VECTORIZATION OF URBAN MLS POINT CLOUDS: A SEQUENTIAL APPROACH USING CROSS SECTIONS

E. Barçon[1]*, T. Landes[2], P. Grussenmeyer[2] G. Berson[1]

[1]Innovation department, TT Géomètres-Experts, 75011 Paris, France - (e.barcon, g.berson)@ttge.fr
[2]Université de Strasbourg, CNRS, INSA Strasbourg, ICube Laboratory UMR 7357,
Photogrammetry and Geomatics Group, 67000, France (tania.landes, pierre.grussenmeyer)@insa-strasbourg.fr)

Commission II, WG II/4

KEY WORDS: Mobile Laser Scanning, Vectorization, Point Clouds, Automation, Road features

ABSTRACT:

Dense point clouds acquired with a mobile laser scanning system (MLS) device become usual raw data for different surveyor applications: topographic maps, 3D models, road inventories, risk assessment of vegetation on road or railroads. Thanks to important evolutions in technologies, MLS devices became powerful and very popular. In the meantime, the need for point cloud automatic processing tools is growing. However, the available tools have not yet reached a sufficient level of maturity. Using MLS point clouds to produce topographic maps, BIM model or other deliverables, requires very often manual vectorization (or digitalization) work. In the road context, the transition from point cloud to road map that consists in delineating curb or road edges, road markings, pole, trees, facades etc. is currently performed manually. To reduce these time-consuming operations, several solutions have been proposed in the literature. In this paper we present the first results of a method consisting in vectorizing urban point cloud scene. The originality of this work is to propose a global approach aiming to detect and vectorize simultaneously multiple objects. The developed algorithm uses cross-section analysis to detect road curbs and vertical objects. The first results are promising, since an F-score higher than 80% has been reached, even before applying road logic rules or additional knowledge. The detection and extraction of vertical objects including facades, trees, and poles, is more challenging but the detections also present a recall greater than 85%.

1. INTRODUCTION

Mobile Laser Scanning (MLS) became a very popular technique to obtain, on large areas, dense points clouds associated with panoramic images (Ma et al., 2018). This acquisition technique is very fast and secured in an urban or rural road context compared to conventional surveying techniques with which the surveyor acquires only the points of interest. Usually, these raw data do not constitute the final product. Topographic map, 3D building models or other deliverables require to process the raw data captured with MLS devices. Detection and modeling of objects based on MLS data are time consuming operations because they are mostly performed manually. In this paper, the operation so-called "vectorization" consists in delineating road features in the point cloud so that they can be represented in a topographic map. Linear objects such as curbs or guardrails are described with 3D polylines whereas punctual objects like streetlamps or trees are defined by their insertion point in 3D on the ground.

2. RELATED WORKS

Point cloud processing has received increasing attention, especially since the successful application of Deep-Learning (DL) methods in the domain of image processing. However, the application of DL techniques to 3D point clouds is still an open research topic. Different approaches have been proposed in the literature and are briefly summarized in the following paragraphs. If a large amount of works is based on DL, more conventional approaches continue to be proposed and show also great performances.

A huge problem usually related in the literature dealing with MLS dense point clouds is the lack of natural spatial structure in the data. Different approaches have been proposed in order to define spatial relationships between points. In this part, we first describe the DL approaches and then focus on conventional methods.

2.1 Learning-based methods

2.1.1 Image-based methods: To benefit from the great results of Convolutional Neural Networks (CNN) on 2D images, Boulch et al. (2018) proposed to generate screenshots of point clouds and then perform a semantic segmentation using SegNet (Badrinarayanan et al., 2017). Finally, the semantic labels are reprojected on the points, knowing the positions and orientation of the snapshots. The number of required screenshots and their position is critical to perform an exhaustive segmentation of the scenes. The back projection of the labels from the screenshots to the point cloud is also a touchy operation. Error transmission can occur during back propagation of the label for two reasons: semantic segmentation errors are amplified during the reprojection. Moreover, multiple labels can be affected to the same point according to different screenshots. We can also notice that this approach doesn't really exploit the point cloud as 3D data as a voxel can be.

2.1.2 Voxel-based methods: This data structure is a 3D generalization of 2D pixels in images. The voxelization consists in transforming point cloud into a 3D grid. Zhou and Tuzel (2018) performed feature extraction and boundary box prediction for object recognition using an end-to-end trainable deep learning network based on a voxel grid structure.
Meng et al. (2019) combine a regular voxel grid and Radial Basis Function (RBF) to improve the learning capabilities and a

[*] Corresponding author

https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-351-2022

variational auto-encoder with group convolution to perform point cloud segmentation.
This data structure requires a huge amount of computing memory and leads to a loss of details equivalent to a spatial sub-sampling that smooths all details smaller than the voxel size.

2.1.3 Point-based methods: PointNet (Qi, Su, et al., 2017) and its multiscale improvement PointNet++ (Qi, Yi, et al., 2017) constitute the first neural networks able to deal with raw point clouds. Using shared Multi-Layer Perceptrons (MLPs), it learns pointwise features that allow classification and segmentation tasks. A lot of published work are based on this pioneer work but all of them can only handle a fixed number of points at the same time. That limitation makes the use of PointNet or PointNet++ for high density and large-scale point clouds very difficult. To deal with it, RandLA-Net (Hu et al., 2021) particularly use a random sampling technique, that reduces the need for memory and computation time. A specific local feature aggregation module is also described to compensate the loss of key features during the sampling step. We also note a recent proposal with lightweight Neural Networks (Puang et al., 2022).

Convolutional approaches such as KPConv proposed by Thomas et al. (2019) also deal directly with raw points. The authors define a rigid and deformable convolution operation. The weights of the kernel are localized in the Euclidean space. The kernel spatial continuity defined in a radius neighborhood allows to operate directly on the points without any other data structure.

2.1.4 Other structures

Landrieu and Simonovsky (2018) present a graph-based method for semantic segmentation. Basics shapes of the point cloud called “superpoints” constitute the vertices of the graph while the edges describe relationships between them. This structure allows a great reduction of the point cloud dimensionality while preserving local geometric complexity.
We can also cite lattice structures, as proposed in Alexandru Rosu et al. (2020). The sparse permutohedral lattices required less memory and computation since the simplices are only allocated if they are associated with point describing a surface of interest. Compared with a voxel approach, this structure allows to process larger scale point clouds.

2.2 Conventional approaches

As introduced before, most of the DL approaches have not solved the point cloud processing problem yet and that’s why conventional methods are still proposed and used. Actually, the neighborhood definitions and features extraction are currently too limited to reach the goal of 3D interpretation and vectorization of complex point cloud scenes. Two categories can be considered among the conventional methods: a) the scan line structure-based approaches and b) works that aim to realize road features inventory. Contrary to DL that aim to perform a full classification or segmentation of point clouds, these approaches focus on one or more specific objects.

2.2.1 Scanline-based approach: This data structure is linked to the acquisition process with rotative lidar heads that usually equip MLS devices. A scan-line is a set of points that are almost align along a straight line because they were acquired during the same spin of the lidar head. The scan-line structure is illustrated in figure 1. Most of the time, the scan-lines are oriented with a 45-degrees angle to the road direction. They are independent from the acquisition speed or point density and constitute a simple 2-dimensionnal structure.
Honma *et al.* (2019) propose a scanline-based technique that allows to segment flat zones, such as roads or sidewalks from irregular zones like curbs, facades, vegetation, or poles.
Using the same data structure, Yao *et al.* (2021) segment the road pavement with a height threshold along the scanline. In a second step, the authors threshold the intensity of the rasterized pavement’s points using (Bradley and Roth, 2007) adaptative thresholding method to segment road markings. Gézero and Antunes (2019) propose an automated curb break lines extraction process.

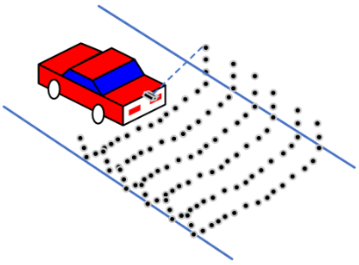

Figure 1 : scan-line obtained by MLS acquisition after Yao et al. (2021)

2.2.2 Road Features Inventory:

Some studies focus on a specific object. Usually, the proposed method depends on the nature and geometry of the search object. For instance, Safaie et al. (2021) propose a method using horizontal cross sections to locate and extract geometric properties of trees (trunks et foliage heights and diameters). Gao et al. (2020) describe a workflow using a modified DBSCAN clustering method for the extraction of guardrails in an urban context. Rodríguez-Cuenca et al. (2015) propose an approach to detect and delineate street curbs from MLS 3D point cloud data using a raster approach. Mi et al. (2021) propose another procedure on the same topic. A “supervoxel” generation enable to extract curb candidates, then a modified Euclidian clustering method allows to obtain long and linear clusters. The curbs are vectorized using Bezier curve fitting and Kalman filter is employed to complete road boundaries. Balado et al. (2020) propose a hybrid method using point cloud and panoramic images to realize a traffic sign inventory. They observe that without using the images, the classification of all the type of signs is not achievable.

Other studies propose workflows to detect simultaneously multiple road features. Umehara et al. (2021) for example, propose a two-step approach. In the first step, after a segmentation of the ground, buildings and utility lines, the remaining point cloud is divided into individual road features. In a second time, using a snapshot of the individual clusters, their natures are identified using YOLO v3. Justo et al. (2021) present an automatic solution for the inventory of traffic signs and guardrail. The method allows to generate IFC model of the road using predefined 3D customizable models. In a previous work, Barçon and Picard (2021) propose two different workflows aiming to extract road features from MLS data. The first one describes a process that extracts and vectorizes pole like objects in an urban context, whereas the second one performs the extraction of road markings and guardrails in a high-way environment.

In this part, the cited methods are very varied. Each of them focusses on one or a few objects and their outcomes are different (inventory, IFC model, 3D points or polylines). Therefor the proposed methods are diverse (raster, supervoxel, hybrid point cloud and image…). This show that point cloud processing is an active research topic and no standard or preferred method already exist. It also shows that DL isn’t the only approach to deal with point clouds, especially if the outcome isn’t a segmentation or classification. Some tendencies can be noticed, the use of horizontal slices for tree or pole-like objects or the use of horizontal ortho image for road marking detection. Unlike these studies, our goal requires global approach that can perform a vectorization as exhaustive as possible of all the objects in a point cloud scene.

3. DATASET

The tests are performed on a very dense point cloud captured along a Parisian Street with a Riegl VMX-450 MLS device. The point cloud used for the experiments covers an area of 75 x 26 m and contains 50 Mo points with a density around 64000 points per square meter. As the MMS device is equipped with two lidar heads oriented differently and the streets are scanned in both ways, the scan-line approach cannot be used. The dataset presents a typical road environment with trees, streetlamps, parked cars, facades, curbs, and other urban furniture. Screenshots of the dataset are presented in Figure 8.

4. PROPOSED APPROACH

As explained in the introduction, the goal of this work is to create a fully automatic processing chain for a systematic identification and vectorization of road features in an urban scene. The objects of interest are structural elements of streets, curbs or roads edges, facades, poles, and trees. The approach is entirely based on point clouds, without using sensors features or panoramic images, and should allow a large-scale use. As the objective is to perform a point cloud vectorization, a subsampling of the data is not being considered for now, as it can lead to a loss of details.

4.1 Data structure

To perform the vectorization of road features, the first obstacle encountered is the lack of natural spatial structure as mentioned in section 2. Therefore, it is very difficult to vectorize a point cloud directly in 3D space. It implies to detect (object recognition) and then extract or select one or more specific 3D position(s) on each object. A great challenge is that those precise points might be missing in the point cloud due to low density or occlusions.
That is why we propose to study the point cloud in three steps using multiple representations of the point clouds. For a given area, regular cross-sections, and a Bird-Eye-View (BEV) are generated as intermediary data. The choice of raster structures is obviously comfortable because we can take benefit from well-known and proven image processing tools. This structure also allows to visualize and evaluate the processing steps quickly and easily. We consider that using only BEV or cross-sections obtained from point cloud is insufficient to properly process the 3D point clouds.

4.1.1 Cross-sections are needed to finely observe vertical objects such as curb, poles, facades, guardrails, trees. It allows to handle vertical superpositions and to estimate objects heights. The best way to define the cross-sections is to orient them orthogonally to the street direction. Using that orientation, the street structure decomposition is easy to perform because street cross-sections are almost always identical. This specific orientation also allows to reduce the geometrical distortions associated with the further projections (section 4.3).
The use of external data such as open-source national streets databases appears to be the simpler and more efficient way to properly define and orient the cross-sections. Polylines in this database provide the direction of the road, from which the orthogonal direction is used to realize cross-sections. This data source is considered as more robust than the car trajectory which can include a lot of variations, such as loops, lane changes or multiple passes. It must be noticed that the cross sections are not relying on the sensors features such as GPS time or scan angles as it is the case in scan-line approaches (Gézero and Antunes, 2019; Honma et al., 2019; Yao et al., 2021)

4.1.2 Bird-Eye-Views provide a general overview of the area and counterbalance the loss of information associated with the study of independent cross-sections. It permits to complete the object detection because it allows a better detection of planar object such as road markings or manhole covers.

The BEV also represents a very helpful base map for the final step of the workflow that consists in a compilation of the cross-sections and BEV predictions. The alignment and correlation of the detected basic predictions will lead to the creation of 3D polylines and 3D points corresponding to the road features. In order to compute this final vectorization, a spatial and statistical study will be performed. The rules will enable to delete false positives and to perform some further detections to improve the recall.
This paper mainly focusses on the first step involving the study of the street-cross sections. Next steps will be described in upcoming articles.

4.2 Point cloud orientation

As mentioned in section 4.1.1, an open-source national streets database is used to locate the cross sections perpendicularly to the road. For that purpose, the point cloud is oriented such as the Y axis corresponds to the direction of the road. The cross section corresponds therefor to the (XOZ) plan as shown in Figure 2.

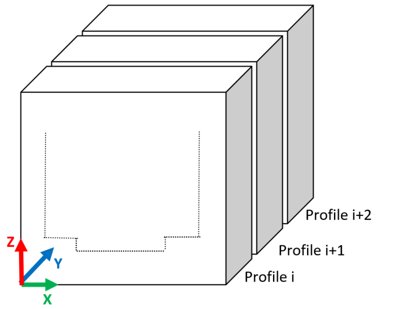

Figure 2: Reference system used for orienting the point cloud and consequently the cross sections

4.3 Generation of regularly spaced cross-sections

Considering the street-oriented point clouds, a profile is a projection on the (XOZ) plane of the points having a Y coordinate according to a defined interval. The optimal thickness of the point cloud is a compromise between the following statement:

- A thinner interval reduces objects' width, especially if they're not aligned with the road direction or presenting a curved profile, like curved curbs, making the identification and vectorization step easier. An example of shape distortion is the planter in Figure 3.
- A larger interval implies that more points are available to describe each object and make the vectorization easier.
- Larger intervals also allow to reduce the impact of density variations.

Regarding the dataset presented in Figure 0, a 5 cm profile thickness has been defined for the profile. A cross-section is considered every 5 cm along the road direction. This high spatial sampling frequency allows to maximize the number of

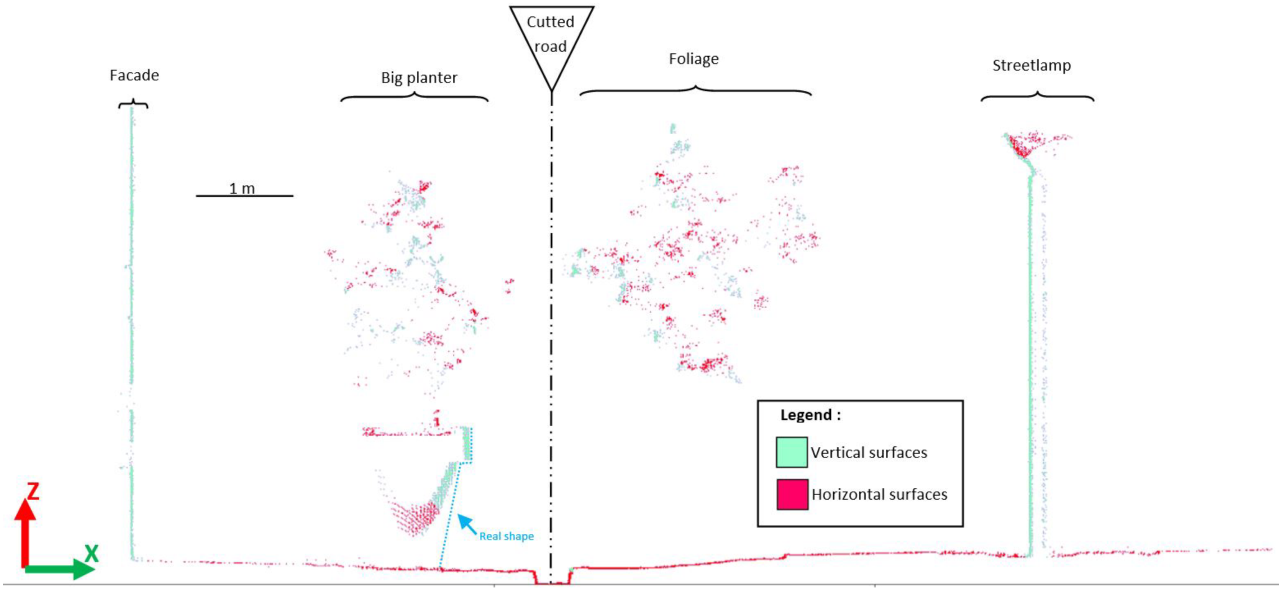

Figure 3: Classification of the points belonging to a cross-section into vertical (green) and horizontal (red) surfaces.

evaluations of the algorithm on the available dataset.
The pixel size has been fixed to 1cm to limit the smoothing of the point cloud. Choosing a small pixel size also allow to enhance the precision of the final outcomes, considering the pixelwise treatment and estimations.

4.4 Segmentation of vertical and horizontal image features

As illustrated in Figure 3, a cross-section is an almost empty image. To avoid misclassifications, several steps have been considered to reduce the noise and acquisition artifacts. For this purpose, different features have been extracted such as point density, point intensity and covariance features, 2D linearity $\frac{\lambda_2-\lambda_1}{\lambda_1}$ and orientation (angle) of the first eigenvector. We also considered the two dominant orientations of a neighborhood. Using those features and considering the labelled reference data, criteria have been searched to realize a first raw selection of the desired points. These attempts were unsuccessful for multiple reasons. The principal one is due to the density variations that complicates the extraction of a satisfying 2D neighborhood. The size of the neighborhood is also critical. It should be large enough to include a sufficient number of points but not too large, to remain sensitive to small structures or geometries.

After investigation, it can be observed that the orientation of the neighborhood, called θ, is considered as the most robust and meaningful feature on the profile for our purpose. θ is used to distinguish points belonging to vertical or horizontal surfaces and to adapt the further process. After thresholding using the natural 45° threshold, a gaussian mixture model is then applied to refine and smooth the segmentation of the pixels. The result of this segmentation is color-coded in Figure 3.

4.4.1 Raw vectorization of the CC: After the previous step, pixels on the image belong to one of 2 classes, vertical or horizontal objects. Points of each class are subdivided into local clusters using the DBSCAN algorithm (Martin et al., 1996). For the clarity of the description, we describe the process applied to the points describing horizontal surfaces. The process is adapted for the vertical elements by performing an inversion of the axis.
In order to improve the clustering process to finely distinguish horizontal clusters with different elevations, biased coordinates are introduced. The Z coordinates are multiplied by 4 for the horizontal points.
Then each cluster is skeletonized and vectorized. For each horizontal cluster, a sliding window along the X axis determine the median value of the Z coordinate of the clusters and leads to the creation of a raw polyline. This technique has several advantages: it is robust, simple and does not require any prior knowledge at this time.

4.4.2 Ground/non-ground segmentation: Considering the set of horizontal raw polylines obtained in previous step, a segmentation into ground/non ground classes is proposed using following considerations:

- Ground can be approximated as a 1-dimensional vector with a length equal to the profile width (along X axis).
- The set of horizontal polylines is sorted depending on their elevation. The first element, so the one with the smaller elevation among the whole set is added by default. It fills the corresponding X extent of the 1-dimensional vector describing the ground altitude.
  The following polylines complete the ground profile one by one, if they do not overlap with the existing parts. In other words, we assume that the ground can be simplified as the combination of the lower polylines for each X position of the profile.

In most of the cases in an urban context, this simplification is verified.
The obtained elevation isn't continuous because the ground is described by polyline portions at this stage. An interpolation is made between available parts. Even if this operation introduces approximations, dealing with continuous data is necessary for the following steps, because a derivation of the ground profile must be carried out to detect the areas of interest.

4.4.3 Extraction of areas of interest: This subsection presents the selection of different areas of interest that may contain the searched objects.

1. By intersection

Having an estimation of the ground profile and a set of vertical objects, intersections are searched. Considering a buffer of 10 cm from the ground profile each intersection with vertical polylines is analyzed. If the vertical polyline has an elevation range less than 25 cm, it's supposed to be a curb, if not, it is considered as unclassified (three, façade, pole…).

2. By derivation

To complete the search of areas of interest, the derivation of the ground profile is calculated and then thresholded. The defined threshold allows to select significant elevation shift corresponding to curb jump around 15 cm. Local peaks are then considered as potential locations for curbs.

4.5 Vectorization of curb-like elements

Having a list of position for potential curbs, the vectorization can be started in each of these areas and is performed in 3 steps. If the sigmoid curve fitting is convincing (paragraph 4.5.2), the vectorization is saved.

4.5.1 Sigmoid curve fitting: Inspired by Zhou and Vosselman (2012), the vectorization of curbs is based on a curve fitting process between the points and a sigmoid function parametrized as shown in equation 1.

$$f(x) = Z_{min} + \frac{r}{1+e^{-a(x-b)}} \qquad \text{(Eq. 1)}$$

With

- Z_{min} : minimum altitude of the curb
- r: elevation range of the curb
- b: position of the inflection point
- a: sharpness of the edges
- x: position against the X axis of the profile

This parametrization without prior defined intervals for the parameters, allows a great flexibility. The function is fittable to any straight-line orientation, and of course, to a curb profile regardless of its height or orientation.

For each area of interest, all the points in a 30x30 pixels square are used as input data for the curve fitting (in black on Figure 4). The conditioning of the problem (inverse of an exponential) requires normalizing the input data in the [0;1] interval along both X et Z axis. The final values of the parameters are obtained later, during a “denormalization” step.

The regression is performed using a RANSAC estimator (relative residual threshold of 10%, 300 max trial, square error loss, and the inlier data should represent at least 70 % of the data).

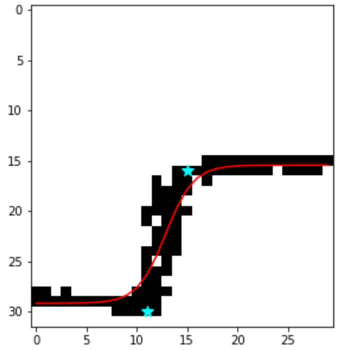

Figure 4: Vectorization of a curb with sigmoid curve fitting.

4.5.2 Filtering of the area of interest: The areas of interest are then filtered using the fitted parameters $Z_{min}, r, a,$ and b.

The proposed filters are the following:

- The inflexion point parametrized by b should be in the neighborhood. In other terms, the elevation jump should be in the defined 30x30 pixels neighborhood.
- The r value corresponding to the elevation shift of the curb should be greater than 6 cm.
- The a parameter should be greater than 9 unless the detected slope isn't sufficient to characterize a curb.
- The a parameter should be lower than 40 unless the curve is “too sharp” to describe a curb.

4.5.3 Vectorization: Each remaining zone of interest is now classified as a curb. Curbs are vectorized with two points as shown in Figure 4.

The X position of these two points correspond to the extremum of the second derivative of the sigmoid function using the optimum parameters. This X position might not correspond to any of the point in the profile, due to occlusions for instance. In order to choose an existing point, a search is started to find the nearest point available. This search is constrained as follows: The research radius increases one pixel at a time, until at least one point is reached. If multiple points are available, the minimum or maximum elevation value is chosen considering the searched point is the bottom or top point of the curb. In this way there is no risk for the algorithm to vectorize missing parts of an object.

4.6 Vectorizing vertical objects

4.6.1 Vertical raw vectorization filtering. If the ground surfaces are solid and opaque, vertical objects may be transparent (glass façade) or reflective (street signs). Generally, their shape is not well defined and trail points can appear. That's why vertical polylines delivered from the raw vectorization (section 4.4.1) are smoothed using a median filter of size 5 to reduce the noise.

4.6.2 Vectorization of vertical objects using ground intersection. Each remaining vertical polyline is vectorized as a segment using a straight-line equation and a RANSAC estimator. Very often, the vertical objects are incomplete on the profiles, due to occlusions. It is also recurrent that the lowest part of facades is missing (cf. Figure 5). To deal with this situation, a point of intersection between the segment and the ground profile is computed. The ground profile is extrapolated by 50 cm on each side to compute a projected intersection in low points areas where the ground profile has not been completed.

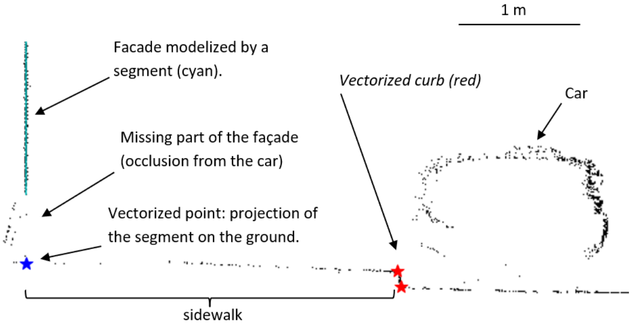
1 m
Facade modelized by a segment (cyan).
Vectorized curb (red)
Car
Missing part of the façade (occlusion from the car)
Vectorized point: projection of the segment on the ground.
sidewalk

Figure 5: Facade vectorization with missing parts.

4.6.3 Filtering: To be considered as a potential vertical road feature, each vertical segment should respect the following conditions:

- The elevation range should be greater than 50 cm.
- The angle between the Z axis and the segment should be less than 15°.
- The distance between the ground profile and the lower part of the segment should be less than 100 cm.

The vectorized point is the intersection point calculated between the ground and the segment. The estimated heigh of the object are especially stored for further treatments.

5. ASSESSMENT

5.1 Reference data

The 1401 profiles have been vectorized manually using “an interactive application" created for this purpose. This program displays the successive profiles and automatically position the cursor according to the previous validated vectorization. The user is invited to correct the position, validate, or cancel the vectorization for the displayed profile. It allows to quickly generate reference data. The manually vectorized road features are the curbs and staircase steps, facades, poles, trees, and wall or fence.

5.2 Evaluation method

The automatic vectorization is compared to the reference vectorization for each profile. The references are associated with the closest automatic vectorization. A maximum search distance of 10 cm is used. Within this tolerance, the automatic vectorization is accepted as a True Positive (TP). If a reference is not associated with any automatic result, it's a False Negative (FN). If an automatic result does not correspond to any reference, it is considered as a False Positive (FP).

In a second step, some further evaluations are performed on the TP. The shift along X and Y as well as the 2D distance between reference and obtained prediction is computed.

6. RESULTS

The image approach facilitates the visualization of intermediate and final results. However, the presented evaluation method is needed to perform a global evaluation and to have quantified indicator.

6.1 Quantitative results

The evaluation method is applied to the 1401 profile, the results are presented in the Table 1 and Table 2 to quantify the quality of the vectorization.

6.1.1 Curb detection and vectorization

The obtained results for this dataset are very satisfying and encouraging. We notice a potential bias between the manual and automatic vectorization for the bottom point. This could be caused by the fact that the automatic algorithm only vectorizes a positive pixel. This constraint is not respected in the reference data.

TP = 1246	FP = 312	FN = 296
Recall = 80.8%	Precision = 80.0%	F-score = 80.4%

Bottom point			Top point		
Δ*X* (cm)	ΔY (cm)	Distance (cm)	Δ*X* (cm)	ΔY (cm)	Distance (cm)
Mean					
-0.3	-0.9	1.9	-0.0	-0.1	1.6
Median					
-0.2	-1.0	1.9	0.0	-0.0	1.6
Standard deviation					
1.4	1.4	1.2	1.4	1.3	1.3

Table 1: Quantitative results for curb vectorization.

The mean and median distance between prediction and reference data are less than 2 cm (2 pixels), corresponding to the order of magnitude of the point cloud noise on a solid surface. This also corresponds to the user vectorization accuracy estimated between 1 and 2 pixels.

6.1.2 Vertical objects detection and vectorization

Quantitative results for vertical object vectorization are reported in Table 2, which presents a high number of FP. This bad score is explained by the lack of filtering as soon as the criteria listed in section 4.6.3 are fulfilled.

In fact, classify vertical object just using one profile in which they appear is a challenging task. Shapes of different objects might look very similar without further context. For example, anti-parking posts and car doors, or façades and electric poles can have respectively the same dimensions. The classification of those objects will be performed in further processing steps using the BEV and "road logic rules".

The statistics of the shifts between manual and automatic vectorization does not reveal any bias.

TP = 1124	FP = 1203	FN = 186
Recall = 85.8%	Precision = 48.3%	F-score = 61.8%

Δ*X* (cm)	ΔY (cm)	Distance (cm)
Mean		
-0.2	0.0	0.5
Median		
0	0	0
Standard deviation		
1.6	0.1	1.5

Table 2: Quantitative results for vertical object vectorization.

6.2 Computing time

The following computation times are given for a laptop with the following specifications: Intel i7-10750H @2.6GHz CPU, 32 Go RAM.

The first operation consists in generating a profile and compute per point neighborhood orientation and save it as a file. It takes 3 seconds per profile. As the vectorization of the profiles are independent processes, they can be parallelized. With 4 processes in parallel, the vectorization of a profile takes 3,2 second in average.

6.3 Qualitative results

Some of the labelled data come from the user's knowledge and user's analysis rather than from a purely geometrical recognition. That's why a 100% score is not feasible. The great majority of the FP for vertical object is associated with cars or vegetation. These false detections will be refined in further steps. Poles and facades are globally well detected. Their vectorization is sometimes imperfect due to noise or acquisition artifacts (trails at the edges, glass surface noise…) (Figure 7).

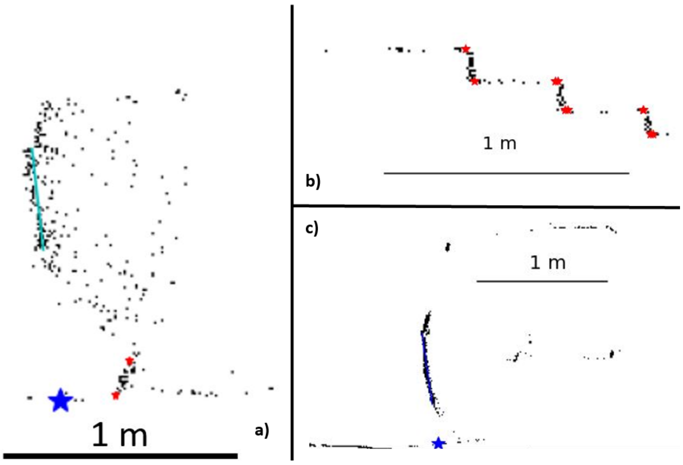

Figure 6: Examples of vectorization results. a) Bad vectorization for a brush generating 2 FP (curb and vertical object); b) Successful vectorization of staircase steps; c) FP generated by a car's door.

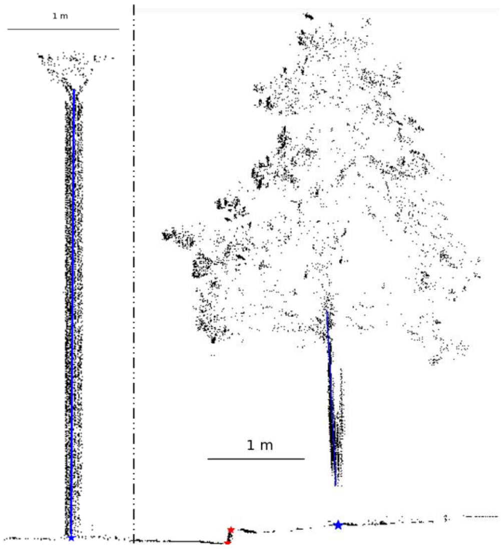

Figure 7: Some vectorization results, a streetlamp (right) and a tree (left).

7. DISCUSSION

At this time, a complete critical review can not be performed because only the first stage of the global approach consisting in the study of street cross-section has been evaluated. Indeed, the processing steps using the BEV and the application of the "road logic rules" are under progress. Nonetheless, some first remarks can be done on this first study.

Whereas the 2D treatment of the point cloud using profile is comfortable in term of data structure, the challenges raised by the point cloud are always present on the images: density variations, noise, occlusions, etc. The extraction of pixelwise features is complicated because contrary to a "normal" image, the profile image is mostly empty.
To deal with these challenges, some interpolations and extrapolations are locally performed. The majority of the criteria and hypothesis made in this study are based on basic geometric considerations (continuity of the ground, orientation, superposition) and on order of magnitude (dimension of the objects, distance between them…).

The sigmoid curve fitting has the advantage to be adjustable to a lot of situations. The optimized parameters allow to discuss the nature of the points. However, in very noisy areas or in partial occlusion situations, the curve fitting results could be incorrect and lead to misclassifications. It could be noticed that the curb detection and vectorization provide very good results on curb and staircase steps (Figure 6b and Figure 7) even if the point density is relatively low. In light of the results of the curb detection, the usage of the sigmoid curve fitting is validated.

As mentioned before, vertical objects' detection is very simple and not very constrained. Any dense vertical surface is detected, leading to a lot of FP. The vectorization of those elements is difficult due to the varying width of the object on the profile. As visible in Figure 7 , the trunk is incomplete and the width inconstant. The obtained vectorization corresponds to the left side of the trunk (the right was not visible from the scanner). On the same figure, the streetlight is vectorized on its axis.

At this time, no prior knowledge has been introduced yet, set aside from the geometry of the curb modelized as a sigmoid function. The oriented profiles could allow to introduce a notion of a typical profile. The proposed approach is flexible and can be enhanced later.
Several iterations of vectorization can also be envisaged in which the results of the previous and further profiles could be used to refine the detection and vectorization of the current profile. In this way, the continuity and linearity of the object can be used as prior knowledge.

8. CONLUSION AND PERSPECTIVES

This paper described a general approach for the inventory and vectorization of road features from point cloud scenes. An algorithm has been developed to detect and vectorize curbs and vertical objects from MLS point cloud in an urban environment. This algorithm only constitutes one stage of the global approach. It consists in the structuration and study of the point cloud scene using vertical cross-sections. The usage of cross-sections oriented perpendicularly to the road direction allows to manage vertical superpositions and to finely deal with objects having a vertical amplitude. The presented results are promising. Some parameters tuning can be done to enhance them. In the future, a set of road rules describing spatial relationships between road features in a knowledge database will allow to improve the overall process performances. As describe in 4.1.2 the BEV image processing will also complete the cross-sections' results.

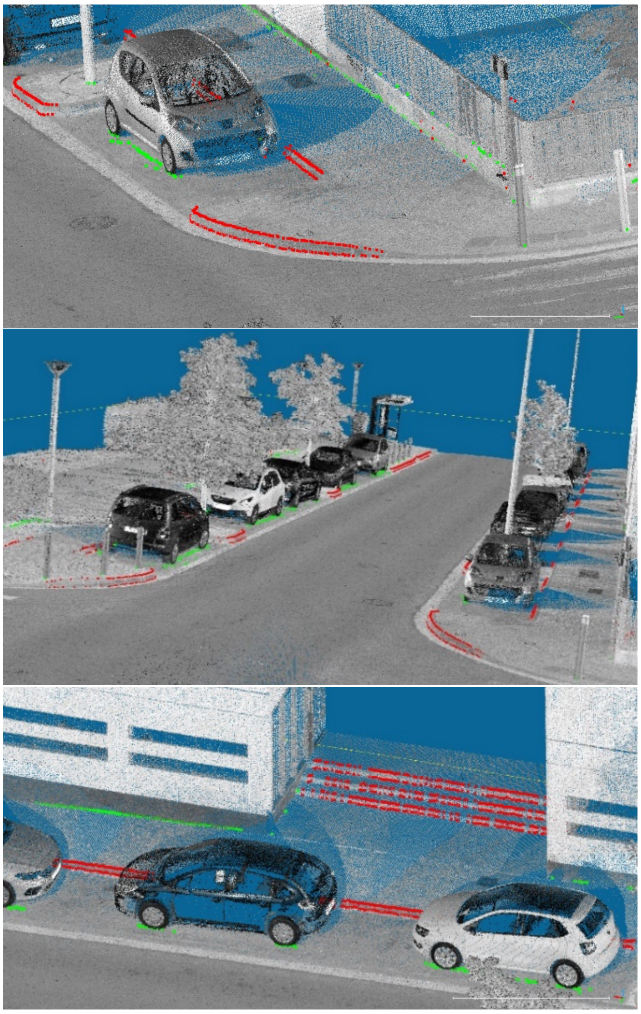

Figure 8: Point cloud obtained by MMS in an urban area. Top and bottom point of curb in red; Vertical object projections on the ground in green.

9. REFERENCES

Alexandru Rosu, R., Schütt, P., Quenzel, J., and Behnke, S. (2020). *LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices*. https://doi.org/10.15607/rss.2020.xvi.006

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Balado, J., González, E., Arias, P., and Castro, D. (2020). Novel Approach to Automatic Traffic Sign Inventory Based on Mobile Mapping System Data and Deep Learning. *Remote Sensing*, *12*(3). https://doi.org/10.3390/rs12030442

Barçon, E., and Picard, A. (2021). Automatic detection and vectorization of linear and point objects in 3D point cloud and panoramic images from mobile mapping system. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B2-2021), 305–312. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-305-2021

Boulch, A., Guerry, J., Le Saux, B., and Audebert, N. (2018). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers and Graphics*, *71*, 189–198. https://doi.org/10.1016/j.cag.2017.11.010

Bradley, D., and Roth, G. (2007). Adaptive Thresholding using the Integral Image. *Journal of Graphics Tools*, *12*(2), 13–21. https://doi.org/10.1080/2151237x.2007.10129236

Gao, J., Chen, Y., Junior, J. M., Wang, C., and Li, J. (2020). Rapid Extraction of Urban Road Guardrails From Mobile LiDAR Point Clouds. *IEEE Transactions on Intelligent Transportation Systems*, 1–6. https://doi.org/10.1109/TITS.2020.3025067

Gézero, L., and Antunes, C. (2019). Automated road curb break lines extraction from mobile LiDAR point clouds. *ISPRS International Journal of Geo-Information*, *8*(11). https://doi.org/10.3390/ijgi8110476

Honma, R., Date, H., and Kanai, S. (2019). MLS point cloud segmentation based on feature points of scanlines. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W13), 1007–1013. https://doi.org/10.5194/isprs-archives-XLII-2-W13-1007-2019

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., and Markham, A. (2021). Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–21. https://doi.org/10.1109/TPAMI.2021.3083288

Justo, A., Soilán, M., Sánchez-Rodríguez, A., and Riveiro, B. (2021). Scan-to-BIM for the infrastructure domain: Generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data. *Automation in Construction*, *127*(April), 103703. https://doi.org/10.1016/j.autcon.2021.103703

Landrieu, L., and Simonovsky, M. (2018). Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4558–4567.

Ma, L., Li, Y., Li, J., Wang, C., Wang, R., and Chapman, M. A. (2018). Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sensing*, *10*(10), 1–33. https://doi.org/10.3390/rs10101531

Martin, E., Hans-Peter, K., Jörg, S., and Xiaowei, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, *96*(34), 226–231.

Meng, H.-Y., Gao, L., Lai, Y.-K., and Manocha, Di. (2019). VV-Net: Voxel VAE Net With Group Convolutions for Point Cloud Segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, *2019-Octob*, 8499–8507. https://doi.org/10.1109/ICCV.2019.00859

Mi, X., Yang, B., Dong, Z., Chen, C., and Gu, J. (2021). Automated 3D Road Boundary Extraction and Vectorization Using MLS Point Clouds. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. https://doi.org/10.1109/TITS.**2021**.3052882

Puang, E. Y., Zhang, H., Zhu, H., and Jing, W. (2022). *Hierarchical Point Cloud Encoding and Decoding with Lightweight Self-Attention based Model*. 1–8. https://doi.org/10.1109/LRA.2022.3149569

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 77–85. https://doi.org/10.1109/CVPR.2017.16

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, *2017-Decem*, 5100–5109.

Rodríguez-Cuenca, B., García-Cortés, S., Ordóñez, C., and Alonso, M. C. (2015). An approach to detect and delineate street curbs from MLS 3D point cloud data. *Automation in Construction*, *51*(C), 103–112. https://doi.org/10.1016/j.autcon.2014.12.009

Safaie, A. H., Rastiveis, H., Shams, A., Sarasua, W. A., and Li, J. (2021). Automated street tree inventory using mobile LiDAR point clouds based on Hough transform and active contours. *ISPRS Journal of Photogrammetry and Remote Sensing*, *174*(February), 19–34. https://doi.org/10.1016/j.isprsjprs.2021.01.026

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, *2019-Octob*, 6410–6419. https://doi.org/10.1109/ICCV.2019.00651

Umehara, Y., Tsukada, Y., Nakamura, K., Tanaka, S., and Nakahata, K. (2021). Research on Identification of Road Features from Point Cloud Data Using Deep Learning. *International Journal of Automation Technology*, *15*(3), 274–289. https://doi.org/10.20965/ijat.2021.p0274

Yao, L., Qin, C., Chen, Q., and Wu, H. (2021). Automatic road marking extraction and vectorization from vehicle-borne laser scanning data. *Remote Sensing*, *13*(13). https://doi.org/10.3390/rs13132612

Zhou, L., and Vosselman, G. (2012). Mapping curbstones in airborne and mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation*, *18*(1), 293–304. https://doi.org/10.1016/j.jag.2012.01.024

Zhou, Y., and Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4490--4499.