1 Objective

1.1 Context

The Intel RealSense 2 can capture colour point clouds in .ply and .bag files. The goal is to use this data to reconstruct high quality models in spite of the noisy input data.

Poisson surface reconstruction [1] may be suited for quality offline reconstruction.

1.2 Related work

Radial basis function reconstruction [2] is another technique for point cloud reconstruction.

Another class of techniques use Voronoi diagrams [3] and similar graph construction-based techniques.

1.3 Selected approach

Poisson surface reconstruction uses the gradient of the normal field to solve for an indicator function. The indicator function can then be used to extract a surface.

2 Methodology

I frankly have no idea how Poisson reconstruction works yet.

3 Implementation

3.1 Programming Languages/Libraries/Data Structures

- Intel RealSense SDK
- Point Cloud Library, or possibly
- CGAL

3.2 Data sets

- The Stanford Bunny
- Data acquired from the RGB-D Camera

4 Timeline

So far, I've toyed with a few examples bundled with the RealSense SDK. I am able to take snapshots and "depth videos" using the camera.

1. Study the paper on Poisson Reconstruction and the provided implementation (2-3 days)

2. Come up with a way to try to improve accuracy – either by eliminating noise or detecting and preserving features. Possibly use multiple input frames to smooth readings, or make use of colour information. (1-2 days)

3. Implement and test (2 weeks)

References

[1] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[2] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 67–76.

[3] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," 1998.