

# Starting project report

Ines VATI

## 1 Article summary

The goal of the studied article [thomas\_semantic\_2018] is to classify each point of a 3D point clouds using an innovative approach to design features which will be employ by a random forest classifier.

**Main contributions.** The authors leverage a multiscale approach that has been proven to be more accurate. Their features computing with multiscale spherical neighborhoods are more effective than state of the art features and scales well to large scale datasets.

## 2 First Implementation results

My code is available on <https://github.com/InesVATI/npm3d-project>.

I want to implement the method from scratch as no code were available. I will apply their method on the Paris-rue-Cassette dataset<sup>1</sup> (12 millions points). The ground truth raw contained a large number of classes. I had to parse an XML file to group together some label classes to fairly compare the results with those obtained in the articles. For this dataset, the number of points for each classes is given in the table ???. The classes are highly unbalanced.

**Multiscale features computation.** Let consider  $N$  points of interest for which we want to compute the multiscale features. The features are given in the table ???.  $S$  is the number of scales. The features are computed using the following steps:

1. At scale 0, I compute the eigenvalues and eigenvectors of the covariance matrix of the neighborhood with a radius  $r_0$  and I compute the 18 features for each point of interest at this scale
2. For each scale  $s$ , I apply a grid subsampling on the original cloud with a voxel size  $r/\rho$ . Each original point is assigned to the closest voxel center and will have the same features as the voxel center.
3. I compute the 18 features for each voxel center associated with at least one point of interest at this scale, on a spherical neighborhood of radius  $r_s = r_0 * \rho^s$
4. I then obtain a feature matrix of size  $N \times 18 * S$ .

For now, the feature computation is quite long. I applied the classification on the MiniLille (1,901,853 and 2,500,428 points) and MiniParis (4,159,318 points) datasets from [NPM3D benchmark](#). This dataset is small but it allows me to evaluate how the method generalizes to unseen data. It took 370s to compute the features of 9,600 points.

I thought about computing the feature of all points in the Paris-rue-Cassette dataset and save the result but it take too much time. I am thinking about using JAX for improve the computation time but I am not sure that it will be enough.

---

<sup>1</sup><http://data.ign.fr/benchmarks/UrbanAnalysis/>

### 3 Perspectives

The things that I plan to do are :

- improving computation time using JAX ?
- plotting histograms of the neighborhood size for different scales to check they are not too small
- comparing the proposed method with a state of the art method on a small dataset (MiniLille or MiniParis) like the one proposed by [hackel\_fast\_nodate] or on a more recent method

**Improvement proposals.** For the project, we also have to propose improvements. I will propose to add height features as suggested by Hackel et al. [hackel\_fast\_nodate]. They are said to better describe the vertical and/or thin objects like traffic signs or tree trunks. I am also thinking of using another classifier different from a random forest like a Multilayer Perceptron algorithm.

#### A Label points distribution in the Paris-rue-Cassette dataset

Class name	Label	Number of points
Ground	1	4229639
Building	2	7027016
Traffic Signs	3	1355
Pedestrian	4	23999
Cars	5	368271
Vegetation	6	212131
Motorcycles	7	38330

Table 1: Number of points for each class for the Paris-rue-Cassette dataset. The label 0 is for the remaining unclassified points.

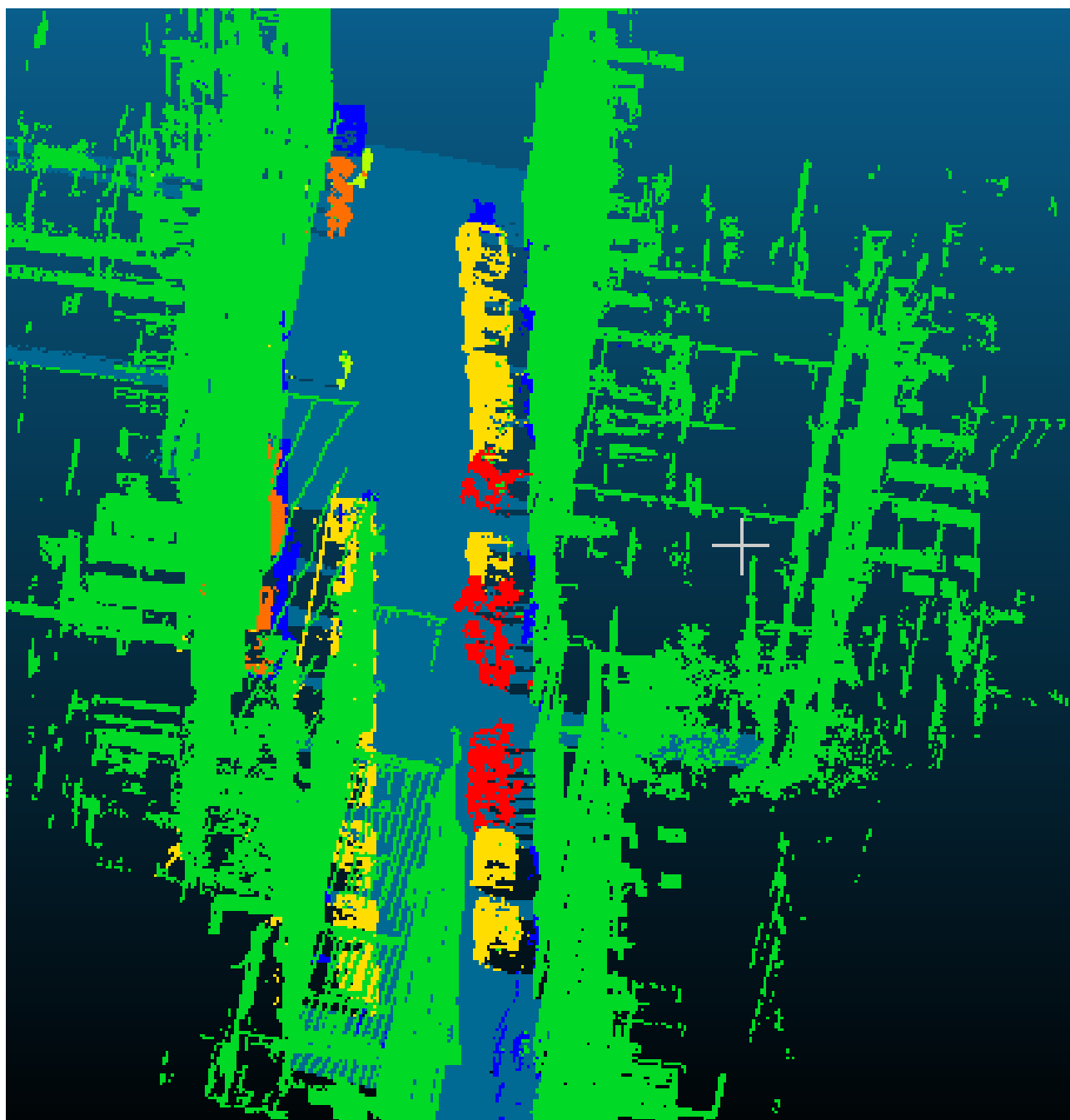


Figure 1: Ground truth for the Paris-rue-Cassette dataset with the 7 classes.

## B Multiscale features computation

Sum of eigenvalues	$\sum \lambda_i$
Omnivariance	$(\prod \lambda_i)^{(1/3)}$
Eigenentropy	$-\sum \lambda_i \ln(\lambda_i)$
Linearity	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity	$\frac{\lambda_3}{\lambda_1}$
Change of curvature	$\lambda_1 - \lambda_3$
Verticality ( $\times 2$ )	$ \arcsin(\langle e_i, e_z \rangle) _{i=1,3}$
Absolute moment ( $\times 6$ )	$\frac{1}{ \mathcal{N} }  \sum_{j \in \mathcal{N}} \langle p_j - \mathbf{p}, e_i \rangle^k _{k=1,2; i=1,2,3}$
Vertical moment ( $\times 2$ )	$\frac{1}{ \mathcal{N} }  \sum_{j \in \mathcal{N}} \langle p_j - \mathbf{p}, e_z \rangle^k _{k=1,2}$
Number of points	$ \mathcal{N} $

Table 2: Geometric features [thomas\_semantic\_2018] of point  $\mathbf{p}$  whose neighborhood is  $\mathcal{N}$ .