도의 종류, 원점 지정, 배율등을 지정한 후 FILE NAME 을 키-인하면 해당 FILE 의 데이터를 사용한 입체도를 얻을 수 있다.

```
1 *****
2 *****
3   PROGRAM NAME : MAIN PROGRAM (FOR DATA CONTROL PROGRAM)
4 *****
5 *****
9  CHAIN MERGE "SUB1" ,ALL,DELETE 9
10 GOSUB 2100 : GOSUB 2900
20 INPUT " INPUT THE DATA-FILE NAME : " ; F$
30 OPEN F$ FOR INPUT AS #1
40 IF EOF(1) THEN 800
50 INPUT #1,X,Y,Z,DL$
60 X(1)=X : Y(1)=Y : Z(1)=Z : GOSUB 3160 : GOSUB 3260 : GOSUB 3650
70 IF DL$="P" THEN GOSUB 3900 : GOTO 40
80 GOSUB 3940
90 GOTO 40
800 CLOSE #1
950 END
960 DATA 100
```

5. 은선처리 방법

5.1 최대, 최소치의 비교에 의한 방법

입체도형을 작도할 때 보이지않는 선을 소거해서 입체같은 그림을 그리는 것은 은선처리(HIDDEN LINE ELIMINATION)라 한다. 퍼스날 컴퓨터에서는 여러가지 은선처리 방법이 사용되어지고 있지만, 여기에서는 화면상에 있어서 임의의 X 값에 대해 Y의 값이 이미 입력이 끝난 Y의 최대치나 최소치와 비교해서 작도할까 만가를 판정한다.

화면에서는 x 방향의 최대치로서 639, 또 y 방향에 399를 선정할 수가 있다. 그러나 X 좌표치의 작도범위를 0 부터 630, Y 좌표치의 최대치를 0, 최소치를 390으로 정하고, [120]에서 [140]까지로서, 우선 초기치를 입력한다. 다음에 [2090]과 [2100]에 의해서 화면의 좌표치가 계산되어지면 아래 그림에서와 같이, X의 i 번째에 있어서 Y의 i 번째의 계산치와 이미 입력이 되어있는 최대치 또는 최소치와의 비교를 행하고 다음의 세가지 경우로 나누어 처리한다. 아래의 (A), (B), (C)에 대응하는 설명을 그림 (a)에, 그 실행예의 약도를 (b)에 표시한다.

(A) 프로그램의 [5210]에서는 Y의 값이 최대치보다 큰 경우로서 이미 입력이 되어있는 최대치와 새로운 Y의 값을 바꾸고 [6000]으로 뛰어 작도하라는 명령.

(B) [5220]에서는 Y 값이 최소치보다 작은 경우로서 새로운 Y 값이 최소치가 되고 최소치를 새로운 Y의 계산치로 교체한 다음 [6000]으로 뛰어 작도하라는 명령.

(C) [5230]에서는 Y 값이 최대치보다 작고, 최소치보다 큰 경우로서 그림에서와 같이 은선이 되는 부분이기 때문에 작도하지 말고 H에 1을 대입해서 [6040]으로 가라는 명령.

이상, (A), (B), (C)의 처리에 의해 간단히 은선처리가 가능하다. 다만, 이 수법을 사용하는 경우에는 작도화면에 있어서 도형의 형태를 잘 나타낼 수 있는 부분을 그릴 수 있도록 프로그램할 필요가 있다. 게다가, 어떤 X의 값에 대해 Y의 값이 2개 이상 존재할 때에는 적용이 불가능하다.

```

1 '=====
2 ' PROGRAM NAME : HIDDEN-LINE ELIMINATION (MAXIMUM,MINIMUM METHOD)
3 '=====
4 '
5 '----- FIRST - VALUE DEFINITION -----
10 CLS 3 : CONSOLE 0,25,0,1 : SCREEN 3,0
20 PI=3.14159
30 T=PI/180
40 TX=-65*T : TY=30*T : TZ=10*T
50 N=0 : CL=0
60 X0=130 : Y0=150
70 DIST = -1
80 DIM MIN(630),MAX(630)
100 '----- MAIN PROGRAM -----
110 GOSUB 1000
120 FOR I=0 TO 630
130 MIN(I)=390 : MAX(I)=0
140 NEXT I
150 N=N+1 : CL=CL+2
160 FOR Y=-80 TO 80 STEP 5 : M=1
170 FOR X=-80 TO 80 STEP 2
180 GOSUB 2000
190 NEXT X,Y
200 ON N GOTO 210,220,230
210 X0=320 : GOTO 120
220 X0=510 : GOTO 120
230 END
240 '
1000 '----- ROTATION -----
1010 XA=COS(TY)*COS(TZ)
1020 YA=SIN(TX)*SIN(TY)*COS(TZ)-COS(TX)*SIN(TZ)
1030 ZA=COS(TX)*SIN(TY)*COS(TZ)+SIN(TX)*SIN(TZ)
1040 XB=COS(TY)*SIN(TZ)
1050 YB=SIN(TX)*SIN(TY)*SIN(TZ)+COS(TX)*COS(TZ)
1060 ZB=COS(TX)*SIN(TY)*SIN(TZ)-SIN(TX)*COS(TZ)
1070 RETURN
1080 '
2000 '----- BASE EQUATION -----
2010 FX=X
2020 FY=Y
2030 FZ=-SQR(X*X+Y*Y)
2040 IF FZ<=-70 THEN FZ=-70
2050 ON N GOTO 2090,2060,2080
2060 IF FX<=-10 THEN FZ=-70
2070 GOTO 2090
2080 IF FZ>=-40 THEN FZ=-40
2090 X2=XA*FX+YA*FY+ZA*FZ +X0
2100 Y2=(XB*FX+YB*FY+ZB*FZ)*DIST+Y0
2110 '
5200 '----- HIDDEN LINE ELIMINATION -----
5210 IF Y2>MAX(X2) THEN MAX(X2)=Y2 : GOTO 6000
5220 IF Y2<MIN(X2) THEN MIN(X2)=Y2 : GOTO 6000
5230 M=1 : RETURN
5240 '
6000 '----- DRAWING -----
6010 IF M=1 THEN M=2 : GOTO 6030
6020 LINE (X1,Y1)-(X2,Y2),CL
6030 X1=X2 : Y1=Y2
6040 RETURN

```

<<EXAMPLE 2>>

HIDDEN-LINE ELIMINATION (MAXIMUM-MINIMUM METHOD)



5.2 PAINTING에 의한 방법

PAINT 문을 사용한 간단한 은선처리에 대해서 설명한다.

처리(1)

우선 그림(1)에 보인대로 배색과는 별도의 색을 사용해서 도형의 윤곽선의 색 C_L 및 그 내부색 C_A 를 같은 색으로 착색한다. 그림에서는 배색이 흑(0), C_L 과 C_A 는 녹(4)를 사용하고 있다.

처리(2)

다음에 작도하고 싶은 색의 번호를 C_L 에 지정하고 C_A 에는 배색과 같은 번호를 대입하면 이미 착색한 색의 위에 지정색이 PAINT 되어지고 겹쳐진 부분에서는 전의 색이 소거되어져서 윤곽선만이 남는다.

처리(3)

그림(3),(4)에는 이미 설명한 수법(1),(2)를 되풀이해서 실행한 상태를 표시하고 있다. 작도할 때에 이렇게 필요한 횟수만 처리(1),(2)를 되풀이해서 적용하면 간단히 은선처리가 가능하다.

단, 이 은선처리 방법을 적용할 경우에는 필히 시점으로부터 먼 부분의 도형을 처음으로 작도하고 순차적으로 가까운 도형을 작도하도록 프로그램하지 않으면 안된다.

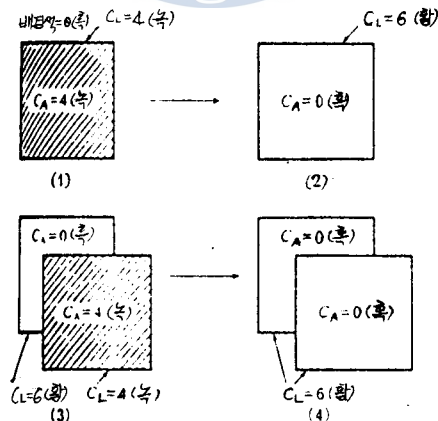


図 8-4 PAINTによる除線処理

```

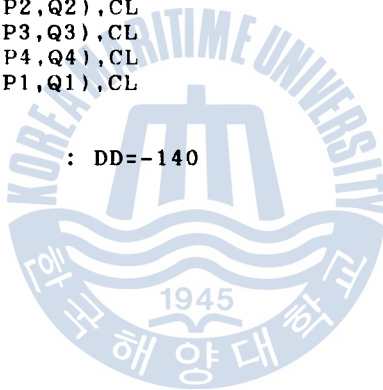
1 '=====
2 ' PROGRAM NAME : HIDDEN-LINE ELIMINATION (PAINTING METHOD) PROGRAM
3 '=====
4 '
5 '----- FIRST - VALUE DEFINITION -----
10 CLS 3 : CONSOLE 0,25,0,1 : SCREEN 3,0
20 P1=9      : P2=11
30 PI=3.14159
40 T=PI/180
50 TX=-65*T : TY=0      : TZ=32*T
60 SF=1.5
70 X0=145   : Y0=315
80 DIST=-1
82 R1=62    : R2=48     : RB=12
84 NN=0
86 DIM SX(P1,P2),SY(P1,P2)
100 '----- MAIN PROGRAM -----
105 NN=NN+1 : ON NN GOTO 110,130,140,170,180,210,240
110 GOSUB 1000 : TB1=-.5 : TB2=1.51 : TB3=1
120 TA1=.5     : TA2=1.1 : TA3=1     : GOTO 220
130          : TA2=-.1 : TA3=-1    : GOTO 220
140 TX=25*T   : X0=255  : Y0=245
150 GOSUB 1000 : TB1=0   : TB2=2.1
160          : TA2=-.51 :          : GOTO 220
170          : TA2=1.51 : TA3=1    : GOTO 220
180 TX=-65*T  : X0=145  : Y0=315
190 GOSUB 1000 : TB1=1.5 : TB2=-.51 : TB3=-1
200 TA1=1     :          : GOTO 220
210 TA1=2     :          : GOTO 220
220 GOSUB 2000 : GOSUB 6000
230 GOTO 105
240 END
250 '
1000 '----- ROTATION -----
1010 XA=COS(TY)*COS(TZ)
1020 YA=SIN(TX)*SIN(TY)*COS(TZ)-COS(TX)*SIN(TZ)
1030 ZA=COS(TX)*SIN(TY)*COS(TZ)+SIN(TX)*SIN(TZ)
1040 XB=COS(TY)*SIN(TZ)
1050 YB=SIN(TX)*SIN(TY)*SIN(TZ)+COS(TX)*COS(TZ)
1060 ZB=COS(TX)*SIN(TY)*SIN(TZ)-SIN(TX)*COS(TZ)
1070 RETURN
1080 '
2000 '----- BASE EQUATION -----
2010 M=0
2020 FOR TA=TA1*PI TO TA2*PI STEP TA3*PI/8 : M=M+1
2030 N=0 : X1=R1*COS(TA) : Y1=R2*SIN(TA)
2040 RA=SQR(X1*X1+Y1*Y1)
2050 FOR TB=TB1*PI TO TB2*PI STEP TB3*PI/5 : N=N+1
2060 RR=RA+RB*COS(TB)
2070 FX=RR*COS(TA) : FY=RR*SIN(TA)
2080 FZ=RB*SIN(TB) : GOSUB 2120
2090 SX(M,N)=XX : SY(M,N)=YY
2100 NEXT TB,TA : RETURN
2110 '
2120 XX=(XA*FX+YA*FY+ZA*FZ)*SF+X0
2130 YY=(XB*FX+YB*FY+ZB*FZ)*DIST*SF+Y0
2140 RETURN

```

```

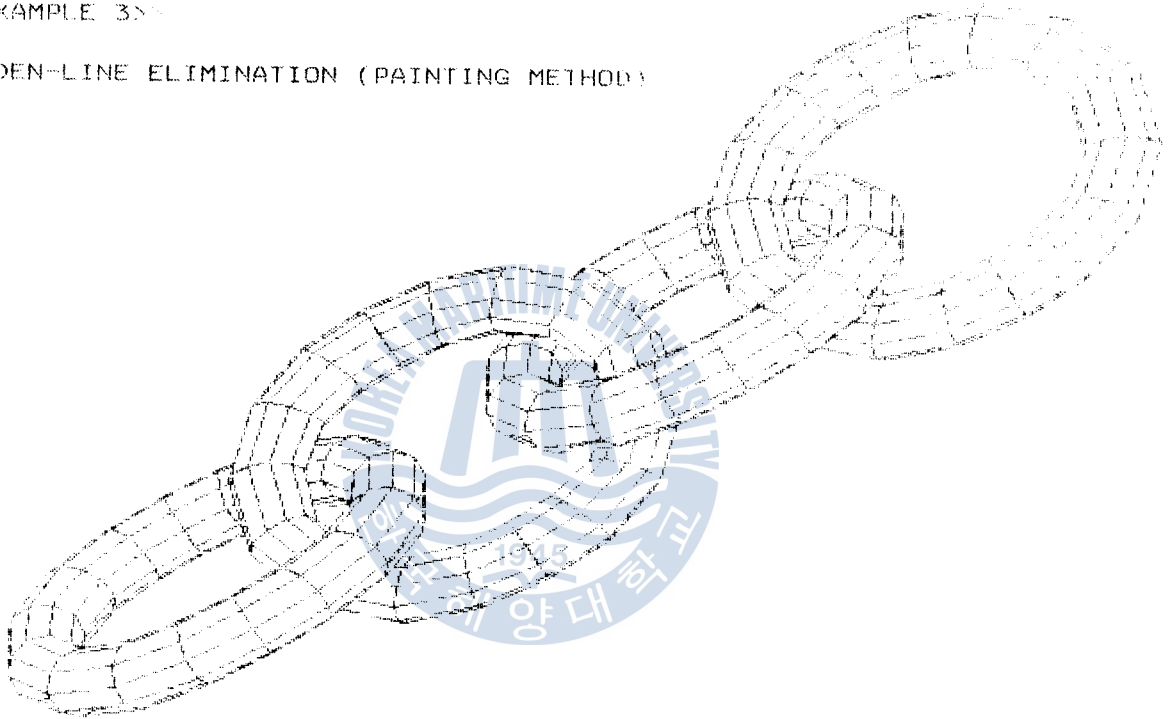
2150 '
5500 '----- HIDDEN LINE ELIMINATION -----
5510 PAINT (PP,QQ),CA,CL
5520 IF CN=2 THEN 6160
5530 CN=2      : CL=6      : CA=0      : GOTO 6110
5540 '
6000 '----- DRAWING -----
6010 CC=0      : DD=0
6020 FOR K=1 TO 2
6030 BB=5              : IF NN=3 OR NN=4 THEN BB=9
6040 FOR I=2 TO BB    : FOR J=2 TO 11
6050 CN=1      : CL=2      : CA=2
6060 P1=SX(I,J)+CC    : Q1=SY(I,J)+DD
6070 P2=SX(I,J-1)+CC  : Q2=SY(I,J-1)+DD
6080 P3=SX(I-1,J-1)+CC : Q3=SY(I-1,J-1)+DD
6090 P4=SX(I-1,J)+CC  : Q4=SY(I-1,J)+DD
6100 PP=(P1+P2+P3+P4)/4 : QQ=(Q1+Q2+Q3+Q4)/4
6110 LINE (P1,Q1)-(P2,Q2),CL
6120 LINE (P2,Q2)-(P3,Q3),CL
6130 LINE (P3,Q3)-(P4,Q4),CL
6140 LINE (P4,Q4)-(P1,Q1),CL
6150 GOTO 5500
6160 NEXT J,I
6170 CC=220          : DD=-140
6180 NEXT K
6190 RETURN

```



EXAMPLE 355

HIDDEN-LINE ELIMINATION (PAINTING METHOD)



6 . 결 론

이상과 같이 PERSONAL COMPUTER 를 이용한 3 차원 입체 그래픽스에 대해, 데이터 관리 프로그램으로 작성된 SEQUENTIAL FILE 로 자동적으로 원점이동, SCALING , 평행이동, 대칭변환, 회전이동이 될 수 있는 하나의 팩키지를 작성하였다.

데이터 관리 프로그램을 통해 모든 데이터들이 간편하고 일목요연하게 작성, 수정, 즉 편집이 될 수 있다는 점과, MAIN PROGRAM과 SUBROUTINE PROGRAM의 유기적인 연결이 본 연구의 특징이다.

3 차원 입체도형 역시 2 차원 평면에 대한 확실한 이해가 바탕이 되어야 하며 서론에서도 언급이 되었었지만 프로그래밍 언어에 대해서도 기본적인 바탕을 전제로 해야만 할 것으로 본다.

실제 은선처리의 문제에 있어서는 이 두 가지 방법 이외에도 외적에 의한 방법, 다각형의 은면 검출에 의한 방법등 여러 방법이 있으나 그 중 간단한 두 방법을 골랐다.

참고로 MAIN PROGRAM과 SUBROUTINE PROGRAM의 연결을 위해 액세스 타임이 많이 걸리나, 화일 결합을 위해 애스키 화일로 세이브시켰다는 것을 밝힌다.

지면관계상 설명이 부족한 곳도 많이 있고, 합리적이지 못한 프로그래밍도 존재 하리라 생각하나, 더 나은 진전의 계기로 삼으려한다.

참고문헌

- (1) 安居院猛・中嶋正之 共著；“パソコン・グラフィクス入門”
- (2) 福永節夫外 3名；“パソコンによる作図の基礎”
- (3) 竹内芳美；“パソコン CAD/CAM”
- (4) 千代倉 弘明；“ソリッドモデリング”
- (5) 山口富士夫；“図形処理工学”
- (6) 최 병규；“NC절삭가공과 CAM시스템”
- (7) 永山かしょう；“3차원그래픽스入門”
- (8) 박 명규；“図形情報技術”
- (9) 정 선모；“CAD/CAM概論”

教育用 振動프로그램 基礎理論
解析에 關한 研究



지도교수 : 김 의 간

1990년 2월 19일

선박기계공학과 4학년

강병수 고명진 김광이
한용대 권영명 황병달