LiDAR Land Classification with Z and I values

by Scott Fletcher

The Plan

- Convert LiDAR files from .las to ASCII containing Header and x,y,z,i point values
- Extract relevant Header information
- Create grid based on header information
  - Min/Max X and Y, cell size
- Group the point data by grid location
- Determine statistics for z and i values
- Create raster files from resulting statistics
- Run classification scheme on resulting rasters

The Study Area

The Study Area

Step #1

Convert LiDAR .las files to ASCII format with header information and x,y,z,i values

The Plan:

- Install liblas Python/C (non-standard) library for parsing .las file

Step #1

Convert LiDAR .las files to ASCII format with header information and x,y,z,i values

The Plan:

- Install liblas Python/C (non-standard) library for parsing .las file

Plan B:

- Download las2txt (C program) and use it to translate .las into ASCII format

Step #2

Extract Relevant Header Information

The Plan:

- Create a Python module to parse the ascii file for header information including:
  - Min/max x and y, number of points,

Step #3

Create grid based on header information

Group the points data by grid location

The Plan:

- Create grid-like data structure, traverse the file, grouping the points by grid cell

Step #3

Create grid based on header information

Group the points data by grid location

The Plan:

- Create grid-like data structure, traverse the file, grouping the points by grid cell

Plan B:

- Create grid-like data structure, traverse half of the file, grouping points by grid cell, write contents to file, empty grid-like data structure, repeat for second half of file

Step #4 (new step)

Merge resulting grid files into one file

The Plan:

- Traverse the two files at the same time, create a third file with combined lists of points from both

Step #5

Determine statistics from z and i values

Create raster files from resulting stats

The Plan:

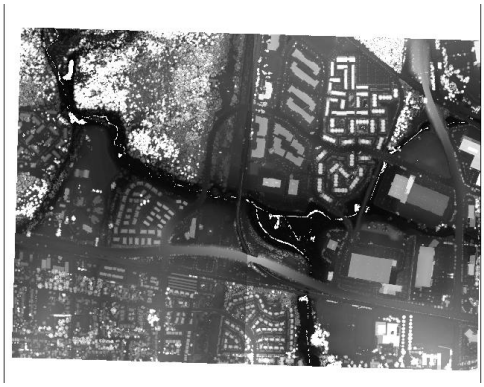
- Traverse the grid file, determine the following:
  - Min, max, range, average, standard deviation

  for z and i values in each cell
- Create a separate raster file for each statistic (5 z raster files, 5 i raster files)

Raster Images

Average Intensity

Raster Images



Average z-value

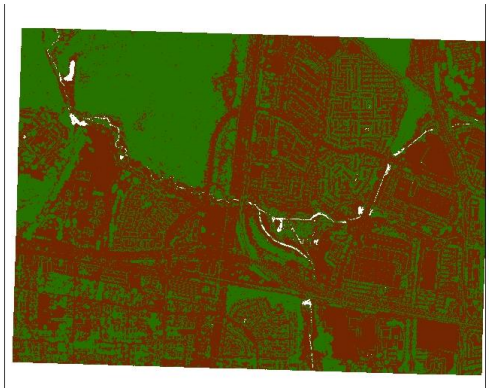
Step #6

Run classification scheme on the resulting raster files

The Plan:

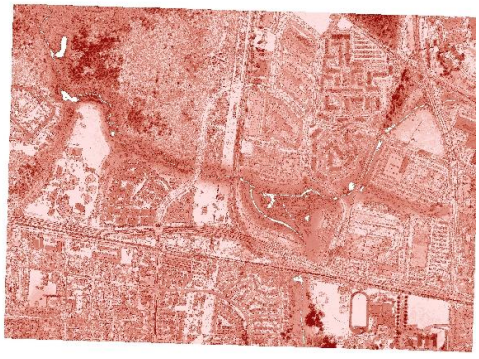
- Use ISO Cluster tool in ArcMap to cluster the values from the resulting raster files
- Run Maximum Likelihood Classification on the results

Classification Results



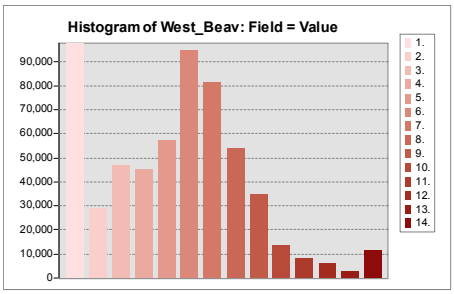
2 Classes

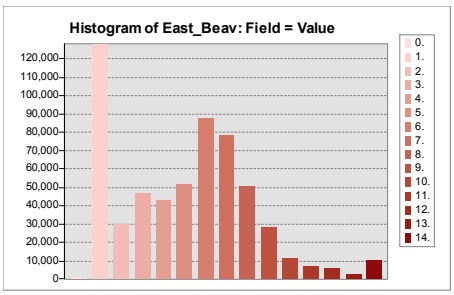
Confidence Raster



Darker Red representing less confidence

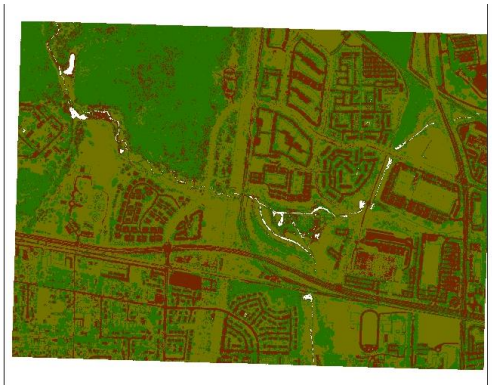
Confidence Histograms





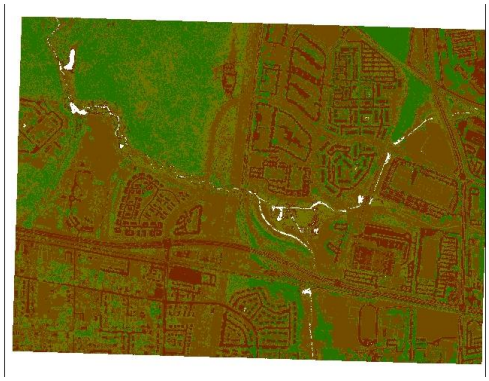
Represents the number of points at confident levels from 100% to 87%

Classification Results



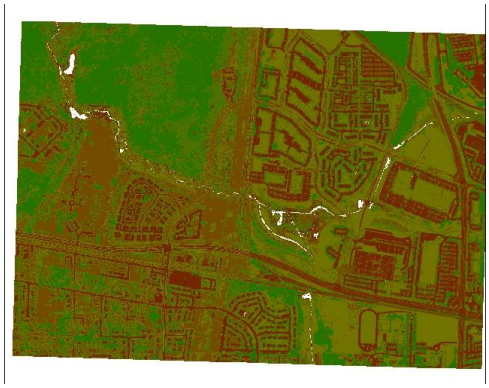
3 Classes

Classification Results



4 Classes

Classification Results

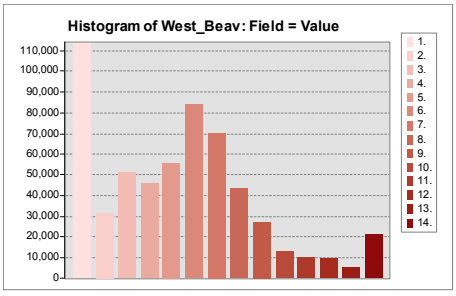


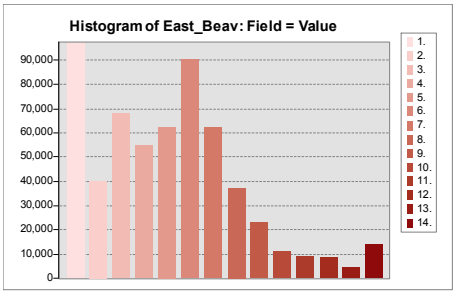
5 Classes

Confidence Raster

Darker Red representing less confidence

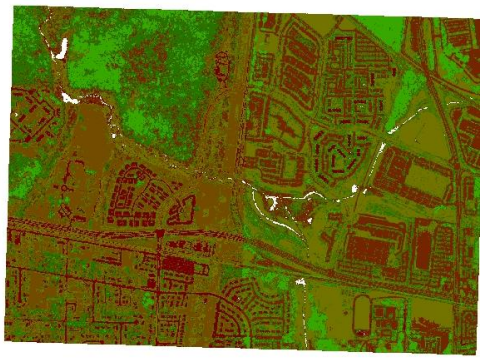
Confidence Histograms





Represents the number of points at confident levels from 100% to 87%

Classification Results



7 Classes

Thoughts for Improvement

- Include other statistics
  - Moran's I, GLCM, Shannon's Diversity, Number of points per cell (in the case of multiple returns)
- Improve application interface / behavior
  - GUI, liblas (or call las2txt from app), more robust (for use with other LiDAR files)

References

las2txt was obtained from the following url:
http://www.cs.unc.edu/~isenburg/lastools/

ESRI Grid format information:
http://en.wikipedia.org/wiki/ESRI_grid

Numpy Python module downloaded from:
http://sourceforge.net/projects/freshmeat_numpy/

Geoffrey Duh:
Cramer Hall 424