Data handling, Interpretation and Visualization for Static and Mobile Terrestrial LiDAR; Qualifying the Problems, Solutions, and Opportunities.

Michael Martin[1], Gregory McQuat[2], and Rob Harrap[3]

1 Geography, Queen's University, Kingston, Ontario, 4mm39@queensu.ca
2 Geography, Queen's University, Kingston, Ontario, 9gmj@queensu.ca
3 Geography, Queen's University, Kingston, Ontario, harrap@geol.queensu.ca

Abstract

LiDAR (Light Detection and Ranging) technology allows for very rapid acquisition of accurate 3d spatial data. All aspects of LiDAR, from data collection to the dissemination of resulting models, are on the cutting edge of geographic information science. LiDAR data is being used across a wide range of academic disciplines from civil engineering to urban planning to GIScience. We focus on urban applications of LiDAR and in particular on effective ways to bring LiDAR data into existing spatial information environments and the subsequent use of LiDAR data with both available and new tools. The research presented here focuses on what can currently be completed using the existing tools and looks at software created at Queen's University to view small LiDAR data sets in a freely distributable engine. This software focuses on a video game approach to LiDAR software programming, providing a visualization environment and algorithmic interface for research.

Background and Relevance

Currently, urban LiDAR scanning methods are in their infancy. While LiDAR technology was first used for bathymetric scanning in the 1960's (Hickman and Hogg 1969) and on airborne systems in the mid 1980's (Krabill et. al. 1984), urban applications of the technology are just beginning to emerge; new sensors, sensor platforms, and tools for using the resulting data sets are under development by many research groups both within the GI community and within the computer science community. As part of research on high resolution urban models for simulation, researchers at Queen's have collected high resolution urban LiDAR data using the Terrapoint TITAN mobile terrestrial LiDAR system. We collected a dense point based representation of the Kingston city centre. These point clouds were found to be too large for traditional software such as commercial GIS software to handle, and even in specialized software tools for LIDAR processing, pushed the capacities of desktop computers to their limits.

Using software such as Lieca Cyclone in conjunction with Google Sketchup, we were able to produce building and streetscape models from urban LiDAR data, however, this general workflow was time consuming and the resulting models lost some of the accuracy that the LiDAR system provided. Workflow testing led to the conclusion that no current consumer level software could efficiently deal with the immense point clouds generated by systems such as TITAN, and these systems furthermore do not allow the testing of custom algorithms or high quality visualization. Furthermore, these tools are quite expensive, so that distributing LiDAR data to interested parties is impossible. As a result a software project was started to address the problems with LiDAR data processing. The goals of this software development was to create a system that would allow for efficient visualization and algorithm development. In the first phase of this project, described here, we focused on the development of a lightweight, distributable test environment for visualization and algorithm testing.

Methods and Data

The data used in this study was collected using a mobile LiDAR scanner, the Terrapoint TITAN mobile scanning system. TITAN uses precision GPS and an Inertial Movement Unit coupled to 4 Riegl scanners to allow mobile scanning while driving at normal flow-of-traffic speeds. TITAN can scan large areas by driving streets without special requirements for surveying, at up to 5cm accuracy. This scanner has been used in an urban context to collect 3d data for representative sections of Kingston, Ontario, Canada.

The delivered data that Queen's University is using is approximately 4Gb in size, divided into 100-250Mb data tiles. These data tiles can contain up to 30 million data points and as a result the data was subset in order to allow for efficient computation. The subsets of data were approximately one block in size, and contained up to 5 million points. While high-end LiDAR tools are capable of dealing with such data volumes, they are not redistributable and preclude testing of custom algorithms.

In this study, the software development took an approach from the video game community. Using this paradigm the software would be optimized for performance over accuracy, where render speed is paramount. While video game software development can be done in many computer programming languages, this study utilizes the C# language and the Microsoft XNA platform. XNA builds upon Microsoft's Dot Net Framework, a codebase used for its ability to rapidly develop programs. Microsoft XNA provides the programmer with a standard code base to work from, specifically designed to create games for desktop computers and the Xbox and Xbox360 gaming consoles. The software development environment used for this study was Microsoft Visual Studio, a powerful and intuitive program designed for working with the Dot Net Framework.

Once work on the basic software visualization environment was completed, the development moved to encompass an interface for programmatically interacting with the point clouds and to development of algorithms which could find objects within the clouds, such as building walls using computer vision methods.

Results

The program created in this project is able to efficiently view and move LiDAR point clouds. It can load 5 million data points, and manipulate them at 24 rendered frames per second, along highly interactive manipulation of clouds (Figure 1). The visualization is from the first person perspective and allows a user to 'fly' around the data, inspecting it however they wish. Built into the software is an interface to interact with the data and sample algorithms which utilize this interface.

The interface developed for interaction with the data allows researchers to develop their own custom algorithms, and to visually animate the progress of the algorithm. The interface exposes the X, Y, Z and intensity values that the LiDAR data reports and allows access to draw functions in the application window.

The algorithms created in this study focus in the detection of building walls. The algorithms focused on a marching cubes and collision technique. However, these algorithms are still currently being refined.

The visualization system and algorithms are not without certain limitations. The visualization environment is limited by the amount of points that it can display at any given time, by 10 million data points the rendered frame rate drops to 6 frames per second. The algorithms developed suffer from changes in typography and the orientation of UTM grid layout in Kingston.

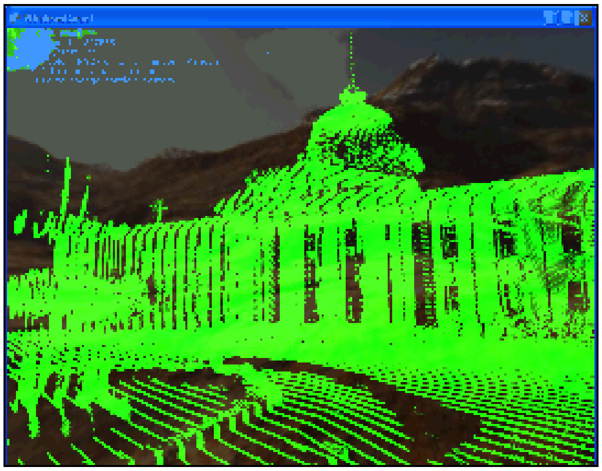

Figure 1. Kingston City Courthouse.

Conclusions

This research contributes to the field of urban geomatics research by addressing the limitations of consumer level software for LiDAR, and providing a new tool for visualization and algorithm development. Specifically, this tool emphasizes interactive performance over data volume handling.

The development approach used in this study has been directly adapted from best practices in the game development community. The result allowed a first person perspective environment to run significantly large data clouds through algorithms at fluent frame rates.

The visualization and research interface software developed in this project proves the validity of using the game technology approach and shows that consumer level computers can scientifically interact with LiDAR data. Although at the current stage of development the tool has limited abilities, the possibilities that can result from a further developed project are exciting. Further development would allow researchers and consumers to be able to use a full complement of visualization techniques and vectorization methods allowing the MTL data to be converted into meaningful, semantically complete, virtual spaces.

References

Google Sketchup. http://sketchup.google.com

Hickman GD, Hogg JE. Application of an airborne pulsed laser for near shore bathymetric measurements. Remote Sensing of Environment. 1969. vol. 1, pp. 47–58.

Krabill WB, Collins JG, Link LE, Swift RN, Butler ML. Airborne laser topographic mapping results. Photogrammetric Engineering and Remote Sensing. 1984. vol 50, 685–694.

Leica Geosystems Cyclone. www.leica-geosystems.com

Microsoft Visual Studio 2008. http://www.microsoft.com/visualstudio

Microsoft XNA. http://www.xna.com

Riegl Laser Measurement Systems. http://www.riegl.com

Terrapoint Inc. TITAN Mobile Terrestrial LiDAR Scanner. http://www.terrapoint.com.