Analysis of Drone Data of Small Elephant Impact Sites

Sunniva McKeever, Isabella Metz and Maximilian Merzdorf

Internship Kruger WS23/24

Data Import

Load the required libraries.

```
library(terra)
library(ggplot2)
library(lidR)
library(mapview)
library(sf)
library(knitr)
```

Set working directory and hyperparameters and import drone data.

```
setwd("C://Users/avinn/Documents/Master/Semester3/ElephantTransects/")

## (only once!!!) create empty data frame
params_df <- data.frame(matrix(ncol = 7, nrow = 0))
colnames(params_df) <- c("name",
                         "numtrees",
                         "treedens",
                         "treeheight_min",
                         "treeheight_max",
                         "treeheight_mean",
                         "canopyarea"
                         )

params_df_indices <- data.frame(matrix(ncol = 4, nrow = 0))
colnames(params_df_indices) <- c("ndvi_mean",
                                 "evi_mean",
                                 "gci_mean",
                                 "lai_mean")


# hyperparameters
mw_size <- 15
crs_epsg <- "epsg:32736"
area <- 0.11 # area of each EIA in km2
EIA_name <- "EIA2 C1"

# load drone data
# EIA2C1
```

```r
DSM <- terra::rast("./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_dsm.tif")
DTM <- terra::rast("./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_dtm.tif")
Ortho <- terra::rast(c(
  "./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_transparent_mosaic_group1.tif",
  "./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_transparent_mosaic_green.tif",
  "./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_transparent_mosaic_red.tif",
  "./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_transparent_mosaic_red edge.tif",
  "./ElephantTransectSites/Pix4d/20230810_EIA2_C1/20230810_EIA2C1_transparent_mosaic_nir.tif"),
  lyrs = c(1,2,3,5,7,9,11)
)

# rename bands
names(Ortho) <- c("red", "green", "blue", "MSgreen", "MSred", "MSrededge", "MSnir")

# load other data
aoi <- st_read("./data/other/polygons.shp")
```

```
## Reading layer 'polygons' from data source
##   'C:\Users\avinn\Documents\Master\Semester3\ElephantTransects\data\other\polygons.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 30 features and 3 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 31.29662 ymin: -25.14499 xmax: 31.89291 ymax: -24.9575
## Geodetic CRS:  WGS 84
```

Data Preprocessing

The data needs to be reprojected (as lidR package requires a projection in meters) and cropped to the extent of the EIA. Then, an absolute tree height raster is calculated by subtracting the DTM from the DSM.

```r
## DATA PREPROCESSING
# reproject
# lidR package requires projection in m
DSM <- terra::project(DSM, crs_epsg)
```

```
## |---------|---------|---------|---------|=========================================
```

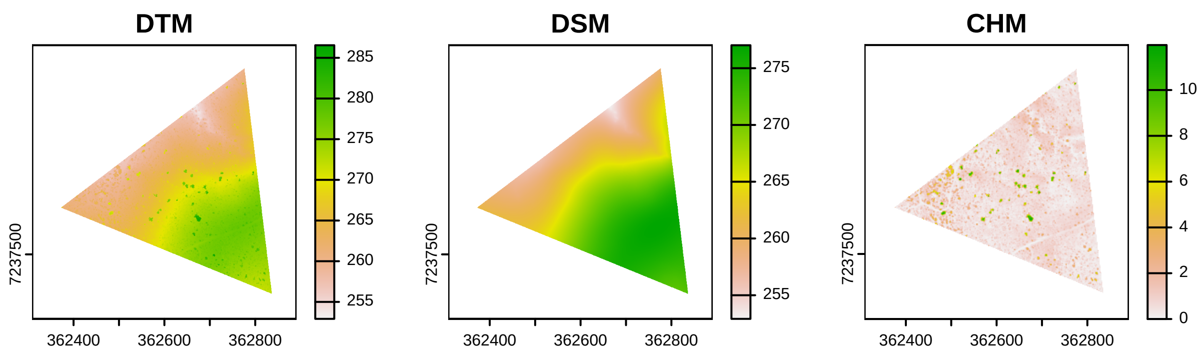
```r
DTM <- terra::project(DTM, crs_epsg)
aoi <- sf::st_transform(aoi, crs = crs_epsg)

# crop
DSM <- terra::mask(DSM, aoi[aoi$FieldID == EIA_name,])
```

```
## |---------|---------|---------|---------|=========================================
```

```r
DTM <- terra::mask(DTM, aoi[aoi$FieldID == EIA_name,])
Ortho <- terra::mask(Ortho, aoi[aoi$FieldID == EIA_name,])
```

```
## |---------|---------|---------|---------|=========================================
```

```r
# calculate Canopy Height Model (CHM) from DSM and DTM
DSM <- resample(DSM, DTM)
CHM <- DSM - DTM
CHM <- aggregate(CHM, 10) # lower resolution to limit computational time

# plot
par(mfrow = c(1,3))
plot(DSM, main = "DTM")
plot(DTM, main = "DSM")
plot(CHM, main = "CHM")
```



Analysis

Basic indices are calculated to compare the structure of different EIAs among each other. Using the lidR package, individual trees are then detected and segmented. Tree tops can be detected by applying a Local Maximum Filter (LMF) on the loaded data set.For a given point, the algorithm analyzes neighborhood points, checking if the processed point is the highest. The size of the moving window determines the size of the analysed neighborhood.

```r
# calculate indices
ndvi <- (Ortho$MSnir - Ortho$red) / (Ortho$MSnir + Ortho$red)
```

```
## |---------|---------|---------|---------|=========================================
```

```r
evi <- 2.5 * ((Ortho$MSnir - Ortho$red) / (Ortho$MSnir + 6 * Ortho$red - 7.5 * Ortho$blue + 1))
```

```
## |---------|---------|---------|---------|=========================================
```

```r
gci <- (Ortho$MSnir / Ortho$MSgreen) - 1
```

```
## |---------|---------|---------|---------|=========================================
```

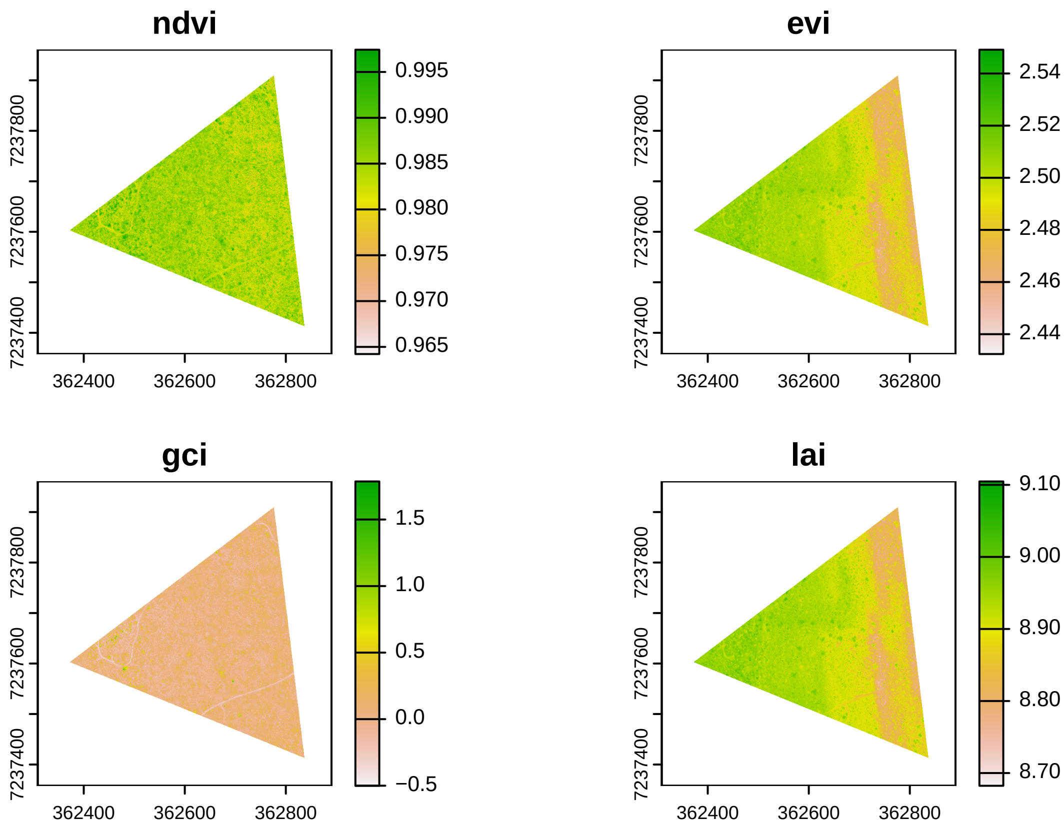
```r
lai <- 3.618 * evi - 0.118
```

```
## |---------|---------|---------|---------|=========================================
```

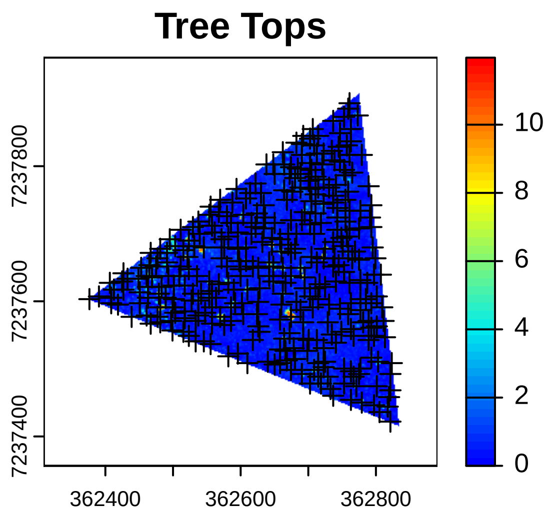
```r
# locate tree tops
# a tree in savannah is everything > 1.5m
ttops <- locate_trees(CHM, lmf(ws = mw_size, hmin = 1.5))

# segment trees
algo <- lidR::dalponte2016(CHM, ttops)
crowns <- algo()

# calculate parameters
numtrees <- round(nrow(ttops), digits = 2)
treedens <- round(numtrees/area, digits = 2) # number of trees per km2
treeheight_min <- round(min(ttops$Z), digits = 2)
treeheight_max <- round(max(ttops$Z), digits = 2)
treeheight_mean <- round(mean(ttops$Z), digits = 2)
canopyarea <- terra::expanse(crowns) # crown area in m2
ndvi_mean <- terra::global(ndvi, 'mean', na.rm = T)
evi_mean <- terra::global(evi, 'mean', na.rm = T)
gci_mean <- terra::global(gci, 'mean', na.rm = T)
lai_mean <- terra::global(lai, 'mean', na.rm = T)

canopyarea <- round(canopyarea, digits = 2)
ndvi_mean <- round(ndvi_mean, digits = 2)
evi_mean <- round(evi_mean, digits = 2)
gci_mean <- round(gci_mean, digits = 2)
lai_mean <- round(lai_mean, digits = 2)
```

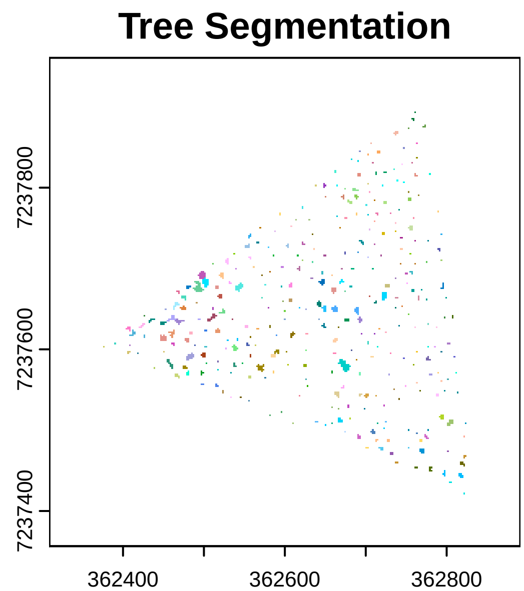
Results

```r
# plot indices
par(mfrow = c(2,2))
plot(ndvi, main = "ndvi")
plot(evi, main = "evi")
plot(gci, main = "gci")
plot(lai, main = "lai")
```




```r
# plot tree tops
par(mfrow = c(1,2))
plot(CHM, col = height.colors(50), main = "Tree Tops")
plot(sf::st_geometry(ttops), add = TRUE, pch = 3)
plot(crowns, col = pastel.colors(200), legend = FALSE, main = "Tree Segmentation")
```






```
# save all data in one df
params_df[nrow(params_df) + 1,] <- c(EIA_name,
                                     numtrees,
                                     treedens,
                                     treeheight_min,
                                     treeheight_max,
                                     treeheight_mean,
                                     canopyarea$area)
params_df_indices[nrow(params_df_indices) + 1,] <- c(ndvi_mean,
                                                     evi_mean,
                                                     gci_mean,
                                                     lai_mean)
kable(params_df)
```

name	numtrees	treedens	treeheight_min	treeheight_max	treeheight_mean	canopyarea
EIA2 C1	341	3100	1.5	11.96	3.59	4287.13

```
kable(params_df_indices)
```

ndvi_mean	evi_mean	gci_mean	lai_mean
0.98	2.49	0.04	8.9