

AUTOMATIC DETECTION AND VECTORIZATION OF LINEAR AND POINT OBJECTS IN 3D POINT CLOUD AND PANORAMIC IMAGES FROM MOBILE MAPPING SYSTEM

E. BARÇON, A. PICARD

Innovation department, TT Géomètres-Experts, 75011 Paris, France – (e.barcon, a.picard)@ttge.fr

Commission II, WG II/4

KEY WORDS: Automatization, vectorization, point clouds, vectorization, segmentation

ABSTRACT:

Surveys of roadways with Mobile Laser Scanning (MLS) are nowadays the faster and more secured way to collect topographic data compared with conventional techniques. To deliver topographic plans, the voluminous data collected by the MLS device need to be processed. If the acquisition step is quite fast, the second part of interpretation and vectorization of the LiDAR data and the panoramic images is laborious and time consuming. This paper proposes two approaches that have been developed in order to reduce the time required to process roadway MLS data. The first one is about automatic detection of pole like objects, and the second one is about the detection of linear objects. The presented workflow try to automatically extract a 3D position for each object from MLS Data.

1. INTRODUCTION

Mobile Laser Scanning (MLS) is a very popular technique to carry out extended topographic survey. In addition to the high acquisition speed, they are used to accomplish secure surveys on urban or rural roads and highways. Most of the time, point clouds are not the final goal of the survey mission but constitute intermediate data. The objective is to produce, topographic maps, 3D or BIM models, textured meshes, and other deliverables. Though the acquisition step is quite fast compared to conventional techniques, raw data are time-consuming and costly to process. The interpretation, digitalization (or vectorization) of LiDAR data remains a hand performed step in most of cases. In a road context, an exhaustive inventory of the punctual objects is often requested (streetlamps, road signs, street furniture...). Linear objects such as curbs, road markings, guardrail, walls or building facades, need to be synthesized into 3D polylines to be appearing on a map. These tasks currently have to be performed by an operator from a point clouds and the corresponding panoramic images. This paper proposes two approaches that have been developed in order to reduce the time required to process roadway MLS data.

The goal is to obtain automatic 3D vectorization of road scenes (urban or highway environment). It means that all the interest's objects should be identified and precisely located in the data (images & point clouds). The precision should be better or equal to a manual vectorization. A high-level representation of the elements must then be calculated. Each object is assigned to an object class and a 3D geometry representation (point or polyline). This 3D representation allow to use them in a CAD software and represent them in topographic map. This paper focuses on streetlamps in an urban environment (Section 4), road markings and guardrail in a highway environment (Section 5).

2. RELATED WORKS

The point clouds vectorization embraces different fields: pattern recognition, classification, segmentation, computer vision,

point clouds processing... In this section, a brief review of the existing techniques is presented.

Most of the time, a road surface extraction is a preliminary step for the extraction of other elements (road marking, road manhole, poles, trees...). Because of the influence of this step on the further treatments, the method should be efficient and adapted to the data. The employed technique also depends on the data structuration. Different techniques have been proposed. The planar geometry of the road is frequently used. (Smadja et al., 2010, Wu et al., 2017, Jung et al., 2019) implement Random Sample Consensus algorithm (RANSAC) to perform the road extraction. A scan-line structure, based on the GNSS timestamp or scanning angle field, is used by (Yu et al., 2015, Yao et al., 2018) to realize the segmentation and the following extractions steps. The altitude along the scan-line is analyzed to identify the road's points. If most of the techniques, directly extracts pavement, (Kumar et al., 2017) proposed a raster method to extract the edges and then segment the point clouds even in a rural environment without curb. After achieving this preliminary step, further and thinner treatments can be performed to detect the searched objects.

2.1 Road Marking Extraction

The road markings are extracted using the high radiometric contrast of the markings with the asphalt surrounding. Most of the studies that propose road marking extraction method use a 2D approach because it reduces the complexity of the data and permit to use image processing tools. Different thresholding technique have been proposed to extract the marking from the image (or scanline). Because the laser pulse intensity decrease with the increase of the scanning range and the incident angle between the scanner and the scanned objects, (Jaakkola et al., 2008, Kumar et al., 2014, Yu et al., 2015, Soilán et al., 2017) adapt their threshold using these information. Others adaptive thresholding methods have been proposed by different authors (Cheng et al., 2017, Yao et al., 2018) to deal with the inhomogeneous intensity observed in the point clouds.

To realize a classification and a vectorization of the marking road, different techniques have been proposed. (Jung et al., 2019) describes geometrical criteria to achieve line association and reduce over-segmentation. It allows to reconstruct the traffic lane. (Yao et al., 2018) performed a skeletonization followed by a template matching method to identify specific markings like arrows, text or other specific road markings.

2.2 Streetlamp detection

Different approaches have been proposed to segment streetlamps. The knowledge-based techniques try to match the point cloud with known model of the objects. (Lehtomäki et al., 2010) proposed a method based on scan-line analysis and cylinder fitting. After Applying an Euclidian clustering method to the non-ground points, (Yu et al., 2015) realize a shape matching with prototype objects to isolate streetlamp.

(Li, Elberink and Vosselman, 2018) proposed a feature-based technique. The method is founded on the analysis of horizontal slices of the non-ground points. Different geometrics rules allow to distinguish streetlamps from three and other above ground components.

Other methods can be regrouped into the Deep Learning techniques based on labeled data. (Wu et al., 2017) use the point clouds and the 2D images to realize the segmentation. The method is divided into three steps: raw localization map generation, "ball falling" and position of detection.

2.3 Systematic segmentation

The systematic segmentation of the point clouds is another way to extract the searched objects. They do not focus on one specific object type but aim to classify all the points. These techniques are based on machine learning or neural networks.

(Tchapmi et al., 2018) proposed a voxel approach to deal with LiDAR data. Following a similar idea (Riegler et al., 2017) proposed an octree-voxel structure with an adaptative voxel size to handle more details. The segmentation is then achieved by a 3D Fully Convolutional Neural Network (3D FCNN).

(Landrieu and Simonovsky, 2018) proposed a graph-based approach. Points belonging to geometrically homogeneous elements like plans are gathered to constitute the nodes of a graph called "SuperPoint". This approach allows to have a compact representation of data without simplifying the relationship between objects parts. The graph is then given as an input to a specific neural network.

The famous neural network Point-Net introduced by (Qi et al., 2017) is capable of dealing with raw point clouds directly as input. This work is very popular because of its ability to extract a brief description of the scanned objects ("critical points") and is therefore reused in lots of other works.

We can also mention the view-based methods (Badrinarayanan et al., 2017) that benefit from trained CNN on point cloud's screenshot (Boulch et al., 2018). The semantic informations in the image are then reprojected on the points. This is performed knowing the position and orientation of the images.

3. GENERAL INFORMATION

3.1 Data Source

The point clouds presented in this paper have been acquired with a Riegl VMX450 placed on the roof of a vehicle. Panoramic images were acquired by a FLIR LadyBug LB5+ associated with the IMU and DMI of the Riegl device.

3.2 Data Storage

Because point clouds are massive and unstructured data, a database storage has been chosen to perform the different treatments. First, the raw point clouds in "las" format is decomposed into smaller groups of 400 point named "patches". Those clusters are then spatially indexed and put in a postGIS postgresSQL database. This process allows to access quickly to specific and localized areas without dealing with heavy files. Obviously, the database setup has a computational cost, but this can be achieved without human intervention.

3.3 Data Storage

3.4 Computer specifications

The technical specifications of the used hardware and software are given below:

Processor	Intel Xeon CPU E5-1620v4 3050 Ghz
RAM	16.0 Go
GPU	NVIDIA Quadro M2000, 4 Mo RAM
Python	V3.5.2
CUDA	V9.0
CuDNN	V7.1
PostGIS	Bundle PostGis 10 v2.5.1-1
Tensorflow	GPU v1.9.0

Table 1. Computer specifications.

4. AUTOMATIC DETECTION OF POLE LIKE OBJECTS IN URBAN POINT CLOUD

4.1 Objectives and approach

The goal of this first part is to propose an automatic method that aim to segment pole like objects in point clouds and then compute their corresponding 3D insertion point. After a study of the state of the art, a hybrid method using both point clouds and panoramic images has been chosen. Figure 1 resume the main points of the proposed method.

4.2 Pre-processing

The pre-processing phase is made of two different and independent workflows. On the first hand, the panoramic images are decomposed into cubemaps to discard spherical distortions. Instance segmentation is then performed on these distortion-free images. A pre-trained model of Mask R-CNN proposed by (He et al., 2017) is used (Figure 2). A specific training with 345 manually labelled images has been achieved to enhance performance on the searched objects. On the other hand, point clouds are ground/non-ground segmented using the method proposed by (Zhang et al., 2003). This segmentation based on mathematical morphology allows to eliminate all the ground points that are irrelevant as they do not contain the searched objects.

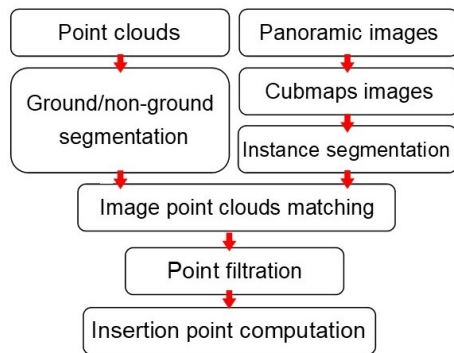


Figure 1. Streetlamp's detection and vectorization workflow.

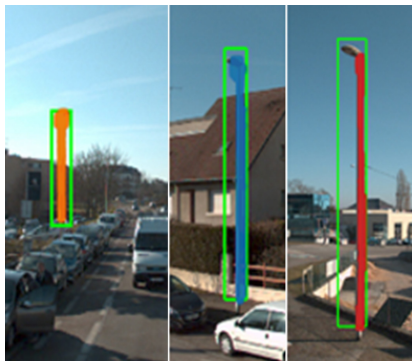


Figure 2. Examples of the instance segmentation results.

4.3 Images and point clouds matching

The segmented mask and the bounding boxes obtained are then used to identify the corresponding points on the LiDAR data. The masks are reprojected into the spherical system of the panoramic images. Knowing the position and orientation of each image, non-ground points are projected back on the panoramic images. It can be seen as a coordinate conversion, from cartesian (X, Y, Z) to spherical coordinates (λ , ϕ , ρ). The origin of the spherical coordinate is given for each image by the principal point of the spherical camera. By doing so, pixel coordinates can be associated to each point. Multiple points can have the same pixel coordinates, but points of interest are those at the foreground in relation with the shooting position. A Hidden Point Removal (HPR) filtration proposed by (Katz and Tal, 2015) is then performed. This technique is more robust than a simple filtration based on the distance between the shooting position and the points and can deal with noisy points. The drawback of this technique is that it eliminates points describing the back of the object. Since the same object is visible from different images, the multiple selection enables to finally obtain an almost complete segmentation of the desired objects.

The selected points are those among the remaining points which intersect the semantic masks. This selection may contain others object than road signs and streetlamps. These unwanted elements are describing nearby items, vegetation, electric box, noise... They need to be removed because they may disturb the inserting point calculation.

4.4 Point filtration

To realize a filtration of the selected points, a DBSCAN (Ester et al., 1996) clustering method is used. This technique based

on point density allows to reduce noise and to regroup objects into clusters. These clusters are then filtered by the following assignment:

- The distance between the point of view and the cluster's center must be less than 100 m. Indeed, at this distance objects are very small on the pictures.
- A score is then calculated for each remaining cluster, thanks to formula (1) given below:

$$score = score_{MaskR-CNN} + \frac{1}{d_{2D}} + nb_{points} \quad (1)$$

Where $score_{MaskR-CNN}$ is the confidence score obtained by the semantic mask CNN.

d_{2D} is the planar distance between the point of view and the cluster centroid,

nb_{points} is the cluster's number of points.

The remaining points are finally divided into groups by a coalescence algorithm. Some results are given in Figure 3.

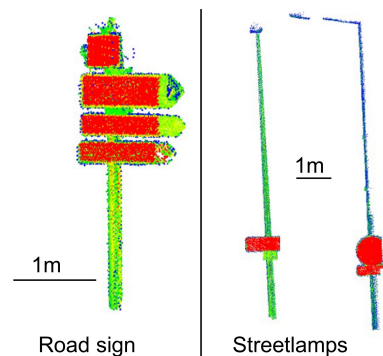


Figure 3. Example of the point clouds segmentation.

4.5 Computation of the insertion point

The simplest way to compute an insertion point is to calculate the XY barycenter of the group and then to get the ground corresponding altitude. But despite the efforts to reduce the outlier's points, their presence can lead to errors. This is the reason why this technique cannot be considered as robust enough. To go further, we have to use the shape of the searched object, their pole. Most of the time, the pole's profile is cylindrical. The horizontal section of a pole is a circle provided that the pole is vertical, an ellipse otherwise. To deal with non-vertical pole, a Principal Component Analysis (PCA) is achieved to determine the main axis of the pole. Using that direction, the points are sliced into different cylinders. For each one, the points are fitted to a circle with the RANSAC method (Fischler and Bolles, 1981). The different computed centers are finally averaged to obtain a final XY position in the original coordinate system as shown on Figure 4.

4.6 Test data

The test data consists in 435 panoramic images and the corresponding point clouds with the characteristics presented in table 2. The data correspond to an urban residential area.

Number of points	498 349 258
Mean density	1 628 pts/m ²
XY surface	0.29 km ²
X extent	1,02 km
Y extent	1,708 km
Z extent	157 m

Table 2. Point clouds specifications.

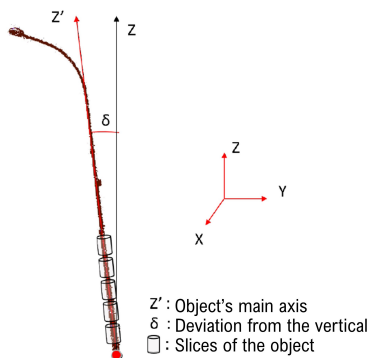


Figure 4. Inserting point calculation.

4.7 Results

An instance segmentation evaluation was performed with ground truth images generated manually on the dataset. As there were too few road signs in the dataset, the evaluation only concerns streetlamps. The results are given below in table 3. The bounding-box IoU with a 63% score allows to say that the detector operates correctly for most of the objects. A qualitative study explains the gap with the mask IoU score: the semantic masks are always smaller than the objects. The others values seem to indicate that the detector is balanced, and not in favor of certain situations. The average score shows that the prediction model can be improved.

	IoU (bbox)	IoU (mask)	Precision	Recall	F-Score
Mean	0.60	0.23	0.53	0.56	0.54
Min	0.07	0.00	0.00	0.00	0.00
Max	0.86	0.62	0.91	0.95	0.91
Median	0.63	0.2	0.56	0.64	0.61

Table 3. Instance segmentation results.

Then an evaluation of the computed insertion points with ground truth data obtained by a manual vectorization has been achieved. The fact that the same object is present on several images allows to obtain a better recall score than the prediction model. The precision goal is practically achieved. The elevation estimation is still perfectible because the technique used is not as robust as the XY determination. Indeed, the ground elevation estimation could be difficult.

The total time of treatment is around 13h, 2h for the point cloud segmentation, 8h for the instance segmentation and finally 2,5h for the insertion points.

Recall	σx	σy	σz	emqX	emqY	emqZ
71 %	3,0	2,7	9,1	0,8	0,2	5,0
	cm	cm	cm	cm	cm	cm

Table 4. Inserting point precision results.

4.8 Conclusion and prospects

The results obtained using our detection workflow are promising. Despite the small training data for the instance segmentation, the results reach a recall of 70 % of the detected candelabras, with an accuracy of less than 5 cm in planimetry and 10 cm in altimetry. However, the proposed engine can be enhanced with more training data. It can also be generalized to other objects.

5. AUTOMATIC DETECTION OF LINEAR OBJECTS IN HIGHWAY POINT CLOUD

The goal of this second part is to detect and vectorize pavement markings and guardrail in highway point clouds. The purpose of this study is to obtain 3D polylines that describe the position of these elements. The needed precision had to be better or equal than a manual vectorization. After studying the state of the art, the image approach has been chosen. A highway is considered as a simple environment because vertical superpositions are occasional. Converting the point cloud into an enhanced elevation raster does not compromise the data. The Figure 5 present the main points of the proposed method.

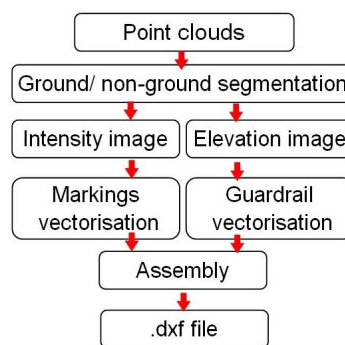


Figure 5. Main points of the proposed guardrail and marking road extraction and vectorization.

5.1 Pre-processing

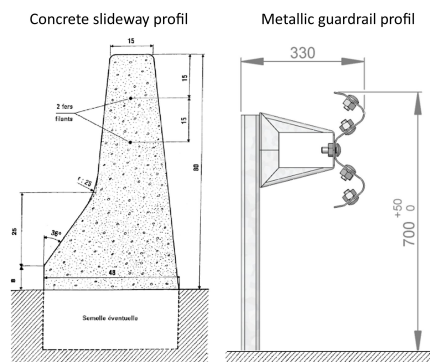


Figure 6. Profiles of the most encountered guardrail in France.

The image approach requires a discretization of the point cloud into contiguous images. Due to the complexity of this step considering the numerous specific cases, the discretization is performed manually. An AutoCAD's function has been created to perform this step. The user visualizes the point cloud in AutoCAD in top-view mode and has to draw the outlines of the future images. Thus, the user is able to optimize the position of the images with respect to the point clouds and can target the area of interest by reducing them to the road only. An optimal image length of 100 meters has been chosen after computational tests realized with the computer used in this study (Cf. 3.3). We assume that the results would have been different with another configuration.

5.2 From point clouds to images

The outlines drawn by the user are used to perform queries in the point cloud database. The recovered points are ground/non-ground segmented with the Cloth Simulation Filter (CSF) algorithm (Zhang et al., 2016). More precisely, the CSF is performed twice. The first implementation is used to remove all points presenting a height above 1.5 m. These points are irrelevant because they cannot correspond to interest objects. The second implementation is used to truly perform a ground/non-ground segmentation of the remaining points. Then, the road direction is determined using a Principal Component Analysis (PCA). The point cloud is rotated in order to align the road direction with the width of the future image. This rotation allows to optimize the image size so to reduce computation cost. The points belonging to the ground class are rasterized into an intensity and elevation image with a pixel size of 2 cm. This intensity image will be used to detect road markings. The remaining points belonging to the non-ground class, are rasterized into an elevation image with a pixel size of 1 cm. This elevation raster contains two channels, corresponding to the minimum and maximum point altitude available for each (2D) pixel location. This elevation image is used to detect guardrail. The guardrails are non-planar objects, a smaller pixel size is used to avoid a shape smoothing of the objects.

5.3 Road marking detection

As mentioned earlier, the road marking detection lies on the intensity image processing. Because some areas contain any point, the raw intensity images contain no-data pixels. A first pre-processing step consists in carrying out an interpolation. The nearest neighbor technique is used. A maximal search radius is fixed at 2 pixels (4 cm) to avoid extrapolation and to limit calculation time. A default value is attributed to the no-data remaining pixels. This value corresponds to the most represented value in the histogram of the image. Assuming that this value always matches with a mean road color, those pixels are then indirectly classified as road-pixels.

After these pre-processing steps, all the pixels have a value. The road marking extraction is performed by an adaptive thresholding method presented by (Bradley and Roth, 2007). This thresholding method is especially able to deal with road color or light exposure changes. The obtained binary image is then filtered to reduce false positives with geometrics criteria. Road markings, as defined by traffic regulation, can have different width and length that lead to different meanings. The filtering criteria must take this into account and be generic. The Connected Component (CC) filters used are listed below:

- The minimum area is fixed to 0,15 m²

- The length over width ratio must be greater than 4.
- The CC verify the inequity:

$$\frac{P}{2(w+h)} > 0,11 \quad (2)$$

Where P is the perimeter,
 w = width of the CC,
 h = height of the CC,

The CC presenting a length greater than 4 m or a width greater than 0.5 m are difficult to handle because of their extensive shape. These are cut in pieces of 1 m length maximum that are easier to process. This procedure is inspired by the "divide and conquer" method. A long marking can be the conglomeration of several markings or the gathering of a marking with another object. The cutting step allows to treat successfully healthy parts of a noisy marking.

5.4 Road marking vectorization procedure

Two different vectorization methods are used depending on the width of the filtered connected component. In most cases, CC's width is lower than 0,5 cm so they are modelled by straight lines. Each pixel of a CC is considered as an observation with x and y coordinates. The modelling is performed by least square method if the outlines are regular otherwise by RANSAC method. The use of the robust RANSAC method allows to detect the outliers and to enhance the vectorization result. The regularity criterion used is based on the standard deviation of the CC's width along its major axis.

In other cases, when the connected component's width is greater than 0,5 m, the vectorization is different. In this case, the outlines of the CC are modelled. It enables to perform a successful vectorization of the markings that separate the main track from the insertion and deceleration lane for example (Figure 7). Individuals marking lines filtered are then linked together in order to rebuild traffic lanes. It is performed by an assembling program based on geometric criteria. These criteria, inspired by (Jung et al., 2019) are based on the relative orientation and position between two individual marking lines. The thresholds are established to limit assembly possibilities. A cost function is used to select the best association between the remaining possibilities. As final step, some geometrics verifications are performed. If two lines overlap, a procedure is launched to fix that issue.

5.5 Detection of the guardrails

As mentioned before, the guardrail detection is based on the elevation image obtained after the vertical projection of the non-ground points. The first objective is to reduce the research area on the images. The searched objects (Figure 6) present a vertical amplitude. The first step is a 2D rough vectorization (in XY plan) of the element having a vertical amplitude. This is achieved using the fact that points describing vertical surfaces create high density areas after a vertical projection on a plan, as illustrated in Figure 8. Vertical surfaces are simply detected by thresholding the point density on the image. The obtained connected components after the thresholding are then vectorized, with the same technique used for the road markings. The obtained lines contain false-positive that mostly correspond to vegetation and vehicles. To deal with possible occlusions and density differences, a specific algorithm tries to lengthen the

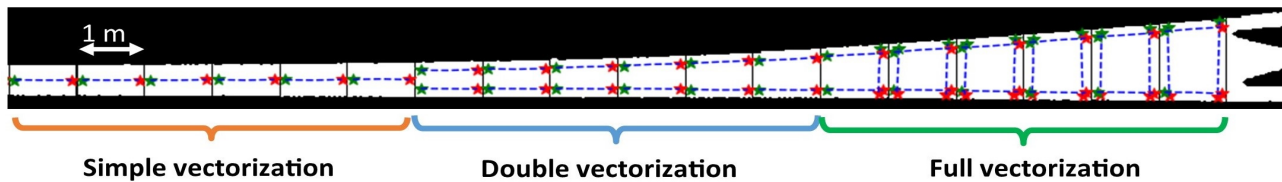


Figure 7. Different vectorization of the detected markings road.

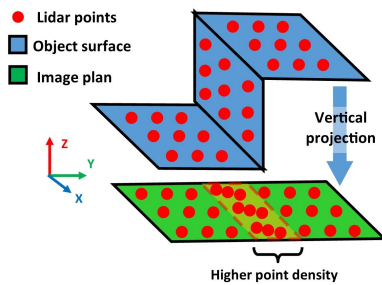


Figure 8. Vertical surfaces create high point density after a vertical projection.

individual straight lines previously obtained. This sequential algorithm, based on the growing surface algorithm, is working step by step. It identifies the direction of the line and detect points that might be continuing the considered line. If the required conditions are met, it lengthens the current line.

After the obtention of those rough 2D polylines, a fine study of the neighborhood of those lines is completed. Profiles are regularly extracted along the lines. They are used to identify the nature of the objects and to vectorize them. The extraction of the profiles is performed in a neighborhood of 10 cm along the line and about 30 cm on both sides in order to get enough data to apprehend successfully the shape of the object (Figure 9).

The nature identification is performed with a template matching method. A profile library is available for the program. For each registered profile, a class and an inserting point is given. The method is performed in two steps. The nature of the object is identified by testing all the known templates and choosing the one who gets the maximum score. The position of the object is then deduced from the correlation image obtained by a normalized cross-correlation between the profile and the selected reference template. The use of the normalized cross-correlation especially permits to determine a position even if the object profile is incomplete or noisy. Due to the regularity of the profile extraction, and after a connection of the different 3D vectorized points, we obtain 3D polylines describing the guard rail. The final phase consists in a connection of the partial results obtained for each image. The final 3D polylines describing the guardrail and the road markings are then exported in DXF format.

5.6 Evaluation

The 3D polylines are compared to 2D polylines manually vectorized and considered as ground truth. The first step is a 2D comparison of the polylines. Each polyline is discretized in infinitesimals lines. For each small segment, the distance to the corresponding reference polyline is computed. Each one is classified according to a precision class as shown in table 5. For each tolerance (T), the recall (R), precision (P), and F-score (FS) can be computed. The presented results have been

Object	T(cm)	35	25	10	5	3
Road Marking	R(%)	97	96	92	84	67
	P(%)	735	51	327	69	94
	FS(%)	94	94	90	82	66
Metallic guardrail	R(%)	83	82	78	69	53
	P(%)	73	72	68	60	46
	FS(%)	77	76	73	64	49
Concrete guardrail	R(%)	92	91	88	74	50
	P(%)	86	86	83	70	48
	FS(%)	89	82	85	72	49

Table 5. Linear object detection and vectorization results.

obtained for a cumulated length of 14 km for the road marking, and 10 km for the guardrail. The used test dataset are composed of different sample of highways in different configuration (various traffic lane number, straight or curved trajectory, tunnel...).

The decreasing of the score with the augmentation of the precision mean that the objects are correctly detected but their vectorization must be enhanced. The recall values are greater than 80% for all the objects at 10 cm precision. Only 66% percent of the road marking and 49% of the guardrail are vectorized with a 3 cm precision.

Different causes have been identified. The transition between a concrete and a metallic guardrail cause program failure because it requires a specific management. We also have to consider that the handmade vectorization cannot be perfect and sometimes simplifies the trajectory. The template matching procedure can be enhanced to avoid false positives.

The 3D evaluation of polylines has been performed qualitatively with a simultaneous visualization of the point clouds and the results (Cf. figures 10 & 11). It shows that the vectorized elevation is successful most of the time. The typical error (around 2 cm) can correspond to the LiDAR acquisition noise. Because the vectorization is performed in a unique stage, elevation errors are highly correlated with 2D errors. This qualitative study shows that our method can vectorize the guardrail even if it has been rugged (Figure 10). This ability causes an underestimation of the quantitative result because the handmade vectorization has smoothed those trajectory variations.

Concerning processing time, the execution speed from a ".las" format point clouds to a dxf file represents between 0,5 and 1 km/h. The pre-processing step, which is the more time consuming, has been optimized with a parallelization. The other steps have not been computationally optimized yet.

5.7 prospects and improvements

The described workflow presents interesting results but needs improvements. If the vectorization process is efficient in most

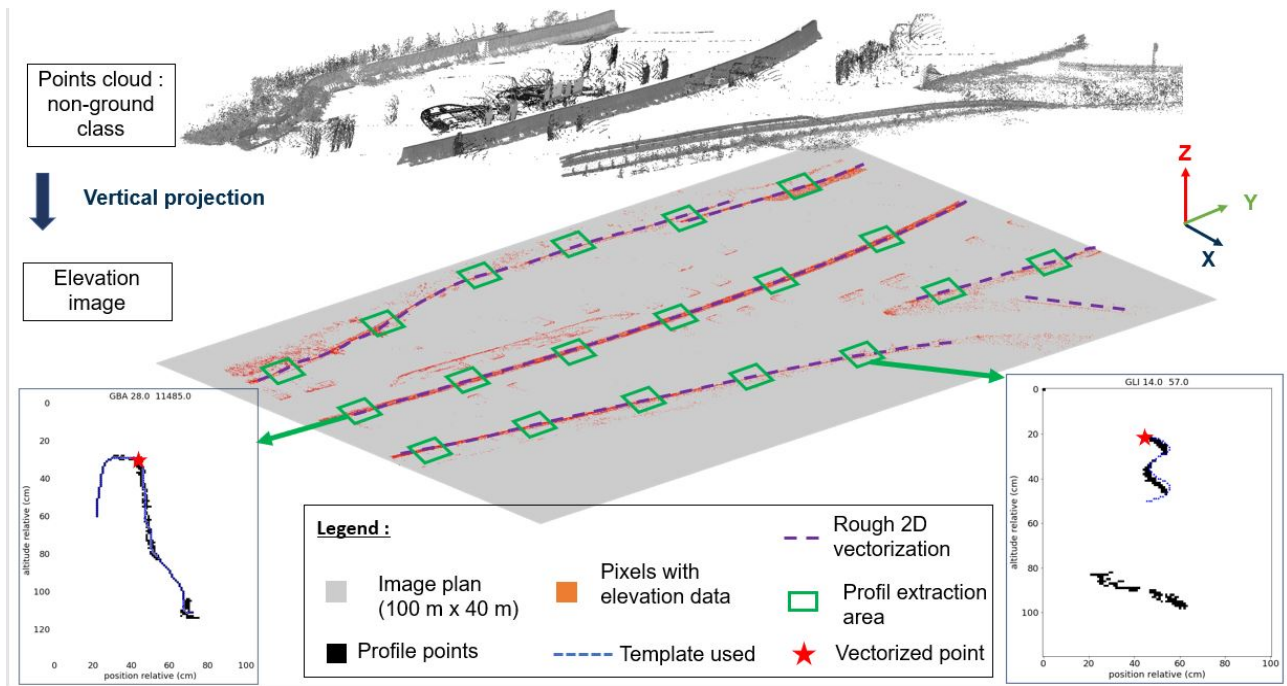


Figure 9. Guardrail extraction process.

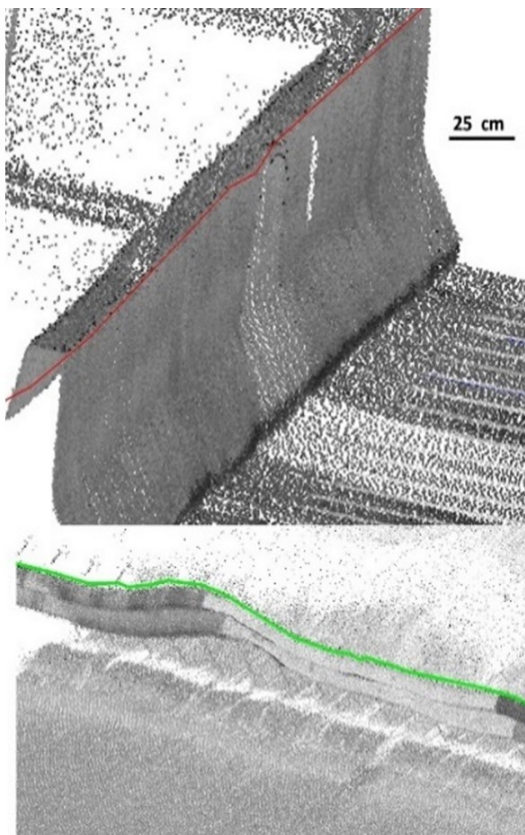


Figure 10. Results of the guardrail vectorization.

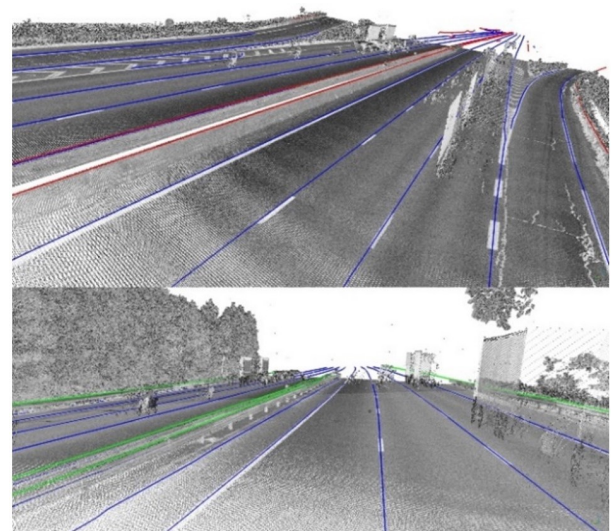


Figure 11. Results of the road marking vectorization in blue and guardrail in green or red.

of the situations, some specific cases disturb the correct execution of the program. More robustness is needed to reduce manual rework. The vectorization process can be enhanced for all the objects. The object detection reaches very promising scores, but the vectorization part needs to be enhanced. The program can also be generalized to other items like walls and curbs by adding them to the known profiles.

6. CONCLUSION

These two studies show that achieving point clouds processing automatization is possible but needs more research and development to be truly efficient. The proposed method for the automatic vectorization of the streetlamps, marking road and the guardrail presents encouraging results. If our results are not as good as those of the state of the art, this difference can partly be explained by the different objectives and evaluation methods used. Obtaining a high level representation (3D vectorization) of the searched objects is difficult, the proposed algorithm needed more robustness. The workflow presented in this paper can be optimized and also generalized to other objects. The second study that concentrate on markings and guardrail points out that deep learning method are not the only way to deal with these complex issues. To go further, a combination of the two detection engines is being considered. The addition of a knowledge base of the objects and their relationship could enhance the detection sensibility and leads to better results.

REFERENCES

- Badrinarayanan, V., Kendall, A., 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.
- Boulch, A., Guerry, J., Le Saux, B., Audebert, N., 2018. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers and Graphics (Pergamon)*, 71, 189–198.
- Bradley, D., Roth, G., 2007. Adaptive Thresholding using the Integral Image. *Journal of Graphics Tools*, 12(2), 13–21.
- Cheng, M., Zhang, H., Wang, C., Li, J., 2017. Extraction and Classification of Road Markings Using Mobile Laser Scanning Point Clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(3), 1182–1196.
- Ester, M., Kriegl, H.-P., SANDER, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 96(34), 226–231.
- He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 386–397.
- Jaakkola, A., Hyypä, J., Hyypä, H., Kukko, A., 2008. Retrieval algorithms for road surface modelling using laser-based mobile mapping. *Sensors*, 8(9), 5238–5249.
- Jung, J., Che, E., Olsen, M. J., Parrish, C., 2019. Efficient and robust lane marking extraction from mobile lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147(November 2018), 1–18.
- Kumar, P., Lewis, P., McElhinney, C. P., Boguslawski, P., McCarthy, T., 2017. Snake Energy Analysis and Result Validation for a Mobile Laser Scanning Data-Based Automated Road Edge Extraction Algorithm. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2), 763–773.
- Kumar, P., McElhinney, C. P., Lewis, P., McCarthy, T., 2014. Automated road markings extraction from mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation*, 32(1), 125–137.
- Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. 4558–4567.
- Lehtomäki, M., Jaakkola, A., Hyypä, J., Kukko, A., Kaartinen, H., 2010. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sensing*, 2(3), 641–664.
- Qi, C. R., Su, H., Mo, K., 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.
- Riegler, G., Ulusoy, A. O., Geiger, A., 2017. OctNet: Learning deep 3D representations at high resolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 6620–6629.
- Smadja, L., Ninot, J., Gavrilovic, T., 2010. Road Extraction and Environment Interpretation From Lidar Sensors. *IAPRS*, 38(1), 281–286.
- Soilán, M., Riveiro, B., Martínez-Sánchez, J., Arias, P., 2017. Segmentation and classification of road markings using MLS data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 123, 94–103.
- Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2018. SEGCloud: Semantic segmentation of 3D point clouds. *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, 537–547.
- Wu, F., Wen, C., Guo, Y., Wang, J., Yu, Y., Wang, C., Li, J., 2017. Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach. *IEEE Transactions on Intelligent Transportation Systems*, 18(2), 292–305.
- Yao, L., Chen, Q., Qin, C., Wu, H., Zhang, S., 2018. Automatic extraction of road markings from mobile laser-point cloud using intensity data. 42Number 3, International Society for Photogrammetry and Remote Sensing, 2113–2119.
- Yu, Y., Li, J., Guan, H., Jia, F., Wang, C., 2015. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(2), 709–726.
- Zhang, K., Chen, S. C., Whitman, D., Shyu, M. L., Yan, J., Zhang, C., 2003. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4 PART I), 872–882.
- Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, 8(6), 501.