Script Review: '14_lidR-Processing_to_Git'

SRM-LQ

04/02/2022

Contents

Action

This is an R Markdown document showing the LiDAR processing steps taken to derive landscape metrics and raster covariates from a continuous point cloud using the lidR package. The following markdown along with a virtual environment were also stored in the github repository titled "tcc-lidR-processing".

1 Import LiDAR

LiDAR downloads for the Ahbau region were imported as zip files and unpacked from their top-directory folder and subdirectory folder using the unzip function. I could not find any published R packages that deal with .7z archive files. Instead custom-written function was adopted from the RAMP project. This was done with the following code chunk:

```
zip_file_ahbau = ("./14_LiDR-Processing_GitRepo/Data/Ahbau.zip")
zip_dir_ahbau_top = ("./14_LiDR-Processing_GitRepo/Data/")
unzip(zip_file_ahbau,
      exdir = zip_dir_ahbau_top,
      overwrite = TRUE)
zip_dir_ahbau_sub = ("./14_LiDR-Processing_GitRepo/Data/Ahbau/Las_v12_ASPRS")
zip_file_ahbau_sub = list.files(
  zip_dir_ahbau_sub,
  full.names = T,
```

```r
  recursive = F,
  pattern = '.7z$')
# Write RAMP function and extract
un7zip = function(archive, where) {
  archive <- normalizePath(archive)
  current_path <- setwd(where)
  system(paste("7zr x", archive, sep = " "))
  setwd(current_path) }
#un7zip(zip_file_ahbau_sub, zip_dir_ahbau_top)
```

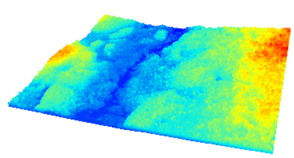
2 Read, Validate, and Assemble LiDAR Collection

LiDAR data for the Ahbau region included 311 .las tiles. These were assembled, imported and processed into the R environment as a LAS collection and processed using the LAScatalog processing engine. A LAX index file was added to the LAScatalog using system memory to speed up processing. For visualization purposes, a single las tile was extracted from file '093g030122ne" and presented directly below the following code. Subsequent illustrations were also rendered using this tile chunk.

```r
las_ctg_ahbau = readLAScatalog("./Data/Ahbau/Las_v12_ASPRS/", select = "xyzc")
lidR:::catalog_laxindex(las_ctg_ahbau)
las_check(las_ctg_ahbau)
crs(las_ctg_ahbau) = 3005
plot(las_ctg_ahbau)

las_tile_ahbau = readLAS("./Data/Ahbau/Las_v12_ASPRS/093g030122ne.las", select = 'xyzc')
crs(las_ctg_ahbau) = 3005
plot(las_tile_ahbau bg = "white")
```

```
#>   |                                                                                |==================
```



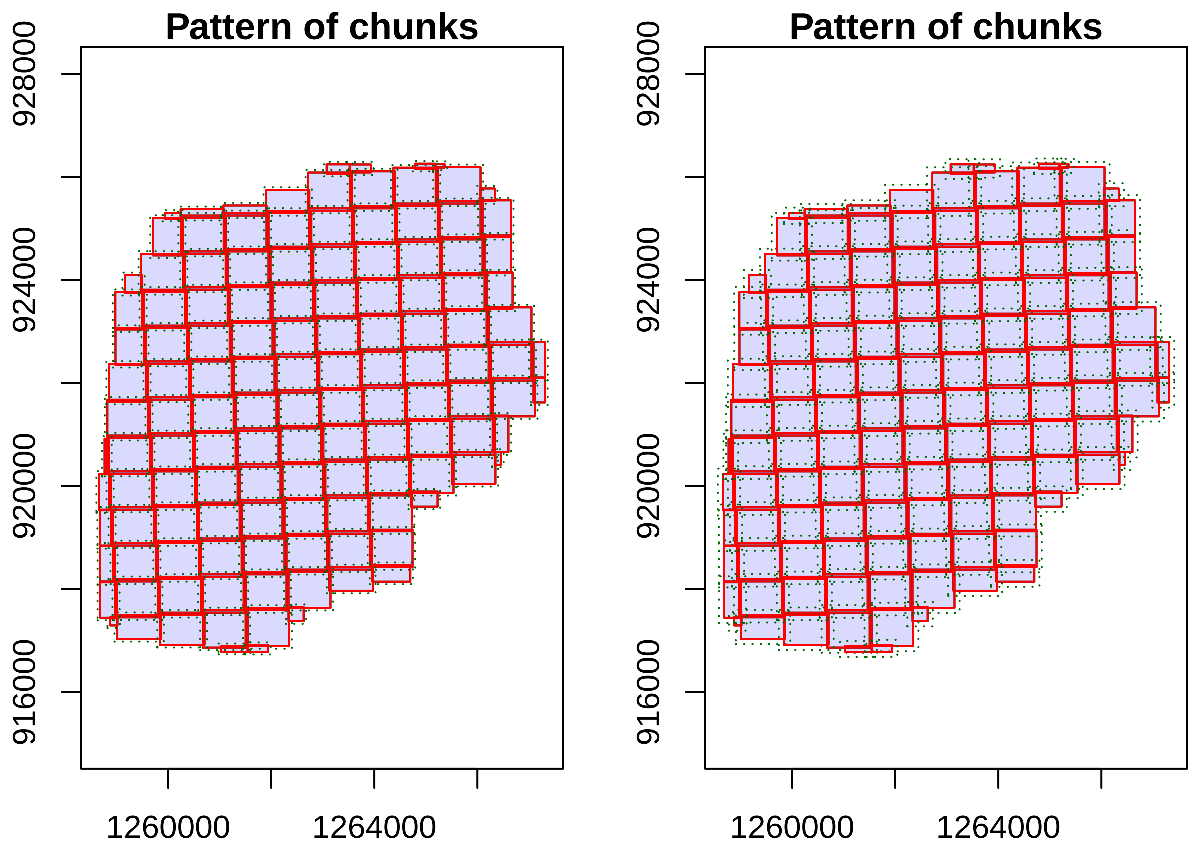
3 Buffer and Chunk Processing

Buffer distance for chunk processing was compared visually for alignment violations between 50m buffer (left) and 100m buffer (right). A 100m buffer was adopted for all subsequent processing operations.

```
par(mfrow=c(1,2))
opt_output_files(las_ctg_ahbau) <- paste0(tempdir(), "{*}_classified")

opt_chunk_buffer(las_ctg_ahbau) = 50
plot(las_ctg_ahbau, chunk = TRUE)

opt_chunk_buffer(las_ctg_ahbau) = 100
plot(las_ctg_ahbau, chunk = TRUE)
```



4 Classification of Ground Points

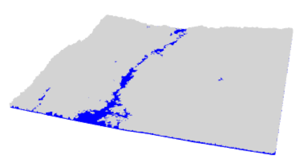
Classification of ground points was applied using the 'csf()' Cloth Simulation Filter algorithm (Zhang et al. 2016). The lidR package also calls on the RCSF package for backend operations here, though in case new R installations are being used by clients, this packaged was also installed in this output's virtual environment now available for cloning on the github repo (see link in Action section). To account for variable topography in the study site and to reduce errors during any potential post-processing, the sloop_smooth argument was adopted over a cloth resolution of 1m at a rigidness factor of 1.

```
mfrow3d(1,2)
#las_ctg_ahbau_classified = classify_ground(las_ctg_ahbau,
#                                              csf(sloop_smooth=TRUE,
#                                                  cloth_resolution = 1,
#                                                  rigidness = 1))

las_tile_ahbau_classified_csf = classify_ground(las_tile_ahbau,
                                                 csf(sloop_smooth = TRUE,
                                                 cloth_resolution = 1,
                                                 rigidness = 1))

util_makeZhangParam()
#> $ws
#> [1]  5  9 13 17
#>
#> $th
#> [1] 3 3 3 3
las_tile_ahbau_classified_pmf = classify_ground(las_tile_ahbau,
                                                 pmf(ws = seq(5, 9, 13),
                                                     th = seq(3, 3, 3)))
#> Morphological filter: 1% (1 threads)Morphological filter: 2% (1 threads)Morphological filter: 3% (1

plot(las_tile_ahbau_classified_csf, color = "Classification", size = 3, bg = "white")
plot(las_tile_ahbau_classified_pmf, color = "Classification", size = 3, bg = "white")
rglwidget(scene3d(minimal = TRUE))
```



5 Classification of Noise

Classification of photon noise was applied using the 'sor' Statistical Outlier Removal Algoirhtm fitted with 15-point neighbourhood and a multiplier of 4. Point neighbourhood was selected as a higher range numeric from the mean density of points (m^2).

```r
mfrow3d(1,2)
rglwidget(scene3d(minimal = TRUE))
```

Zhang, Wuming, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xie, Xiaoyan Wang, and Guangjian Yan. 2016. "An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation." *Remote Sensing* 8 (6): 501.