line segment

Title: Visualizing Line Segment Intersection Algorithms

Team Members

The project is not simple enough for one person. I would be woking on it all by myself.

Following are the team members:

- Dhrushit Raval - dr9703

Problem Description

One of the very basic problems in computational geometry, is to find intersection of two lines segments. In this project, I will build a tool that would help visualize the algorithm that is used to find the intersection in hopes to make it easy to understand. The algorithm used is a simple line sweep algorithm.

In this project, I will implement the line sweep algorithm as well as a brute algorithm for finding intersections for n line segments. This would also allow user to compare and understand how things work. Further details about how this algorithms is provided in the literature tab, and a visualization, will soon be provided in the visualization tab.

The complexity of this algorithm is $O((n + I) * log(n))$. The explanation of the same can be found in the literature section.

Timeline

As per the deadlines set by the university, the final submission will be done in week 11. Currently, this is week 4. And as I start building up the tools, I would have 6 weeks to complete the project. Following timeline is what I estimate would be followed in this project:

Week 5:

- Iterate over the algorithm details

- Implement data structures needed

Week 6:

- Implement the algorithm with terminal output
- Implement brute force with terminal output
- Assemble the components

Week 7:

- Look for existing data set to test the algorithm and test extensively
- Add corner case handling
- Start search for the visualization library and learn about it

Week 8:

- Use the visualization library to map the algorithm to the visual output
- Figure out a way to put stop points, add undo and step-forward functions

Week 9:

- Finishing touches and complete what is left
- I am expecting some delays in previous weeks due to other projects and midterms

Week 10:

- Final presentation preparations

Week 11:

- Final submissions

Algorithm Notes

- Re-iterate over the notes and literature
- Overall Implementation

    - O(n^2) for comparison
    - O((n+k) logn) the actual algorithm
- Implementation Details
    - Segment List
        - Data stored in slope, intercept format (m_i, b_i)
            - red-black tree
            - skiplist
        - For Insert, Delete, Find Successor, Find Predecessor in O($ln(n)$)
            - Splay Tree
            - AVL Tree
    - Event Queue
        - Events
            - Start of segment
            - End of segment
            - Intersection of segments
        - functions - **priority queue**
            - insert
            - findmin
            - deletemin
- Verify with existing data set of inputs
- Handle corner cases of vertical lines, overlapping lines, multiple segment intersection(long term)
- Make visualisation

References

- Notes from CMU: https://www.cs.cmu.edu/~15451-f17/lectures/lec21-sweepline.pdf

- Notes from UMD: https://www.cs.umd.edu/class/spring2020/cmsc754/Lects/lect04-intersection.pdf