Report for Computer GraphicII, HW2 2D Skeleton extraction by Chordal Axis Transform (CAT)

Sixun Dong, 2021233155

April 2, 2022

Acknowledgements:

Deadline: 2022-03-31 23:59:59

You should answer the questions in **English**

You can choose C++ or Python, and no restrictions on programming framework. You can freely use frameworks such as openGL.

The **report** submits as a PDF file to gradscope, the programming part should package all the files include code, input files, executable file, readme.txt, and report. The **package** name is **your_student_name+student_id.zip**.

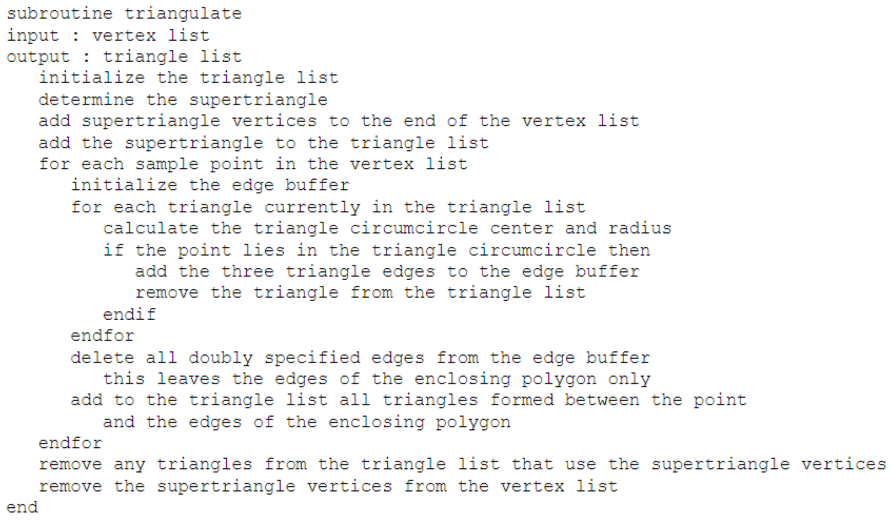
You will get Zero if the code not passing the plagiarism check.

1 2D Skeleton extraction by CAT

1.1 Delaunay Triangulation (40 points)

Implement 2D Delaunay Triangulation for any given sampling points. You are required to provide at least five examples (sample on the contours of some 2D shapes) and corresponding DT visualization.

Solution: The algorithm is from the paper[1] and refer to the note[2]. Pseudo code is as fig. 1 shown.



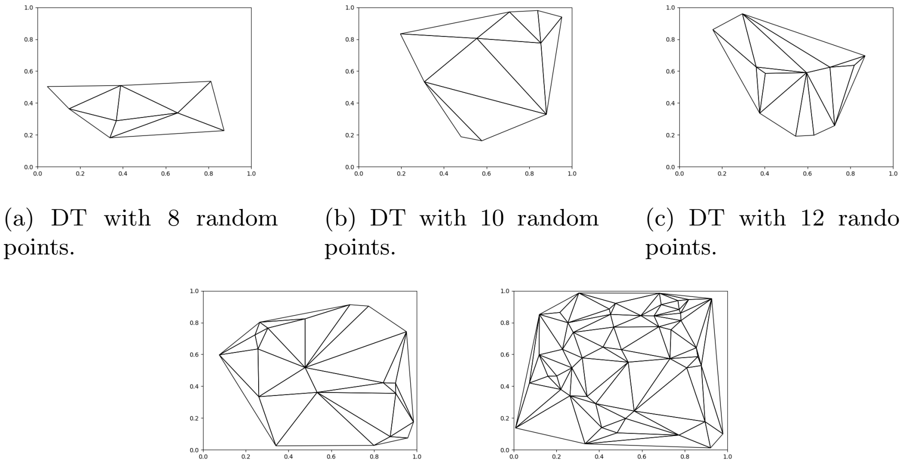
```
subroutine triangulate
input : vertex list
output : triangle list
   initialize the triangle list
   determine the supertriangle
   add supertriangle vertices to the end of the vertex list
   add the supertriangle to the triangle list
   for each sample point in the vertex list
      initialize the edge buffer
      for each triangle currently in the triangle list
         calculate the triangle circumcircle center and radius
         if the point lies in the triangle circumcircle then
            add the three triangle edges to the edge buffer
            remove the triangle from the triangle list
         endif
      endfor
      delete all doubly specified edges from the edge buffer
         this leaves the edges of the enclosing polygon only
      add to the triangle list all triangles formed between the point
         and the edges of the enclosing polygon
   endfor
   remove any triangles from the triangle list that use the supertriangle vertices
   remove the supertriangle vertices from the vertex list
end
```

Figure 1: The pseudo code of DT, which is from the paper[1].

The result as fig. 2 shown, it contains five examples figs. 2a to 2e, which are generated by DT with different numbers of random points.

[1] http://paulbourke.net/papers/triangulate/

[2] https://www.jianshu.com/p/172749e6116a



(a) DT with 8 random points.

(b) DT with 10 random points.

(c) DT with 12 random points.

(d) DT with 20 random points.

(e) DT with 50 random points.

Figure 2: Five examples of DT.

1.2 CAT (30 points)

Implement CAT and visualize the skeleton results of the above examples.

Solution: The implementation uses *Triangle*[3] module to help generating input data, implementing CDT algorithm and visualization.

The implementation have four step.

(1) Use *Triangle* to generate the input data which is a face.

(2) Use *Triangle.triangulate()* to generate CDT.

(3) Traverse[4] the triangles and divide them into three kinds. They independently have one , two, or three edges on the boundary.

(4) Use the CAT algorithm in the 3^{rd} slide to connect the middle points of the edges.

[3] https://rufat.be/triangle/

[4] The algorithm refers to the 10^{th} page of the paper: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.3204&rep=rep1&type=pdf

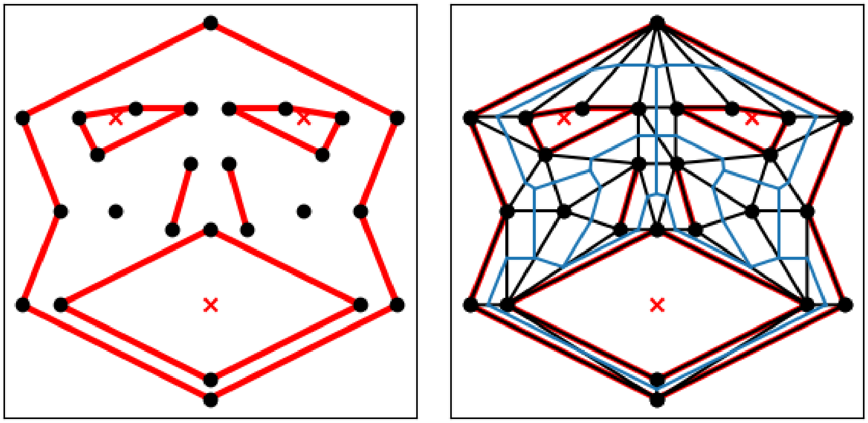
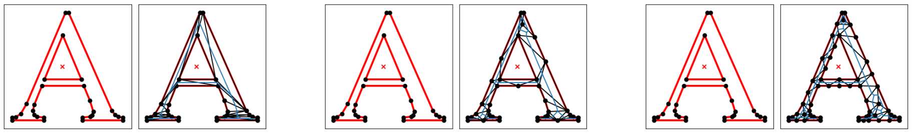


Figure 3: The result of Q2.

1.3 Comparison (30 points)

Compare and analyze the skeleton generated by CAT and by the connected centers of circumcircles of DT triangles on different sampling densities on the contours of 2D shapes.

Solution: By using *Triangle* to generate different numbers of triangles. The result as figs. 4 and 5 shown, by limiting the minimum angle of triangles, we could abtain the different numbers of triangles. And the result figure out that more triangles do not necessarily lead to the best results. CAT can achieve better results with the least number of triangles.

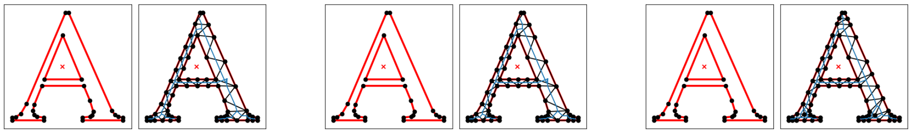


(a) The result of CDT with 29 triangles and the minimum angle of triangles is 0.

(b) The result of CDT with 43 triangles and the minimum angle of triangles is 10.

(c) The result of CDT with 77 triangles and the minimum angle of triangles is 20.

Figure 4: CAT: The different numbers of triangles by limiting the minimum angle of triangles.



(a) The result of DT with 66 triangles and the minimum angle of triangles is 0.

(b) The result of DT with 66 triangles and the minimum angle of triangles is 10.

(c) The result of DT with 82 triangles and the minimum angle of triangles is 20.

Figure 5: DT: The different numbers of triangles by limiting the minimum angle of triangles.