Computer Graphics Exercises - EDA221, HT1 2013

Jacob Munkberg

Abstract

Here is a short list of exercises that test transforms, parametric surfaces and camera setup. These topics are essential for understanding computer graphics, and a good way of get an intuition for these concepts is to work through a set of exercises.

1 Camera Setup

A camera is placed at the position $E = (1, 2, 3, 1)$ with up vector $(0, 1, 0)$, and is looking at the point $C = (0, 0, 0, 1)$.

a) Derive the view matrix

b) Derive the view matrix if $E = (1, 1, 1, 1)$

c) Given the view matrix in **b)**, what is the *camera space* position of the *world space* point $P = (5, 4, 3, 1)$?

d) What is an affine transform? Give one example of a transform that is not affine.

2 Rotation

Describe the matrix that rotates 30 degrees around the y axis.

a) When the center of rotation is the origin.

b) When the center of rotation is (3,0,6).

c) Are the matrices in **a)** and/or **b)** orthonormal? Motivate.

3 Interpolation

Smoothstep is cubic interpolation between 0,1 on the interval $x \in [a, b]$, with the constraint that the derivative is zero at $x = a$ and $x = b$.

a) Derive the cubic interpolant.

4 Transform from Image

a) Describe the transform needed to transform the triangle from A to B in Figure 1, either in RenderChimp syntax or as a (set of) matrices.

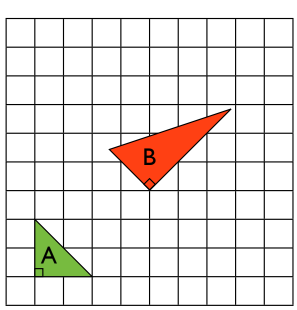

Figure 1: *Transform from A to B*

5 Rotation around a Vector

a) Derive the formula for rotating the vector $\mathbf{v}$ around a vector $\mathbf{u}$.

b) If you rotate the vector $\mathbf{v} = (1, 0, 3)$ 10 degrees around the vector $\mathbf{u} = \frac{1}{\sqrt{3}}(1, 1, 1)$, what is the new vector $\mathbf{v}_{rot}$?

6 Projection

A camera with up-vector $(0, 1, 0)$, placed at $E = \frac{1}{\sqrt{2}}(0, 10, 10)$, and that is looking at the origin, has a view matrix:

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\pi/4) & -\sin(\pi/4) & 0 \\ 0 & \sin(\pi/4) & \cos(\pi/4) & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The projection matrix is given by:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3 & -2 \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

a) Given a camera space coordinate $P_{cam} = (2, 1, -1, 1)$, what is the corresponding clip space coordinate?

b) Given a camera space coordinate $P_{cam} = (6, 4, -2, 1)$, what is the corresponding NDC coordinate?

c) What are the NDC coordinates of the world space triangle with vertices $P^0 = (0, 0, 0, 1)$, $P^1 = (-1, 1, 0, 1)$ and $P^2 = (1, 1, 0, 1)$?

d) The triangle in **c)** has a 90 degree angle at P^0 in world space. Prove this. Is the angle at P^0 in NDC still 90 degrees (i.e., for the 2D projection of the triangle at the image plane)? Motivate your answer.

7 Parametric Surfaces

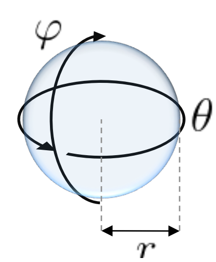

Figure 2: *A parametric representation of a sphere.*

As shown in Figure 2, a sphere with radius ($r = 1$) is given on parametric form

$$s(\theta, \varphi) = \begin{bmatrix} \sin\theta \sin\varphi \\ -\cos\varphi \\ \cos\theta \sin\varphi \end{bmatrix}, \tag{1}$$

where $\theta \in [0, 2\pi[$ and $\varphi \in [0, \pi]$. We pick a tangent space such that the tangent, $\mathbf{t}$, is aligned with $\frac{\partial \mathbf{s}}{\partial \theta}$ and the binormal, $\mathbf{b}$, is aligned with $\frac{\partial \mathbf{s}}{\partial \varphi}$.

a) Define the position and tangent space at the parametric value $(\theta, \varphi) = (\frac{\pi}{2}, \frac{\pi}{2})$

b) Define the position and tangent space at the parametric value $(\theta, \varphi) = (\frac{\pi}{4}, \frac{3\pi}{4})$

8 Bump Mapping

For the point derived in Exercise 7b, we perform a texture lookup into a bump map texture.

a) If the texture lookup returns the color $(100, 10, 175)$, what is the perturbed normal in object space?

b) If the world matrix is given by the non-uniform scaling,

$$\mathbf{M} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

what is the perturbed normal vector in *world space*?

9 Basis Vectors

Given a *orthonormal* basis defined by the basis vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$.

a) Define the 3×3 matrix that rotates $\mathbf{x} = (1, 0, 0)$ to align with $\mathbf{a}$, $\mathbf{y} = (0, 1, 0)$ to align with $\mathbf{b}$, and $\mathbf{z} = (0, 0, 1)$ with $\mathbf{c}$.

b) Both the basis $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ are orthonormal. Derive the inverse of the matrix from question **a)**. What is this matrix useful for?

10 The View Matrix

A view matrix is defined by

$$\mathbf{M} = \begin{bmatrix} 0.36 & 0 & -0.93 & 0 \\ -0.44 & 0.88 & -0.17 & 0 \\ 0.82 & 0.47 & 0.32 & -10.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and its inverse is given by:

$$\mathbf{M}^{-1} = \begin{bmatrix} 0.36 & -0.44 & 0.82 & 8.23 \\ 0 & 0.88 & 0.47 & 4.71 \\ -0.93 & -0.17 & 0.32 & 3.18 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

a) What is the camera position in world space?

11 Concatenation of Transforms

Let the matrix $\mathbf{A}$ represent a translation along the vector $(1, 2, 3)$, $\mathbf{B}$ a rotation with 45 degrees around the x axis and $\mathbf{C}$ a scaling with $(4, 6, 1)$ in (x, y, z).

a) Write the three matrices on 4x4 matrix form.

b) Given the point P = (3,4,5,1), what is $P_{ABC} = \mathbf{ABC}P$?

c) Given the point P = (3,4,5,1), what is $P_{CAB} = \mathbf{CAB}P$?

d) Given the point P_{CAB}, defined as in **c)**, describe how to get back to the point P using *three* steps, where a transform is applied in each step.

Solutions

Disclaimer This is the first version of these exercises and the solutions may contain errors or typo(s). if you find something that seems wrong, please email me at `jacob@cs.lth.se`, so I can update the document.

1. Camera Setup

a) $E = (1, 2, 3, 1)$ gives

$$\mathbf{V} \approx \begin{bmatrix} 0.95 & 0 & -0.32 & 0 \\ -0.17 & 0.85 & -0.51 & 0 \\ 0.27 & 0.53 & 0.80 & -3.74 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

See derivation of the `LookAt` function in Lecture 5. p 5-12.

b) $E = (1, 1, 1, 1)$ gives

$$\mathbf{V} \approx \begin{bmatrix} 0.71 & 0 & -0.71 & 0 \\ -0.41 & 0.82 & -0.41 & 0 \\ 0.58 & 0.58 & 0.58 & -1.73 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

c) $P_{cam} = (1.41, 0, 5.20, 1)$.

d) See lecture & book p. 182-185.

2. Rotation

a)

$$\mathbf{R}_y(\theta = 30 \text{ deg}) = \begin{bmatrix} \cos\frac{\pi}{6} & 0 & \sin\frac{\pi}{6} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\frac{\pi}{6} & 0 & \cos\frac{\pi}{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b)

$$\mathbf{T}((3,0,6))\mathbf{R}_y\mathbf{T}(-(3,0,6)) \approx \begin{bmatrix} 0.87 & 0 & 0.5 & -2.60 \\ 0 & 1 & 0 & 0 \\ -0.5 & 0 & 0.87 & 2.30 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

See lecture 2 for details.

c) The matrix in **a)** is a pure rotation, and is orhonormal, $\mathbf{M}^{-1} = \mathbf{M}^T$. The matrix in **b)** is a concatenation of two translations and one rotation and is not orthonormal ($\mathbf{M}^{-1} \neq \mathbf{M}^T$).

3. Interpolation

a) Set $t = \frac{x-a}{b-a}$, and denote the cubic interpolant $f(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$. The values and derivatives at $t = 0$ and $t = 1$ give us the four equations:

$$\begin{aligned} f(t=0) = 0 \quad &\Rightarrow \quad c_0 = 0 \\ f(t=1) = 1 \quad &\Rightarrow \quad c_0 + c_1 + c_2 + c_3 = 1 \\ \\ f'(t=0) = 0 \quad &\Rightarrow \quad c_1 = 0 \\ f'(t=1) = 0 \quad &\Rightarrow \quad c_1 + 2c_2 + 3c_3 = 0 \end{aligned}$$

Now use these four equations to solve for the four coefficients c_i. In general, this is a system of equations $\mathbf{Ax} = \mathbf{b}$ that can be solved using a matrix inverse, i.e., $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, but in this case, both c_0 and c_1 are zero, we are left with the two equations $c_2 + c_3 = 1$ and $2c_2 + 3c_3 = 0$. Solving for c_2 and c_3, we get $c_2 = 3$ and $c_3 = -2$, and the cubic interpolant is then:

$$f(t) = 3t^2 - 2t^3, \quad (5)$$

where $t = \frac{x-a}{b-a}$.

4. Transform from Image

a) One possible solution is:

Assume the origin is placed in the lower left corner of triangle A, and that Figure 1 shows the xy-plane. First, scale the triangle at A with two in x. Then, rotate the triangle 45 degrees around the origin (around the z-axis, i.e., using the $\mathbf{R}_z$ rotation matrix). Finally, transform the lower left corner to point (4,3,0). The matrix applied to the triangle at A is then:

$$\mathbf{T}(4,3,0)\mathbf{R}_z(45)\mathbf{S}(2,1,1) \quad (6)$$

The full matrix is given by:

$$\mathbf{T}(4,3,0)\mathbf{R}_z(45)\mathbf{S}(2,1,1) = \begin{bmatrix} \sqrt{2} & -\frac{1}{\sqrt{2}} & 0 & 4 \\ \sqrt{2} & \frac{1}{\sqrt{2}} & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In RenderChimp, each node has a $\mathbf{TRS}$ transform matrix, where the matrices are applied in that order, so for this example, given that triangle A is defined with the three vertices $P_0 = (0,0,0)$, $P_1 = (2,0,0)$ and $P_2 = (0,2,0)$, we can simply write

```
tri->setScale(2,1,1);
tri->setRotateZ(M_PI/4.0f);
tri->setTranslate( 4, 3, 0);
```

Note that if we want to multiply the matrices together in *another* order, e.g., $\mathbf{SRT}$, we must create new nodes in RenderChimp to handle this case.

5. Rotation around a Vector

a) See lecture slides (Lecture 2).

b) $\mathbf{v}_{rot} \approx (1.31, -0.18, 2.87)$

6. Projection

a)

$$P_{clip} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3 & -2 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

b) First, compute the clip space coordinate as above. This gives us $P_{\text{clip}} = (6, 4, 4, 2)$ Normalized device coordinates are obtained by performing the perspective divide on the clip space coordinate. Given a clip space coordinate P, the NDC coordinate P_{NDC} is:

$$\frac{P}{P_w} = (\frac{P_x}{P_w}, \frac{P_y}{P_w}, \frac{P_z}{P_w}, 1)$$

In our case $P_{\text{clip}} = (6, 4, 4, 2)$ and

$$P_{NDC} = (3, 2, 2, 1)$$

c) We first apply view matrix to transform from world space to camera space, then the projection matrix to go from camera space to clip space, (or more compact: directly apply the **ViewProjection** matrix $P_{\text{clip}} = [\mathbf{Proj}][\mathbf{View}]P$) on the world space triangle vertices to obtain the clip space positions:

$$\begin{aligned} P^0_{\text{clip}} &= (0, 0, 28, 10) \\ P^1_{\text{clip}} &\approx (-1, 0.71, 25.88, 9.29) \\ P^2_{\text{clip}} &\approx (1, 0.71, 25.88, 9.29) \end{aligned}$$

The NDC positions are obtained by dividing each position with its w component:

$$\begin{aligned} P^0_{\text{NDC}} &= (0, 0, 2.8) \\ P^1_{\text{NDC}} &\approx (-0.11, 0.076, 2.78) \\ P^2_{\text{NDC}} &\approx (0.11, 0.076, 2.78) \end{aligned}$$

d) In world space, the angle at P^0 is obtained by forming the two edge vectors $\mathbf{e}_1 = P^1 - P^0 = (-1, 1, 0)$, and $\mathbf{e}_2 = P^2 - P^0 = (1, 1, 0)$. Now: $\text{dot}(\mathbf{e}_1, \mathbf{e}_2) = 0$, which implies that the angle between the two vectors are 90 degrees.

In NDC, again form the edge vectors

$$\begin{aligned} \mathbf{e}_1 &= P^1_{\text{NDC}} - P^0_{\text{NDC}} \approx (-0.11, 0.076, -0.015) \\ \mathbf{e}_2 &= P^2_{\text{NDC}} - P^0_{\text{NDC}} \approx (0.11, 0.076, -0.015) \end{aligned}$$

Now note that $\text{dot}(\mathbf{e}_1, \mathbf{e}_2) \neq 0$, so in NDC, i.e., *after* projection, there is no longer a 90 degree angle at P_0. Alternatively, we could directly look at the NDC xy-coordinates, and check the angle between the edges $P1P0$ and $P2P0$ of the 2D triangle. Again, we see that this angle is not 90 degrees. Projective transforms do not preserve angles, nor parallel lines.

7. Parametric Surfaces

a)

$$\mathbf{t} = \frac{\partial \mathbf{s}}{\partial \theta} = \begin{bmatrix} \cos\theta \sin\varphi \\ 0 \\ -\sin\theta\sin\varphi \end{bmatrix} = [\text{normalize}] = \begin{bmatrix} \cos\theta \\ 0 \\ -\sin\theta \end{bmatrix}$$

$$\mathbf{b} = \frac{\partial \mathbf{s}}{\partial \varphi} = \begin{bmatrix} \sin\theta\cos\varphi \\ \sin\varphi \\ \cos\theta\cos\varphi \end{bmatrix}$$

$$\mathbf{n} = \frac{\partial \mathbf{s}}{\partial \theta} \times \frac{\partial \mathbf{s}}{\partial \varphi} = \begin{bmatrix} \sin\theta\sin\varphi \\ -\cos\varphi \\ \cos\theta\sin\varphi \end{bmatrix}.$$

b) $(\theta, \varphi) = (\frac{\pi}{2}, \frac{\pi}{2})$ gives $P = \mathbf{s}(\frac{\pi}{2}, \frac{\pi}{2}) = (1, 0, 0)$ and

$$\begin{aligned} \mathbf{t}(\frac{\pi}{2}, \frac{\pi}{2}) &= (0, 0, -1) \\ \mathbf{b}(\frac{\pi}{2}, \frac{\pi}{2}) &= (0, 1, 0) \\ \mathbf{n}(\frac{\pi}{2}, \frac{\pi}{2}) &= (1, 0, 0) \end{aligned}$$

c) $(\theta, \varphi) = (\frac{\pi}{4}, \frac{3\pi}{4})$ gives $P = \mathbf{s}(\frac{\pi}{4}, \frac{3\pi}{4}) = (\frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2})$ and

$$\begin{aligned} \mathbf{t}(\frac{\pi}{4}, \frac{3\pi}{4}) &= (\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}) \\ \mathbf{b}(\frac{\pi}{4}, \frac{3\pi}{4}) &= (-\frac{1}{2}, \frac{1}{\sqrt{2}}, -\frac{1}{2}) \\ \mathbf{n}(\frac{\pi}{4}, \frac{3\pi}{4}) &= (\frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2}) \end{aligned}$$

Note that for a unit sphere, the normal and position have the same xyz coordinates for all values of (θ, φ).

8. Bump Mapping

a) Map $[0, 255] \rightarrow [-1, 1]$. The color $\mathbf{c} = (100, 10, 175)$ then maps to :

$$(\alpha, \beta, \gamma) = \frac{2\mathbf{c}}{255} - (1, 1, 1) \approx (-0.22, -0.92, 0.37)$$

The object space normal is then:

$$\mathbf{n}_o = \alpha\mathbf{t} + \beta\mathbf{b} + \gamma\mathbf{n} \approx (0.49, -0.39, 0.80).$$

Finally, normalize $\mathbf{n}_o$: $\hat{\mathbf{n}}_o = (0.49, -0.38, 0.79)$.

b) If $\mathbf{M}$ is the world matrix, i.e., the matrix that transforms points *from* object space *to* world space, then the normal vectors should be transformed with the matrix $\mathbf{M}^{-T}$ (the inverse transpose of $\mathbf{M}$):

$$\mathbf{M}^{-T} = \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{1}{10} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

The world space normal is then

$$\mathbf{n}_w = \mathbf{M}^{-T}[\hat{\mathbf{n}}_o, 0]^T$$

$$\mathbf{n}_w = \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{1}{10} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_{o_x} \\ n_{o_y} \\ n_{o_z} \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.16 \\ -0.038 \\ 0.39 \\ 0 \end{bmatrix}$$

Finally, normalize $\mathbf{n}_w$: $\hat{\mathbf{n}}_w \approx (0.38, -0.089, 0.92)$.

9. Basis Vectors

a)

$$\mathbf{M} = \begin{bmatrix} | & | & | \\ \mathbf{a} & \mathbf{b} & \mathbf{c} \\ | & | & | \end{bmatrix}.$$

See Lecture 3, slides 5-9.

b) The matrix $\mathbf{M}$ is orthogonal, thus:

$$\mathbf{M}^{-1} = \mathbf{M}^T = \begin{bmatrix} - & \mathbf{a} & - \\ - & \mathbf{b} & - \\ - & \mathbf{c} & - \end{bmatrix}.$$

This matrix rotates $\mathbf{a}$ to align with $\mathbf{x}$, $\mathbf{b}$ to align with $\mathbf{y}$, and $\mathbf{c}$ with $\mathbf{z}$. For example $\mathbf{M}^{-1}\mathbf{b} = (0, 1, 0)$.

10. The View Matrix

a) The camera position in camera space is at the origin. Furthermore, the view inverse matrix transforms a point from camera space to world space. The camera position in world space is then $C_{\text{world}} = \mathbf{M}^{-1}\mathbf{o}$:

$$\begin{bmatrix} 0.36 & -0.44 & 0.82 & 8.23 \\ 0 & 0.88 & 0.47 & 4.71 \\ -0.93 & -0.17 & 0.32 & 3.18 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 8.23 \\ 4.71 \\ 3.18 \\ 1 \end{bmatrix}$$

or simply the fourth column of the view inverse matrix.

11. Concatenation of Transforms

a)

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

.

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} & 0 \\ 0 & \sin\frac{\pi}{4} & \cos\frac{\pi}{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

.

$$\mathbf{C} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

.

b) $P_{ABC} \approx (13.0, 15.44, 23.51, 1)$

c) $P_{CAB} \approx (16, 7.76, 9.36, 1)$

d)

$$\begin{aligned} P_{CAB} &= \mathbf{CAB}P \\ P &= (\mathbf{CAB})^{-1}P_{CAB} \\ P &= \mathbf{B}^{-1}\mathbf{A}^{-1}\mathbf{C}^{-1}P_{CAB} \end{aligned}$$

The three steps are then:
1) Scale P_{CAB} with $\frac{1}{4}$ in x and $\frac{1}{6}$ in y.
2) Translate along the vector (-1,-2,-3).
3) Finally, rotate -45 degrees around the x-axis.