Lidar Classification

Christopher Crosby
San Diego Supercomputer Center / OpenTopography

Mike Oskin
UC Davis / KeckCAVES

(with content adapted from Ralph Hagerud (USGS))

2011 SCEC LiDAR Short Course:
Imaging & Analyzing Southern California’s Active Faults with High-Resolution Lidar Topography
October 24-25, 2011
Overview

1. Lidar technology

2. Data collection workflow

3. Data products, formats, metadata

4. Lidar and vegetation

5. QA/QC, artifacts, issues to keep in mind

6. DEM generation from lidar point cloud data
all surveyed points

ground points identified by semi-automatic processing

Nookachamps Creek, east of Mount Vernon, Washington – R. Hagerud, USGS
What is ground?

Three assumptions:
Can be used to guide automated processing approaches

1. Ground is smooth
   – despiking, iterative linear interpretation algorithms

2. Ground is continuous (single-valued)
   – No-multiples algorithm

3. Ground is lowest surface in vicinity
   – Block-minimum algorithms

Modified from: R. Hagerud, USGS
Ground is smooth ⇒ despike algorithm

Approach:

1. flag all points as ground

2. repeat:
   - build TIN (triangulated irregular network) of ground points
   - identify points that define strong positive curvatures
   - flag identified points as not-ground

3. until no or few points are flagged

R. Hagerud, USGS
Start with mixed ground and canopy returns (e.g. last-return data), build TIN
Flag points that define spikes (strong convexities)
Rebuild TIN
Flag points that define spikes
(strong convexities)
Rebuild TIN

R. Hagerud, USGS
Flag points that define spikes (strong convexities)

R. Hagerud, USGS
Rebuild TIN

R. Hagerud, USGS
Despike algorithm

Benefits:

• It works

• It’s automatic
  – Cheap(!)
  – All assumptions explicit

• It can preserve breaklines

• It appears to retain more ground points than other algorithms

R. Hagerud, USGS
Despike algorithm

Problems:
• Removes some corners
• Sensitive to negative blunders
• Computationally intensive
• Makes rough surfaces

Cross-section of highway cut

R. Hagerud, USGS
In the real world…

- Almost all return classification is done with proprietary codes (Terascan the standard)

- Successful classification uses a mix of
  - Sophisticated code
  - Skilled human
    - To adjust code parameters
    - To identify and remedy problems

- Let somebody else do it! - *then carefully check their work*

- We have no useful metrics for accuracy of return classification

R. Hagerud, USGS
Do it yourself:

Open Source - Automated:

- LASTools – lasground.exe & lasclassify.exe
- MCC-lidar (Evans & Hudak, 2007) http://sourceforge.net/apps/trac/mcclidar/
- BCAL lidar tools (requires ENVI): http://bcal.geology.isu.edu/tools-2/envi-tools

More discussion: http://www.opentopography.org/index.php/blog/detail/tools_for_lidar_point_cloud_filtering_classification#comments

Open Source - Manual:

- LidarViewer (KeckCAVES)
When Automatic Classification Goes Wrong:
Dumay Slip-Rate Site, Enriquillo Fault, Haiti

This data set was processed quickly for assessing urban area, not faults
Manual classification practical for small areas using a 3D environment