Report Number:

2019-4

Computational Geometry Teaching Tool

Yujie Zhou and Tao Ju

When students are taking Computational Geometry course which covers many geometry algorithms, most of them are difficult to follow because these algorithms are very abstract even if authors draw pictures to illustrate. In order to help students to get a better understanding of these algorithms, we decide to design Computational Geometry Teaching Tool. This tool is a web application that covers 8 geometry algorithms : Graham Scan, Quick Hull, Line Segment Intersection, Dual, Line Arrangement, Voronoi Diagram, Incremental Delaunay Triangulation and Kd Tree. First, this tool is developed by using JavaScript so that users don't need to install any...
Read complete abstract on page 2.

Recommended Citation
Zhou, Yujie and Ju, Tao, "Computational Geometry Teaching Tool" Report Number: (2019). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/1172

Computational Geometry Teaching Tool

Yujie Zhou and Tao Ju

Complete Abstract:

When students are taking Computational Geometry course which covers many geometry algorithms, most of them are difficult to follow because these algorithms are very abstract even if authors draw pictures to illustrate. In order to help students to get a better understanding of these algorithms, we decide to design Computational Geometry Teaching Tool. This tool is a web application that covers 8 geometry algorithms : Graham Scan, Quick Hull, Line Segment Intersection, Dual, Line Arrangement, Voronoi Diagram, Incremental Delaunay Triangulation and Kd Tree. First, this tool is developed by using JavaScript so that users don't need to install any software or package. Furthermore, it breaks down the algorithm and go step by step so that students can move forward and backward on their own pace. Finally, all demos in this tool have same layout so that when students learn how to use the first one, they will know how to use others.

Computational Geometry Teaching Tool

---Yujie Zhou

1. Motivation:

Computational geometry covers many geometry algorithms used for solving geometry problems. However, when students are taking this course, most of them are hard to follow even if the author draw the pictures to illustrate algorithms. Furthermore, most of demos from the Internet cannot go step by step.

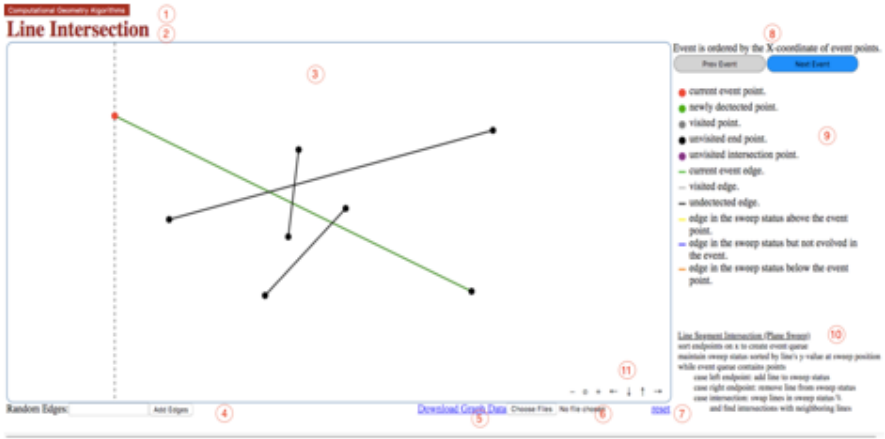
In order to help students to get a better understanding of those algorithms, we decide to develop Computational Geometry Teaching Tool.

2. Features

1. Using JavaScript to develop this tool. Because JavaScript is a platform-Independent language, students don't need to install any software or package. Students can directly open the link through browser.
2. Breaking down the algorithm and go step by step. Students are able to move forward and backward on their own pace.
3. Having consistent look and feel. So that when students learn how to use the first one, they will know how to use others.

3. Contributions

1. A framework for making demos: GUI, CodeBase
   - GUI : All demos in the tool have same layout



Event is ordered by the X-coordinate of event points.

● current event point.
● newly dectected point.
● visited point.
● unvisited end point.
● unvisited intersection point.
— current event edge.
— visited edge.
— undectected edge.
— edge in the sweep status above the event point.
— edge in the sweep status but not evolved in the event.
— edge in the sweep status below the event point.

```
Line Segment Intersection (Plane Sweep)
sort endpoints on x to create event queue
maintain sweep status sorted by line's y-value at sweep position
while event queue contains points
    case left endpoint: add line to sweep status
    case right endpoint: remove line from sweep status
    case intersection: swap lines in sweep status \
        and find intersections with neighboring lines
```

1 Section: link of other algorithms' app
2 Title
3 Graph: display diagram
4 Set random points or edges
5 Download the diagram as JSON file
6 Upload JSON file and display it on the graph
7 Reset: remove diagram
8 Button bar
9 Legend: description of different color
10 Algorithm Description
11 Side Bar: zoom in or zoom out, go left, right, up and down.

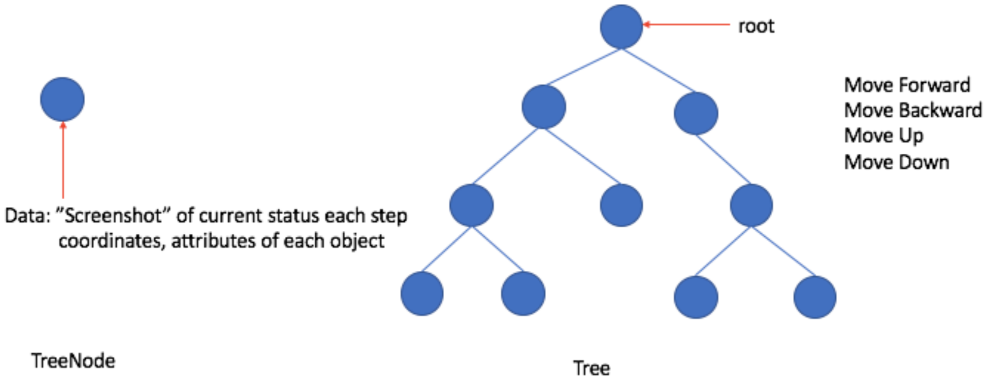
   - CodeBase

(1) Graph Container: graphContainer.js, desContainer.js
Create a new graph: var graph = new GraphContainer("graph title");
It supports that all demos will have same layout.

(2) Elementary Objects: Point.js, Edge.js, Face.js, Circle.js, Triangle.js
Create a new point: var point = new Point(coordinates, attributes);
We wrap the access code of JSXGraph so that developers don't' need to directly use JSXGraph code.

(3) Help Functions
e.g. Edge.intersection(e1, e2) : check if these 2 edges intersect.
Triangle.prototype.update(tri) : update the vertices of triangle

(4) Mechanism of step by step: Tree.js, TreeNode.js


root
Move Forward
Move Backward
Move Up
Move Down
Data: "Screenshot" of current status each step
coordinates, attributes of each object
TreeNode
Tree

TreeNode will store the data(the coordinates, attributes of each object inside the diagram). And the tree will adopt these treenodes so that students can move forward, backward, up and down.

2. A set of demos that cover common algorithms types.
(1) Convex Hull: Graham Scan --> Line Sweep Algorithm
Quick Hull --> Recursive Algorithm
(2) Line Segment Intersection -- > Line Sweep Algorithm
(3) Line Arrangement -- > Incremental Algorithm
(4) Delaunay Triangulation -- > Incremental Algorithm -- (Yujie zhou)
(5) Range Query -- > Recursive Algorithm – (Yujie zhou)
(6) Interactive demos : Dual, Voronoi Diagram(Yujie zhou)

4.Link

(1) WashU Server Path: http://students.engineering.wustl.edu/comp_geo_algorithms/devBranch/
Graham Scan : graham_scan/grahamScan.html
Quick Hull : quick_hull/quickHull.html
Line Segment Intersection : line_sweep/lineSweep.html
Line Arrangement : line_arrangement/lineArrangement.html
Dual : dual_point/dualPoint.html
Voronoi Diagram : Voronoi_diagram/VoronoiDiagram.html
Delaunay Triangulation : Delaunay_Triangulation/DelaunayTriangulation.html
KD Tree : KD-Tree/kd.html
(2) GitHub: https://github.com/RDerber/computationalGeometry/tree/layout_overhaul