컴퓨터에 의한 透視図 作図

曺 鐵 鎬 – 건축사 · 建国大교수

PERSPECTIVE BY CAD SYSTEM

CHUL–HO CHO / KONKUK UNIV.

어떤 간단한 建物을 컴퓨터에 의해 透視図를 作図할 수 있도록 간단하게 나타낸 기본개념도가 『그림－32』이다.

14. 任意의 觀測点에 대한 隱線除去

이 방법은 Roy E. Myers에 의한 것이다.

앞에서 설명한 두 隱線除去 방법들은 관측점을 설정하는 데 제약점이 있었다. 보다 일반적인 隱線除去方法은 面의 방향을 이용하여 얻을 수 있다.

『그림－33』에서 (a)와 같은 관측각도에서 보면 피라미드의 일부는 가리워지면서 정육면체가 보다 앞에 있는 것 처럼 보인다. 『그림－33(b)』에서는 피라미드가 정육면체의 일부를 가리우고 있다. 여기서 문제가 되는 것은 건물들 각각에 우선 순위(Priority)를 결정해 주는 것이다. 두 건물이 画像으로 나타날 경우 다른 건물에 의해 가리워지지 않는 건물에 높은 순위를 부여해야 할 것이다. 부분적으로 가리워진 건물은 두번째 순위를 갖게 된다. 우선 순위는 관측점으로부터 각 건물의 중심까지의 거리를 서로 비교하여 가까운 곳에 있는 건물로부터 높은 순위를 부여할 수 있다.

우선 순위를 부여하는 두번째 방법으로는 두 물체를 분리시키는 分離平面을 이용하는 방법이다. 높은 순위는 分離平面에 대하여 관측점과 같은 쪽에 있는 건물에 부여한다. 건물들을 분리할 수 있는 平面은 어떤 것이든지 분리平面에 대하여 관측점과 같은 쪽에 있는 건물에 부여한다.

건물들을 분리할 수 있는 平面은 어떤 것이든지 분리 평면으로 선택할 수 있다. 분리평면은 건물의 한 점이나 한 辺, 또는 한 面을 포함할 수도 있다.

만약 분리 평면을 바꾸면 건물들의 우선 순위도 바뀔 수 있다. 그러나, 이것을 염려할 필요는 없다. 이런 경우에는 서로 상대방을 가리지 않게 되기 때문이다.

일단 우선 순위가 결정되면 먼저 높은 순위를 갖는 건물을 그린다. 여기서는 관측자에게 보여지는 面들만을 그리기 위해 面의 방향을 이용한 방법을 사용하기로 한다.

面의 방향을 이용하는 방법은 2번째 순위의 건물에 대하여서도 관측자 쪽을 향하는 面들을 결정하기 위해 사용될 수 있다. 눈에 보이는 面의 각 辺들은 첫번째 우선 순위를 갖는 물체에 의해 양 끝점들 중의 어떤 것이 가리워지지 않는가를 조사해야 한다.

만약 양 끝점이 모두 가리워지지 않는다면 그 변은 그려진다. 만약 양 끝점이 모두 가리워졌다면 그 변은 무시된다. 만약 어느 한 쪽만 가리워졌다면, 높은 순위 건물에 대한 화상의 경계선과 이 변과의 교점을 계산해야 한다. 다음에 눈에 보이는 부분만을 그리면 된다.

이러한 것을 『그림－33』과 같이 정육면체와 피라미드의 변들에 대해 적용시켜 보면 다음과 같다.

『그림－34(a)』에서는 피라미드의 안 보이는 변은 점선으로 나타낸 것이다.

이러한 변들의 보이는 부분은 보이는 끝점으로부터 정육면체 화상의 외곽 경계선들과의 교점까지이므로, 정육면체의 내부 경계선들을 무시하여 『그림－34(b)』과 같이 단순화시킬 수 있다.

面 배열에 들어갈 값들의 순서를 정의할 때, 정해진 정육면체의 외곽 변들의 방향은 『그림－34(c)』와 같다.

여기에서 정육면체의 외곽 경계선들 안쪽에 놓여 정육면체에 의해 가리워지는 피라미드의 꼭지점 H는 이 외곽경계선 각각에 대해 좌측반평면에 놓이게 됨을 알 수 있다. 가리워지지 않는 꼭지점 V는 어떤 일부 경계선들에 대해서는 좌측 반평면에 놓이게 되고, 나머지 다른 경계선들에 대해서는 우측 반평면에 놓이게 됨을 알 수 있다.

선 VH 중의 보이는 부분을 결정하기 위해서 점 H로부터 점 V로의 線上을 따라 움직이며 각 점에 대하여 可視度検査를 한다. 점 T를 적어도 정육면체의 한 외곽 경계선에 대하여 우측반평면에 놓이게 되는 첫번째 점으로 결정한다. 그러면 점 V로 부터 점 T 까지의 선은 그려질 수 있다.

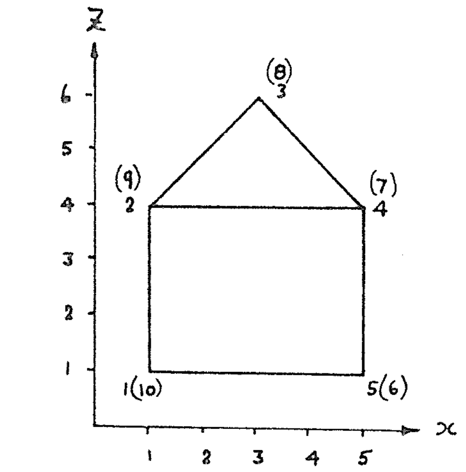
예를 들어서 설명하면 『그림－35』와 같은 건물에 『그림－36』과 같은 굴뚝을 첨가하는 경우로 선정한다.

『그림－35』에서 건물의 좌표는 대개 정하고 『그림－36』의 굴뚝도 마찬가지로 좌표를 정한다.

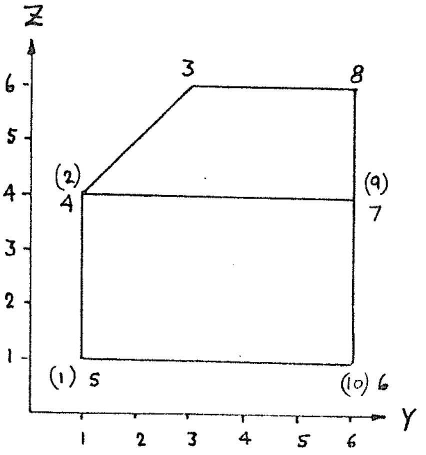
굴뚝은 여섯개의 면들(윗면, 아랫면, 4개의 옆면)로 구성되는데, 각 면은 4개의 변들로 구성되고 이 변들은 양끝점으로 나타내진다.

배열 V(I, J)와 SV(I, J)로서 18개의 점들에 대한 Data를 처리할 수 있도록 D/M V(18, 3), SV(18, 3)로 정할 수 있다. 꼭지점 배열 V(I, J)에 대해서는 앞에서부터 1～18까지 V(I, 1), V(I, 2), V(I, 3)는 각각 I번 꼭지점의 X, Y, Z좌표값을 나타낸다.

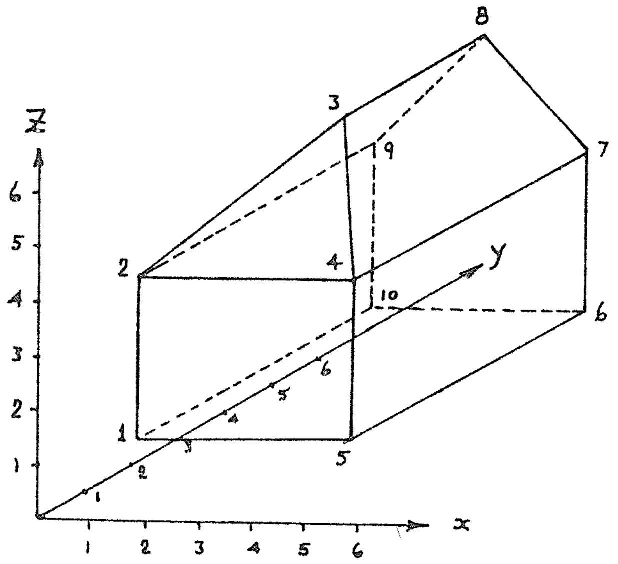
SV(I, 1)과 SV(I, 2)의 값은 V(I, 1), V(I, 2), V(I, 3)이 구해질 때 함께 계산되어 구해진다. 각각의 I에 대하여 SV(I, 1)과 SV(I, 2)는 그 점(X, Y, Z)의 화면 좌표값(Screen Coordinates)을 나타낸다. 세번째 값



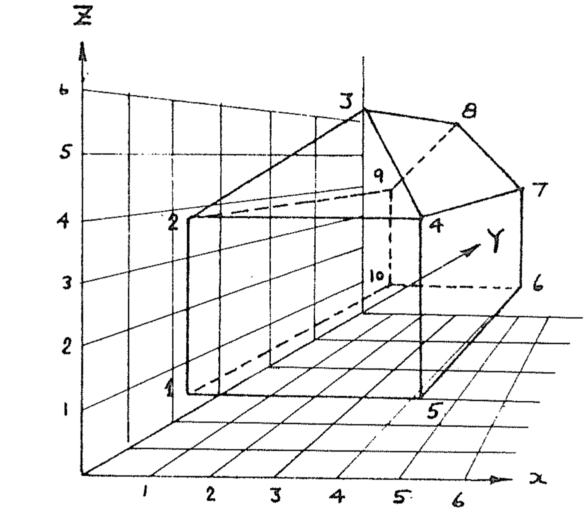
(a) 정면도



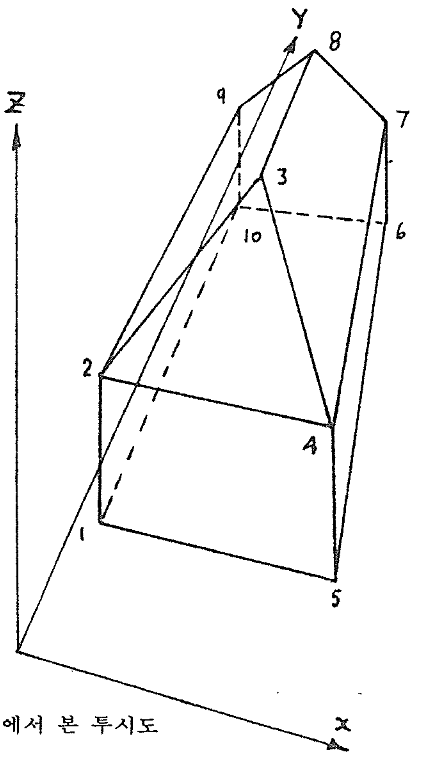
(b) 측면도



(c) Isometric view



(d) 정면에서 본 투시도

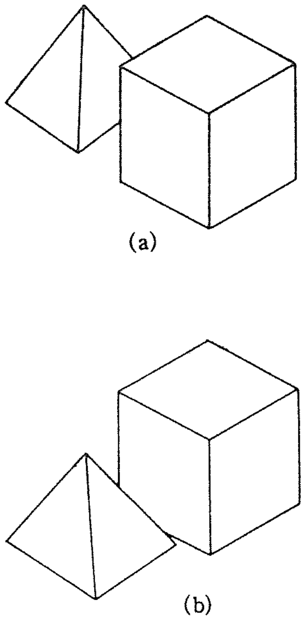


(e) 위에서 본 투시도

Point	Co-ordinates		
	x	y	z
1	1	1	1
2	1	1	4
3	3	3	6
4	5	1	4
5	5	1	1
6	5	6	1
7	5	6	4
8	3	6	6
9	1	6	4
10	1	6	1

(f) 좌표

「그림 - 32」



『그림-33』 관측 각도에 따른 화상의 변화

SV(I, 3)는 초기에는 0으로 되어 있으나, 나중에 어떤 꼭지점들에 대해서는 그들의 可視상태를 明示하기 위해 수정하게 되는 것이다.

〔面配列(Surface Array)〕

각 좌표값에 대하여 꼭지점 번호로 面배열을 구성해야 한다.

다음 표에서와 같은 건물은 面1에서 7까지 결정할 수 있고, 굴뚝에 대해서는 面8에서 13까지 결정할 수 있다.

이 표는 물체의 밖에서 보았을 때 꼭지점들을 반시계 방향 순서로 연결한 것이다. 面8은 굴뚝의 윗면, 面9는 아랫면에 해당하고, 面10, 11, 12, 13은 굴뚝의 옆면들이다.

〔法線 配列(Normal Array)〕

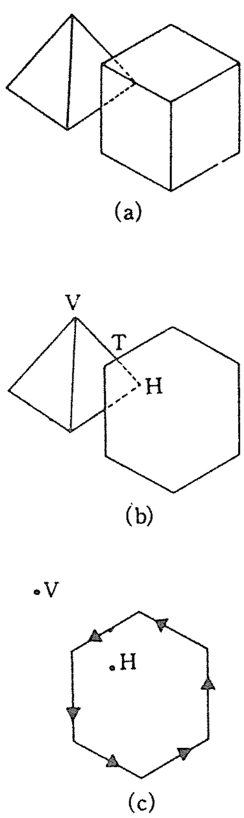
法線 배열은 앞의 방법과 같은 방법으로 구한다. 여기에서 N(I, 1), N(I, 2), N(I, 3)로 I번째 면에서 바깥쪽을 향하는 法線 벡터의 3개의 좌표값을 나타낼 수 있다.

表-1 面의 구성

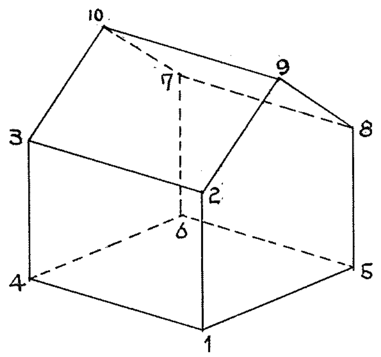
S(I, J)	1	2	3	4	5	6
1	1	2	3	4	1	0
2	1	5	8	9	2	1
3	5	6	7	8	5	0
4	4	3	10	7	6	4
5	3	2	9	10	3	0
6	7	10	9	8	7	0
7	1	4	6	5	1	0
8	11	12	13	14	11	0
9	15	18	17	16	15	0
10	11	14	18	15	11	0
11	12	16	17	13	12	0
12	11	15	16	12	11	0
13	14	13	17	18	14	0

〔辺 配列(Edge Array)〕

辺 배열은 건물이 보이는 12개의 辺들을 결정할 수 있는 공간을 갖도록 해야 한다. 굴뚝의 변들을 나타내기 위해서도 12개의 공간을 마련해 놓아야 한다. 실제로는 어떤 각도에서 굴뚝을 보더라도 9개 이상의 辺들이 보이지 않을 것이다.

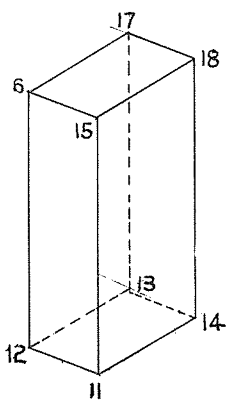


『그림-34』 보이지 않는 꼭지점의 결정과정



꼭지점	좌	표	
1	7.5	10.5	−7.5
2	7.5	10.5	7.5
3	7.5	−10.5	7.5
4	7.5	−10.5	−7.5
5	−7.5	10.5	−7.5
6	−7 5	−10.5	−7.5
7	−7.5	−10.5	7.5
8	−7.5	10.5	7.5
9	0	10.5	12 0
10	0	−10.5	12.0

『그림-35』 건물의 꼭지점과 그 좌표



꼭지점	좌	표	
11	−3.0	−10.5	−7.5
12	−3.0	−13.5	−7.5
13	−7.5	−13.5	−7 5
14	−7.5	−10.5	−7.5
15	−3.0	−10.5	13.5
16	−3.0	−13.5	13.5
17	−7.5	−13.5	13.5
18	−7.5	−10.5	13.5

『그림-36』 굴뚝의 꼭지점과 그 좌표

멀리 있는 건물의 어떤 변들은 가까운 곳의 물체에 의해 부분적으로 가리워 질 수 있으므로, 각 辺들은 일단 건물의 번호에 따라 각각 분리하여 사용하는 것이 편리할 것이다.

그러므로 辺 배열은 E(2, 12, 3)으로 정의해야 한다. 辺 배열에서의 어떤 한 변에 대한 데이터를 찾는데 있어서 E(I, J, k)를 사용하게 된다. 여기에서 I는 1이나 2의 값을 갖는데, 이들은 각각 1번은 건물, 2번은 굴뚝을 지칭하는데 쓰인다. 다음에 J는 1에서 12까지의 값을 가질 수 있는데, 이들은 각각 I에 의해 지정된 물체의 변들을 지칭하는데 사용된다.

건물에 대해서는 J가 12까지의 값을 가질 수도 있으나, 굴뚝에 대해서는 많아야 9의 값을 가진다. I번 물체의 J번 변에 대하여, E(I, J, 1)과 E(I, J, 2)는 그 변의 양 끝점을 나타내는 것이고, E(I, J, 3)는 그 辺을 경계로 갖는 면들 중에 눈에 보이는 면의 수를 나타내는 상태 변수이다.

『그림-37』은 辺 배열에서의 辺의 구성상태를 나타낸다. 이 변 배열의 내용은 선택되는 관측점에 따라 바뀌어진다. 관측점을 바꿈에 따라서 변의 갯수와 배열에서의 변들의 나열순서, 그리고 상태 변수의 값이 변경될 것이다.

이 물체의 각 면들은 가시도 검사를 받아 관측점으로부터 멀어지는 쪽을 향하는 면들은 제거된다. 그리고 눈에 보이는 한 면의 변들을 처리하는 부분이다. 일단 상태변수 E(0, N, 3)은 1로 지정해 두었다가, 만일 어떤 변이 두번 나오면, 그 변이 눈에 보이는 두개의 면들의 경계선이 되면 상태변수 E(0, N, 3)는 2가 되게한다.

1번 물체(건물)의 모든 면에 대하여 가시도 검사가 끝났으면 다음에는 2번 물체(굴뚝)의 면에 대하여 같은 과정을 계속한다.

건물과 굴뚝의 상태 변수의 결과를 나타낸 것이 表-2이다.

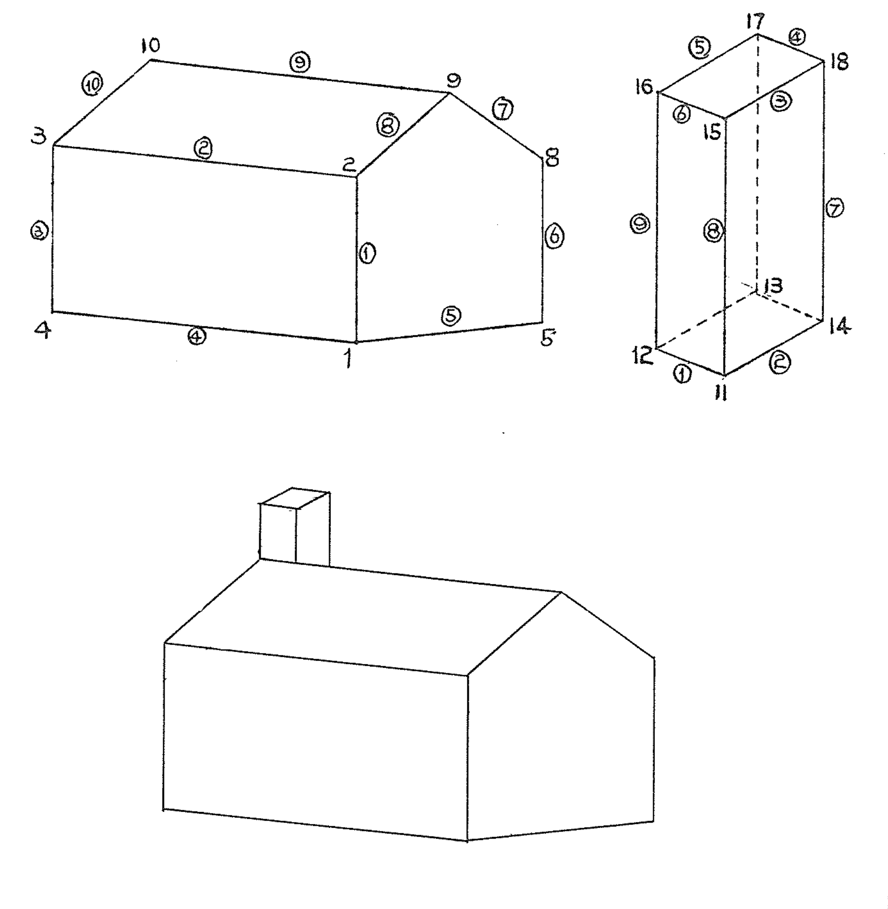
表-2

I=1 : 物体1(建物) I=2 : 物体2(굴뚝)

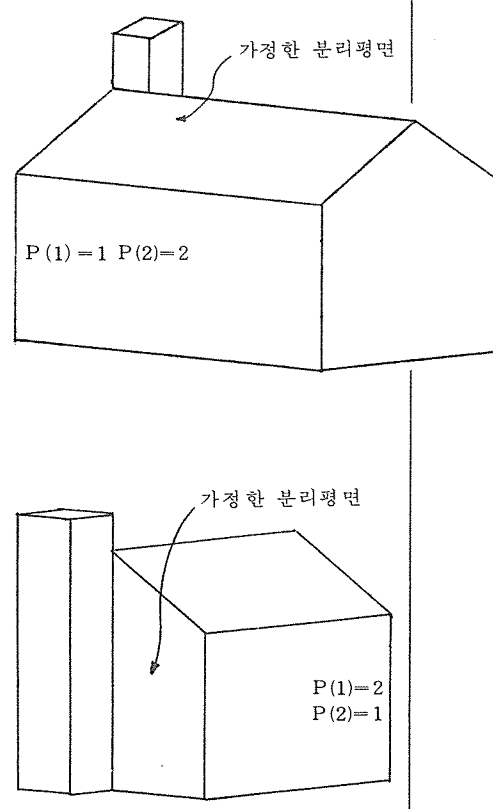
辺	辺의끝점		상태변수	辺	辺의끝점		상태변수
1	1	2	2	1	11	12	1
2	2	3	2	2	14	11	1
3	3	4	1	3	15	18	2
4	4	1	1	4	18	17	1
5	1	5	1	5	17	16	1
6	5	8	1	6	16	15	2
7	8	9	1	7	14	18	1
8	9	2	2	8	15	11	2
9	9	10	1	9	16	12	1
10	10	3	1	10	0	0	0
11	0	0	0	11	0	0	0
12	0	0	0	12	0	0	0

〔物体의 優先順位(Priority)〕

먼 곳의 물체는 가까운 곳의 물체에 의하여 부분적으로 가리워질 수 있으므로, 물체들을 상대적인 위치에 따라 분류할 필요가 있다. 두 물체의 우선순위를 결정하기 위하여 分離平面을 사용하는 것이 좋다. 1번 물체



『그림-37』 변의 구성상태



『그림-38』 관측점과 분리평면에 따라 결정되는 화상

의 꼭지점들은 분리 평면상에 있든가 또는 2번 물체의 꼭지점들과는 반대쪽에 위치하기 때문에 분리평면으로 Y=−10.5인 평면을 사용하는 것이 좋겠다.

만일 관측점(Xe, Ye, Xe)이 分離平面에 대하여 건물과 같은 쪽에 있다면, 집의 변들은 굴뚝의 어떤 부분에 의해서도 가리워지지 않는다. 그러므로, 집은 첫번째 우선 순위를 갖게 되고, 집에 의해 부분적으로 혹은 완전히 가리워질 수 있는 굴뚝은 두번째 우선 순위를 갖는다. 만일 관측점이 굴뚝쪽에 있다면, 굴뚝이 첫번째 우선 순위를 갖고, 건물이 두번째 우선 순위를 가질 것이다.

〔첫번째 우선 순위의 변들〕

첫번째 우선 순위를 갖는 물체의 보이는 변들은 이제 그릴 수 있게 된다. 이 변들의 각각은 눈에 보이는 면들을 이루는 경계선들이다.

이 변들은 첫번째 우선 순위를 갖는 물체의 변들이기 때문에 다른 물체에 의해 전혀 가리워지지 않게 된다.

〔두번째 우선 순위의 변들〕

두번째 물체에 관련된 변들은 보다 주의깊게 다루어져야 한다. 어떤 변들은 첫번째 우선 순위의 물체에 의해 부분적으로 또는 완전히 가리워질 수 있기 때문이다.

『그림 39』에서 건물과 굴뚝에 대하여 살펴 보면 1번과 2번 변은 굴뚝의 옆면들을 구성하는 것으로서 그 면들의 가시도 검사에서 통과되어 눈에 보이는 변들이다. 그렇지만 이것들은 건물에 의해 완전히 가리워진다.

따라서 이들은 그려져서는 안된다. 3, 4, 5, 6번 변들은 완전히 보이는 변들이므로 화면에 그려져야 한다. 반면에 7, 8, 9번 변들은 단지 일부분만이 보이는 변들로서, 이들에 대해서는 어느 부분에서부터 보이기 시작하는지를 구해야 한다. 그 다음에 보이는 부분만을 그릴 수 있다.

따라서, 두번째 우선 순위를 갖는 굴뚝의 보이는 면들은 다음과 같은 3가지 형태의 변들로 구성됨을 알수 있다.

① 어떤 변들은 가까운 곳의 건물에 의해 완전히 가리워지고

② 어떤 변들은 단지 일부만이 보이며

③ 어떤 변들은 완전히 보인다.

〔변의 끝점들의 可視상태〕

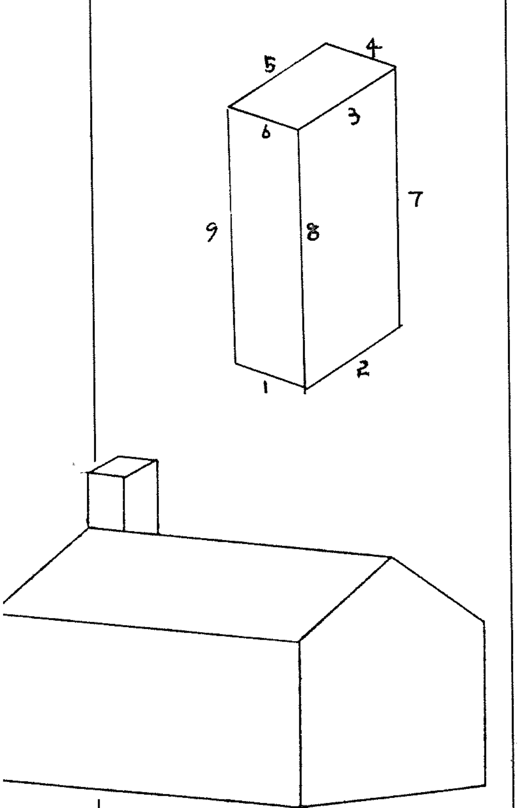
『그림−40(a)』에는 건물의 보이는 변들이 그려져 있고, 또 굴뚝의 한변의 양 끝점이 나타나 있다.

『그림−40(b)』에서 보면, 건물에 의해 가리워진 끝점 H는 건물의 외곽 경계선의 안쪽에 위치하고, 다른 한 끝점 V는 바깥쪽에 위치하고 있다.

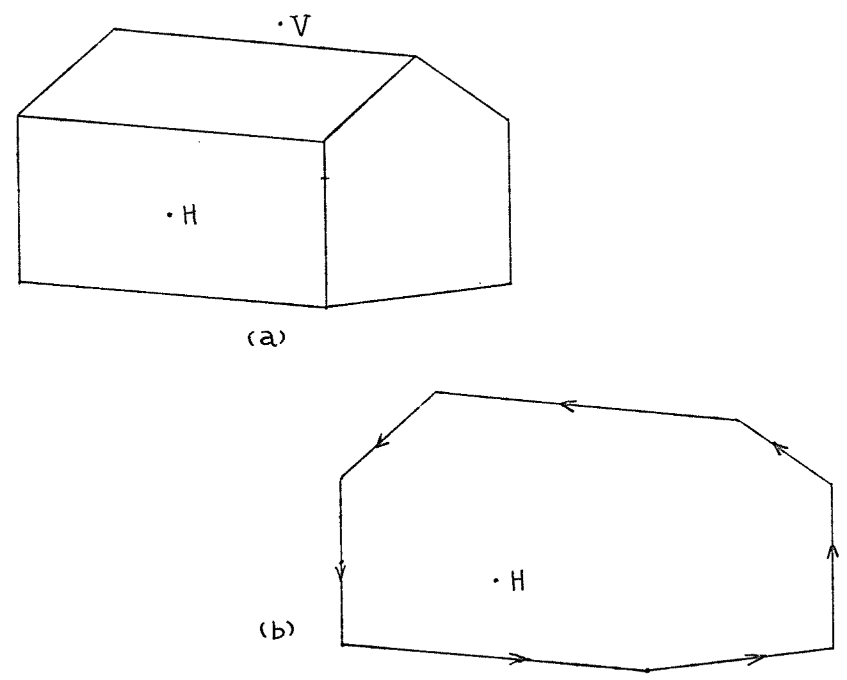
그런데 집의 외곽 경계선들은 반시계 방향으로 돌아가도록 정의되어 있으므로, 안쪽의 끝점 H는 각각의 경계선을 이루는 변들에 대하여 좌측 반평면에 놓이게 되며, 바깥쪽의 끝점 V는 적어도 한변에 대해서는 우측 반평면에 있게 된다.

구분 서브루틴(Subroutine)에서는 可視변수 VI를 보이는 끝점 (건물의 외곽 경계선의 바깥쪽에 있는 점)에 대해서는 +1을 지정하고 안보이는 끝점 (경계선의 안쪽 점)에 대해서는 −1을 지정하여 해결할 수 있다.

만일 건물의 외곽 경계선들 중 적어도 하나에 대하여 우측 반평면에 있게 되는 끝점은 보이게 되므로 이것을 알아보기 위하여 끝점(Xt, Yt)를



『그림−39』



『그림−40』 끝점의 가시상태검사

건물의 외곽 경계선들과 일일이 비교해야 한다.

〔부분적으로 안 보이는 변들의 Clipping〕

만일 어떤 변의 한 끝점은 보이고 다른 한 끝점은 안 보일 경우, 보이는 부분만을 디스플레이하려면 그 변을 Clipping해야 한다.

여기에서 보이는 점은 (X_v, Y_v)로 나타내지고 안보이는 점은 (X_h, Y_h)로 나타내진다. 2진 검색(Binary Search)을 이용하여 주어진 변을 두 부분으로 분리하는 점(X_t, Y_t)를 구하는 부분이다. 그러면 점(X_v, Y_v)에서 점(X_t, Y_t) 까지는 보이는 부분이 되고 점(X_t, Y_t)에서 점(X_h, Y_h)까지는 안보이는 부분이 된다.

2진 검색과정은 처음에는 매우 느리고 불규칙한 시도를 하는 듯이 보이나, 실제에 있어서는 매우 효율적인 것으로 그 과정은 다음과 같다.

1. 점(X_v, Y_v)에서 점$(X_h Y_h)$로 향하는 크기가 두 점간의 거리의 반에 해당하는 벡터(V_1, V_2)를 결정해야 한다.
2. 벡터(X_v, Y_v)에 벡터(V_1, V_2)를 더하여 두 점(X_v, Y_v)와 (X_h, Y_h)의 중점 (X_t, Y_t)를 구한다.
3. 점(X_t, Y_t)에 대해 가시도 검사를 한다.
4. $(V_1, V_2) = (V_1, V_2)/2$로 다시 (V_1, V_2)를 정의한다.
5. 만약 점(X_t, Y_t)가 보이면, 그 점을 점(X_h, Y_h) 쪽으로 이동시킨다.
$(X_t, Y_t) = (X_t, Y_t) + (V_1, V_2)$
만약 점(X_t, Y_t)가 보이지 않으면, 그 점을 점(X_v, Y_v) 쪽으로 이동시킨다.
6. 앞의 3과정으로 돌아간다.

점(X_t, Y_t)에 대한 가시도 검사를 하기 위해서는 서브루틴을 호출하여 사용하는 것이 좋다.

앞에서와 마찬가지로 이 루틴은 만약 점(X_t, Y_t)가 보이면 VI=+1을 되돌려 주고 그렇지 않으면 VI=−1을 되돌려 준다. 이 VI 값에 의하여 다음의 점(X_t, Y_t)를 결정하는데, 만약 VI=+1이면 점(X_h, Y_h) 쪽으로 이동시키고, VI=−1이면 점(X_v, Y_v) 쪽으로 이동시킨다.

위의 3과정에서 5과정까지의 루우프에서 빠져나오는 것은 점(X_t, Y_t)의 7번째의 값이 계산되었을 때 일어난다. 그러면 벡터(V_1, V_2)의 길이는 점(X_v, Y_v)와 점(X_h, Y_h) 간의 거리의 1/128이 된다. 좀 더 정교하게 할 경우 오차를 좁히면 되겠지만 Desk-Top 컴퓨터 디스플레이 화면에서의 대부분의 화상을 나타내는 데 별로 문제가 되지 않는다.

물론 루우프에서의 반복 횟수를 증가시킴으로써 보다 정확한 값을 얻을 수 있다.

루우프에서 빠져나올 때 점(X_v, Y_v)에서 점(X_t, Y_t) 까지의 선이 그려진다.

15. 컴퓨터에 의한 透視図 프로그램을 위한 기본개념

지금까지 컴퓨터에 의한 透視図의 実例를 들면서 프로그램 할 수 있는 시스팀분석(System Analysis)을 설명하였다. 실제 프로그램 작성을 위하여 Computer Graphics에 관한 기본개념을 참고로 설명하고자 한다.

15. 1 좌표축(Coordinate Axes)

해석 기하학에서는 일반적으로 좌표축시스팀(Coordinate Axis System)에 많이 의존하여 해석되므로 좌표기하학(Coordinate Geometry)이라고도 한다. 보통 중앙에서 서로 수직교차하는 두개의 직선을 긋는다. 이 경우 이 직선들은 방향성을 갖게 된다.

Desk-Top 컴퓨터에서 이러한 좌표축을 제공받아 좌표축의 원점을 중앙에 두고 X축은 오른쪽 방향을 증가 방향으로 하고, Y축은 위쪽 방향을 증가방향으로 하도록 구성해 줄 수 있다.

좌표시스팀을 구성하려면 좌표축 선형 이동 및 대칭이동(Reflection) 등의 조작이 필요하다.

예를 들면

```
PLOT 140, 0 TO 140, 191
PLOT   0, 96 TO 279, 96
```

라고 해 둔 뒤에 좌표축의 원점을 정의한다.

```
CX=140
CY= 96
```

그리고 다음에 사용될 PLOT X, Y 들은 모두 PLOT X+CX, CY−Y로 치환해 준다. 이때 PLOT명령이 받아들이는 X, Y 값의 범위는 다음과 같다.

$-140 \leq X \leq 139$
$-96 \leq Y \leq 95$

여기에서PLOT X,Y명령을 PLOT X+CX, CY−Y로 치환함으로써 X축에 대한 대칭이동 변환과 (C_X, Y_Y) 만큼의 선형 이동변환이 이루어짐을 알 수 있다. 이들 각각에 대한 변환 행렬은 다음과 같다.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ C_X & C_Y & 1 \end{bmatrix}$$

또 이들의 곱은 다음과 같다.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ C_X & C_Y & 1 \end{bmatrix}$$

그러므로 이들의 변환관계는 다음과 같다.

$$(X, Y, 1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ C_X & C_Y & 1 \end{bmatrix} = (X+C_X, C_Y-Y, 1)$$

지정된 X, Y값에 대하여 $S_x = X + C_X$와 $S_y = C_y - Y$에 의해 화면 좌표 값(S_x, S_y)를 얻거나 또는 크기조절자(S_C)를 사용한$S_x = S_C * X + CY$, $S_y = C_y - Y$로써 (S_x, S_y)를 구할 것이다.

15. 2 선(線)

기하학에서 점 이외의 가장 간단한 대상물은 선으로서, $Y = MX + B$로 표현될 수 있다. 여기에서 M은 선의 기울기이고 B는 이 선과 Y축이 교차하는 점의 Y값, 즉 Y의 절편이 된다. (X_1, Y_1)에서 (X_2, Y_2)까지 직선을 그으려면 X는 $DX = X_2 - X_1$만큼 Y는 $DY = Y_2 - Y_1$만큼 이동하여야 한다.

15. 3 원(圓, Circles)

원은 中心(H, K)로 부터 같은 距離(R)만큼 떨어져 있는 모든 점(X, Y)들의 집합이다. 이것을 대수적으로 표현하면 $(X-H)^2 + (Y-K)^2 = R^2$이 되는데 원점을 중심을 둔 원의 방정식은 다음과 같다.

$X^2 + Y^2 = R^2$

이러한 원의 방정식을 이용하여 원을 그리는 데에는 여러가지 방법이 있는데 여기서는 그 중에서 몇가지 방법만을 설명하기로 한다.

1) 직각 좌표법에 의한 원(圓)

방정식 $X^2+Y^2=R^2$ 을 Y에 대하여 정리하면 두개의 방정식

$YP=\sqrt{R^2-X^2}$ 과

$YN=-\sqrt{R^2-X^2}$ 이 구해진다.

이 방정식에 의하여 원을 그릴 수 있다.

2) 등각도 증가법에 의한 圓

원($X^2+Y^2=R^2$) 상의 한점(X, Y)는 각도 θ와 다음과 같은 관계를 갖는다.

$X=R \cdot \cos\theta$
$Y=R \cdot \sin\theta$

이렇게 등간격의 점들을 이용하여 원을 그릴 수 있는 것이다.

3) 순환식(Recursive Generation)을 이용한 반복법에 의한 圓

위의 방법에서는 매번 $\cos\theta$값과 $\sin\theta$값을 구하기 위해 많은 시간이 소요된다. 그러므로 여기에서 다루어질 방법에는 단지 한 각도에 대해서만 sin 값과 cos값을 구하면 된다.

먼저 원을 그리는데 사용되는 점들에 번호를 붙여보면 다음과 같다.

(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ……
(X_{22}, Y_{22})

두개의 연속된 점(X_n, Y_n)과 (X_{n+1}, Y_{n+1}) 사이에는 다음과 같은 관계가 성립한다.

$$X_n = R\cos\theta$$
$$Y_n = R\sin\theta$$
$$X_{n+1} = R\cos(\theta+d\theta)$$
$$Y_{n+1} = R\sin(\theta+d\theta)$$

여기서 $d\theta$는 점을 구하는 루우프에서의 증가분을 나타낸 것이다. 이것들을 삼각함수를 이용하여 다시 정리하면 다음과 같은 관계식을 얻을 수 있다.

$$X_{n+1}=R\cos(\theta+d\theta)=R\cos(\theta)\cos(d\theta)-R\cos(\theta)\sin(d\theta)$$
$$=X_n\cos(d\theta)-Y_n\sin(d\theta)$$
$$Y_{n+1}=R\sin(\theta+d\theta)$$
$$=R\sin(\theta)\cos(d\theta)+R\cos(\theta)\sin(d\theta)$$
$$=Y_n\cos(d\theta)+X_n\sin(d\theta)$$

그러므로 이 관계식으로부터 단지 일정한 값 $\cos(d\theta)$와 $\sin(d\theta)$를 한번만 계산하여 구해놓고 초기 값 X_n과 Y_n이 주어지면 다음에 그려질 점 (X_{n+1}, Y_{n+1})을 구할 수 있다. 이과정을 이용하여 圓을 그릴 수 있다.

원들의 중심이 원점($0, 0$)에 있는 것을 어떤 점(H, K)로 중심을 옮기려면 계산에 의해 구해지는 각 점들을 선형 이동시켜 주면 된다. 각 점(X, Y)는 다음의 행렬 곱에 의하여 ($X+H$, $Y+H$)로 이동된다.

$$(X, Y, 1)\begin{bmatrix}1 & 0 & 0\\ 1 & 0 & 0\\ H & K & 1\end{bmatrix}$$
$$=(H+X, Y+K, 1)$$

15.4 타원(Ellipse)

타원은 두개의 고정된 점으로부터의 거리의 합이 일정한 점들의 집합이다. 두개의 고정점들을 각각 그 타원의 초점(Focus)이라고 한다.

타원에서는 중점을 $C(H, K)$에 잡고, 길이 PC 또는 QC를 A로 나타내고 길이 TC 또는 SC를 B로 나타내는 것이 일반적이다. (『그림-41』 참조)

그러면 타원의 초점은 중점으로부터 PQ선상에서 좌우로 각각 $\sqrt{A^2-B^2}$ 만큼씩 떨어진 곳에 있게된다. 그리고 끈의 길이는 2A가 된다. 이러한 표기법에 의하여 타원을 대수적으로 표현하면 다음식과 같다.

$$\frac{(X-H)^2}{A^2}+\frac{(Y-K)^2}{B^2}=1$$

이 식은 가장 일반적인 형태로서 타원을 정확하게 표현하는 식이지만 좀 간단한 형태의 식을 생각해 볼 수 있다.

1) 원점 ($0, 0$)에 중심을 갖는 타원

타원의 중심을 원점에 위치시킴으로써 대수식과 프로그래밍을 간단하게 할 수 있는데 나중에 이것을 선형 이동시켜 원하는 위치에 중심을 갖는 타원을 얻을 수도 있다.

위의 식에서 H와 K값을 0으로 하면 원점에 중심을 갖는 타원의 식은 다음과 같이 구해진다.

$$\frac{X^2}{A^2}+\frac{Y^2}{B^2}=1$$

이 식을 이용하여 타원을 그리려면 사용될 점들을 상당히 많이 계산하지 않으면 타원은 모가 나게 그려질 것이다. 다음 식은 위의 식과 동일하다.

$$X=A\cos(\theta)$$
$$Y=B\sin(\theta) \qquad 0\leq\theta<2\pi$$

2) 線型移動과 回転

선형이동과 회전 변환율을 이용하여 보다 자유스러운 타원을 얻을 수 있다.

회전 변환 행렬은

$$\begin{bmatrix}\cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \sin(\theta) & 0\\ 0 & 0 & 1\end{bmatrix}$$

이고 (H, K) 만큼씩의 선형 이동 변환행렬은

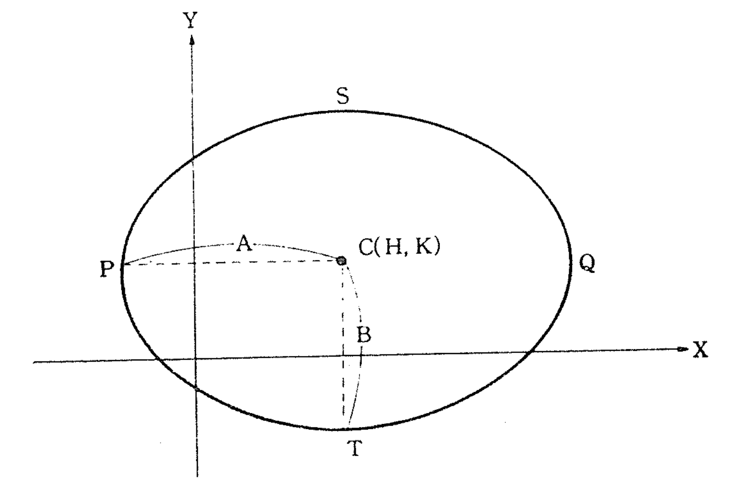
$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ H & K & 1\end{bmatrix}$$

이며 임의의 어떤 화상을 θ만큼 반시계 방향으로 회전시킨 후 다시 (H, K) 만큼 선형 이동시키기 위한 변환행렬은 다음과 같다.

$$\begin{bmatrix}\cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ H & K & 1\end{bmatrix}$$

$$=\begin{bmatrix}\cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \cos(\theta) & 0\\ H & K & 1\end{bmatrix}$$

임의의 어떤 점(X, Y)는 다음과 같



『그림-41』 타원

이 변환된다.

$$(X, Y, 1)\begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ H & K & 1 \end{bmatrix}$$
$$= (Xcos(\theta) - Ysin(\theta) + H,\ Xsin(\theta) + Ycos(\theta) + K, 1)$$

이것을 반영시켜 타원을 그릴 수 있다.

3) 원으로부터 변형되어진 타원

타원은 원이 변형된 것으로 생각할 수 있는데, 원을 X방향이나 Y방향으로 확대 또는 축소시킴으로써 타원을 만들 수 있는 것이다. 예를 들면 반지름이 1 인 원($X^2+Y^2=1$)의 X좌표 값은 A배로 크기 조절해 주고, Y좌표 값은 B배로 해 줌으로써 다음과 같은 타원을 얻을 수 있다.

$$\frac{X^2}{A^2} + \frac{Y^2}{B^2} = 1$$

이러한 크기조절 행렬은 다음과 같다.

$$\begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

이것을 이용하여 타원을 그릴 수 있다.

15. 5 媒介函数(Parametric Equations)

다음과 같은 함수를 매개함수라고 한다.

$X = T^2 + 10$
$Y = T^2 - 10T$

이들은 X, Y값을 계산하기 위해 매개변수 T를 사용하고 있다. 한 예로서 원을 그리기 위해 $R=50$으로 하여 다음과 같은 매개함수를 사용 할 수 있다.

$X = Rcos(T)$
$Y = Rsin(T)$

또 이러한 것은 타원을 그리는 프로그램에서도 사용되었다.

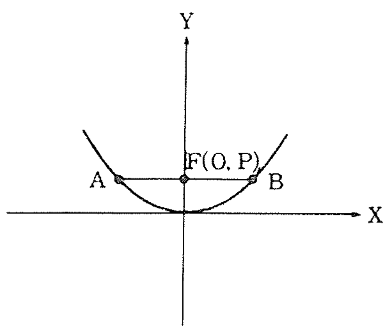
이러한 X, Y의 매개함수 관계를 사용함으로써 보다 유연한 곡선을 짧은 연산시간내에 구할 수 있게된다.

또 이들을 사용하면 색다른 곡선도 구할 수 있다.

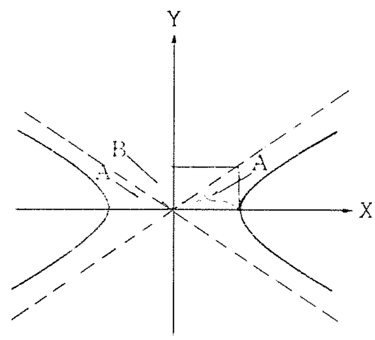
다음과 같은 것이 그 일례가 될 수 있을 것이다.

$X = 50\,sin(2(T - \frac{\pi}{13}))$
$Y = -70\,cos(T)$

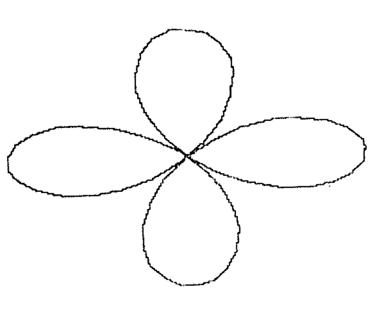
$A=50$, $B=2$, $C=13$, $D=-70$로 두면



『그림－42』 포물선



『그림－43』 쌍곡선



『그림－44』 장미모양의 극좌표 곡선

$X = A\,sin(B\,(T - \frac{\pi}{C}))$
$Y = D\,cos(T)$

15. 6 포물선

포물선은 많은 면에서 실제적으로 응용되는 곡선이다. 예를 들어 만약 돌이나 공을 위쪽으로 던지면, 그것이 지나간 자취는 포물선이 된다. 바람이나 공기의 저항이 없다고 가정하면 이들은 포물선 운동을 한다.

포물반사경(Parabolic Reflectors)은 반사된 빛을 한 점, 즉 초점에 집중시켜 주므로 매우 유용하다. 또한 혜성들은 포물섭의 초점에 해당하는 태양 근처를 포물선을 그리며 지나간다.

방정식 $Y = X^2/4P$는 『그림－42』와 같은 포물선을 나타낸다. 원점은 포물선의 꼭지점이 된다. 그리고 초점은 점$(0, P)$에 있다. 포물선에서의 초점과 꼭지점을 지나는 곡선을 그 포물선의 축이라고 한다. P값에 의해 초점과 꼭지점간의 상대적인 위치가 결정되고 포물선의 모양이 결정된다.

초점을 지나면서 포물선의 축과 수직인 선분 AB의 길이는 $4/P/$이다. $4/P/$의 값에 의해 포물선의 넓이(Width)가 결정된다.

방정식 $Y = X^2/4P$에 의해 포물선이 정의되지만, 이것보다 더 유용한 표현식으로서 다음과 같은 매개할 수가 있다.

$X = 2PT$
$Y = PT^2$

이러한 것을 이용하여 포물선을 그릴 수 있다.

15. 7 쌍곡선(雙曲線, Hyperbolas)

원, 타원, 포물선 등을 원뿔의 단면 곡선이라고 한다. 이 곡선들은 원뿔과 평면과의 교차 곡선들이기 때문이다.

쌍곡선도 원뿔의 단면 곡선들 중의 하나이다. 쌍곡선은 대수적으로 다음과 같이 표현된다.

$$\frac{X^2}{A^2} - \frac{Y^2}{B^2} = 1$$

쌍곡선의 모양은 A, B의 값을 바꿈으로써 여러가지로 바뀐다. 쌍곡선의 매개 함수적 표현은 다음과 같으며 이것을 사용하면 보다 쉽게 프로그램을 작성할 수 있게 된다.

$X = A\,sec(T)$
$Y = B\,tan(T)$

15. 8 極座標 曲線(Polar Coordinate Curves)

반지름 R이 50인 원을 다음과 같이 매개함수로써 표현할 수 있다.

$X = Rcos(T)$
$Y = Rsin(T)$

만일 여기에서 R의 값을 일정하게 놓지 않고 여러가지로 변할 수 있게 한다면, 즉 R을 변수로 사용하면 재미있는 곡선들을 얻을 수 있다.

R을 T에 대한 함수로 정의하여 프로그램을 작성할 수 있다.

이를 이용하여 장미모양의 도형 등을 그릴 수 있다.