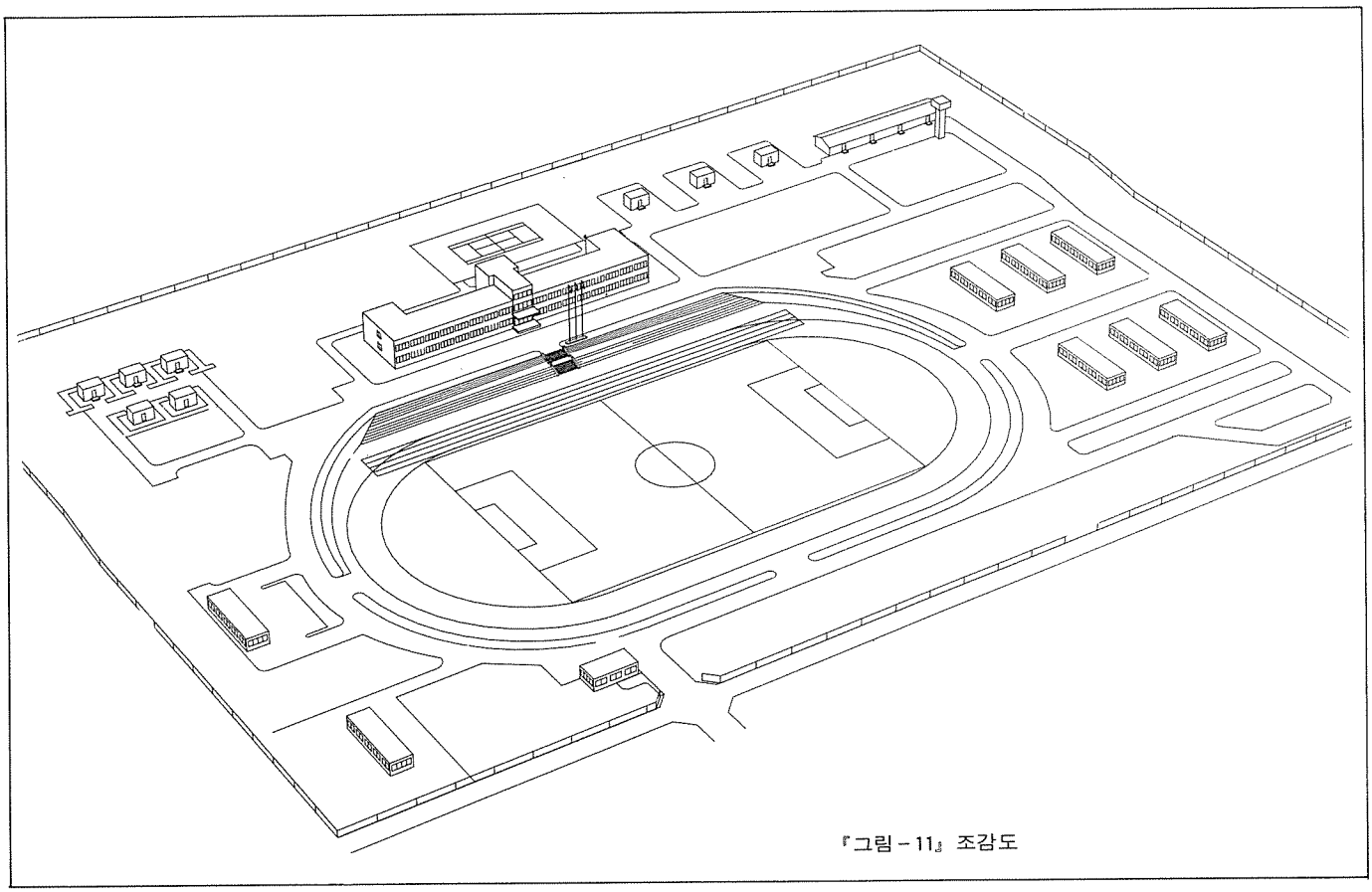
# 컴퓨터에 의한 透視図 作図

曺 鐵 鎬 – 건축사 · 建国大교수

# PERSPECTIVE BY CAD SYSTEM

**CHUL-HO CHO** / KONKUK UNIV.



『그림-11』 조감도

### 8. 조감도의 実例

建物의 투시도는 보는 사람의 눈높이를 建物보다 낮게 잡을 수도 있고 높게 잡을 수도 있게 된다. 建物의 群은 눈높이를 낮게 잡아 표현할 수는 없고 높게 잡아야 한다. 이 경우 조감도라고 하는데 컴퓨터로 표현할 때는 투시도와 다른 技法이 아니라 같은 技法으로 표현한다. 『그림-11』이 조감도로 표현한 한 예이다.

투시도에서와 마찬가지로 눈높이며 방향 거리를 자유로 조정이 가능하다. 프롯터에 그리기 전에 CRT 화면상에서 좋다고 생각하는 위치를 결정해서 컴퓨터에 연결된 프롯터로 그려내면 될 것이다.

투시도나 조감도나 눈의 위치만 다를 뿐 근본원리는 앞에서 소개한 바와 같다. 다만 컴퓨터에서는 숨어 있는 線(隱線·Hidden Line)을 해결하는 것이 가장 어려운 점이며, 또 번거로운 것이 된다. 이러한 것은 컴퓨터 自体内에서 해결할 수 있으므로 利用者는 신경 쓰지 않아도 된다.

지금까지의 내용은 筆者 자신이 開発한 프로그램에 대한 것이나, 外国에서 이미 開発되어 활용하고 있는 Package를 소개함으로써 참고가 되도록 하고자 한다. 여기에 소개하는 것은 英国 ARC社에서 開発한 BDS (Building Design System) 중의 투시도(Perspective) 부분이다.

### 9. 外国Package의 利用実例

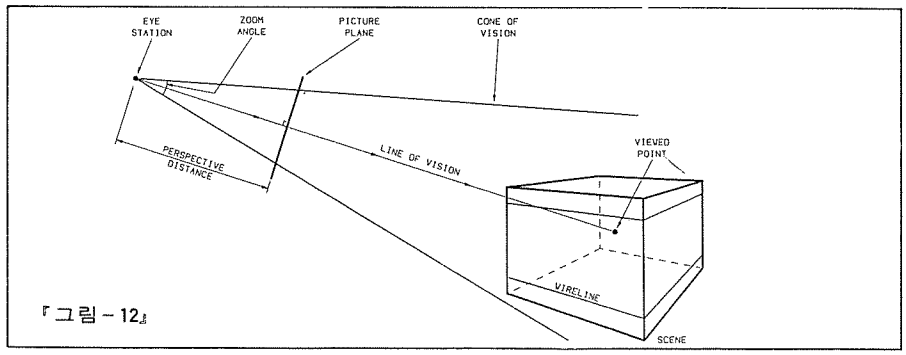
ARC社의 BDS에서는 XPER라는 명령을 주어 3次元으로 기억 하고 있는 建物 형상을 투시도나 조감도로 표현할 준비를 하게 한다.

『그림-12』와 같이 대상물과 눈의 위치를 정의하고 있다.

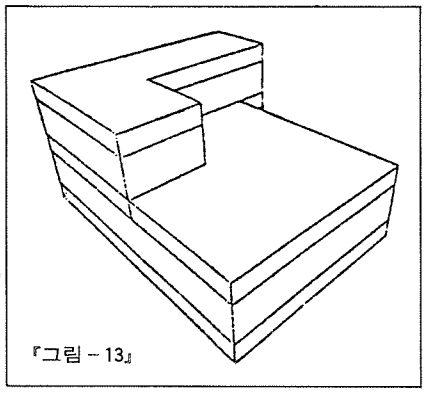
YMODE BUILDING이라는 명령에 의하여 『그림-13』과 같은 건물이 CRT의 일종인 GRAPHICS DISPLAY에 나타날 수 있다. 여기서 YMOVE Z 15라는 명령어로 눈높이를 15m 높여 볼 수 있는 예가 『그림-14』이다.

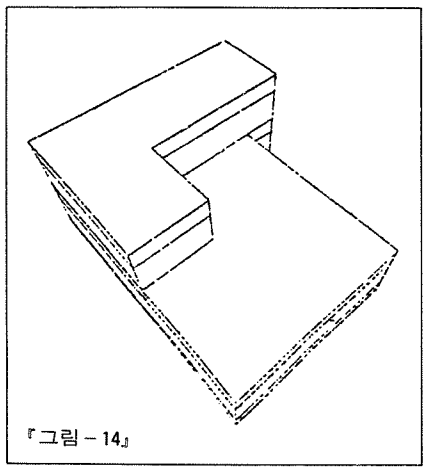
건물을 돌려서 볼 수 있도록 YROUND 50을 하면 50°(度)로 회전하게 된다. 50度씩 4번 회전시킨 것이 『그림-15』이다.

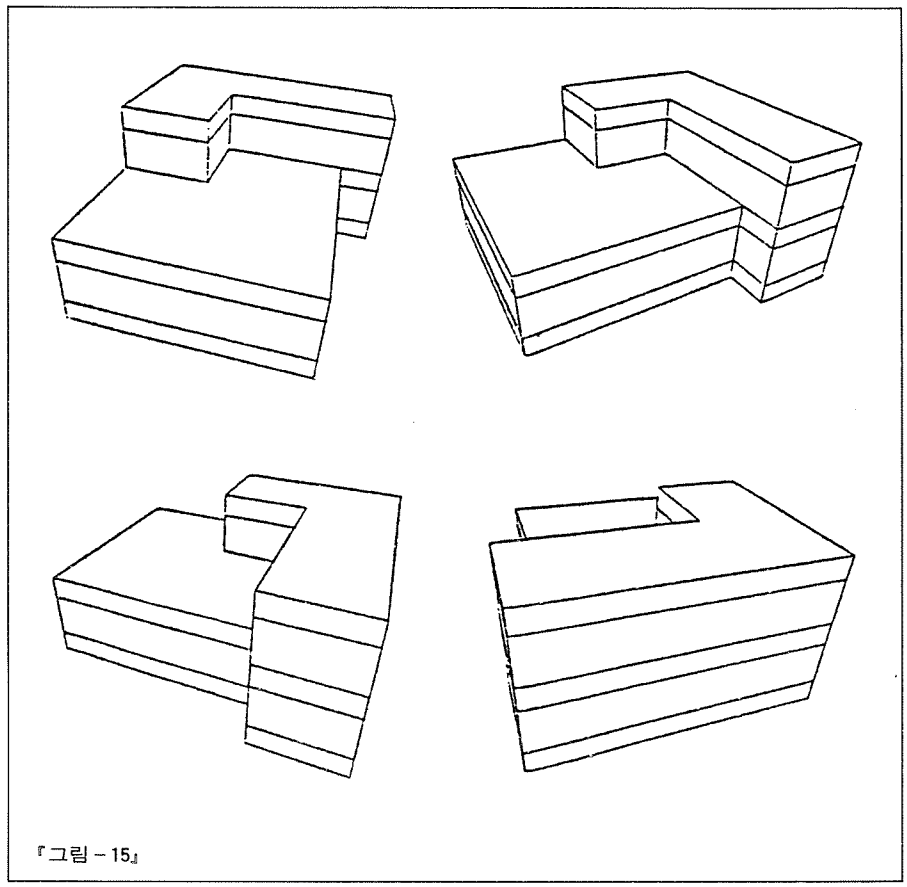
건물바닥에 線을 나타낼 경우 YCARPET로 線의 간격이며 범위를 줄 수 있다.



『그림－12』


『그림－13』


『그림－14』


『그림－15』

그 예가 『그림－16』이다.

숨어 있는 線을 보기 위해서는 YHIDENO로 하면 『그림－17』처럼 된다.

YTHREE OFF를 하면 3点 투시에서 2点 투시로 바꾸어 지는데 그 예가 『그림－18』이 된다.

이미 작성된 건물 투시도를 원점에서 이동 회전을 위하여 YTRANS를 쓸 수 있다. 그 예가 『그림－19』이다.

建物에서는 X, Y, Z 方向의 크기를 변동시킬 필요가 거의 없지만 크기를 변동하는 SCALE의 기능도 있다.

室內 회전계단의 室內 투시도를 작도하기 위해서 그 과정을 살펴 보면 다음과 같다. 회전계단의 平面·立面을 결정하여 입력한 내용이 『그림－20』이다.

벽을 90° 회전해서 본 투시가 『그림－21』이다. 벽과 계단 중심부를 나타낸 것은 『그림－22』가 된다. 계단판을 나타낸 것이 『그림－23』이고, 핸드레일을 나타내면 『그림－24』와 같게 된다. 바닥을 나타낸 것이 『그림－25』이고, 위에서 내려다 본 경우가 『그림－26』이 된다.

실이용자는 3次元 건물이 2次元의 투시도로 변하는 내용을 몰라도 이용이 가능하지만 컴퓨터를 활용하기 위해서는 그 내용을 알아 두는 것이 좋다.

### 11. 3次元 建物에 대한 2次元 画像

『그림－8』에서 눈(점 P) 위치는 직각 좌표로는 H, K, L로 구 좌표로는 $\rho, \theta, \phi$로 정해진다. 투영된 평면이 관측자의 눈으로부터 D만큼의 거리로 떨어져 있다고 가정하면 투영평면, 즉 화면은 관측자의 눈과 원점을 연결하는 선과 수직을 이루도록 하면 점 P에 새로운 좌표시스팀을 하나 구성할 수 있게 되는데, 이것을 눈좌표시스팀(Eye Coordinate System)이라 한다. 이 점들은 눈좌표시스팀으로(Xe, Ye, Ze)라고 표현될 수도 있고, 표준좌표시스팀으로(X, Y, Z)라고 표현될 수도 있는 것이다.

눈좌표시스팀에서 관측자의 보는 방향이 표준좌표시스팀의 원점을 향하고 있을 때, Ze軸은 보는 사람의 시선 방향을 향하고, Xe軸은 오른쪽을 향하며, Ye軸은 위쪽을 향한다. 눈좌표시스팀은 왼손 좌표시스팀이 된다.

투시도로 작도해 낼 수 있도록 하

기 위해서 다음과 같은 두가지 과정을 거쳐야 한다.

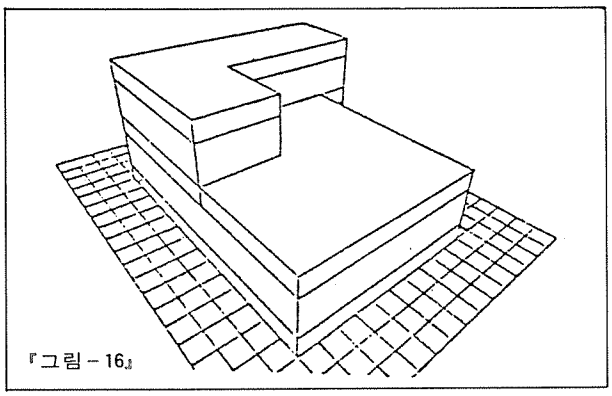
1. 표준좌표시스팀에서의 점($x, y, z$)을 눈좌표시스팀에서의 좌표(Xe, Ye, Ze)로 변환시켜야 한다.

2. 눈좌표시스팀에서의 점(Xe, Ye, Ze)로부터 화면 좌표(Sx, Sy)를 구해야 한다. 여기에서 2번째 과정은 앞 절에서 설명한 바 있는 식으로 부터 $Sx = D(Xe / Ye)$, $Sy = D(Ye / Ze)$로 쉽게 구할 수 있다.

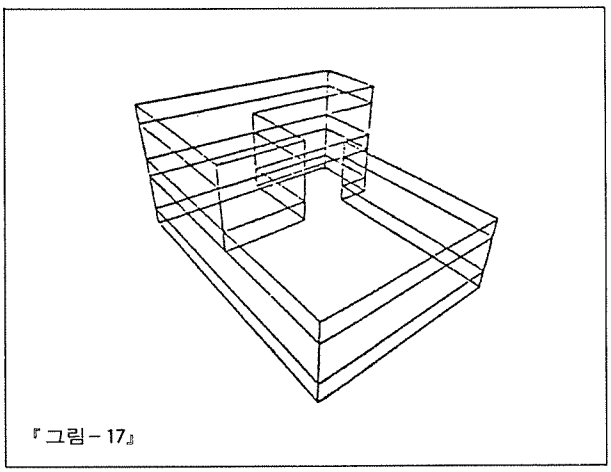
표준좌표시스팀과 눈좌표시스팀 사이의 관계를 구함으로써 첫번째 과정을 해결할 수 있다. X, Y, Z좌표 시스팀에서 Xe, Ye, Ze좌표시스팀으로 변환하려면 4단계의 변환과정을 거쳐야 한다.

중간단계에서의 좌표시스팀들은 모두 $x', y', z'$로 나타낼 수 있게 된다. 그리고 최종단계에서의 좌표시스팀 $x', y', z'$은 눈좌표시스팀 Xe, Ye, Ze가 된다.

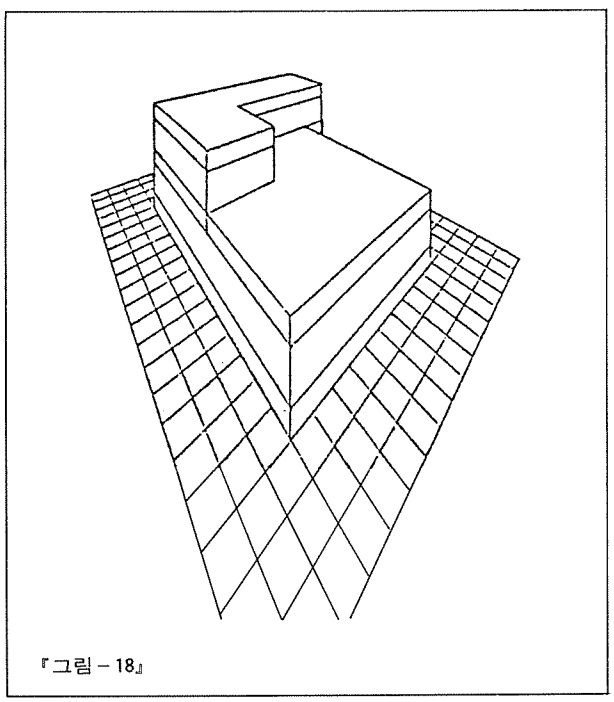
새로운 좌표시스팀을 얻기 위한 각 변환들에서 관측자의 눈에 해당하는 점 P는 표준좌표시스팀과 관련하여 직각 좌표(H, K, L) 또는 구좌표($\rho, \theta, \phi$)로 표현할 수 있다. 이 경



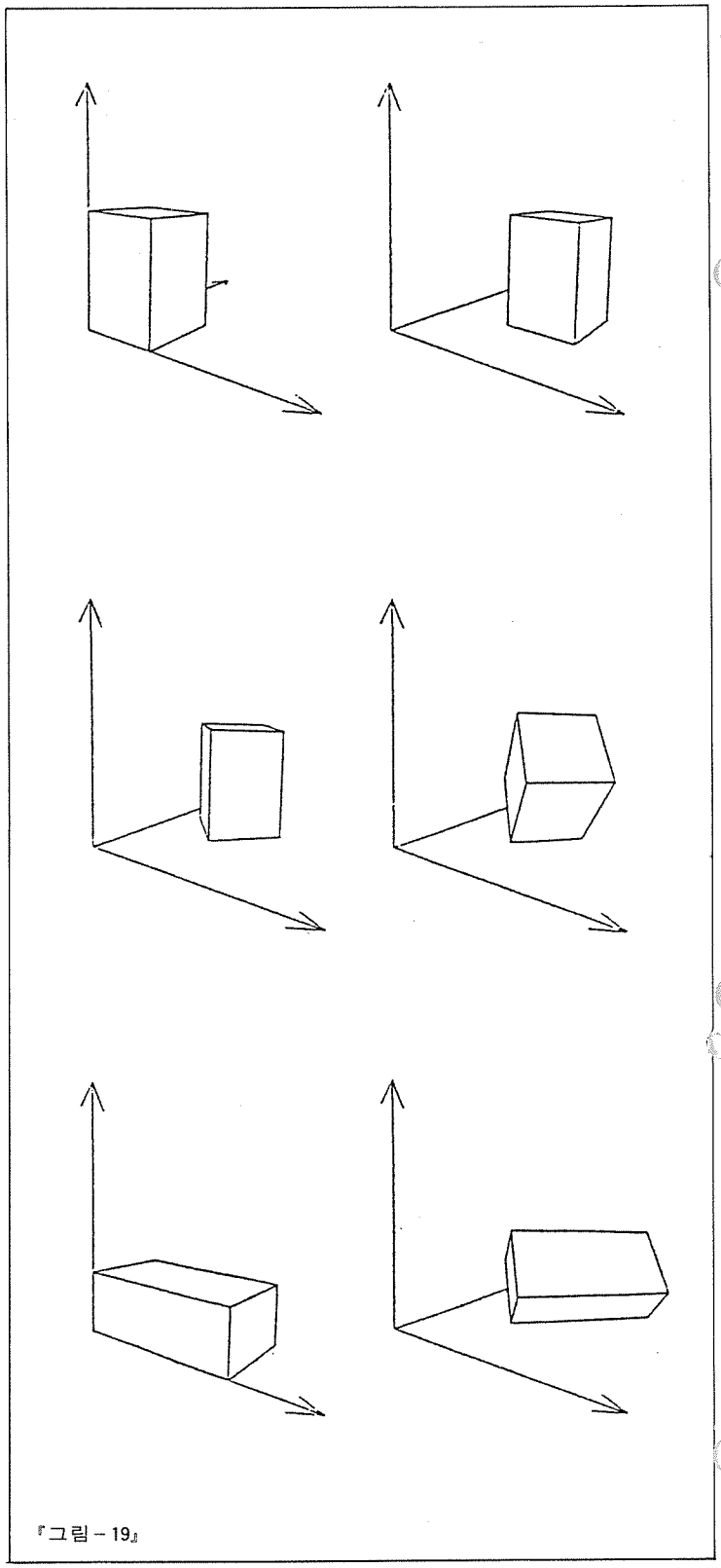
『그림-16』



『그림-17』



『그림-18』



『그림-19』

우 이들 사이의 관계는 다음과 같다.

$H=\rho \sin\phi\cos\theta$

$K=\rho \sin\phi\sin\theta$

$L=\rho \cos\phi$

이들 좌표변환 4단계를 간단히 설명하면 다음과 같게 된다.

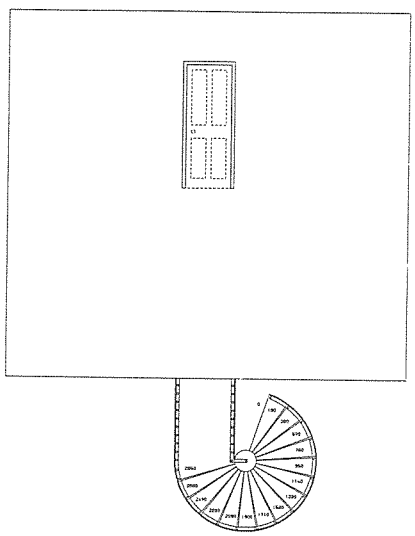
1. 제 1 단계에서는 보는 사람의 눈 위치에 원래의 좌표시스팀과 동일한 방향의 軸을 갖는 새로운 좌표시스팀을 구성한다. 이러한 과정을 변환行列 A로 나타낼 수 있다.

2. 제 2 단계와 제 3 단계에서는 제 1 단계에서 좌표축들을 회전 시켜서 Z′축이 표준좌표시스팀의 원점을 향하고, Y′축이 앙의 Z축과 교차하도록 해서 보는 사람의 위치에서 볼때에 Y′축은 위쪽을 향하고 X′축은 왼쪽을 향하게 된다.

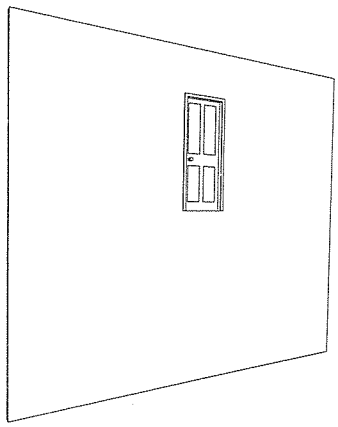
3. 제 4 단계에서는 X′축의 방향을 오른쪽으로 향하도록 하여 왼손 좌표시스팀으로 변환한다. 이러한 과정을 거침으로써 보는 사람의 위치에서 보다 편한 좌표시스팀이 구성된다.

각 단계에서의 변환 행렬을 구해보면 다음과 같다.

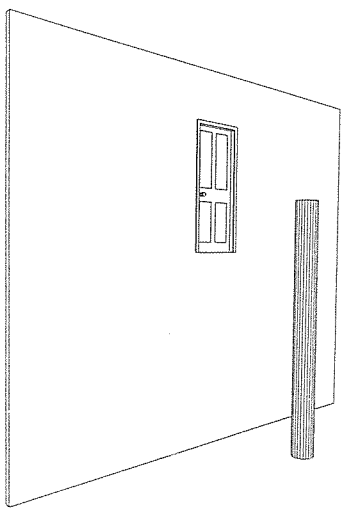
제 1 단계



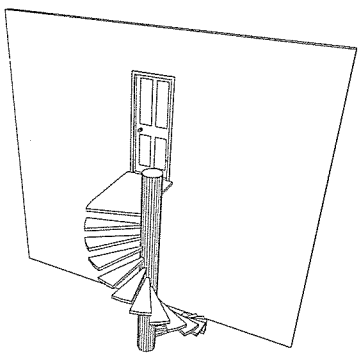
『그림 - 20』



『그림 - 21』



『그림 - 22』



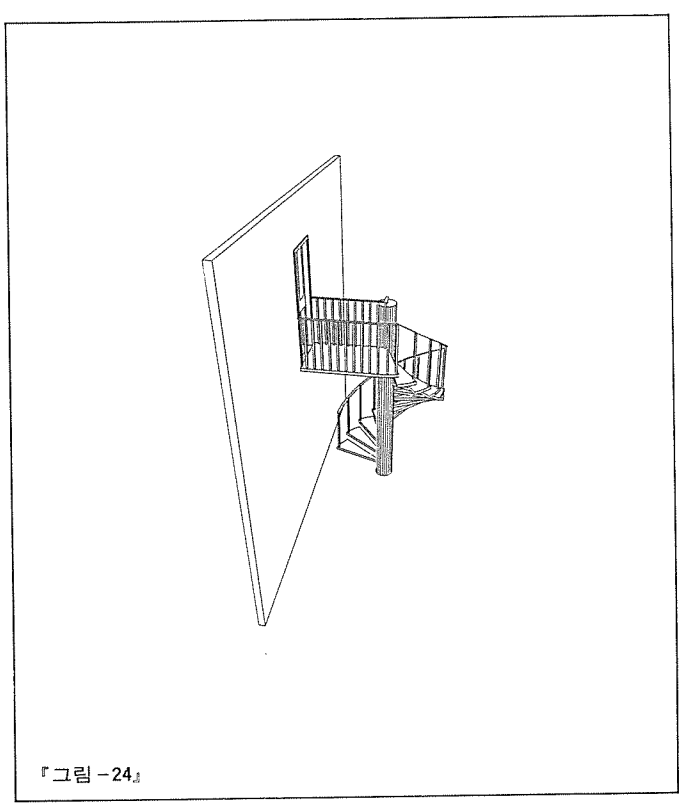
『그림 - 23』

원점을 (H, K, L) = ($\rho sin\phi\ cos\theta$, $\rho sin\phi\ sin\theta$, $\rho cos\phi$)에 위치시킨다. 변환 行列은 다음과 같게 된다.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -H & -K & -L & 1 \end{bmatrix}$$

또는

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



『그림 - 24』

$$\begin{bmatrix} -\rho sin\phi\ cos\theta & -\rho cos\phi \\ -\rho sin\phi\ sin\theta & 1 \end{bmatrix}$$

여기에서 H, K, L에 대한 구좌표표현이 보다 불편하게 보이지만 계산을 간단하게 할 수 있게 해줌으로써 좋게 된다.

제 2 단계

좌표시스팀을 Z′축에 대하여 시계방향으로 (90 − $\theta$) 만큼 회전시킨다.

회전결과로, 음의 Y′軸은 Z 軸과 교차하게 된다.

이 경우의 변환행렬은 다음과 같다.

$$B = \begin{bmatrix} sin\theta & cos\theta & 0 & 0 \\ -cos\theta & sin\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

여기에서 좌표시스팀을 회전시킬때의 변환행렬은 한 좌표시스팀 내에서 점을 회전시킬 때의 변환행렬에 대한 역행렬이 된다.

제 3 단계

좌표시스팀을 X′軸에 대하여 반시계 방향으로 (180 − $\phi$) 만큼 회전 시킨다.

Z′軸은 X, Y, Z 좌표시스팀의 원점을 향하게 될 것이다. 이 경우의 변환 行列은 다음과 같다.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -cos\phi & -sin\phi & 0 \\ 0 & sin\phi & -cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

제 4 단계

다음과 같은 변환행렬로서 X′軸의 방향을 바꾸어 왼손 좌표시스팀을 구성할 수 있다.

$$D = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

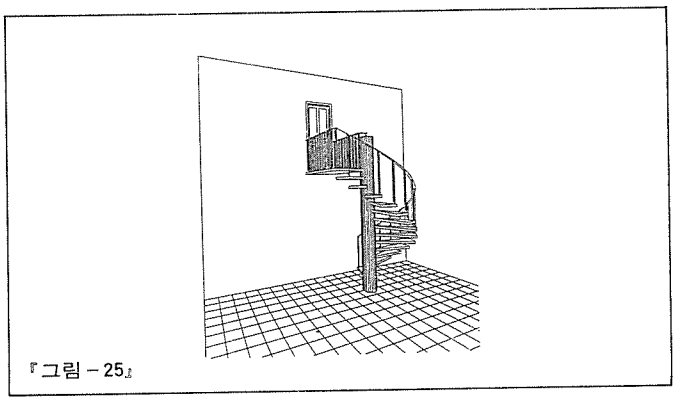
만약 어떤 점에 대한 표준 좌표(X, Y, Z)를 알고 있다면 눈 좌표(Xe, Ye, Ze)는 (Xe, Ye, Ze, 1) = (X, Y, Z, 1) ABCD로 구해진다. 계산을 간단하게 하기 위해서 하나의 변환 행렬 T= ABCD를 구해 사용할 수 있다.

(Xe, Ye, Ze, 1) = (X, Y, Z, 1)

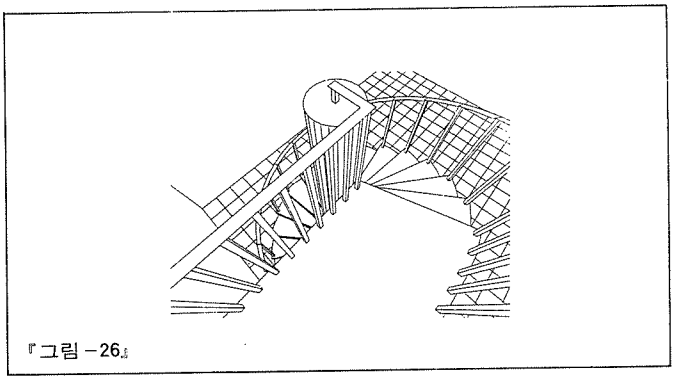
$$\begin{bmatrix} -sin\theta & -cos\theta\ cos\phi & -cos\theta\ sin\phi & 0 \\ cos\theta & -sin\theta\ cos\phi & sin\theta\ sin\phi & 0 \\ 0 & sin\phi & -cos\phi & 0 \\ 0 & 0 & \rho & 1 \end{bmatrix}$$

## 11. 隱線 및 隱面을 위한 面의 方向 決定

隱線(Hidden Line)이나 隱面(Hidden Surface)을 제거하는 것은 컴퓨터 그래픽에서의 커다란 과제중의 하나가 된다. 隱線제거는 많은 연산시간과 기억용량을 필요로 한다. 大型 컴퓨터에서는 이러한 것을 만족시켜 주지만 Desk-top 컴퓨터와 같은 개인용 컴퓨터에서는 이러한 것들을 충분히 만족시켜 줄 수 있는 여유가 없게 된다. 그러므로 여기에서는 여러 기법들을 소개하고 이에 따른 간단한 예제 프로그램을 사용해서 간단한 3次元 물체에 대한 투시도를 그려보도록 하겠다. 범용으로 쓰이는 隱線 제거방법은 매우 비효율적이다. 컴퓨터에서 각각의 특정한 응용에 대하여 그것에 맞는 技法을 사용하여 간단한 프로그램을 작성하는 것이 보다 효율적이 되겠다.



『그림 - 25』



『그림 - 26』

建物의 隱線이나 隱面을 제거하기 위해서는 먼저 보이는 隱과 隱線을 구분하는 방법을 알아야 할 것이다.

建物의 보이는 부분만을 디스플레이하는 가장 확실한 方法은 컴퓨터에 보이는 부분만의 데이터를 入力시켜 주어 그것만을 디스플레이 시키는 方法이겠다.

이 技法에서는 프로그래머가 먼저 특정한 관측점을 구해서 그곳으로 부터 보이는 물체의 꼭지점들을 미리 구해야 하는 불편함은 있지만 이러한

프로그램은 매우 간단하며 매우 빠른 시간 내에서 투시도를 그려준다. 그러나 여기에서는 또다른 제약점들이 있다. 만일 다른 관측점에서 본 투시도를 그리려면 프로그램에서의 DATA 문과 線을 그리는 부분을 수정할 필요가 있다.

### 11.1 面의 方向決定 동작원리

주사위를 보면 『그림-27』과 같이 두面이나 세面이 보이게 된다. 다른 面들은 우리들로부터 멀어지는 쪽을 향하게 되므로 우리에게는 보이지 않는다.

우리가 隱線 제거방법을 사용하려면 먼저 面의 方向을 결정해야 한다.

정육면체의 각 면에 대해서는 두개의 수직방향을 설정할 수가 있다.

한 방향은 정육면체의 안쪽을 향하는 것이고, 다른 한 방향은 정육면체의 바깥쪽을 향하는 것이다. 『그림-28』에서 이 方向들을 수직 벡터로 표현하고, 또한 면의 방향 벡터로서는 바깥쪽을 향하는 벡터를 사용하기로 한다.

물체의 각 면에 대해 두번째 벡터로서 視線벡터를 정의할 수가 있다. 이 벡터는 면의 한 꼭지점에서 보는 사람의 눈의 위치인 관측점으로 향하는 벡터이다.

각 면에 대해서 우리는 視線벡터와 方向벡터가 이루는 角을 측정할 수있다. 만일 이 角이 0°에서 90°사이의 角이면, 그 面은 관측자쪽을 향하게 되어 눈에 보이게 되므로 디스플레이 될 것이다. 그렇지만 측정각이 90°에서 180° 사이의 각이면, 그 면은 관측자로부터 멀어지는 쪽을 향하게 되므로 우리에게는 보이지 않게 된다. 따라서 이 面은 디스플레이되지 않는다. 『그림-29』에 이러한 것을 나타내었다.

### 11.2 方向벡터(Orientation Vector)

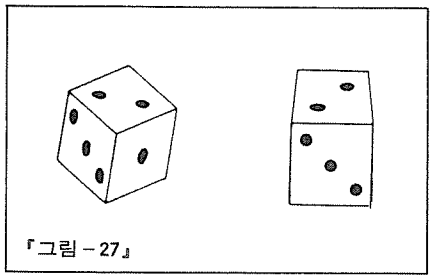
『그림-30』과 같이 면의 꼭지점들에 각각 번호를 붙이면, 이 경우 번호 1에 해당하는 꼭지점은 임의로 선택할 수 있지만 나머지 꼭지점들은 정육면체를 밖에서 볼 때 반시계 방향으로 돌아가며 번호를 붙여야 한다. 다음에 1번 꼭지점에서 2번 꼭지점으로 향하는 벡터 $\vec{u}$와 1번 꼭지점에서 3번 꼭지점으로 향하는 벡터 $\vec{v}$를 구해야 한다. 그러면 이 두 벡터의 外積(Cross Product)에 의한 벡터($\vec{n}=\vec{u}\times\vec{v}$)는 면에 수직하게 되며, 정육면체의 바깥쪽을 향하게 된다.

이 때 방향 벡터 $\vec{n}$의 크기는 우리에게 별 의미가 없게 된다.

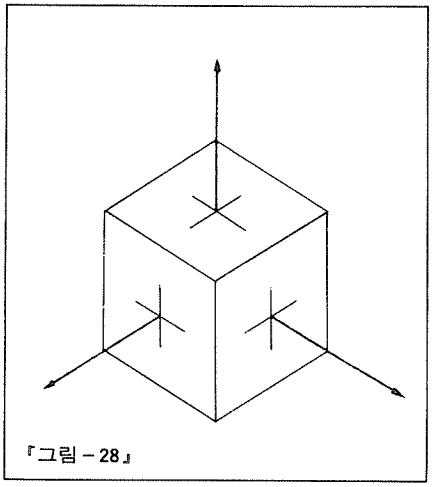
### 11.3 可視度(Visibility)

벡터 $\vec{w}$와 $\vec{n}$의 內積은 다음과 같이 정의할 수 있다.

$$\vec{w}\cdot\vec{n}=|\vec{w}||\vec{n}|\cos\theta$$



『그림-27』



『그림-28』

여기에서 $\theta$는 벡터가 이루는 角인데, 만일 $\theta$가 0°~90°사이의 角이면 內積 $\vec{w}\cdot\vec{n}$은 正의 값을 가질 것이고 90°~180° 사이의 각이면 $\vec{w}\cdot\vec{n}$은 負(−)의 값을 갖게 된다.

우리는 $\vec{w}$를 어떤 면의 視線벡터로 하고 $\vec{n}$을 方向벡터로 함으로써, 그 面의 可視度를 조사해 볼 수 있다.

『그림-29』의 (b)에서와 같이 보이는 面은 $\vec{w}$와 $\vec{n}$이 이루는 角이 0°~90°가 되어 $\vec{w}\cdot\vec{n}$은 正의 값을 갖는다.

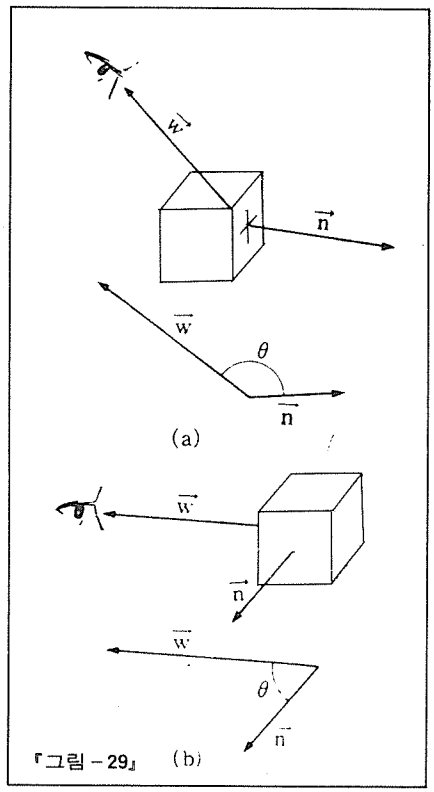
『그림-29』의 (a)에서와 같이 안 보이는 面, 즉 隱面은 $\vec{w}$와 $\vec{n}$이 이루는 角이 90°~180°가 되어 $\vec{w}\cdot\vec{n}$은 負의 값을 갖게 된다.

## 12. Blackout에 의한 두 建物 사이의 隱線除去

여기서는 여러 物体 사이에서 서로 가리우는 부분은 제거하고 보이는 부분만을 그리는 方法으로 해결할 수있다. 따라서 한 물체에서의 보이는 면을 결정하는 과정 이외에는 가까운 곳의 물체에 의해 가리워지는 부분도 결정해야만 한다.

어떤 물체의 가리워진 부분도 일단 그렸다가, 다음에 가까운 물체, 즉 가리우는 물체를 그리면서 가리워진 부분은 지워지도록 하는 方法이다.

『그림-31』은 이 方法에 의하여 그려진 것이다.

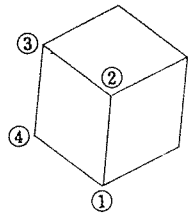


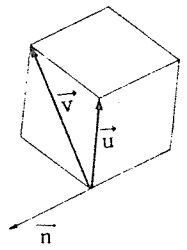

『그림-29』

## 13. 2進 検索(Binary Search)에 의한 두 물체 사이의 隱線除去

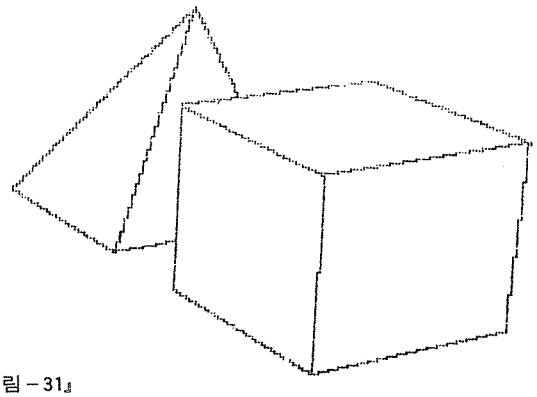
『그림-32』에서 피라미드와 같은 꼭지점 R는 정육면체에 의해 가리워진다. 피라미드의 辺 PR는 부분적으로 가리워져 단지 PS부분만이 보이게된다. 꼭지점 P의 화면 좌표값은 그대로 쓸 수 있지만, 점 S의 좌표값은 계산하여야 한다. 점 S는 정육면체의 변 QT와 피라미드의 변 PR의 교차점이므로, 점 S의 좌표값은 선 PR과 QT의 방정식으로부터 구할 수 있다. 이에 관련된 대수식은 간단하지만 효율적인 것은 아니다. 그러므로 여기에서는 S의 좌표를 구하기 위해 2진검색(Binary Search)을 이용하는 방법을 쓰고 있다.

점 S에 대한 가시도 검사를 하기 위해 먼저 점 T에서 점 Q로 향하는 벡터 $\vec{u}$와 점 T에서 점 P로 향하는

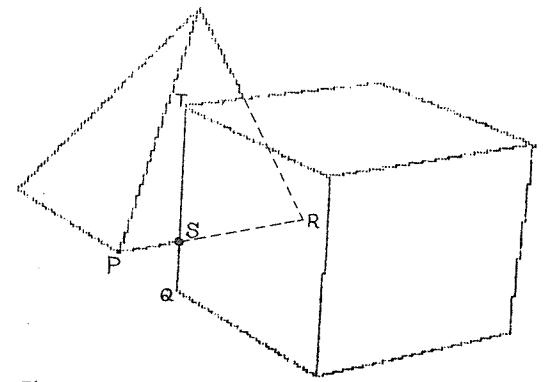



『그림 - 30』



『그림 - 31』




『그림 - 32』

벡터 (XP - XT, YP - YT), 그리고 점 T에서 점 S로 향하는 벡터(XS - XT, YS - YT)를 구한다. 만일 점 S가 보이는 점이면, 점 S는 선 QT에 대하여 점 P와 같은 쪽에 놓이게 된다. 그러면 두 벡터(XS - XT, YS - YT)와 $\vec{u}$의 外積의 Z軸 成分은 두 벡터(XP - XT, YP - YT)와 $\vec{u}$의 外積의 Z軸成分과 같은 부호를 갖게 된다. 만일 보이지 않는 점이라면 外積의 결과는 서로 다른 부호가 되므로, 프로그램에서 이 결과를 比較하면 된다.

위의 과정을 7번 반복하여 구해진 (XS, YS) 값을 점 S의 좌표 값으로 취하게 된다. 이때 벡터($P_1$, $P_2$)의 크기는 두점(XP, YP)와 (XR, YR) 간의 거리의 약 1/128에 해당한다. 이렇게 함으로써 컴퓨터 디스플레이 화면에서 만족할만한 정확도를 갖는 画像을 얻어 낼 수 있다. 정확도는 반복회수를 증가함으로써 얻을 수 있게 된다. 반복 루우프에서 빠져나오면 점 P에서 점 S로 선을 긋도록 하는 方法을 취하고 있다.

다음에는 부분적으로 가리워진 다른 선에 대하여 같은 과정을 반복하면 된다.