Road Side Detection and Reconstruction Using LIDAR Sensor

Conference Paper · June 2013

DOI: 10.1109/IVS.2013.6629637

2 authors:

Alexandre Hervieu
Institut national de l'information géographique et forestière
15 PUBLICATIONS **170** CITATIONS
SEE PROFILE

Bahman Soheilian
University of Paris-Est
42 PUBLICATIONS **589** CITATIONS
SEE PROFILE

2013 IEEE Intelligent Vehicles Symposium (IV)
June 23-26, 2013, Gold Coast, Australia

Road Side Detection and Reconstruction Using LIDAR Sensor

Alexandre Hervieu and Bahman Soheilian

Abstract— **Road edge localization is key knowledge for automatic road modeling and hence, in the field of autonomous vehicles. In this paper, we investigate the case of road border detection using LIDAR data. The aim is to propose a system recognizing curbs and curb ramps and to reconstruct the missing information in case of occlusion. A prediction/estimation process (inspired by Kalman filter models) has been analyzed. The map of angle deviation to ground normal is considered as a feature set, helping to characterize efficiently curbs while curb ramps and occluded curbs have been handled with the proposed model. Such a method may be used for both road map modeling and driver-assistance systems. A user interface scheme has also been described, providing an effective tool for semi-automatic processing of a large amount of data.**

I. INTRODUCTION

Embedded systems provide rich information on roads and obstacles, allowing for applications in various domains such as driver assistance, autonomous vehicle and automatic road map modeling. To identify drivable areas, automatic schemes require road edge localization. This road side information is also crucial for distinguishing obstacles on the street from those beside the street. Therefore, automatic road edge recognition may be used to model streets. Such models are of prime importance for city maintenance, providing tools for automatic road maintenance and path planning. Indeed, the current paper is part of a research project called Terramobilitawhich aims at identifying practicable trajectories (i.e., accessibility) for any kind of mobility such as the wheelchair, the bicycle or the disabled person's mobility.

In order to understand the road scene, techniques such as LIDAR, GPS or computer vision techniques have been used. At first, the computer vision community investigated the issue of semantic road scene understanding. In the context of computer vision, 3D reconstruction is usually computed using stereoscopic images, to provide important geometrical knowledge of the road scene. Graphical models have been used to deal with road scene modeling. In [1], Floros and Leibe proposed an image-based multi-class segmentation of the scene relying on Conditional Random Field (CRF) modeling. The authors developed a model that considers the temporal continuity of the observed images and the corresponding 3D information in a unified framework. This method belongs to a set of image-based methods that do not deal with obstacles (see also [2], [3], [4]) and do not reconstruct missing parts of the road side. Siegemund et al. [5] also examined CRF modeling to detect 3D curbs and process road on road-adjacent segmentation in images via a Kalman filtering that predicts the current position of a curb from estimated results in the previous step. However interesting, this method is not able to deal with curb ramps since it relies on the elevation between roads and road-adjacent areas. Hence it may lose the curb and create erroneous curbs. Traffic isles are also a limitation to that technique since the method relies on the assumption that the curbs are collateral to the vehicle driving corridor. Other methods were proposed by Oniga et al. [4], [6] for road scene analysis. In [6], a road surface and obstacles detection method is described where 3D information extracted from the stereoscopic images was used to produce digital elevation maps. Furthermore, this map has been processed using a RANSAC algorithm combined with a region growing method that helps recognize roads and obstacles. The method seems to work well, but curb ramps are detected as road surface, whereas our aim is to distinguish curb ramps from the road.

Hence, image processing has been extensively used to process road scene analysis. However, despite the real interest of the proposed methods, it still remains highly uncertain about 3D information extracted from stereoscopic images. Besides, vision systems suffer from unfavorable light or weather conditions, i.e, overcast sky, sun glare, shadows, etc..

Several approaches to automatic detection of road sides and traffic islands from aerial images have also been proposed, relying usually on active contours and snake modeling (for an example of traffic island detection, see [7]). Our aim here is to propose a method that provides results of road edge detection for path planning in urban environments, which require high precision of the 3D aspects of the road. Thus aerial images are not suitable for curbs detection.

Hence, to ensure reliability of the processed information, LIDAR sensors will be examined further.

A method dedicated to autonomous mobile systems that relies on the fusion of both LIDAR and image was proposed by Anderson et al. (see [8]). This work aims at detecting reachable areas (regarding mobile entities, i.e., cars, humans, robots, etc.) and use this information to further retrieve road edges. In the paper, the road is defined as a reachable part of the ground in front of the mobile. LIDAR information has been used to estimate seven different features on the road (such as curvature, slope, etc.). Relying on a road visual homogeneity hypothesis, image information is used to recognize the road and non-road areas, hence providing knowledge on road sides.

Unlike such reachable area detection systems, our aim is to identify road edges while dealing with potential obstacles. Hence, while these methods will consider a curb ramp to be part of the road, we want a method that detects road and non-road in the presence of obstacles and curb ramps.

LIDAR systems have been proposed for road boundary

Alexandre Hervieu and Bahman Soheilian are with the Université Paris-Est, IGN/SR, MATIS, 73 avenue de Paris, 94160 Saint Mandé, France firstname.lastname@ign.fr

detection based on road network models and GPS ([9], [10]). Stückler et al. [9] used detection of curbs for in-lane localization. GPS waypoints were here used as well as road models that provide lane width information to produce a model of expected curbs. The aim of the method is to reduce the offset between the model and the real observed curbs. First, LIDAR information is transformed into height images where curbs are detected using edge filters. Then an Iterative Closest Point (ICP) algorithm used to determine the rotation and translation that minimizes a function error describing the sum of the observed offsets. Similarly, Wijesoma et al. [10] proposed a road edge detection method requiring approximate road width at the successive positions (given by a GPS). Road area detection was here processed by a Kalman filtering method that tracks the two road sides.

However, our aim is a road side detection method that does not require any knowledge about the width of the road.

Finally, LIDAR-only road edge detection method was proposed in [11], [12]. Maye et al. [11] designed a method where ground is processed using a piecewise planar model. First, a DEM was computed on the LIDAR measurements. Then, each cell of the DEM was processed using a Conditional Random Fields algorithm. An Expectation-Maximization algorithm for mixture models (here a planar mixture model) was then used to get the corresponding piecewise planar model. Curbs were finally detected as the boundaries between plane segments.

Most of the existing LIDAR-only road boundary detection methods (except few very recent works [13], [14]) do not handle curb occlusion and curb ramps, whereas our aim is to handle both of these issues to provide efficient road edge detection in the presence of obstacles.

The goal of the approach described below is to detect curbs and curb ramps in urban road scenes. The method will be used later on to produce path planning tools for distinct kind of mobility (i.e., the wheelchair, the roller skates, the bicycle, etc.). Hence, its aim is also to handle occlusions caused by prospective obstacles, i.e., to reconstruct road side positions when no information is available. The process should propagate curb positions inside occlusion areas and be robust enough to deal with small obstacle residuals.

The rest of the paper is structured as follows. Section II presents the proposed approach, describing processed feature map and introducing the road edge detection algorithm. In section III, experiments and results are described.

II. OUR APPROACH

The feature map and some further pre-processings are first presented. Then, we describe the model that was applied on the feature map.

A. Angular distance to ground normal map

LIDAR data described in the previous subsection have been processed to compute an original feature: the angular distance to ground normal (Fig. 1). Let us denote as N the normalized normal to the ground and V_p the normal vector at a given 3D point p, the angular distance θ is defined by :

$$\theta = \arccos(V_p \cdot z).$$

The direction of the normal vector V_p is so that $V_p \cdot z > 0$, i.e., the normal vector is always oriented within the same half-space than z and hence $\theta \in [0, \frac{\Pi}{2}]$.

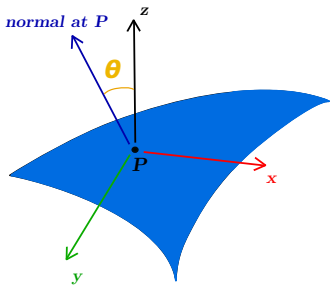

Fig. 1. Angular distance θ to ground normal on a given point p.

Indeed, such a feature describes well the curb 3D pattern that we wish to retrieve (see Fig. 2). It enabled us to distinguish the 3D curb pattern from other ones, such as the ground, facades or stairs of higher heights.

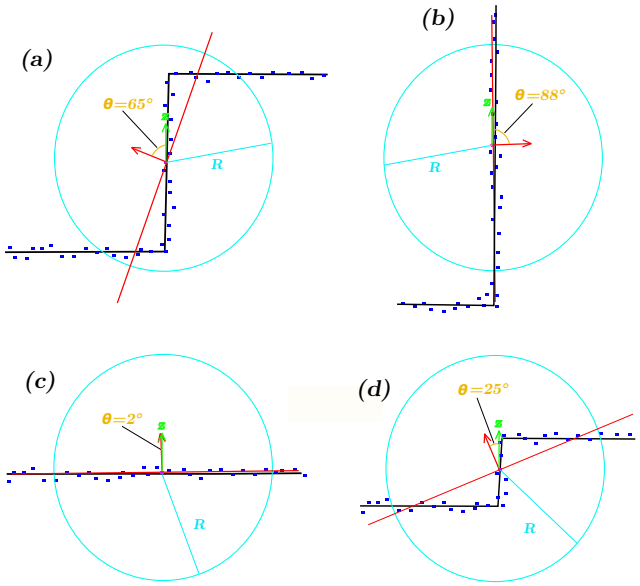

Fig. 2. Four different profile patterns depicted here in 2D for better understanding. (a), (b), (c) and (d) schemes respectively represent a stair, a facade, the ground and a road side (usually lower than a stair). The red arrows correspond to computed normals whereas the green one is the z vector. Blue points are LIDAR points, the red line denotes the estimated plane (using RANSAC) based on the LIDAR points within the blue circle. The yellow arrow is the θ feature, i.e., the angle deviation with the z vector.

As shown in Fig. 2, normal calculation requires a radius value parameter (R was set to 0.2 meters). This radius, denoted by R, defines an area around the point p (the blue area in Fig. 2). All the points inside the area are taken into account to fit a plane (least-square fitting), and the normal to this plane was computed. Fig. 3 illustrates the feature map processed on a road curve. This has been done by an interesting 3D point cloud processing library (Point Cloud

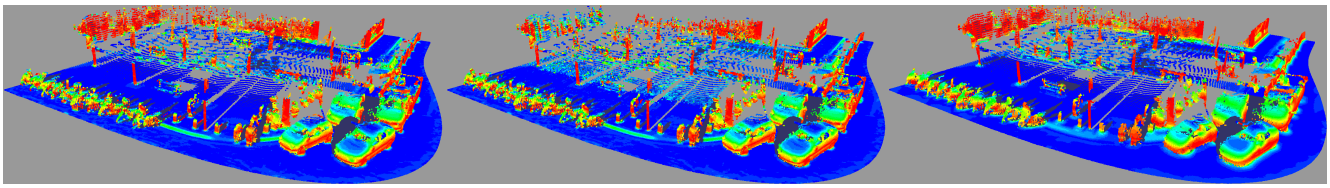

Fig. 3. Three feature maps. On the right the post-processed feature map, the middle one was obtained using a low R value while the left one is given using a high R value.

Library, or PCL) which provides several processing tools. A presentation of this library can be found in [15].

Although quite effective, the normal calculation process requires post-processing to handle changes of point densities. Indeed, depending on the speed of the Stereopolis, the distance between two successive lines of scanning may vary considerably. Furthermore, the density of points is variable depending on the distance to the LIDAR sensor, i.e., the greater the distance from a point to the RIEGL, the lower the density around that point. If there are too few points in the area defined by R, the normal is not reliable.

To handle such situations, a simple post-processing step was set up. For each point p, the density D_p of neighboring points (i.e., the number of points in the area defined by R) was calculated. If the density D_p is lower than a given parameter P_D, the radius value R is enlarged until D_p reaches the desired density (D_p was set to 15). Hence, each point p was given a radius value R_p. Fig. 3 shows three feature maps. The first one is the post-processed one, whereas the two other ones are unprocessed ones with two distinct R values. The map in the middle has been obtained using a low value of R, which gives a good level of detail for points close to the LIDAR sensor. For example here the road sides were clearly visible while further point values were inaccurate. On the contrary, the lower map was obtained with a higher R value, and details were lost close to the sensor but feature values are better further away from the sensor.

B. Prediction/Estimation model

To process the feature map described in the previous subsection, a prediction/estimation model inspired by the well known Kalman filter was designed. In the next step, the goal was to detect road edges in the 2D top view, i.e., projected in the (x, y) plane. The z component of the resulting road edges was also found out, relying on the z values of the selected observations (see below in the section).

The processing requires an initialization. This is an ordered sequence of points (or a line in the top view (x, y) 2D map) that indicates the direction of the curb processing.

Hence, the model described in this section was used to detect the positions of the road edges in the (x, y) top view. The abscissa x and ordinate y components of the position are gathered in a state vector called X_k such that $X_k = (x_k, y_k)$. In the rest of the section, we denote by X_k the result of the processing at time k. The studied model is composed of three distinct steps which will be described separately: the prediction step, the observation selection step and the result step. Fig. 4 illustrates the successive steps used to obtain result X_{k+1} from the previous results $\{X_t\}_{t=1,\dots,k}$.

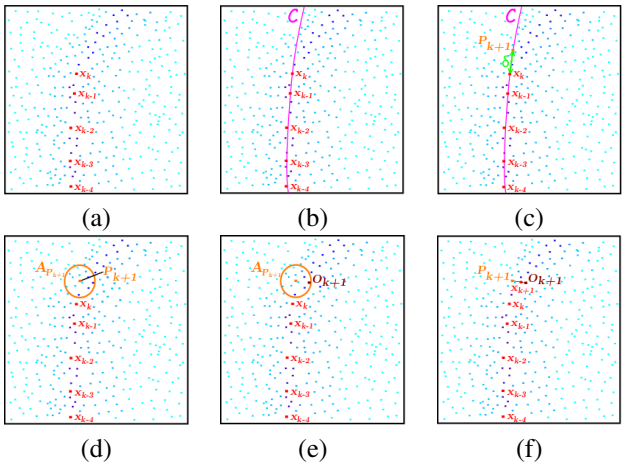

Fig. 4. The road side detection steps from time k to time $k+1$. (a) LIDAR points and previous results $x_{k-4}, \dots, x_k$. (b) polynomial fitting by a curve C (c) selection of the prediction P_{k+1}. (d) $A_{P_{k+1}}$ the search area around P_{k+1}. (e) selection of the best observation O_{k+1} (f) new result x_{k+1}.

1) prediction step:

Let P_{k+1} be the prediction of the model at time $k+1$, we have

$$P_{k+1} = f(\{X_t\}_{t=k-N_p-1,\dots,k}, D, \delta) \quad (1)$$

f denotes the process that provides the prediction using the N_p previous results $\{X_t\}_{t=k-N_p-1,\dots,k}$ and the parameters D and δ. A polynomial curve $\mathcal{C}$ of degree D was used to fit the N_p (N_p was set to 3) previous results $\{X_t\}_{t=k-N_p-1,\dots,k}$ of the process. Then, the curve $\mathcal{C}$ was used further to get the prediction relying on δ, a parameter that defines the distance between the previous result X_k and the current prediction P_{k+1} (δ was around 0.5 meters). The resulting prediction P_{k+1} is found by selecting the point distant of δ from the previous result X_k when following the curve $\mathcal{C}$. It is illustrated in Figs. 4(b) and 4(c).

2) observation selection step:

Let O_{k+1} be the selected observation of the process at time $k+1$. Let us define A_p and A'_p two areas around a given point P, respectively relying on two radius parameters r and r' (In practice, we set r and r' to 0.2 meters and 0.1 meters respectively). O_{k+1} was selected within the area $A_{P_{k+1}}$. In Figs. 4(d), (e), $A_{P_{k+1}}$ are depicted by a orange circle. O_{k+1} is chosen by:

$$O_{k+1} = g(P_{k+1}, \theta_{curb}, r, r') \quad (2)$$

where

$$g(P_{k+1}, \theta_{curb}, r, r') = \arg \max_{i \in A_{P_{k+1}}} W_i$$
$$= \arg \max_{i \in A_{P_{k+1}}} \left(\sum_{j \in A'_i} \mathcal{N}(i, j, \sigma_1) \mathcal{N}(\theta_i, \theta_{curb}, \sigma_2) \right) \quad (3)$$

g denotes the process that selects an observation relying on the prediction P_{k+1} and three parameters θ_{curb}, r and r'.

The θ_{curb} parameter corresponds to the feature value (see section II-A) characterizing a classical road curb (In the experiments, θ_{curb} value was 20) (as described in Fig. 2). A_i is the area around the point i of radius r, and $\mathcal{N}(a, b, \sigma)$ denotes the value at a position a of a Gaussian function centered in b and of standard deviation σ (here σ_1, σ_2 are two standard deviation parameters, σ_1 and σ_2 were set to 5). Fig. 5 describes the observation selection process, highlighting the fact that such a model helps select an observation for which its own feature value and the close neighborhood feature values follow the parameter θ_{curb}.

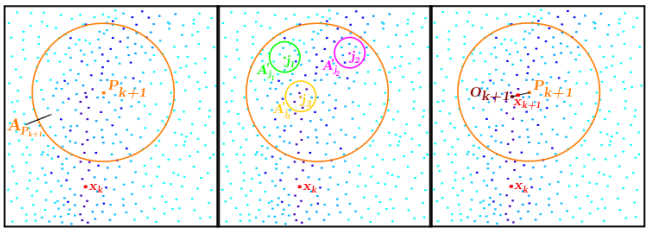

Fig. 5. Description of the observation selection step. The blue points are the LIDAR points, and the darker blue they are, the closer of θ_{curb} they are. The A'_p regions are the regions within which points are used to compute W_p (with a Gaussian weighting regarding the distance to p, see Eq. 3) and are shown for three points j_1, j_2 and j_3. Among all the points in $A_{P_{k+1}}$, j_3 is the chosen observation.

Moreover, to take into account the depth continuity of a road side, a filtering process was included. In Eq. 3, the observation selected in the A_P area should have a z value in the neighborhood of the selected observation at previous time k, i.e., $z_{O_{k+1}} \in [z_{O_k} - \epsilon, z_{O_k} + \epsilon]$ (ϵ value was 0.3 meters). In Fig. 4(e), the selected observation O_{k+1} that verifies Eq. 3 is depicted in brown.

3) result step:

The resulting road curb localization at time $k+1$ was the weighted mean of the prediction and the selected observation, i.e., that verifies

$$X_{t+1} = (1 - \alpha(W_{Obs})) \ . \ P_{t+1} + \alpha(W_{Obs}) \ . \ O_{t+1} \quad (4)$$

where $W_{Obs} = \max_{i \in A_{P_{k+1}}} W_i$ (see Eq. 3 for the definition of W_i).

The function $\alpha(W_{Obs})$ characterizes the reliability given to selected observation regarding the model predictions. It was defined by a parameter S that correspond to a multiplying factor (S value was set to 1.5) of W_{Obs}. As soon as W_{Obs} is higher than $\frac{1}{S}$, $\alpha(W_{Obs})$ is equal to 1 and the model only follows the observation:

$$\alpha(W_{Obs}) = \begin{cases} S \ . \ W_{Obs} & \text{if } S \ . \ W_{Obs} < 1 \\ 1 & \text{otherwise} \end{cases}$$

If no observation is available, i.e., in case of an occlusion, the model follows the prediction until it founds a new observation. In addition, when an occlusion occurred, the radius parameter r' that defines the observation 3D search area was increased by a multiplying parameter M until it reached a limit L_M (In practice, M value was 1.125 and L_M value was 0.25 meters). When a new observation was found, the r' parameter was set to its initial value.

As described above, this whole process was applied for both x and y components (i.e., the components of the top view) and z_k height value of X_k was given by the selected observation height z_{O_k}. In case of occlusion, the z value remained constant and equal to the height value of the last observation. When a new observation was selected (end of the occlusion), a linear model was applied to the z values around the occlusion to obtain heights of occluded X_k.

The prediction was a function of the N_p previous points (Eq. 1), where a Kalman filter needed the prediction to be a function of only the previous point. Hence, some properties of the Kalman filters for variances and gains are not verified.

This model is dedicated to LIDAR-based road side detection since it helps to handle occlusions and curb ramps. Indeed, when occlusions occur, no observation is available, and the model follows the prediction only as long as no observation is found. In the case of curb ramps, selected observations have a feature value θ_{obs} which is far away from the parameter value θ. Hence, these observations have little influence on the weighted result from prediction to selected observation, and the model still (mainly) follows predictions.

C. Road graph

To process road side detection with an increased efficiency, the point cloud to process was divided into pieces using road graph data. Indeed, as presented in Fig. 6, a road graph that describes the road in a top view is available for all roads[1]. Using this information, point clouds to process were divided into two classes of points: the "road crossing" class and the "linear road" class. Points that are close to a crossing (thus containing road bends) belonged to the "road crossing" class. Those that belonged to part of the road without a crossing, that are usually quite linear parts of the road, corresponded to "linear road" areas. Road crossing parts of the top view map were defined as parts of the roads that are at a distance lower than a given Δ parameter from the location where roads cross (Δ was set around 5 meters). This helps to process point clouds by using two distinct sets of parameters dedicated, respectively, to crossing and linear parts of the road.

The model described in the previous sections works for both crossing and linear parts of the road. In particular, when processing a linear road part, the prediction was computed using a linear model, i.e., $D = 1$. In the case of road crossing parts, a second order polynomial function (i.e., $D = 2$), which helps to better follow the variations of the sides of road bends, was selected.

[1]BD TOPO® is an IGN database containing the road graph and is freely available for research purposes

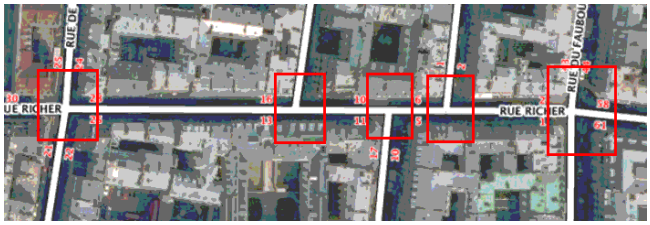

Fig. 6. A part of Paris, and the corresponding road graph in white segments. Areas corresponding to road crossing are depicted by red squares.

III. RESULTS AND EVALUATIONS

LIDAR data were obtained by a multi-sensor mobile mapping system called Stereopolis (see [16]) as shown in Fig. 7. The Stereopolis system is composed of a navigation device hybridating measurements from two GPSs, an inertial measurement unit, and a wheel odometer. The navigation device produces georeferencing of the platform that allows to locate all data and imagery in a global reference datum.

Fig. 7. The Stereopolis mobile mapping system.

Data processed in this article had been acquired using two road scanners (RIEGL LMS-Q120i) placed at the back of the system. Those LIDARs have an accuracy of about one centimeter in depth with a field of 80 degrees and they digitize up to 100 scan lines per second. Thus, the spatial sampling in the along track direction is about 10 cm.

The proposed model has been tested on several parts of the Parisian road network. Fig. 8 shows a result that is typical of those usually obtained. Indeed, when no curb ramps and no occlusions occur, the model provides accurate road side detection results. The use of the road graph model gives flexibility to the overall method since it allows to specialize the parameter sets (especially concerning the prediction step), providing increased efficiency of the detection. Indeed, if no road graph is used, and only depending on the parameter set choice, the system may diverge from real road side locations in either crossing or linear parts of the road. A constrained model dedicated to linear curves will overcome difficulties in following parabolic curves that occur in road bends. Moreover, on the other hand, a model with more freedom in the prediction step would be unstable in the presence of occlusion or road curbs in linear parts of the road.

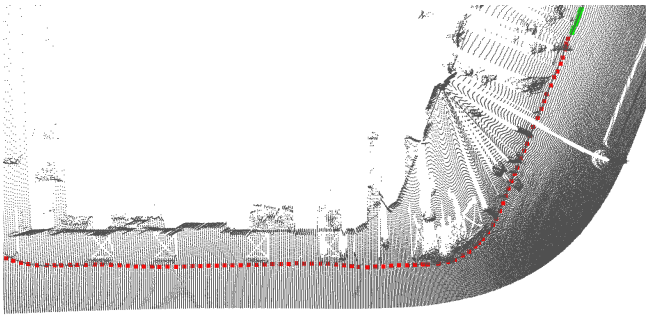

Fig. 8. Result of the prediction/estimation road side detection model in a crossing part.

Furthermore, the model is quite robust since it may recover from small errors. It is shown in Fig. 9 where the model fails to efficiently detect the whole road side. Indeed, before the second bend, a curb ramp occurs in a area close to a barrier. The feature value of the barrier temporally attracts the model since it is more similar to a curb profile pattern than the curb ramp. However, having passed the barrier and the curb ramp, the model managed to reach the part of curb that is found after the curb ramp to efficiently detect the second bend.

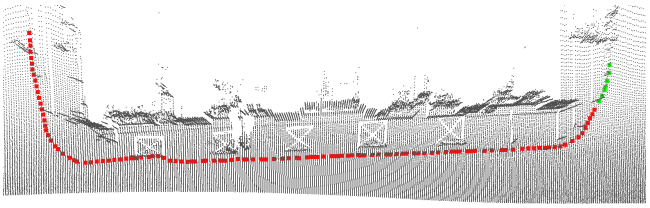

Fig. 9. Result of the prediction/estimation road side detection model in a part of the road composed of two crossing parts and one linear part.

Satisfying results have been obtained, even in the presence of occlusions and curb ramps, especially in the linear parts of the road. Fig. 10 illustrates the right behavior of the model regarding occlusions. In this example, many vehicles parked between the LIDAR sensor and the road side to detect. However, the model uses the previous results to get a fine prediction of the road side. Following these predictions, the model passes through the occlusions and finds the newly observable part of the road side.

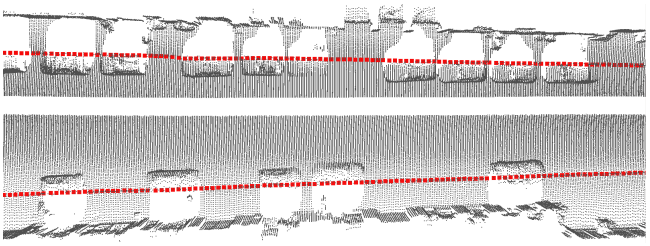

Fig. 10. Result of the prediction/estimation road side detection model in two linear parts with many occlusions.

However, there is still a situation that is difficult to handle when using the proposed model. When a curb ramp or an occlusion occurs in the bending part of the road, the system usually fails to follow the road side. Indeed, if no 3D curb profile pattern is observed in a bend, the prediction of the

model will fail to retrieve the shape of the road side. To cope with this issue, several paths may be explored. A road side detection in a top view image may be possible (if no vehicle stand over the road side). Considering several laser scans of the same road would also be an option.

However, to handle those last problematic situations, a semi-automatic version of the proposed model is presented.

A. Semi-automatic version

Our system is quite efficient in most cases of road side, but may with curb ramps or occlusions that may occur in bends. Hence, we decided to investigate a method of easily correcting small errors by adding a user-machine interface step to provide an efficient semi-automatic tool to process large point cloud data.

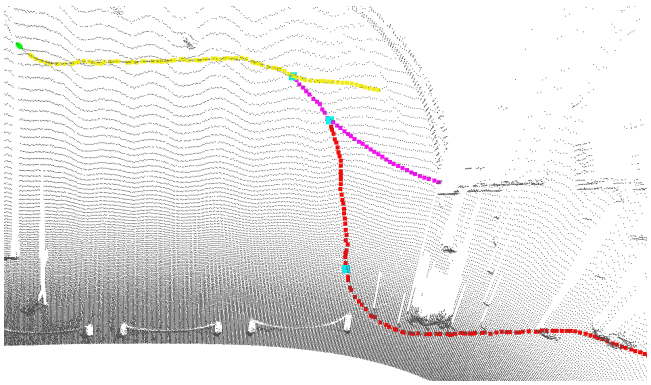

Fig. 11. Result of the prediction/estimation road side detection model with two control points. The yellow curve is the initial result, the purple curve is the result with one control point and the red one is obtained using two control points. The cyan points correspond to user mouse clicks. The second cyan point has been clicked two times: as second click for the first redirection and as first one for the second redirection.

It is presented in Fig. 11 which presents a curb ramp occurring between two road side bends. The model does not manage to follow the road side (the yellow curve) and an interaction with the user is needed. This process is very simple, it only requires two mouse clicks. The first one tells the system from where to reprocess the road side (i.e., when the system begins to fail) while the second one indicates a position further of the right road side. This latter one will attract the process by enforcing the system to make predictions in the direction of this "good" road side (i.e., the second mouse click). This first user-interaction produces the purple curve, that also drifts away from the road side. Adding the second interaction produces an efficient result in road side detection on a typically difficult part of the road side: a curb ramp occurring between two close road side bends. In Fig. 11, user mouse clicks are depicted in cyan. The second cyan point has been clicked two times: as second click for the first redirection (i.e, the one indicating the right the direction) and as first one for the second redirection (i.e, where the process fails to follow the road side).

IV. CONCLUSION

In this paper, a road side detection method for LIDAR data is presented. It is based upon a prediction/estimation model, that helps handle occlusions due to obstacles such that vehicles parking between the LIDAR sensors and the road side, and curb ramps that may be encountered on the road sides. In such situations, the system reconstructs the missing parts of road edges. The system has been successfully tested on large parts of the Parisian road network. A simple user-machine interaction step is also proposed, hence providing an efficient tool for processing of large amounts of data.

A perspective to this work is to use road side retrieval for precise road reconstruction. The aim for the future is to use road side points of both sides to compute the shape of the road. The road reconstruction may even be integrated in the whole prediction/estimation to directly compute the couple of road side points and the road shape, hence providing a road shape model in presence of obstacles.

REFERENCES

[1] G. Floros and B. Leibe, "Joint 2d-3d temporally consistent semantic segmentation of street scenes," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[2] T. Michalke, R. Kastner, J. Fritsch, and C. Goerick, "A self-adaptive approach for curbstone/roadside detection based on human-like signal processing and multi-sensor fusion," in *In Proceedings of the IEEE Intelligent Vehicles Symposium*, 2010.

[3] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," *In Proceedings of the European Conference on Computer Vision*, 2008.

[4] F. Oniga, S. Nedevschi, and M. Meinecke, "curb detection based on a multi-frame persistence map for urban driving scenarios," in *IEEE Conference on Intelligent Transportation Systems*, 2008.

[5] J. Siegemund, U. Franke, and W. Förstner, "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields," in *In Proceedings of the IEEE Intelligent Vehicles Symposium*, 2011.

[6] F. Oniga and S. Nedevschi, "processing dense stereo data using elevation maps: road surface, traffic isle and obstacle detection," in *IEEE Transactions on Vehicular Technology*, 2010.

[7] M. Ravanbakhsh, C. Fraser, and M. Awrangjeb, "Extraction of traffic islands via active contours without edges," in *International Archives of Photogrammetry and Remote Sensing*, 2010.

[8] J. C. Andersen, M. R. Blas, O. Ravn, N. A. Andersen, and M. Blanke, "Traversable terrain classification for outdoor autonomous robots using single 2D laser scans," *Integrated Computer-Aided Engineering - Informatics in Control, Automation and Robotics*, vol. 13, no. 3, pp. 223–232, 2006.

[9] J. Stückler, H. Schulz, and S. Behnke, "In-lane localization in road networks using curbs detected in omnidirectional height images," in *Robotik 08*, 2008.

[10] W. Wijesoma, K. Kodagoda, and A. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 456–464, june 2004.

[11] J. Maye, R. Kaestner, and R. Siegwart, "Curb detection for a pedestrian robot in urban environments," in *In Proceedings of the International Conference on Robotics and Automation*, 2012.

[12] W. Zhang, "Lidar-based road and road-edge detection," in *In Proceedings of the IEEE Intelligent Vehicles Symposium*, 2010.

[13] L. Zhou and G. Vosselman, "Mapping curbstones in airborne and mobile laser scanning data," *Int. J. Applied Earth Observation and Geoinformation*, vol. 18, pp. 293–304, 2012.

[14] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking lidar sensor," *IEEE Trans. Vehicular Technology*, vol. 61(3), pp. 971–985, 2012.

[15] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *In Proceedings of the International Conference on Robotics and Automation*, 2011.

[16] N. Paparoditis, J.-P. Papelard, B. Cannelle, A. Devaux, B. Soheilian, and E. H. N. David, "A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology," *Revue Française de Photogrammétrie et de Télédétection*, vol. 200, pp. 69–79, October 2012.