Collecting and Processing Point Cloud Data for Snow Depth and Density Measurement

Fletcher Wadsworth
Ethan Gutmann, Agbeli Ameko
SIParCS

July 26, 2022

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Introduction

Project: Integrating a LiDAR Sensor into a low-cost Weather Station

Purpose: Collect and process point cloud data into useful snow measurements under
computational and storage constraints.

Approach: Parallel collection and analysis of point cloud measurements using Raspberry Pi.

Background - Snow Measurement

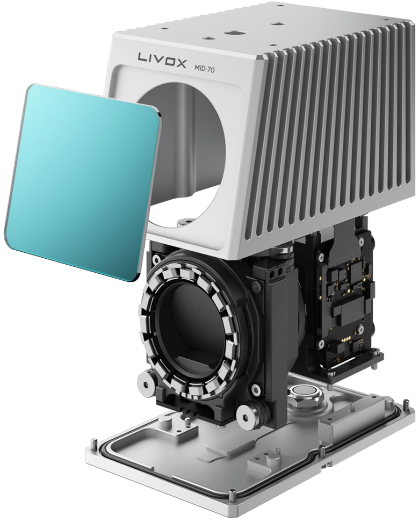
- **Measuring snow is important**
  - Climate modeling/forecasting
  - Water management

- **Measuring snow is difficult**
  - Poorly measured
  - Expensive

- **Our goal is measuring snow with high precision at low cost.**

Background - Hardware

- Livox Mid-70 3D automotive LiDAR sensor
- 100k or 200k measurements per second
- $1099



1

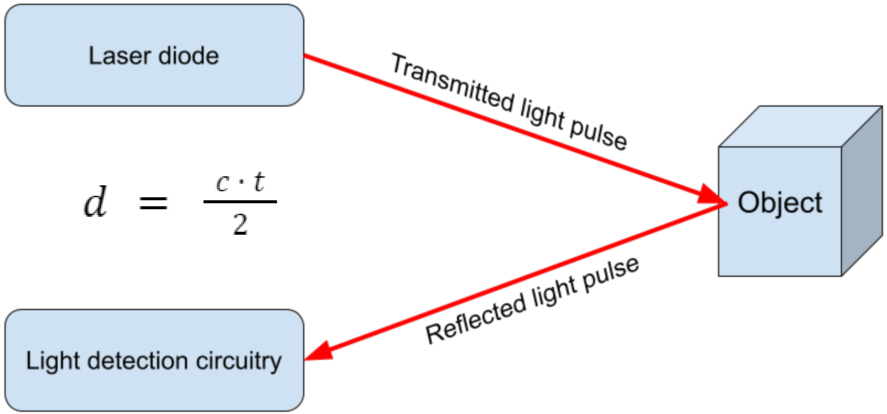
- Raspberry Pi 3 model B+ single-board computer
- SoC 64-bit quad-core processor at 1 GHz
- $30-$40

2

1: https://www.livoxtech.com/mid -70

2: https://www.raspberrypi.com/products/raspberry -pi-3-model-b-plus/

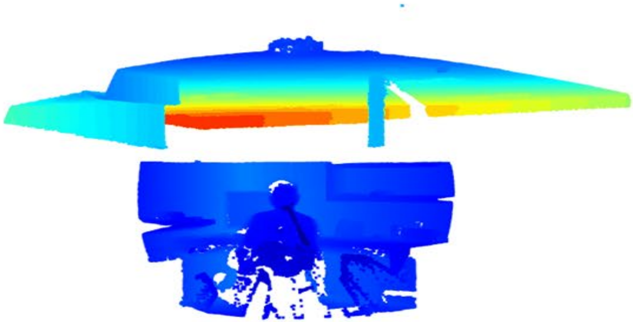
Background - LiDARs and Point Clouds



$$d = \frac{c \cdot t}{2}$$

- Unstructured data array containing x,y,z coordinates, reflectivity, timestamp, and status code
- Ordered (mostly) in the sequence of capture determined by sensor manufacturer

Background - LiDARs and Point Clouds



Example visualization of point cloud

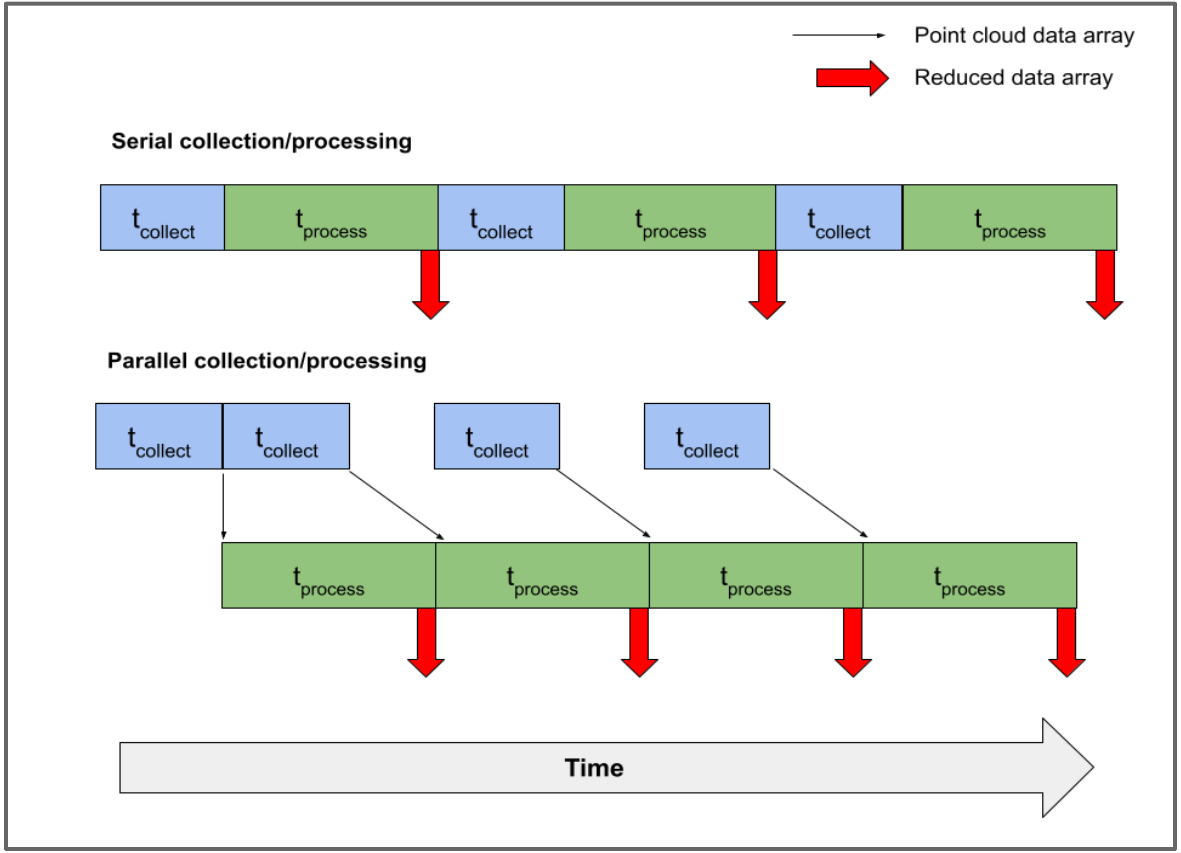
Methods

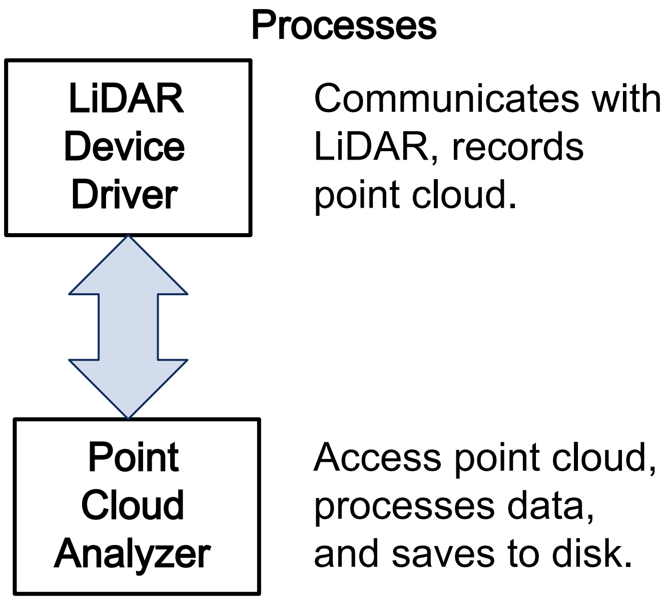
- **Point cloud collection with LiDAR**
  - Update existing device driver to keep point cloud data in memory for processing.
- **Point cloud analysis**
  - Preliminary point cloud processing algorithms to estimate snowpack depth on the ground and snowflake density in the air.
  - Extract key measurements from large data in the interest of disk storage savings.
- **Interplay**
  - Process level parallelism to allow simultaneous data collection and processing/downsizing.

Methods - Why Process in Real -Time?

- Each point requires 12 bytes - one 4 byte integer for each coordinate.
- At collection rate of 100,000 points/sec, it takes **14.91 minutes to fill 1 GB**.
- Intended for deployment in remote locations
- Strategy: collect and process point clouds simultaneously

Methods - Multiprocessing





Results - Data Storage

For a recording session of 3 point clouds of 3 seconds each:

	Raw Data File Size	Ground Height Array Size	In Air Point Cloud Size	**Total Storage**
Without processing	12.8 MB	X	X	**12.8 MB**
With processing	X	1.88 MB	2.86 MB	**4.74 MB**

Only 37% of storage is required and most information is retained!

However, this is an arbitrary benchmark.

Methods - Snowpack Height Measurement

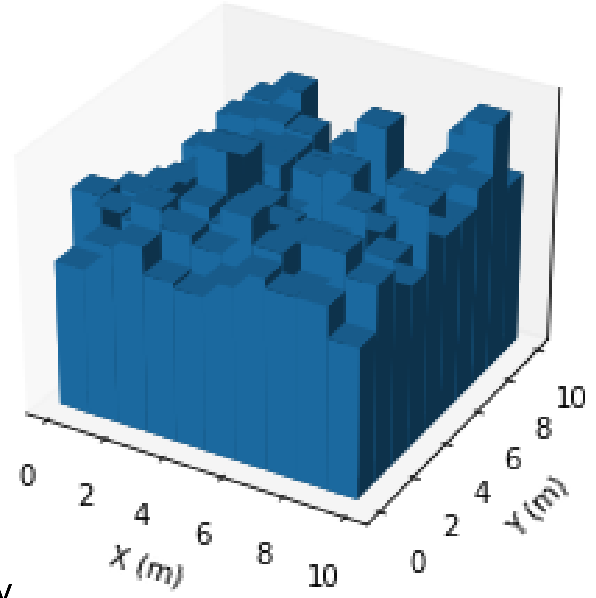
Motivation:

Measure the volume of snowpack over a specified ground area.

Algorithm:

1. First separate points on the ground from points in the air.
    - For each bin, find point with lowest z coordinate value

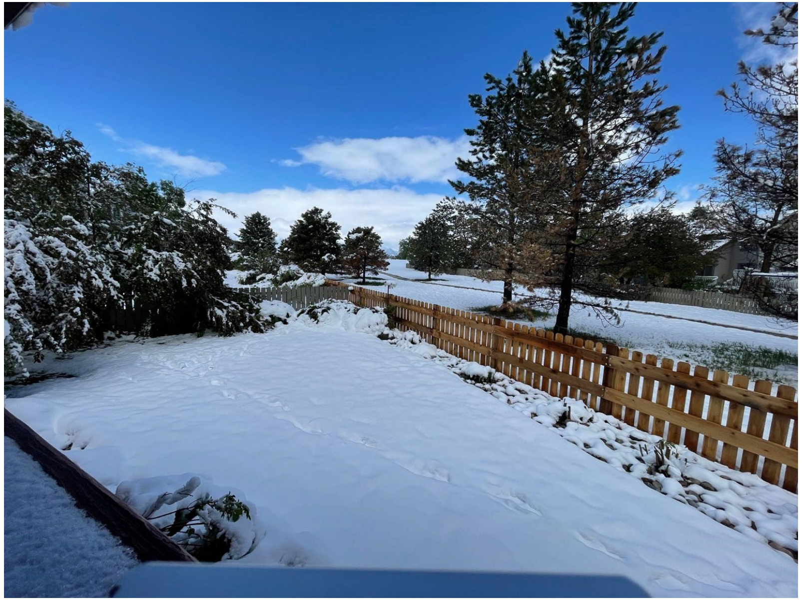
1. Approximate height of ground in each bin
    - Take average of all points within a threshold of minimum value.



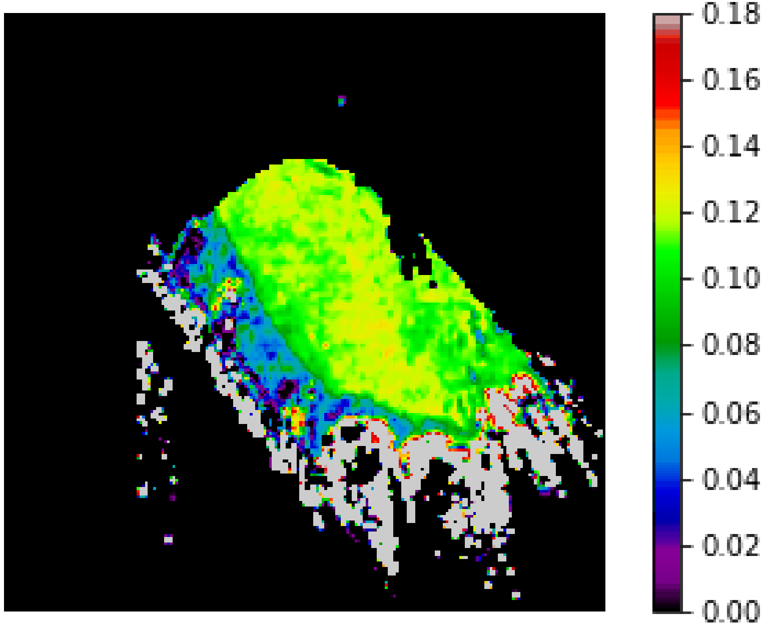
Output:

A 2D array of ground height values in equally size bins in the x,y plane.

Results - Snowpack Height Measurement



LiDAR test environment



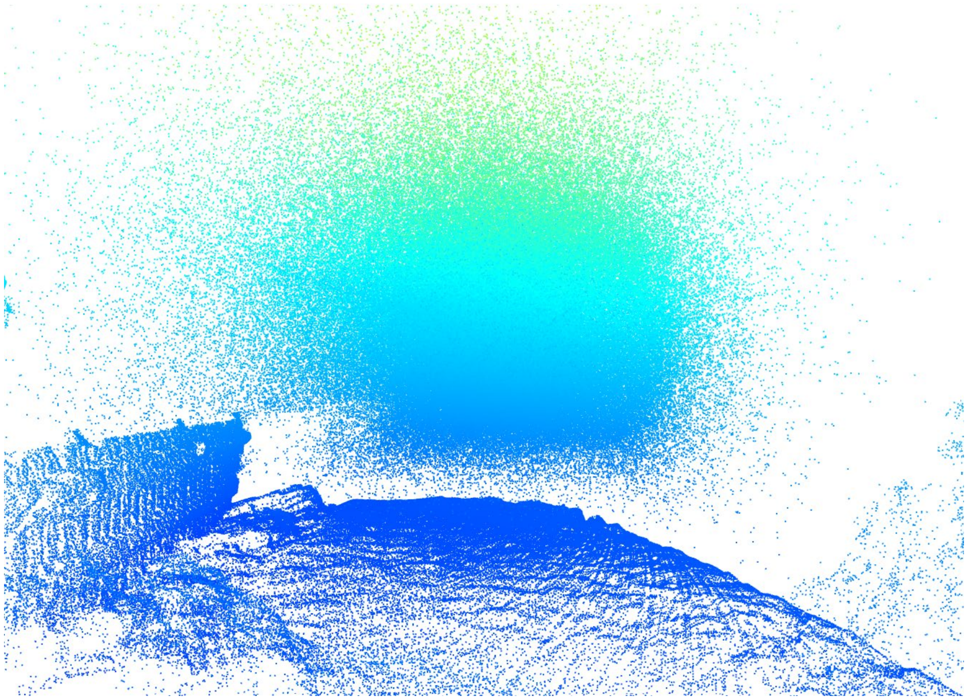
Snowpack height map over time series of point clouds

Methods - Snowflake Density

- Extracting useful observations of snow in the air from point cloud data proves more difficult.

- **Use a 3D histogram to visualize the density of points in the air.**

- Visualization may help lend insight into limitations of LiDAR and possible measurements to investigate.



Methods - Snowflake Density

Motivation:

Calculate the density of snowflakes in the air as a 3D histogram

Algorithm:

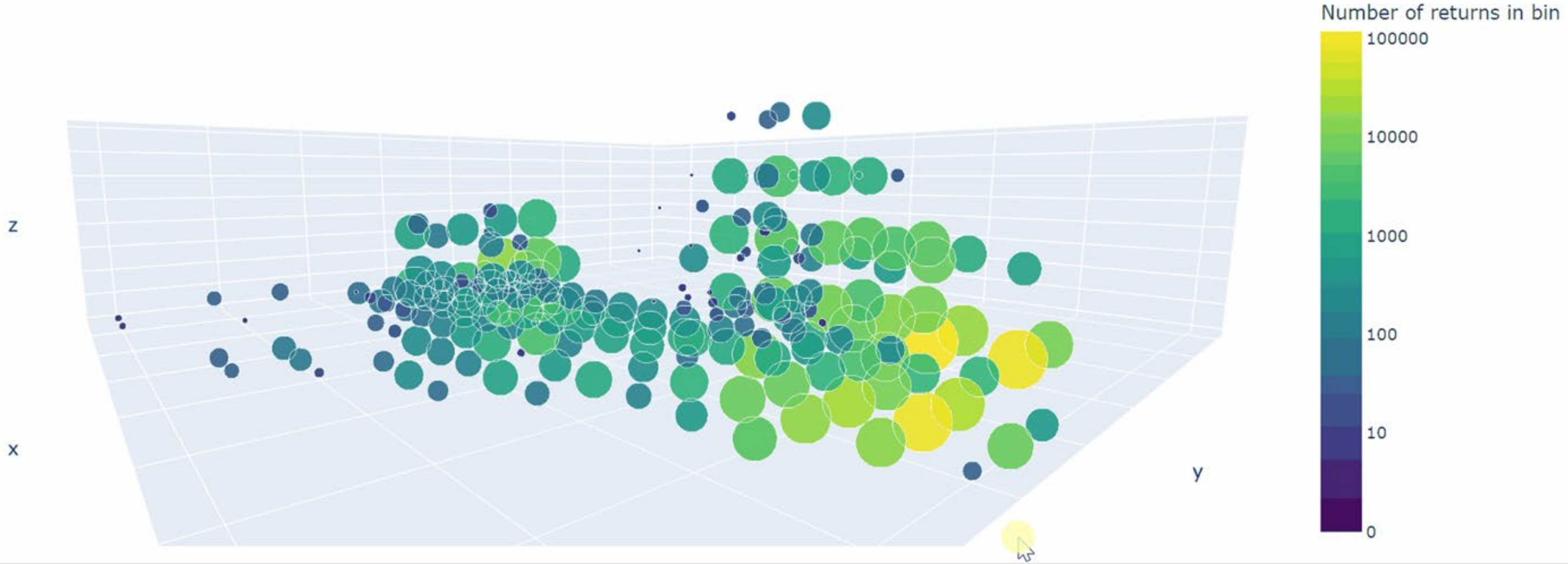
1. Create a grid structure which separates the 3D space into equally sized bins.
    - The size of bins and area of interest are configurable.

1. For every point in the point cloud, add 1 to its respective bin.

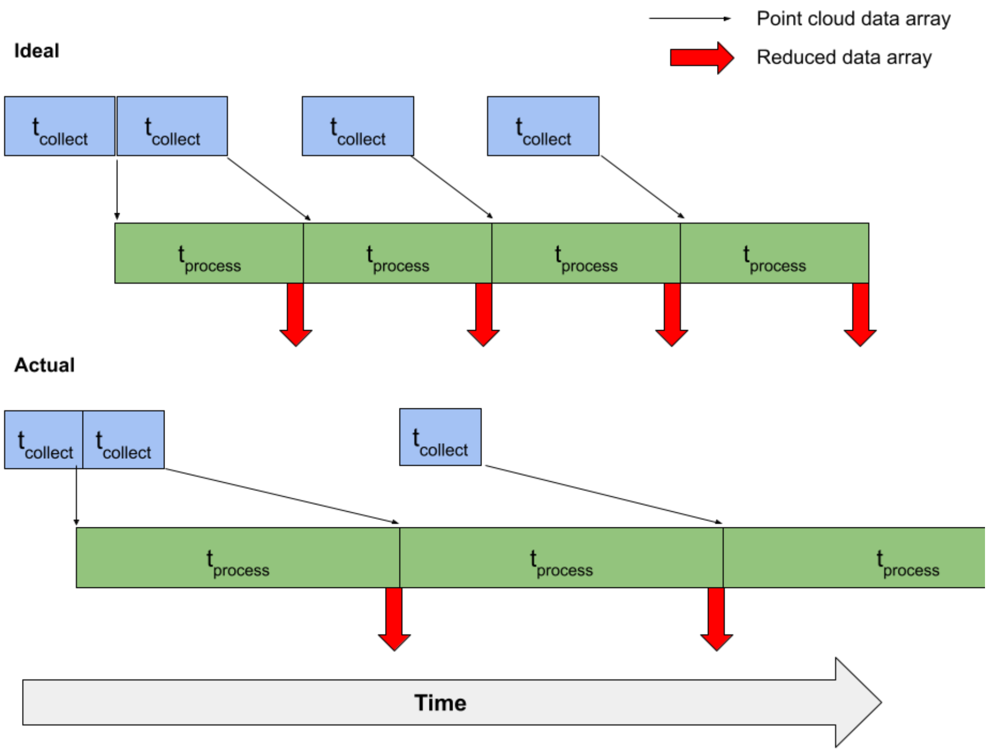
Output:

A 3D array with each element containing the number of snowflake returns in the bin location.

Results - Snowflake Density Visualization



Discussion - Timing



- **Speedup?** Only about 5%.
- Ground height computation is costly.
- Snowflake density is highly optimized.
- Ideal scenario has a better balance.

Conclusion

- Multiprocessing implemented to parallelize point cloud collection and processing.
- Preliminary algorithms developed to extract key observations from point clouds.
- Significant downsizing of memory requirements - less frequent data offloading.

Future Work

- Extensive testing during upcoming winter.
- Optimization of current processing routines.
- Development of new processing routines (dependent on larger dataset for guidance).
- Investigation of data buffers/further multiprocessing to improve data throughput with timing imbalance.

Thank You

Questions?

Fletcher Wadsworth
fwadswor@uwyo.edu