CSC4140 Computer Graphics

Assignment 3&4

March 19, 2022

Student ID: 120090453
Student Name: Haonan XUE

This assignment represents my own work in accordance with University regulations.

Signature:

1 Task 1

In the task, the triangle pixel is walked by BFS method:

1.The head node is set as the barycenter of the triangle.

2.Fetch the node from the head of the queue and pop the head node.

3.Walk through along vertical and horizontal directions by the head node. If the point is in the triangle, then put it into the queue.

4.When the queue is empty, the rasterization of the triangle is done.

The outcome is shown in the Figure 1.

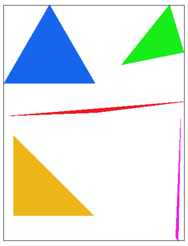

Figure 1. basic/test4.svg

2 Task 2

Supersampling offers more information in boundary of the triangle. It provides higher accuracy and better look.

In the pipline, supersampling is firstly applied. The frame buffer is sampled at 4x resolution. Then downsampling with the average of the supersampling unit pixels. In this way, we get a smoother image.

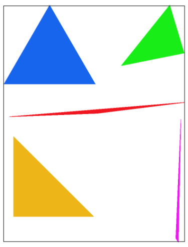

Figure 2. 4x Supersampling

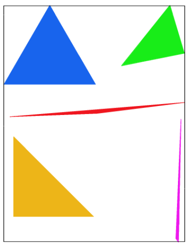

Figure 3. 9x Supersampling

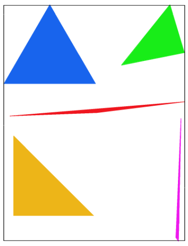

Figure 4. 16x Supersampling

3 Task 3

The transform of the model is described as the following matrix:

$$translate = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

$$scale = \begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Then we can transform the model to the view plane. I changed the direction of the robot's leg. Shown as the following:

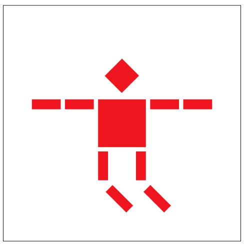

Figure 5. doc/robot.svg

4 Task 4

Baycentric coordinates can be expressed in the way of (α, β, γ) and it can be translated in the following way:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \beta \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \gamma \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

where

$$\alpha = \frac{-(x - x_1) * (y_2 - y_1) + (y - y_1) * (x_2 - x_1)}{-(x_0 - x_1) * (y_2 - y_1) + (y_0 - y_1) * (x_2 - x_1)}$$

$$\beta = \frac{-(x - x_2) * (y_0 - y_2) + (y - y_2) * (x_0 - x_2)}{-(x_1 - x_2) * (y_0 - y_2) + (y_1 - y_2) * (x_0 - x_2)}$$

$$\gamma = 1 - \alpha - \beta$$

The color of the point (x, y) is $color_{x,y}$, where

$$color_{x,y} = \alpha color_{x_0,y_0} + \beta color_{x_1,y_1} + \gamma color_{x_2,y_2}$$

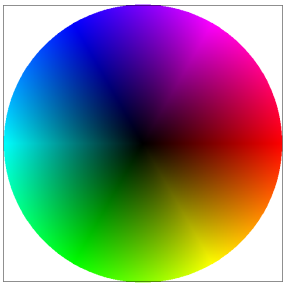

Figure 7. basic/test7.svg

5 Task 5

Pixel sampling is a kind of mapping – mapping the pixel from the texture to the screen. In a texture triangle, each texture pixel corresponds a pixel in the screen in barycentric coordinates. It can be described as the following form:

$$color_{screen}(\begin{bmatrix} x \\ y \end{bmatrix}) = color_{screen}(\alpha \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \beta \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \gamma \begin{bmatrix} x_2 \\ y_2 \end{bmatrix})$$

$$= \text{color}_{texture}(\alpha \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \beta \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} + \gamma \begin{bmatrix} u_2 \\ v_2 \end{bmatrix})$$

$$= \text{color}_{screen}(\begin{bmatrix} u \\ v \end{bmatrix})$$

where

$$\alpha = \frac{-(x - x_1) * (y_2 - y_1) + (y - y_1) * (x_2 - x_1)}{-(x_0 - x_1) * (y_2 - y_1) + (y_0 - y_1) * (x_2 - x_1)}$$

$$\beta = \frac{-(x - x_2) * (y_0 - y_2) + (y - y_2) * (x_0 - x_2)}{-(x_1 - x_2) * (y_0 - y_2) + (y_1 - y_2) * (x_0 - x_2)}$$

$$\gamma = 1 - \alpha - \beta$$

In the task, we can see the difference between nearest sampling and bilinear sampling. Since nearest sampling is discrete, the sampling image of nearest has a sharper edge compared with the bilinear sampling. With the increasing sampling rate, the difference seems to be smaller.

Figure 8. Nearest sampling and bilinear sampling with 1x sampling

Figure 8. Nearest sampling and bilinear sampling with 16x supersampling

6 Task 6

Level sampling offers a dynamic sampling level. For those with larger gradient, the rasterizer samples at a high sampling level (which has a lower sampling times). Tradeoff comparison among the zero, nearest and linear level sampling is listed in the following:

Speed: *Zero* < *Linear* < *Nearest*

Memory usage: *Nearest* < *Linear* < *Zero*

Antialiasing power: *Nearest* < *Linear* < *Zero*

The following figures show the outcomes of different level sampling and pixel sampling.

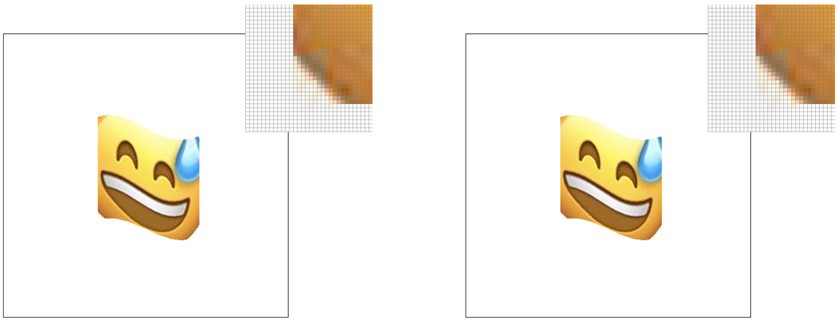

Figure 9. Pixel nearest sampling and bilinear sampling with zero level sampling

Figure 10. Pixel nearest sampling and bilinear sampling with nearest level sampling