Elaborated by : Walid Ghariani
Date : 24.08.2020

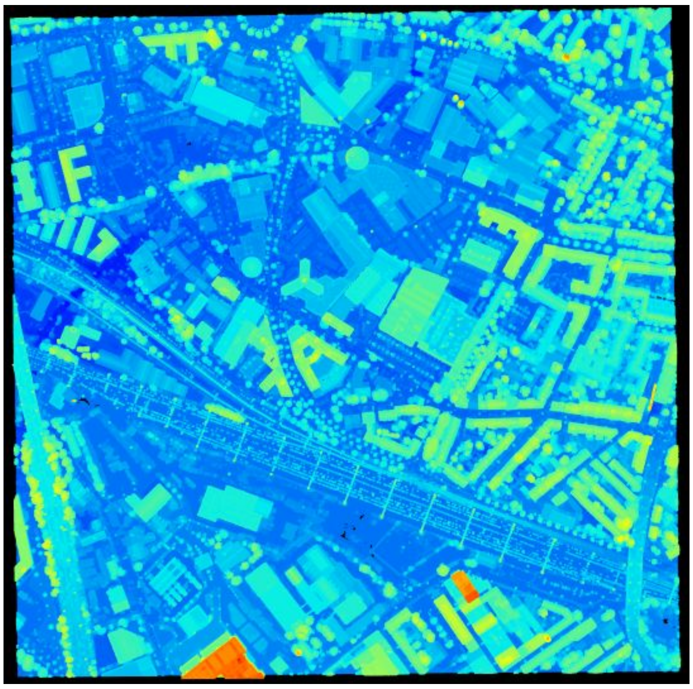
LiDAR Data Processing with R and lidR package:

1. Methodology :

First Step: Read the file:

Laz file characteristics :

class : LAS (v1.2 format 1)
memory : **841.1 Mb**
extent : 364000, 365000, 5622000, 5623000 (xmin, xmax, ymin, ymax)
coord. ref. : +proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs
area : 1 km²
points : **11.02 million points**
density : **11.02 points/m²** : this is the point density of the point cloud data



Second Step : Generate the the DTM

Interpolates the ground points and creates a rasterized digital terrain model.

The algorithm uses the points classified as "ground" (Classification = 2 according to LAS file format specifications) to compute the interpolation. How well the edges of the dataset are interpolated depends on the interpolation method used. :

we used knnidw: It implements an algorithm for spatial interpolation. Interpolation is done using a k-nearest neighbour (KNN) approach with an inverse-distance weighting (IDW).

Parameters of knnidw:

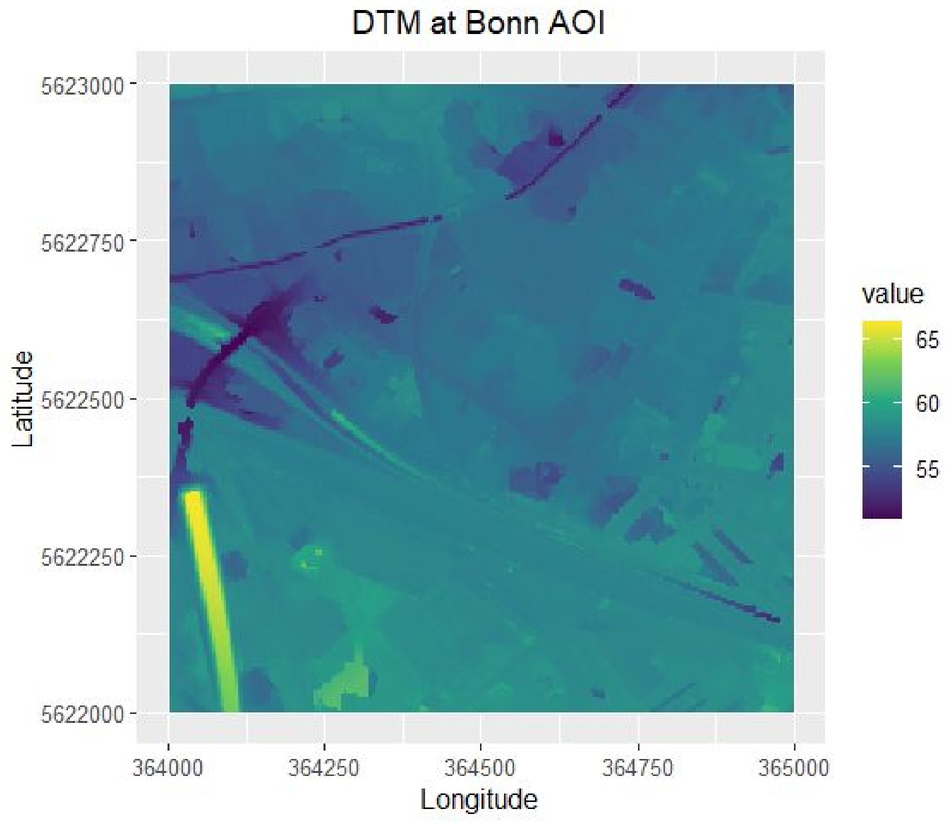
k : integer. Number of k-nearest neighbours: 6L

p: numeric. Power for inverse-distance weighting. 2

DTM Characteristics:

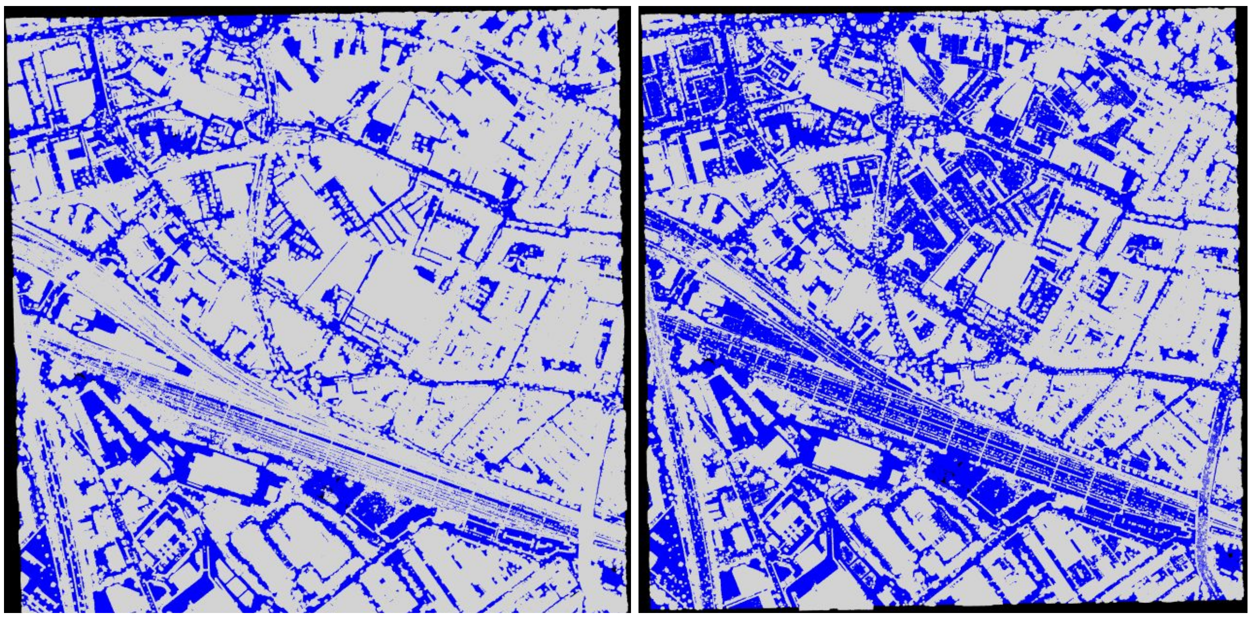
class : RasterLayer
dimensions : 1000, 1000, 1e+06 (nrow, ncol, ncell)
resolution : 1, 1 (x, y)
extent : 364000, 365000, 5622000, 5623000 (xmin, xmax, ymin, ymax)

```
crs        : +proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs
source     : memory
names      : Z
values     : 51.01, 66.5  (min, max)
```



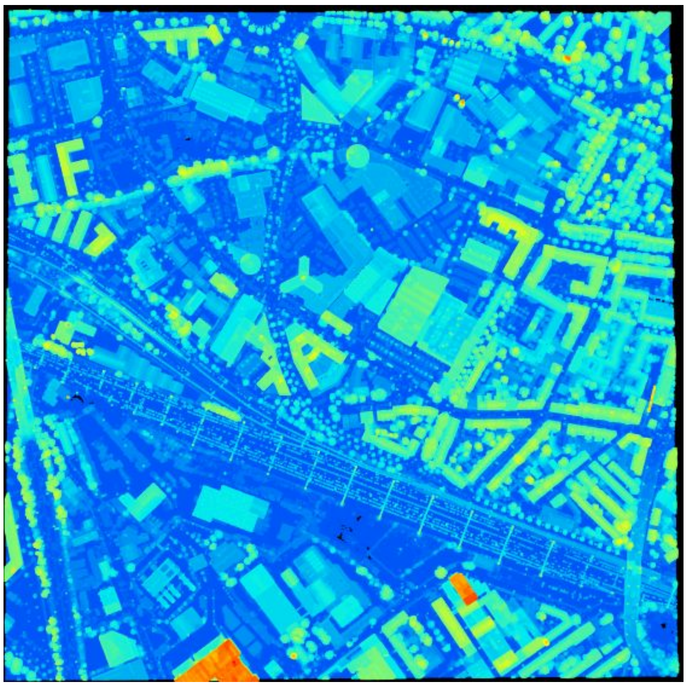
# Optional :
Although the function and the implemented algorithm takes into account the points classified as "ground" (Classification = 2) to generate the DTM
we can also use beforehand either a Progressive Morphological Filter (Zhang et al. 2003) or a Cloth Simulation Filter (Zhang et al. 2016) to classify the point clouds into ground and non-ground.

Progressive Morphological Filter **Cloth Simulation Filter**



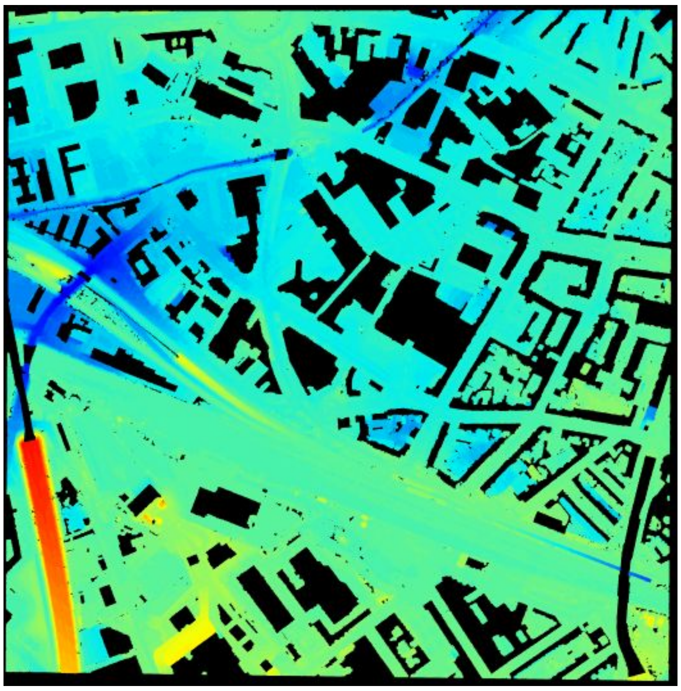
Third Step: We Normalize the lidar data = Remove the topography from a point cloud.
Description
Subtract digital terrain model (DTM) that we already computed from LiDAR point cloud to create a dataset normalized with the ground at 0. In this case the algorithm does not use rasterized data and each point is interpolated. There is no inaccuracy due to the discretization of the terrain and the resolution of the terrain is virtually infinite.
Characteristics:

class	: LAS (v1.2 format 1)
memory	: **967.2 Mb**
extent	: 364000, 365000, 5622000, 5623000 (xmin, xmax, ymin, ymax)
coord. ref.	: +proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs
area	: 1 km²
points	: **11.02 million points**
density	: **11.02 points/m²**

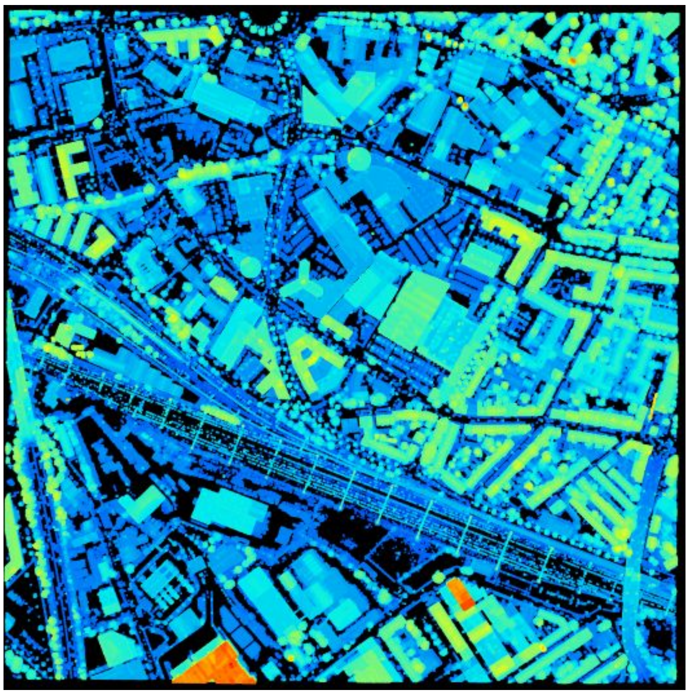


Fourth Step: We filter the point clouds from the first return classified as ground.

The following graph represent the first returns



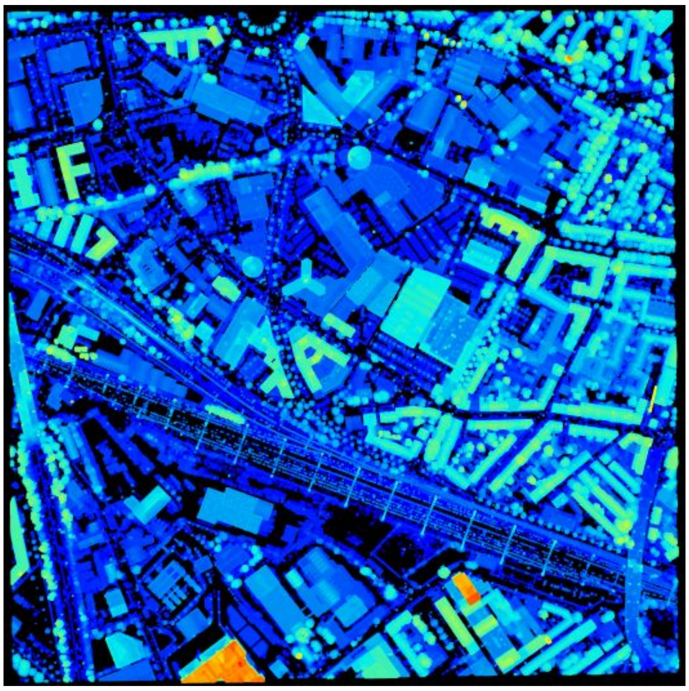
The following graph represent the filtered lidar data from the first return



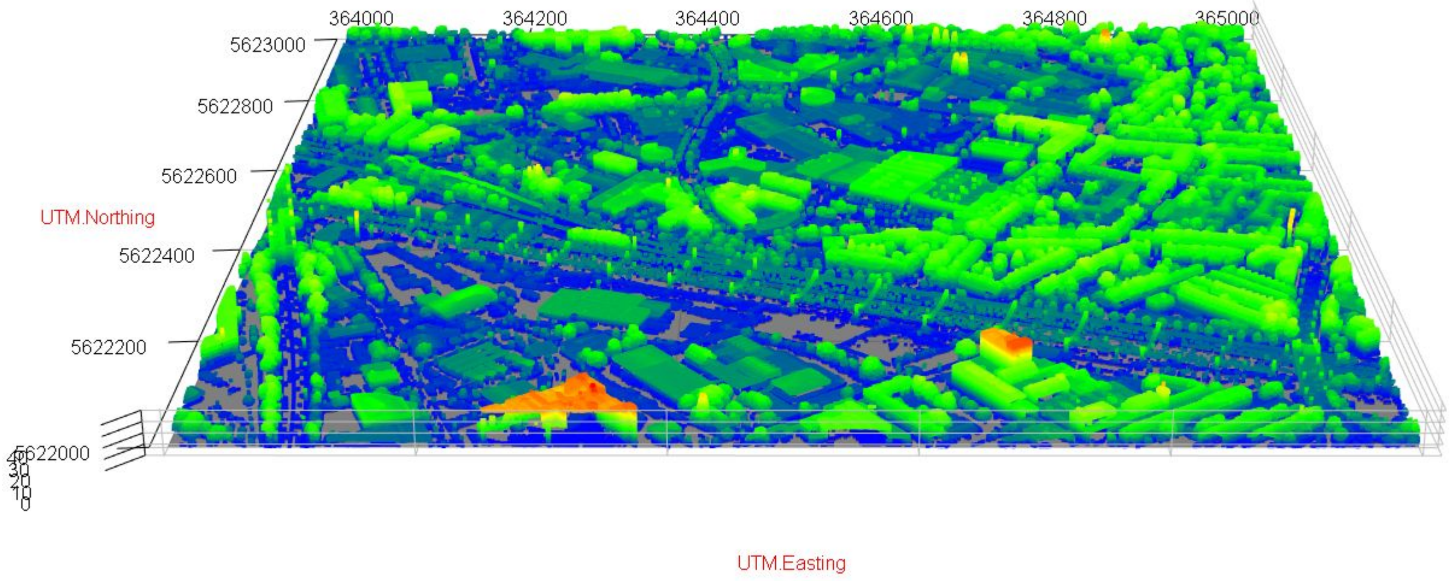
class   : LAS (v1.2 format 1)
memory   : **548.3 Mb** notice decrease fo the memo
extent   : 364000, 365000, 5622000, 5623000 (xmin, xmax, ymin, ymax)
coord. ref. : +proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs
area   : 1 km²
points   : **6.53 million points**
density   : **6.53 points/m²**

Fifth step : We remove the outlines :
we remove/filter some points that are considered as errors/outliers they usually have negative values ; and it can be seen when generating the nDSM.

class : LAS (v1.2 format 1)
memory : **545.3 Mb** (u can see that decease of the memo when removing the outlines)
extent : 364000, 365000, 5622000, 5623000 (xmin, xmax, ymin, ymax)
coord. ref. : +proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs
area : 1 km²
points : **6.5 million points**
density : **6.5 points/m²**

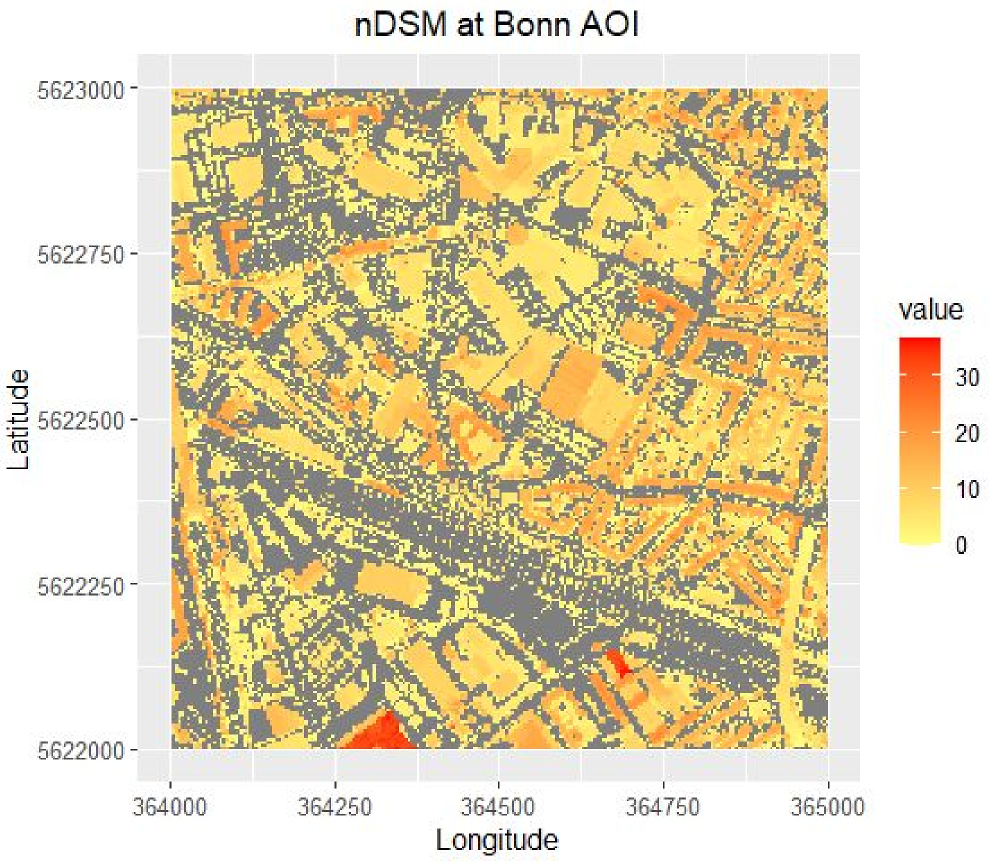


3D. VIZ with rLiDAR



Sixth Step : We generate the nDSM aka Height
the Algorithm used: p2r {lidR}
It implements an algorithm for digital surface model computation based on a points-to-raster method: for each pixel of the output raster the function attributes the height of the highest point found. The subcircle

tweak replaces each point with 8 points around the original one. This allows for virtual 'emulation' of the fact that a lidar point is not a point as such, but more realistically a disc. This tweak densifies the point cloud and the resulting nDSM is smoother and contains fewer 'pits' and empty pixels.
the parameters:
res = 1 (1m)
p2r()
The grey values are not the outlines here they are just the NA values when we removed the Ground.

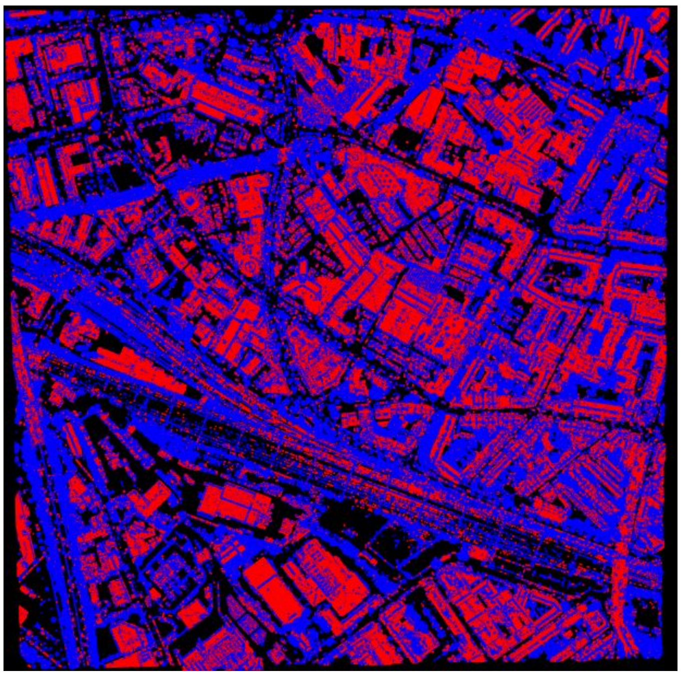


```
class       : RasterLayer
dimensions : 1000, 1000, 1e+06  (nrow, ncol, ncell)
resolution : 1, 1  (x, y)
extent      : 364000, 365000, 5622000, 5623000  (xmin, xmax, ymin, ymax)
crs         : +proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs
source      : memory
names       : Z
values      : 0, 40.83  (min, max)
```

2. Experimental part: LiDAR data segmentation using Algorithms for shape detection of the local point neighborhood

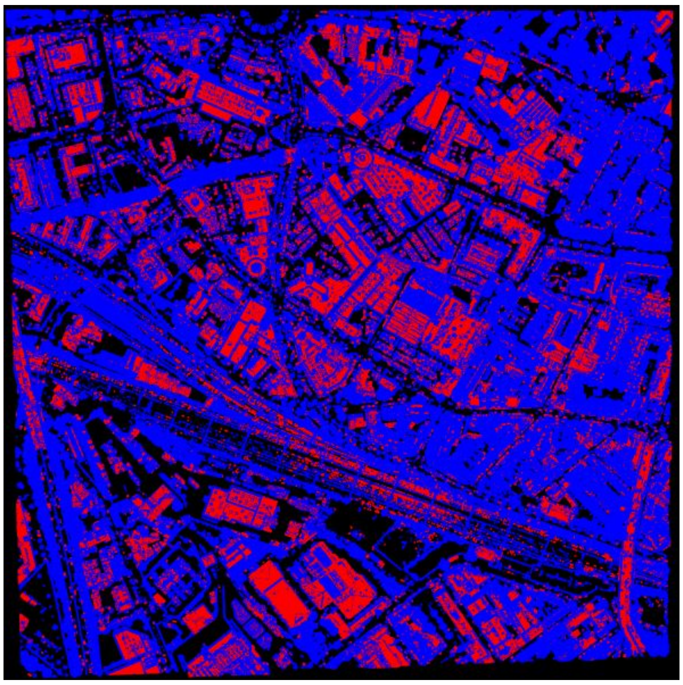
2.1. Detection of plans based on criteria defined by Limberger & Oliveira (2015):

It's possible with this algorithm to get a good segmentation differentiating the rooftops (red color) from the rest of the shapes (blue color): trees, streets, etc.

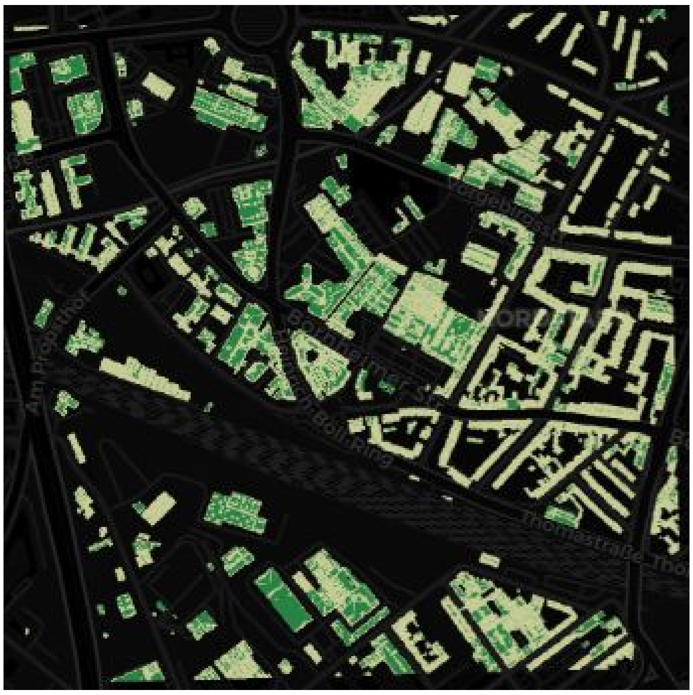


2.2. The same as 'plane' but with an extra test on the orientation of the Z vector of the principal components to test the horizontality of the surface.

The segmentation is also quite good for classifying the rooftops in general but performed much better from the previous algorithm in classifying the flat rooftops.



So the Detection of plans algorithm is suitable for classifying the rooftops in general but the algorithm based on the addition of an extra test on the orientation of the Z vector of the principal components is much suitable for classifying the flat rooftops. Our baseline for this comparison is the slope layer which enabled us to categorize the rooftops in flat rooftops (0-5°) (green color), and non-flat rooftops (> 5°) (pale yellow color).



2.3. Detection of lines inspired by the Limberger & Oliveira (2015).

A point is labelled TRUE if the neighborhood is approximately linear: the detected lines seems to represent the power lines network