

PointMan: A Pointcloud Renderer

Nathan Marshak, Uriah Baalke

November 18, 2013

1 Motivation

Data from 3D range scanners is often represented as a point cloud. Point clouds are used in robotics, geospatial visualization, architecture, and industrial CAD [1]. While point clouds are often converted into a triangular mesh or NURBS, it is often desirable to render raw point data. This is due to several reasons. First, reconstruction algorithms take time, and are notoriously tricky to work with. Second, data sets (especially outdoor data) often must be cleaned by hand before a mesh can be obtained. Cleaning a large data set by hand may require several man-months of time [2]. Finally, a point representation does not require extra memory to store mesh topology information [3]. See full list of point cloud papers and sites that we found by clicking on this link.

2 Project Description

Implement EITHER a point cloud renderer on top of WebGL, OR a from-scratch, alternative rendering pipeline in CUDA. Implement the following features:

1. Load real range scan / LiDAR data files.
2. Fill in the gaps between points with splatting (AKA surfels).
3. Support for color (this allows for “texture”).
4. Implement an optimization to allow for rendering of LARGE point sets. Common optimizations are out-of-core storage [4], and octrees [2]. One very complicated scheme in [5] uses JavaScript for the user interface and camera control, while offloading more expensive operations to the GPU. We would probably want to do an octree, then if there is time, extend it to be an out-of-core octree.

References

- [1] “Making point clouds a fundamental data type in bentley software.” <http://www.bentley.com/en-US/Promo/Point+Clouds/>, Nov. 2013.
- [2] M. Wimmer and C. Scheiblauer, “Instant points: Fast rendering of unprocessed point clouds,” in *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG’06, (Aire-la-Ville, Switzerland, Switzerland), pp. 129–137, Eurographics Association, 2006.
- [3] H. Pfister, M. Zwicker, J. van Baar, and M. Gross, “Surfels: Surface elements as rendering primitives,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, (New York, NY, USA), pp. 335–342, ACM Press/Addison-Wesley Publishing Co., 2000.
- [4] M. Botsch, R. P. (editors, A. Berner, M. Bokeloh, A. Fleck, M. Hoffmann, P. Jenke, B. Maier, D. Staneker, and A. Schilling, “Interactive editing of large point clouds,” 2007.
- [5] “hidof large pointclouds.” <http://hidof.com/project/large-pointclouds/>, Nov. 2013.