

## 6 좌표변환과 컴퓨터 그래픽스

### 6.1 좌표변환

#### 6.1.1 서론

geometric modelling  
computer graphics

-> coordinate transform ; 2차원 -> 2차원, 3차원 -> 3차원  
projection transform ; 3차원 -> 2차원

#### 6.1.2 2차원 좌표변환

translation  
rotation  
scaling

Homogeneous coord.

Homogeneous transform.

Homogeneous 좌표 (a, b, w) -> 실제 좌표  $x = a/w, y = b/w$

Translation;

$$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}, \quad (x', y', 1) = (x, y, 1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Scaling

$$\begin{aligned} x' &= x \cdot s_x \\ y' &= y \cdot s_y \end{aligned}, \quad (x', y', 1) = (x, y, 1) \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation

$$\begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta \end{aligned}, \quad (x', y', 1) = (x, y, 1) \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

예제 6.3)

$p(4,2) \rightarrow t_x=-2, t_y=-1, s_x=s_y=2, t_x=2, t_y=1$

$$M = T_1 S T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ -2 & -1 & 1 \end{bmatrix}$$

$$(x, y, w) = (4, 2, 1) \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ -2 & -1 & 1 \end{bmatrix} = (6, 3, 1)$$

### 6.1.3 3차원 좌표변환

x axis Rotation ( $x'=x$ )

$$\begin{aligned} y' &= y \cos\theta - z \sin\theta \\ z' &= y \sin\theta + z \cos\theta \end{aligned}, \quad (x', y', z', 1) = (x, y, z, 1) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & s\theta & 0 \\ 0 & -s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

y axis Rotation ( $y'=y$ )

$$\begin{aligned} z' &= z \cos\theta - x \sin\theta \\ x' &= z \sin\theta + x \cos\theta \end{aligned}, \quad (x', y', z', 1) = (x, y, z, 1) \begin{bmatrix} c\theta & 0 & -s\theta & 0 \\ 0 & 1 & 0 & 0 \\ s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

z axis Rotation ( $z'=z$ )

$$\begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta \end{aligned}, \quad (x', y', z', 1) = (x, y, z, 1) \begin{bmatrix} c\theta & s\theta & 0 & 0 \\ -s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

\* 3차원 변환식

$$(x', y', z') = (x, y, z)R + T$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad T = (t_1, t_2, t_3)$$

## 6.2 2차원 컴퓨터 그래픽스

### 6.2.1 PC를 이용한 컴퓨터 그래픽스

Turbo-C language

Graphics

- 초기화 -> graphics mode로 전환
- text mode로 전환
- one pixel
- 위치이동
- 선 긋기
- 직사각형 그리기
- 원그리기

예제 6.4)

3점 (300,150), (150,400), (500,350)

```
# include <stdio.h>
```

```
# include <conio.h>
```

```
# include <graphics.h>
```

```
void main(void)
```

```
{
```

```
int gdriver=DETECT, gmode;
```

```
initgraph(&gdriver, &gmode, "c:\\tc");
```

```
moveto(300,150);
```

```
lineto(150,400);
```

```
lineto(500,350);
```

```
lineto(300,150);
```

```
getch();
```

```
closegraph();
```

```
}
```

## 6.2.2 2차원 컴퓨터 그래픽스를 위한 좌표변환

CRT에 도면을 그리기 위한 작업

CRT; 640x480 이므로  $s_x=1\text{cm}/27$ ,  $s_y=1\text{cm}/28$

- (1) 도형의 중심이 원점이 되도록;  $T(10,10)$
- (2) 이동된 도형을 뒤집고  $s_x=1$ ,  $s_y=-1$ ; CRT비율로 축척변환 ( $s_x=27$ ,  $s_y=28$ )
- (3) 변환된 도형을 CRT 중심으로 이동;  $T(320,240)$

$$\begin{aligned} M &= T(-10, -10) \cdot S(1, -1) \cdot S(27, 28) \cdot T(320, 240) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -10 & -10 & 1 \end{bmatrix} \begin{bmatrix} 27 & 0 & 0 \\ 0 & -28 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 320 & 240 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 27 & 0 & 0 \\ 0 & -28 & 0 \\ 50 & 520 & 1 \end{bmatrix} \end{aligned}$$

window 중심;  $(x_w, y_w)$

viewport 중심;  $(x_v, y_v)$

x,y 방향의 중심;  $(s_x, s_y)$

$$\begin{aligned} M &= T(-x_w, -y_w) \cdot S(s_x, s_y) \cdot T(x_v, y_v) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_w & -y_w & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_v & y_v & 1 \end{bmatrix} \end{aligned}$$

$$(x' \ y' \ 1) = (x \ y \ 1)M$$

$$\begin{aligned} x' &= (x - x_w)s_x + x_v \\ y' &= -(y - y_w)s_y + y_v \end{aligned}$$

## 6.2.3 Clipping

viewport로 지정한 영역을 벗어나는 것을 잘라버림

Cohen & Sutherland 의 알고리즘

## 6.3 3차원 컴퓨터 그래픽스

### 6.3.1 서론

edge - vector graphics  
surface(shading) - raster graphics

3차원 물체를 투영면에 투영  
CRT 좌표변환(window-viewport)  
보는 방향 (view angle)

### 6.3.2 관측방향에 따른 좌표변환

물체좌표계(object coord.)  
관측좌표계  
    관측벡터(viewing vector)  
        F; 관측자의 눈  
        T(원점); 투영면

$$F = (f_x, f_y, f_z), \quad T = (t_x, t_y, t_z)$$
$$V = (f_x - t_x, f_y - t_y, f_z - t_z) = (v_x, v_y, v_z)$$

step 1) T의 시작점을 물체좌표계의 원점으로 이동

$$T(-t_x, -t_y, -t_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -t_x & -t_y & -t_z & 1 \end{bmatrix}$$

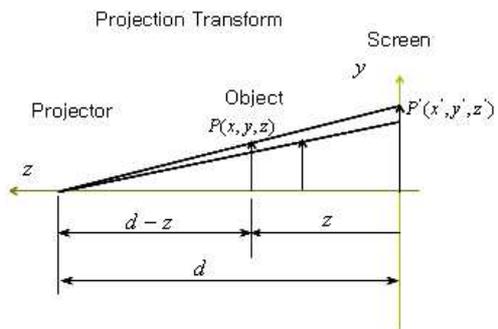
step 2) 관측벡터 V가 yz평면상에 오도록 y축으로  $-\alpha$ 만큼 회전

$$R_y(-\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \alpha = \tan^{-1}\left(\frac{v_x}{v_z}\right)$$

step 3) V가 z축상에 오도록 x축을 기준으로  $+\beta$ 만큼 회전

$$R_x(\beta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \beta = \tan^{-1}\left(\frac{v_y}{\sqrt{v_x^2 + v_z^2}}\right)$$

### 6.3.3 원근투영과 평행투영



d: 관측자의 눈, project 의 렌즈 중심

$$d : y' = d - z : y$$

$$y' = \left( \frac{1}{1 - \frac{z}{d}} \right) y$$

$$x' = \left( \frac{1}{1 - \frac{z}{d}} \right) x$$

$$(x^*, y^*, z^*, 1 - \frac{z}{d}) = (x, y, z, 1) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

원근투영변환(Perspective transform) ; 건축분야  
 평행투영(parallel projection); d=∞ ; 기계분야  
 x\*=x, y\*=y

### 6.3.4 3차원 컴퓨터 그래픽스의 요약

- (1) 관측변환행렬  $M_v$ 를 구함

$$M_v = T(-t_x, -t_y, -t_z) \cdot R_y(-\alpha) \cdot R_x(\beta)$$

평행투영

$$M_v = R_y(-\alpha)R_x(\beta)$$

- (2) 원근투영 변환  $M_p$

평행투영인 경우  $x, y$  값만 선택

- (3) 3차원 물체좌표를 도형좌표로 변환

$$(x^*, y^*, 0, 1) = (x, y, z, 1) \cdot M_v \cdot M_p$$

- (4) 도형좌표를 CRT 좌표(viewport)로 변환

$$x' = (x^* - x_w) \cdot s_x + x_v$$
$$y' = (y^* - y_w) \cdot s_y + y_v$$

- (5) viewport내로 clipping