FUTURE FORESTS RESEARCH

RADIATA MANAGEMENT TECHNICAL NOTE

Number: RTN-011
Date: December 2011

How to use Aerial LiDAR Data

Summary

This technical note gives an overview of creating Digital Elevation Models (DEMs) from aerial LiDAR data, and a step-by-step guide to using the free FUSION software for obtaining LiDAR metrics for sample plots. The guide then shows how to generate these metrics over the entire LiDAR area and create forest parameter maps such as total standing volume.

This procedure can be followed without any programming skills, but would be more efficiently performed with basic scripting knowledge. A sample script – which can be run in the free software Octave – is provided as an appendix, and is available by emailing: dave.pont@scionresearch.com.

The techniques and software shown here are already in common usage, although if any company needs further coding assistance they are advised to contact Scion or the authors.

Authors: Tom Adams, Peter Beets, Thomas Paul, Rod Brownlie

LiDAR Data

Aerial LiDAR data are now being acquired over New Zealand planted forests by both forest owners and regional councils. This tech note provides a guide to how to use those data to create Digital Elevation Models (DEMs), and LiDAR metrics which may be used to map stand attributes such as total standing volume or carbon.

The two New Zealand LiDAR suppliers – New Zealand Aerial Mapping (NZAM) and Aerial Surveys – currently each own the same type of scanner (Optech 3100EA), and deliver outputs in similar forms. The most common use for these data is for DEMs, but some forest owners are investigating the potential of using LiDAR metrics to infer information on the estate, particularly for volume and carbon.

DEM Creation

Nowadays most LiDAR suppliers will supply a ready-to-go DEM with the LiDAR data. If this is not the case, or the data are old or second-hand, there are many software packages with the ability to build DEMs from LiDAR. This process entails:

- filtering the LiDAR data into ground / non-ground returns;
- fitting a surface to the ground points; and
- outputting or displaying the surface in a usable fashion.

Both NZAM and Aerial Surveys provide classified LIDAR data, meaning that the filtering stage has already been done. Both use TerraScan software (regarded as one of the best in the business) and manual checking to identify breaklines such as ridges, valleys, bluffs etc. Figure 1 shows a hill-shaded DEM created from a) FUSION's fully automated GroundFilter procedure, and b) NZAMs manually corrected TerraScan filtering. Using manual breaklines will inevitably lead to slightly idealised, simplified DEMs, whilst fully automated filtering (e.g. FUSION) is likely to add more 'noise', or non-existent features. In the two examples below the continuity of roads (top left) and overall form appear more realistic in the NZAM/TerraScan DEM, but there is the possibility that some of the 'extra features' in the FUSION DEM are also genuine. When a smoother DEM is required, the supplied semi-manual classification can be considered superior to any fully-automated result, although when very small-scale topography is important users may benefit from comparing results from an additional filter.

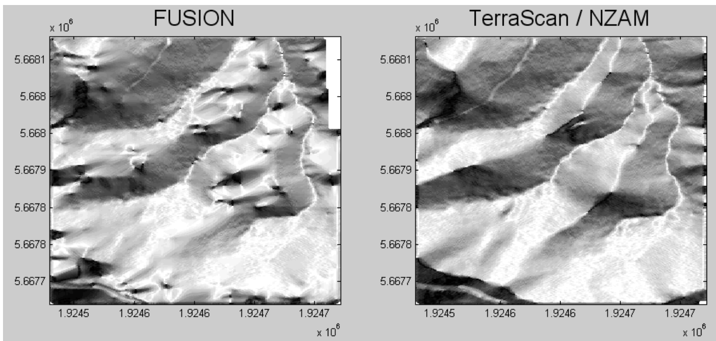

Figure 1 – DEMs generated from a) automated ground filtering in FUSION and b) semi-manual filtering in TerraScan

Fitting a surface to the ground points is facilitated in many software packages, the most common of which is ArcGIS, but many surveying and engineering tools

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

Number: RTN-011
Date: December 2011

also share this facility. Most of these tools have the capacity to display the results, and in some cases to export for other uses. In the next section we show how to do this in FUSION, although FUSION is only necessary if LiDAR metrics are also required.

A surface will be a raster (grid) or a TIN (triangulated irregular network, a network of connected nodes). The choice of TIN or raster depends on the software used, although a TIN is better with sparse but reliable ground filtering whereas a raster may be more suitable when there are many ground points but the filtering is questionable. Arc is compatible with both TINs and rasters. The cell size (resolution) of the DEM should not be much less the average pulse spacing: $\left(\sqrt{\frac{\textit{Number of ground returns}}{\textit{Total area}}}\right)$

LiDAR Plot Metrics

Most studies in which inventory data have been derived from LiDAR – such as volume, biomass and carbon – use LiDAR plot metrics. LiDAR plot metrics are variables defined from subsampled 'LiDAR plots' cut from the complete LiDAR point cloud, and can involve both ground and non-ground returns.

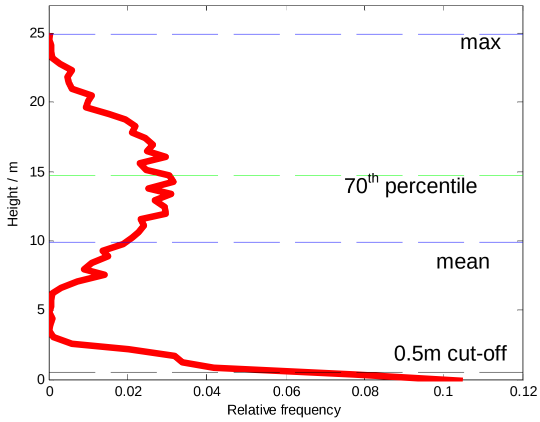

Figure 2 – Vertical distribution of LiDAR returns and typical metrics for a 9-year-old radiata pine stand.
In the following section we describe how to get these metrics from the FUSION software.

Commonly used metrics are canopy cover (the inverse proportion of the returns to hit or get near to the ground), height percentiles and intensity percentiles. Each return has a height and intensity, and percentiles are defined over a given area or 'LiDAR plot'. Thus the 80th height percentile is the height at which 80% of returns are lower, and 20% above. The 80th intensity percentile is similarly the intensity level at which 80% of returns were of a lower intensity. It is not the average intensity at the 80th height percentile. Figure 2 shows a histogram of heights for a LiDAR sample, and several characteristic metrics.

Getting Canopy Metrics from FUSION

FUSION is a free LiDAR software system built by the US Forest Service, and is available for free download from:
http://forsys.cfr.washington.edu/fusion/fusionlatest.html

Also on the same page is a manual which should be used to add depth to this tech note.

FUSION consists of a (fairly limited) user interface which is useful for overlaying LiDAR and aerial imagery, although this is better performed in Arc. The most useful components of FUSION are command-line executables, which can be accessed from the DOS prompt, or run with a batch script or with any scripting language such as Python, Matlab or R. Each executable is described individually in the manual, and tutorials are available on the website.

In the following example we show how to derive a volume map for a forest that has been flown for LiDAR. A set of plots exist in which field crews have determined the variable of interest (in this case volume per hectare). LiDAR has been supplied as a set of tiles, and the positions of the field plots have been accurately georeferenced against the LiDAR data by assigning coordinates for each plot (e.g. centre) taken from a high precision, differentially-corrected GPS (Global Positioning System).

1) Organise your files

Put FUSION and your data in easily accessed places (e.g. D: \FUSION, D:\LiDARdata). Some FUSION commands cannot handle file names or paths with spaces, dots or commas in them, so these are best avoided.

Put all the LAS tiles in their own folder, and create a text file with the full file name (including path) for each, e.g.

```
D:\temp\lidarfiles.txt

D:\LiDARdata\Forest1\LAS001001.las
D:\LiDARdata\Forest1\LAS001002.las
D:\LiDARdata\Forest1\LAS001003.las
D:\LiDARdata\Forest1\LAS002001.las
...
```

Make sure this text file is saved as a plain text (.txt) format, a file per line, and has no headers or footers. This removes the need to type out the name of every file every time you read the raw data.

2) Generate a FUSION DTM

To get canopy metrics in FUSION, you need a FUSION format DTM (even if you have already derived a DTM in another package). FUSION has two commands for making DTMs, one based on a TIN (`TINSurfaceCreate.exe`) and one based on a raster (`GridSurfaceCreate.exe`). The method should match the filtering process. Typically data from NZAM or Aerial Surveys will use TerraScan software, which uses a TIN densification method, so it is best to use the TIN method. If the filtering was performed with another system (such as FUSION's GroundFilter.exe), replace `TINSurfaceCreate` in the following example with `GridSurfaceCreate`.

Type into DOS, or the scripting system

```
C:\FUSION\TINSurfaceCreate.exe  /class:2
[dtm filename] 1 m m 0 0 0 0  [LiDAR
filenames text file]
```

For an explanation of the terms please see the FUSION manual under TINSurfaceCreate. The filenames in *[square brackets]* should be local temporary locations with no spaces.

3) Create a text file of plot locations

To get FUSION to cut out several plots in one go, you need to create a text file with a filename for each sample, and specify the bottom left and top right coordinates of the bounding box containing the plot (Figure 3). If the plot is circular, these are equivalent to the centre point ± the radius. Note that the plots are cut out in a horizontal plane, so the radius is the standard radius, not the slope-corrected one. Note that filenames end in `.lda`, a FUSION format.

```
D:\temp\PlotList.txt

D:\LiDARdata\Forest1\PlotSamples\CP100_01.lda 1974022.474 5774772.209 1974056.474 5774806.209
D:\LiDARdata\Forest1\PlotSamples\CP100_02.lda 1974144.127 5774788.406 1974178.127 5774822.406
D:\LiDARdata\Forest1\PlotSamples\CP100_03.lda 1974247.472 5774771.074 1974281.472 5774805.074
D:\LiDARdata\Forest1\PlotSamples\CP100_04.lda 1973930.246 5774662.876 1973964.246 5774696.876
```

Figure 3 – Example text file of plot locations

4) Cut out plots

Cutting out plots is performed with FUSION's clipdata.exe command. For a full description of the various commands see the FUSION manual.

```
C:\FUSION\clipdata.exe /shape:1 /dtm:[dtm
filename] /height [LiDAR filenames text
file] [Plot location text file]
```

This will cut out each plot and save it in the file location specified in the text file. For many plots this may take some time.

5) Generate plot metrics for each plot

This can be done plot by plot using FUSIONs `cloudmetrics.exe` command. Either each plot must be treated separately, or some scripting or batching software is used to loop the process. To analyse each plot type

```
C:\FUSION\cloudmetrics.exe /new
/firstreturn /above:0.5 /htmin:0.5 [Plot
sample lda file] [Plot sample csv file]
```

which will save the metrics for that plot in a csv file named *[Plot sample csv file]*. In this example a height cut-off of 0.5m is used – returns below this are not included in the metrics except for canopy cover.

6) Compile plot metrics and regress with field measurements

By combining the plot metrics from the LiDAR plots with the field-measured variable of interest (e.g. volume), a regression can be performed with any statistics package such as SAS, minitab or R. The resulting function (with its given precision) can be used for developing maps (e.g. volume maps) across the entire LiDAR area at an appropriate scale. A typical function may be of the form

$$V = \alpha P_{30} + \beta C + \gamma \qquad [1]$$

where V = volume per hectare, P_{30} is the 30th height percentile, C is the canopy cover, and α, β, γ are coefficients found in the regression.

7) Determine metrics across the whole LiDAR area

Once a functional form for volume has been found, it can be applied across the whole forest. First though, the metrics of interest should be found across the

LiDAR area by sampling it as a grid which defines the scale of the final map. This can be done by typing into dos

```
C:\FUSION\gridmetrics.exe /diskground
/minht:0.5 [dtm filename] 0.5 30 [gridded
metrics filename] [LiDAR filenames text
file]
```

Note that the number '30' refers to the grid size. It is recommended that the grid size is similar to the plot size, as many LiDAR equations (such as for volume) depend on plot size. A 0.06-ha plot has a diameter of 28 m. This command will produce a csv with a set of metrics for each grid cell. Note that the metrics are similar, but not exactly the same as from `plotmetrics` (and not quite in the same order either). By applying the function derived in step 5 to the variables found in the gridmetrics csv the volume can be found for each *(x,y)* cell. Any graphing tool can then be used to plot the X and Y coordinates (3rd and 4th column) against the calculated volume.

Octave Example

The two scripts in the appendix will automate the above procedure (except for the regression, which is left to the user), and can be run in either Matlab or the free scripting software Octave which can be downloaded from
http://www.gnu.org/software/octave/download.html

As the code is not complex it would not be difficult to port it to any other language. The first script performs steps 1-5 and produces a spreadsheet of LiDAR metrics for each plot, although users must use their own statistics package to develop a meaningful regression for their forest. The second script then performs step 6, and saves the volume map as a geo-referenced image.

In this code we assume that all LAS files are in their own folder, and that a csv sheet D:\LiDARdata\PlotLocations.csv exists with the first five columns being 'Compartment, Plot, Plot Size (ha), Easting, Northing'. Compartment and plot are both numerical values, with no letters, spaces or other non-digits. This script also requires that FUSION is saved in C:\FUSION.

Once the first script has been run, a csv spreadsheet containing all the LIDAR metrics for all the plots (plotmetrics.csv) will be produced. The user can take these, perform a regression against the field measured volume, and derive an equation such as equation [1]. In script two there are two variables, cols and consts, that must be set to match this equation. Cols relates to the columns for the variables of interest in the gridmetrics.csv output (see Table 1), and consts are the scaling parameters (assuming a linear regression). With these variables adapted to suit, a volume map will be produced. This will be displayed on screen and supplied as a tiff and worldfile (tfw) pair (VolMap.tif).

Table 1 – Variable and column numbers in gridmetrics output

Column	Variable	Column	Variable
5	Total return count above 0.50	36	Elev P90
6	Elev minimum	37	Elev P95
7	Elev maximum	38	Elev P99
8	Elev mean	39	Return 1 count above 0.50
9	Elev mode	40	Return 2 count above 0.50
10	Elev stddev	41	Return 3 count above 0.50
11	Elev variance	42	Return 4 count above 0.50
12	Elev CV	43	Return 5 count above 0.50
13	Elev IQ	44	Return 6 count above 0.50
14	Elev skewness	45	Return 7 count above 0.50
15	Elev kurtosis	46	Return 8 count above 0.50
16	Elev AAD	47	Return 9 count above 0.50
17	Elev L1	48	Other return count above 0.50
18	Elev L2	49	Percentage first returns above 0.50
19	Elev L3	50	Percentage all returns above 0.50
20	Elev L4	51	(All returns above 0.50) / (Total first returns) * 100
21	Elev L CV	52	First returns above 0.50
22	Elev L skewness	53	All returns above 0.50
23	Elev L kurtosis	54	Percentage first returns above mean
24	Elev P01	55	Percentage first returns above mode
25	Elev P05	56	Percentage all returns above mean
26	Elev P10	57	Percentage all returns above mode
27	Elev P20	58	(All returns above mean) / (Total first returns) * 100
28	Elev P25	59	(All returns above mode) / (Total first returns) * 100
29	Elev P30	60	First returns above mean
30	Elev P40	61	First returns above mode
31	Elev P50	62	All returns above mean
32	Elev P60	63	All returns above mode
33	Elev P70	64	Total first returns
34	Elev P75	65	Total all returns
35	Elev P80		

Appendix

Octave (or Matlab) Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                          %
%                                                                          %
%                               LiDARdemo1                                 %
%                                                                          %
%                                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%   Generate LiDAR plot metrics using FUSION

clear all
close all

% Switches
heightbreak=0.5;  % cut-off for canopy cover
minht=0.5; % do not include values above this for metrics except cover

% User defined folders
LiDARdir='D:\LiDARdata\Forest1\LAS';
outputpth = 'D:\LiDARdata\Forest1';

%File paths
tempfolder=[outputpth,'\temp'];
plotsfolder=[outputpth,'\plots'];
lasfn=[tempfolder,'\FilesList.txt'];
dtmfn=[outputpth,'\groundDTM.dtm'];
plotlocationsCsv=[outputpth,'\PlotLocations.csv'];
plotMetricsCsv=[outputpth,'\plotmetrics.csv'];

mkdir(tempfolder);
mkdir(plotsfolder);

% Generate a list of all the las filenames
r=dir(LiDARdir);
r(1:2)=[];
lasfnms=cell(numel(r),1);
for i=1:numel(r);
    lasfnms{i}=r(i).name;
end

Nfiles=numel(lasfnms);
disp(['Found ',int2str(Nfiles),' LAS files in folder ',LiDARdir])

% Write text file of LAS filenames
f1 = fopen(lasfn, 'w');
for i=1:Nfiles
    fprintf(f1, '%s\r\n', [LiDARdir,'\',lasfnms{i}]);
end
fclose(f1);

% Make DEM with TerraScan classifiaction
ds = ['C:\FUSION\TINSurfaceCreate.exe /class:2 ',dtmfn,' 1 m m 0 0 0 0 ',LiDARdir,'\*.las'];
dos(ds);

% Cut out plots

% Load up plot data file cooridinates
PlotData = dlmread(plotlocationsCsv,',',1,0);
Compartment=PlotData(:,1);
plotNum = PlotData(:,2);
radius=sqrt((PlotData(:,3)*100^2)/pi);
X = PlotData(:,4);
Y = PlotData(:,5);
nplots=numel(plotNum);
plotname=cell(nplots,1);
```

```
for i=1:nplots
    plotname{i}=[int2str(Compartment(i)),'_',int2str(plotNum(i))];
end

% Build corner list and filename list
fn=[tempfolder,'\PlotList.txt'];
f2 = fopen(fn,'w');
for i=1:nplots
    cnrs=[X(i)-radius(i),Y(i)-radius(i),X(i)+radius(i),Y(i)+radius(i)];
    fprintf(f2,'%s\n',[plotsfolder,'\',plotname{i},'.lda ',num2str(cnrs,11)]);
end
fclose(f2);

% clip data
ds=['C:\FUSION\clipdata.exe /shape:1 /dtm:',dtmfn,' /height ',lasfn,' ',fn];
dos(ds);

% Generate Plot Metrics

% generate metrics into csv files
for i=1:nplots
    ldanm=[plotsfolder,'\',plotname{i},'.lda'];
    csvnm=[plotsfolder,'\',plotname{i},'.csv'];
    ds=['C:\FUSION\cloudmetrics.exe /new /above:',num2str(heightbreak),' /htmin:',num2str(minht),'
',ldanm,' ',csvnm];
    dos(ds);
end

% Compile plot metrics

f3=fopen(plotMetricsCsv,'w');
for i=1:nplots
    csvnm=[plotsfolder,'\',plotname{i},'.csv'];
    f4=fopen(csvnm);
    [a,count]=fscanf(f4,'%c',inf);
    fclose(f4);
    b=find(double(a)==13,1);    % Find line feed
    if i==1
        fprintf(f3,'%s',a(1:end-2));
    else
        fprintf(f3,'%s',a((b+1):(end-2)));
    end
end
fclose(f3);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                            %
%                                                                            %
%                                LiDARdemo2                                  %
%                                                                            %
%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Generate LiDAR volume map using FUSION

clear all
close all

% Switches
gridsize=20;    % What gridsize?
heightbreak=0.5;  % cut-off for canopy cover
minht=0.5; % do not include values above this for metrics except cover

% User defined folders
LiDARdir='D:\LiDARdata\Forest1\LAS';
outputpth = 'D:\LiDARdata\Forest1';
```


```
%File paths
tempfolder=[outputpth,'\temp'];
plotsfolder=[outputpth,'\plots'];
lasfn=[tempfolder,'\FilesList.txt'];
dtmfn=[outputpth,'\groundDTM.dtm'];
plotlocationsCsv=[outputpth,'\PlotLocations.csv'];
plotMetricsCsv=[outputpth,'\plotmetrics.csv'];
GridMetricCsv=[outputpth,'\gridmetrics.csv'];

% Volume equation - columns in gridmetrics.csv and constants
cols=[29,50]; %P30 and %all returns >0.5m
consts=[20,2,0];    %i.e. V = 20*P30 + 2*canopy cover + 0

% Gridmetrics of all data
ds=['C:\FUSION\gridmetrics.exe /diskground /minht:',num2str(minht),' ',dtmfn,' ',num2str(heightbreak),'
',int2str(gridsize),' ',GridMetricCsv,' ',lasfn];
dos(ds);

% Read in gridmetrcis
gmfn=[GridMetricCsv(1:end-4),'_all_returns_elevation_stats.csv'];
hdrfn=[GridMetricCsv(1:end-4),'_all_returns_elevation_stats_ascii_header.txt'];
C=dlmread(gmfn,',',1,0);
griddta=dlmread(hdrfn,' ',0,1);
b=zeros(size(C,1),1);
dta=[C(:,cols),b]';
V=consts*dta;

% Remove undefined areas (-9999)
[i,j]=find(C==griddta(6));
b=unique(i);
V(b)=0;

% convert to a grid
s=griddta(1:2);
r=griddta(5);
V=reshape(V,s(1),s(2));
V=V';
yv=linspace(griddta(3),griddta(3)+r*(s(2)-1),s(2))-r/2;
xv=linspace(griddta(4),griddta(4)+r*(s(1)-1),s(1))-r/2;
figure
imagesc(xv,yv,V); caxis([0 1000]); colormap(gray)

% Write tif and tfw output
V8bit=uint8((V/4));
tfw=[gridsize;0;0;-gridsize;griddta(4);griddta(3)+r*(s(2)-1)];
imwrite(V8bit,[outputpth,'\VolMap.tif'],'tif');
dlmwrite([outputpth,'\VolMap.tfw'],tfw,'precision',11,'delimiter',' ');
```


Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz