

# VISUALIZING LINEAR ALGEBRA USING THE HTML5 CANVAS: MATRICES AND QUATERNIONS

Paul Bouthellier

Department of Computer Science and Mathematics  
University of Pittsburgh-Titusville  
Titusville, PA 16354  
pbouthe@pitt.edu

The HTML5 canvas is a workspace that can be added to a web page to perform many of the same features as Flash. However, unlike Flash, which is no longer viewable on many devices, any device which can view web pages with a current browser can view canvas animations and interactive content. The canvas also has the advantages of requiring no extra software, no plug-ins, and hence no extra cost to view or create for it.

In order to design for the canvas however, one must understand the concepts of matrix transformations such as: translations, scaling, rotations, skewing, and the composition and inverse of matrix transformations. These are needed to properly place and animate objects on the canvas. A basic understanding of the JavaScript programming language is also required.

In this paper the canvas will be used to interactively illustrate rotations in  $\mathbb{R}^3$  using matrices and quaternions.

Such HTML5 canvas based tutorials may be viewed on any device-computers, tablets, and mobile phones-which can view a web page.

## What is the HTML5 Canvas?

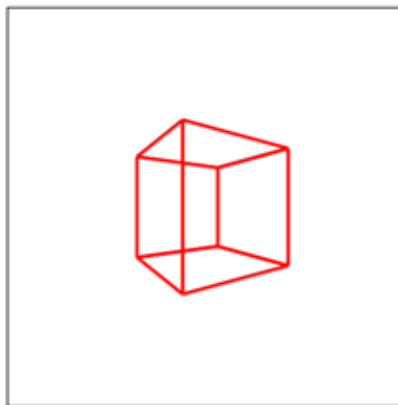


Figure 1-A 3D cube created in the HTML5 canvas

Answer: A rectangle in a web page where the following may be placed:

- text
- lines
- curves
- images
- videos
- sounds

These objects may then be transformed by: rotations, scaling, skewing, and translating. It will be noted that the basic canvas only deals with 2D figures. To deal with 3D objects, as shown in Figure 1, one either needs to download a JavaScript file, such as three.js and learn how to implement it, or to create your own JavaScript code to create 3D models and project them onto a 2D computer screen. (Warning: the learning curve can be a bit steep-but is well worth it.)

Some great things about the canvas:

- Can create interactive animations
- Can create interactive tutorials
- Accessible-Anyone with a modern browser already has it-for free
- Growing Rapidly
- Displays Quickly
- Can alter individual pixels

The key point here is that it is free. The main problem in the past with creating interactive 3D models is that it required proprietary software. Problem solved.

### Reasons to Develop for the Canvas

- Develop once-run (almost) anywhere
- Toolkits available in 2D/3D
- Well-accepted standard
- Demand for interactivity and animations
- Creating tutorials
- The mobile market
- It's Free
- It's Free
- It's Free!!!!!!!!!!!!

The y-Axis is Reverse from What We Are Used To

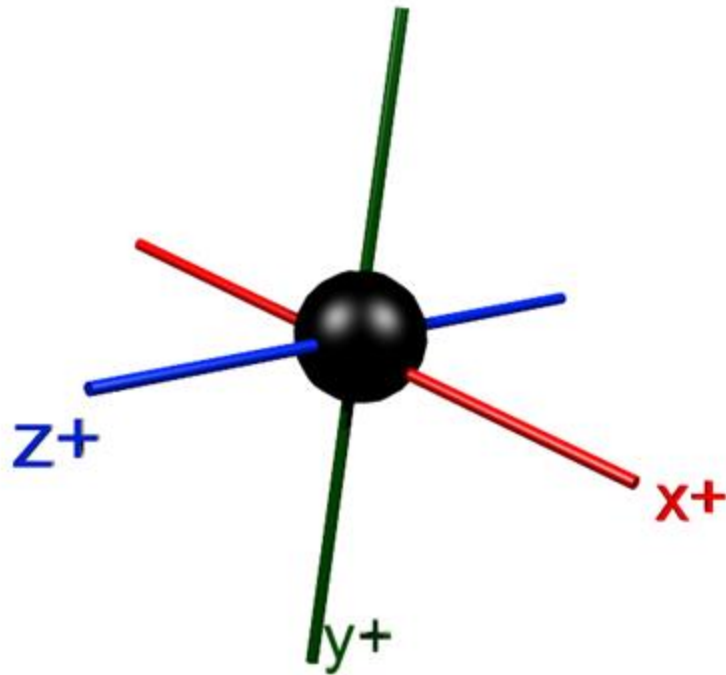


Figure 2-The positive y-axis is down

When writing programs one always has to remember the y-axis is inverted.

### Matrix Rotations About the x-, y-, and z- Axes

An excellent application of the canvas is to illustrate the mathematics of rotations in 3D. We will look at Euler angles and quaternions.

Euler Angles:

The usual rotation matrices about the x, y, and z- axes are as follows:

$$R_x = \begin{bmatrix} \cos(\theta_x) & -\sin(\theta_x) & 0 \\ \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix}$$

$$R_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_z) & \sin(\theta_z) \\ 0 & -\sin(\theta_z) & \cos(\theta_z) \end{bmatrix}$$

Rotation about an Arbitrary Axis  $\langle n_x, n_y, n_z \rangle$  (the normal vector) where

$$c = \cos(\theta) \text{ and } s = \sin(\theta)$$

$$R_n(\theta) = \begin{bmatrix} c + (1-c)n_x^2 & (1-c)n_x n_y - s n_z & (1-c)n_x n_z + s n_y \\ (1-c)n_x n_y + s n_z & c + (1-c)n_y^2 & (1-c)n_y n_z - s n_x \\ (1-c)n_x n_z - s n_y & (1-c)n_y n_z + s n_x & c + (1-c)n_z^2 \end{bmatrix}$$

Proof:

- Breaking point "vector" into parallel and orthogonal components or using
- Exponential Maps  $e^{A\theta}$

Using the above matrices we can write code to allow for the interactive rotation of an object about any axis, and show students the mathematics and the code which makes it work.

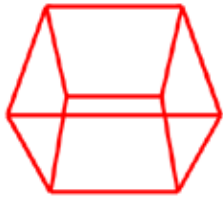


		
Rotation About x-Axis	Rotation About y-Axis	Rotation About z-Axis

Figure 3 Rotations about the x, y, and z-axis in the HTML5 canvas

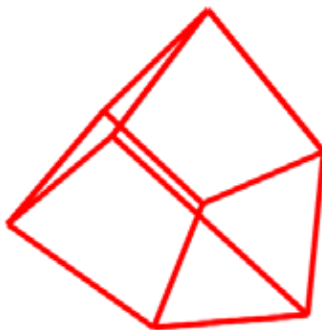


Figure 4: Rotation About the Axis  $\langle 1,1,1 \rangle$

An important fact about Euler angles is that they can create rotations about an arbitrary axis using one of 12 possible combinations as given by the following theorem by Euler:

Theorem [2]:

Any two independent orthonormal coordinates frames can be related by a sequence of not more than three rotations about coordinate axes, where no two successive rotations may be about the same axis.

Euler Angle-Axis Sequences

xyz	xzy	xyx	xzx	yzx	yxz
yzy	yxy	zxy	zyx	zxz	zyz

Using the HTML5 canvas we may illustrate the above theorem to students interactively rather than just writing a sequence of matrices and multiplying them together. The students can see each step of the rotation visually.

Possible Problems with Euler Angles:

- Round-off errors (matrix creep)
- Interpolation between angle-axis orientations
- Gimbal Lock

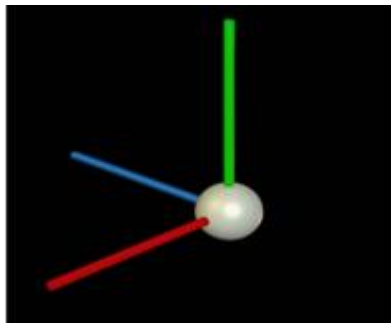
### What is Gimbal Lock?

Depends on who you ask.

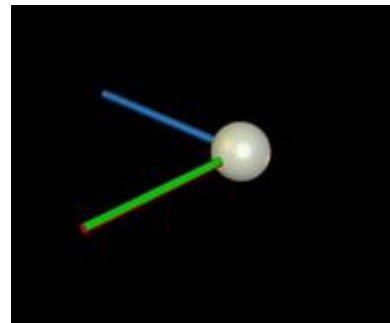
#### Graphic Design Definition:

Gimbal lock occurs when one axis is rotated onto another axis thus causing a loss of a degree of freedom in the rotations.

Why: When the angle of the second rotation is  $\pm 90^\circ$  the third axis of rotation is rotated onto the first axis of rotation.



Blue Axis (left) Rotates Green Axis (up) onto Red Axis (forward)



Gimbal Lock-Loss of a Degree of Freedom in Rotation

Figure 5-Computer Graphics Definition of Gimbal Lock

#### Aerospace Definition:

Gimbal lock occurs when multiple combinations of canonical angles rotations produce the same rotation matrix and one is not able to determine the pitch, heading, and bank uniquely.

Why: When the second angle of rotation is  $\pm 90^\circ$  terms in the rotation matrix reduce to:

$$R_x R_y R_z$$

$$= \begin{bmatrix} \cos(x) & -\sin(x) & 0 \\ \sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(y) & 0 & -\sin(y) \\ 0 & 1 & 0 \\ \sin(y) & 0 & \cos(y) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(z) & \sin(z) \\ 0 & -\sin(z) & \cos(z) \end{bmatrix}$$

$$\text{Where } y = \pm \frac{\pi}{2}$$

$$= \begin{bmatrix} 0 & \sin(z-x) & -\cos(z-x) \\ 0 & \cos(z-x) & \sin(z-x) \\ 1 & 0 & 0 \end{bmatrix}$$

And now we cannot uniquely determine the rotations in the z- and x-axis.

Notes:

- In Gimbal lock nothing is ever "locked".
- In various books and forums there is a great deal of confusion over gimbal lock as people are often using different definitions.

A mathematical technique that allows us a way out of Gimbal lock, if used properly, is that of quaternions.

## A Brief Look at Quaternions

Quaternions were invented in 1843 by Hamilton originally as a way to manipulate vectors.

Define  $q = q_0 + iq_1 + jq_2 + kq_3 = q_0 + \mathbf{q}$  where

$$i^2 = j^2 = k^2 = ijk = -1$$

$$ij = k = -ji$$

$$jk = i = -kj$$

$$ki = j = -ik$$

Quaternion multiplication:

where

$$q = q_0 + iq_1 + jq_2 + kq_3 = q_0 + \mathbf{q}$$

$$p = p_0 + ip_1 + jp_2 + kp_3 = p_0 + \mathbf{p}$$

$$pq = p_0q_0 - \mathbf{q} \bullet \mathbf{p} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}$$

Given pure quaternions:

$$q = iq_1 + jq_2 + kq_3 = \mathbf{q}$$

$$p = ip_1 + jp_2 + kp_3 = \mathbf{p}$$

$$pq = -\mathbf{p} \bullet \mathbf{q} + \mathbf{p} \times \mathbf{q}$$

**Rotations:**

Consider a unit quaternion

$$\cos\left(\frac{\theta}{2}\right) + u \sin\left(\frac{\theta}{2}\right)$$

Where  $u$  in a unit vector in  $R^3$

Defining

$$q^{-1} = q_0 - iq_1 - jq_2 - kq_3 = q_0 - \mathbf{q}$$

and where  $v \in R^3$  is represented as a pure quaternion  $(0+v)$

Then

$$qvq^{-1}$$

is a right-handed rotation of  $v$  by an angle of  $\theta$  about the vector  $\mathbf{q}$ .



As quaternions simply rotate a point about a single axis we cannot get Gimbal lock.

A note about illustrating rotation matrices and quaternions using the HTML5 canvas: If one is trying to optimize the code and simplifying the mathematics as much as possible, the equations one get from rotations matrices and quaternions are of course the same.

To illustrate the original equations for rotations and quaternions-and show we get the same results-one must put up with some inefficiencies in the program.

A final topic that one can illustrate with the HTML5 canvas is that of interpolation between coordinate frames using various forms of matrix interpolation and quaternions methods such as SLERP (Spherical Linear Interpolation).

References:

- [1] Andrew Hanson, "*Visualizing Quaternions*" San Francisco, CA: Morgan Kaufmann, 2006.
- [2] Jack B. Kuipers, "*Quaternions and Rotation Sequences*" Princeton, NJ: Princeton University Press, 2002.