Julius-Maximilians-
UNIVERSITÄT WÜRZBURG

EFFICIENT POINT CLOUD COLLISION DETECTION AND ANALYSIS IN A TUNNEL ENVIRONMENT USING KINEMATIC LASER SCANNING AND K-D TREE SEARCH

Johannes Schauer, Andreas Nüchter

Objective & Challenge

- The minimum clearance outline or structure gauge has an important place in the planning of railroad tracks.
- Unfortunately the structure gauge is an unsatisfying measure within turns of the tracks model moved along a trajectory
- The goal is to determine which points of the environment collide with the model on its path, given a certain safety margin and how deep any colliding points penetrate the model.

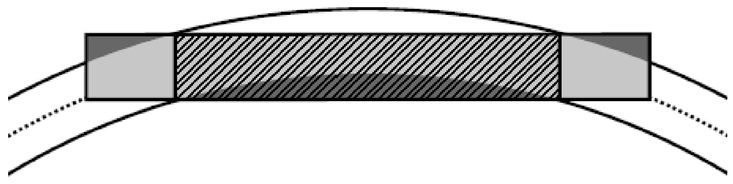
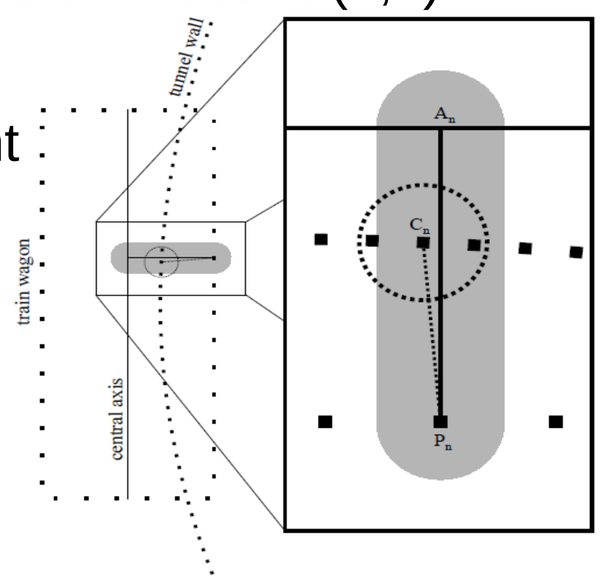


Figure: Top view of the train wagon (in dark and light gray) and its curved loading gauge as it passes through a turn. The dark gray areas mark the volumes of the train wagon outside of its loading gauge. The striped volume indicates the volume of the train wagon between its two bogies. The dotted line indicates the wagon's trajectory.

Approach

- Efficient *k*-d tree for Nearest Neighbor Search in **3DTK – The 3D Toolkit** (1,2)
- Heuristic for protruding objects
- `segmentSearch_1NearestPoint` closest point to a line segment
- Cf. Fig.: The gray area is the segment search volume. The dotted line is the distance between P and C which is the point that was found to be closest to P within the search area. The dotted circle shows the search radius around C. All points within this radius are updated with the the same distance that C has to P if that distance is greater than the previously stored one.



Results

- Reliable and fast calculations
- Optech Lynx Mobile Mapper data
- 3D point cloud of a train wagon (VZ-400)

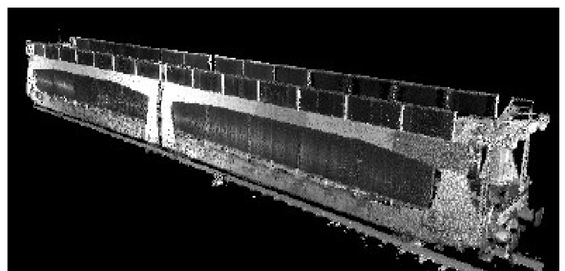
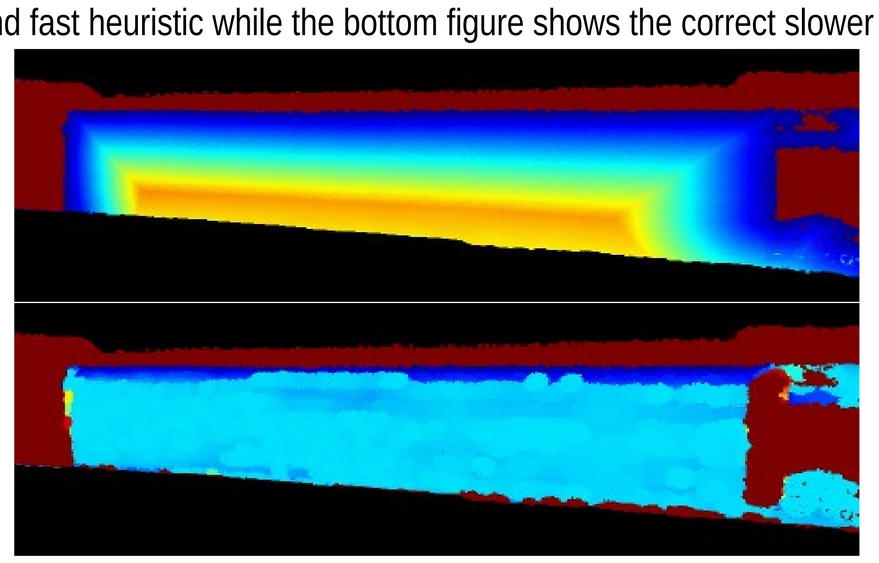


Figure: A comparison of the penetration depth as calculated by the two variants. Both figures show a narrow piece of tunnel from the outside with the calculated penetration depth indicated by the point color. Non-colliding points are shown in dark red. The top figure shows the simple and fast heuristic while the bottom figure shows the correct slower variant.



- Benchmark for computations:
  - 18,92 mill. points from mobile mapping
  - 19,392 x 28,622 = 555,037,824 *k*-d tree searches require 77 seconds.

Future Work

- Research which checks to abort the *k*-d tree traversal for different search geometries and input data perform best
- Employ OpenMP for multiple cores
- Present a GPU based method, based on (3)

References

(1) 3DTK – The 3D Toolkit, `http://threedtk.de`

(2) Elseberg, J., Magnenat, S., Siegwart, R. and Nüchter, A., 2012. Comparison of Nearest-Neighbor-Search Strategies and Implementations for Efficient Shape Registration. Journal of Software Engineering for Robotics (JOSER) 3(1), pp. 2 – 12.

(3) Qiu Deyuan, May S. May, and Nüchter, A . GPU-accelerated Nearest Neighbor Search for 3D Registration. In Proc. of the 7th International Conference on Computer Vision Systems (ICVS '09). LNCS 5815, pages 194-203, Liege, Belgium, 2009.

Contact: Prof. Dr. Andreas Nüchter **andreas@nuechti.de**