ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/277622377

Reconstruction of a 3D Polygon Representation from full-waveform LiDAR data

Conference Paper · September 2014

CITATIONS
0

READS
184

5 authors, including:

Milto Miltiadou
Cyprus University of Technology
7 PUBLICATIONS **3** CITATIONS

SEE PROFILE

Mark Warren
Plymouth Marine Laboratory
22 PUBLICATIONS **46** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

GlobCurrent View project

SEO-DWARF H2020 RISE project View project

All content following this page was uploaded by Milto Miltiadou on 03 June 2015.

The user has requested enhancement of the downloaded file.

Reconstruction of a 3D Polygon Representation from full-waveform LiDAR data

Milto Miltiadou[1,2], Michael Grant[2], Matthew Brown[1], Mark Warren[2], Emma Carolan[2]

[2]*Centre for Digital Entertainment, University of Bath, Bath UK, mm841@bath.ac.uk*

[1]*Remote Sensing Group, Plymouth Marine Laboratory, Plymouth, UK, mmi@pml.ac.uk*

Corresponding author name: *Milto Miltiadou, mmi@pml.ac.uk, 07549700928*

ABSTRACT

This study focuses on enhancing the visualisation of FW LiDAR data. The intensity profile of each full-waveform pulse is accumulated into a voxel array, building up a fully-3D representation of the returned intensities. The 3D representation is then polygonised using functional representation (FRep) of geometric objects. In addition to using the higher resolution FW data, the voxels can accumulate evidence from multiple pulses, which confers greater noise resistance. Moreover, this approach opens up possibilities of vertical observation of data, while the pulses are emitted in different angles. Multi-resolution rendering and visualisation of entire flightlines are also allowed.

Introduction: The most common approach of interpreting the data, so far, was decomposition of the signal into a sum of Gaussian functions and sequentially extraction of points clouds from the waves (Wanger, Ullrich, Ducic, Malzer , & Studnicka, 2006). Neunschwander et al used this approach for Landover classification (Neuenschwander, Magruder, & Tyler, 2009) while Reightberger et al applied it for distinguishing deciduous trees from coniferous trees (Reitberger, Krzystek, & Stilla, 2006). In 2007, Chauve et al proposed an approach of improving the Gaussian model in order to increase the density of the points cloud extracted from the data and consequently improve point based classifications applied on full-waveform LiDAR data (Chauve, Mallet, Bretar, Durrieu, Deseilligny, & Puech, 2007).

In this research, particular attention is drawn on the visualisation of the data. Previous work in visualising FW LiDAR has used transparent objects and point clouds. Inserting the waveforms into a 3D Volume and visualising them using different transparencies across the voxels was proposed by Perssion et al in 2005. In "FullAnalyze", for each waveform sample, a sphere with radius proportional to its amplitude is created (Chauve et al, 2009). However, both publications are based on small regions of interest, while entire flightlines can be visualised using our approach.

Here it worth mentioning that the full-waveform LiDAR data are provided by NERC ARSF. The data was collected on the 8th of April in 2010 at New Forest in UK using a small footprint Leica ALS50-II system. The backscattered signal was saved into LAS1.3 files after being digitised using 256 samples at 2ns intervals. This corresponds to 76.8m of waveform length.

Method: A volumetric approach of polygonising FW LiDAR data is proposed here. Voxelisation is chosen over Gaussian decomposition, to decrease the amount of information reduced while discretisation and allow multi resolution regular sampling of the data. First, the waveforms are inserted into a 3D Volume, then an FRep object is defined by the Volume and by the end the FRep object is polygonised using the Marching Cubes algorithm. More details are given below.

The waveforms are converted into voxels by inserting the waves into a 3D volume, similar to Person et al, 2005. But in our case, low level filtering is applied to discard noise first.

Further, to overcome the uneven number of samples per voxel, the average amplitude of the samples that lie inside each voxel is taken, instead of selecting the sample with the highest amplitude. Therefore:

$$Voxel\ Intensity\ Value = \frac{(\sum_{i=1}^{n}(\mathrm{Ii}))}{n},$$

where n is the number of samples inserted into that voxels and Ii is the intensity of the sample i.

The results of the normalisation are shown on the following thickness maps generated from the same flightline; A thickness map is an image, where each pixel value represents the number voxel between the first and the last non-empty voxels of each column (z-axis). As shown below, the quality of the output image is significantly increased when normalisation is applied.

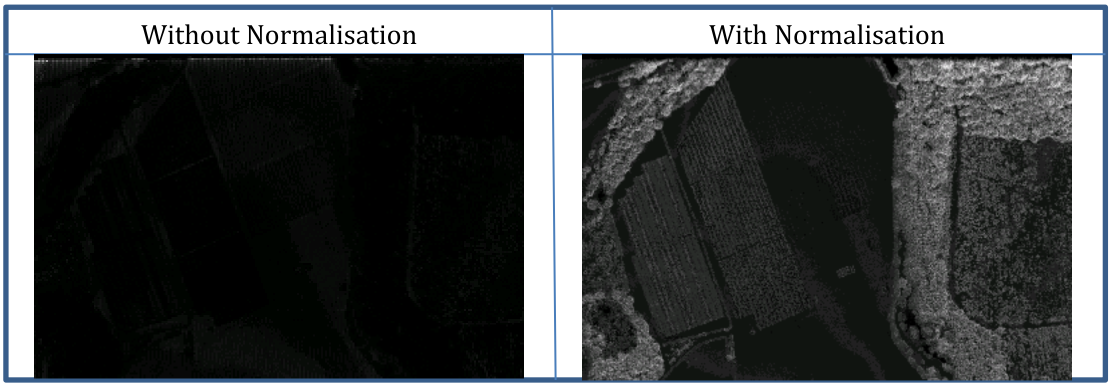


Figure 1: Thickness map, before and after Normalisation

Another problem to be addressed is the noise. The system records and digitises 256 samples per pulse. When the pulse doesn't hit any objects, the systems still records low signals which are as noise. For that reason low level filtering is applied and the samples with amplitude lower than the noise level are discarded. Aliasing also seems to appear on areas with small thickness like the ground. But addressing this problem is beyond the scope of this paper.

Once the pulse samples are inserted into a 3D Volume, the volume is then used as a discrete density function (f(X) , a) to represents an FRep object. Recalling form Pasko et al, an FRep object is defined by a continuous function f(X) where:

$f(X) = \alpha$, when X lies on the surface of the object
$f(X) > \alpha$, when X lies inside the object and
$f(X) < \alpha$, when X lies outside the object (Pasko & Savchenko, 1994)

f(X), in our case, is a discrete density function that takes as input a 3D point X and returns the accumulated intensity value of the voxel that X lies in.

X is 3D point (x, y, z) and here x, y and z are longitude, latitude and height respectively.

α is the isolevel of the object and defines the boundary of the object. f(X) is equal to α iff X lies on the surface of the object . On the original paper α=0, but in this case, α thresholds some of the noise from the actual object. α is also a user defined parameter and can vary depending on the amount of noise that exists in the data. As shown later at the results, while α decreases, the number of non-empty voxels classified as noise increases and the amount of information preserved decreases.

An FRep object is defined by a continuous function and the quality of it is not defined. On the one hand, this is useful on reducing storage memory and it also allows multiple rendering resolutions of the same object. But on the other hand, the object has no discrete values (vertices, faces and edges). So, processing is required before rendering/visualising. This problem is either address by ray-tracing or by polygonising the object. In this case we chose polygonisation using the Marching Cubes Algorithm, which allows direct rendering with commodity 3D-accelerated hardware.

The Marching cubes algorithm is an algorithm used to construct surfaces from implicit objects using a search table. Let's assume that $f(X)$ defines an object to be polygonised. At first a 3D volume is divided into cubes, named voxels. Each voxel is defined by eight corner points and each point lies either inside or outside the object. This is calculated from the function $f(X)$, as explained above. Then, by enumerating all the possible cases and linearly interpolating the intersections along the edges, the surface of the implicit object is constructed (Lorensen & Cline, 1987).

According to Lorensen and Cline, the normal of each vertex is calculated by measuring the change of gradient in that area. In our case, this does not lead to a smooth looking surface, due to the high gradient changes that exist in the Volume, especially where trees exist. Therefore, for each vertex we get the average normal of its adjacent triangles.

Results and Experiments:

The output of our system is a 3D polygon mesh. The area of interest is user defined, so either an entire flightline or a small area can be visualized (Figure 1). Further, the output could either be derived from FW LiDAR or discrete LiDAR, but as shown on Figure 2, polygon meshes created from the FW data contain more information.

By the end, Figure 4 shows how the results are modified while increasing or decreasing the rest three user-defined parameters of our system: Voxel Length, Isolevel and Noise Level. The voxel Length controls the resolution of the output; the bigger the voxel length is the lower the resolution is. The isolevel is the boundary that defines whether a voxel is inside or outside the object. While isolevel increases the number of voxels that are inside the object decreases. For that reason if this value is set too high, the object seems to disappear. The Noise level, here, serves low level filtering. All the samples with intensities less than the noise level are ignored. If the noise level is too low, then the noise is obvious on the results and if it is too high important information are discarded and the object seems to disappear again.

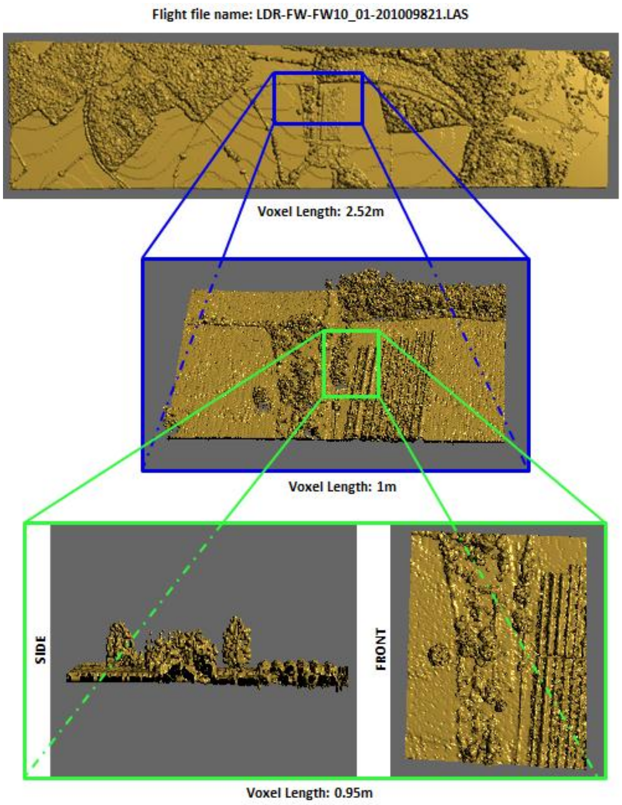


Figure 2: selecting region of interest

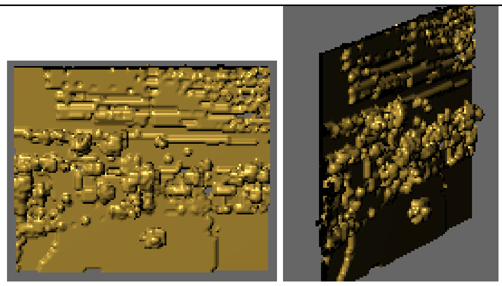
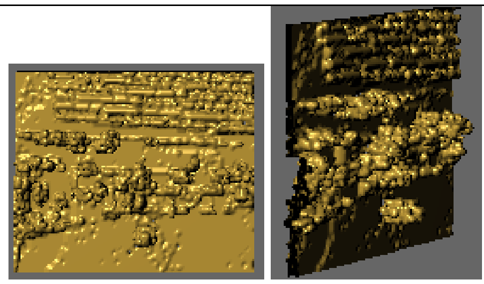
	Discrete LiDAR	Full-wavefrom LiDAR
Voxel Length = 1.7m		

Figure 3: selecting region of interest

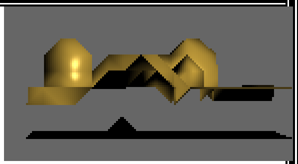
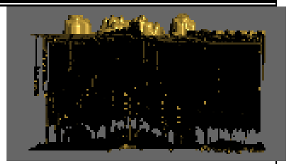
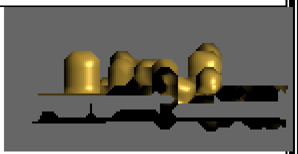
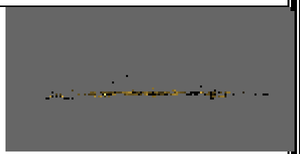
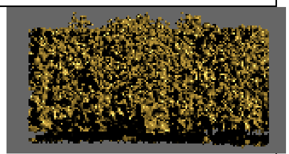
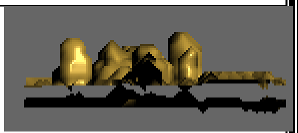
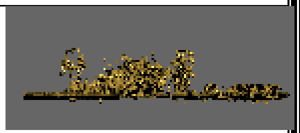
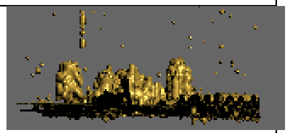
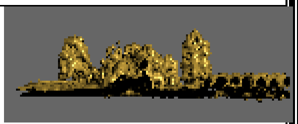
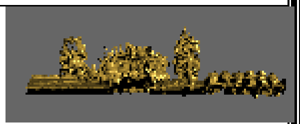
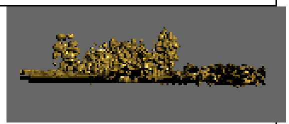
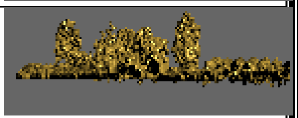
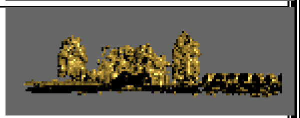
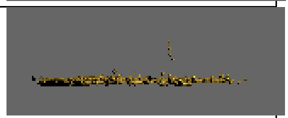
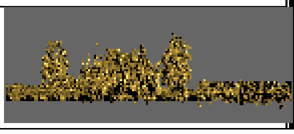
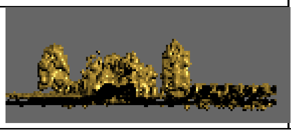
Voxel Length	Visualisation with different voxel lengths	Isolevel	Visualisations with various isolevels	Noise Level	Visualisations with various noise levels
10.0 m		45		5	
5.7 m		15		15	
4.44 m		-45		17	
1.43 m		-60		30	
1.0m		-85		75	
0.67 m		-100		135	

Figure 4: Switching the user defined parameters

Conclusions: To sum up, previous work on visualisation uses either transparent voxels or spheres while at this paper, an approach of generating fully 3D polygon representations of FW data was presented. A 3D Volume representation of FW LiDAR data is firstly generated, by accumulating the intensity profile of each recorded full-waveform into a voxel array. The 3D representation is then polygonised using functional representation of objects (FReps).

The output is a 3D-polygon representation of the selected data, showing well-separated structures such as tree canopies and greenhouses. The polygon is suitable for direct rendering with commodity 3D-accelerated hardware, allowing smooth visualisation. Furthermore, comparing the results of applying the same method on discrete LiDAR, the polygons generated from FW LiDAR contain more detail. The user-defined parameters (resolution, noise-level, isolevel and region of interest) also increase the flexibility of our system. Finally, this method is particularly beneficial for various resolutions rendering of the data, while entire flightlines can be visualised.

References:

Chauve, A., Bretar, F., Durrieu, S., Pierrot-Deseilligny, M., & Puech, W. (2009). FullAnalyze: A research tool for handling, processing and analysing full-waveform LiDAR data. IEEE International Geoscience & Remote Sensing Symposium. Cape Town: South Africa.

Chauve, A., Mallet, C., Bretar, F., Durrieu, S., Deseilligny, M. P., & Puech, W. (2007). Processing full-waveform LiDAR data: Modelling raw signals. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.

Lorensen, W. E., & Cline, H. E. (1987). Marching Cubes: A High Resolution 3D Surface Construction Algorithm. General Electric Company Corporate Research and Development Schenectady. New York: ACM.

Neuenschwander, A., Magruder, L., & Tyler, M. (2009). Landcover classification of small-footprint, full-waveform lidar data. Jounal of Applied Remote Sensing, Vol. 3, 033544.

Pasko, A., & Savchenko, V. (1994). Blending operations for the funcitonally based constructive geometry.

Persson, A., Soderman, U., Topel, J., & Ahlberg, S. (2005, September). Visualisation and Analysis of full-waveform airborne laser scanner data. V/3 Workshop "Laser scanning 2005". Enschede, the Neverlands.

Reitberger, J., Krzystek, P., & Stilla, U. (2006). Analysis of full waveform LiDAR data for tree species classification.

Wanger, W., Ullrich, A., Ducic, V., Malzer , T., & Studnicka, N. (2006). Gaussian decompositions and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. ISPRS Journal of Photogrammetry and Remote sensing 60, 100-112.

Wanger, W., Ullrich, A., Melzer, T., Briese, C., & Kraus, K. (2004). From single-pulse to full-waveform arborne laser scanners: potential and practical challenges.

Keywords: *Visualisation, full-wavefrom LiDAR, Voxelisation,* FRep*, 3D-polygon*

Indicate here preferred conference session: *LiDAR*

Indicate here preference between oral and poster presentation: *ORAL*

View publication stats