

## UM EECS 487

### Lab 2: Triangle Rasterization

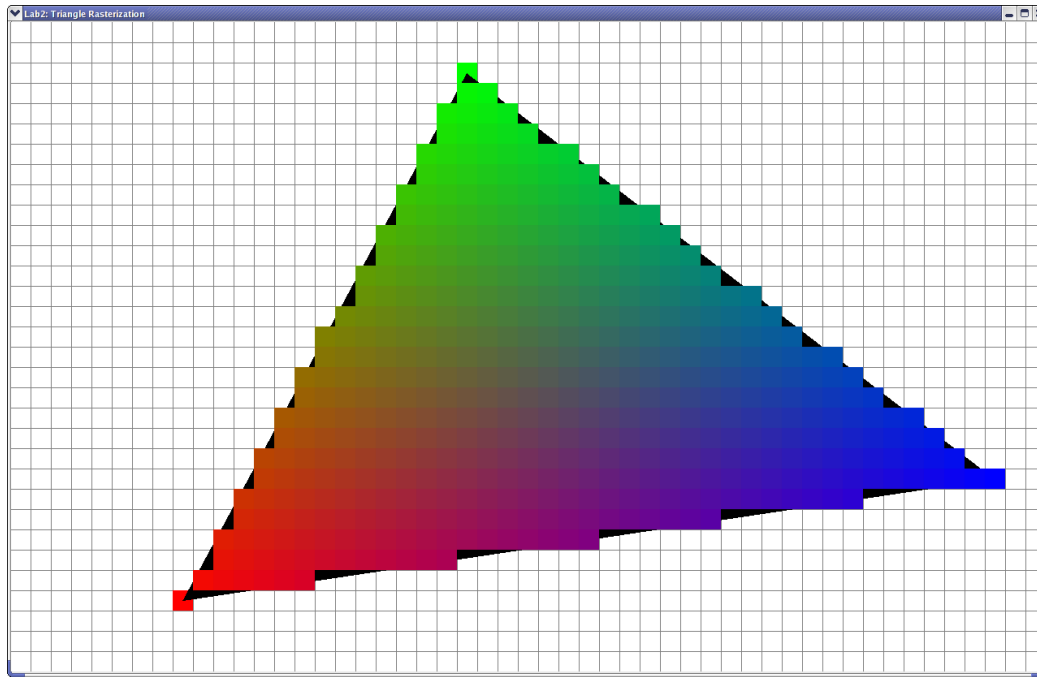


FIGURE 1. A rasterized triangle with the color interpolated from the three vertices.

In this lab you will implement triangle rasterization with color interpolated from the triangle's three vertices. The support code provided for this lab displays a grid of squares, each representing a *virtual pixel*. 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; an additional click clears and resets the screen. A black, solid triangle with vertices at the center of the selected virtual pixels is also rendered, as a guide.

Your task is to implement an algorithm to calculate and plot the virtual pixels that form (are on or inside) the triangle between its three vertices. The virtual pixels must also interpolate the color between the vertices. You can start with the `display()` function that is registered as the display callback function with GLUT. You may add helper functions as necessary.

The one function you will absolutely need to *plot a virtual pixel* with a specified 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.

One thing to consider, are there any *boundary cases* with this lab? Those would be specific cases where the algorithm “falls apart” or has “undesirable” behavior.