Contents


```
function display_cube()


%
% This function displays a 3-D cube.
%
%
%
%
```

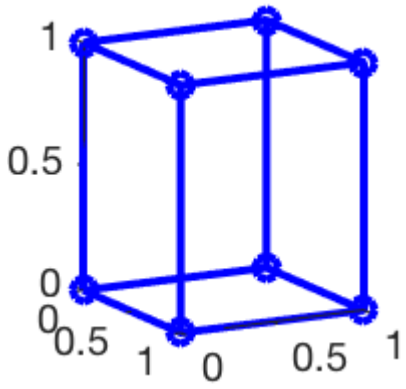
The cube has 8 vertices with 3-D coordinates $\mathbf{p}_i = (x_i, y_i, z_i)^T$, for $i = 1, \ldots, 8$. We store the coordinates into a matrix, i.e.:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_8 \\ y_1 & y_2 & \cdots & y_8 \\ z_1 & z_2 & \cdots & z_8 \end{bmatrix} = [\, \mathbf{p}_1 \quad \mathbf{p}_1 \quad \cdots \quad \mathbf{p}_8 \,]. \tag{1}$$

```
% The shape to be transformed
pts = [ 0.0  0.0  0.0;    % 1
        0.0  0.0  1.0;    % 2
        0.0  1.0  0.0;    % 3
        0.0  1.0  1.0;    % 4
        1.0  0.0  0.0;    % 5
        1.0  0.0  1.0;    % 6
        1.0  1.0  0.0;    % 7
        1.0  1.0  1.0 ]'  % 8


pts =

     0     0     0     0     1     1     1     1
     0     0     1     1     0     0     1     1
     0     1     0     1     0     1     0     1
```

First, we display the original shape in blue. The shape will then be transformed and the result is displayed in red. All shapes will be displayed superimposed on a single plot.

```
% Create a new figure dialog and set the background color to white.
figure;
set(gcf, 'color','w');
set(gcf, 'Position', [0, 0, 100, 100])

% Show original shape in blue
showCube(pts,'b');
view(62,11)
```



The following matrix represents the transformation. In this example, we use the scaling transformation, which is given by:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}. \tag{2}$$

This is the transformation in its original form. We want to convert it to its form in homogeneous coordinates, i.e.:

$$\tilde{S} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

This convertion to homogeneous coordinates allows us to combine linear transformations (i.e., scaling, rotation, shear, reflection) with translations (i.e., spatial shifts) by using matrix multiplications.

```
% Transformation matrix (scaling)
S_tilde = [ 0.5 0.0 0.0 0.0;
            0.0 0.5 0.0 0.0;
            0.0 0.0 0.5 0.0;
            0.0 0.0 0.0 1.0 ];
```

We also convert the coordinates of the shape from cartesian to homogeneous, i.e.:

$$\tilde{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_8 \\ y_1 & y_2 & \cdots & y_8 \\ z_1 & z_2 & \cdots & z_8 \\ 1 & 1 & \cdots & 1 \end{bmatrix} = [\, \tilde{\mathbf{p}}_1 \quad \tilde{\mathbf{p}}_1 \quad \cdots \quad \tilde{\mathbf{p}}_8 \,]. \tag{4}$$

```
% Convert points to homogeneous coordinates
pts_tilde = [ pts; ones(1,size(pts,2)) ]
```

```
pts_tilde =

     0     0     0     0     1     1     1     1
     0     0     1     1     0     0     1     1
     0     1     0     1     0     1     0     1
     1     1     1     1     1     1     1     1
```

To transform the shape, we multiply the transformation matrix by the shape matrix, i.e.:

$$\tilde{X}' = \tilde{S}\tilde{X}. \tag{5}$$

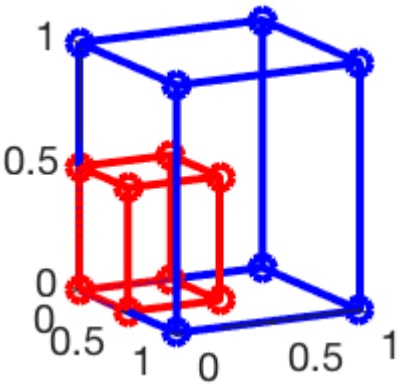
```
% Apply transformation
pts_prime = S_tilde * pts_tilde;

% Show transformed shape in red
hold on;
showCube(pts_prime,'r');

view(62,11)
return


function showCube(x, c )
```

showCube

This function plots the cube shape in 3-D.

Input: x: (x,y,z) coordinates as 3xM matrix c: line color

```
hold on;

% Indices of bottom square
idx1 = [ 1 5 7 3 1 ];
plot3(x(1,idx1),x(2,idx1),x(3,idx1),'Color',c, 'Marker','o','LineWidth',2);

% Indices of top square
idx2 = [ 2 6 8 4 2 ];
plot3(x(1,idx2),x(2,idx2),x(3,idx2),'Color',c, 'Marker','o','LineWidth',2);

% Link the two squares
plot3(x(1,1:2),x(2,1:2),x(3,1:2),'Color',c, 'Marker','o','LineWidth',2);
plot3(x(1,5:6),x(2,5:6),x(3,5:6),'Color',c, 'Marker','o','LineWidth',2);
plot3(x(1,7:8),x(2,7:8),x(3,7:8),'Color',c, 'Marker','o','LineWidth',2);
plot3(x(1,3:4),x(2,3:4),x(3,3:4),'Color',c, 'Marker','o','LineWidth',2);

hold off;

return
```


1
0.5
0
0
0.5
1
0
0.5
1