Identification of tree seedling with high resolution UAV lidar

NRS 504 Final Project

Anthony Martinez

# Background/Introduction

A critical component of sustainable forest management is the regeneration of trees for the continuance of future forests. To effectively monitor the condition of a forest stand, it is necessary to characterize tree seedlings and saplings. Current methods of seedling quantification involve resource intensive manual seedling surveys. As technology advances, an automated method of quantifying tree regeneration could reduce the cost forest measurements and free up resources for other forest management projects.

Lidar is a technology that can quickly provide precision forest measurements over large areas and it is being used with greater frequency in forestry applications. Lidar has traditionally been collected by mounting sensors on planes. These plane-mounted systems are expensive to own and operate, which makes the data they collect prohibitively expensive for many land managers. Recently, with advances in UAV (unmanned aerial vehicles) capabilities and availability, UAV lidar systems are reaching the market, increasing the accessibility of high-precision data to land managers.

To make UAV lidar data a worthwhile investment for land managers, it is necessary to develop techniques to extract as much information from it as possible. A literature search in both Google Scholar and Web of Science yielded no results on the use of UAV lidar in the detection and characterization of tree seedlings and saplings. This information could be useful to land managers who are interested in forest regeneration, including post-disturbance or post-harvest regeneration. UAV detection and quantification of tree regeneration could reduce the tedious field work involved in seedling counting, and increase the accuracy of these traditional methods, which rely on extrapolating field data, often over large areas.

## Objectives

Identify the locations of tree saplings using high-resolution UAV lidar.

# Methods

## Study area

The study area was in the University of Idaho Experimental Forest on Moscow Mountain, in Latah County, Idaho. The site was located in a recently harvested mixed conifer stand. In addition to conifer seedlings, the forest floor contained a grasses and light logging slash (Figure 1).

|  |
| --- |
|  |
| Figure 1: Aerial image of study area from UAV. |

## Data

### Field data

Field data was collected on September 2, 2017. 40 seedlings were measured, and their GPS locations were recorded using a Topcon GPS system. In addition to their location, seedling species, height, max/min crown width, number of whorls (a proxy for age) and cover class (open, semi suppressed, and suppressed) were recorded.

### Lidar data

The UAV flight mission was performed by Alta Science & Engineering on September 24, 2017. The data was collected using an HDL-32E lidar sensor (Volodyne Acoustics, Inc. Morgan Hill, CA) mounted onto a the 6-rotor Matrice 600 UAV (DJI, Los Angeles, CA). The typical flight time was approximately 20 minutes, at which point memory cards and batteries were exchanged. The sensor collects approximately 700,000 points/second with an accuracy of ± 0.787 in. (2 cm). The average point density within the study area 46.76 points/ft2 (522.88 points/m2), with an average point spacing of 0.15 ft (0.05 m).

## Analysis

Analysis was completed using LAStools, Fusion, and the lidR and rLiDAR R packages; it was visualized using ArcGIS. Although LAStools and Fusion are executed using command line functions, I called these functions through R system calls, so all code could be compiled into a single R script.

### Preprocessing

The lidar data delivered from the vendor in 53 separate and overlapping LAZ files (compressed LAS files (Figure 2a). These files were first merged into a single LAS file, then tiled using LAStools (Figure 2b). From there, the point cloud was clipped to include only the study area (Figure 2c).

The raw point cloud provided by the vendor included point locations for the UAV position for each pulse, which left 12 flight line paths in the LAS data which had to be removed prior to modeling the canopy (Figure 3). A histogram of the point heights was used to identify a threshold above which to filter out points (using lidR::lasfilter).

|  |  |  |
| --- | --- | --- |
| a) | A picture containing text  Description generated with high confidenceb) | A necklace hanging on a wall  Description generated with high confidencec) |
| Figure 2: Preprocessing steps: the original 53 overlapping LAS files (a) were merged and tiled (b) before being clipped to the study location (c). Purple dots in (c) indicate sample tree locations. | | |

|  |
| --- |
|  |
| Figure 3: Example of lidar point cloud profile with UAV position returns in red. An elevation histogram (right) allowed for threshold determination to remove these points. |

### Ground model – Digital Terrain Model (DTM)

Since 1998, when the ground filtering algorithm that Fusion uses was developed (Kraus and Pfeifer 1998), there have been advances in ground filtering methods. Serifoglu et al. (2016) found that the progressive morphological filter (PMF; Zhang et al. 2003) was particularly effective for identifying ground points in high density UAV lidar point clouds. With lidR::lasground, I classified ground points using the PMF method. I then created a 0.25 ft. resolution DTM using these ground points using the k-nearest neighbor inverse-distance weighting method with lidR::grid\_terrain.

### Digital surface model

Using LAStools las2dem, I created a digital surface model (DSM) using the spike-free methodology, developed by Khosravipour et al. (2016). This algorithm preferentially selects lidar points to be used in creating the DSM not only on their x and y location (as has been the standard), but also in the z dimension (Khosravipour et al. 2016). This process reduces the occurrence of “spikes” and “pits” (i.e. sudden, and often unrealistic, high and low points on the DSM, respectively) during the point selection phase in the raw point cloud. Previous algorithms rely on smoothing the DSM rasters to remove spikes and pits, which can reduce the accuracy of the final product. The freeze constraint parameter, was set to 0.45, about 3 times the average point spacing per Isenburg (2016). The final DSM was smoothed using a 0.75 ft. Gaussian smooth (rLiDAR::CHMsmoothing) to help reduce noise.

### Tree detection

I first attempted to detect tree tops using local maxima filters on the LAS point cloud itself (lidR::tree\_detection). I later used the same command to detect tree tops using local maxima filters on the smoothed DSM. I decided to use the DSM as opposed to the CHM (ground normalized DSM) for the tree detection because Khosravipour et al. (2015) found that attempting to detect trees with a CHM could lead to incorrect tree identification, especially areas of complex topography. I used a widow size set to 6 ft., the approximate maximum crown width in the sample data; I used a minimum tree height of 1 ft., the approximate minimum sapling height in the sample data. I approximated these predicted tree heights by ground-normalizing the DSM at the tree top locations.

|  |
| --- |
|  |
| Figure 4: When more than one predicted tree (yellow) fell with the buffer (blue) around the sampled tree (green), the predicted tree with the height closed to sampled tree was marked as the correctly identified tree. |

### Accuracy assessment

To assess correct identification of seedlings, the locations of the sampled trees were compared to the locations of the predicted trees. To account for the spatial error in the tree locations, field sampled trees were buffered (1 ft,. 2 ft., …, 5 ft.) and predicted trees were tested whether they were within the buffer. In cases where a predicted tree fell with the buffer of more than 1 tree, it was assumed to be correctly identified as the sampled tree with the closest height (Figure 4). Predicted trees that were identified as matching one of the sampled trees were marked as true positive predictions.

100 points were randomly generated throughout the study area. These points were visually assessed using high resolution imagery for the presence or absence of a tree. The first 40 absence points were then selected. Using the same process described above, these points were buffered and predicted trees were tested whether they were within the buffer. The predicted trees within the “absence” buffer were marked as false positive predictions.

# Results

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 5: The three primary outputs: (1) the DTM, from brown to blue; (2) the DSM, from blue to red; and (3) the DSM with the 2080 predicted tree locations. | | |

There final product was the location of all the predicted trees, and the intermediate products were the DTM and the DSM (Figure 5).

|  |
| --- |
|  |
| Figure 6: Distribution of predicted sapling heights (for trees < 9 ft.). |

After trying several different parameters in Fusion’s ground model functions and having low quality DTMs, I did more research and found that the PMF method may be the best method to use ground filtering UAV lidar points. This, coupled with the k-nearest neighbor method of rasterizing, resulted a high quality DTM with no visible artifacts such as trees or logs.

There were also many failed attempts when creating a DSM. The canopy height model functions in lidR and Fusion resulted in “pitted” DSMs. The spike-free method resulted in a smooth and realistic surface. I did still decide to smooth the final raster, which reduced the number of trees predicted from 4000+ to 2080.

|  |  |
| --- | --- |
| Table 1: Number of correctly predicted seedlings within various sized buffers | |
| **Buffer radius (ft)** | **Correctly predicted saplings** |
| 1 | 1 |
| 2 | 1 |
| 3 | 7 |
| 4 | 13 |
| 5 | 16 |

|  |
| --- |
|  |
| Figure 7: Distribution of height differences between ground sampled seedlings and their correctly identified analogues. |

2,080 trees were identified with lidR::tree\_detection. Although this includes full sized trees, there were relatively few of these on the site. A large proportion of the predicted tree heights were less than 2 ft. (Figure 6), although there was some error introduced into these heights during the ground normalization process. With a 5-ft. radius around sampled trees, there were as 16 correctly identified seedlings. The difference in height between the correctly predicted seedlings and their sampled counterparts was generally about one ft. (Figure 7).

# Discussion and conclusion

Overall, there is a good start to producing meaningful results (Figure 8), but I think it will take quite a bit of work to get there. These methods seemed to have overpredicted the number of trees in the study area, which is not very surprising given the presence of grass and logging slash on the site.

One difficulty in assessing the accuracy of the tree detection method is the spatial error of the sampled tree locations and the UAV point cloud. Predicted tree/observed tree mismatches may be because of modeling error; however, it could also be a correctly identified tree, but the locations just do not align.

I suspect the low detection accuracy was primarily caused by two factors. First, is the spatial error of the two datasets, which would be compounded when comparing them against one another. Second, ground at the study site was vegetated by tall grass and logging slash. This biomass added a great deal of noise to the point cloud, obscuring tree seedlings and potentially resulting in a high number of false positives.

Further research may be able to overcome these challenges. I suspect that including the lidar pulse return intensity in the analysis may yield improved results. Because photosynthetically active vegetation reflects a higher amount of NIR light, this may help in distinguishing between logging slash and dry grass and conifer seedlings. Additionally, increasing the number of tree seedlings sampled may also help in adjusting the parameters and improve the accuracy assessments. With additional time, I think I could improve these by studying and understanding the algorithms that underly these methods. With a better understanding of how they work, I could then optimize the input parameters without having to guess-and-check, which is very time consuming, as many of these processes can take hours to run.

|  |
| --- |
|  |
| Figure 8: Example comparison of predicted tree locations and sampled tree locations using the widest buffer. Correctly predicted tree symbol increases with its closeness to the sampled tree (i.e. the nearer the correctly predicted tree is to the sampled tree, the larger its symbol). |

# References

Isenburg M (2016) Generating Spike-Free Digital Surface Models from LiDAR. https://rapidlasso.com/2016/02/03/generating-spike-free-digital-surface-models-from-lidar/

Khosravipour A, Skidmore AK, Isenburg M (2016) Generating spike-free digital surface models using LiDAR raw point clouds: A new approach for forestry applications. Int J Appl Earth Obs Geoinf 52:104–114. doi: 10.1016/j.jag.2016.06.005

Khosravipour A, Skidmore AK, Wang T, et al (2015) Effect of slope on treetop detection using a LiDAR Canopy Height Model. ISPRS J Photogramm Remote Sens 104:44–52. doi: 10.1016/j.isprsjprs.2015.02.013

Kraus K, Pfeifer N (1998) Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS J Photogramm Remote Sens 53:193–203. doi: 10.1016/S0924-2716(98)00009-4

Serifoglu C, Gungor O, Yilmaz V (2016) Performance evaluation of different ground filtering algorithms for uav-based point clouds. Int Arch Photogramm Remote Sens Spat Inf Sci - ISPRS Arch 2016–Janua:245–251. doi: 10.5194/isprsarchives-XLI-B1-245-2016

Zhang K, Chen SC, Whitman D, et al (2003) A progressive morphological filter for removing nonground measurements from airborne LIDAR data. IEEE Trans Geosci Remote Sens 41:872–882. doi: 10.1109/TGRS.2003.810682

# Acknowledgements

Thank you to Carlos Silva and Arjan Meddens for pointing me in the right direction as I hit several snags along the way.

# **Appendix A: R code used for analyzing data**.

|  |
| --- |
| #=======================================================================#  #=======================================================================#  # Using UAV lidar to identify conifer seedlings #  # NRS 504 Final Project #  # Anthony Martinez #  #=======================================================================#  #=======================================================================#  # Set directories  mainDir <- "S:/COS/PyroGeog/amartinez/UAV\_seedlings/Lidar/LAS"  setwd(mainDir)  #=======================================================================#  # Inital preparation (Fusion and LAStools)  #=======================================================================#  # Merge LAZ files into a single point cloud  system(paste(file.path("C:","Fusion", "mergedata.exe"),  "S:\COS\PyroGeog\amartinez\UAV\_seedlings\Lidar\fullLAZ\filelist.txt",  file.path(mainDir, "Seedling\_merge.las"),  sep = " "))  # Tile pointcloud into tiles of 2 million points  system(paste(file.path("C:","Fusion", "lastile.exe"),  "-i", file.path(mainDir, "Seedling\_merge.las"),  "-o", file.path(mainDir, "Tiles", "tile.las"),  "-refine 20000000", "-cores 6",  sep = " "))  system(paste("dir/s/b",  file.path(mainDir, "Tiles", "\*.las"),  ">", file.path(mainDir, "TileList.txt"),  sep = " "))  # Clip tiles to seedling area  system(paste(file.path("C:","Fusion", "clipdata.exe"),  "/shape:0 /index",  file.path(mainDir, "TileList.txt"),  file.path(mainDir, "MoscowMtn\_clip.las"),  "2380665 1893999 2381231 1894722",  sep=" "))  # View LAS information using LAStools and Fusion  system(paste(file.path("C:","LAStools", "bin", "lasinfo.exe"),  "-i", file.path(mainDir, "LAS\_norm.las"),  "-last\_only",  "-cd",  sep = " "))  system(paste(file.path("C:","Fusion", "catalog.exe"),  "/density:0.125,5,10",  file.path(mainDir, "MoscowMtn\_clip.las"),  sep=" "))  #=======================================================================#  # Create ground model -- 0.25 ft (3 in) resolution  #=======================================================================#  # Load packages  library(rLiDAR)  library(lidR)  library(raster)  library(rgeos)  library(rgdal)  library(data.table)  # Import LAS  las <- lidR::readLAS(file.path(mainDir, "MoscowMtn\_clip.las"))  # Identify ground points (takes approx. 20 hours to run on this dataset)  dem.las <- grid\_terrain(las, res = 0.25, method = "knnidw", k = 10, p = 2,  keep\_lowest = FALSE)  dem.raster <- as.raster(dem.las)  writeRaster(dem.raster, file.path(mainDir, "Ground", "pmf\_dem.tif"))  dem.raster <- raster(file.path(mainDir, "Ground", "pmf\_dem.tif"))  #=======================================================================#  # Create canopy model -- 0.25 ft (3 in) resolution  #=======================================================================#  # Remove power lines  hist(las@data$Z) # identify cutoff threshold  las.pwr.rm <- lasfilter(las, Z < 3050)  writeLAS(las.pwr.rm, file.path(mainDir, "LAS\_pwr\_rm.las"))  las.pwr.rm <- lidR::readLAS(file.path(mainDir, "LAS\_pwr\_rm\_class.las"))  # Create "spike free" canopy height model (with LAStools)  system(paste(file.path("C:","LAStools", "bin", "las2dem.exe"), # CHM with normalized LAS  "-i", file.path(mainDir, "LAS\_pwr\_rm.las"),  "-spike\_free 0.45", # freeze distance: ~ 3x the average pulse spacing  "-step 0.25",  "-o", file.path(mainDir, "chm", "CHM\_spike\_free.asc"),  sep=" "))  chm.sf <- raster(file.path(mainDir, "chm", "CHM\_spike\_free.asc"))  chm.sf <- CHMsmoothing(chm.sf, filter = "Gaussian", ws = 3, sigma = 0.5)  writeRaster(chm.sf, file.path(mainDir, "chm", "CHM\_spike\_free\_sm.asc"))  chm.sf <- raster(file.path(mainDir, "chm", "CHM\_spike\_free\_sm.asc"))  #=======================================================================#  # Locate trees  #=======================================================================#  # Identify tree tops -- 6 foot radius, 12 inch minimum seedling  tree.top.ras <- tree\_detection(chm.sf, ws = 25, hmin = 1)  tree.top.pts <- rasterToPoints(tree.top.ras, spatial = T)  tree.top <- as.data.frame(rasterToPoints(tree.top.ras, spatial = F))  colnames(tree.top) <- c("x", "y", "id")  shapefile(tree.top.pts, filename = file.path(mainDir, "Trees", "tree\_tops.shp"),  overwrite = T)  # Classify trees in point cloud  #lastrees\_silva(las.pwr.rm, chm.sf, tree.top, max\_cr\_factor = 0.8)  #writeLAS(las.pwr.rm, file.path(mainDir, "LAS\_pwr\_rm\_class.las"))  # Determine tree heights  chm.norm <- chm.sf - dem.raster  writeRaster(chm.norm, file.path(mainDir, "chm", "chm\_norm.tif"), overwrite = T)  chm.norm <- raster(file.path(mainDir, "chm", "chm\_norm.tif"))  tree.top.pts@data$ht <- extract(chm.norm, tree.top.pts)  ############################################################################  # Sampling  ############################################################################  # Import sampled trees  tree.samp <- readOGR("S:/COS/PyroGeog/amartinez/UAV\_seedlings/Lidar/GIS/SampledTrees.shp", stringsAsFactors = FALSE)  proj4string(tree.top.pts) <- proj4string(tree.samp)  SapID <- function(x, y, z){  buff <- gBuffer(y, byid = T, width = z)  a <- gContains(buff, tree.top.pts, byid = T)  b <- as.data.frame(which(a, arr.ind = T))  colnames(b) <- c("tree.top.id", "sample.no")  b$sample.name <- y$OBJNAME[b[,2]]  b$tree.top.ht <- x@data$ht[b$tree.top.id]  b$sample.ht <- y@data$Ht\_\_in\_[b$sample.no]/12  b$diff <- abs(b$tree.top.ht - b$sample.ht)  b <- data.table(b)  b <- b[ , .SD[which.min(diff)], by = tree.top.id]  assign(paste0("tree.samp.", z), tree.samp[unique(b$sample.no),], envir = .GlobalEnv)  assign(paste0("tree.top.", z), x[b$tree.top.id,], envir = .GlobalEnv)  data.frame(b)  }  SapID(tree.top.pts, tree.samp, 1)  SapID(tree.top.pts, tree.samp, 2)  SapID(tree.top.pts, tree.samp, 3)  SapID(tree.top.pts, tree.samp, 4)  SapID(tree.top.pts, tree.samp, 5)  ############################################################################  # END #  ############################################################################ |