

CSE 590 PROJECT 1 CP1 REPORT

IMPLEMENTATION OF COLLISION DETECTION FOR 3D PRINTERS

Name		UB Person #
1	Libing Wu	50042948
2	Amaan Akhtarali Modak	50206525
3	Shashank Suresh	50208025

Submitted On: 3/27/2017

I. ABSTRACT

LINE INTERSECTION:

We need to design a module that takes in a line segment coordinate pairs (Unsigned 8 bits, $[x_1, y_1, z_1, x_2, y_2, z_2]$) and whether they intersect (Boolean 1bit). Initially we design 2D printing collision detection. We will try to implement 3D printing collision detection in the next checkpoint.

Preconditions:

- All the coordinates are positive integers
- All the line segments are partitioned by layers
- All the line segments are in ascending order

Design Explanation:

Step 1:

We design a memory module to store the voxel coordinate pairs for each line segment on the same layer. Set current layer height as $Z=0$.

Step 2:

We read one voxel coordinate pair (6 integers) into our Verilog module. We set the layer height of the coordinate pair as $Z=0$. If the layer height of this line segment is the same as Z (that is, we check if the 'z' coordinates of both these points are the same), then we determine whether the current line segment has intersection point with other line segments whose layer height is also equal to Z . If there is intersection, we output the line segment index number as malicious G-code. Otherwise, we store the current line segment into memory. If the layer height is larger than Z , it means another layer is starting. We then update Z to current layer height and clear the memory module and store the current line segment into memory.

Step 3:

This keeps repeating until no line segments are left

II. DESIGN FLOW CHART

The example flow chart consists of sub-modules is shown in Fig. 1.

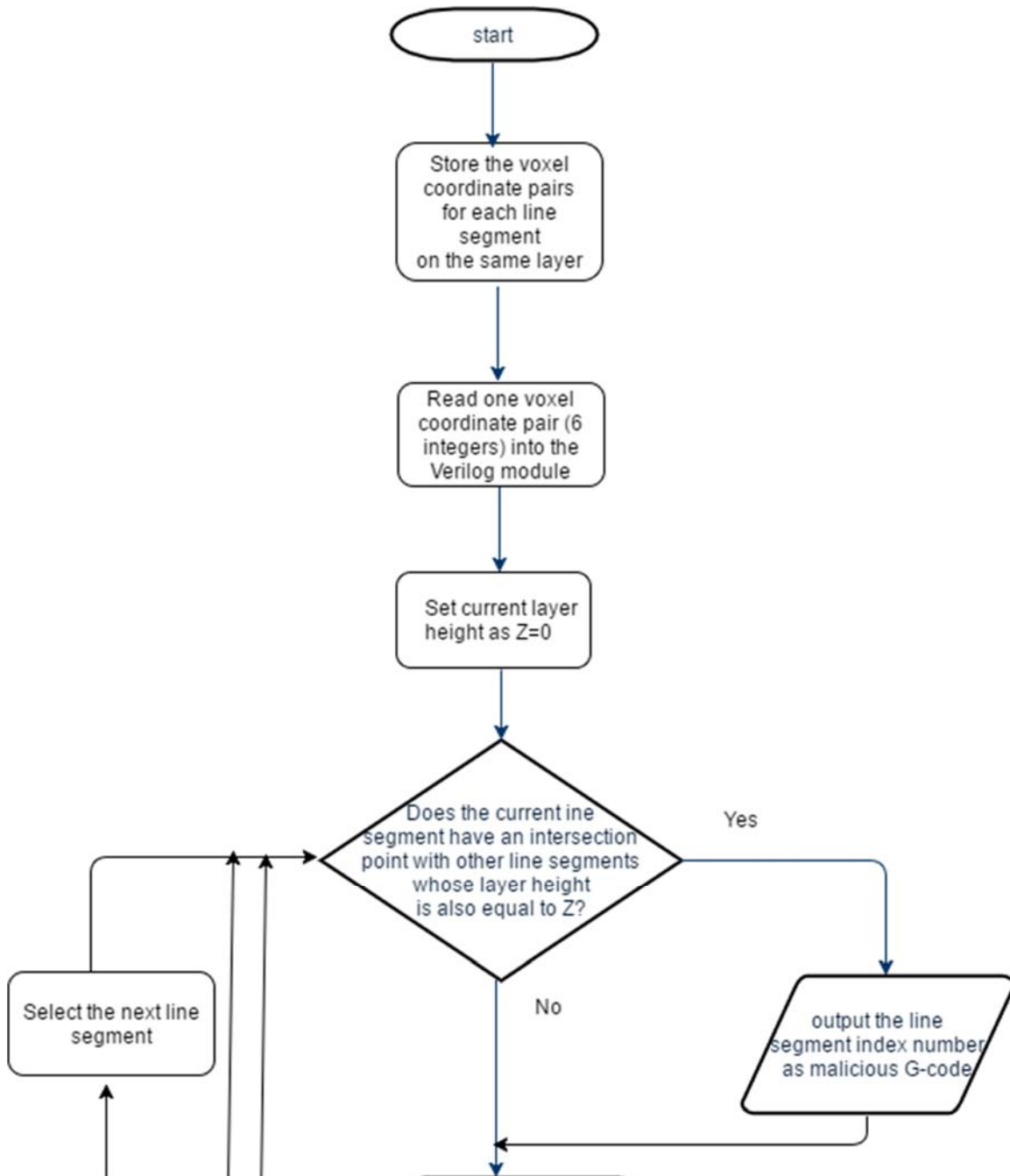
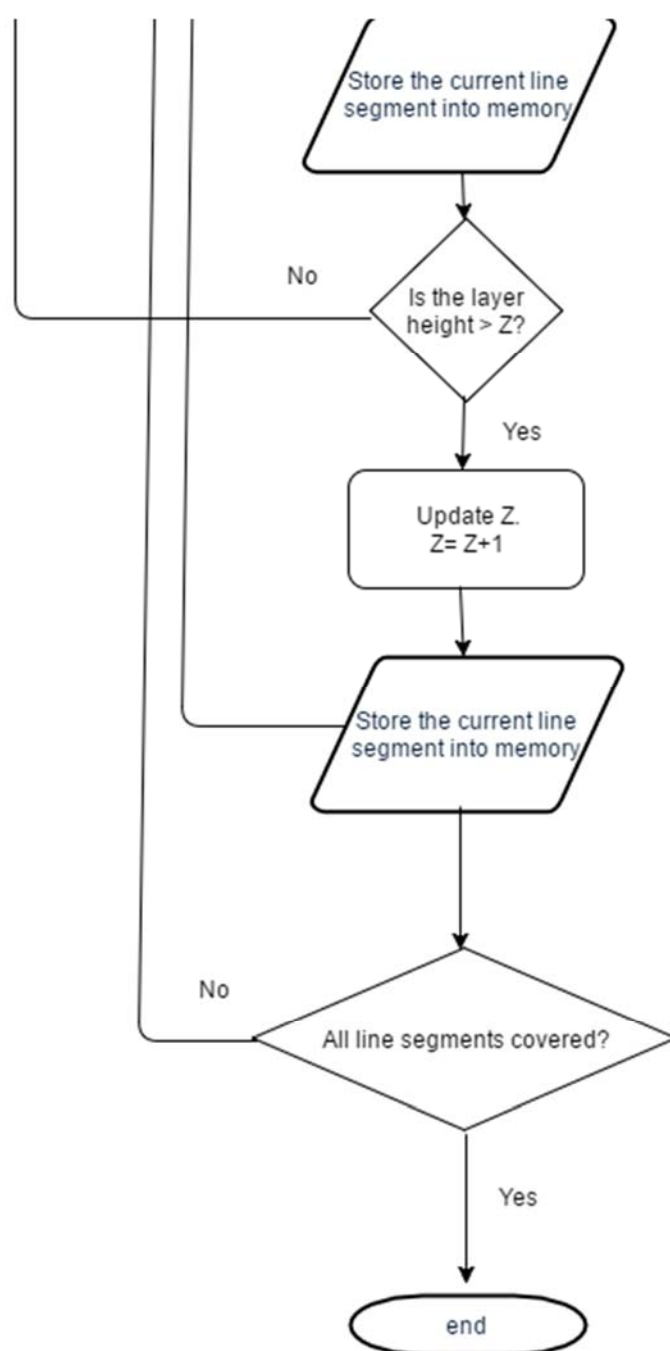


Fig.1a. Flow Chart



III. SIMULATED RESULTS

This is the simulated waveform from Xilinx ISE webpack. The work is still being carried out and below screenshots are to check if two lines are intersecting each other or not.

Input: 4 2D points

Output: 2 states (0-no collision,1-collision)

