Assignment 2: Triangles and Z-buffering

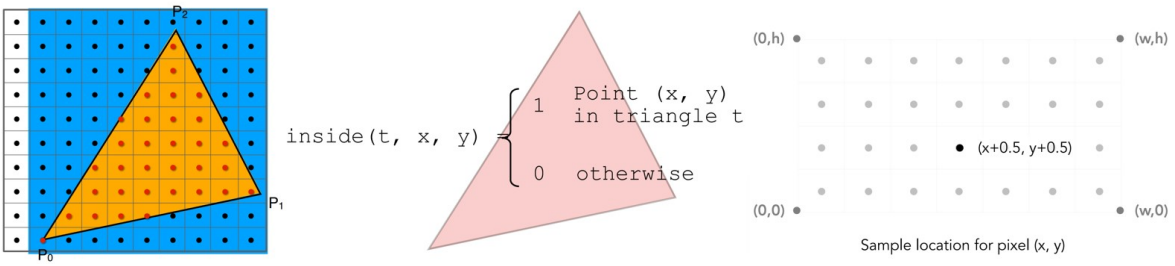
Mengzhu Wang

January 23, 2024

1 Z-buffer

1.1 Overview

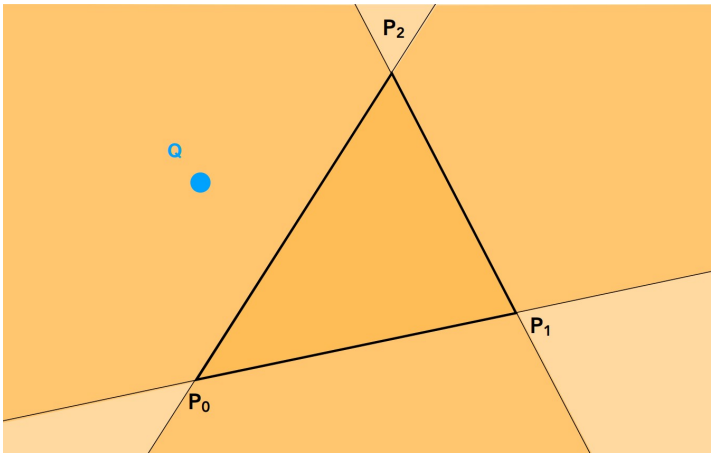
- Create 2D bounding box for triangle according to its vertex coordinates.
- Sample if each pixel center is inside triangle.
- If is inside triangle, compare interpolated depth value and corresponding value in depth buffer.
- If current point is closer to camera, set the pixel color and update depth buffer.



1.2 Evaluate inside(tri, x, y)

Three Cross Products

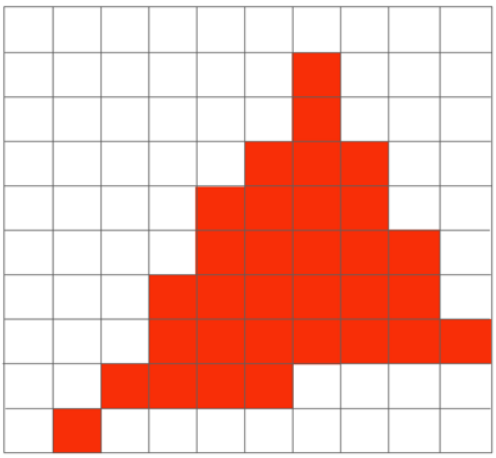
- The direction of $P_1P_2 \times P_1Q$ is outside $\rightarrow$ z is positive $\rightarrow$ Q is left to P_1P_2.
- If Q is left to all sides of the triangle, Q is in the triangle.



2 Antialising By Supersampling

2.1 Aliasing

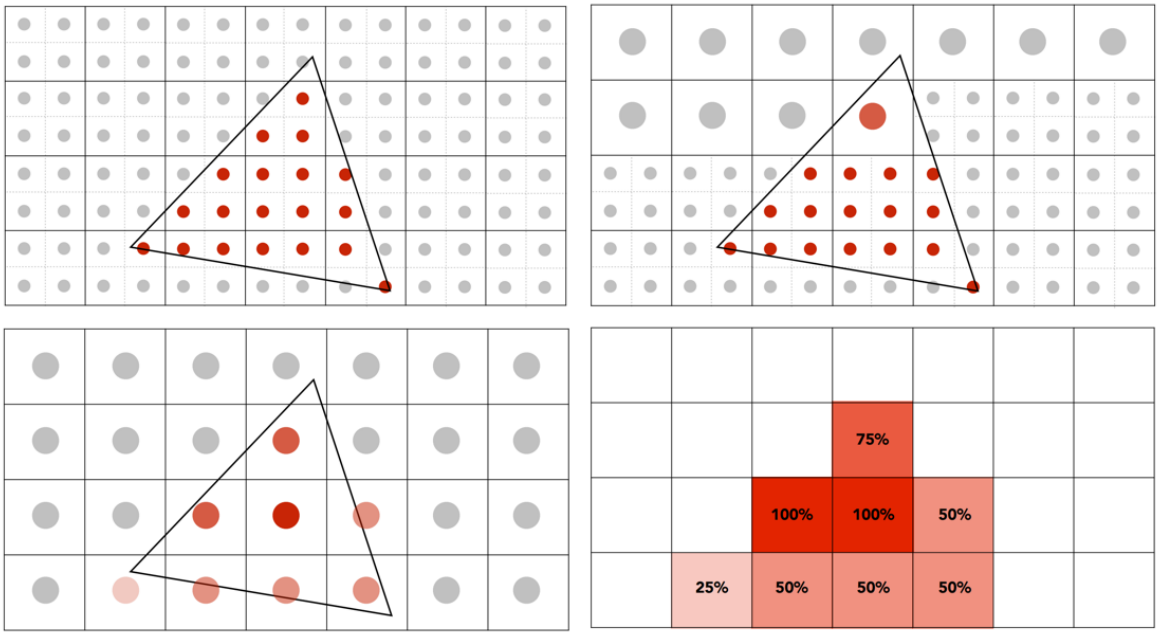
Pixels are uniformly-colored squares, leading to Jaggies!



2.2 MSAA (Multi-sample Anti-Aliasing)

Approximate the effect of the 1-pixel box filter by sampling multiple locations within a pixel and averaging their valuse.

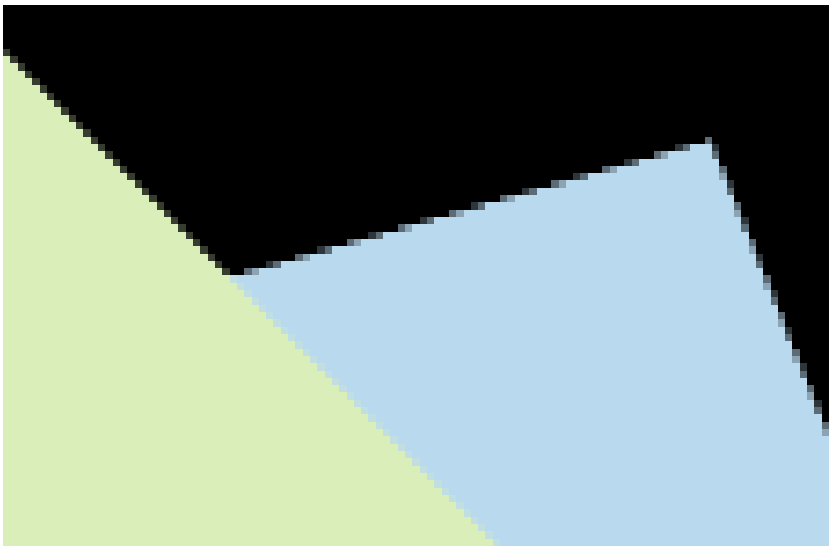
- Take $N \times N$ samples in each pixel.
- Average the $N \times N$ samples inside each pixel



2.3 Black Border

At the border, front green triangle only covers 1/4 pixel, and back blue one should cover 3/4 pixel. However, the depth of blue triangle is larger and can not cover the color of this pixel. The pixel color is finally set to 1/4 green, which is closer to black background.

Each pixel is super-sampled with 2 * 2 samples. Therefore, the original image can be enlarged up to four times. If at least one sample needs to update depth in buffer, we also need to update color.



3 Puzzle

zNear and zFar in funciton get_projection_matrix is positive, while they are negtive in class. Therefore, it leads to symmetric flip.

Sol.

- Add a minus sign to t.
- Change input zNear and zFar to negative value. Change z() = vert.z() * f1 + f2 in funciton draw() of rasterizer.cpp to z() = -vert.z() * f1 + f2.