Image Processing

Due date: Monday, February 22, 11:59 pm

1 Overview

Write a program that performs two tasks: line drawing and scan conversion for polygons with texture mapping. The program should be interactive, that is, the endpoints of the lines and the vertices of the polygon should be defined interactively by the user. Switching between the two modes can be done using the pull-down menu.

In this assignment only, you should not use OpenGL functions for drawing lines and polygons. Create an internal array of RGB values, and do all drawing there; display the image using OpenGL raster functions `glRasterPos`,`glDrawPixels`.

Your program should do the following:

- Draw lines using Bresenham algorithm. The user picks two endpoints with the mouse; when the second endpoint is picked, the line is drawn; old lines should not be erased as the new ones are added. An important thing to remember: in the `draw` callback in your code you should not assume that the results from the previous calls are visible; you should start from scratch by calling `glClear`.
- Read an image in `ppm` format and display it. User can pick four points in the image; calculate the transformation that would map the original image into the quadrangle picked by the user, and compute the warped image using scan conversion and filtering as it was explained at the lecture. After the image was computed, write it out (the name for the output image should be supplied on the command line).

For the purposes of calculating the transformation, assume that the coordinates in the image extend from -1 to 1 in x and y directions.

Assume that vertices of the quadrangle are picked in order, i.e. the point of the image with coordinates (-1,-1) should map to the first point, (1,-1) to the second picked point, (1,1) to the third and (-1,1) to the last. Check the input for validity, i.e. detect and refuse to accept concave and self-intersecting quadrangles, and quadrangles with three vertices on one line.

2 What to turn in

A working version of your program; also, generate images of the checkerboard with four vertices approximately in the following positions:

- (-0.9,-0.9), (0.9,-0.9), (0.9,0.9), (-0.9,0.9)
- (-0.2,-1), (0.8,0), (0.2,0.8), (-1,0.1)
- (-1,-1), (1,-1), (0.1,1), (-0.1,1)

- (-1,-1), (-0.8,-1), (1,1), (0.8,1)

All coordinates are normalized to the ranges -1..1, -1..1; to get pixel coordinates, use appropriate translation and scaling.

The checkerboard image is NOT the one from the previous assignment; a new one will be posted on the net for this assignment.