3 × 3 transformation ma- trix applied to a 3-element homogeneous column vector representing a 2-D point will generate a new 3-element homogeneous column vector represent- ing the transformed 2-D point

Scale

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+c \\ dx+ey+f \\ 1 \end{bmatrix}.$$

$$\begin{bmatrix} s & & \\ & s & \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} sx \\ sy \\ 1 \end{bmatrix}.$$

(To avoid visual clutter, we omit matrix elements that are zero.)

$$\begin{bmatrix} h & & \\ & v & \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} hx \\ vy \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} 1 & & a \\ & 1 & b \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ 1 \end{bmatrix}$$

Relative Scale

Using vector algebra, this operation would be

$$s((x,y)+(-a,-b))+(a,b) = (sx+(1-s)a, sy+(1-s)b). \qquad (5.6)$$

Using transformation matrices, this becomes

$$\begin{aligned} T_{a,b}\ S_s\ T_{-a,-b} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= \begin{bmatrix} 1 & & a \\ & 1 & b \\ & & 1 \end{bmatrix} \begin{bmatrix} s & & \\ & s & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & -a \\ & 1 & -b \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} s & & (1-s)a \\ & s & (1-s)b \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \end{aligned} \qquad (5.7)$$

Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & \\ \sin\theta & \cos\theta & \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Polar Coordinate (r, ϕ)

$$\begin{aligned} r &= ||(x,y)|| = \sqrt{x^2+y^2}, \\ \phi &= \arctan\frac{y}{x} \\ x &= r\cos\phi \\ y &= r\sin\phi. \end{aligned}$$

$$\begin{aligned} x' &= r\cos(\phi+\theta), \\ y' &= r\sin(\phi+\theta). \end{aligned}$$

$$\begin{aligned} \cos(\phi+\theta) &= \cos\phi\cos\theta - \sin\phi\sin\theta, \\ \sin(\phi+\theta) &= \sin\phi\cos\theta + \cos\phi\sin\theta, \\ x' &= r(\cos\phi\cos\theta - \sin\phi\sin\theta) \\ &= x\cos\theta - y\sin\theta, \\ y' &= r(\sin\phi\cos\theta + \cos\phi\sin\theta), \\ &= y\cos\theta + x\sin\theta. \end{aligned}$$

Inverse Transformation

Identity matrix

$$I = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

Opposite direction

$$T^{-1} = \begin{bmatrix} 1 & & a \\ & 1 & b \\ & & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & & -a \\ & 1 & -b \\ & & 1 \end{bmatrix}$$

R is a rotation matrix with angle

$$\begin{aligned} R^{-1} &= \begin{bmatrix} \cos\theta & -\sin\theta & \\ \sin\theta & \cos\theta & \\ & & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \cos-\theta & -\sin-\theta & \\ \sin-\theta & \cos-\theta & \\ & & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & \sin\theta & \\ -\sin\theta & \cos\theta & \\ & & 1 \end{bmatrix} \\ &= R^T, \end{aligned}$$

Scale matrix S might not be invertible, if it is then

$$S^{-1} = \begin{bmatrix} h & & \\ & v & \\ & & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{h} & & \\ & \frac{1}{v} & \\ & & 1 \end{bmatrix}.$$

If it is not, then it is a projection

If an arbitrary 3 × 3 transformation matrix M is invertible, then its inverse is

$$\begin{aligned} M^{-1} &= \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix}^{-1} \\ &= \frac{1}{ad-bc} \begin{bmatrix} d & -b & bf-de \\ -c & a & ce-af \\ 0 & 0 & ad-bc \end{bmatrix} \\ &= \frac{1}{\det M}\,\mathrm{adj}M. \end{aligned}$$

CanvasToScreenTransformation

$$C2S = \begin{bmatrix} \frac{H-1}{r-l} & & \\ & \frac{V-1}{t-b} & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & -l \\ & 1 & -b \\ & & 1 \end{bmatrix} = \begin{bmatrix} \frac{H-1}{r-l} & & -l\frac{H-1}{r-l} \\ & \frac{V-1}{t-b} & -b\frac{V-1}{t-b} \\ & & 1 \end{bmatrix}$$

$$C2S_{-1,1} = \begin{bmatrix} \frac{H-1}{2} & & \frac{H-1}{2} \\ & \frac{V-1}{2} & \frac{V-1}{2} \\ & & 1 \end{bmatrix}.$$

Plotting fields

$$S2C = C2S^{-1} = (S \times T)^{-1} = T^{-1} \times S^{-1}.$$

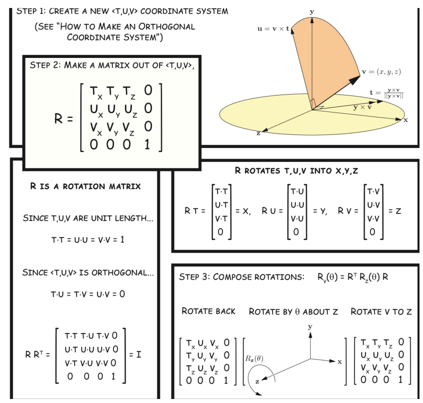
3D transformational geometry

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+cz+d \\ ex+fy+gz+h \\ ix+jy+kz+l \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} s & & & \\ & s & & \\ & & s & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} sx \\ sy \\ sz \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & & & a \\ & 1 & & b \\ & & 1 & c \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ z+c \\ 1 \end{bmatrix}$$



$$\mathbf{a}\cdot\mathbf{b} = (a_x, a_y, a_z)\cdot(b_x, b_y, b_z) = a_x b_x + a_y b_y + a_z b_z.$$

$$\mathbf{a}\cdot\mathbf{b} = A^T B = [a_x\ a_y\ a_z\ 0] \begin{bmatrix} b_x \\ b_y \\ b_z \\ 0 \end{bmatrix}$$

$$\mathbf{a}\cdot\mathbf{b} = ||\mathbf{a}||\ ||\mathbf{b}||\cos\theta$$

$$\mathbf{a_b} = \frac{\mathbf{a}\cdot\mathbf{b}}{\mathbf{b}\cdot\mathbf{b}}\mathbf{b}.$$

CROSS PRODUCT

$$(a_x, a_y, a_z) \times (b_x, b_y, b_z) = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x).$$

$$(a_x, a_y, a_z) \times (b_x, b_y, b_z) = \begin{vmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$

Given the vector a = (ax,ay,az), the skew-symmetric matrix

$$\mathsf{X}_{\mathbf{a}} = \begin{bmatrix} & -a_z & a_y \\ a_z & & -a_x \\ -a_y & a_x & \end{bmatrix}$$

$$\begin{bmatrix} & -a_z & a_y & \\ a_z & & -a_x & \\ -a_y & a_x & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \\ 0 \end{bmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \\ 0 \end{bmatrix}$$

$$||\mathbf{a} \times \mathbf{b}|| = ||a||\ ||b|| \sin\theta.$$

ROTATION AROUND AXIS

$$R(\theta, \mathbf{z}) = \begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

ABOUT X

$$R(\theta, \mathbf{x}) = \begin{bmatrix} 1 & & & \\ & \cos\theta & -\sin\theta & \\ & \sin\theta & \cos\theta & \\ & & & 1 \end{bmatrix}.$$

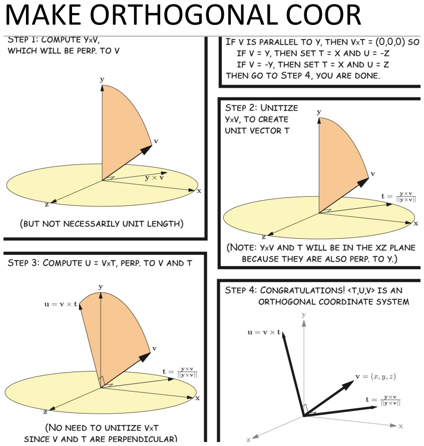
ABOUT Y

$$R(\theta, \mathbf{y}) = \begin{bmatrix} \cos\theta & & \sin\theta & \\ & 1 & & \\ -\sin\theta & & \cos\theta & \\ & & & 1 \end{bmatrix}$$

Cayley's Formula

$$\mathsf{X}_{\mathbf{v}} = \begin{bmatrix} & -v_z & v_y \\ v_z & & -v_x \\ -v_y & v_x & \end{bmatrix}$$

MAKE ORTHOGONAL COOR



ROTATION SUIYI AXIS

$$R(\theta, \mathbf{v}) = R^{-1}(\phi) R(\theta, \mathbf{z}) R(\phi)$$

Given a unit vector $\mathbf{v}$, we want to find $R(\phi)$, a rotation matrix such that $R(\phi)\left[\begin{smallmatrix}\mathbf{v}\\0\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{z}\\0\end{smallmatrix}\right]$. We will do this by first rotating $\mathbf{v}$ into the xz-plane, then rotating that into the z-axis.

We first rotate $\mathbf{v}$ into the xz-plane, by calculating an appropriate rotation about the x-axis, $R(\phi_x, \mathbf{x})$. We find this angle ϕ_x by projecting $\mathbf{v} = (x, y, z)$ onto the yz-plane as $(0, y, z)$. Note that the same rotation $R(\phi_x, \mathbf{x})$ that rotates $\mathbf{v}$ into the xz-plane would also rotate $(0, y, z)$ into the z axis. In the yz-plane, we form the right triangle with vertices $(0, 0, 0)$, $(0, y, z)$ and $(0, 0, z)$, and the angle its hypotenuse makes with the origin is precisely ϕ_x. The length of the adjacent edge is z, of the opposite edge is y and of the hypotenuse is $d = \sqrt{y^2 + z^2}$. We find ϕ_x with the trigonometry

$$\cos\phi_x = z/d, \tag{6.53}$$
$$\sin\phi_x = y/d, \tag{6.54}$$

$$R(\phi_x, \mathbf{x}) = \begin{bmatrix} 1 & & & \\ & z/d & -y/d & \\ & y/d & z/d & \\ & & & 1 \end{bmatrix}$$

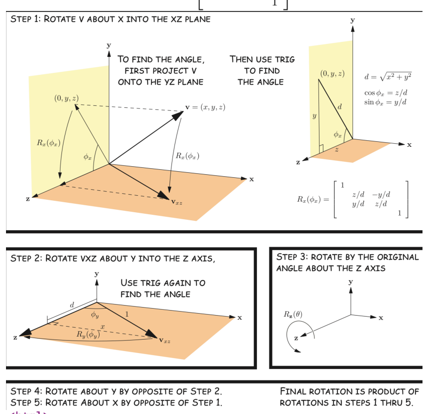
$$\begin{bmatrix} \mathbf{v}_{xz} \\ 0 \end{bmatrix} = R(\phi_x, \mathbf{x}) \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}.$$

Next, we find a rotation about the y-axis $R(\phi_y, \mathbf{y})$ that rotates $\mathbf{v}_{xz}$ from the xz-plane into the z-axis. Note that $\mathbf{v}_{xz}$ is a unit vector because it is a rotated version of $\mathbf{v}$, so when it is rotated into the z-axis, it will be simply $(0, 0, 1)$. We again create a right triangle, now in the xz-plane with vertices $(0, 0, 0)$, $\mathbf{v}_{xz}$ and $(0, 0, d)$. The point $(0, 0, d)$ is the projection of $\mathbf{v}_{xz}$ onto the z-axis, and is the result of rotating the previous hypotenuse by ϕ_x about the x-axis. We use trigonometry again to find the angle ϕ_y of this triangle at the origin, given a hypotenuse of length one ($||\mathbf{v}_{xy}|| = ||\mathbf{v}||$), an adjacent length of d and an opposite length x (since rotation about the x-axis does not change the x coordinate). A positive right-handed rotation would rotate the positive z axis toward the positive x axis, so the angle ϕ_y should be negative when used to specify a rotation[1]. Hence

$$\cos\phi_y = d, \tag{6.56}$$
$$\sin\phi_y = -x, \tag{6.57}$$

and

$$R(\phi_y, \mathbf{y}) = \begin{bmatrix} d & & -x & \\ & 1 & & \\ x & & d & \\ & & & 1 \end{bmatrix}. \tag{6.58}$$




```
<html>
<body>
    <canvas id="myCanvas" width="400" height="200"></canvas>
  <br/>
  <input id="slider" type="range" min="0" max="100" />
</body>
<script> "use strict"
  function setup() {
    var y = 50;
    var canvas = document.getElementById('myCanvas');
    var slider = document.getElementById('slider');
    slider.value = 20;
    function draw() {
      var context = canvas.getContext('2d');
      //context.clearRect(0, 0, canvas.width, canvas.height);
      canvas.width = canvas.width;
      // use the slider to get the position
      var x = slider.value;
      // this actually draws a square
      context.beginPath();
      context.beginPath();
      context.rect(x,y,50,50);
      context.fill();
      y = (y + 2) % 100;
      window.requestAnimationFrame(draw);
    };
    // we don't need an event listener - we'll update all the time
    // slider.addEventListener("input",draw);
    // we don't need to draw - since requestanimationframe does that
    // draw();
    window.requestAnimationFrame(draw);
    };
  window.onload = setup;
</script>
<head>
  <meta name="description" content="E4 - Slider Animation">
  <meta charset="utf-8">
  <title>Slider Animation Example</title>
</head>
</html>
```

Specifically, the changes have opened up two specific aspects of the graphics hardware pipeline. Programmers now have the ability to modify how the hardware processes vertices and shades pixels by writing *vertex shaders* and *fragment shaders* (also sometimes referred to as *vertex programs* or *fragment programs*). Vertex shaders are programs that perform the vertex and normal transformations, texture coordinate generation, and per-vertex lighting computations normally computed in the geometry processing stage. Fragment shaders are programs that perform the computations in the pixel processing stage of the graphics pipeline and determine exactly how each pixel is shaded, how textures are applied, and if a pixel should be drawn or not. These small shader programs are sent to the graphics hardware from the user program (see Figure 18.5), but they are executed on the graphics hardware. What this programmability means for