

Lab 2: Triangle Rasterization

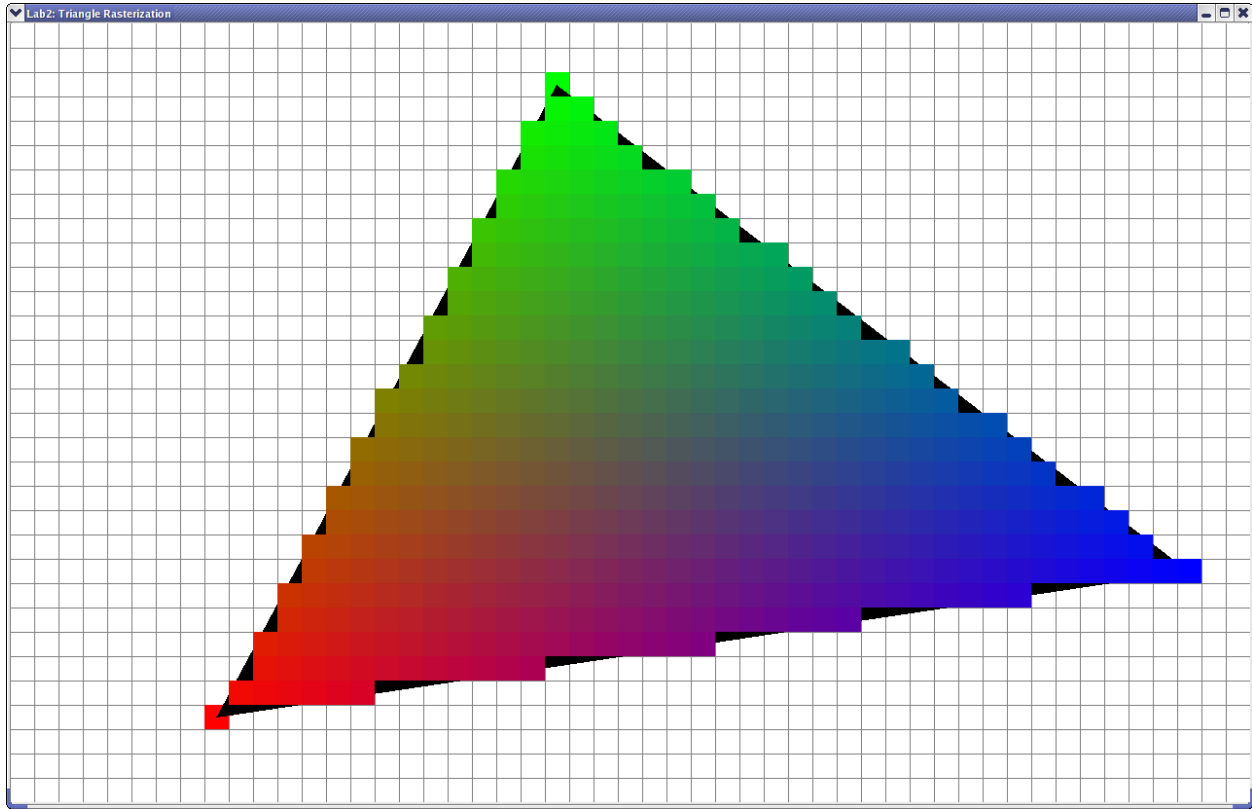


Fig 1: Screenshot of a coarse grid in which each square represents a virtual pixel. The vertices of the triangle (in red, green, and blue) are selected by mouse-clicks. The black, solid triangle rendered using OpenGL is a guide to the virtual pixel area to be shaded. Notice the gradients achieved using Gouraud color interpolation.

This lab will explore triangle rasterization with Gouraud shading (interpolation of the vertex colors across the triangle). The support code provided for this lab displays a grid of squares, each representing a virtual pixel; the term *virtual pixel* will refer to such a square. As shown in Fig 1, the vertices of the triangle appear in red, green, and blue respectively when the user clicks to select them on the grid and an additional click resets the screen. A black, solid triangle with vertices at the virtual pixel centers is rendered as a guide.

Your task is to implement an algorithm to calculate and plot the virtual pixels that lie on and inside the triangle and to color them appropriately (by interpolating vertex colors). Your starting point is the `disp()` function that is registered as the display callback with GLUT. You may add helper functions as you see fit.

The function that you will need to use to *plot* a *virtual pixel* with a specific RGB color is

```
void drawpoint(int x, int y, Glclampf r, Glclampf g, Glclampf b);
```

You are encouraged to examine how this function is implemented.