Lab 10: Comparison of Lidar

Anthony Stewart

2023-05-30

Objectives

- Comparison of lidar
    - Cloud Metrics / Rumple / Tree Segmentation between ALS & TLS
    - Cloud Metrics and RH between ALS and GEDI

- Data and Software:
    - LAB10 data
        - ALS – From the WA DNR lidar portal
        - TLS (included in zip file)
            - 2019-08-18 Pack Forest TMLS.laz
            - 2019-08-22 Hall of Mosses TMLS.laz
        - GEDI – (included in zip file)
    - RStudio
    - CloudCompare
    - ArcGIS Pro

- What you will turn in:
    - **You will submit a report with several required figures included.**

Introduction

So we've made it this far into the quarter and it's time for the last lab. There is not final exam but this lab will consist of many of the past lessons from the previous 9. By now you should have a repository of code and lidar processing examples in the form of your past assignments. Most of this final lab comes from those past assignments and reuses their code. So there will be more limited examples here meaning less copy/paste.

The data in this lab also consists of past lab examples except for some extra TLS data from Pack Forest and the Hall of Mosses in the Hoh Rainforest: `2019-08-18 Pack Forest TMLS.laz` & `2019-08-22 Hall of Mosses TMLS.laz`. You will need to download extra data from the lidar portal in Part 1.

Part 1: ALS & TLS comparison processing steps with required figures

You will have 2 plots for this part centered at:

- Pack Forest - 1185950 555180 (EPSG 2927) / -122.31625, 46.84141 (WGS 84):

  https://lidarportal.dnr.wa.gov/#46.84141:-122.31625:15

- ONP - 797785 941189 (EPSG 2927) / -123.93314, 47.86372 (WGS 84):

  https://lidarportal.dnr.wa.gov/#47.86372:-123.93314:15

Download the ALS point clouds at those two locations from the Washington DNR lidar portal website.

- Read in the two ALS laz files and the two TLS laz files into R
    - ALS files (you downloaded from the DNR) (Assuming Pack 2013)
        - WA_031_rp.laz
        - q47123g8201.laz
    - TLS files (provided via canvas):
        - 2019-08-18 Pack Forest TMLS.laz
        - 2019-08-22 Hall of Mosses TMLS.laz

```
PFALS <- readLAS()

HMALS <- readLAS()


PFTLS <- readTLSLAS()

HMTLS <- readTLSLAS()
```

- Define the `crs` in R, we need them all to be in the same crs

```
crs(PFTLS) <- NEWCRS_FILL_IN_YOURSELF

crs(HMTLS) <- NEWCRS_FILL_IN_YOURSELF

crs(PFALS) <- NEWCRS_FILL_IN_YOURSELF

crs(HMALS) <- NEWCRS_FILL_IN_YOURSELF
```

- `clip_circle` the files to the plot extent:
  - Pack Forest ALS & TLS (epsg 2927)
  - x = 1185950, y = 555180, radius = 50
 - ONP ALS & TLS (epsg 2927)
  - x = 797785, y = 941189, radius = 50

```
?clip_circle
PFTLSc <- clip_circle()
HMTLSc <- clip_circle()
PFALSc <- clip_circle()
HMALSc <- clip_circle()
```

Future steps for the TLS will only use these clips

`writeLAS` the four lidar clips and bring them into **CloudCompare** to produce a figure.

```
?writeLAS
writeLAS(PFTLSc)
writeLAS(HMTLSc)
writeLAS(PFALSc)
writeLAS(HMALSc)
```

REQUIRED FIGURES: An image of the ALS and TLS plotted together for both the Pack Forest (PF) and Hall of Mosses (HM) sites (two screen shots). Example given has the TLS cloud in gray scale and the ALS cloud in multicolor for the Hall of Mosses forest plot.
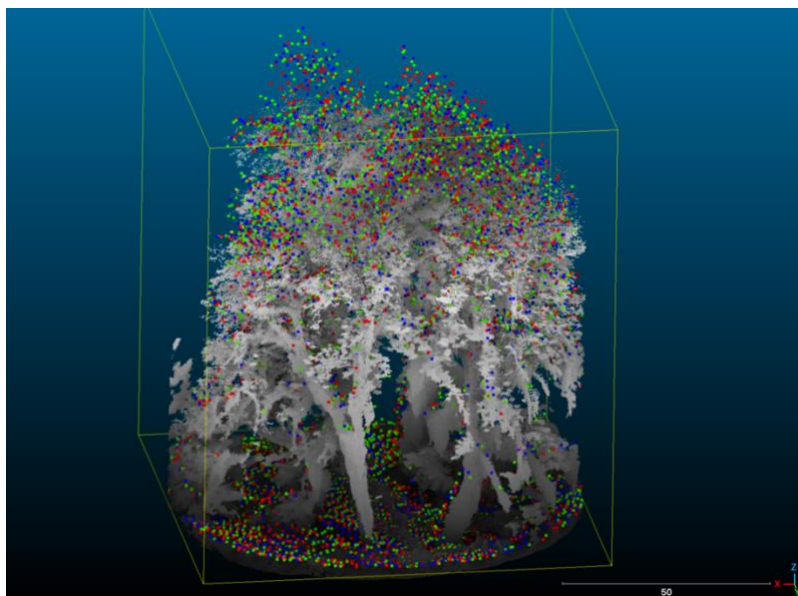
Now back in R:

- Classify the points in both TLS point clouds using the either pmf or csf and the classify_ground command.

```
mycsf <- csf()

OR

mypmf <- pmf()


PFTLSg <- classify_ground(PFTLSc, _____)


HMTLSg <- classify_ground(HMTLSc, mycsf)
```

- Use rasterize_terrain to make DTMs for both the TLS point cloud clips

```
PFDTM1 <-grid_terrain(PFTLSg, res = _____, algorithm = _____)


HMDTM1 <-grid_terrain(HMTLSg, res = _____, algorithm = _____)
```

- plot_dtm3d to make sure your DTMs are good quality. You will likely need to try different values in the production of the DTMs.

```
plot_dtm3d(PFDTM1)
plot_dtm3d(HMDTM1)
```

REQUIRED FIGURES: Two screen shots of your two produced dtms. Fully captioned with the technique used to make them.

- Normalize your point clouds using the DTMs from the TLS.
  - `normalize_height()`
  - Use the Pack forest TLS DTM to normalize both the ALS and TLS clips from Pack forest
  - Use the ONP TLS DTM to normalize both the ALS and TLS clips from ONP.

```
PFTLScN <- _____(PFTLSg,_____)

HMTLScN <- _____(HMTLSg,_____)

PFALScN <- _____(PFALSg,_____)

HMALScN <- _____(HMALSg,_____)
```

- Create a DSM for all four clips using `rasterize_canopy`, Your choice what algorithm to use.
  - Plot each DSM and grab screen shots. You will be including them in the report

```
?rasterize_canopy

PFTLS_DSM <- rasterize_canopy()

PFALS_DSM <- rasterize_canopy()

HMTLS_DSM <- rasterize_canopy()

HMALS_DSM <- rasterize_canopy()
```

- Generate the rumple_index for each of the four DSMs.

```
?rumple_index

rumple_index()

rumple_index()

rumple_index()

rumple_index()
```

REQUIRED FIGURES: Four screen shots. One of each of your produced DSMs with the rumple index included in the caption along with the algorithm used to produce the DSM.
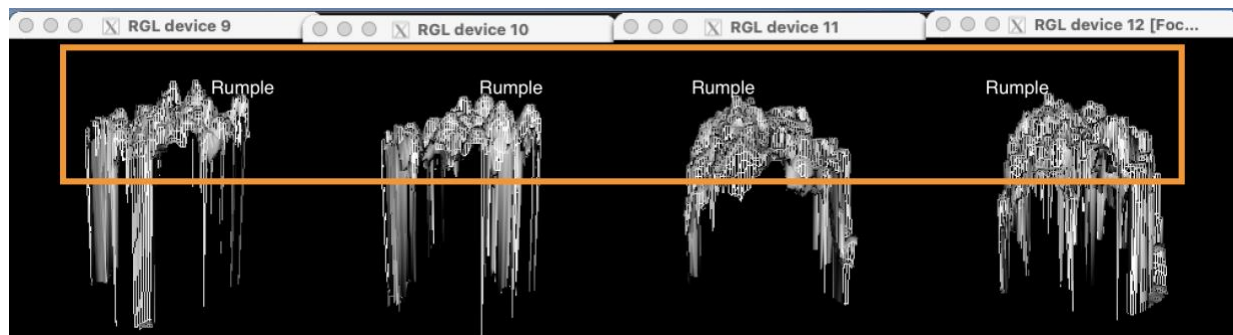

Use `focal` statistics to smooth your DSM. You can do this step before generating the screenshots and rumple index but the smoothing may cause an error with the rumple index.

```
?focal
PFTLS_DSMs <- focal(_____, w = _____, fun = _____)

PFALS_DSMs <- focal(_____, w = _____, fun = _____)

HMTLS_DSMs <- focal(_____, w = _____, fun = _____)

HMALS_DSMs <- focal(_____, w = _____, fun = _____)
```

- segment_trees using the normalized las clips and the smoothed DSMs.

```
?segment_trees
PFTLStrees <- segment_trees()


PFALStrees <- segment_trees()


HMTLStrees <- segment_trees()


HMALStrees <- segment_trees()
```

REQUIRED FIGURES: Four screen shots: for the ALS & TLS in each plot of the lidar data colored by the tree segmentation for each of the normalized lidar clips. Fully captioned with the algorithms used to create the tree segmentation.
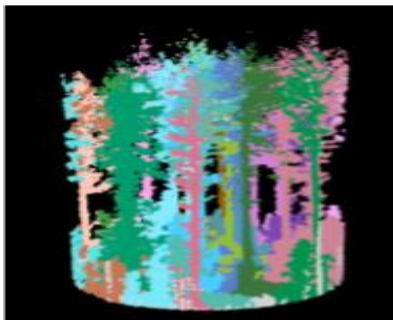
Calculate cloud metrics on the normalized point cloud for the ALS and TLS above 2m height (6.56ft)

```
?cloud_metrics

PFTLS_cm <- cloud_metrics((filter_poi(_____,___ > 6.56)), ~_____)

PFTLS_cm

PFALS_cm <- cloud_metrics()

PFALS_cm

HMTLS_cm <- cloud_metrics()

HMTLS_cm

HMALS_cm <- cloud_metrics()

HMALS_cm
```

- **Fill in the table below with cloud_metrics from each of the clips. Remember, we only want to use points above 2m for the cloud metrics. Deriving canopy cover values is required and you must report the cloud metrics for zmax, zmean, zsd, zq25, zq95**

name	Zmax	Zmean	Zsd	Zq25	Zq95	CanopyCover
Pack Forest ALS						
Pack Forest TLS						
ONP ALS						
ONP TLS						

That is all for the processing of the ALS/TLS plots

Part 2: ALS & GEDI comparison processing steps with required figures for the report

This part is dealing with some more recent coding so I'll provide it here. The questions will refer to the outputs

You will be comparing ALS cloudmetrics and Canopy Height Models (CHMs) to GEDI for the Hall of Mosses area of the Olympic National Park

For the purpose of this lab, we are going to assume that there is no error in the GEDI pulse locations. In reality there is some error and that can account for oddities in the data.

To complete this portion, of the lab you will need to:

Download The DTM and DSM from the Olympic Peninsula
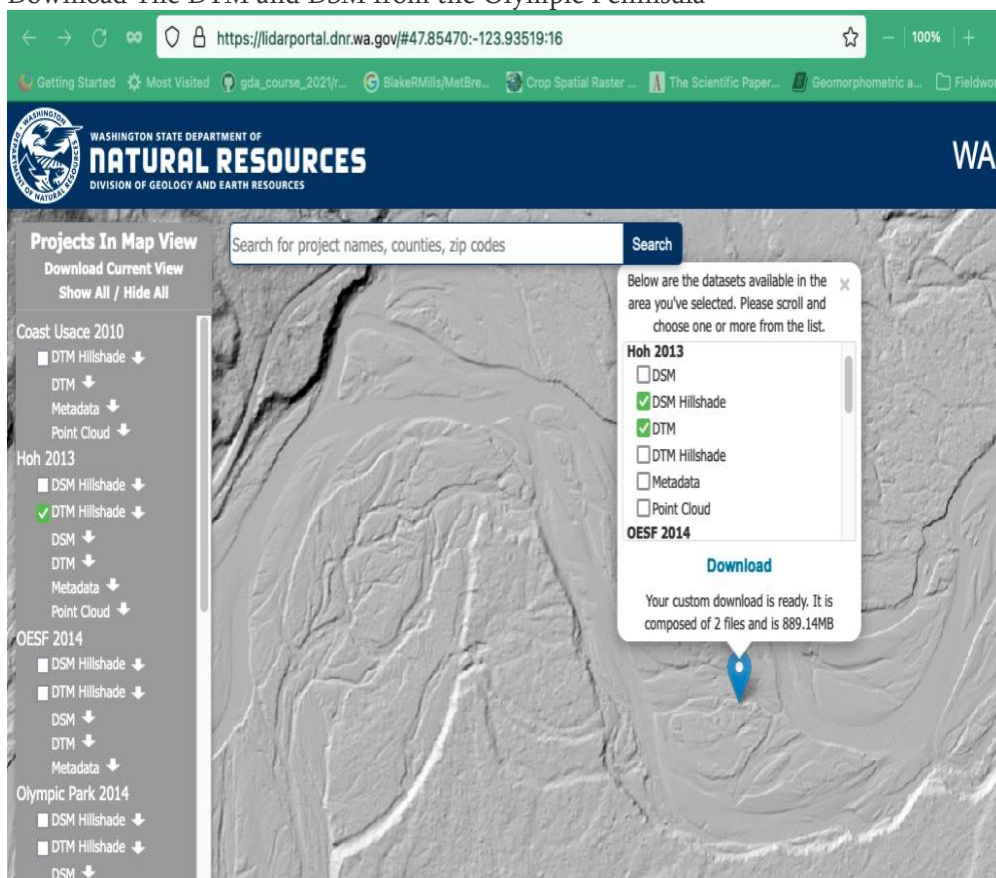
Read them into R and assign the ALS the correct crs

```
ONPDTM <- rast('...................hoh_2013_dtm_4.tif')
ONPDSM <- rast('...................hoh_2013_dsm_4.tif')
```

Subtract the DTM from the DSM for an easy Canopy Height Model

- save the file for later

```
ONPCHM <- ONP_DSM-ONP_DTM

plot(ONPCHM)


writeRaster(ONPCHM, "....................ONPCHM.tif")
```

Now read in GEDI level 2a data from the data folder for this lab

```
?getLevel2AM

gedilevel2a<-readLevel2A(level2Apath = file.path ("...................HM_level2a_clip.
h5"))
```

Compute a series of statistics of GEDI Level2A RH100 metric

```
#Get GEDI Elevation and Height Metrics (GEDI Level2A)

?getLevel2AM

level2AM<-getLevel2AM(gedilevel2a)

head(level2AM[,c("beam","shot_number","elev_highestreturn","elev_lowestmode","rh100")]
)
```

Convert shot_number as "integer64" to "character" and the make the GEDI Level2AM points into a spatial object with sf

```
level2AM$shot_number<-as.character(level2AM$shot_number)

level2AM_sf<-st_as_sf(level2AM, coords = c("lon_lowestmode", "lat_lowestmode"), crs =
4326)
```

Define a function for computing GEDI stats

```
mySetOfMetrics = function(x)

{

  metrics = list(

    min =min(x), # Min of x

    max = max(x), # Max of x

    mean = mean(x), # Mean of x

    sd = sd(x)# Sd of x

  )

  return(metrics)
```

```
}
```

Create a RH100 raster with 0.005 res in degrees of lat / long.

- For the required figures you can use the plotting code below but it's difficult to get good interpretation in R. I would use ArcGIS for visualization after we export

```
?gridStatsLevel2AM

rh100metrics<-gridStatsLevel2AM(level2AM = level2AM, func=mySetOfMetrics(rh100), res=0
.005)

crs(rh100metrics) <- "EPSG:4326" #same as GEDI

rh100metrics_prj <- terra::project(rast(rh100metrics), "EPSG:2927") #transform to WA s
tate plane south


plot(ext(770000, 850000, 900000, 970000), type = "lines", buffer = TRUE)

plot(rh100metrics_prj$max, axes = F, add = TRUE, legend =T)


plot(ext(770000, 850000, 900000, 970000), type = "lines", buffer = TRUE)

plot(ONPCHM, axes = F, add = TRUE, legend =T)


writeRaster(rh100metrics_prj, "............rh100metrics_prj.tif")
```

REQUIRED FIGURES: Create 2 figures: a map from the ONPCHM and a map of the same area using the GEDI RH100 max data. Write out the ONPCHM and GEDI RH100 to open in ArcGIS Pro and put them next to each other for comparison and qualitatively assess the spatial patterns. In your caption explain if they generally agree or disagree with each other

Extract ONPCHM elevation values with the new projected GEDI Level2A points

- use `bind=TRUE` for combining extracted elevation values to the points dataframe
- remember this variable for later modeling

```
level2AGeo_sf_prj <- st_transform(level2AM_sf, "EPSG:2927") #transform to WA state pla
ne south

ONP_CHM_GEDI<- (terra::extract(ONPCHM, level2AGeo_sf_prj, bind = TRUE))

ONP_CHM_GEDI_FILTER <- ONP_CHM_GEDI[-c(is.na(ONP_CHM_GEDI$hoh_2013_dsm_4)),] #removing
NA values

ONP_CHM_GEDI_FILTER <- ONP_CHM_GEDI_FILTER[ONP_CHM_GEDI_FILTER$rh100 > 0,] #lets remov
e the weird GEDI rh100 values that are 0

mapview(st_as_sf(ONP_CHM_GEDI_FILTER))
```

Create a regression model between GEDI and ALS CHM

```
reg <- lm(ONP_CHM_GEDI_FILTER$hoh_2013_dsm_4 ~ (ONP_CHM_GEDI_FILTER$rh100))

summary(reg)

plot((ONP_CHM_GEDI_FILTER$rh100), ONP_CHM_GEDI_FILTER$hoh_2013_dsm_4 , xlab="GEDI rh10
0", ylab="ALS CHM (ft)",

    main="Veg Height")

abline(reg, col="blue")
```

REQUIRED FIGURES: First screenshot of the filtered GEDI points using mapview and second, a screenshot of the final regression model.

PUTTING IT ALL TOGETHER

In total for the lab so far you should have:

- 2 Figures from cloud compare for PF and HM TLS & ALS
- 2 Figures of 3D DTMs
- 4 Figures of DSMs with Rumple
- 4 Figures of TLS & ALS segmented Trees
- 1 Table of cloud metrics
- 2 Figures for the ONP CHM and RH100 rasters
- 2 Figures for the GEDI points in mapview and the final regression plot

Now for the last part of the lab

ESRM 433:
All your figures and tables need to be fully captioned and submitted in one PDF document. All figures need to be high quality and captions need to describe in brief what the image is of, what data was used, and how it was created.
In addition please explain in 300 words or less:

- What is lidar?
- What is ALS?
- What is TLS vs MLS?
- What is GEDI?
- How has lidar been used for quantifying forest structure and topography?
    - Feel free to include figures from your past labs as examples.
    - Include citations of some of the optional reading from the class or some of the papers used in the lidR functions.

SEFS 533:
You will include all of the required figures and tables as well as the short essay, same as 433.
In addition, please explain in 300 words or less:

- Where are your areas of interest that led you take the lidar class?
- Will lidar or lidar derived data help your work?
- Find and summarize one research publication or government report that uses lidar. How would you adapt their analysis or use of lidar for your own work?