Deep Learning for 3D Vegetation Analysis
or
How I Struggled with Trees for 2 Years

Ekaterina Kalinicheva

It is not about the destination, but about the journey

Figure: Me very much enjoying the journey while swimming in a very-very cold lake after walking in the rain somewhere in the Pyrenees.

Project Overview. (What we thought I would do)

Goal

Automatic analysis of 3D vegetation coverage using multi-modal data:

1. Creation of tree database
2. Species classification, individual tree extraction
3. Extraction of morphological parameters

Project Overview. (What we thought I would do)

Goal

Automatic analysis of 3D vegetation coverage using multi-modal data:

1. Creation of tree database
2. Species classification, individual tree extraction
3. Extraction of morphological parameters

Multi-modal data

Study area close to Bordeaux:

- 3D forestry data (LiDAR)
- VHR hyper-spectral images
- Satellite image time series (Sentinel-2)

What have I actually done?

1. Database creation (a lot of classes of trees and bushes).
2. Trying a lot of different DL methods for indoor scene analysis.
    - Spoiler: nothing works.
3. Reflection on what can be done with data.
4. Multi-layer vegetation analysis using only 3D data.
5. Test of different multi-modal approaches.
6. Individual tree detection.

WildForest 3D. What's inside:

WildForest3D dataset:

- 2000 individual trees and bushes with morphological parameters and species (Forest Inventory).
- 29 plots of dense forest, 7 million 3D points, 2.1 million individual labels.
- 31 classes of trees and bushes.

WildForest 3D. What's behind:

What we had to start with?!

- Excel table with tree positions and some parameters.
- Aerial LiDAR 3D point clouds (low point density close to the ground).

What we thought would work?!

- Semi-automatic tree extraction based on tree coordinates.

What we got?!

- Coordinate errors up to 5 meters or even no tree at all.
- 3-5 trees that grow from the same point.

What I did?!

- Manual clipping and reclipping of the "preliminary dataset".

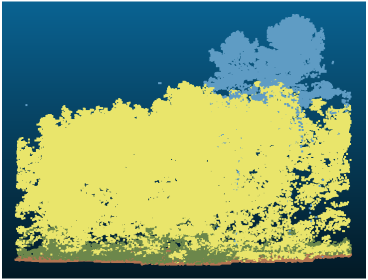
What would be better?!

- Do everything manually from the beginning.

Initial Plan

Do semantic segmentation for 4 classes:

- Expectation: Trees (deciduous + coniferous), bushes, ground.
- Reality: Almost everything is classified as deciduous trees, classified bushes are in reality grass (which is not annotated). Fail!☹
- Why Fail? Some bushes are too tall and some trees are too small, ground vegetation is not annotated.
  $\rightarrow$ Classes are not well defined.

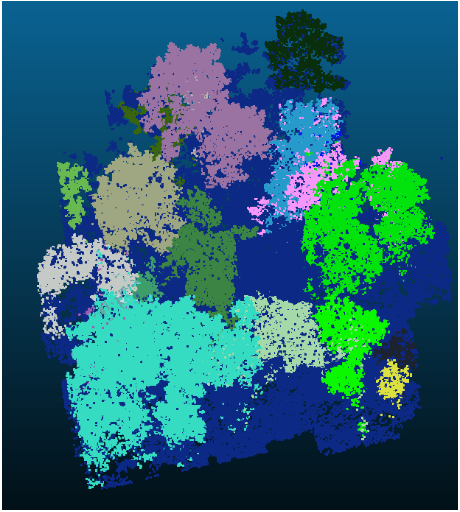


Brown - ground, Green - bushes, yellow - deciduous, blue - coniferous

Second Plan

Do Instance Segmentation
= Tree identification

- PointGroup,
  based on MinkowskiNet
- Research of the instances in the segmented results
- Shapes are ok, but
  trees are not separated
- Fail again! ☹ ☹ ☹

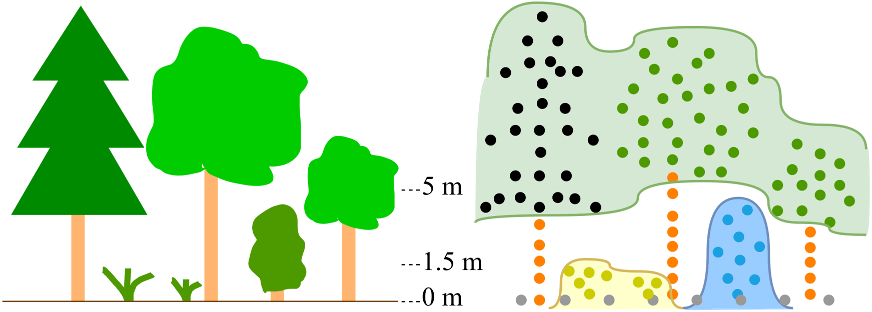


Is there anything I can do?

Multi-layer Vegetation Analysis from 3D Point Clouds

- A deep learning method to model the **multi-layer structure of dense forests** from airborne LiDAR scans.
- Generate **high resolution meshes and occupancy maps** of three different vegetation layers.
- WildForest3D adaptation to the multi-layer analysis.

Vegetation Multi-Layer Structure



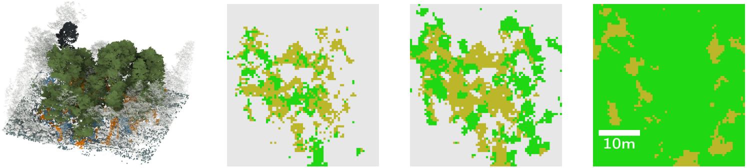
point labels		vegetation layers
ground	coniferous	overstory
ground vegetation	deciduous	understory
understory	stem	ground veget.

WildForest3D

Adaptation to multi-layer structure analysis (6 classes + 3 layers):

- 3D annotations :
  ground (z=0m), ~~ground vegetation~~,
  understory, deciduous, coniferous, stems.
- 2D vegetation occupancy maps for 3 layers:
  $\Rightarrow$ derivation of ground vegetation.

WildForest3D



(a) Annotated 3D Point Cloud.

(b) GV Occupancy.

(c) Understory Occupancy.

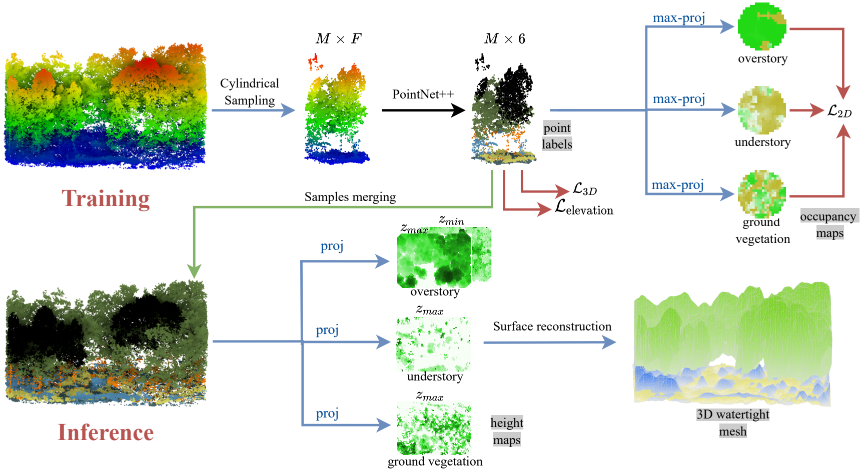
(d) Overstory Occupancy.

full

empty

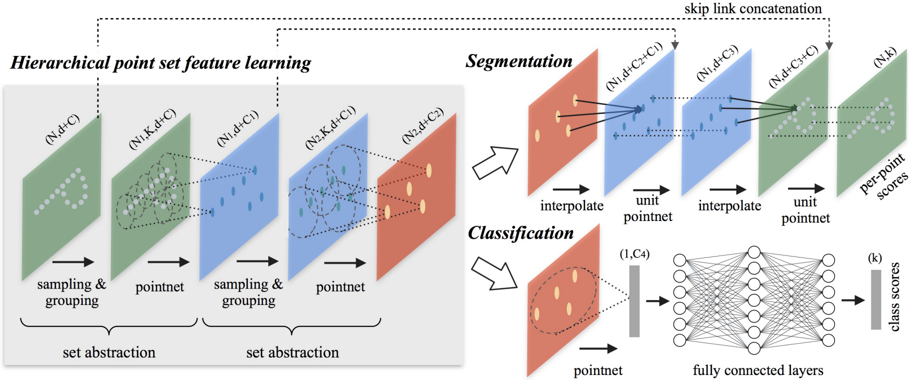
no data

Model



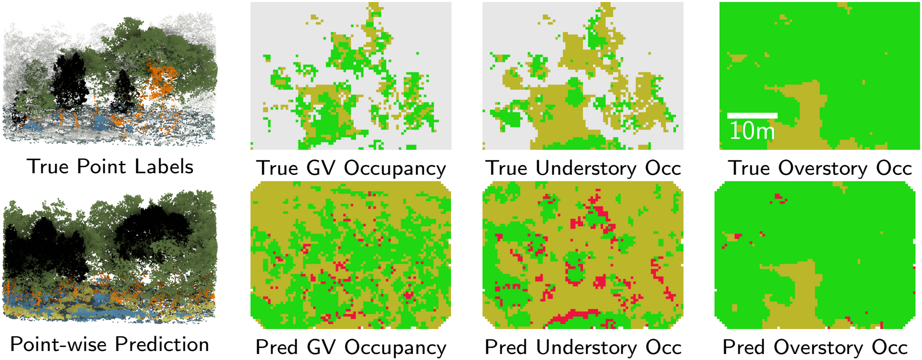
$M \times F$
$M \times 6$
Cylindrical Sampling
PointNet++
point labels
max-proj
max-proj
max-proj
overstory
understory
ground vegetation
occupancy maps
$\mathcal{L}_{2D}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{elevation}$
Training
Samples merging
Inference
proj
proj
proj
z_{min}
z_{max}
overstory
z_{max}
understory
z_{max}
ground vegetation
height maps
Surface reconstruction
3D watertight mesh

PointNet++



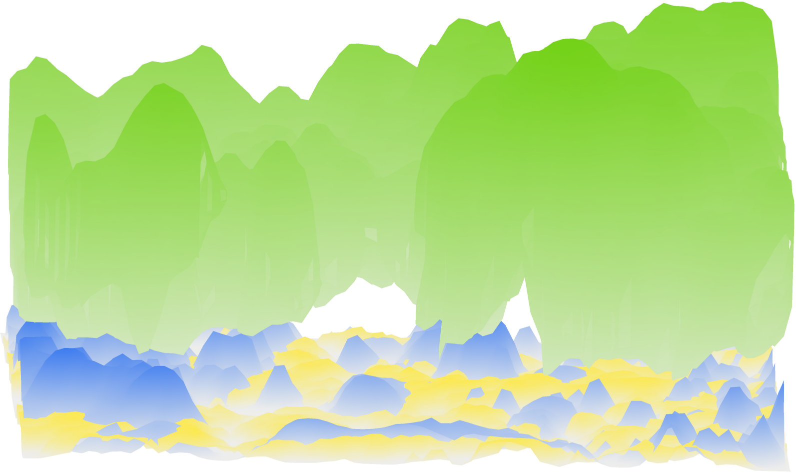
Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: deep hierarchical feature learning on point sets in a metric space, NIPS'17.

Results



Qualitative Results. Ground truth (top row) and prediction (bottom row) for the point labels and layer occupancy maps.
The prediction errors are in red.

Results



Three vegetation layers modeled as a **3D mesh**.

Results

- Difficult to evaluate as GT is incomplete and subjective.
- Height Regression Performance: MRE from 3 to 25%.
- 2D Occupancy maps performance:
  - GV, Overstory - very good, Understory - decent IoU=60%
- 3D Classification performance
  - IoU coniferous = 25%
  - IoU stems = 15%
  - IoU understory = 45%

What can we do to improve the results?

Multimodal and Bi-temporal Data

Data:

- 3D point clouds Summer 2019 (annotated),
- 3D point clouds Winter 2021 (non-annotated),
- VHR Aerial images 2019 (annotated from point clouds).

4 possible combinations :

- 3D point clouds Summer,
- 3D point clouds Summer + Images,
- 3D point clouds Summer + Winter,
- 3D point clouds Summer + Winter + Images,

Multimodal and bi-temporal approach

Expectation:

- Improvement of precision,
- Detection of more classes (7 vs 6) :
    - Deciduous : oaks + others (mostly alder),
- Classification of Winter data.

Multimodal and bi-temporal approach

Expectation:

- Improvement of precision,
- Detection of more classes (7 vs 6) :
  - Deciduous : oaks + others (mostly alder),
- Classification of Winter data.

Reality:

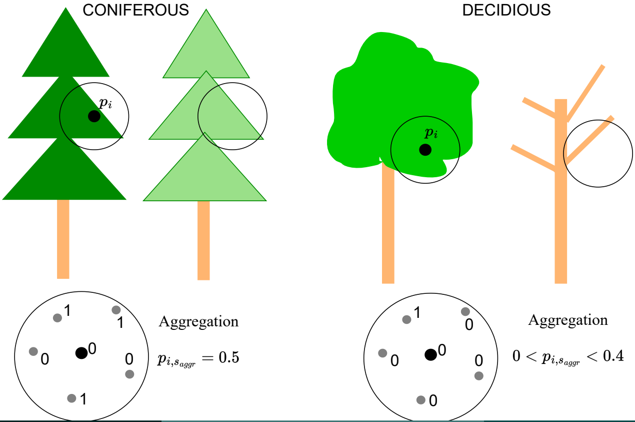
- It worked! (But not in the way we thought)

Annotated Aerial Images

- SegNet for image feature extraction F_{im},
- F_{im} are concatenated with point cloud features F_{3D} at the input of the model,
- Features are attributed only to points "visible from the top" (1 m),
- Two networks are trained at the same time.

Non-annotated 3D Winter Data

- Create a binary feature "season": 0-summer, 1-winter,
- Merge two point clouds,
- The network aggregates points at each step,
- If ratio of points summer/winter is 50/50 $\rightarrow$ coniferous, etc ...



Multimodal Results. All available data

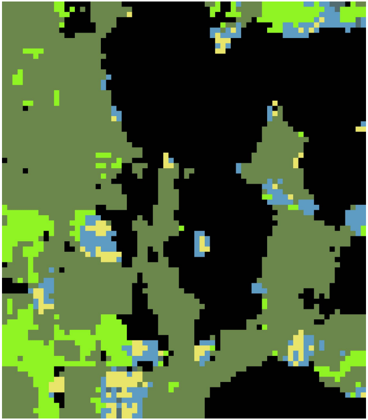
Output of the model with all the available data:

- Classification 3D - summer and winter,
- Vegetation occupancy maps for each stratum (summer and winter),
- Classification of aerial images (different from 3D classification, view from the top) $\rightarrow$
    - Explication of the bad performance.

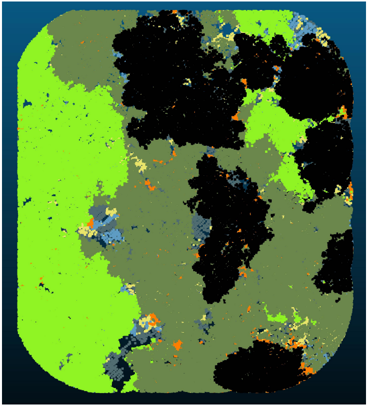
Multimodal Results

- The best : 3D summer + winter (mIoU = 59%),
- Okay : 3D summer + winter + image (mIoU = 56%):
  - Not a lot of annotated 2D data vs a lot of annotated 3D points,
- Bad : 3D summer + image,
- The worst : 3D summer.

Multimodal Results. Data : 3D summer + winter + image.



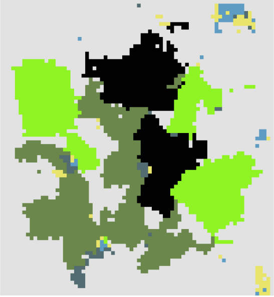
2D Image classification.



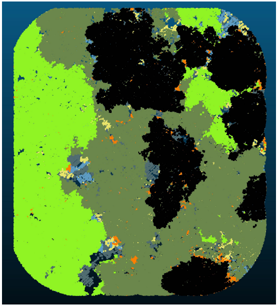
3D classification (top view). Summer.

oaks	coniferous	stem	understory
alder + others		ground veg.	ground

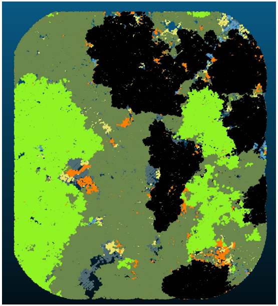
Multimodal Results. 3D Classification. Summer



Ground Truth.



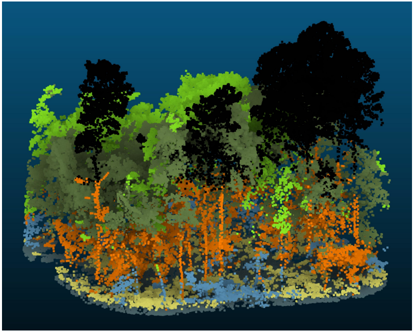
3D classif. Input data: 3D summer + winter + image.



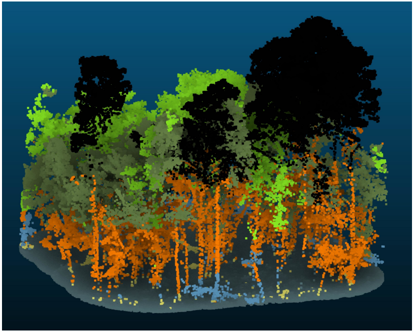
3D classification. Input data: 3D summer + winter.

oaks
alder + others
coniferous
stem
ground veg.
understory
ground

Multimodal Results. Input Data: 3D Summer + 3D Winter



3D Classification. Summer.



3D Classification. Winter.

oaks	coniferous	stem	understory
alder + others		ground veg.	ground

Multimodal Conclusion

The best results with bi-temporal approach :

- Winter data classification is visually good and “copy” the summer $\rightarrow$
  - Possible to identify seasonal vegetation (fougère, ...),
  - Analysis of species growing,
  - Change detection, etc...
- Coniferous are better detected :
  - Test : mIoU = 80% now vs 25% before,
- Identification of alder trees,
- Stems and understory are better classified.

I still have some time left...

Individual Tree detection

What can we do?

- Panoptic segmentation?
- Object detection?
- What algorithm if nothing worked before?
- Partial data annotation, how to deal with it?

Individual Tree detection

What can we do?

- Panoptic segmentation?
- Object detection?
- What algorithm if nothing worked before?
- Partial data annotation, how to deal with it?

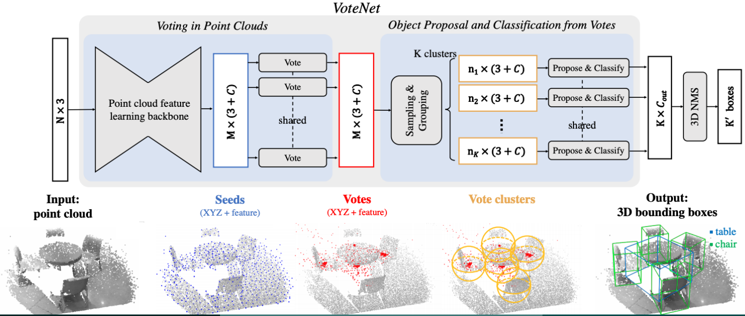
Individual tree detection with 3D bounding boxes:

- VoteNet (network based on PointNet)
- Only trees with H>5 m (overstory)
- 2 classes : deciduous and coniferous

VoteNet

Votenet predicts BB parameters on the sampled points:

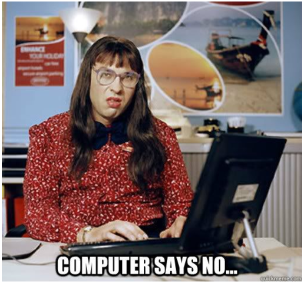
- Feature extraction with the backbone model (PointNet++)
- Compute offsets and residual features for each seed (1D conv)
- Compute clusters-propositions (SA)
- For each proposition, compute BB parameters (1D conv):
  - center offset, size parameters of the box, objectness score, class
- Postprocessing



How to adapt Votenet to Trees?

Problems:

- Partially annotated data
  $\rightarrow$ Background is not well defined
  $\rightarrow$ Difficult to train and validate
- No obvious boundaries between trees
- BB occupation is not homogeneous, bad symmetry
  $\rightarrow$ computer struggles to interpret the GT
- Trees all look the same!!!



VoteNet Losses

A lot of them, computed for different outputs.
Each output is associated to the real points from the initial point cloud!

- Votes loss (dist from seed GT votes to predicted vote coordinates)
- Objectness Loss (Cross Entropy)
- Classification Loss (Cross entropy)
- Size and ~~Heading~~ loss (Huber Loss)
- Center loss (Euclidean distance to GT from predicted center of BB)

VoteNet Losses

A lot of them, computed for different outputs.
Each output is associated to the real points from the initial point cloud!

- Votes loss (dist from seed GT votes to predicted vote coordinates)
- Objectness Loss (Cross Entropy)
- Classification Loss (Cross entropy)
- Size and ~~Heading~~ loss (Huber Loss)
- Center loss (Euclidean distance to GT from predicted center of BB)

Solution:

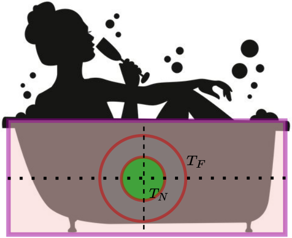
- Loss is computed only for annotated points (summer) and their neighbours (winter)
- Close to ground points are penalized

How to compute objectness loss??

Objectness Score and Objectness Loss

How to create objectness labels?

- Define Near and Far Thresholds ($T_N = 0.3m$ and $T_F = 0.6m$)
- For each aggr. vote compute distance D_s to the closest GT BB center
- If $D_s < T_N \rightarrow s$ can vote for an object (1)
- If $D_s > T_F \rightarrow s$ can not vote for an object (0)
- If $T_N < D_s < T_F \rightarrow s$ in the grey area (does not participate in loss)



Objectness Score and Objectness Loss

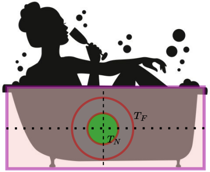
How to deal with dense forests with partially annotated tall trees?!

Objectness Score and Objectness Loss

How to create objectness labels?

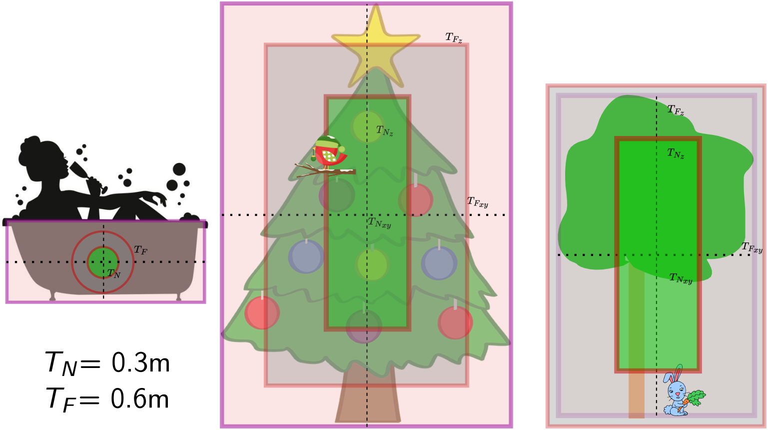
- Only annotated points and their close neighbours participate in loss
- Define Near and Far Thresholds in XY and Z planes
- First compute horizontal distance, then vertical
- Almost no restrictions in height, far threshold is really far!
    - Non annotated neighbouring trees can not vote for the annotated neighbour,
      but they will not be penalized in the loss function
    - Center of BB is more important than height
    - Almost no non-voting labels are produced!

Objectness Labels. Imagine it is in 3D.



$T_N = 0.3$m
$T_F = 0.6$m

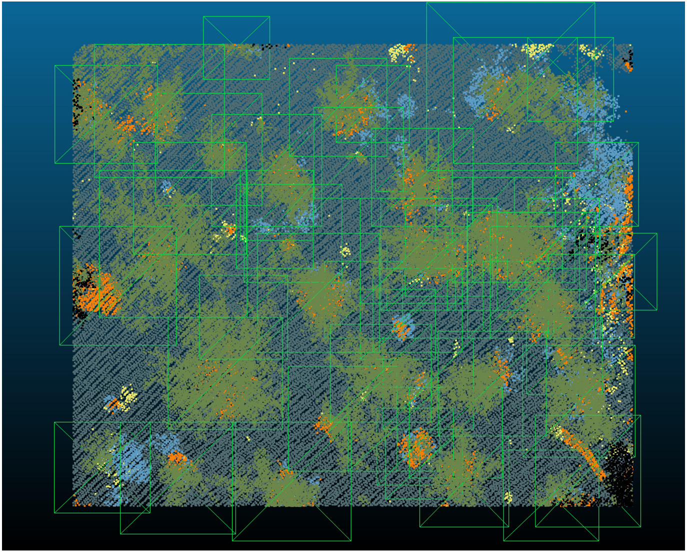
Objectness Labels. Imagine it is in 3D.



$T_N = 0.3\text{m}$
$T_F = 0.6\text{m}$

$T_{N_{xy}} = 1.5\text{m},\ T_{N_z} = 7\text{m}$
$T_{F_{xy}} = 4\text{m},\ T_{F_z} = 15\text{m}$

It worked, but results could be better



Perfect plot with sparse trees. Good for qualitative validation.

Non-maximum Suppression

Non-maximum Suppression (NMS) did not do its job properly?

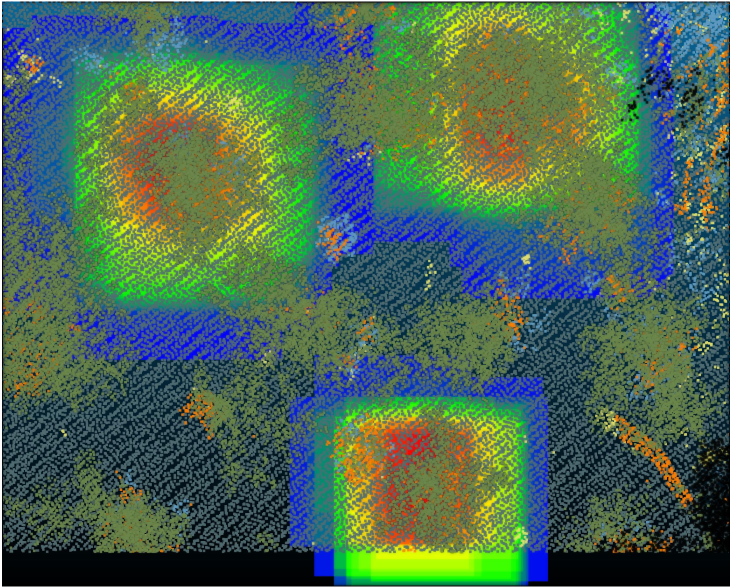
1. A lot of BBs are predicted
   - Within each cylinder
   - Cylinders overlap, grid<=R
2. Predictions are sorted in descending order
3. Take the first, suppress all the overlapping boxes with $O >= 25\%$
4. Take the second, suppress...
5. Repeat until nothing left

BB Heatmaps instead of Non-maximum Suppression

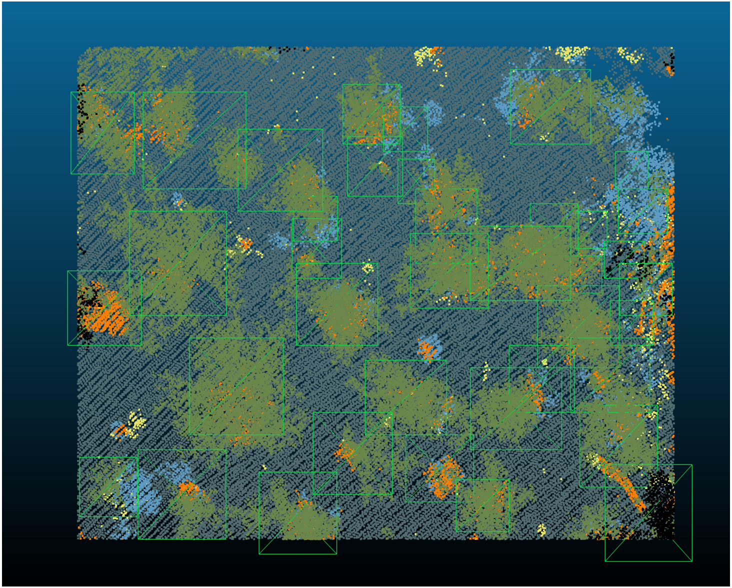
BB center and size from heatmaps:

1. A lot of BBs are predicted
2. Predictions are sorted in descending order
3. Take the first BB,
   group it with all the overlapping boxes in XY with $O >= 25\%$
4. Delete all those boxes from the list
5. Take the second, group...
6. Repeat until nothing left
7. For each group create a heatmap using prediction probabilities
8. Compute mean and std for each axis from PDF
9. Height=weighted height of the group

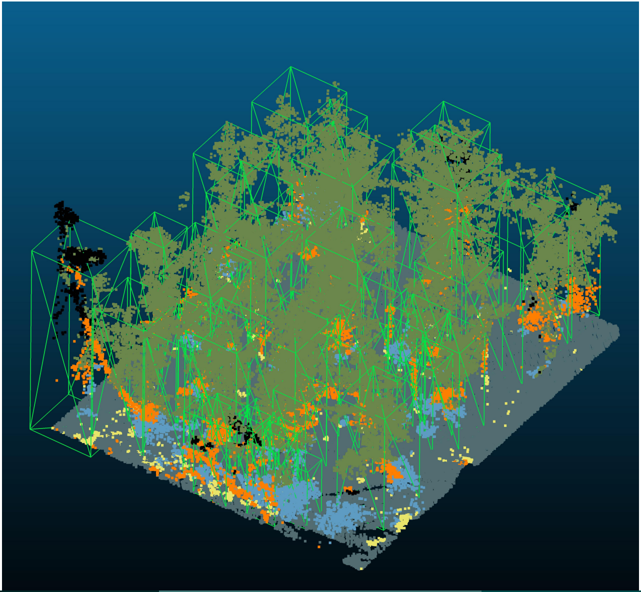
BB Heatmaps instead of Non-maximum Suppression



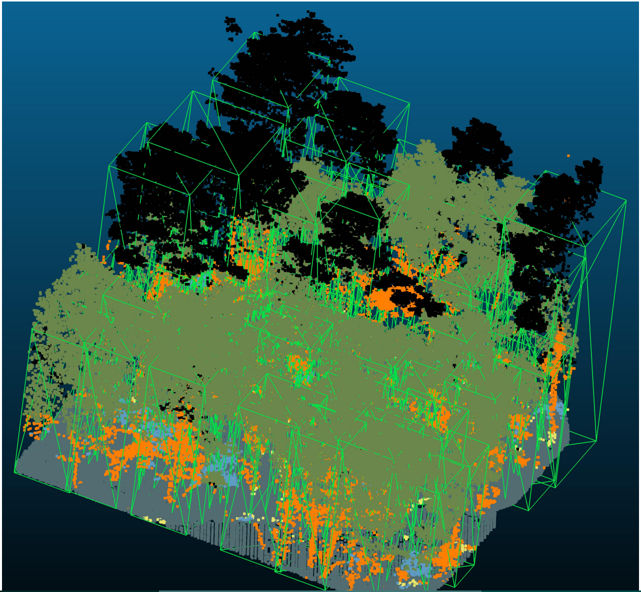
BB Heatmaps Results



BB Heatmaps Results



BB Heatmaps Results



BB Results. Problems

- Predictions are better than GT
  $\leftarrow$ Predictions are centered on “tree barycenter”
- GT is not perfect
  $\rightarrow$ Quantitative results (IoU) are better for bigger predicted boxes
- Qualitative results are good, but difficult do estimate for dense areas
- How to justify that the results are good?

BB Results. Problems, but Not Problems

- Can be combined with multimodal stratum analysis model:
  - Somehow does not improve BB predictions☹
  - But... The whole model looks impressive☺
- Model works better with bi-temporal data,
  but changes are not detected☹
- Ecologists are not that much interested and impressed by individual tree detection☹
- A lot of time-consuming post-processing is needed to obtain good results😐

Final Conclusion

Don't lose hope, it will work somehow one day!