1 Graphics Pipeline, Linear & Affine Transformations in $\mathbb{R}^2$

1.1 Inverses of Linear and Affine Transformations

We begin with a definition.

Definition 1.1

Let A and B be transformations of $\mathbb{R}^2$. We say that B is the **inverse** of A, written $B = A^{-1}$, if:

1. $A \circ B = I$.
2. $B \circ A = I$.

1.1.1 Finding Inverses of Linear Transformations

How do we compute inverses? One way to do so is to do it by matrices. Let $A : \mathbb{R}^2 \mapsto \mathbb{R}^2$ be represented by the matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Then, A^{-1} is represented by M^{-1}, where

$$M^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \frac{1}{ad - bc}.$$

1.1.2 Finding Inverses of Affine Transformations

Let $A(\mathbf{x})$ be defined by

$$A(\mathbf{x}) = B(\mathbf{x}) + \mathbf{u},$$

where B is linear and $\mathbf{u} \in \mathbb{R}$ so that A is affine. How do we compute its inverse? Now, to find A^{-1}, we need to solve

$$\mathbf{y} = B(\mathbf{x}) + \mathbf{u}$$

to get $\mathbf{x}$ in terms of $\mathbf{y}$. Now, we have

$$\begin{aligned} \mathbf{y} &= B(\mathbf{x}) + \mathbf{u} \\ &\implies B(\mathbf{x}) = \mathbf{y} - \mathbf{u} \\ &\implies \mathbf{x} = B^{-1}(\mathbf{y} - \mathbf{u}) \\ &\implies \mathbf{x} = B^{-1}(\mathbf{y}) - B^{-1}(\mathbf{u}) \\ &\implies \mathbf{x} = \underbrace{B^{-1}(\mathbf{y})}_{\text{Linear Part}} + \overbrace{(-B^{-1}(\mathbf{u}))}^{\text{Translation Part}}. \end{aligned}$$

1.2 Compositions and Generalized Rotations

Recall that we have

- Rotations: R_θ.
- Scalings: $S_{\langle \alpha, \beta \rangle}$.
- Reflections.

We now want to discuss generalized rotations. Recall again that R_θ is a rotation about the **origin** θ. A **generalized rotation**, denoted $R_\theta^{\mathbf{u}}$, where $\mathbf{u} \in \mathbb{R}^2$ and $\theta \in \mathbb{R}$, is a rotation about some arbitrary point $\mathbf{u}$. In other words, we hold $\mathbf{u}$ fixed and then rotate about $\mathbf{u}$ counter-clockwise through angle θ. Consider the following example:

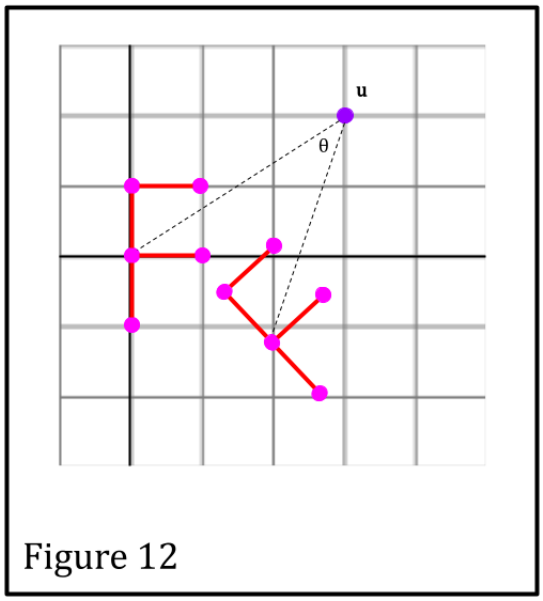

Figure 12

In the figure above[1], we have the *image* of "F" under $R_\theta^{\mathbf{u}}$. Our goal is to be able to express $R_\theta^{\mathbf{u}}$ in terms of rotations (about the origin) and translations. For an example like the one above, we could:

1. Move $\mathbf{u}$ to the origin $\mathbf{0}$; we call this $T_{-\mathbf{u}}$ (translation by $-\mathbf{u}$).
2. Perform the rotation about the origin $\mathbf{0}$; we call this R_θ (counter-clockwise rotation about the origin θ degrees).
3. Then, move $\mathbf{0}$ back to $\mathbf{u}$; we call this $T_{\mathbf{u}}$ (translation by $\mathbf{u}$).

We can write this as a **composition**. In particular[2]:

$$R_\theta^{\mathbf{u}} = T_{\mathbf{u}} \circ R_\theta \circ T_{-\mathbf{u}}.$$

So, we have:

$$R_\theta^{\mathbf{u}}(\mathbf{x}) = T_{\mathbf{u}}(R_\theta(T_{-\mathbf{u}}(\mathbf{x}))) = R_\theta(\mathbf{x}fx + -\mathbf{u}) + \mathbf{u}.$$

Visually, this might look like:

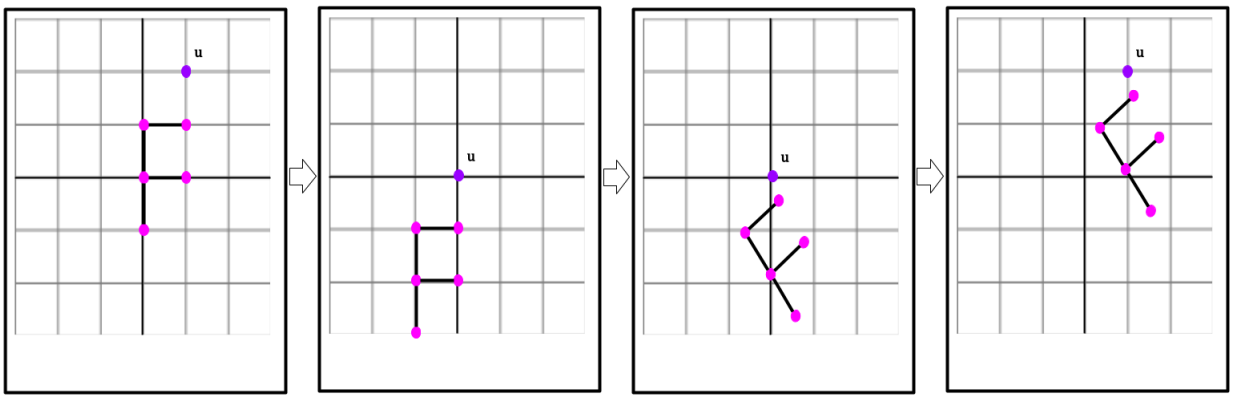

Figure 13

In general, rotations and translations **do not commute**. In general, we have

$$T_{\mathbf{u}} \circ R_\theta \neq R_\theta \circ T_{\mathbf{u}}.$$

[1] Not drawn to scale.

[2] Remember that if we did something like $(f \circ g)(x)$, this is equivalent to $f(g(x))$. In other words, the composition that we want to perform first should be on the right side.

1.3 Homogeneous Coordinates & Matrix Representations

Definition 1.2

Let $x, y, w \in \mathbb{R}$ (scalars), assumiwhere $w \neq 0$. Then, the triple

$$\langle x, y, w \rangle \text{ or } \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

is a **homogenerous** representation of the point

$$\langle \frac{x}{w}, \frac{y}{w} \rangle \in \mathbb{R}^2.$$

(Example.) Some examples of homogeneous coordinates for $\langle 2, 1 \rangle \in \mathbb{R}^2$ are:

- $\langle 2, 1, 1 \rangle$.
- $\langle 4, 2, 2 \rangle$.
- $\langle 8, 4, 4 \rangle$.
- $\langle -2, -1, -1 \rangle$.

How do we give a matrix representation of an affine transformation, over homogeneous coordinates? Let $A(\mathbf{x})$ be affine; in particular,

$$A(\mathbf{x}) = B(\mathbf{x}) + \mathbf{u}.$$

Let B be a linear map, represented by the matrix

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}.$$

Let $\mathbf{u} = \begin{bmatrix} e \\ f \end{bmatrix}$. Let N be the 3×3 matrix

$$\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix}.$$

Let $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ be a representation of $\begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$. Then,

$$A\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} ax + cy + e \\ bx + dy + f \end{bmatrix}.$$

This gives us the transformation under A. Now, we note that

$$N\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + cy + e \\ bx + dy + f \\ 1 \end{bmatrix}.$$

This represents a homogeneous representation of the same point as $A(\mathbf{x})$.

Let us now suppose that w is something other than 1 (the general case). More namely, let us consider $\begin{bmatrix} wx \\ wy \\ w \end{bmatrix}$,

where $w \neq 0$. This is another homogeneous representation of the point $\begin{bmatrix} x \\ y \end{bmatrix}$. Then,

$$N \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = wN \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} w(ax+cy+e) \\ w(bx+dy+f) \\ w \end{bmatrix}.$$

This is another homogeneous representation of the point $A\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)$. *Additionally*, for any $\alpha \neq 0$, αN is a 3×3 matrix that also represents $A(\mathbf{x})$ over homogeneous coordinates.

(Example.) Suppose we're given the following affine transformation:

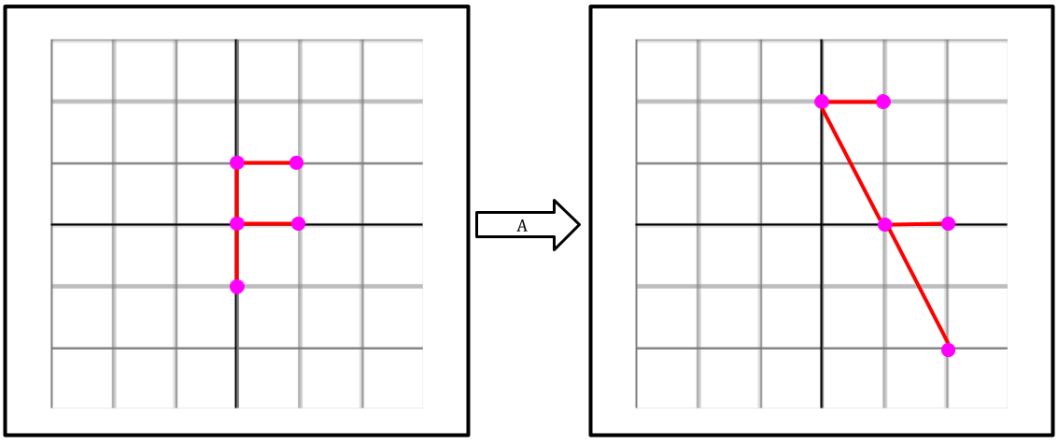

Figure 14

Find a matrix representation of an affine transformation.

The matrix representation for A is given by:

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

In other words, the above matrix is the 3×3 matrix representation of A. To see how we got this answer, we note that the

$$\begin{bmatrix} A(\mathbf{i}) - \mathbf{i} & A(\mathbf{j}) - \mathbf{j} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$$

is just the 2×2 matrix representation of the image, and the

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

is the translation. We keep the bottom

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

as is.

1.4 Hierarchical Transformations

Why do we need hierarchical transformations? Suppose we are simulating a moving object which has parts that are moving relative to the objects itself. We want to have some transformations – representing the

movement of the overall object – and then transformations controlling the cars. So, as an example, if we have a car that is moving, we might have matrices controlling the position of the car; we might also have matrices controlling where, say, the steering wheel or the wheels or the car door are.

Consider the following example:

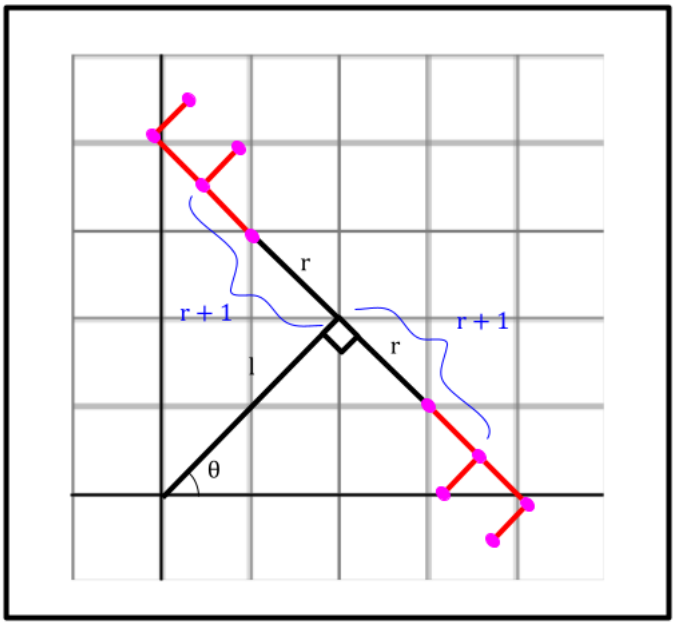

Figure 15

How do we move a standard "F"-shape to the upper "F" position? If we imagine the right-angle at $(2, 2)$ to be the "origin," then:

1. We can move the standard "F"-shape upwards $T_{\langle 0, r+1 \rangle}$.
2. Then, we can move the standard "F"-shape right by $T_{\langle \ell, 0 \rangle}$.
3. Then, we will rotate the "F"-shape by θ.

How do we represent this (upper "F") in terms of compositions?

$$R_\theta \circ T_{\langle \ell, 0 \rangle} \circ T_{\langle 0, r+1 \rangle}.$$

How about the lower "F" position? Again,

1. We rotate the "F"-shape by π (180 degrees).
2. We then move the "F"-shape down $T_{\langle 0, -(r+1) \rangle}$.
3. We translate the "F"-shape by $T_{\langle \ell, 0 \rangle}$.
4. Finally, rotate the "F"-shape by R_θ.

This gives us the transformations, in terms of compositions, as

$$R_\theta \circ T_{\ell, 0} \circ T_{\langle 0, -(r+1) \rangle} \circ R_\pi.$$

1.4.1 Application to OpenGL

How would this (roughly) look in OpenGL?

```
Let T = Theta
// Identity matrix
M_1 = Identity
// Compose with rotation about angle T
M_1 = M_1 o R_T
// Compose with translation by (l, 0)
M_1 = M_1 o T_(l, 0)
// Compose with translation by (0, r + 1)
// This gives us the first composition below Figure 15.
M_2 = M_1 o T_(0, r + 1)
Render ''F'' with M_2 as the model view matrix to render upper ''F''

M_3 = M_1 o T_(0, -(r + 1))
// This gives us the second composition below Figure 15.
M_3 = M_3 o R_pi
Render ''F'' with M_3 as the model view matrix.
```

In a real OpenGL program,

- Transformations are done as matrices acting on homogeneous coordinates. In $\mathbb{R}^3$, we use 4×4 matrices.
- Compositions is just matrix multiplication.
- Model view matrices get send to shader program as uniform variables.
- The vertex positions are vertex attributes stored in the VBO.