Weak Line Simplification Project
[Spring 2003]

Instructors:
Ouri Wolfson, Goce Trajcevski
Group Members:
Xin Li, Yuanyuan Jia, Feihong Hsu
{xli1, yjia, fhsu}@cs.uic.edu

Background

- Line simplification reduces the number of points while preserving the basic "shape" of trajectories.
- Two types of line simplification:
  - Strong: Use existing vertices
  - Weak: Don't use existing vertices
- Different metrics & algorithms:
  - We only implement 2D due to complexity of weak simplification algorithm
  - We only measure space savings

Weak vs. Strong

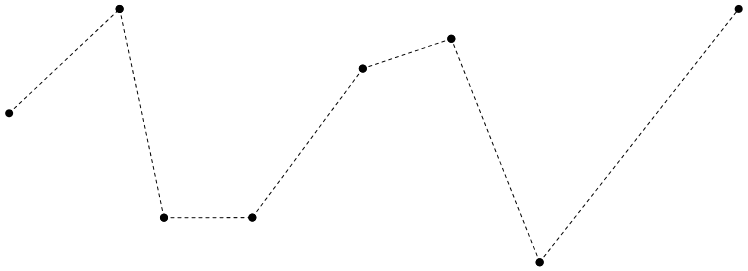


Figure 1. Original trajectory

Original trajectory

Strong line simplification

Weak line simplification

Weak vs. Strong

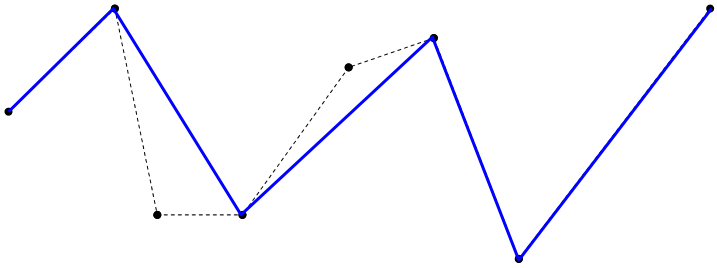


Figure 2. Original & strong simplified trajectory

Original trajectory

Strong line simplification

Weak line simplification

Weak vs. Strong

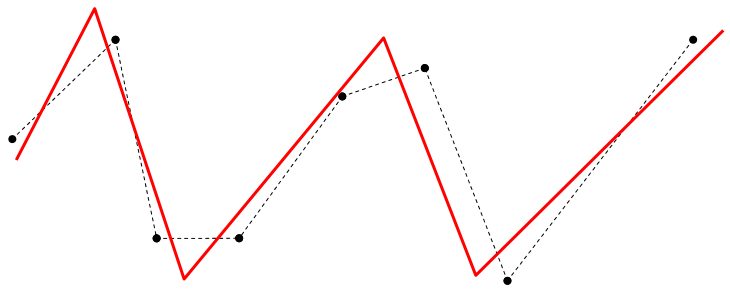


Figure 3. Original & weak simplified trajectory

Original trajectory

Strong line simplification

Weak line simplification

Weak vs. Strong

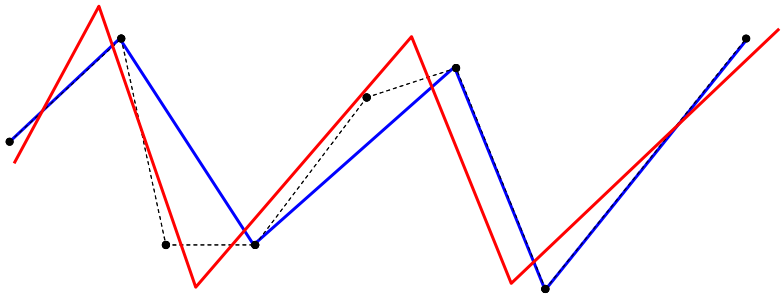


Figure 4. Original, strong simp. & weak simp.

Original trajectory

Strong line simplification

Weak line simplification

Goals

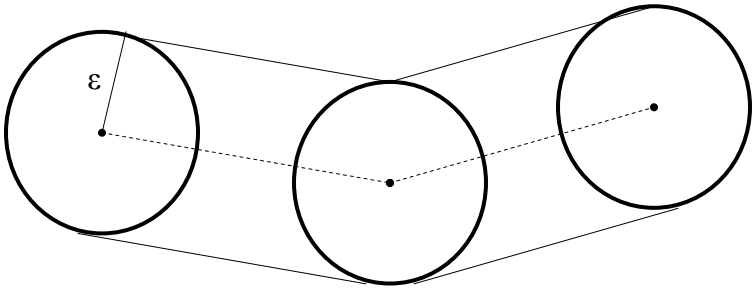
1. Implement basic 2D line stabbing algorithm from Guibas et al[1]
2. Use line stabbing to generate simplified trajectories
3. Improve line stabbing to handle very close vertices (overlapping circles)
4. Compare weak vs. strong simplification in terms of their average savings (in # of vertices)

Approach

- Modules:
  - Geometric and trigonometric functions
  - Line stabbing & weak simplification algorithm
  - Visualization
  - Integration
- Tools:
  - Visual C++ (Windows)
  - g++ (UNIX)
  - gnuplot for visualization

Line Stabbing Overview(1)

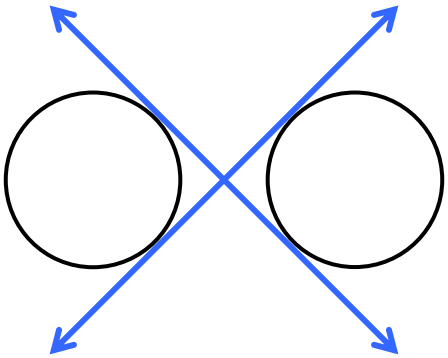
ε defines a "tube" around trajectory:



In the strictest form of line simplification, the simplified trajectory must be within the ε-tube.

Line Stabbing Overview(2)

Inner tangents of two circles:

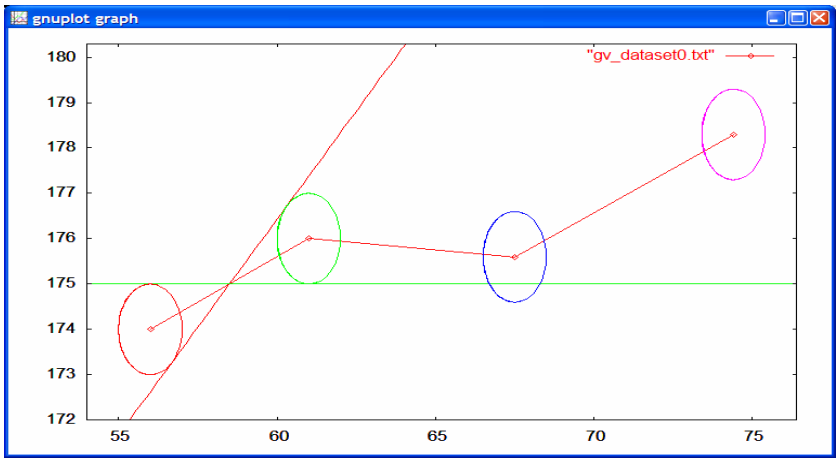


Line Stabbing Overview(3)

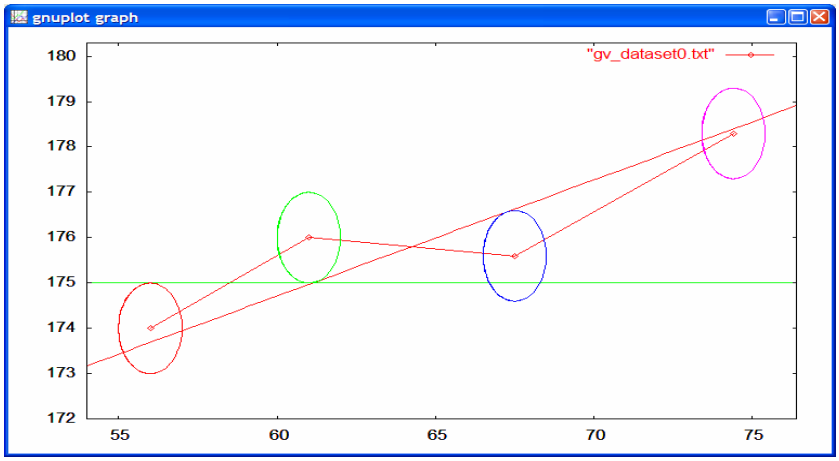
Basic intuition:

- Iterative algorithm, examines one new circle at each step
- For each new circle, compute the inner tangents which will intersect every circle examined so far (stop if impossible)
- Any line that is placed between the inner tangents can be said to "stab" the circles

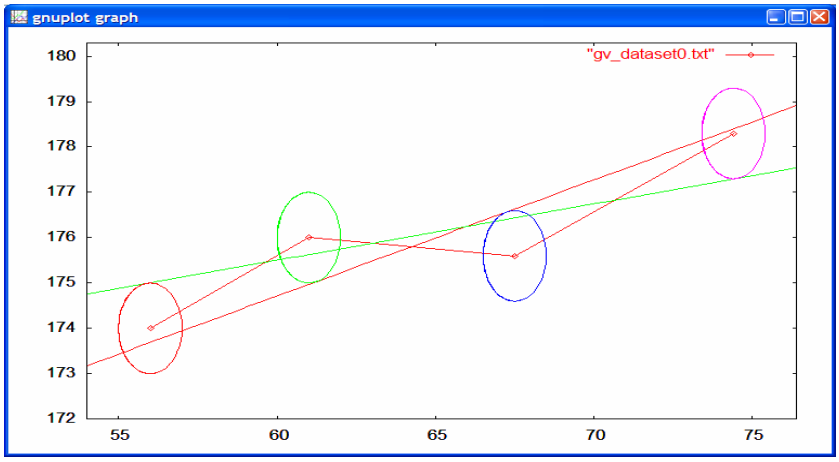
Line Stabbing Overview(4)



Line Stabbing Overview(5)



Line Stabbing Overview(6)

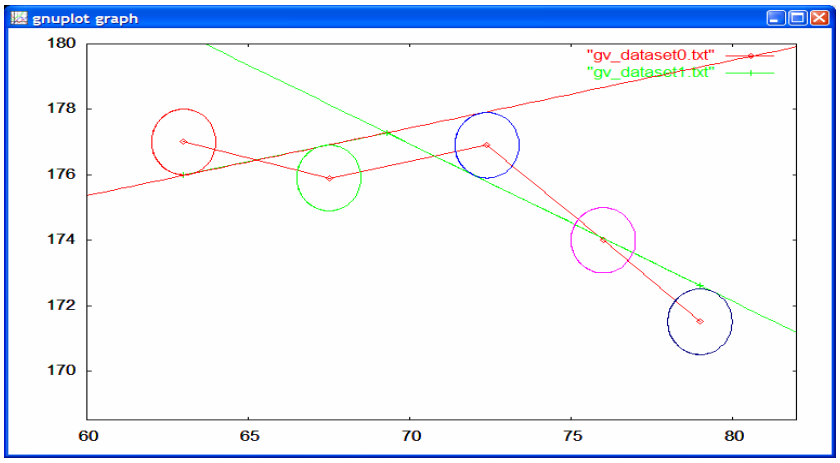


Weak Simplification Overview(1)

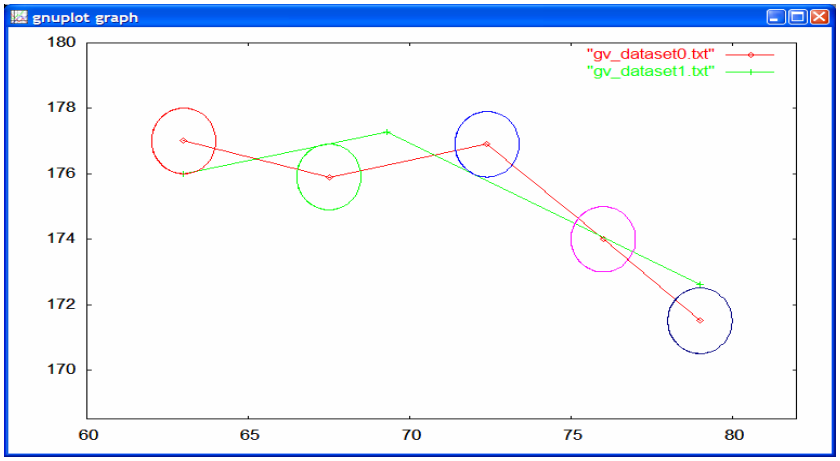
Basic intuition:

- Run the line stabbing algorithm as many times as necessary
- Just choose one of the inner tangents as the line stabber
- Take the intersection of the line stabbers as the vertices of the simplified trajectory

Weak Simplification Overview(2)



Weak Simplification Overview(3)



Intermediate Summary

- What we have done so far
  - Generated weak simplified trajectories for test cases having only disjoint circles (vertices > ε apart).
- What remains to be done:
  - Handle cases having overlapping circles.
  - Design experiments and run them
    - Evaluate average savings over entire set of trajectories as function of epsilon (ε)
    - Report on relationship between epsilon and overlapping
  - Submit a final report along with all source code

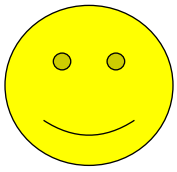
References

[1] Guibas, L.J., Hershberger, J.E., Mitchell, J.S. and Snoeyink, J.S. Approximating Polygons and Subdivisions with Minimum-link Paths. *International Journal of Computational Geometry & Applications*, 3: 383-415, 1993.

The End

Questions or comments?

Thank you!



Have a nice day!