3-D (Linear) Geometric Transformations

Overview

In these notes, we introduce 3-D linear transformations. Because these transformations are directly extensions of their 2-D counterparts, these notes will just present some basic transformations such as scaling, rotation, and shear.

Three-dimensional linear transformations

As expected, a 3-D linear transformation transforms three-dimensional vectors $\mathbf{x} = (x, y, z)^\mathsf{T}$ onto three-dimensional vectors $\mathbf{x}' = (x', y', z')^\mathsf{T}$. In this case, we can write $f : \mathbb{R}^3 \to \mathbb{R}^3$ such that:

$$\begin{aligned} x' &= f_1(x, y, z) = a_1 x + a_2 y + a_3 z, \\ y' &= f_2(x, y, z) = a_4 x + a_5 y + a_6 z, \\ z' &= f_3(x, y, z) = a_7 x + a_8 y + a_9 z. \end{aligned} \tag{12}$$

This 3-D linear transformation consists of three separate equations, each resulting in one component (or coordinate) of the transformed vector. Equation 12 can be written in matrix form as:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{13}$$

or, in short:

$$\mathbf{x}' = A\mathbf{x}. \tag{14}$$

Here, the transformation is encoded in matrix A, i.e.:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \tag{15}$$

Some important 3-D geometric transformations

The matrix of some important geometric 3-D transformations are:

Scaling:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}. \tag{16}$$

Rotations (about the x-axis, y-axis, and z-axis):

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \tag{17}$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{18}$$

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

By multiplying these matrices, we can obtain general rotations such as:

$$R = R_z\,R_y\,R_x = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{yaw}} \underbrace{\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}}_{\text{pitch}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}}_{\text{roll}}. \tag{20}$$

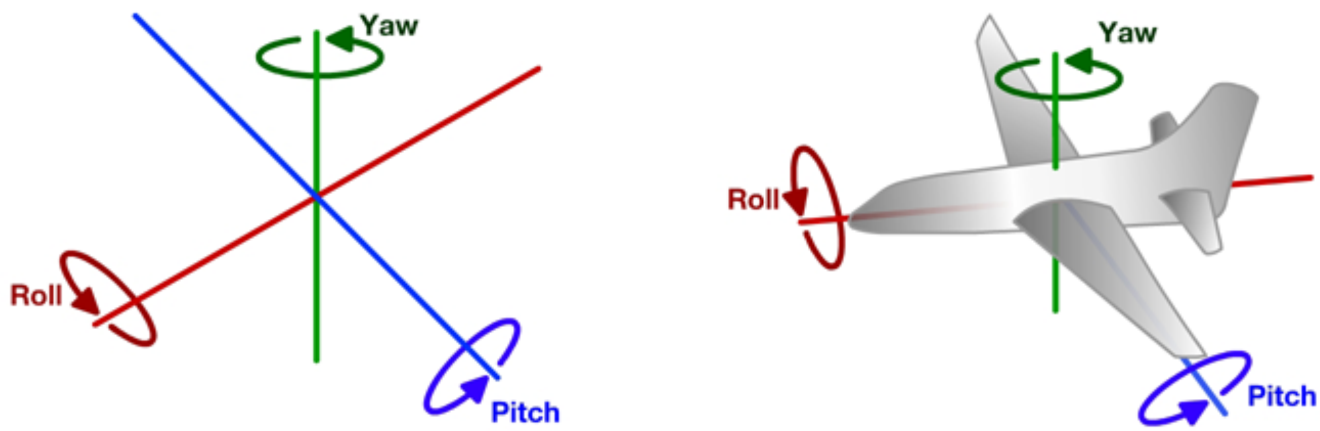


Figure from https://www.touringmachine.com/Articles/aircraft/6/

Example

Transforming a 3-D shape

The following matrix stores the 8 vertices of a rectangular prism:

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{21}$$

We can scale the shape by half as follows:

$$\begin{aligned} X' &= S\,X \\ &= \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}. \end{aligned} \tag{22}$$

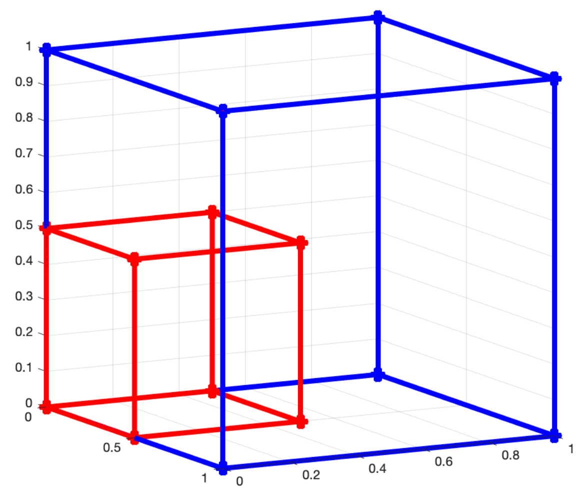
Matlab basic code:

```
% The shape to be transformed  (transposed)
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

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

% Apply transformation
pts_prime = S_tilde * pts
```

Plot:



Whole code (with plotting functions)

```
function display_cube()
%
% This function displays a 3-D cube.
%

% 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


% 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)

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


% Apply transformation
pts_prime = S_tilde * pts;

% 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',4);
```

```
% 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',4);

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


hold off;

return
```