CSC4140 Assignment III

Computer Graphics

March 19, 2022

Midterm

Student ID: 119010434

Student Name: Zhang Qihang

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

Signature:

1 Overview

In this assignment, I realized the rasterization of triangle, realized the anti-aliasing effect through SSAA and MSAA, and realized three kinds of transforms at the same time. I also realized the calculation of the color data obtained by interpolation through the barycenter coordinates. Finally, I realized the corresponding pixel color data obtained by mapping according to the texture. And realize mipmap and linear interpolation to get better image effect.

2 Task1

2.1 Walk through how you rasterize triangles in your own words.

Firstly, a range that can completely surround the triangle is obtained according to the coordinates of the three vertices of the triangle. Then all pixels within this range are traversed to determine whether the pixel is inside the triangle. If so, the pixel will display the correct color.

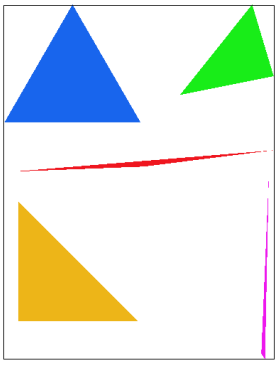

Figure 1: Test4.svg default

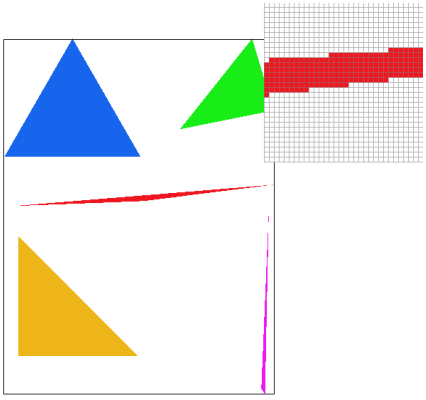

Figure 2: Test4.svg with Zoom

3 Task2

3.1 Walk through your supersampling algorithm and data structures.

For supersampling, we need to increase the original judgment points during traversal. For example, quadruple supersampling needs to sample four points, and these points divide the whole square pixels into four squares equally. Then, a more appropriate pixel color can be obtained according to the data of the four points.

3.2 Why is supersampling useful?

The color of each pixel can be displayed more reasonably with the same total number of pixels by using the supersampling technology. The problem of image sawtooth can be greatly alleviated by using the supersampling technology, but it will also greatly increase the amount of computation.

3.3 What modifications did you make to the rasterization pipeline in the process?

When the sampling rate is greater than 1, the points traversed need to be multiplied. Therefore, we redesigned the FOR loop and added a new for loop to super-sample each pixel on the screen, calculate and store the color value of adjacent points, and then determine the appropriate color data of this pixel through MSAA or SSAA. Instead of expanding the entire sample Buffer, I chose to store the calculated data into the sample Buffer, which greatly reduced the space required.

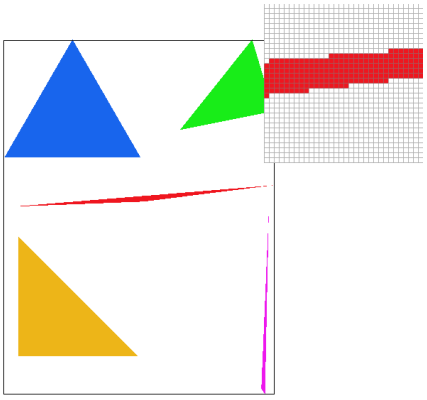

Figure 3: Test4.svg with X1

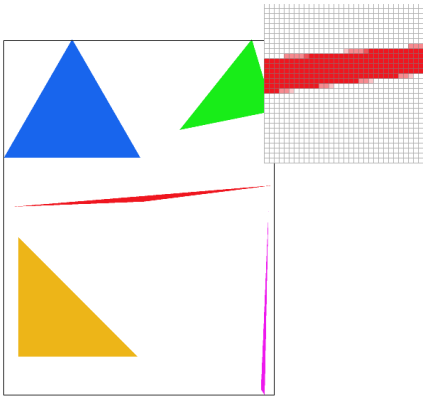

Figure 4: Test4.svg with X4

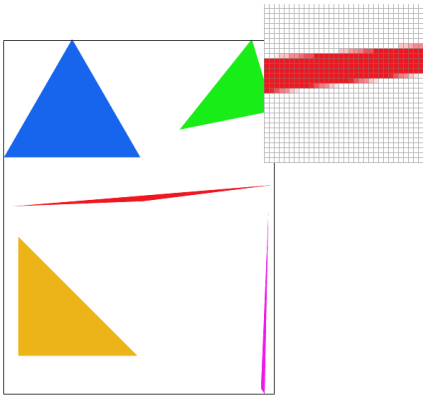

Figure 5: Test4.svg with X16

We observe that as the sampling rate increases, the triangle edge flattens out as a whole.

3.4 Extra

1.Two methods (SSAA MSAA) are used, but due to the svg problem(all are in one color), the effect of the two methods is similar. 2.Reduces the space used using array to get color before stored in the sample buffer.

4 Task3

Realized a man with six arms and a small tail.

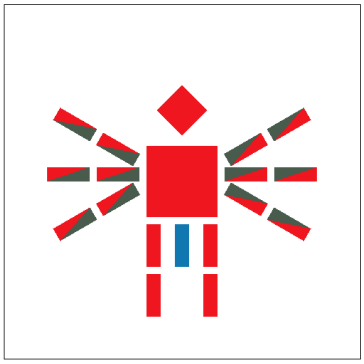

Figure 6: Transform.svg

5 Task4

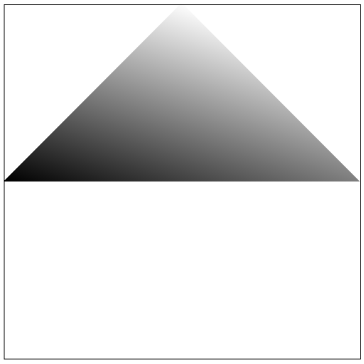

Figure 7: Color Tri in svg

This image shows the effect of barycentric interpolation on color calculation. In this image, the colors of the three corners are black, white and gray. From the barycentric coordinates, the proportion of three colors at any point in the triangle can be calculated, and this proportion is quantified from the values of the barycentric coordinates.

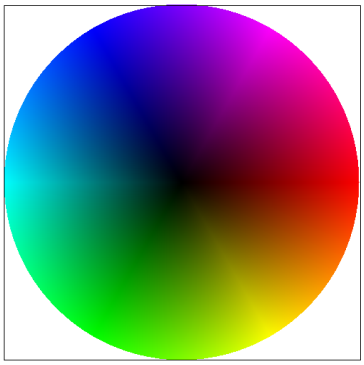

Figure 8: Color Cycle in svg

6 Task5

6.1 Explain pixel sampling in your own words and describe how you implemented it to perform texture mapping.

The triangle in the rasterizer, three vertices on the map correspond to the three points, according to the practice in similar to the last task, we through the barycentric coordinates to get corresponding to each pixel in the points on the map, but the point is not necessarily lies squarely on the integer point, so you need to find nearest or using bilinear interpolation to get the color of the corresponding data.

6.2 Briefly discuss the two different pixel sampling methods, nearest and bilinear.

The nearest integer point to a float can be found using round, where as bilinear is the color of the four points around the float and their proportion.

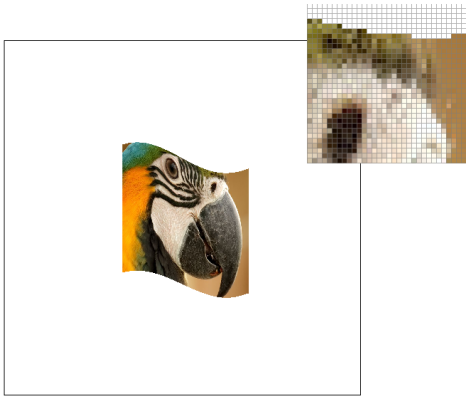

Figure 9: Nearest Sampling

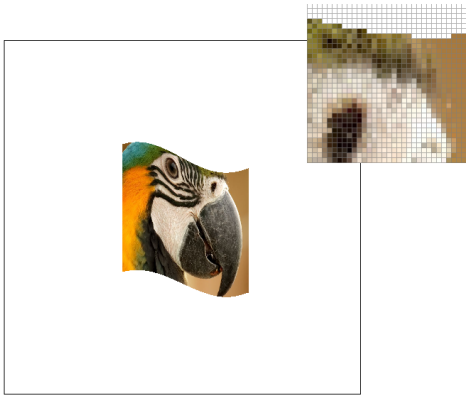

Figure 10: Bilinear Sampling

7 Task6

7.1 Explain level sampling in your own words

A complete map in the distance may be only a few pixels in screen space, so a pixel in screen space must correspond to a range of points on the texture map, leading to serious errors. Therefore, we can use The Level MipMap to solve this problem. The higher level map already combines the information of a large range of points, so the corresponding graphics are better.

7.2 Describe how you implemented it for texture mapping.

Then calculate the length of displacement in the texture based on the maximum displacement. If the level is different, just change the former level 0 texture to the corresponding level texture.

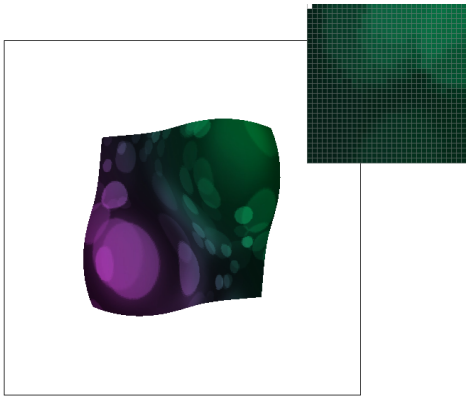

Figure 11: L ZERO and P NEAREST

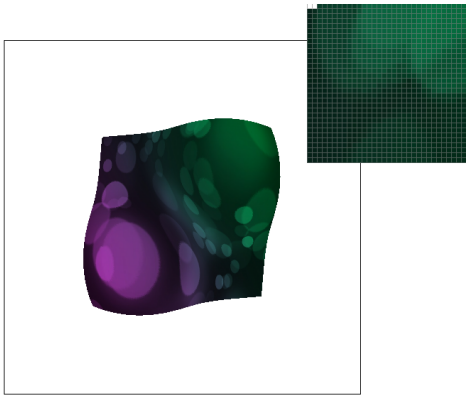

Figure 12: L ZERO and P LINEAR

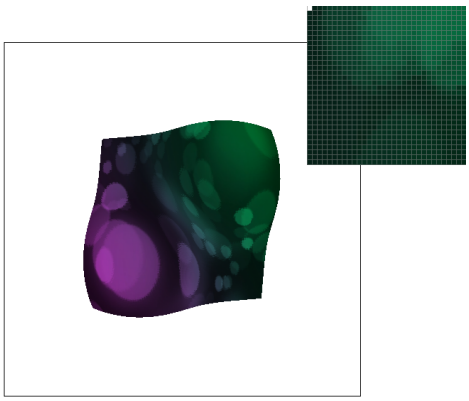

Figure 13: L NEAREST and P NEAREST

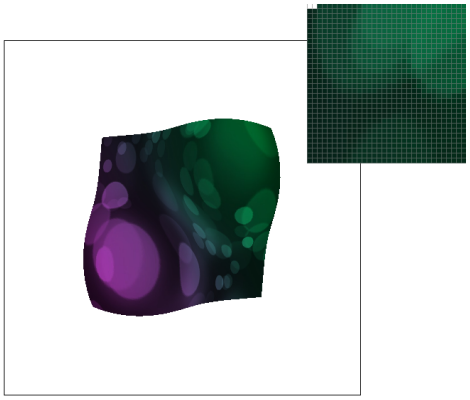

Figure 14: L NEAREST and P LINEAR

With L ZERO to L NEAREST and P NEAREST to P LINEAR, time and space used is increasing, but picture become better.

8 Extra

1.Two methods (SSAA MSAA) are used, but due to the svg problem(all are in one color), the effect of the two methods is similar. 2.Reduces the space used using array to get color before stored in the sample buffer.