function Display3D(DATA, T)

% scaling factor

스케일링을 위한 눈행렬의 마지막 요소 대입

s = T(4,4);

% draw the X, Y, Z axis

x, y, z 축의 전치행렬

X = [1,0,0,1]';

Y = [0,1,0,1]';

Z = [0,0,1,1]';

O = [0,0,0,1]';

데이터와 회전행렬을 곱하여 데이터에 갱신

DATA = T\*DATA;

마지막줄 0으로 변경

T0 = T;

T0(1,4) = 0;

T0(2,4) = 0;

T0(3,4) = 0;

% transform

방향 변경

X0 = T0\*X;

Y0 = T0\*Y;

Z0 = T0\*Z;

Origin = T0\*O;

% display is the y-z plane

hold off;

plot([-2,2],[-2,2],'wx');

hold on;

% Project onto the YZ axis

행렬 이동

Tx = s\*[0,1,0,0];

Ty = s\*[0,0,1,0];

출력

plot(Tx\*[Origin, X0], Ty\*[Origin,X0], 'g');

plot(Tx\*[Origin, Y0], Ty\*[Origin,Y0], 'r');

plot(Tx\*[Origin, Z0], Ty\*[Origin,Z0], 'm');

% display the data

plot(Tx\*DATA,Ty\*DATA, 'b')

end

% Translation

X, Y, Z 의 변환 행렬 및 화살표 행렬

X = [0,0,0,0,0,0,0,0]';

Y = [-0.5,0.5,0.5,1,0,-1,-0.5,-0.5]';

Z = [0,0,0.5,0.5,1,0.5,0.5,0]';

ARROW = [X,Y,Z,0\*X+1]';

회전값 Ry, Rz 도출

c = cos(-25\*pi/180);

s = sin(-25\*pi/180);

Ry = [c,0,-s,0;0,1,0,0;s,0,c,0;0,0,0,1];

c = cos(-45\*pi/180);

s = sin(-45\*pi/180);

Rz = [c,-s,0,0;s,c,0,0;0,0,1,0;0,0,0,1]

회전행렬

Tdisp = Ry\*Rz

4x4 눈행렬을 만들어 1~180도의 방향만큼 x축으로 화살표를 회전

T = eye(4,4);

**for** i=1:180

dx = sin(i\*pi/90);

T(1,4) = dx;

Display3D(T\*ARROW, Tdisp);

pause(0.01);

end

4x4 눈행렬을 만들어 1~180도의 방향만큼 y축으로 화살표를 회전

**for** i=1:180

dy = sin(i\*pi/90);

T(2,4) = dy;

Display3D(T\*ARROW, Tdisp);

pause(0.01);

end

4x4 눈행렬을 만들어 1~180도의 방향만큼 z축으로 화살표를 회전

**for** i=1:180

dz = sin(i\*pi/90);

T(3,4) = dz;

Display3D(T\*ARROW, Tdisp);

pause(0.01);

end