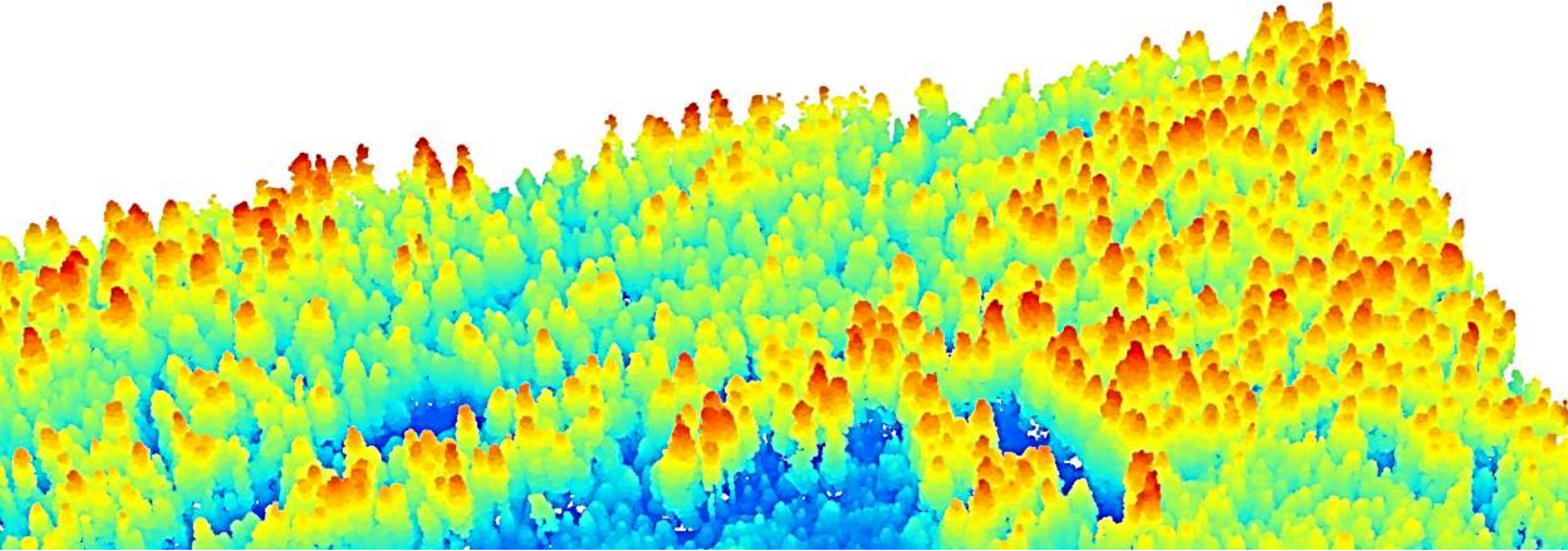
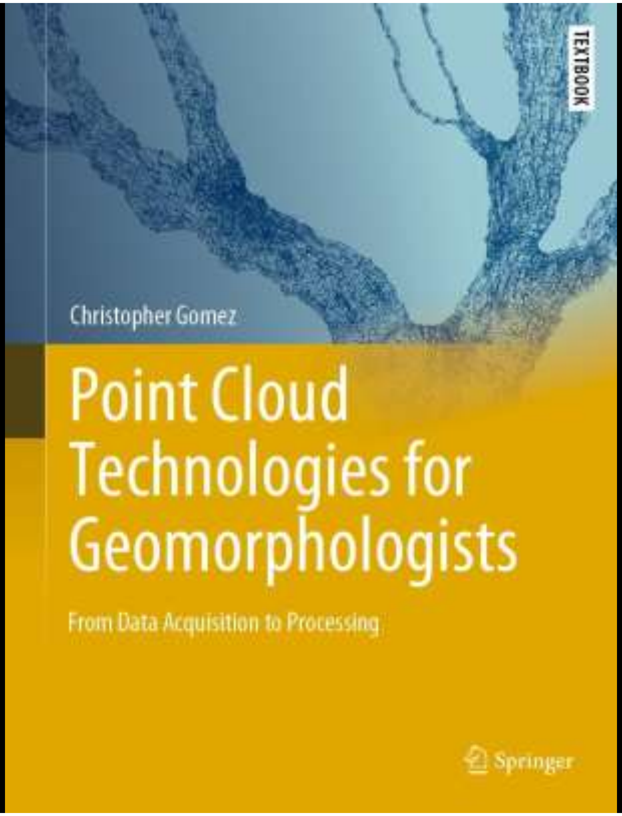
DAY 2

LiDAR data processing (November 2023)

Professor Christopher Gomez
christophergomez@bear.kobe-u.ac.jp
SABO Laboratory @ Kobe University (Japan)
PSBA Research Centre & Geography @ UGM University (Indonesia)





砂防2
点群データ:入門
LiDAR data processing: from points to the world

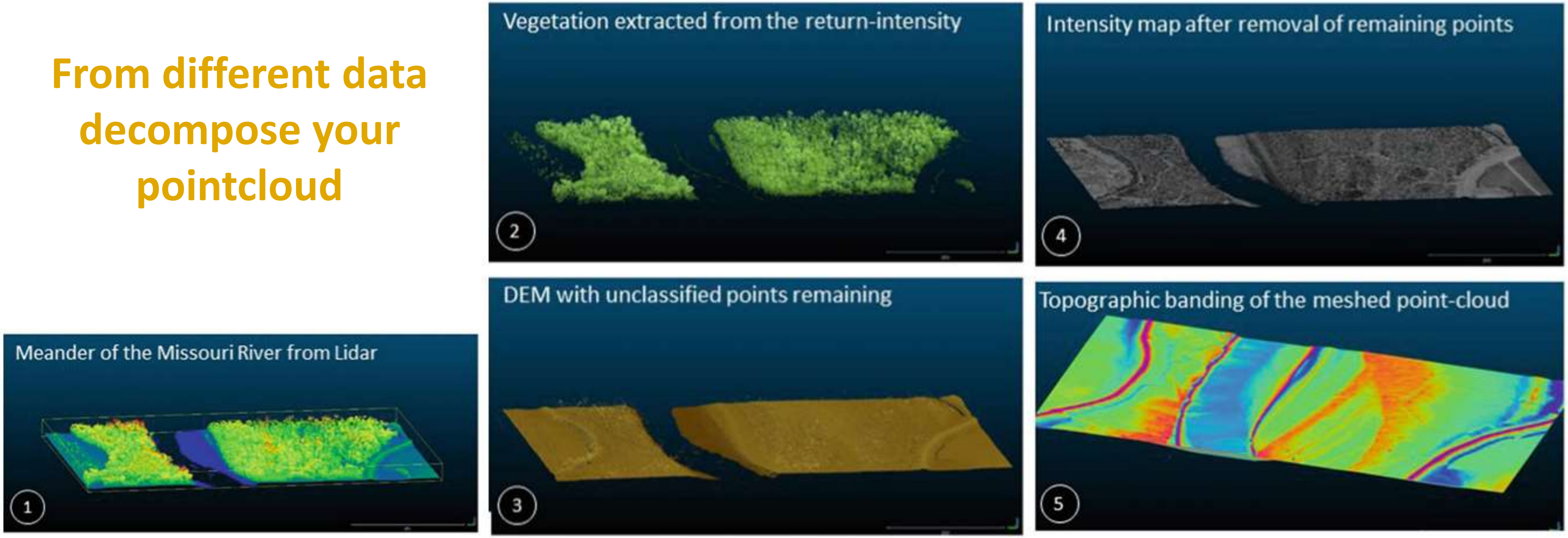
Christopher Gomez

christophergomez@bear.kobe-u.ac.jp

1. Extracting elements with CloudCompare

christophergomez@bear.kobe-u.ac.jp

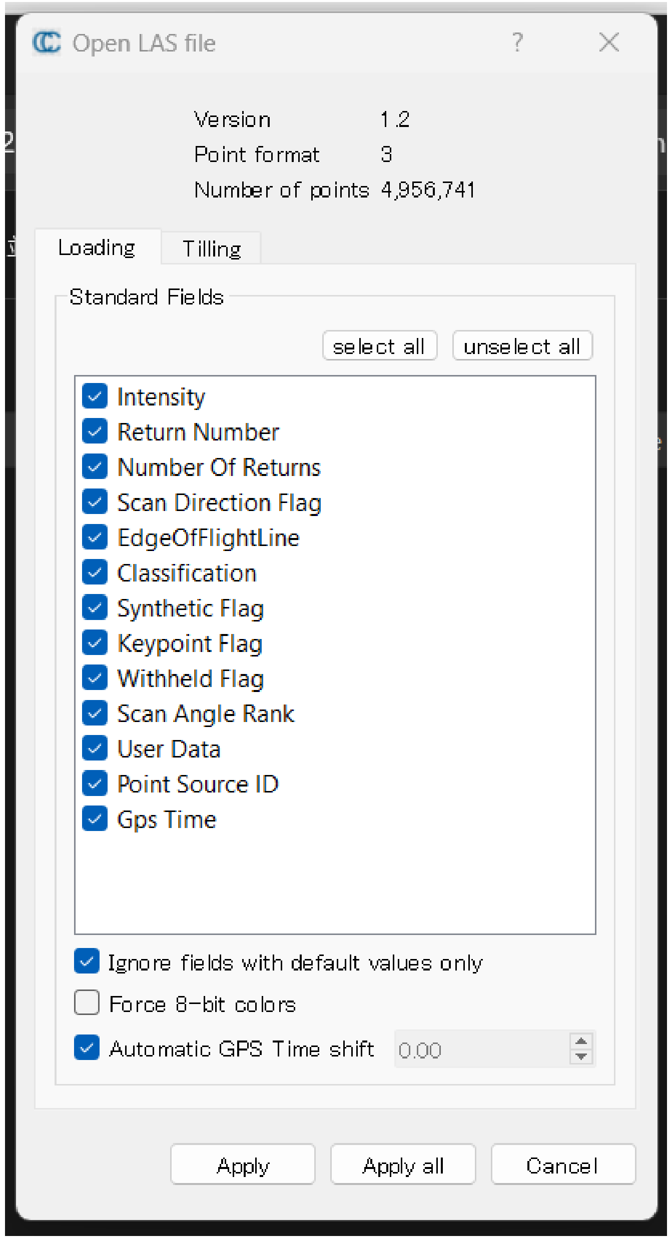
From different data decompose your pointcloud

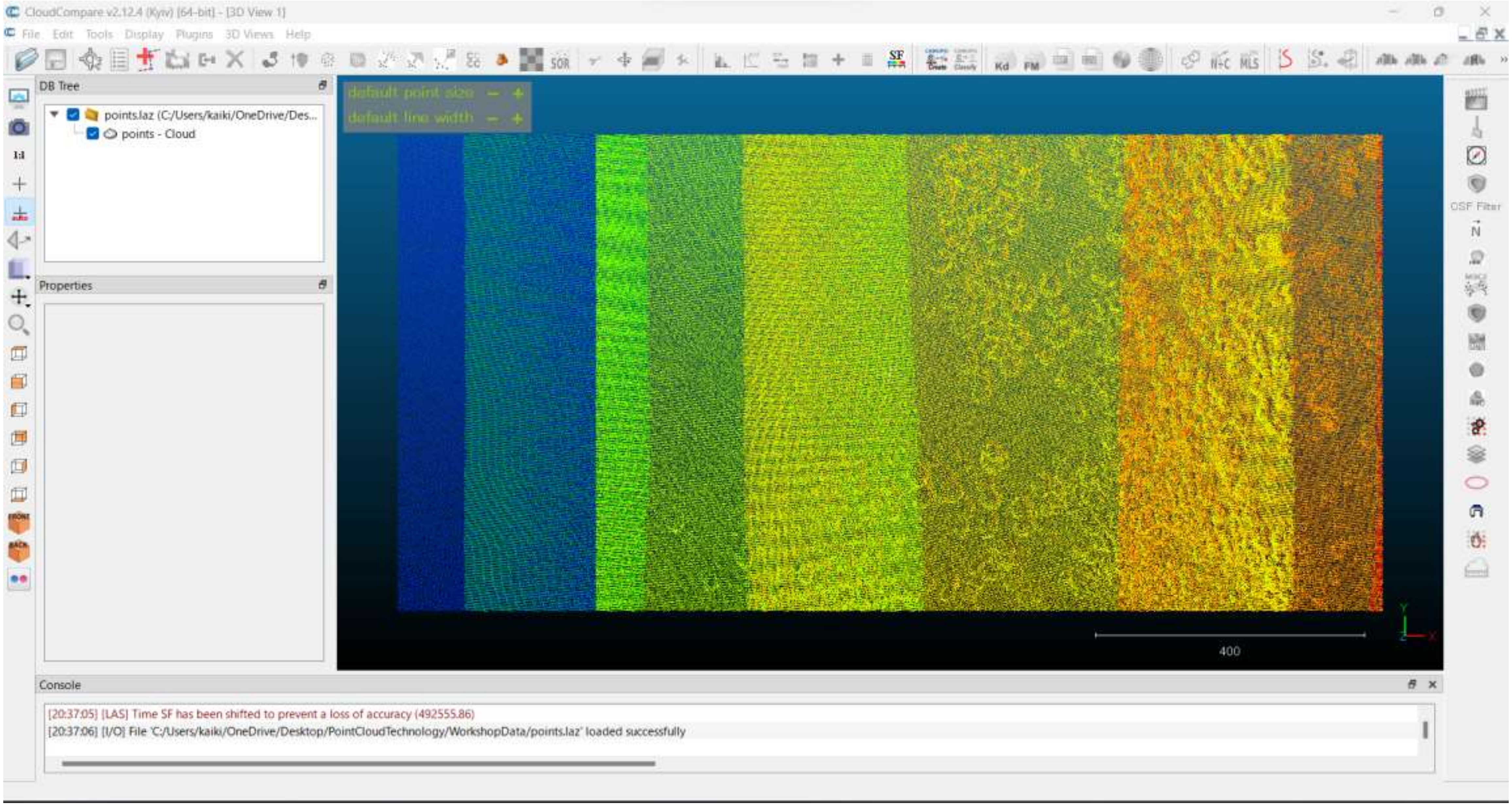


1. Extracting elements

… to do so you use the characteristics of the pointcloud.

$$P_n = \sum_{i=1}^{n} [X_i, Y_i, Z_i, Att_i]$$

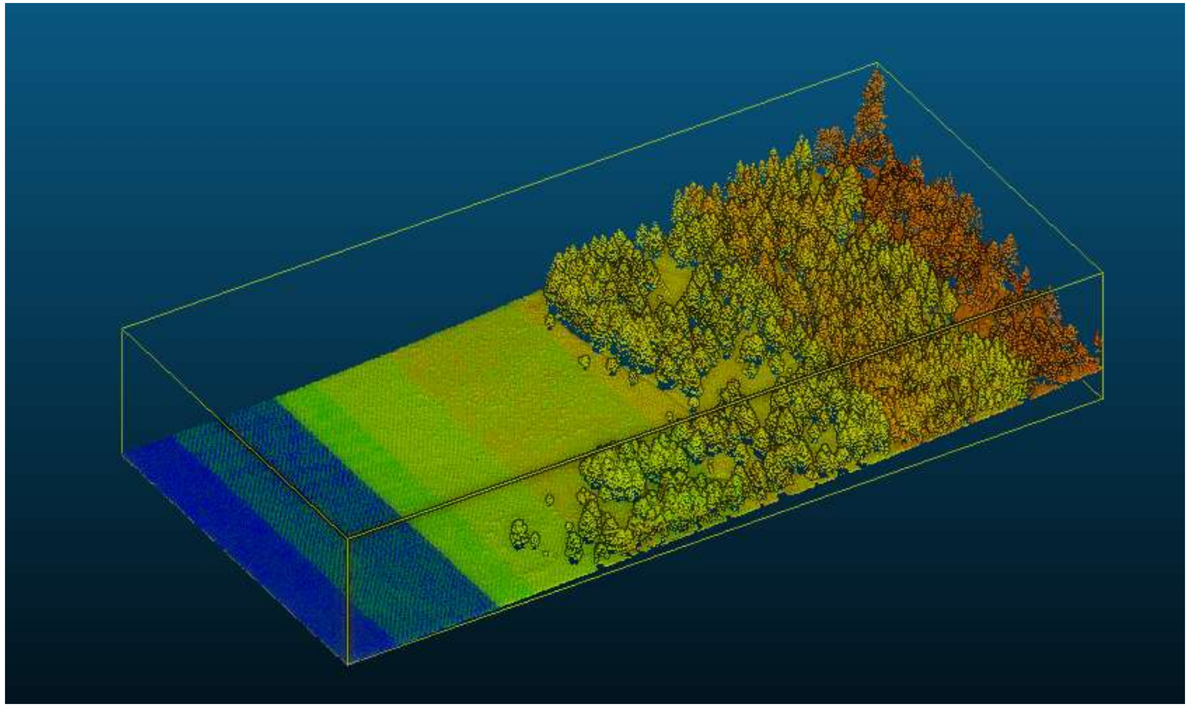


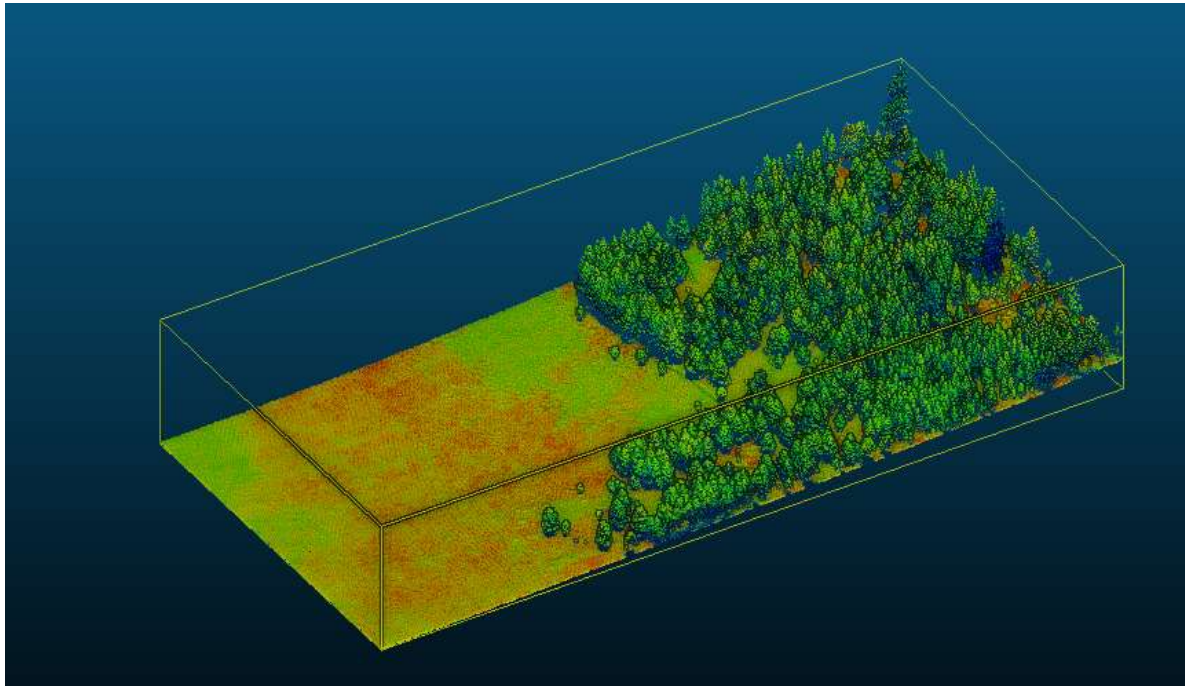


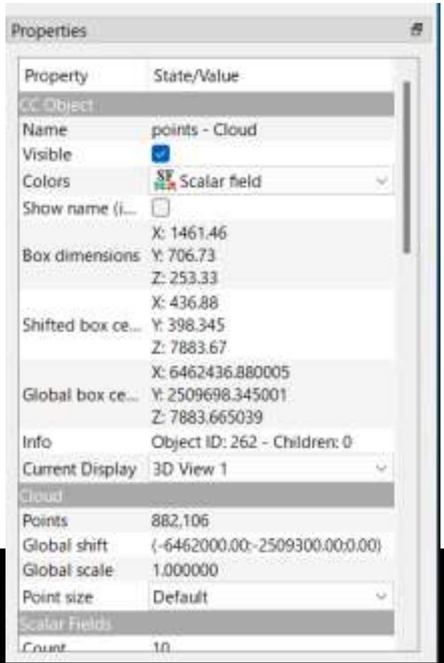
1. Extracting elements

1. Extracting elements

1. EDL shader (real time rendering)

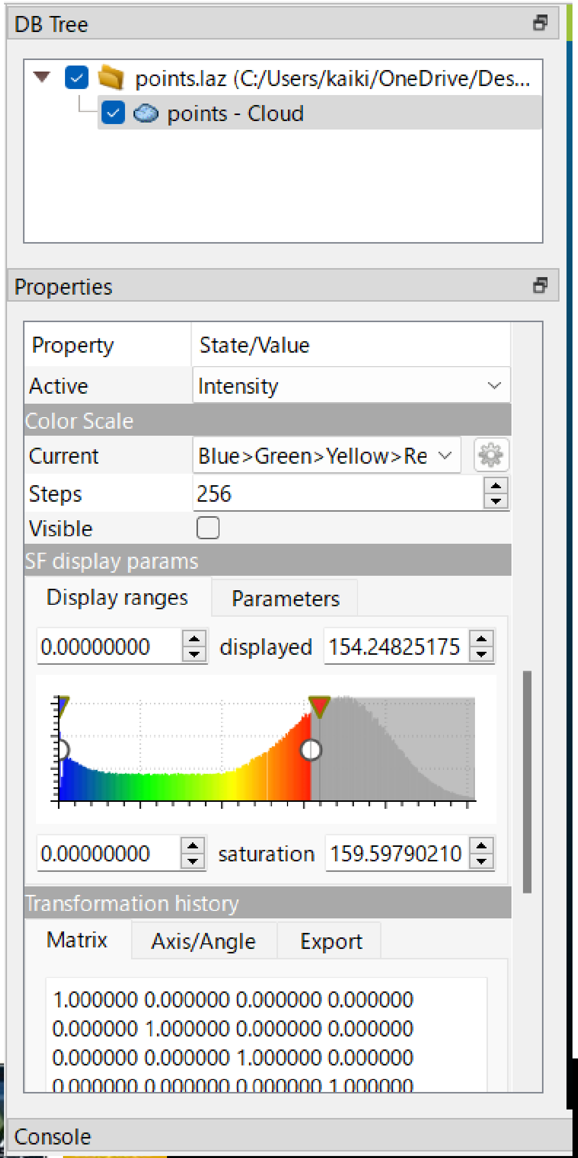


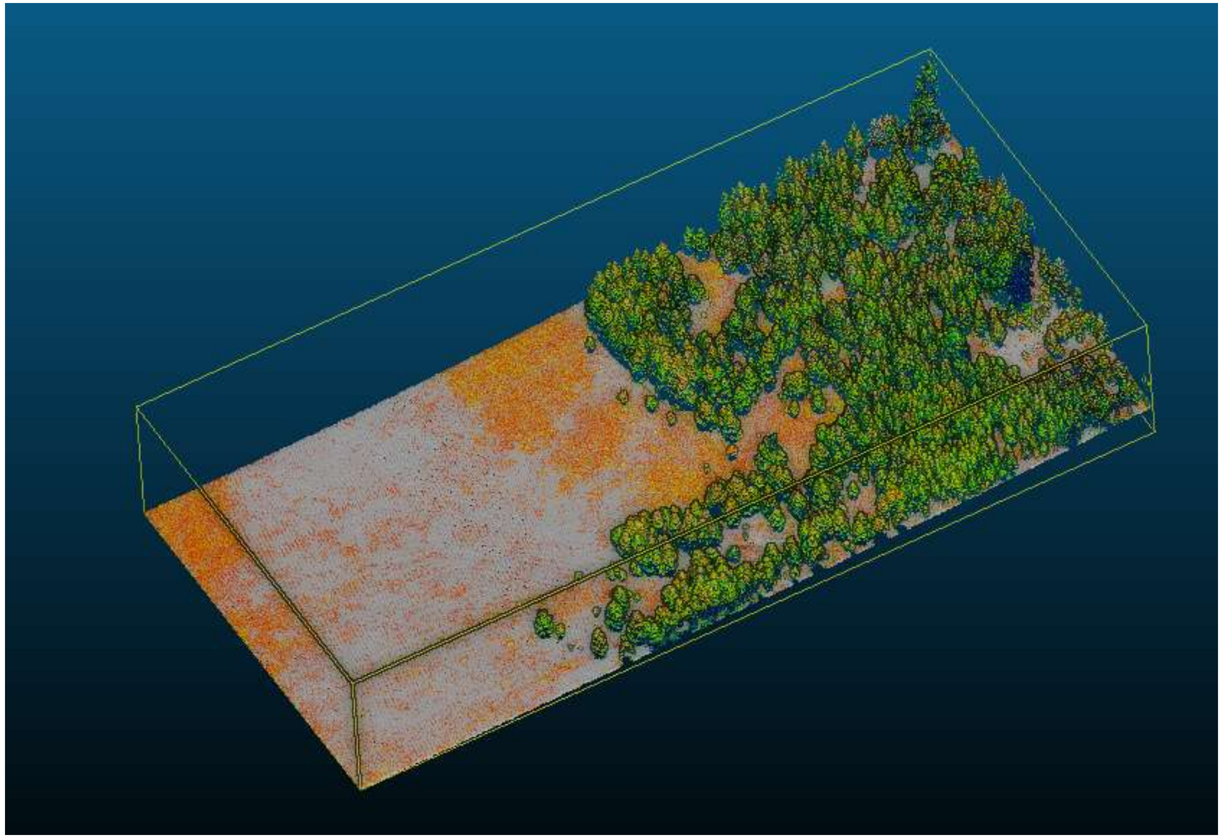


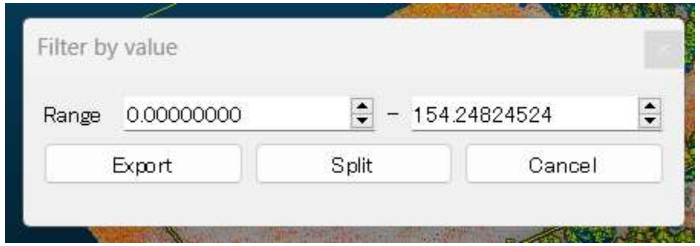


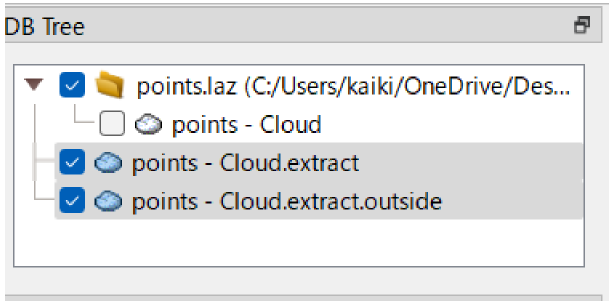
2. Display the intensity return of the Lidar

Intensity-based classification



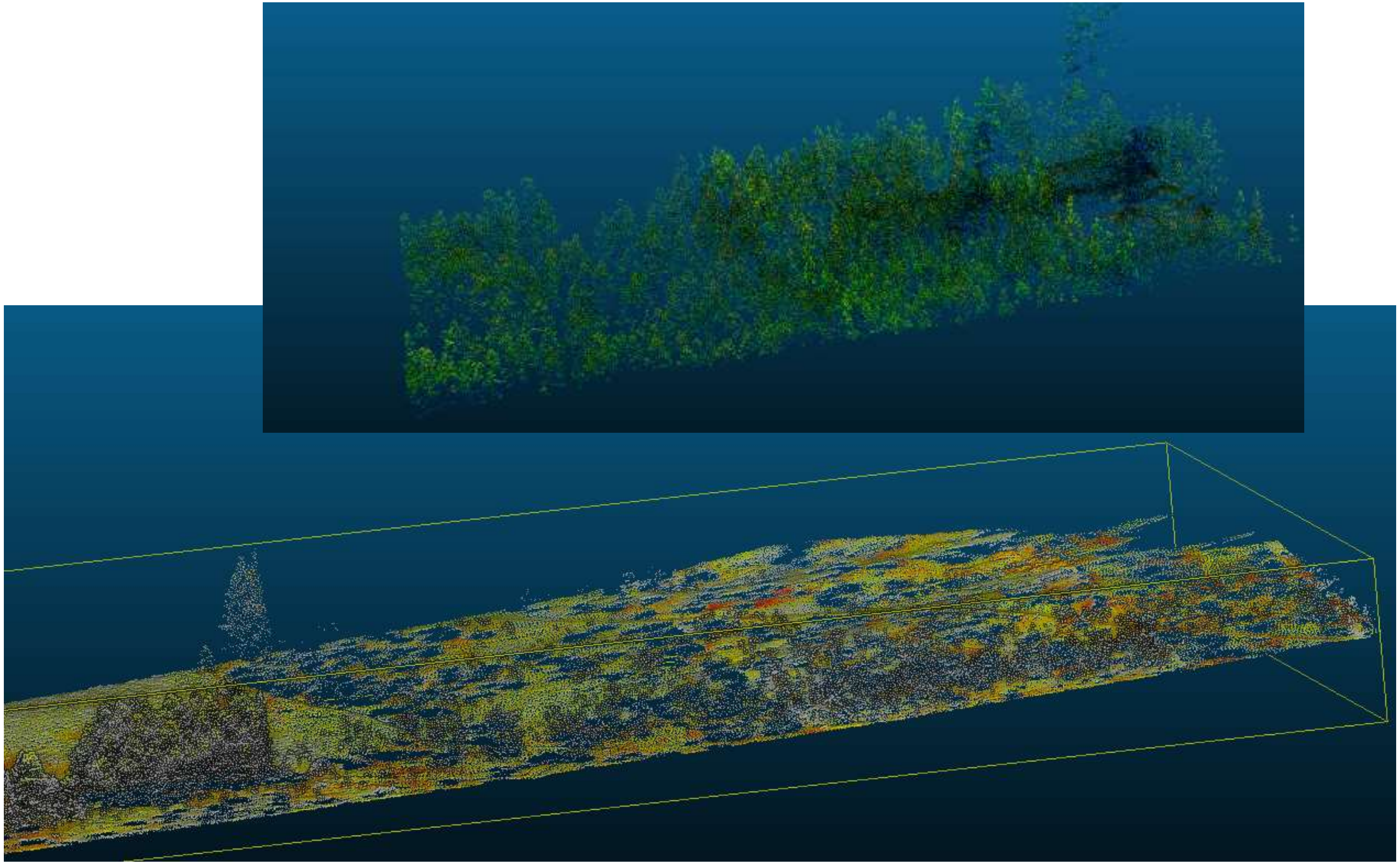






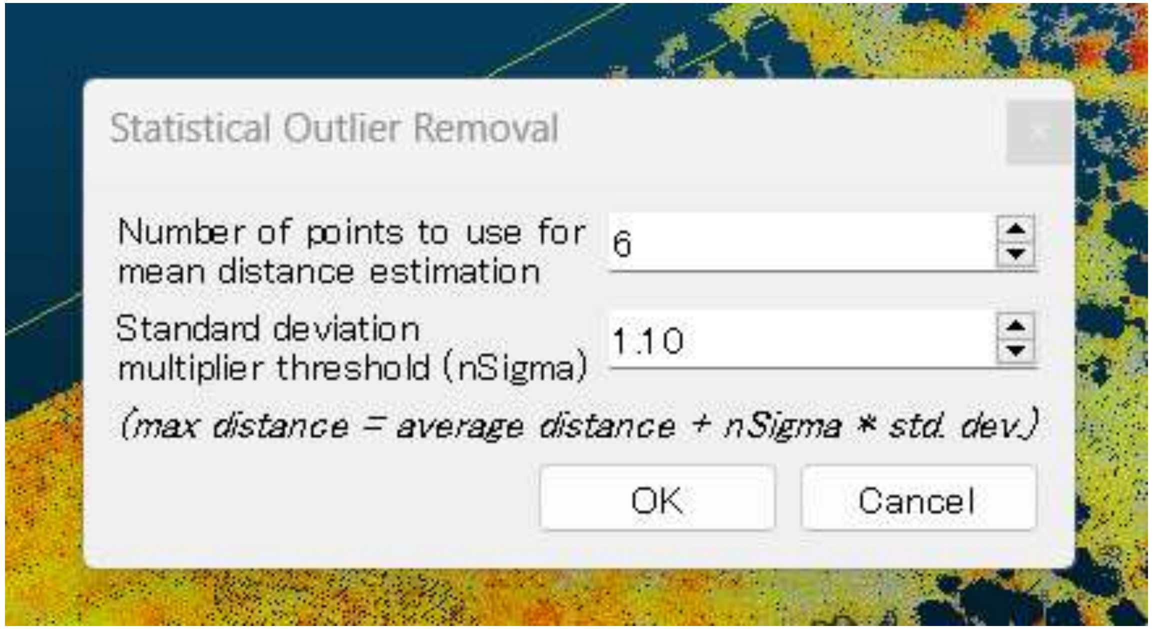
1. Extracting elements

Or cut “manually”



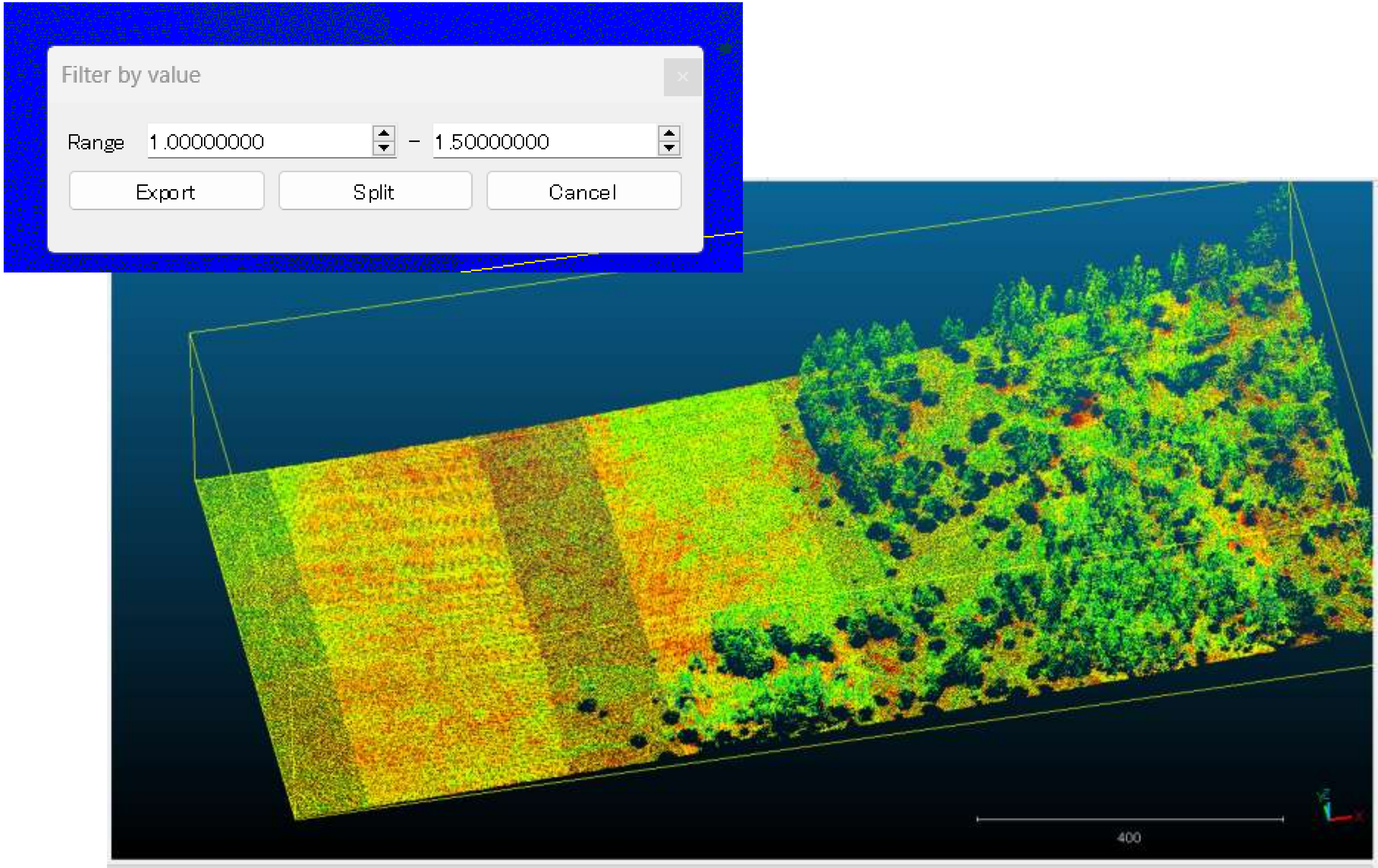
1. Extracting elements

… when you are happy with your classification,
then you can eliminate the outliers:

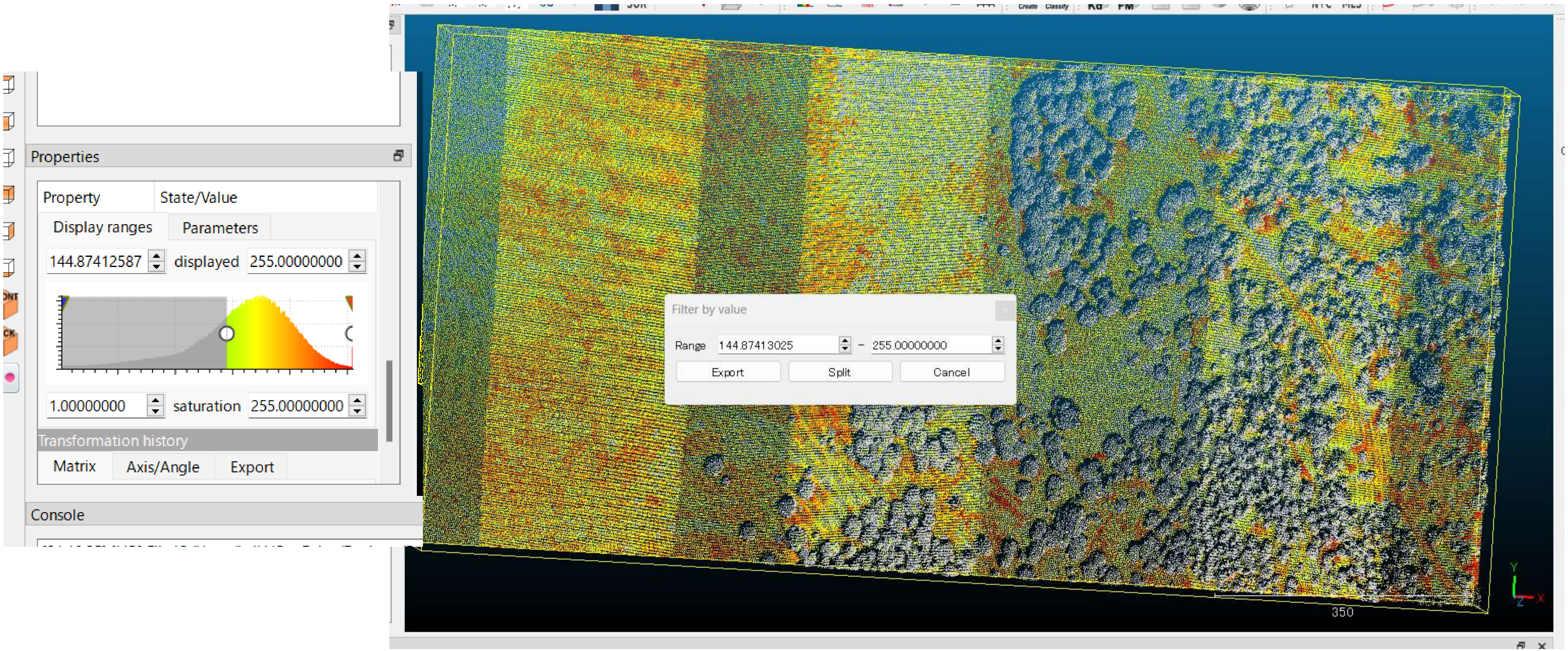


Other option:

Classification Based on the Number of returns

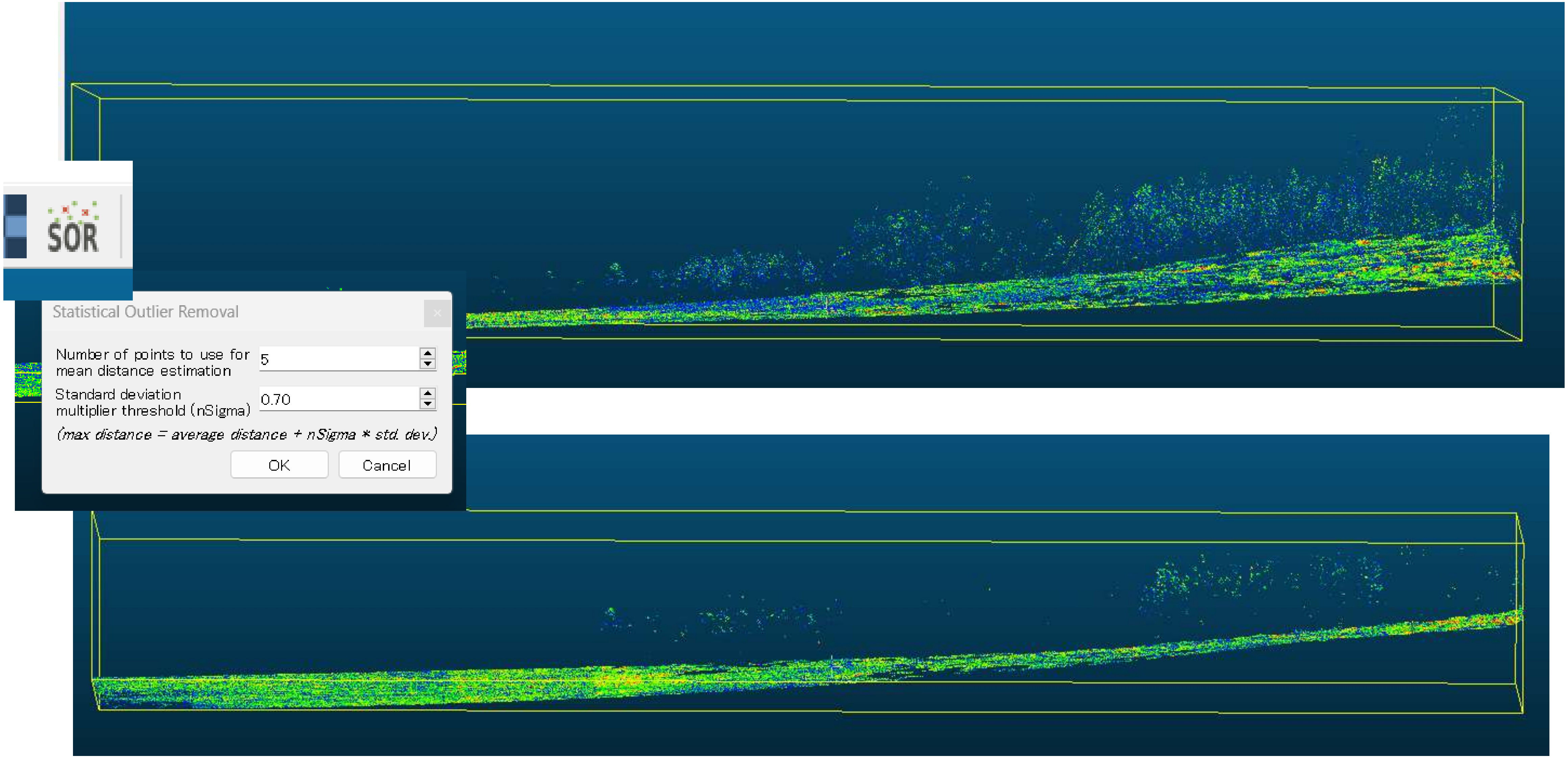


Eliminate the first return on top of the trees:



1. Extracting elements

But it is not perfect yet, we still have a lot of remaining noise.

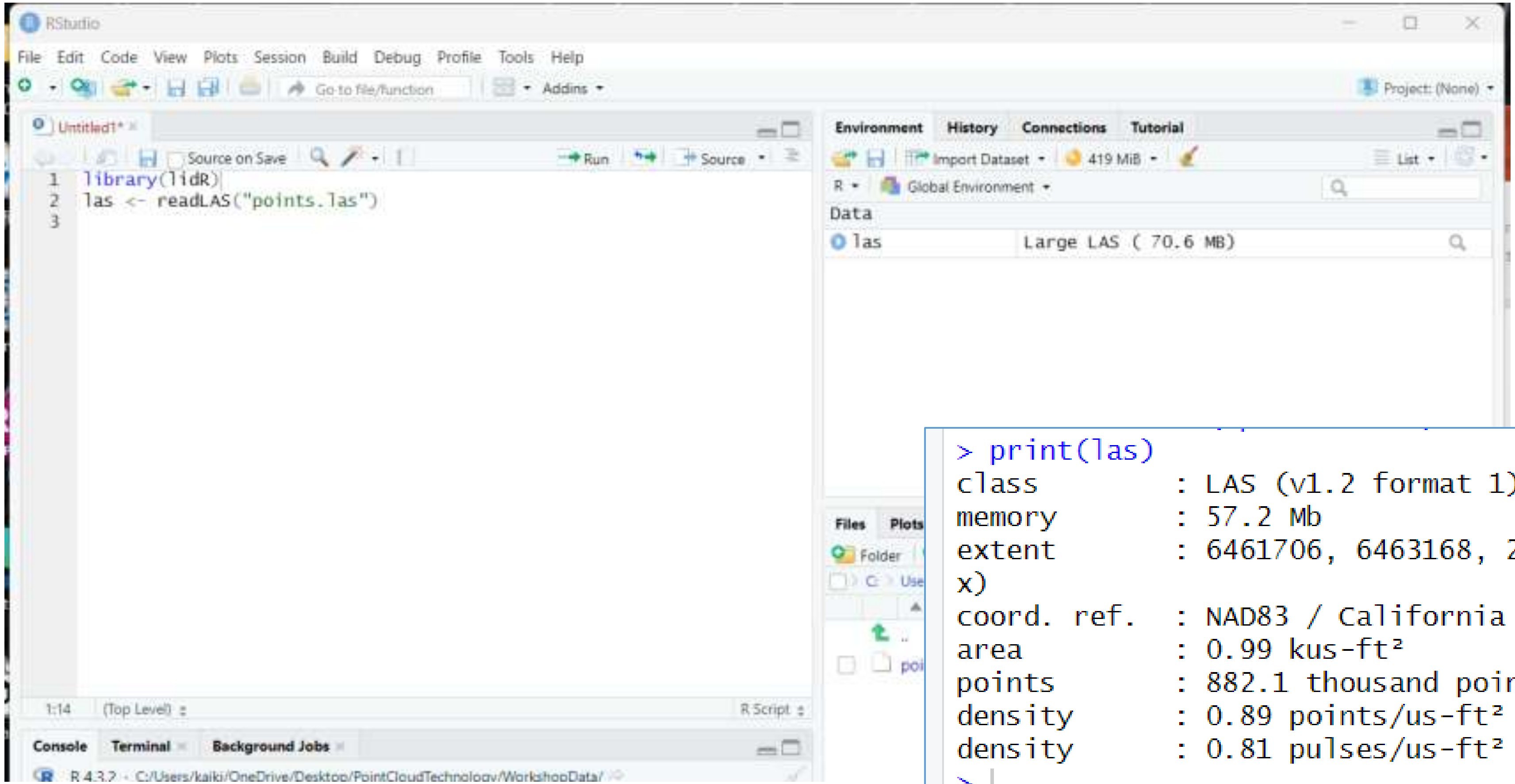


+ some Manual cleaning

2. Extracting elements with LidR

christophergomez@bear.kobe-u.ac.jp

In R-Studio:



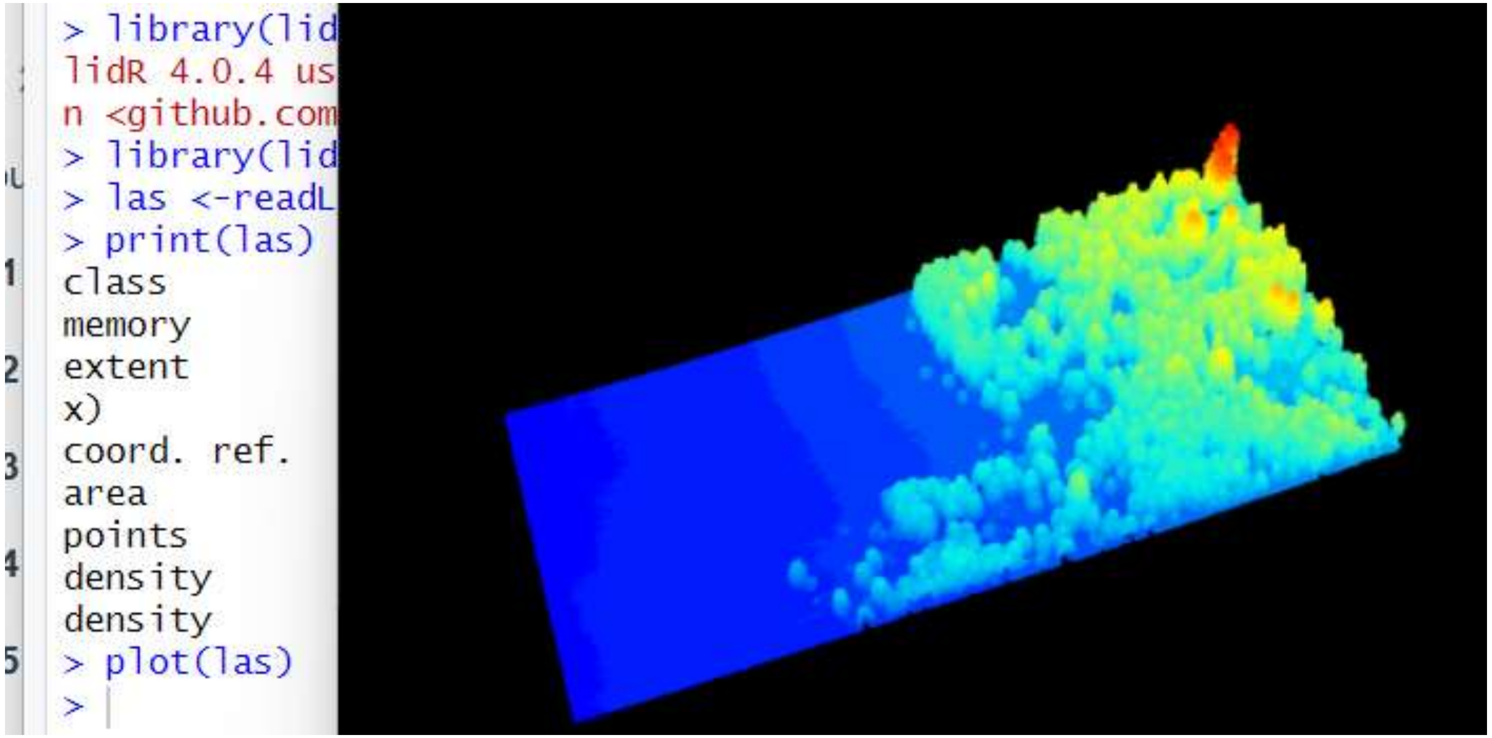

```
> print(las)
class        : LAS (v1.2 format 1)
memory       : 57.2 Mb
extent       : 6461706, 6463168, 2509345, 2510052 (xmin, xmax, ymin, yma
x)
coord. ref.  : NAD83 / California zone 5 (ftUS)
area         : 0.99 kus-ft²
points       : 882.1 thousand points
density      : 0.89 points/us-ft²
density      : 0.81 pulses/us-ft²
>
```

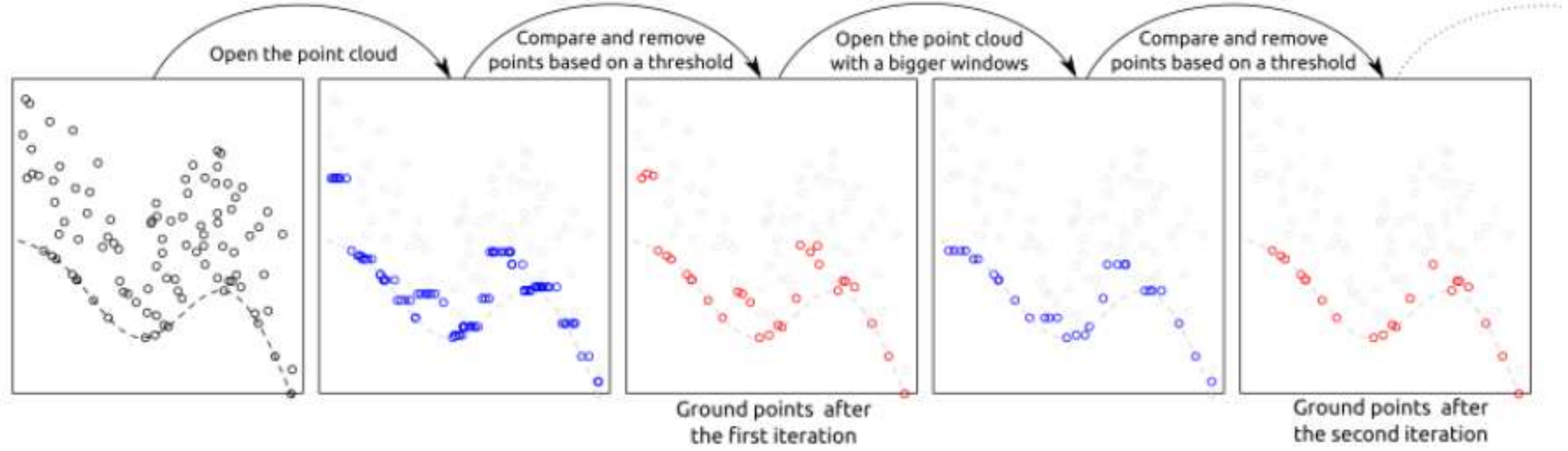
```
library(lidR)
las <- readLAS("points.las")
```

… you can do by coding, what we just did with the GUI:

las <- readLAS("file.las", filter = "-keep_first") # Read only first returns

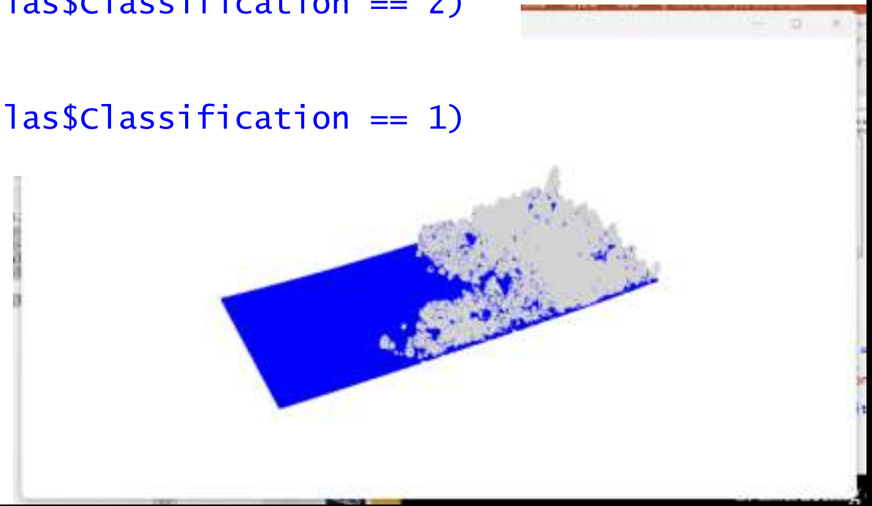
> plot(las)

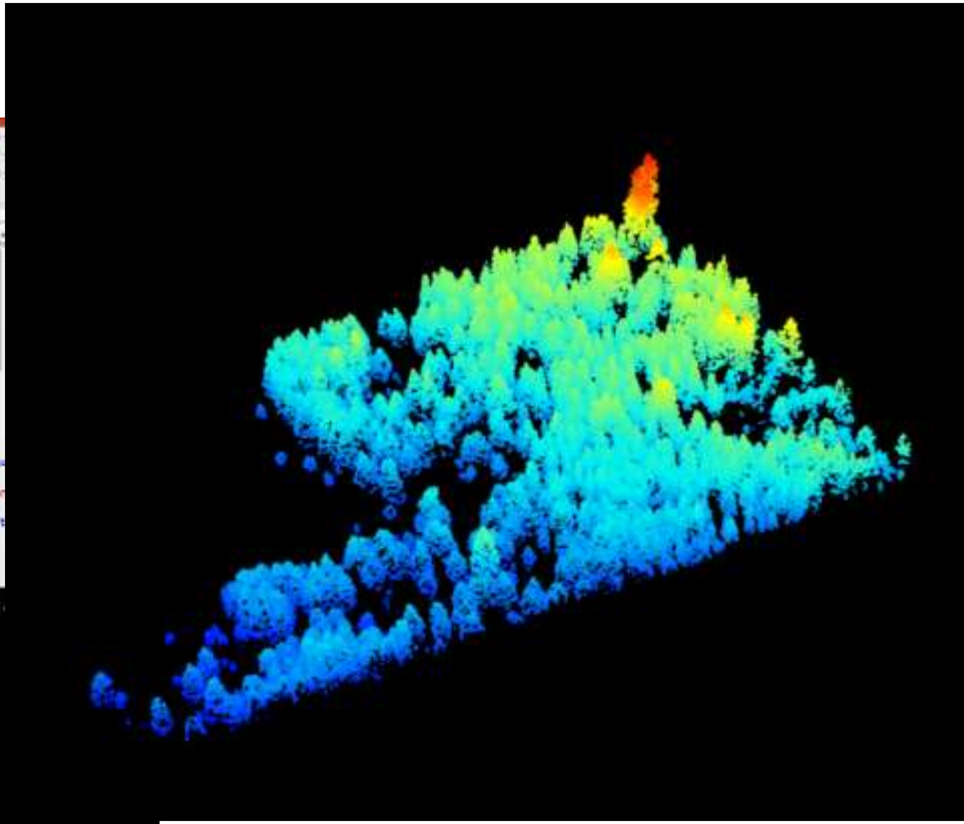


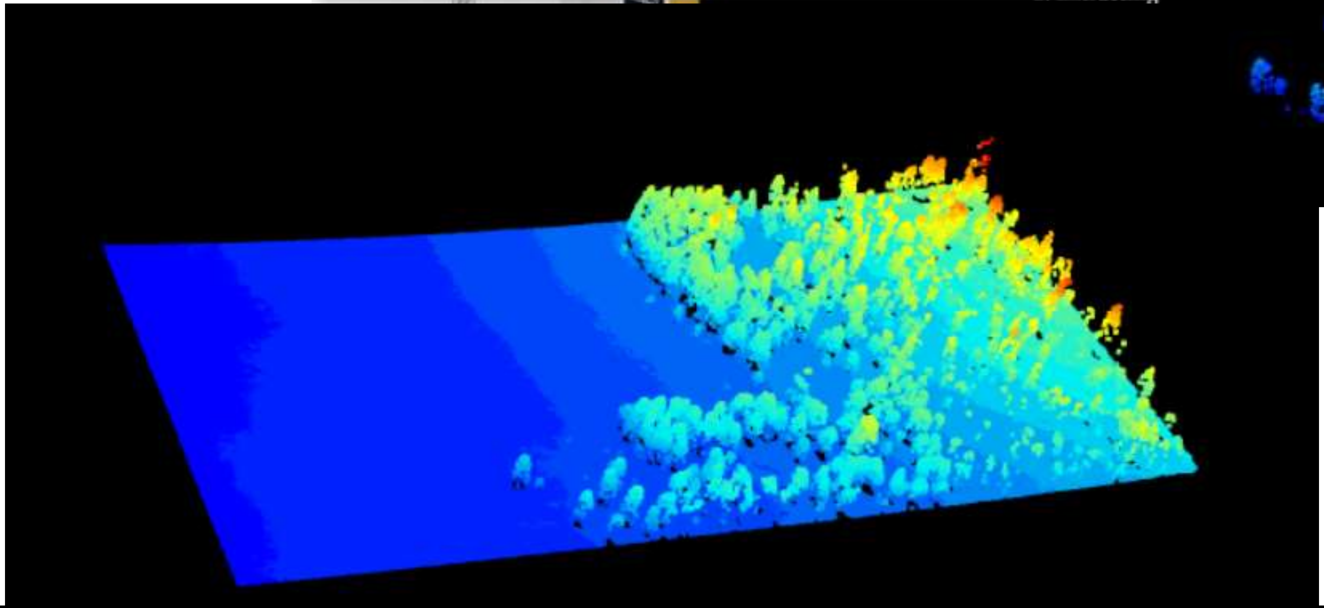



```
las <- readLAS("points.las")
las <- classify_ground(las, algorithm = pmf(ws = 5, th = 3))
Original dataset already contains 277737 ground points. These
points were reclassified as 'unclassified' before performing a
new ground classification.
plot(las, color = "Classification", size = 3, bg = "white")
```

```
➢ filtered_points <- subset(las, las$Classification == 2)
➢ plot(filtered_points)
➢ plot(las)
➢ filtered_points <- subset(las, las$Classification == 1)
➢ plot(filtered_points)
```







2. Extracting elements with LidR

Open Access | Article

An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation

by Wuming Zhang [1], Jianbo Qi [1,*], Peng Wan [1], Hongtao Wang [2], Donghui Xie [1], Xiaoyan Wang [1] and Guangjian Yan [1]

1 State Key Laboratory of Remote Sensing Science, Beijing Key Laboratory of Environmental Remote Sensing and Digital City, School of Geography, Beijing Normal University, Beijing 100875, China

2 School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454003, China

* Author to whom correspondence should be addressed.

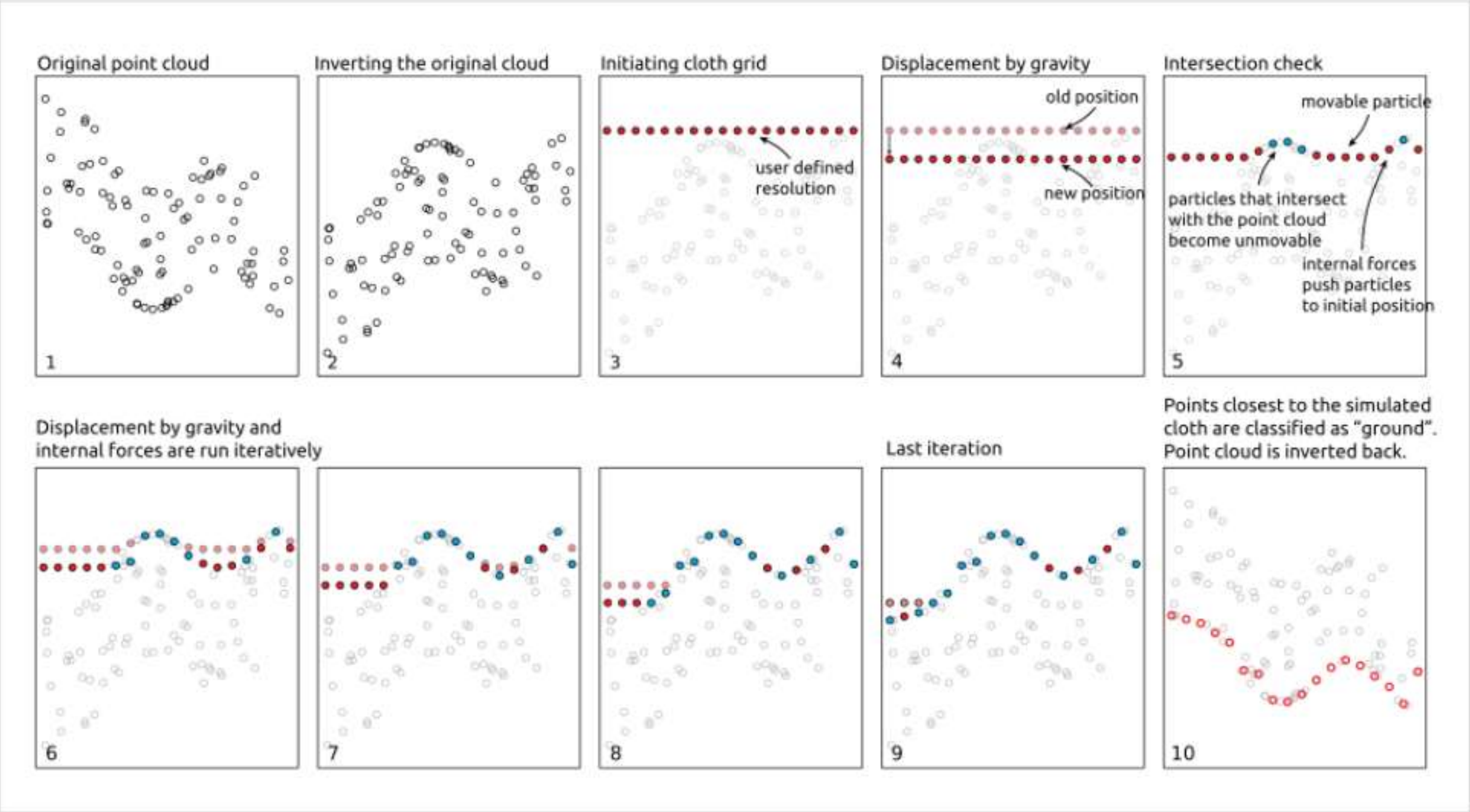
Remote Sens. **2016**, *8*(6), 501; https://doi.org/10.3390/rs8060501

Received: 13 March 2016 / Revised: 19 May 2016 / Accepted: 3 June 2016 / Published: 15 June 2016

(This article belongs to the Special Issue **Airborne Laser Scanning**)

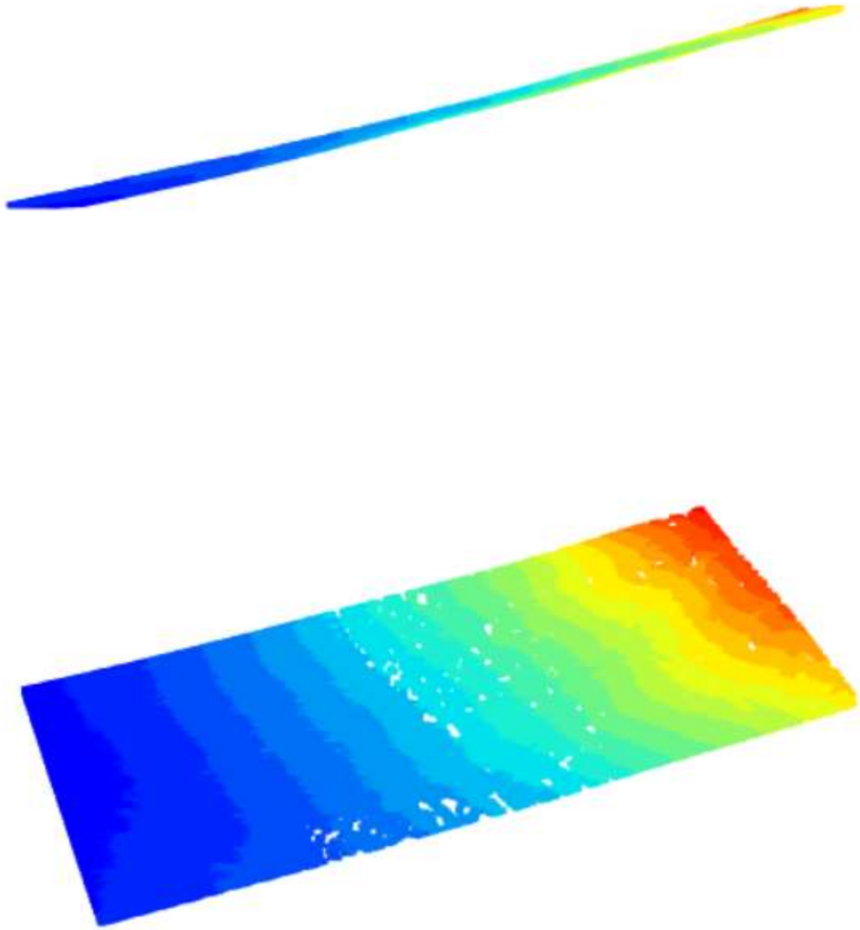
Download | Browse Figures | Versions Notes

2. Extracting elements with LidR

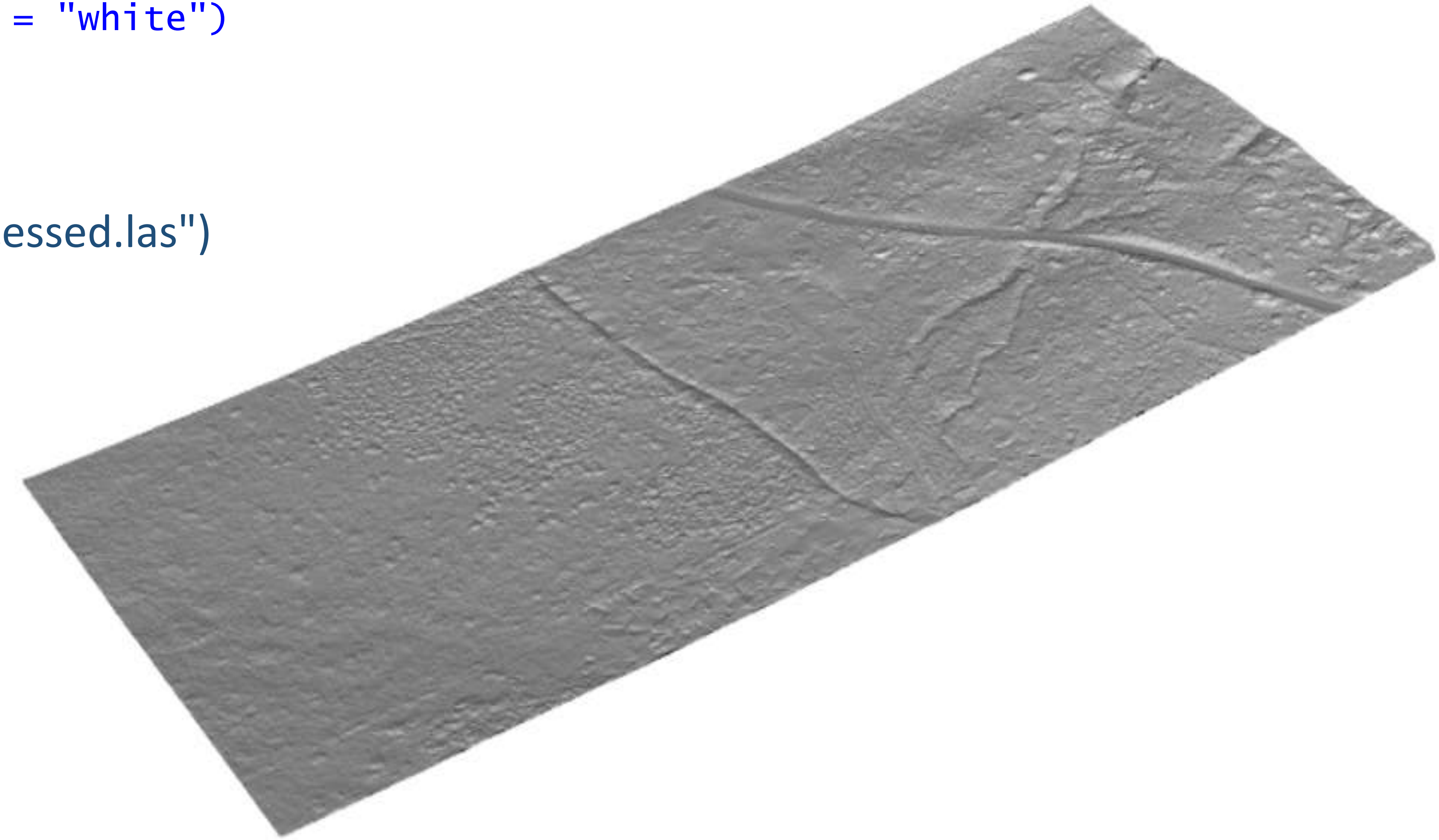



```
las2 <- classify_ground(las2, algorithm = csf())
```

```r
gnd <- filter_ground(las2)
plot(gnd, size = 3, bg = "white")
```



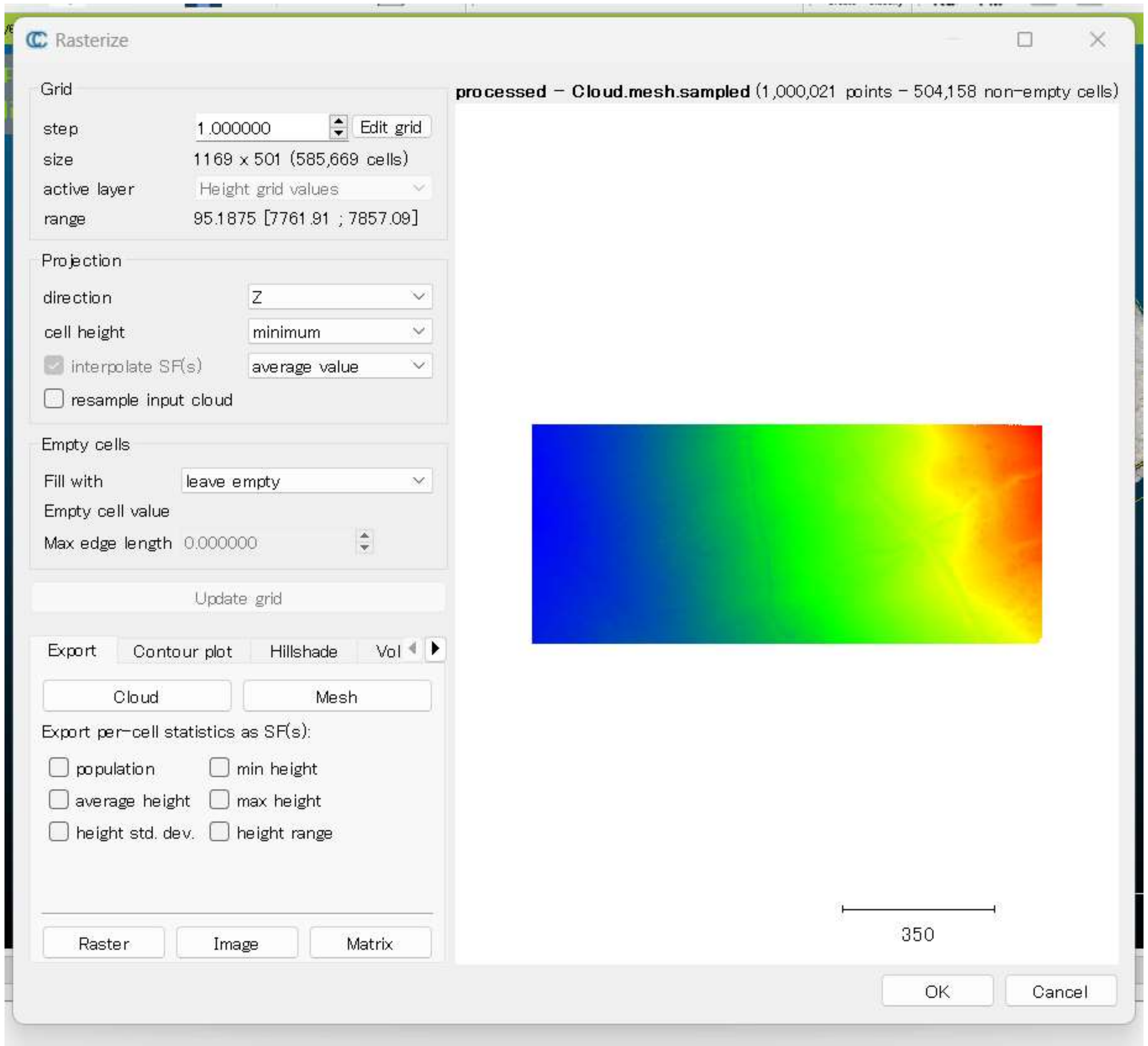
```r
dem <- rasterize_terrain(gnd, res = 1, algorithm = tin())
plot_dtm3d(dem, bg = "white")
```



```r
writeLAS(gnd, "processed.las")
```

1. Mesh Delauney 2.5
2. Point-sampling on Mesh
3. From Pt to DEM

… then you can also add population for your DEM to assess the quality.

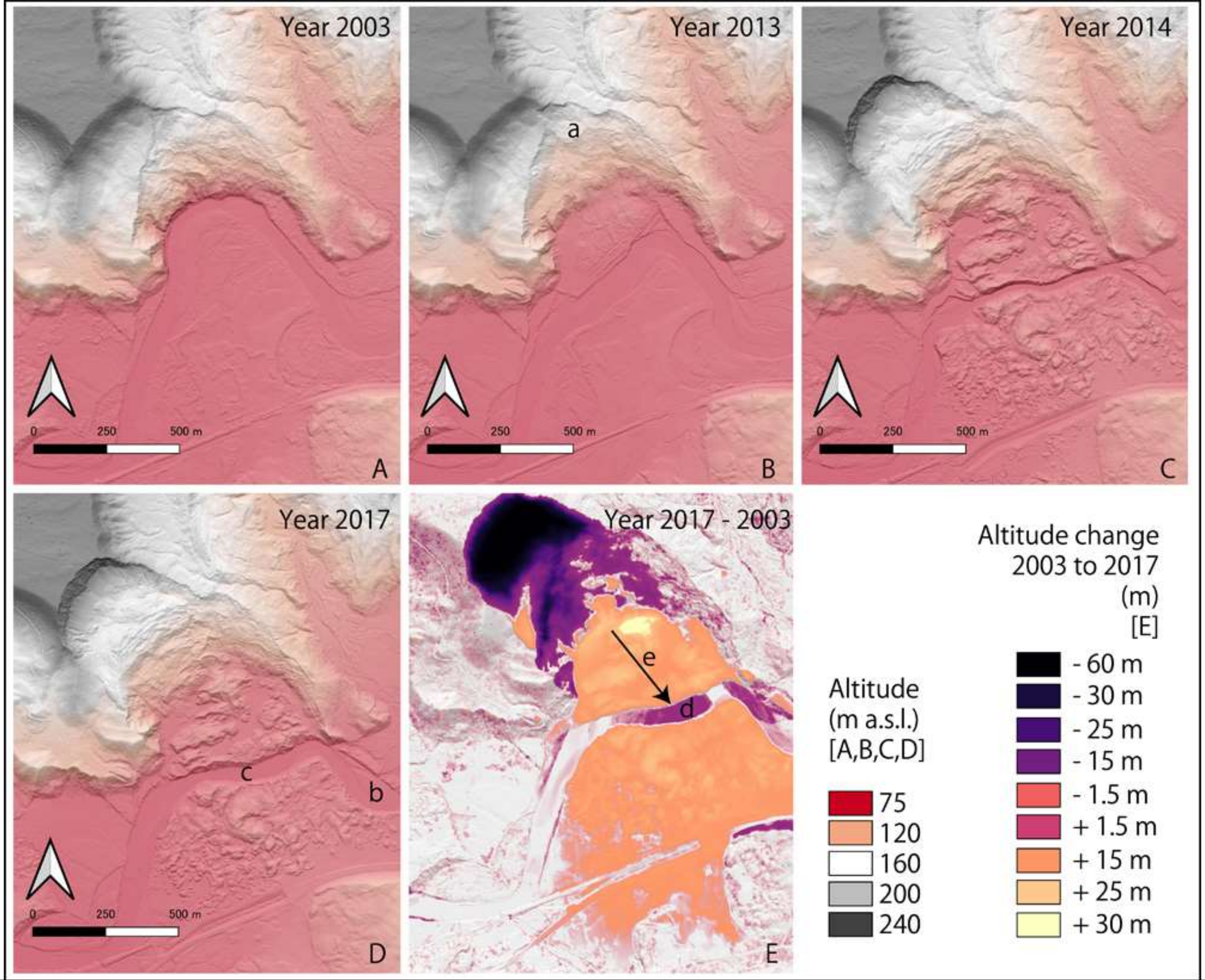


2. Extracting elements with LidR

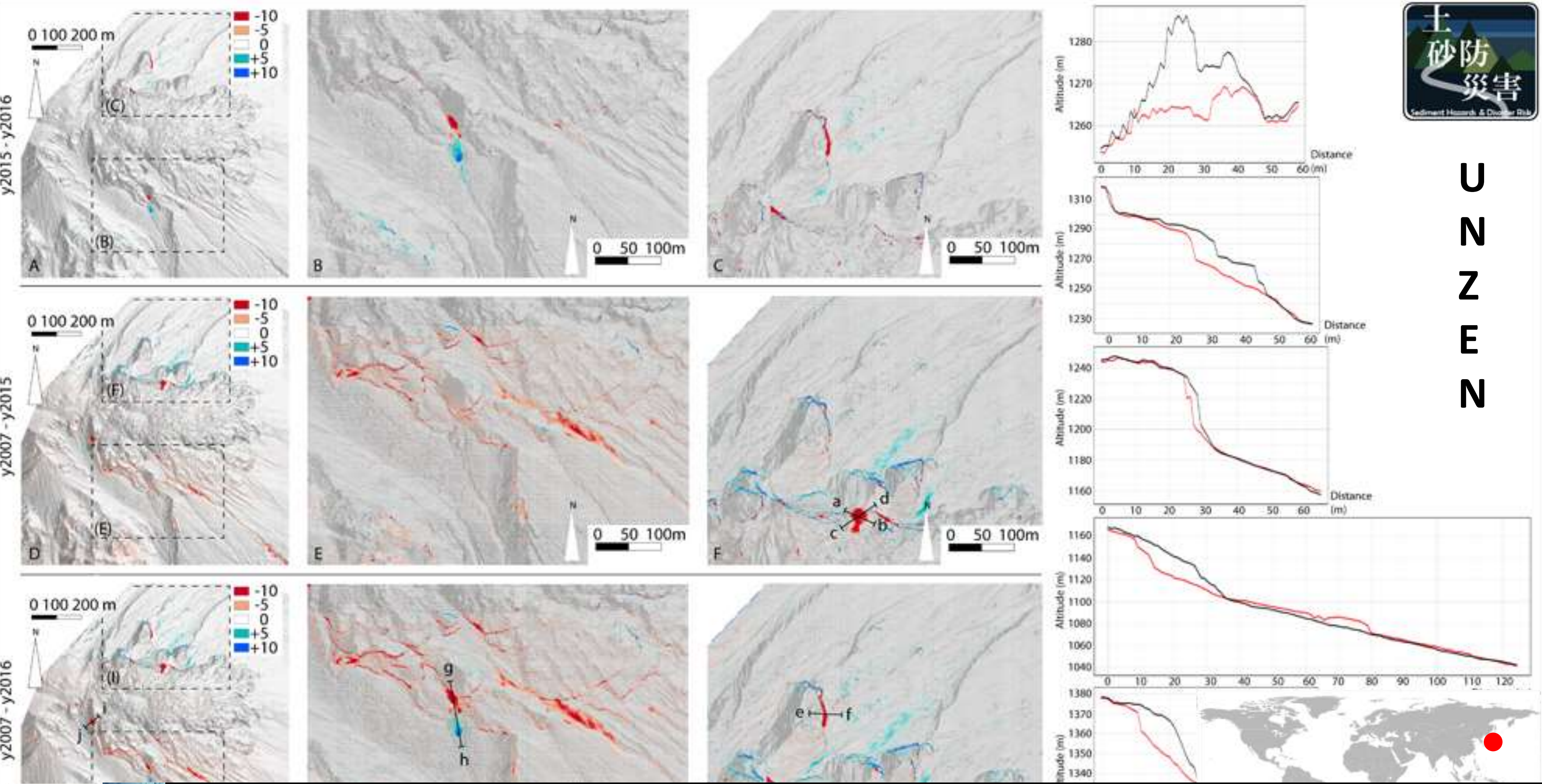
3. Comparing pointclouds

christophergomez@bear.kobe-u.ac.jp

Temporal evolution from HRT (High-resolution Topography) PCL and DEMs differentiation at Oso Landslide in USA (LiDAR from the USGS and the OpenTopography community).

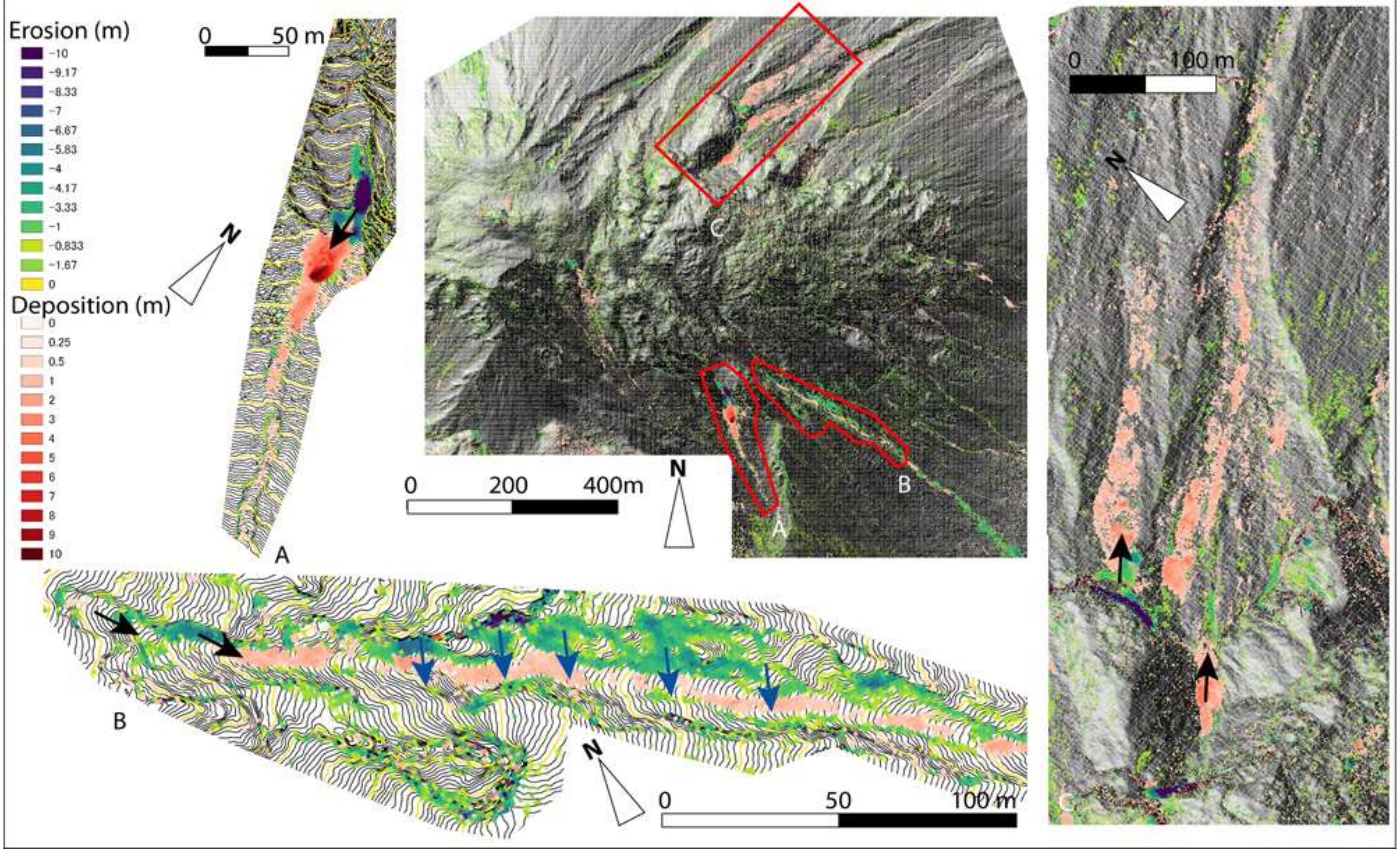


3. Comparing pointclouds



UNZEN

2. Comparing pointclouds



2. Comparing pointclouds

The cone filtering method

$$\forall P_j \in A: hp_i - \Delta h\big(d(p_i, p_j), m\big) \leq hp_j$$

The adaptive-cone filtering method

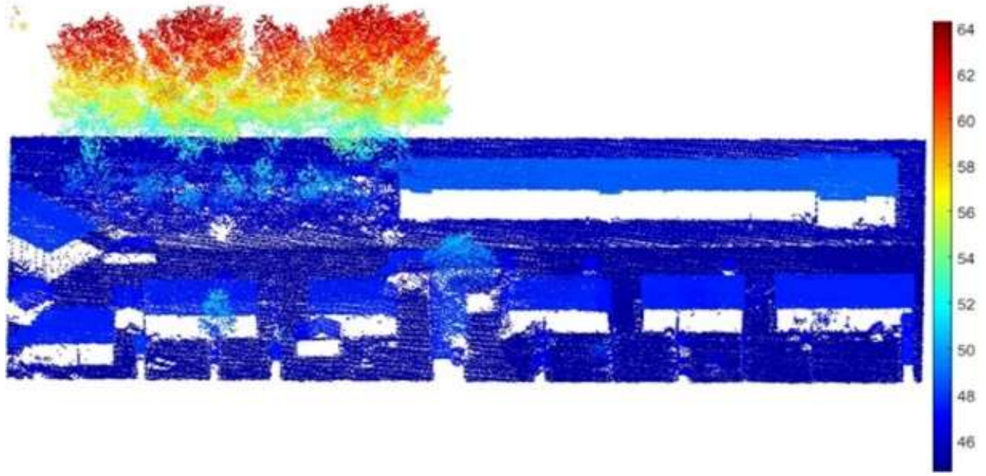
$$\forall P_j \in A: hp_i - \Delta h\big(d(p_i, p_j), m_i\big) \leq hp_j$$

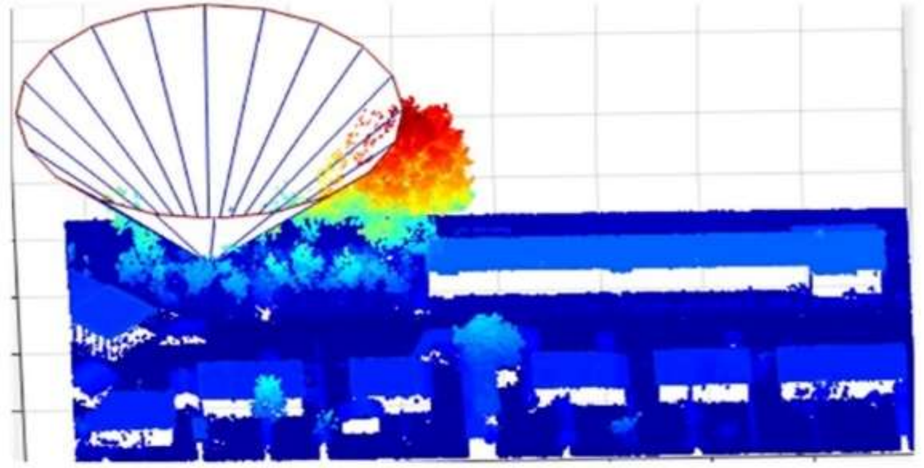
Where, P_j is a point in the dataset within the window excluding the point being examined;

hp_i and hp_j are the heights of the i points in the window and of the j point being examined,

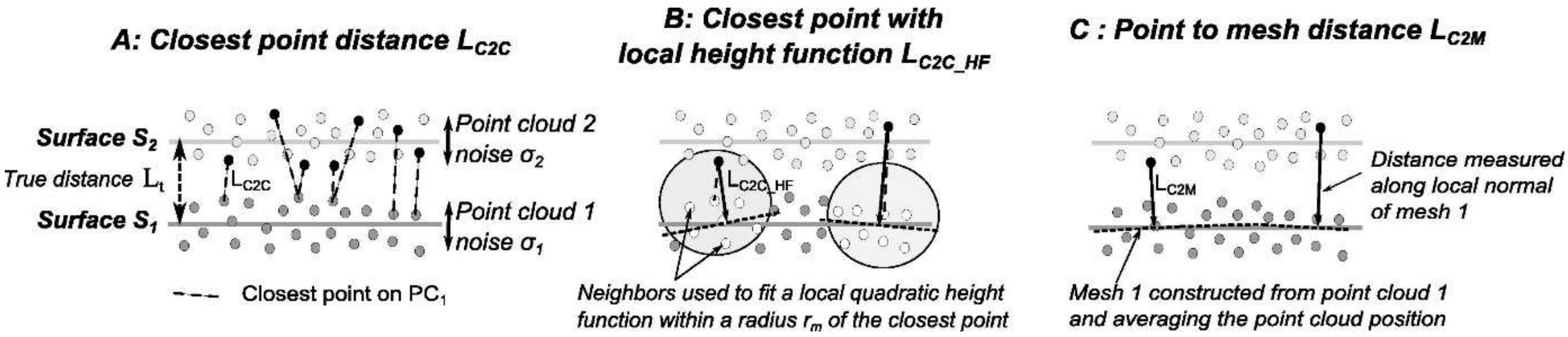
m is the absolute value of the cone's gradient, and the negative value of m is the height of the cutoff plane,

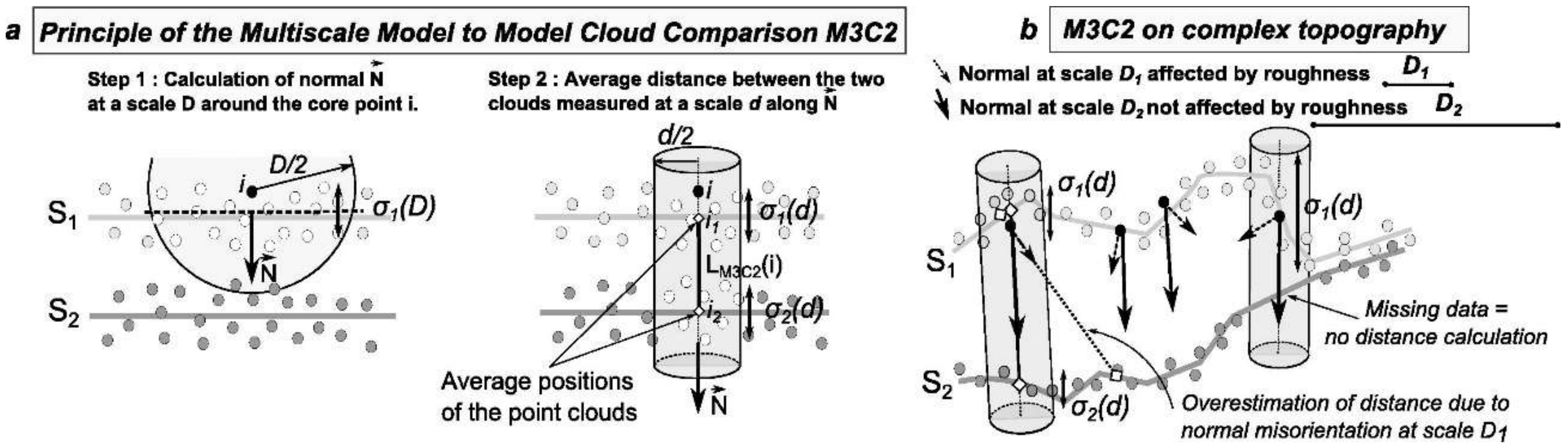
while A is the set of points to be filtered out.





Figures after Mahphood and Arefi, 2020





Figures from: D. Lague et al., 2013. ISPRS Journal of Photogrammetry and Remote Sensing 82, 10-26.

2. Comparing pointclouds

We will learn how to do this second part, during the afternoon workshop with Mr. Rikuto Daikai.

Conclusion of Lecture 04

What you should know:

(a) Understand what the LiDAR data are made of
(b) Know how to manipulate them in CloudCompare and with LidR
(c) Know how to extract elements from the dataset
(d) Be able to transform your raw data from the LiDAR into a DEM and a DSM

christophergomez@bear.kobe-u.ac.jp