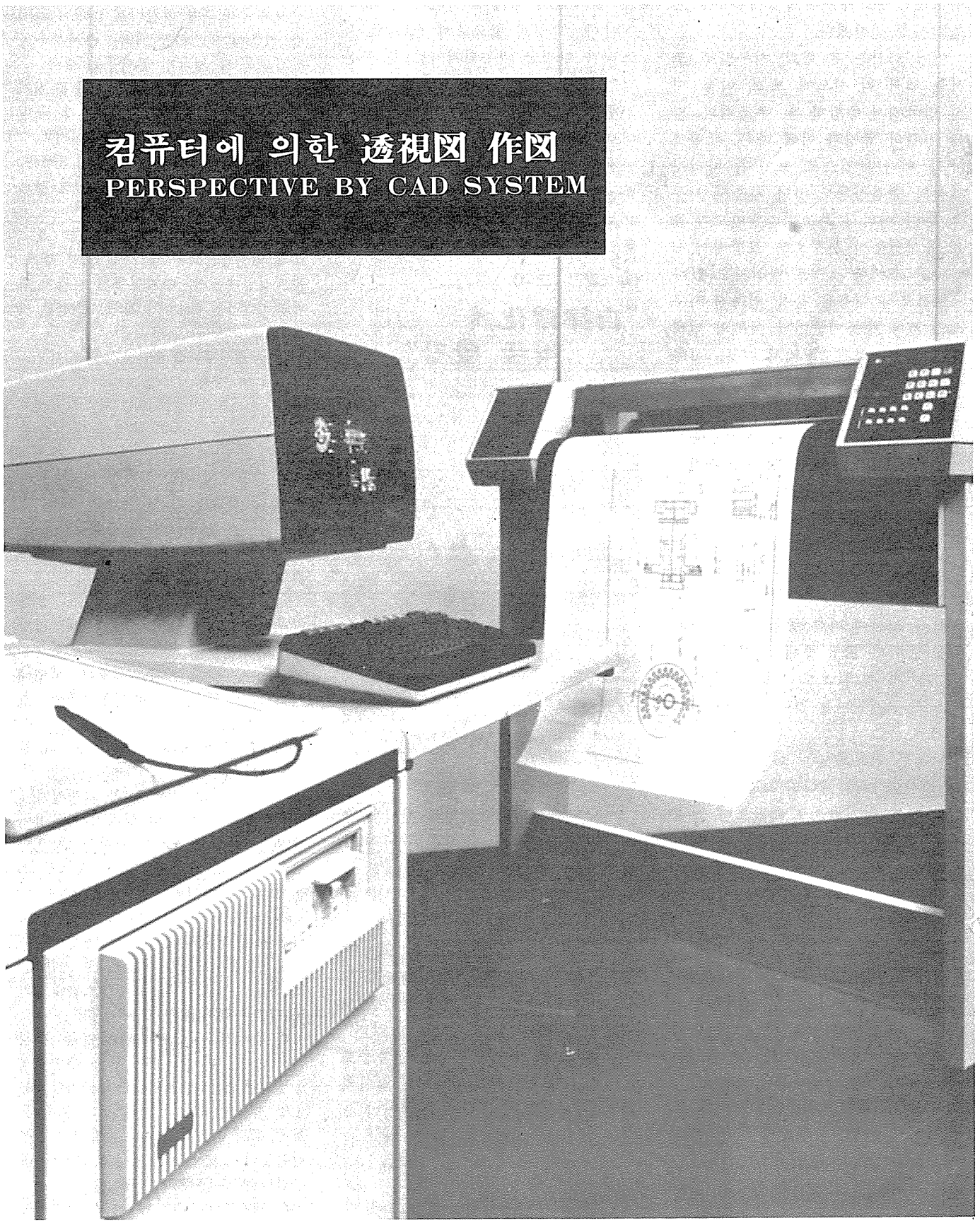
컴퓨터에 의한 透視図 作図
PERSPECTIVE BY CAD SYSTEM



『그림－1』컴퓨터 시스템(CAD 用)

曺 鐵 鎬 － 건축사・建国大교수
CHUL－HO CHO / KONKUK UNIV.

各分野에 걸쳐 컴퓨터의 活用에 대한 관심이 높아지고 있다. 建築士誌에서는 지난 4月号부터 5차례에 걸쳐 『設計의 컴퓨터手法』(日本·飯塚英雄)을 소개한 바 있다.

建築作品活動에 직접 참여 하시는 분들의 컴퓨터에 의한 設計에 관하여 관심은 많으나 실감을 느끼시기가 어려운 듯하다. 筆者나름대로 이 분야에 경험한 것을 토대로 컴퓨터에 의한 建築設計(Compute Aided Design)의 実際를 소개함으로 앞으로 몇년 후 이러한 CAD 시스팀에 접할 기회가 있을 때 다소나마 도움이 된다면 퍽 다행이라 생각하면서 筆者의 拙文을 전개하려 한다.

컴퓨터에 의한 建築分野의 活用은 우리나라에서 1970년 초반 構造解析에 大型컴퓨터를 이용해 오다가 1970년 후반 몇 분의 構造技術者가 개인용 컴퓨터(Desk -top Computer)를 직접 소유하여 構造解析 및 斷面算定에 활용하면서 상당한 진보가 있었다.

建築계획에 있어서는 1980년 초반부터 外国의 BDS, GDS, CADAM 등의 Package 의 활용이 이루어 지고 있으나 아직 본격적인 이용에는 시간을 要하는 것으로 보여진다.

그러나 컴퓨터의 利用度는 建築에서도 점차 높아지고 컴퓨터의 가격이 점차 저렴해져 建築設計研究所에서도 Ao(대판) 크기의 프롯터(Plotter)를 도입하여 活用될 날이 머지 않았으며 몇년 후에는 보편적으로 이용될 전망이다.

筆者는 평소 컴퓨터에 관심이 많아 1982년 12월 Ao 크기의 프롯터를 구입하여 建築構造骨組図面의 프로그램을 개발한 바 있다.

建築構造骨造의 형태를 透視図作図技法을 이용해 본 결과 視覚的이어서 퍽 효과적이었다.

여기에 사용했던 프로그램의 일부를 독립시켰더니 이것은 建築設計의 計劃단계에서 透視図나 모형의 효과를 同一하게 얻을 수 있는 시뮬레이션(Simulation)을 TV 화면과 비슷한 CRT에 나타낼 수 있었다. 筆者자신이 構造를 전공한 사람으로 設計分野에 속하는 내용을 소개하는 것이 어떨지 모르겠으나, 会員 여러분들의 많은 指導편달을 바라면서 透視圖의 컴퓨터応用에 대하여 소개하고자 한다.

1. 透視図의 原理

平行線은 하나의 점에 集合된다는 약속에 의해 一定한 法側에 따라 透視図를 그리게 된다.

透視図의 基本的인 개념은 눈과 物体의 各点을 直線으로 연결하고 그 直線의 모임을 任意의 点에서 切斷한 面에 나타나는 畵像이다라 할 수 있다.

즉 유리창을 통하여 보는 건물을 그 유리면에 그대로 나타낸 그림이 透視図이다. 카메라로 찍은 사진과 같은 것이다. 하나의 눈을 통해 3次元의 物体를 2次元의 平面으로 대치하는 것이고 그 대치하는 방법이 図法이다.

2. 透視図 図法의 種類

透視図는 対象物이 되는 立体(3次元)를 보는 位置에 따라 대개 3가지로 나눌 수 있다.

1点 透視…… 対象物을 正面으로 본 경우이며 室内透視図에 쓰인다.

2点 透視…… 対象物을 斜面으로 角을 지워 본 境遇로 建物透視図에 많이 쓰인다.

高層建物인 경우 윗面이 넓어 보이므로 이 경우는 3点 透視図法으로 作図하는 것이 좋다.

3点 透視…… 対象物을 斜面으로 내려다 보거나 올려다 보는 경우로 실제에 거의 가까운 図法이다.

3 컴퓨터에 의한 透視図의 実例

높이 79.5m, 길이 65.4m, 폭 36.6m인 20층 建物을 前面 160m(건물높이의 2倍), 높이 160m에 60度의 角으로 본 透視図의 実例가 『그림-2』와 같다.

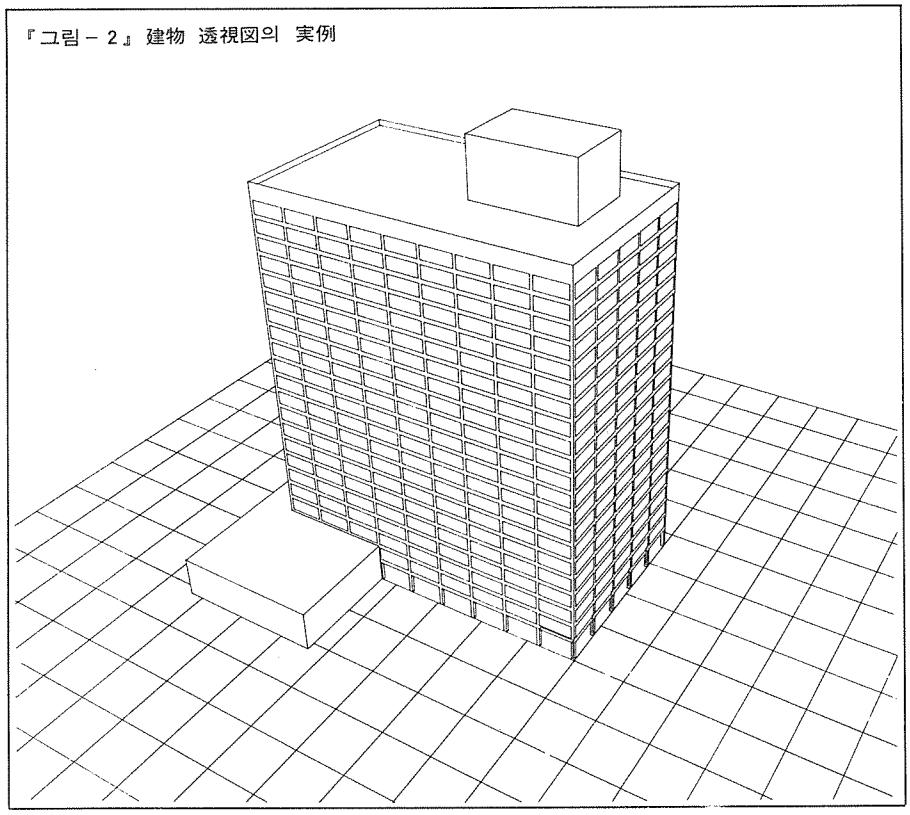
컴퓨터에 의하면 建物의 모든 点을 3次元으로 入力시켜 2次元으로 변환시키므로 透視図의 作図가 가능하다.

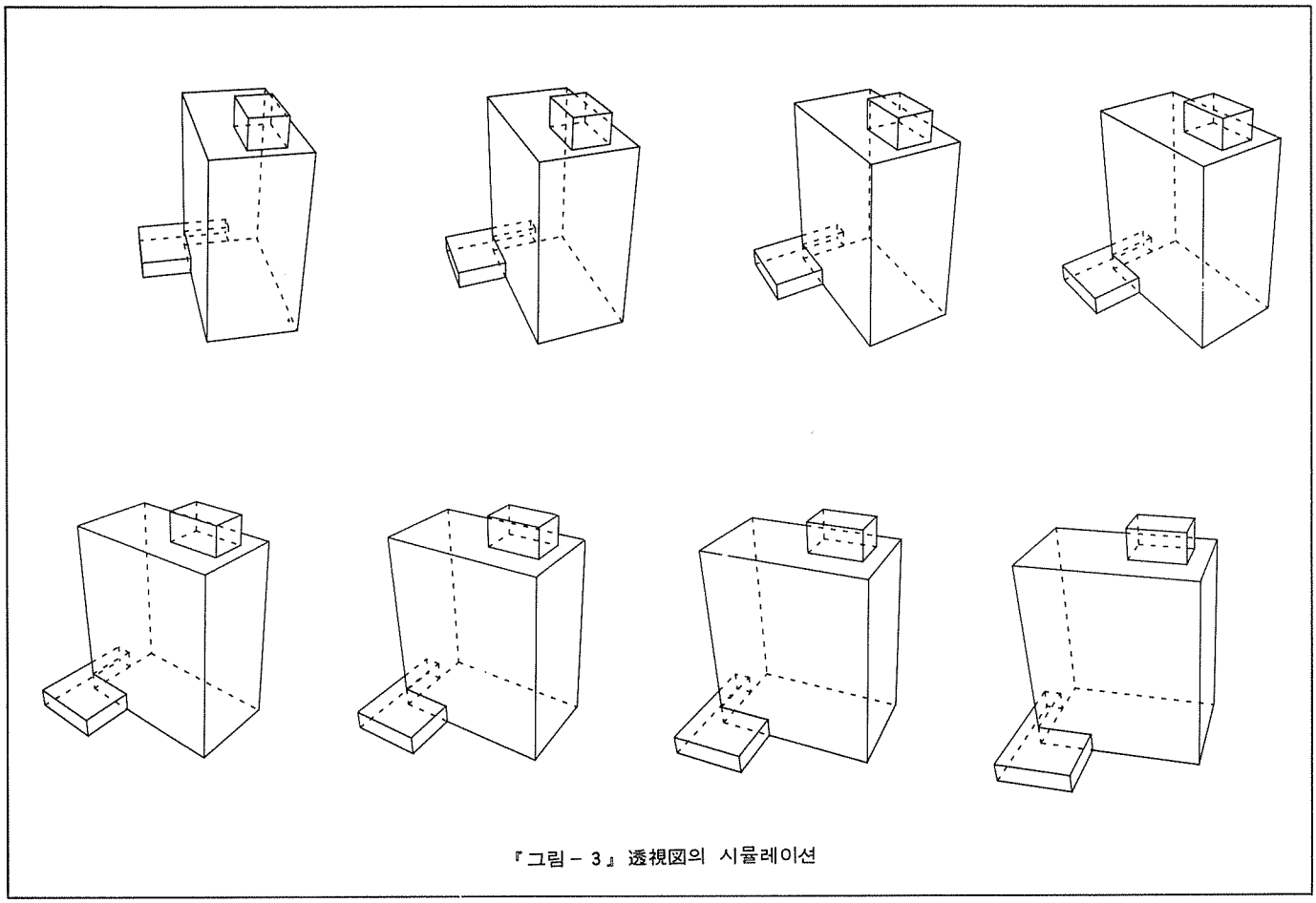
한번 入力시킨 点들의 資料는 변화시키지 않아도 임의 위치에서의 2次元으로 平面化하는 것은 간단하다.

『그림-3』과 같이 同一한 임의의 位置에서 角을 10度에서 80度까지 변화시키면 本透視図를 그리기 전에 변화하는 모양을 볼 수 있다.

여기서 가장 보기가 좋다고 생각되는 角度를 골라 本透視圖로 作図할 수 있는 시뮬레이션(Simulation)이 가능한 것이다. 물론 建物의 回轉은

『그림-2』 建物 透視図의 実例





『그림－3』 透視図의 시뮬레이션

360度 모두 가능해서 後面의 透視도 나타낼 수 있다. 눈높이도 조정할 수 있어 建物높이보다 낮게할 수도 있고 심지어 建物 밑에서 올려다 볼 수도 있다.

컴퓨터를 이용할 경우, 하드 웨어(Hard Ware) 가격은 본체가 CRT 포함해서 약 $30,000, Ao 크기의 프롯터가 $30,000로 高價이나 점차 가격이 떨어지고 있는 추세이다.

建築設計를 위한 선진국의 소프트웨어(Soft Ware) 가격이 $100;000 ~$200,000로 高価라 아직 建築을 전공하는 사람들이 보편적으로 CAD (Computer Aided Design) 시스팀을 활용하는 것은 쉬운 일이 아니라고 보나 몇년 후에는 모든 조건이 좋아져 가능하리라 본다.

4. 컴퓨터에 의한 透視図 作図

透視図에 컴퓨터를 이용하는 시도는 초기단계부터 시행되어 왔다. 그러나 線画때문에 표현력이 부족하고 숨어있는 線인 隱線(Hidden Line) 처리를 한 경우 시간이 많이 걸렸으므로 完成図를 이용하는 것보다 그 당시 컴퓨터의 特性을 살려서 外部시퀜스 경관이나 볼륨 체크 등에 이용되어 왔었다.

최근에 와서는 컴퓨터의 고성능화·고속화에 따라 시간단축과 메트릭스法(Matrix Method)의 解析이 쉽게 되고 CRT의 粒度가 1024×1024로 선명해 지고 색채画質이 양호해져 손으로 그리는 透視図 못지않게 되었다.

『그림－2』는 隱線을 처리한 것이고, 『그림－3』은 隱線을 점선으로 나타낸 것이다.

일반적인 透視図의 作図方法은 消点에서 線을 구해 面積으로 나타내므로 作図되는 범위가 한정되어 그를 확대해야 하는 불편이 따른다.

컴퓨터에 의하면 1点 透視·2点 透視·3点 透視의 방법을 쓰는 것이 아니라, 카메라로 사진을 찍으므로 얻을 수 있는 영상과 동일한 画像을 얻을 수 있는 長点이 있다.

構造解析의 技法의 하나인 메트릭스法(Matrix Method)을 써서 座標를 변환함으로 간단히 3次元의 立体建物을 2次元의 平面透視図로 나타낼 수 있게 된다.

따라서 컴퓨터에 의한 透視図는 3点 透視 방법보다 実際와 같은 画像인 것이다.

또 일단 2次元으로 구해 놓은 透視図의 모든 点을 이용하여 자유자재로 확대 축소가 가능하다.

그리고 현재까지 써온 방법에 의하면 그 기본형태가 육면체 정도로 제한되어 있어서 형태가 복잡하면 육면체로 분할하여 해결하였으므로 부정형의 형태는 정확히 그리기가 어려웠다.

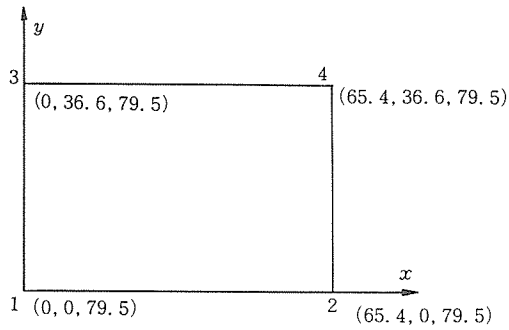
그러나 컴퓨터에 의하면 거의 모든 형태(원형·타원형 등)의 透視図를 0.012mm 까지의 오차인 精度로 정확히 作図할 수 있다. 또 다른 위치에서 본 透視度로 Feed Back 을 짧은 시간 내에 할 수 있다.

5. 資料의 入力方法

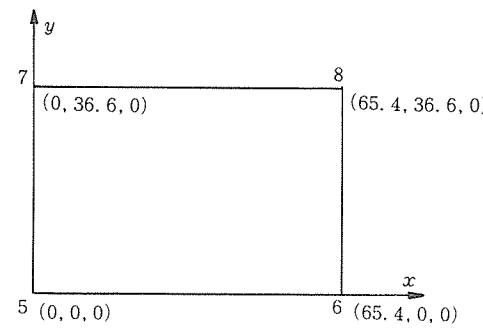
立体構造物의 各節点을 入力하는 방법과 同一하게 $x \cdot y \cdot z$ 座標로 3次元 入力으로 建物의 形態를 컴퓨터가 알 수 있게 한다.

직교좌표계가 아닌 원통좌표계(거리·경도·높이)나 구좌표계(거리·위도·경도)로 入力도 가능하다.

『그림－2』와 같은 建物의 예를 들면 다음과 같다.

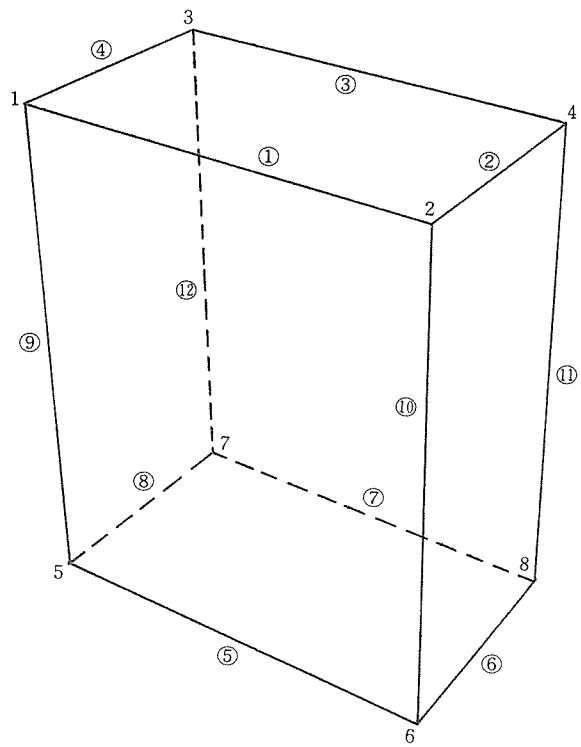


옥상부분 (G. L +79. 5)



지층부분 (G. L +0. 0)

『그림 - 4』 건물의 좌표



『그림 - 5』 건물의 線

点의 座標

点	x 座標	y 座標	z 座標
1	0	0	79.5m
2	65. 4	0	79. 5
3	0	36. 6	79. 5
4	65. 4	36. 6	79. 5
5	0	0	0
6	65. 4	0	0
7	0	36. 6	0
8	65. 4	36. 6	0

線의 表現

線	앞 点	뒷 点
1	1	2
2	2	4
3	4	3
4	3	1
5	5	6
6	6	8
7	8	7
8	7	5
9	1	5
10	2	6
11	4	8
12	3	7

『그림 - 3』과 같은 建物은 点의 数가 34, 線의 数가 40 정도이지만 『그림 - 3』과 같은 実際建物은 창문 등의 座標를 다 나타내면 点의 数가 1400, 線의 数가 약 1500 정도로 入力資料를 만드는데 약 4 시간 정도 要한다.

点의 번호를 규칙적으로 붙여 入力을 간결화하면 時間단축이 可能하다.

3 次元 座標를 2 次元 平面 座標로 바꾸어 透視図로 만드는데 16 Bit Desk -top 컴퓨터로 약 20분이 소요된다. 32Bit 로는 약 2 분 정도로 단축이 된다.

6. 座標시스템

3 次元인 実物에 해당하는 建物을 2 次元으로 나타내기 위해서는 우선 3 次元 공간의 点들을 어떻게 표현할 것인가에 대한 약속을 정해 익힐 필요가 있다. 点의 위치를 나타내기 위하여 座標시스템이 필요하다.

3 次元 座標시스템으로 가장 많이 쓰이는 것은 直角座標시스템과 球座標시스템이다.

6. 1 直角座標시스템

2 次元 座標시스템에서 사용되던 X, Y軸에 원점을 지나면서 XY 平面에 수직인 Z軸을 설정함으로써 3 次元 座標시스템이 구성된다. 따라서 3 次元 座標시스템은 서로 수직으로 교차하는 세개의 軸으로 구성된다.

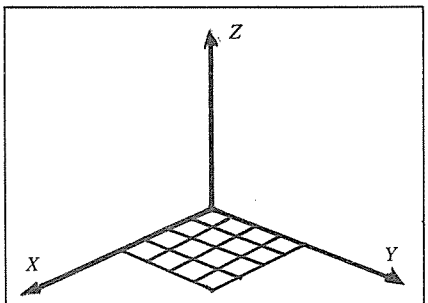
Z軸의 方向을 정하는 방법에는 2

가지 방법이 있다. 그 한가지 방법은 오른손 시스템(Right-hand System)을 구성하는 것이다. 오른손의 집게 손가락과 가운데 손가락이 각각 X, Y축을 가리킬 때 엄지 손가락은 Z축을 가리키는 시스템이다.

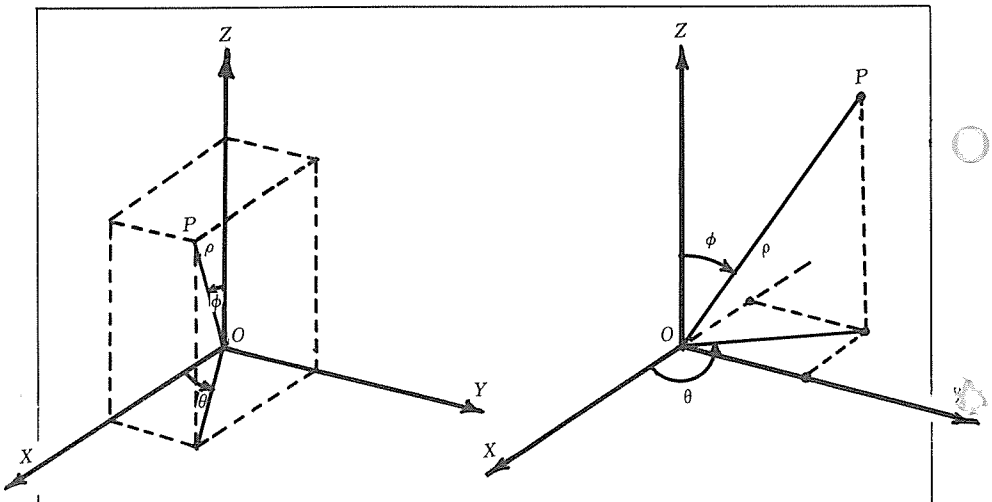
왼손 시스템(Left-hand System)에서는 Z축의 방향이 오른손 시스템에서와 반대이다.

보통 편의상 오른손 시스템을 표준으로 채택하고 있다. 오른손 시스템이나 왼손 시스템을 선택하는 데에는 특별한 원칙은 없다. 어느 시스템을 선택하든지 원하는 연산을 충분히 처리할 수 있어 상관없다.

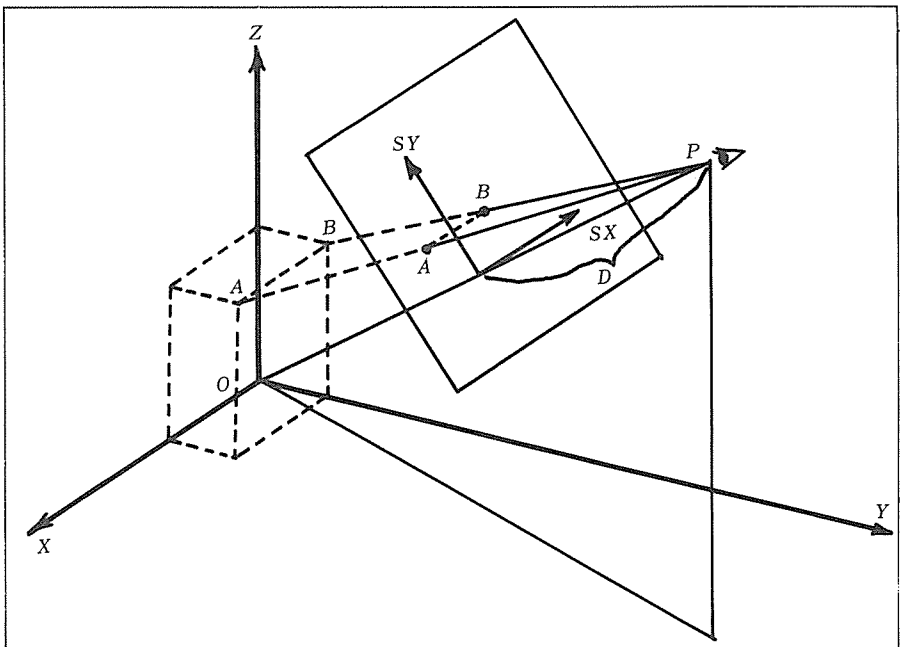
3개의 좌표축 중에서 2개는 좌표평면이라고 하는 하나의 평면을 구성하게 된다. 이 좌표 평면에는 XY평면, YZ평면, XZ평면이 있다. 공간상의 点들은 x, y, z의 3개의 숫자로서 (a, b, c)와 같이 표현되는데, 이때의 a, b, c 값은 각각 YZ평면, XZ평면, XY평면으로부터의 수직 거리이다.



『그림－6』 오른손 좌표시스템



『그림－7』 球좌표시스템



『그림－8』 관측자와 물체 사이에 놓인 화면

6.2 球座標시스템(Spherical Coordinate System)

球座標시스템은 『그림－7』과 같이 표현된다.

이 시스템에 의하면 点 P는 3개의 값 ρ, θ, ϕ로 표현된다. ρ는 원점으로부터 그 点까지의 거리를 나타내고, ϕ는 선분 OP와 Z축의 正(+)의 軸과 이루는 角을 나타낸다. θ는 X軸의 正의 軸과 선분 OP를 XY평면에 투영해서 얻어진 선분과 이루는 角을 나타낸다. 이 경우 θ는 Z축의 正의 方向에서 볼 때 반시계 방향으로 이루는 角으로 표현된다.

3次元 공간의 어떤 点은 (X, Y, Z)나 (ρ, θ, ϕ)로 표현될 수 있어, 다음과 같이 삼각함수를 이용하여 변환방법의 관계식을 얻을 수 있다.

$$x = \rho \sin\phi \cos\theta$$
$$y = \rho \sin\phi \sin\theta$$
$$Z = \rho \cos\phi$$
$$\rho^2 = x^2 + y^2 + z^2$$

6.3 화면 좌표의 구성

3次元 物体의 모양을 2次元 그래픽 디스플레이 화면(CRT화면)에 나타내기 위해서는 화면 좌표(SX, SY)와 직각 좌표(X, Y, Z) 또는 球좌표(ρ, θ, ϕ)간의 관계를 알아야 한다.

『그림－8』에서 관측하는 사람의 위치는 点 P로 나타나 있다. 디스플레이 화면은 物体가 투영될 평면인데, 이 투영평면은 선분 OP와 수직이고 점 P로 부터 D만큼 떨어져 있다고 하면, 물체의 각 점들이 투영평면에 투영됨에 따라, 각 점(X, Y, Z)에 대한 화면 좌표 (SX, SY)가 얻어진다.

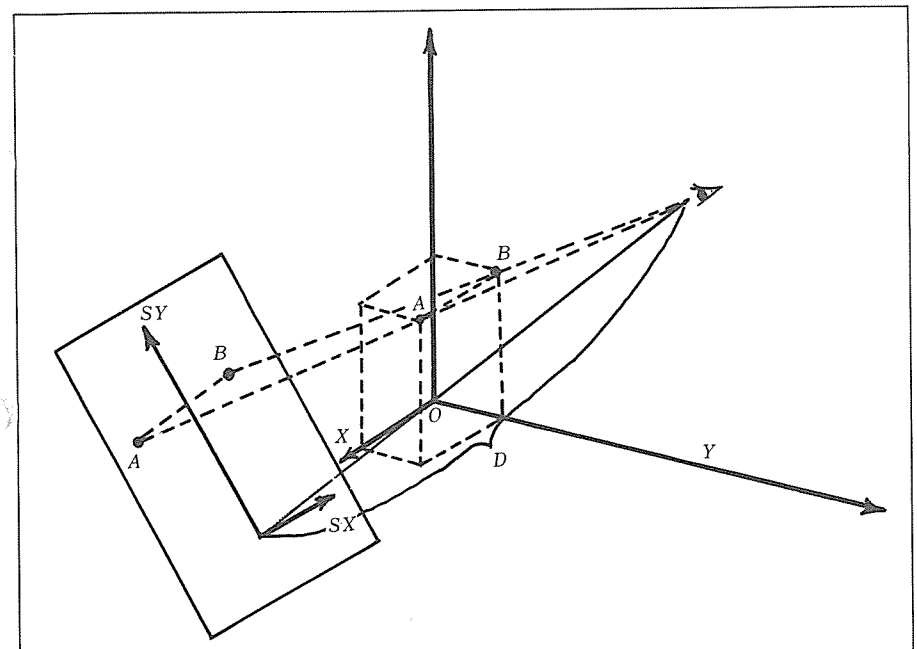
여기서 관측점의 위치는 球좌표로 표현할 수도 있다. 球좌표로 직각좌표 표현법보다 쉽게 관측점의 위치를 지정 변경할 수 있다.

球좌표에서 ρ는 관측자와 원점과의 거리를 나타낸다. ρ값을 증가시키면 관측자는 물체로부터 멀어지게 된다.

따라서 물체는 보다 작게 보인다.

θ와 ϕ는 관측자가 물체를 보는 방향을 지정해 주게 된다.

『그림－8』에서는 투영화면이 관측자와 물체 사이에 놓여 실물보다 작게 보인다. 『그림－9』에서는 투영화면이 관측자로 부터 물체의 반대쪽에 놓여 있어 실물보다 큰 화상을 얻을



『그림－9』 물체에 대하여 관측자와 반대 쪽에 놓인 화면

수 있다.

『그림－8』에서 투영화면을 관측자 쪽으로 가까이 가져올수록 투영된 화상은 점점 더 작아지고, 『그림－9』에서 투영화면이 관측자로부터 멀어질수록 투영된 화상은 점점 더 커진다.

그러므로 여기서 관측 변수 ρ, θ, ϕ, D를 변경시킴으로써 화상의 위치와 방향 크기를 바꿀 수 있음을 알 수 있다.

建物과 같은 것은 実物이 크므로 『그림－8』과 같은 방법을 이용하고 있다.

7. 座標의 各種 変換 (Transformations)

3次元의 空間에 위치한 点을 동차좌표 (Homogeneous Coordinate) 라는 좌표로 나타낸다. 직각좌표로 표현된 점 (X, Y, Z)은 동차좌표로는 $(X, Y, Z, 1)$로 표현된다. 따라서 선형이동 변환도 4×4行列로 표현시킬 수 있다. 자주 사용되는 変換을 알아보면 다음과 같다.

7.1 크기조절 변환(Scaling)

대각선 行列은 크기 조절 行列로서 쓰인다. 대각선상의 각 값들은 해당 좌표축 방향으로의 크기 조절에 각각 사용된다.

$$(X, Y, Z, 1)\begin{bmatrix} A & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= (AX, BY, CZ, 1)$$

모두 같은 크기로 조절하려면 다음과 같이 하면 된다.

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

$$= (KX, KY, KZ, 1)$$

어떤 점 (X, Y, Z, D)은 표준 형태로 $(X/D, Y/D, Z/D, 1)$이므로 모두 같은 크기로의 변환行列은 다음과 같은 방법도 가능하다.

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

7.2 回転 変換(Rotation)

3次元에서는 物体를 각 좌표축에 대해 회전시키는 行列들만 구하면 충분하다.

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

벡터 $(X, Y, 1)$에 R을 곱하여, 그 벡터를 원점을 중심으로 반시계 방향으로 θ만큼 회전시킨 경과로 얻을 수 있다. 만약 Z축을 XY평면으로부터 우리들을 향하는 방향으로 있다고 가정하면, 위와 같은 회전은 Z축에 대하여 실제 회전시키는 것과 마찬가지가 된다.

3次元 벡터에 대한 회전에는 다음과 같은 변환 行列을 사용한다.

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

마찬가지로 X, Y축에 대한 회전에는 각각 다음과 같은 행렬을 사용한다.

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

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

각각의 회전에 있어서, θ는 회전축의 正(+, Positive)의 방향에서 원점을 향하여 바라볼 때, 반시계 방향으로 회전한 각을 나타낸다.

7.3 선형이동 변환(Translation)

만일 선형이동 변환이 필요하지 않다면 동차 좌표(Homogeneous Coordinate) 표현법을 쓸 필요가 없다.

동차 좌표가 선형 이동 변환을 행렬로 나타내는데 매우 유용하다.

어떤 점들을 (k, l, n) 만큼 이동시키려면 아래와 같이 하면 되는데, 이러한 것을 다른 측면에서 보면 원점을 점 $(-k, -l, -n)$로 이동 시키는 것과 같음을 알 수 있다.

$$(X, Y, Z, 1)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & l & n & 1 \end{bmatrix}$$

$$= (X+k, Y+l, Z+n, 1)$$

7.4 反射 변환(Reflection)

어떤 물체를 어떤 한 平面에 대하여 反射, 즉 対称移動시키는 것은 그 평면의 반대쪽에 그 물체가 거울에 반사되는 것처럼 위치시키는 것과 같은 것이다. 따라서, 어떤 점을 XY 평면에 대해 대칭이동시키는 것은 그 점의 Z좌표 값의 부호를 바꾸는 것과 같다.

이와 같은 反射 변환 行列은 다음과 같다.

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

XZ 평면에 대한 反射 行列은

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

YZ평면에 대한 反射 行列은

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

7.5 順次変換 (Sequential Transformation)

앞에서의 변환 행렬들을 순차적으로 변환시킴으로써 다양한 변환 효과를 얻을 수 있다. 이 때의 변환행렬은 각각의 단위 변환 행렬들을 순서대로 곱함으로써 하나의 행렬 형태를 가지도록 할 수 있다.

예를 들어, 다음과 같이 선분 AB ($A=(0,2,3)$, $B=(1,2,3)$)를 반시계 방향으로 30° 회전시키는 과정을 생각하면 다음과 같다.

1. 선형이동 변환을 $(0,-2,-3)$ 만큼 한다.

$$T=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&-2&-3&1\end{bmatrix}$$

2. 30° 만큼 반시계 방향으로 회전 변환한다

$$R=\begin{bmatrix}1&0&0&0\\0&cos30°&sin30°&0\\0&-sin30°&cos30°&0\\0&0&0&1\end{bmatrix}$$

3. 다시 원래의 위치 $(0,2,3)$로 선형이동한다

$$T^*=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&2&3&1\end{bmatrix}$$

위의 순서를 다음과 같은 하나의 변환행렬로 만들 수도 있다.

$$M=TRT^*=\begin{bmatrix}1&0&0&0\\0&cos30°&sin30°&0\\0&-sin30°&cos30°&0\\0&-2cos30°+3sin30°+2&-2sin30°-3cos30°+3&1\end{bmatrix}$$

$$=\begin{bmatrix}1&0&0&0\\0&.866&.500&0\\0&-.500&.866&0\\0&1.768&-.598&1\end{bmatrix}$$

변환 순서는 매우 중요한 것이다.

일반적으로 변환 순서가 바뀌면 결과적으로 변환 효과도 바뀌게 되므로 순서에 유의해야 한다.

선형 이동한 후에 회전하는 것과, 회전을 먼저 한 후 선형 이동하는 것은 전혀 다른 효과를 나타낸다.

7.6. 逆行列(Inverse Matrix)

위의 예에서 T와 T^*는 서로 반대의 효과를 나타내고 있다. 行列 T는 $(0,-2,-3)$만큼 선형이동시키는 것이고, 行列 T^*는 $(0,2,3)$ 만큼 선형이동시키는 것이다. 이 두 行列을 잇달아 곱하면, 대상물은 원래의 위치로 돌아온다.

$$TT^*=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&-2&-3&1\end{bmatrix}$$

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&2&3&1\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}=I$$

두 行列의 곱이 단위행렬 I(대각선이 모두 1이고 나머지는 0인 行列)가 될 때, 이들은 서로 상대방 행렬의 역행렬이라고 한다, 즉, T는 T^*의 역행렬이고, 또 T^*는 T의 역행렬이다.

어떤 행렬 N의 역행렬을 보통 N^{-1}로 표현한다.

어떤 행렬은 역행렬을 갖지 못하는 경우도 있지만, 컴퓨터 그래픽에서 사용되는 행렬들은 모두 역행렬을 가지고 있다.

예를 들어, X축에 대하여 30° 회전시키는 변환행렬 R은 역행렬로서 X축에 대하여 −30° 회전시키는 변환행렬 R^{-1}을 갖는다.

$$R=\begin{bmatrix}1&0&0&0\\0&cos30°&sin30°&0\\0&-sin30°&cos30°&0\\0&0&0&1\end{bmatrix}$$

$$=\begin{bmatrix}1&0&0&0\\0&.866&.500&0\\0&-.500&.866&0\\0&0&0&1\end{bmatrix}$$

$$R^{-1}=\begin{bmatrix}1&0&0&0\\0&.866&-.500&0\\0&.500&.866&0\\0&0&0&1\end{bmatrix}$$

7·7. 좌표시스팀 변환

좌표축을 변환시켜서 새로운 좌표시스팀을 구성하는 것이 필요하다.

원점을 새로운 위치로 이동시킬 수도 있겠고, 좌표 시스팀을 어떤 한 축에 대하여 회전시킬 수도 있다.

物体의 점들을 원래의 좌표로 나타내고, 변환된 새로운 좌표 시스팀으로 표현할 때 대개 프라임(Prime)기호(′)를 사용하여 구분한다.

따라서 어떤 점 $(3,2,1)$은 표준 좌표시스팀으로 표현한 것이고, $(3,2,1)'$은 새로운 좌표시스팀에 의한 표현이다.

예를 들어, Z축에 대해 반시계 방향으로 30° 회전시키는 변환 行列은 다음과 같다.

$$N=\begin{bmatrix}cos30°&sin30°&0&0\\-sin30°&cos30°&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$=\begin{bmatrix}.866&.500&0&0\\-.500&.866&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

이러한 변환 행렬을 어떤 점의 좌표 벡터에 곱하면, 결과로서 30° 회전된 점이 얻어진다.

예를 들어, 점 $(2,4,1)$은 다음과 같이 회전되어 점 (−0.268, 4.464, 1)로 된다.

$$(2,4,1,1)\begin{bmatrix}.866&.500&0&0\\-.500&.866&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$=(-.268,4.464,1,1)$$

어떤 한점 P'는 점 P를 Z축에 대하여 반시계 방향으로 30° 회전하여 얻어진 점이라고 하고, 좌표시스팀을 Z축에 대하여 시계 방향으로 30° 회전시킨 경우를 생각해 보면, 새로 얻어진 좌표시스팀 X', Y', Z'은 점 P를 새로운 값으로 표현할 것이다. 원래의 X, Y, Z좌표시스팀으로 표시된 점 P'의 좌표값은 점 P를 반시계 방향으로 30° 회전시켜 얻은 점으로 점 P를 새로운 X', Y', Z' 좌표시스팀으로 표시한 좌표값과 같은 것을 알 수 있다.

行列 N은 다음과 같은 두가지 용도로 쓰임을 알 수 있다.

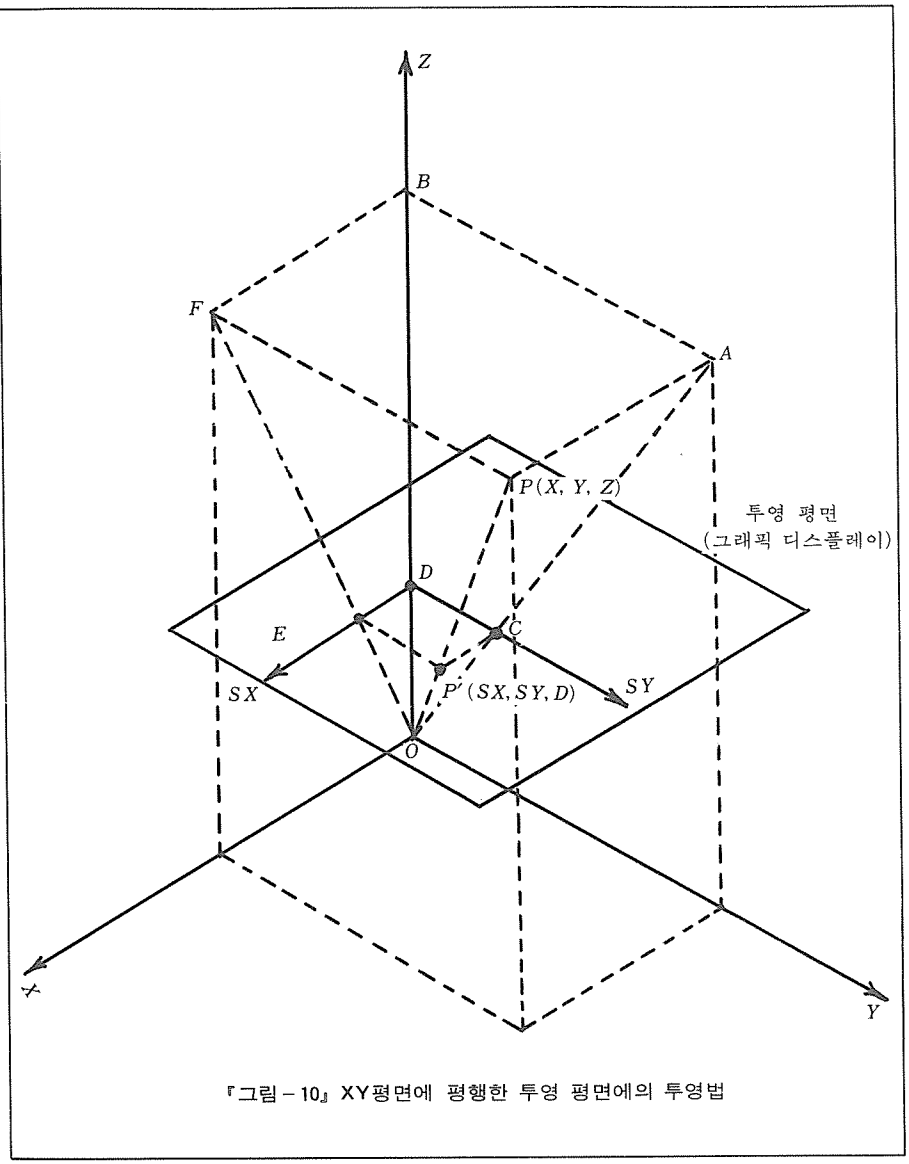
첫째 어떤 점들을 반시계 방향으로 30°만큼 회전시켜 그 때의 좌표값을 계산한다.

둘째 좌표시스팀을 시계 방향으로 30° 만큼 회전시켜 얻어진 새로운 좌표시스팀으로 점들의 좌표값을 계산한다.

위의 예로서 다룬 내용은 일반적으로 적용되는 것이다. 좌표 시스팀을 변환시키는 데 사용되는 行列은 한 좌표시스팀 내에서의 점을 회전시키는 데 사용되는 行列의 逆行列인 것이다.

7·8. 投影(Projection)

3次元인 建物을 2차원으로 표현하기 위하여 다음에는 물체의 각 꼭지점들을 평면에 투영시키는 방법으로 YX평면과 평행한 평면에 투영시키는 것에 대하여 먼저 알아 둘 필요가 있다.



『그림－10』 XY평면에 평행한 투영 평면에의 투영법

『그림－10』에서 투영 평면은 XY평면으로 부터 D만큼 떨어져 있다고 보면, 투영 평면을 컴퓨터 디스플레이 화면으로 생각할 수 있다.

여기서 평면의 각 점들은 화면 좌표(SX, SY)로 표현된다. SX와 SY 축은 각각 X축, Y축과 평행하다고 가정하고, 투영 평면에서의 스케일(Scale)은 원래의 좌표시스템에서와 같다. 따라서 투영 평면상의 점(SX, SY)은 원래 좌표시스템에서의 점(X, Y, D)와 같다. 여기서 $X=SX$, $Y=SY$이다.

투영 방법에는 여러 가지가 있는데, 여기서 사용할 방법은 원점에 투영 중심이 있는 원근투영법(Perspective Projection Method)이다. 어떤 점 $P(X, Y, Z)$는 투영평면과 선 OP 와의 교점으로 투영된다.

『그림 －10』에서 직각 삼각형 OBA와 ODC는 서로 닮은 꼴이므로 다음과 같은 식이 성립된다.

$$\frac{DC}{OD}=\frac{BA}{OB}\rightarrow\frac{SY}{D}=\frac{Y}{Z}\rightarrow SY=D\left(\frac{Y}{Z}\right)$$

같은 방법으로, 직각 삼각형 OBF와 ODE 사이에는 다음의 관계식도 성립한다.

$$\frac{DE}{OD}=\frac{BF}{OB}\rightarrow\frac{SX}{D}=\frac{X}{Z}\rightarrow SX=D\left(\frac{X}{Z}\right)$$