Textured Mesh Reconstruction of Indoor Environments Using RGB-D Camera

Collin Boots

A THESIS

in

Robotics

Presented to the Faculties of University of Pennsylvania in Partial

Fulfillment of the Requirements for the Degree of Master of Science in Engineering

2014

Dr. Daniel D. Lee
Supervisor of Thesis

Dr. Camillo J. Taylor
Graduate Group Chairperson

Abstract

Your abstract goes here... ...

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

As robots continue to be incorporated into human environments, the need for intelligent and high-speed reasoning about the objects around them increases dramatically. At the simplest level, mobile robots need to create a map of their environment for navigation. At a higher level, some robots need to recognize distinct objects in their environment, track object movement, and have some intuitive sense of object geometry that is easily stored and processed. Even more importantly, robots must be able to efficiently generate adaptable models of their environment, or worldview, from sensor data in real time. Many different methods for representing the world have been proposed and implemented, and they will be discussed below in more detail. Like the human brain, the robot should also be able to perform these low level functions with only minimal intervention from higher cognitive functions. Such technology also has potential uses beyond robotics. Applications may include easily modeling indoor environments for interior design concepts, generating 3D tours or maps for various buildings, or low resolution rough mapping of archaeological excavations.

This thesis is intended to work towards such a system based on RGB-D cameras, triangle meshes, and the powerful parallel computing capability modern Graphics Processing Units (GPUs) offer. RGB-D cameras like Microsoft's Kinect provide a great low cost solution for capturing 3D environments. Triangle meshes are efficient to store, simple to manipulate and refine, and very versatile. Meshes have the added benefit of being well suited to GPU hardware (which was originally designed for just that purpose).

1.2 Related Work

A great variety of methods have been applied to RGB-D camera data in an effort to construct a coherent worldview. Each has advantages and disadvantages.

1.2.1 Worldview Storage Models

Point Cloud Models The simplest approach to storing RGB-D data is as a raw point cloud. Each point is stored in a self contained data structure containing at least the point's position and color information. E.g.

$$P_i = \{pos_x, pos_y, pos_z, red, green, blue\}$$

This approach allows a complete record of the raw data to be stored very easily, but the size of the data stored will grow very quickly, and simply storing the data linearly results in very slow queries.

Nearest neighbor approximations have recently become a popular approach for speeding queries on large point clouds[9]. However, achieving reliably fast queries usually requires some form of hierarchical tree structure like K-d trees[11] or octrees[15]. K-d trees are much more adaptable and usually more efficient in terms of memory storage, but octrees have a significant advantage when it comes to incrementally building a point cloud because points can very easily be inserted into the appropriate octree leaf node with no duplication or restructuring. K-d trees are better suited for compressing point clouds offline.

Voxel Space Models Perhaps the most robust and impressive real-time surface reconstruction algorithm to date is Microsoft's Kinect Fusion[10, 4]. An open source implementation called KinFu is also available[12]. Kinect Fusion uses a bounded 3D voxel space where each voxel stores the distance to the nearest detected surface or empty space (the default). Figure 1.1 shows the workflow of the Kinect Fusion system. The point cloud generated by each RGB-D frame is projected into the voxel space and each point updates nearby voxels' distance to nearest surface metric. Implicit surfaces can then be rendered by raycasting through the voxel space and detecting the distance sign crossover point. This results in very high resolution and fidelity reconstructions of the implicit surfaces in the environment. The primary disadvantage of this approach is the workspace size is limited by memory and compute resources which scale with the resolution and dimensions of the voxel space.

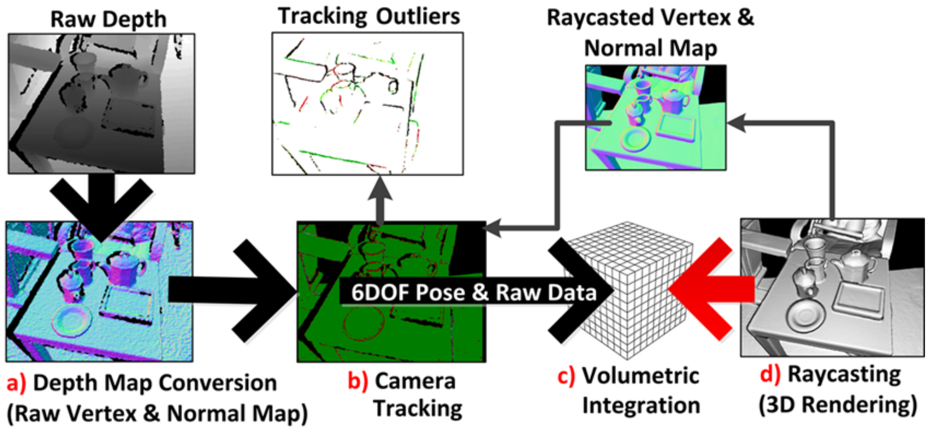
Raw Depth
Tracking Outliers
Raycasted Vertex & Normal Map
6DOF Pose & Raw Data
a) Depth Map Conversion (Raw Vertex & Normal Map)
b) Camera Tracking
c) Volumetric Integration
d) Raycasting (3D Rendering)

Figure 1.1: Kinect Fusion Tracking and Reconstruction Pipeline. Reproduced from[4]

A spacial extension of this system called Kintinuous was introduced in 2012[14, 1]. Kintinuous uses a mobile voxel space which periodically relocates based on camera motion. Voxels that move outside the space are converted to a triangular mesh using a greedy mesh triangulation procedure described by Marton *et al.*[8].

Others have attempted to use sparse voxel octrees (SVOs) to represent larger unbounded spaces using voxels of varying resolution[6, 13]. This approach is very amenable to human environments like buildings which are dominated by free space. However, all of these approaches are still limited to storing discrete data points and contain no inherent information about coherent objects or regions.

Point Cloud Offline Processing Point clouds can also be processed offline to extract surfaces and objects. These systems are not burdened by the strict requirements of real-time computation so they generally can produce more globally optimal solutions than their online counterparts. These systems generally target unordered point clouds. Marton *et al.* proposed an incremental approach to triangulation of noisy point clouds[8]. The approach was amenable to introduction of new registered RGB-D frames by only triangulating points that did not correlate with existing mesh models, but the implementation was far too slow for real-time applications. Ma and Whelan introduced a system for planar simplification of dense point clouds using a QuadTree based algorithm[7] and texture maps. This thesis will rely heavily on this innovation. Other approaches worked towards implicit surfaces like parameterized smooth surface fits [3] or Poisson surfaces[5, 2].

1.2.2 Real-Time Processing

1.2.3 Additional Resources

Although this thesis focuses on detecting and representing only planar elements, the envisioned worldview generating pipeline would need several additional components and capabilities to be successful. Additionally, many data processing technologies exist that can improve the quality of the input data through more sophisticated filtering.

RGB-D Filtering

3D SLAM

Mesh Processing and Modification

1.3 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 will review some basic precepts of parallel algorithm design, along with specific optimization considerations for programming GPUs with CUDA. Chapter 3 will provide an overview of the system design as well as break down the pipeline into sub-modules to be explored in much more detail in Chapter 4. Finally, Chapters 5 and 6 will provide performance analysis, conclusions, and areas of potential improvement for future work.

TODO: Double Check that this hasn't changed

Chapter 2

Parallel Programming Paradigms

2.1 Principles of Parallel Programming

2.2 Parallel Algorithm Building Blocks

2.3 Programming with CUDA

2.3.1 CUDA GPU Architecture

2.3.2 Optimizing CUDA Code

Chapter 3

Problem Formulation and Approach

3.1 Problem Specification

3.2 High Level System Design

3.3 Plane Detection and Meshing Pipeline Design

Chapter 4

Implementation

4.1 RGBD Framework

4.2 Preprocessing

4.3 Plane Segmentation

4.4 Mesh Generation

Chapter 5

Performance Analysis

Chapter 6

Conclusions and Future Work

Bibliography

[1] *Robust real-time visual odometry for dense RGB-D mapping*, May 2013.

[2] Matthew Bolitho, Michael Kazhdan, Randal Burns, and Hugues Hoppe. Parallel poisson surface reconstruction. In *Advances in Visual Computing*, pages 678–689. Springer, 2009.

[3] Kai Hormann. From scattered samples to smooth surfaces. *Proc. of Geometric Modeling and Computer Graphics*, 2003.

[4] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[5] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[6] S. Laine and T. Karras. Efficient sparse voxel octrees. *Visualization and Computer Graphics, IEEE Transactions on*, 17(8):1048–1059, Aug 2011.

[7] Lingni Ma, Thomas Whelan, Egor Bondarev, Peter HN de With, and John McDonald. Planar simplification and texturing of dense point cloud maps. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 164–171. IEEE, 2013.

[8] Z.C. Marton, R.B. Rusu, and M. Beetz. On fast surface reconstruction methods for large and noisy point clouds. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3218–3223, May 2009.

[9] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP (1)*, pages 331–340, 2009.

[10] Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[11] A. Nüchter, K. Lingemann, and J. Hertzberg. Cached k-d tree search for icp algorithms. In *Proceedings of the 6th IEEE international Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM '07)*, pages 419–426, August 2007.

[12] Michele Pirovano. Kinfu – an open source implementation of kinect fusion. http://homes.di.unimi.it/ pirovano/pdf/3d-scanning-pcl.pdf, 2012. PhD student in Computer Science at POLIMI.

[13] Julian Ryde and Jason J Corso. Fast voxel maps with counting bloom filters. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4413–4418. IEEE, 2012.

[14] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. 2012.

[15] K. M. Wurm, A. Hornung, M Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, USA, May 2010.