http://lidar.asu.edu

An Efficient Implementation of a Local Binning Algorithm for Digital Elevation Model Generation of LiDAR/ALSM Dataset

Han Suk Kim[1], J Ramón Arrowsmith[2], Christopher J. Crosby[2], Efrat Frank[3], Viswanath Nandigam[3], Jeffrey Conner[2], Ashraf Memon[3], Scott B. Baden[2], and Chaitan Baru[3]

[1]Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093
[2]School of Earth and Space Exploration, Arizona State University, Tempe, AZ 85287
[3]San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA 92093
hskim@cs.ucsd.edu, ramon.arrowsmith@asu.edu

1. INTRODUCTION

The GEON LiDAR Workflow (GLW) project (http://www.geongrid.org/science/lidar.html) has enabled geoscientific communities to process and download large LiDAR (Light Distance And Ranging), or ALSM (Airborne Laser Swath Mapping) datasets. The high density and random distribution of LiDAR datasets make production of digital elevation models (DEMs) computationally challenging. DEMs are essential representations of the landscape that are imported in various off-the-shelf geoscientific tools.
So far, the GLW has utilized the regularized spline with tension interpolation algorithm for the DEM generation from GRASS GIS successfully for small (<1.6 million points). However, it is compute-intensive; in order to process 5 million LiDAR points, it takes about 50 minutes on the current single processor Linux machines in a GEON cluster.

2. LOCAL BINNING ALGORITHM

The algorithm utilizes the elevation information from only local reference points, the points inside of a circular search area with user specified radius.

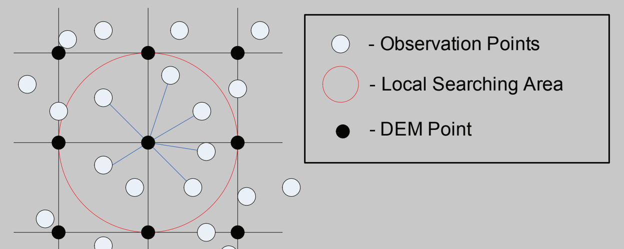


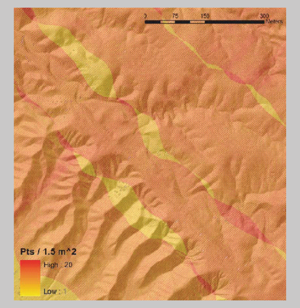
Figure 1. Local Binning Algorithm

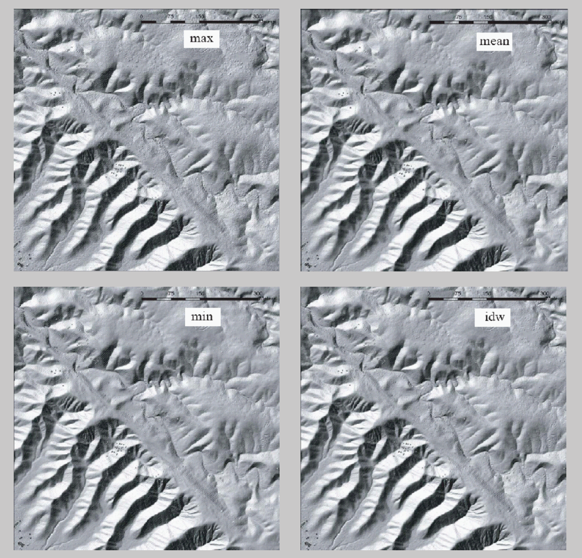
With the local points, five values are computed for each node in a grid:
1) Minimum
2) Maximum
3) Mean
4) Inverse Distance Weighted mean of the local points
5) Density (the number of points in the search area)

$$Z_{min} = min(Z_i)$$
$$Z_{mean} = mean(Z_i)$$
$$Z_{max} = max(Z_i)$$
$$Z_{IDW} = \frac{\sum_{i=1}^{n} \frac{Z_i}{d_i}}{\sum_{i=1}^{n} \frac{1}{d_i}}$$

If the number of points in the search radius is 0, the node is assigned a null value. The noble implementation technique can produce a grid containing those five values within O(N) time, where N denotes the size of the point cloud used. In addition, the space cost for this implementation is O(M), where M denotes the size of the grid. See Performance Section.

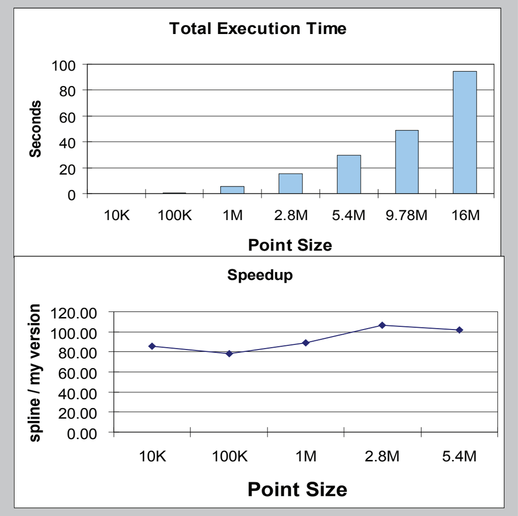
Examples
Right: Figure showing the number of LiDAR returns per search area for aportion of the B4 dataset in the Carizo Plain near the Dragon's Back. For this 1 m DEM, a search radius of 0.707 m was used, giving a search area of 1.5708 m^2. Note the heterogeneous LiDAR return distribution due to swath overlaps and changes in plane orientation. LiDAR return concentrations vary from 1 to 20 per search area. This grid was produced from ~4.7 million LiDAR points.
Four hillshaded 1m DEMs for the same as shown above. All four DEMs are produced by the local binning algorithm implemented in the GLW. The local binning approach works well when the DEM resolution is greater than the LiDAR return spacing. When the DEM resolution approaches the shot spacing, a large search radius or an interpolation algorithm (e.g. spline) must be used to avoid numerous null values

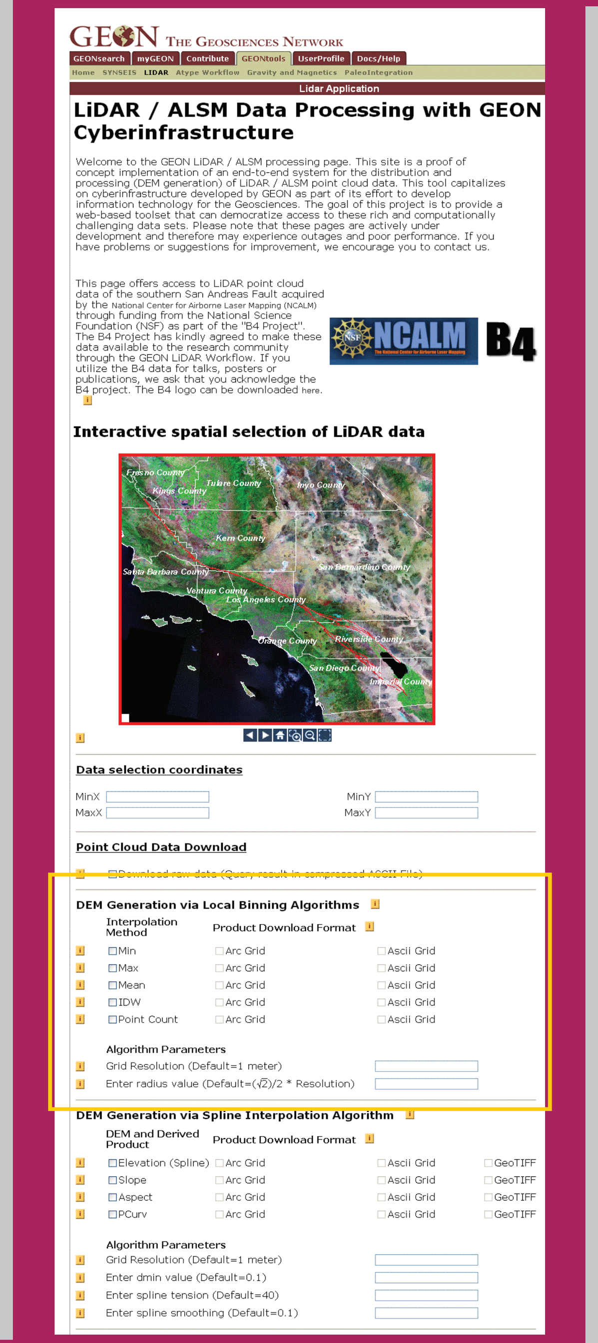




3. PERFORMANCE EVALUATION

5 million LiDAR points can be processed in 30 seconds, which is 100 times faster than the regular spline algorithm in GRASS GIS.
Various test files show about 80~100 X speedup where speedup is defined as a ratio of the execution time of the local binning algorithm to that of the regular spline algorithm implemented in the GRASS GIS.

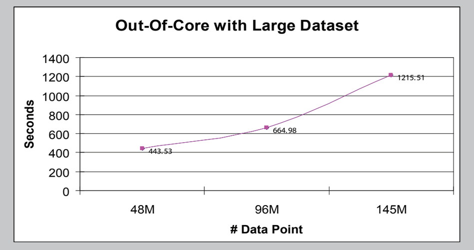


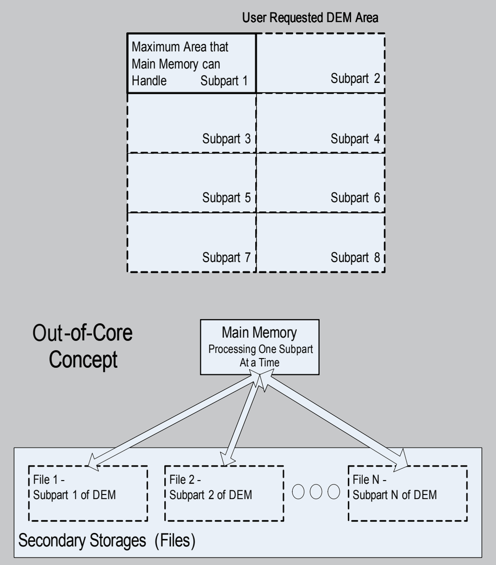


4. IMPLEMENTATION AND DEPLOYMENT

Out-of-Core Implementation
An out-of-core (memory) version of the local binning algorithm has been developed. This implementation exploits secondary storage for saving intermediate results when the size of a grid exceeds that of memory. The large grid is split into several pieces so each piece can be fit in main memory. During the computation, the pieces reside in secondary storage and each piece is fetched to main memory one at a time when needed.

The out-of-core version can compute a DEM from 145 million points in about 20 minutes. The only overhead of this implementation is the cost for swapping pieces. However, in the implementation, the number of swapping is minimized by using in-memory buffer in which interpolation information of other pieces is kept.





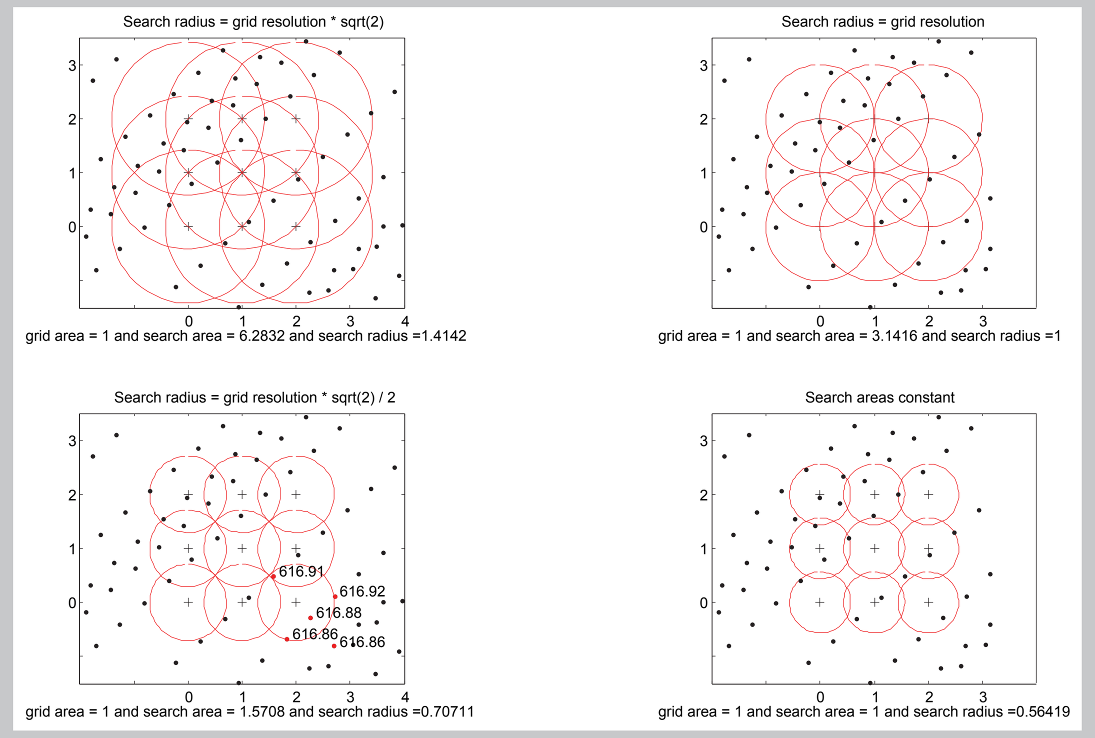
Deployment
The local binning algorithm has been deployed in the GEON LiDAR Workflow. Users can control the grid resolution and search radius as they want.
The products are written in two formats: ArcGrid and Ascii Grid. Both of them are widely used in geoscience communities.

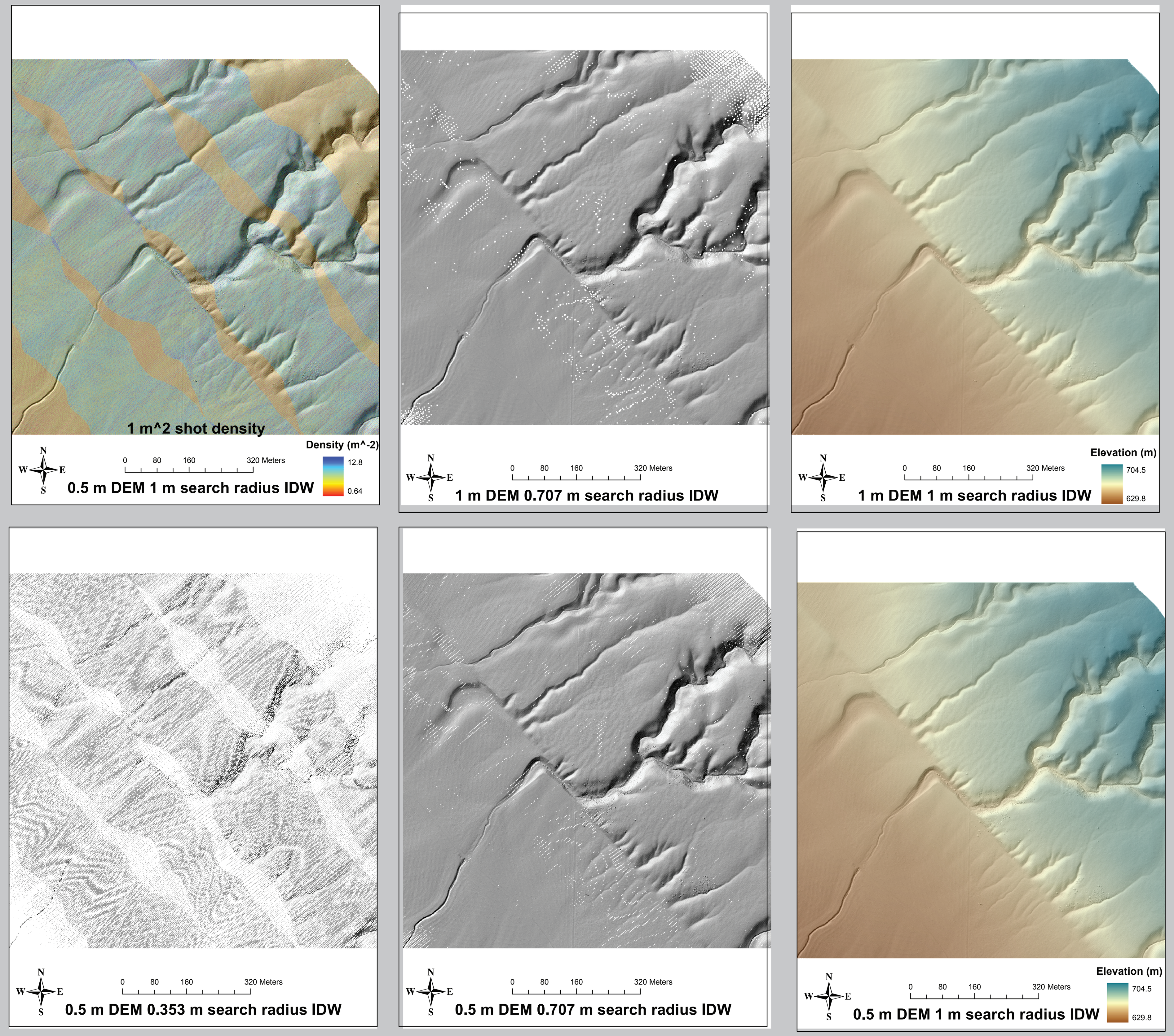
5. ANALYSIS

Given the search radius implementation, what does it actually look like with actual ALSM data, and what are the search radius tradeoffs?
This plot shows different search radius options as a function of the grid/DEM resolution (+ indicates grid node).

The example is for the typical 1 m DEM and the data are actual B4 survey data from the Carrizo Plain (black dots). The red circles show the different search circles, while the annotation below indicates the search area and search radii as a function of grid/DEM area. We concluded that the optimal (and thus default for the GLW) search radius = (grid resolution * sqrt(2) / 2), or in this 1 m DEM case, search radius = 0.707 m. This option covers all of the survey, but does have a slightly larger search area (1.56 m^2) than the grid area 1 m^2. For some locations, however, it leaves a modest but notable amount of null cells in which no points are encountered. Thus, in practice, we have started to use the upper right case where the 1 m DEM is produced with a 1 m search radius (search area = 3.14 m2).

In addition, the figure shows the actual points in red with their elevations that are close to the search radius for the lower right node. Within the search radius, we operate on the selected points to produce the grid with a local minimum, maximum, mean, inverse distance weighted mean, and point count.



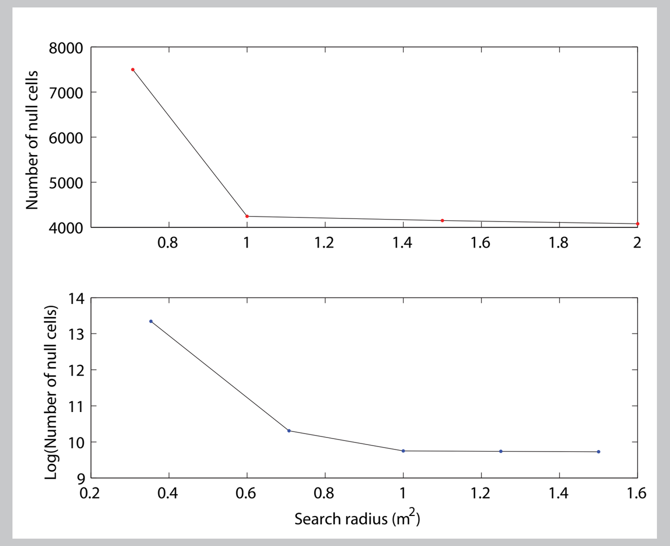


Shot densities in the Carrizo Plain in the B4 airborne laser swath mapping data set have shot densities are as high as 12.8 shots per 1 m^{-2} in areas of swath overlap. Typical densities are 2-3 shots per 1 m^{-2} in areas of single swath and about 5 shots per 1 m^{-2} in double overlap.

Obviously, there is a clear tradeoff between the point density and the DEM accuracy. As the search radius increases, more data from LiDAR return is included in a DEM grid not estimation. In such a way, we can prevent DEMs from having null points. On the other hand, larger radius can include wrong values especially when the slope of the area is steep.

The plots up and to the right show DEMs produced with varying resolution (1 m or 0. 5 m) and varying search radii. As the images show and the plot below indicates, the density here is such that a search radius of about 1 is optimal

The plot at right shows the number of nulls per DEM versus search radius. The first value is the default search radius (grid resolution * sqrt(2) / 2) for the 0.5 m (upper) and 1 m (lower) DEMs. At about the 1 m search radius, the number of nulls become asymptotic to the minimum number of nulls defined but the strip at the edge of the data on the northeast (upper right) of the images.



6. FUTURE WORK

In the near future, we will parallelize this code, which will reduce the execution time further. Then, the parallelized and out-of-core implementation will be able to routinely process thousands of requests over broad areas of interest in the GEON LiDAR Workflow. Another possible work is to build an Win32 executable so users can generate DEMs from their own LiDAR datasets. As the local binning algorithm requires less hardware resources (CPU time and memory), the Win32 version of local binning algorithm will help scientists experimenting with various datasets.

7. ACKNOWLEDGEMENTS

Special thanks to our SDSC/ASU/UCSD colleagues working on the GEON LiDAR Workflow. Thanks also to the B4 team and Mike Oskin.

This material is based upon work supported by the National Science Foundation under grants EAR-0225543 (GEON), EAR-0409745 (Bidart paleoseismology) & the Southern California Earthquake Center