Real-Time LiDAR Terrain Mapping and Analysis

Galen Cochrane
Undergraduate Researcher

Marko Sterbentz
Undergraduate Researcher

John Edwards, Ph.D.
Research Advisor

Objective

Develop an application for real-time LiDAR terrain mapping and analysis.

Capturing LiDAR Data

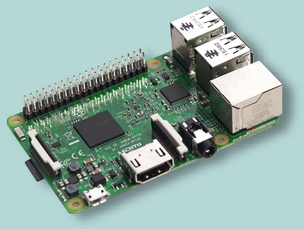
What is LiDAR?

LiDAR is an acronym for Light Detection and Ranging. By shooting an array of lasers hundreds of times a second, it is possible to measure and record distance data. This information can be converted into a cloud of point data that represents an accurate three dimensional view of the immediate environment.

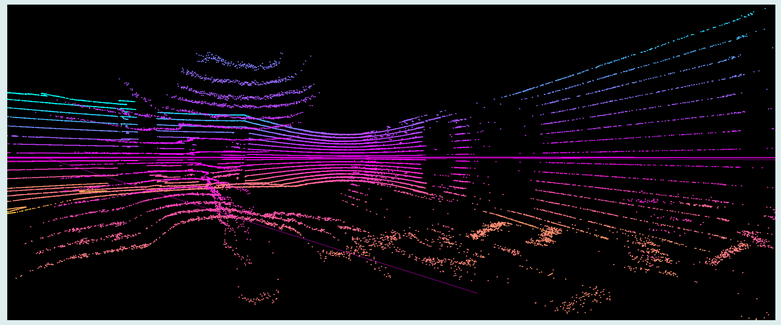
Typical Static Data Capture

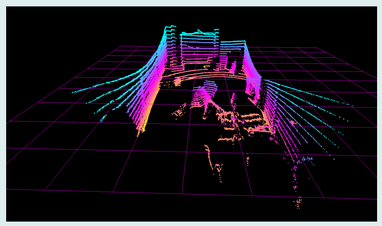
An Unmanned Aerial Vehicle (UAV) can be equipped with a LiDAR sensor to capture a three dimensional image of its surroundings. Such an image is composed of many thousands of sample points which may be computationally analyzed to gain information about the environment they represent. For example, given a LiDAR image of a landscape, it is possible to distinguish plants from rocks, or to highlight areas at high risk of landslides. Using existing approaches, performing such analyses generally requires that the LiDAR image first be captured by the UAV, and then later processed after the UAV has returned to the ground. This workflow is impractical because if any part of the gathered data is incomplete or otherwise unsatisfactory, the flight must be repeated.

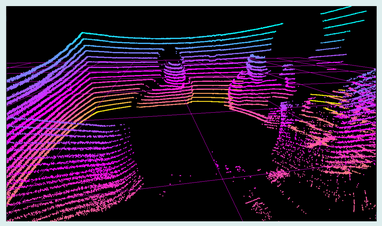
Real-Time Capture and Analysis



This project aims to develop software for the real-time reception and analysis of data gathered by a LiDAR-equipped UAV. The user can view and analyze the incoming data as the UAV is in flight, which enables the flight path to be corrected accordingly. The user will be able to see an interactive model of the landscape with selected analyses applied on demand. This software will provide the capability to easily map the environment, which will facilitate the improvement of many ecosystem services.







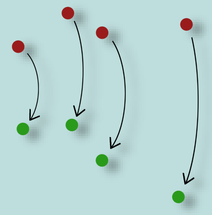
Retrieving and Processing Data

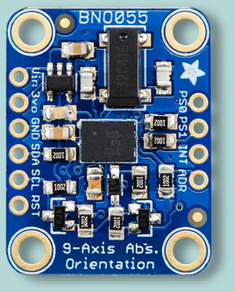
Interfacing with the Velodyne

The Velodyne VLP-16 transmits packets of data containing an angle and a distance for each sample over an Ethernet connection. From there, each sample is converted into Cartesian coordinates for use in constructing a three dimensional representation of the surrounding environment.

Processing Data with Iterative Closest Point

Since the Velodyne is continuously moving, it is necessary to align incoming point clouds to a unified coordinate space. This is accomplished with the Iterative Closest Point (ICP) algorithm, which compares two point clouds in order to produce a transformation that moves the second cloud into the space of the first. Each new point cloud is processed using ICP.





Interfacing with the Inertial Measurement Unit (IMU)

ICP alone produces good results for a short duration, but eventually error propagates. To correct for this drift, an Inertial Measurement Unit (IMU) provides acceleration and orientation data which, combined with ICP and a GPS, ought to alleviate the issue.

Visualizing the Data

Creating a Graphical Interface

Once all of the points exist in a single space, they must be displayed for the user to see. The Open Graphics Library (OpenGL) is used to provide rendering functionality. A 3D navigation interface is available so the user can 'move around' in the virtual environment using a mouse and keyboard. The data shown to the left is colored using height measurements, but more sophisticated processes are available.

Conclusion

Results

As a result of the research and development, a functional and standalone application was created. This software is capable of interfacing with the Velodyne VLP-16 and reading the data that is being transmitted over the wire. It then analyzes it and constructs a point cloud of the surroundings. As the sensor continues to read in new data, the iterative closest point algorithm is run in order to correctly orient the new data within the three-dimensional space defined by the preceding data.

Future Applications and Improvements

In order to achieve higher quality data sets, improvements will be made to the implementation of the ICP algorithm, and new data analysis tools will be added in order to enhance the capabilities of the software. Additionally, the ICP algorithm will be written for the graphics processing unit (GPU) in order to parallelize and speed up the processing.

References

[1] *VLP-16 User's Manual and Programming Guide*, 1st ed. Morgan, CA: Velodyne LiDAR, Inc., 2016, pp. 1 - 45.

[2] P. Besl and H. McKay, "A method for registration of 3-D shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

Acknowledgements

This publication was made possible by the NSF Idaho EPSCoR Program and by the National Science Foundation under award number IIA-1301792.

Contact Information

Galen Cochrane: cochgale@isu.edu
Marko Sterbentz: stermark@isu.edu
John Edwards, Ph.D.: edwajohn@isu.edu

Idaho State University MILES IDAHO EPSCoR