Curb detection in urban traffic scenarios using LiDARs point cloud and semantically segmented color images

Selma Evelyn Catalina Deac, Ion Giosan, and Sergiu Nedevschi, *Member, IEEE*

***Abstract*— In this paper we propose a robust curb detection method which is based on the fusion between semantically labeled camera images and a 3D point cloud coming from LiDAR sensors. The labels from the semantically enhanced cloud are used to reduce the curbs' searching area. Several spatial cues are next computed on each candidate curb region. Based on these features, a candidate curb region is either rejected or refined for obtaining a precise positioning of the curb points found inside it. A novel local model-based outlier removal algorithm is proposed to filter out the erroneous curb points. Finally, a temporal integration of the detected curb points in multiple consecutive frames is used to densify the detection result. An objective evaluation of the proposed solution is done using a high-resolution digital map containing ground truth curb points. The proposed system has proved capable of detecting curbs of any heights (from 3cm up to 30cm) in complex urban road scenarios (straight roads, curved roads, intersections with traffic isles and roundabouts).**

I. Introduction

Curbs are the most common road delimiters in urban environments. Their precise detection is an important requirement in any Advanced Driver Assistance Systems (ADAS) and Autonomous Driving Systems for many tasks. They are a crucial early step for detecting the navigable road area, for trajectory planning, vehicle localization, or parking.

It has been over a decade since the detection of urban curbs was first studied. Throughout time, dedicated methods based on the type of sensors used for the detection were proposed. Monocular cameras were the first sensors used for the curb detection task, followed by stereo cameras, radars and LiDARs.

However, each sensor has its advantages and limitations. Monocular cameras for example offer RGB information, but they are sensitive to illumination variations and lack spatiality. A LiDAR sensor, on the other hand, provides accurate depth measurements for its surrounding environment, being independent on the illumination variations, but does not have color information and offers sparse data. In order to compensate sensors' limitations, the recent researches are focusing on multi-sensor approaches. The most popular combination of all are the LiDARs and camera systems.

The curb extraction for a LiDAR sensor usually implies sweeping each LiDAR row in the search of a region of points which capture the most common curb geometric structure, a step. However, many false positives are present. Small curbs are hard to detect in these systems because of the lack of an a priori information which should provide context.

Semantic segmentation on color camera images is a multi-classification problem, which implies assigning an object class to each pixel of an image. Semantic segmentation using Convolutional Neural Networks (CNNs) applied on a vehicle's monocular color camera images has become a very popular and useful task. The recent optimizations made in the networks design have ensured their run in real time. A first limitation is the expensive annotation required in order to have a large dataset for training. Recently, automatic annotation systems are more and more researched in order to overcome this issue. The second limitation is the lack of a precise spatial positioning of the surrounding objects. Thirdly, the semantic segmentation is memoryless. This means semantic classes of pixels may vary drastically between consecutive frames although the environment did not change much. Also, in order to keep the semantic segmentation running in real time, a perfect classification of each pixel will never be possible and erroneous segmentation will always be present.

This paper, tries to solve the limitations stated earlier by using a multi-sensor fusion approach. The main idea of the proposed method lies in the fusion between the semantic labeled camera images with the LiDAR point cloud. In this way, the 3D point cloud processing for the detection of urban curbs will benefit from a search space reduction and an access to the context information.

Curbs are static elements of the urban environment, which persist in consecutive frames. As a geometrical shape, they are best described using line segments. Following these observations and the work in [1], we propose an improved and novel curb detection method that is capable of detecting curbs of any heights and geometrical shapes in complex urban road scenarios. The main contributions of the proposed system refer to:

- Improved spatial cues computation and filtering module for the detection of curbs – capable of detecting small curbs (as low as 3cm), regular curbs (≈10-15cm), and large curbs (≈25-30cm) with a high precision rate
- A novel outlier removal algorithm – which is based on a local model approach developed by considering both

Selma Evelyn Catalina Deac, Ion Giosan and Sergiu Nedevschi are with the Technical University of Cluj-Napoca Romania, Computer Science Department (e-mail: selma.goga@cs.utcluj.ro, ion.giosan@cs.utcluj.ro, sergiu.nedevschi@cs.utcluj.ro).

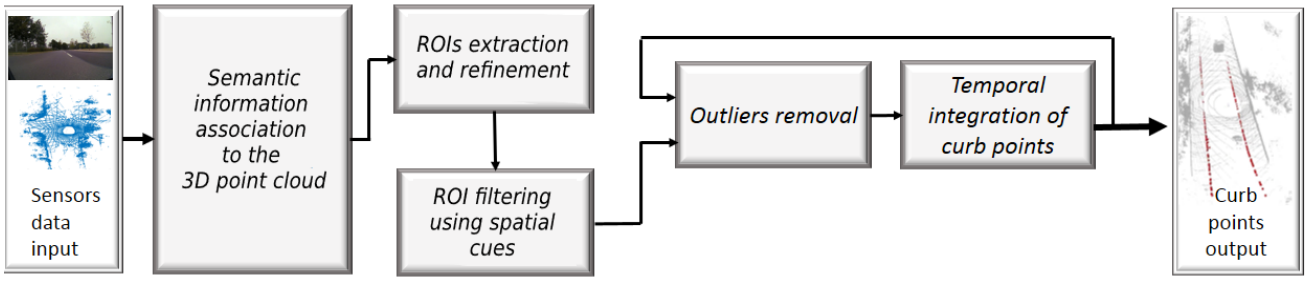

Figure 1. The pipeline of the proposed method.

density and spatial proximity. This increases in the end the robustness and precision of the results.

- A temporal integration module – which tackles the sparsity of the detected curb points from a single frame.
- An objective evaluation of the entire proposed solution, done by using a high resolution digital map which contains ground truth curb points

Along with the added improvements, the algorithm remains independent of the shape of the urban road, detecting curbs with high precision in many road scenarios: straight road, curved road, fork roads and complex intersections.

The research was done in the framework of the UP-Drive H2020 project having as goal the design of an automated vehicle. The sensory architecture of the systems includes five 360 degrees with 16/32 layers LiDARs, and four fish eye area-view cameras. The calibration and cross-calibration of the sensors, the motion correction of LiDARs points and the LiDAR to camera data fusion are presented in [2]. The software architecture include a semantic segmentation module of the fish eye camera color images [3]. The curb detection module is developed in this context.

The paper is classically structured as follows: First, the state of the art for curb detection is presented in Section II. Then, in Section III, the pipeline of the new curb detection algorithm along with the detailed description of each stage in subsections are presented. In Section IV, the evaluation method is presented followed by the experiment results from Section V. Finally, we close with the conclusions and possible further improvement in section VI.

II. Related Work

Curb detection has been a subject of major study in autonomous driving applications throughout time.

The most popular sensors used nowadays for curb detection are 2D/3D LiDAR sensors [4-6],[9-11],[14],[16] mostly because of their high accuracy measurements and because they are independent on lighting conditions.

Authors in [5],[7],[12] use the representation of a grid map in order to detect the curb cells. Because of the sparse measurements, detecting a curb step inside a grid map is harder to achieve and does not offer a precise spatial positioning of curb points.

An advantage 3D LiDAR sensors have when it comes to curb detection, is that each 360 degrees' laser beam spin accumulates ordered set of points. This property is the most exploited in literature [4], [9], [13], [14], [16] and has proved to give the best results when it comes to precise curb detection methods. Authors use various spatial cues in the search of the curb step model on each laser beam spin: elevation difference (as in [4], [9], [14]), normal vector [16], angle [4].

Using a 3D LiDAR as a standalone sensor for curb detection has disadvantages. Authors in [4] noticed that using only a LiDAR sensor, the system it is not capable to provide enough context information for the detection of curbs. In the search of context, authors in [9] and [11] assume curbs are usually the nearest small size obstacle from the vehicle. They build a radial grid and keep the nearest candidate curb point to the car as being curb points. However, vehicle wheel tires which have similar aspect to that of a curb might be detected as false positives using this constraint.

Authors in [9] also propose the usage of Smooth Arc Length Feature which assumes that curbstones are sepparate two smooth surfaces. However, using this constraint the curbs between road side and vegetation will not be considered correctly.

In [14], the authors start from the assumption that urban curbs are on the left and right of the autonomous vehicle. The data is thus split in two sets one from the right and the other from left. Although this assumption holds in many urban road cases, the system will fail to detect more complex curb scenarios when traffic isles and roundabouts are present.

A viable solution for receiving context is fusing the 3D LiDAR sensor with a monocular camera sensor. In [10] authors combine LiDAR depth images with raw camera RGB images in order to detect curbs. A fusion between semantic labeled camera images and LiDAR's point cloud is proposed in [1]. The context provided by the semantic labels to the 3D points successfully reduces the search for the curb points and ensures the detection of curbs in complex road scenes. However, the curb detection system in [1] estimates the curb points independently and does not take into account the entire model of the curbstone segment. Outliers are thus present and not filtered out.

When comparing the curb detection methods using a 3D LiDAR sensor with other used sensors (eg. stereos), the density of the final curb points from a frame is not high.

Authors in [17], [18] detected curbs using a stereo sensor. Although the density of stereo is known to be high, both

authors propose to enrich the curb data points making use of the curbs' static property.

The work of [18] uses a temporal integration of frames in order to enrich the curb points from the current frame. On the enriched data points, a cubic spline model is fitted onto the curb data. Authors in [17] also integrate curb points from multiple frames using a conditional random field (CRF) method. A 3rd order polynomial is fitted onto the curb points in the end and a prediction system is built using a Kalman filter.

Authors in [15] used the temporal integration idea for enriching curb points detected by a LiDAR sensor. They used the dense detected curb points in order to correct the GPS location errors of the intelligent vehicle.

Regardless of the used sensor for curb detection, the modeling of curb points is similar for all methods found in literature. This modeling is usually done for the detection of outliers (as in [9] and [11]), or for the curb reconstruction ([18], [17]). The used models can be of two types: parametric (linear models [9], quadratic polynomials [11], 3rd order polynomials [17], NURBS [8], splines [18]) and non-parametric (free-form [4], [15]).

The authors in [11], [17] use a high order polynomial for curb fitting. Curbs from traffic isles and roundabouts remain hard to detect. The usage of splines [18] might provide a better representation but it is unfortunately sensitive to erroneous curb points. To address the problem of complex urban scenarios, the usage of a non-parametric model remains the most promising solution for now.

III. Proposed Solution

In this section we present the stages of the proposed solution. The pipeline of the system can be seen in figure 1. The proposed algorithm for the detection of urban curbs is based on a top down approach and lies in five stages:

A. *Semantic information association to the 3D point cloud* – where the low level fusion between LiDAR data and semantic images is performed.
B. *ROIs extraction and refinement* – in which curb ROIs are detected based on the semantic labels of the points on each LiDAR scanning line. Each ROI is expanded in order to increase the chance of enclosing inside it a curb's step.
C. *ROI filtering using spatial cues* – in which the ROI boundaries are shrink in order to match the candidate curb edge points.
D. *Outliers removal* – where the candidate curb points which best describe a real curbs' shape are identified.
E. *Temporal integration of curb points* – where the curb object persistency property is used in order to densify the detected curb points.

A non-parametric curb model is obtained in the end in the form of a list of 3D points. The lowest curb point is stored in the list for each detected curb.

A. Semantic information association to the 3D point cloud

In order to obtain the semantically labeled images from the vehicle's monocular color cameras we used the semantic segmentation solution developed in the framework of the UP-Drive project [3]. The set of considered classes was extended with the curb class by improving the annotation of the training set and retraining the segmentation system.

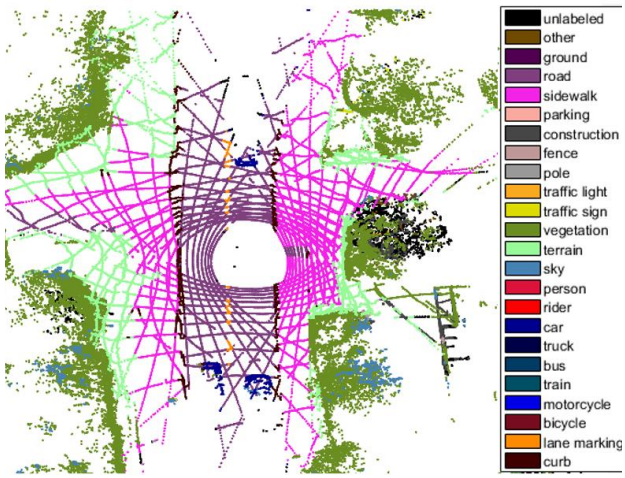

Figure 2. The point cloud with semantic information associated to it.

When the segmented images are available in the system, along with an entire LiDAR 360 degrees' sweep, a fusion between the two is done by projecting each LiDAR point onto the images. The semantic label of a pixel is associated to the corresponding 3D point which falls onto it (see [2]).

After obtaining the semantically enhanced 3D cloud (figure 2), we filter the points based on their semantic classes. We separate the points classified in the ground category (vegetation, road, curb, lane marking, terrain and sidewalk) from the other points (the remaining classes). Only the 3D points from the ground category classes are next processed.

B. Curb ROIs extraction and refinement

We start with the assumption that the point cloud coming from a LiDAR sensor is in an ordered sequence. A curb Region of Interest (ROI) is a chunk of consecutive 3D points extracted from a scanning line of the LiDAR sensor. A scanning line can contain more than one ROIs.

In order to extract a ROI, we will make use of the semantic classes. On each row we search for a succession of points with semantical classes which best describe the presence of a curb. The best guess is taking the region of consecutive points, labeled as curb class. The second option is to extract the small region where a transition between the terrain, vegetation or sidewalk class and the road class is present.

Because we want to make sure a ROI encapsulates the entire curb structure (i.e. the curb step), a further refinement is needed. For this, the ends of the previously obtained ROIs are dilated in both directions with predefined distances: λ_1 for the chunk containing curb labeled points and λ_2 for the chunk marking a transition between terrain, vegetation or sidewalk and road. λ_1 is smaller than λ_2.

C. ROI filtering using spatial cues

The idea is to shrink each ROI, such that it fits as much as possible on a curb's structure: one end of the ROI will be the

lowest curb edge (the point neighboring the road side region), while the other, the highest curb edge point (the point neighboring the sidewalk/vegetation/terrain region).

A curb's shape inside a ROI is an ascending height step between the road region and the vegetation or sidewalk region. We start by assuming all ROIs contain such a structure and we try to find in each ROI the monotonically ascending regions of points starting from road side to the adjacent road area. Because the standard step shape of an urban curb is the same for all curb types (i.e. small curbs, regular curbs, large curbs), we normalize the heights of the points inside the ROI. This allows the detection of curbs of any heights (from 3cm up to 30cm). First order derivative (FOD) is next computed on the normalized heights from the road side to the adjacent road side in order to detect the monotonically ascending regions. A smoothing filter is applied two times on the FOD for eliminating any unwanted spikes. Usually, the point with maximum slope ($peakPoint$) of the curb is found in on the maximum value of the FOD (see figure 3).

The first spatial filter is applied using the FOD:

1. **Normalized height variation feature** (v_h) - the variation on the normalized height (on the Z axis in the xOyOzO coordinate system) of the n points found inside a ROI is computed using the following formula:

$$v_h = peakPoint_Z - mean_Z \quad (1)$$

$$mean_Z = \frac{1}{n}\sum_{i=1}^{n} ROI\_Point_{i_Z} \quad (2)$$

Its value should be inside an interval $[v_h^{min}, v_h^{max}]$ in order for a ROI to be further processed.

The steps for computing v_h can be seen in figure 3. A curb should mark a height transition between a lower region (the road side) and an upper region (the sidewalk or vegetation side). As an intuition, the normalized height variation feature (v_h) tries to check the previous statement, by giving valuable information about the curb in the context of its surroundings. The ROIs which do not pass this filter will be discarded.

The next computed spatial cues will check the vertical structure of the curb independent on its surroundings.

Having the index of the maximum peak, we will start searching in opposite directions for the first local minimums found. The index of the found minimum will represent the best guess of the lowest and highest curb points. Let us denote the point near the road region with p_{low} and the point near the adjacent road region (sidewalk or vegetation) with p_{high}. Figure 3 shows an illustration of finding the p_{low} and p_{high} points.

After finding p_{low} and p_{high} for a ROI, we apply the next two filters:

2. **Height feature** – Δh – a curb should have a height between $H1$ and $H2$, where $H1 < H2$. The value of Δh is the absolute difference of the real measured heights of p_{low} and p_{high}.

3. **Width feature** – the Euclidean distance between p_{low} and p_{high} is also a good indicator along with the height filter. The width and height give valuable indication about the sloppiness of the line segment between p_{low} and p_{high}. This width should be inside a predefined interval $[W1, W2]$.

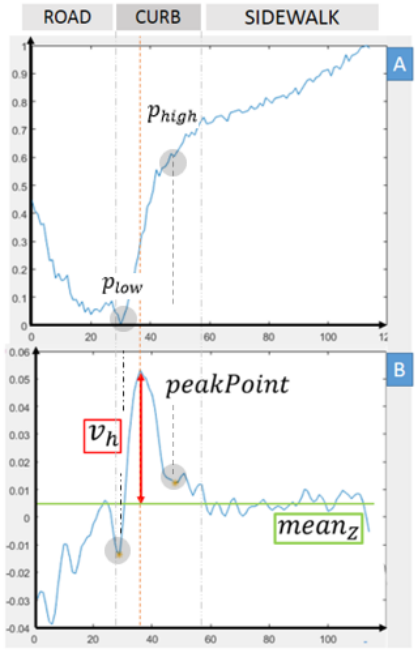

Figure 3. Illustration of *Normalized height variation filter computation*. The vertical axis represents the normalized height while the horizontal axis is the index of points inside a ROI. The upper figure A represents the profile of a Curb ROI. The lower figure B represents the first order derivative (FOD) of the curbs' profile.

The ROIs which passed all the three feature tests will be kept as candidate curb regions while the others will be discarded.

A list of 3D points containing the detected low edge point from each candidate curb ROI is passed to the next pipeline stage.

D. Outliers removal

Although the features computed earlier try to capture as much as possible the real curb points, some erroneous data might still be present, especially in the areas with vegetation. The erroneous points appear as sparse chunks of points which are distributed unevenly on the lateral and longitudinal axis,

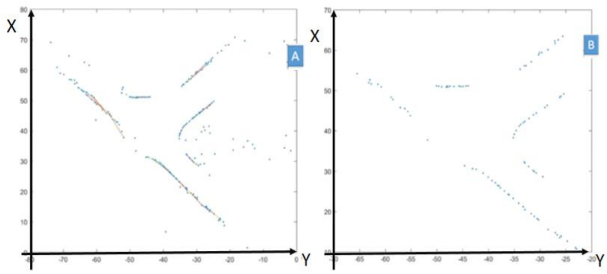

Figure 4. *Applying the outliers' removal algorithm*. In image A, the candidate curb points are seen from a bird's eye view. The best lines found by the local linear based outlier removal algorithm are drawn. In image B, only the point's which fall in the previously found lines are kept as the real curbstones points.

while the true curb points are usually linearly grouped on several short length line segments (see figure 4). A global outlier removal method which relies on statistical methods is impossible to implement and give good results.

From a bird's eye perspective, curbstones found on any shape of road can be described using connected line segments. Even a circular shaped curb can be reconstructed using consecutive small polylines. Because the most obvious feature of a curb is its shape, the proposed outlier removal relies on a local linear fitting model based approach. The spatial proximity plays an important role in the selection of the final curb points.

The algorithm for this method is presented in the figures Algorithm 1 and 2.

Input: P
Output: CP
Parameters: $r, K, D, distTH, inlPcTH$

1. $DG = buildDensityMap(P, r)$
2. $CP = \{\}$
3. $for\ each\ occupied\ cell\ c\ in\ DG$
4. $\quad CellL = \{\}$
5. $\quad CellL = NN(c, D, DG)$
6. $\quad if\ size(CellL) < K$
7. $\quad\quad CellL = KNN(c, K)$
8. $\quad CP.append(RemOuts(CellL, distTH, inlPcTH))$

Algorithm 1. Extracting local chunks of cells using a mixed NN approach for applying the outlier removal algorithm

The outlier removal stage uses as input a merged cloud P composed of 3D points from the current candidate curb points and the past detected 3D curb points. This assures that a high density of points is present near the neighbourhood of the real curb points. Based on this observation we first build a density grid (Line 1 – Alg. 1).

The algorithm next tries to extract chunks of cells in a Nearest Neighborhood (NN) fashion, using the Euclidean distance D as search radius (Line 5 – Alg.1). If enough neighboring cells are identified inside the radius D, then the outlier removal method, $RemOuts()$, is next applied on the chunk of cells to identify the best line model (Alg.2). Else, it searches for the minimum K number of its nearest neighbours and tries to identify the best line model for them (Alg.2). This mixed neighborhood approach makes sure that the best neighboring cells are chosen for applying the outlier removal algorithm.

The line hypotheses are built starting from the first two cells with highest density and ending with the ones with the lowest density (Lines 2-6 – Alg.2). The indices of the cells which fall on each candidate fitting line are stored in *idxF* (Line 9 – Alg. 2). If the percentage of *idxF* cells number from *CellL* cells number is greater than a threshold then the best line is found (Line 10-12 – Alg. 2). In this case the 3D points found inside the *idxF* cells are stored in *Pts* (Line 14 – Alg.2). The final curbs' low edge points CP are all the points found inside the inlier cells (Line 8 – Alg. 1).

Input: $CellL$
Output: Pts
Parameters: $distTH, inlPcTH$

1. $Pts = \{\}$
2. $descendingOrderSortUsing(CellL, "density")$
3. $ok = false$
4. $for\ each\ cell\ c1\ in\ CellL$
5. $\quad for\ each\ cell\ c2\ in\ CellL, c1 \neq c2, ok = false$
6. $\quad\quad [a, b, c] = lineCoefficients(c1, c2)$
7. $\quad\quad distsL = idxL = distF = idxF = \{\}$
8. $\quad\quad [distsL, idxL] = distToLine(CellL, a, b, c)$
9. $\quad\quad [distsF, idxF] = (distsL < distTH)$
10. $\quad\quad if\ size(idxF) * 100 / size(CellL)$
11. $\quad\quad\quad\quad > inlPcTH$
12. $\quad\quad\quad ok = true$
13. $if\ ok = true$
14. $\quad Pts = getPtsInCellsIdx(CellL, idxF)$

Algorithm 2. The outlier removal algorithm *RemOuts*()

E. Temporal integration of curb points

Curbs are static elements of the environment. They tend to persist from one frame to another.

Lidar sensors give sparse 3D clouds. This sparsity tends to increase direct proportionally with the depth of a point. In order to exploit the first statements and to solve the LiDAR sensor's disadvantage, we have found of real help the integration of detected curb points from consecutive frames.

Temporal integration implies accumulating points in the current frame from a fixed sequence of past frames. The past curb coordinates are transformed in the current coordinates system.

For this we used information from the IMU sensor of the car, such as the yaw rate, the linear velocity and the timestamp of a frame. The *Constant Turn Rate and Velocity (CTRV)* motion model was used. Let us consider the position x, y of a low edge curb point from a past frame on X, Y axis. The transformed coordinates are computed using the formula:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R_{xy} \begin{bmatrix} x - T_x \\ y - T_y \end{bmatrix} \quad (3)$$

where,

$$T_x = t_d \cos\frac{\theta}{2} \quad (4)$$

$$T_y = t_d \sin\frac{\theta}{2} \quad (5)$$

$$R_{xy} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (6)$$

θ represents the steering angle of the car from one frame to another and is computed based on the yaw rate and Δt;

t_d represents the linear displacement of the car from one frame to another. It is computed based on the distance d traveled by the car from the past frame to the current one as follows:

$$t_d = 2\frac{d}{\theta}\sin\frac{\theta}{2} \quad (7)$$

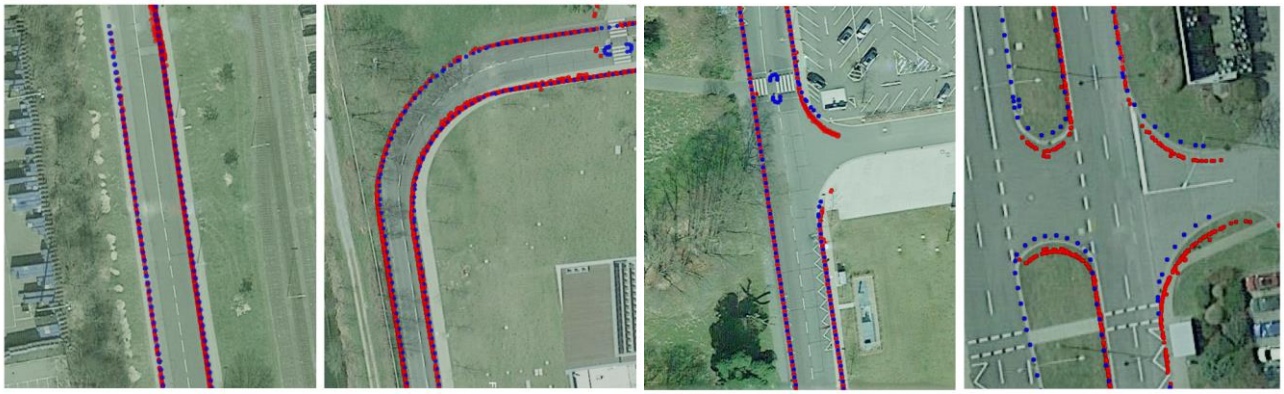

Figure 5. Plotted curb points onto the satellite images overlapping the map points. The curb points are marked with red, while the map points are marked with blue. Major displacement errors appear from the DGPS sensor errors accumulated with the yaw angle errors from the car's IMU sensor.

Although the final points are densified, temporal integration is sensitive to road bumps. A point's displacement on the pitch angle clearly appears in such cases. A temporal solution for this problem is to find the deviation between the pitch from the aligned old frame points and the one from the new frames points and to correct the displacement by realigning the old points in order to correspond to current points pitch angle. However, small error in pitch measurement might be present and, at a large number of temporal integrated frames, they might accumulate in time and might be visible. Research still needs to be done in this direction.

The set of final curb points C is obtained after temporally fusing the curb points from m previous frames with the set of current frames' curb points CP. Based on several experiments, we set the best value for m to 5.

IV. EVALUATION METHOD

In order to evaluate the quality of the detected curb points we have considered two metrics: average distance of the detected curb points to the ground truth curb points ($AVGD$) and the positive predictive value – precision – of the detected curb points (PPV).

In the absence of an agreed benchmark for curb detection evaluation, we used a high-resolution ground truth map privately used in the frame of the UP-Drive research project. The map curb points are sampled with a gap of 10cm between them. Each point M_i from the entire set of map points M is defined by a latitude and a longitude in the Earth's geographic coordinate system. After obtaining the set of curb points C of size p based on the proposed method, we try to match them against the map's points. In order to do this, we first need to convert the curb point's coordinates from the vehicles' coordinate system to the Universal Transverse Mercator (UTM) coordinate system. This implies aligning the two coordinate systems. Once curb points UTM coordinates are available, a conversion to the geographic coordinate system which use latitude and longitude is done.

Due to the fact that the distances between the detected points and the ground truth points are very small we can assume that the Earth has a constant radius R in that vicinity and compute the distances between them. Thus, we use the following formulas for computing the $AVGD$ score, where $Lat(C_j)$ is the latitude and $Lon(C_j)$ is the longitude of a curb point C_j:

$$AVGD = \frac{1}{p}\sum_{j=1}^{p} \min_{i=1,n}\left(dist\left(C_j, M_i\right)\right) \tag{8}$$

$$dist\left(C_j, M_i\right) = \sqrt{distNS\left(C_j, M_i\right)^2 + distEW\left(C_j, M_i\right)^2} \tag{9}$$

$$distNS\left(C_j, M_i\right) = R \cdot distLat\left(C_j, M_i\right) \tag{10}$$

$$distEW\left(C_j, M_i\right) = R \cdot \cos\left(Lat\left(C_j\right)\right) \cdot distLon\left(C_j, M_i\right) \tag{11}$$

$$R = 6378137\ m \tag{12}$$

$$distLat\left(C_j, M_i\right) = \frac{\left|Lat(C_j) - Lat(M_i)\right|}{180} \cdot \pi \tag{13}$$

$$distLon\left(C_j, M_i\right) = \frac{\left|Lon(C_j) - Lon(M_i)\right|}{180} \cdot \pi \tag{14}$$

A point from C is considered to be a true-positive (TP) detection if it is closer than T cm from a point from the ground truth map, where $T = 30cm$. This value was chosen with respect to the existing DGPS and IMU sensor errors that lead to an imperfect mapping of the detected curb points. The PPV is computed by using the following formulas:

$$PPV = \frac{TP}{TP+FP} = \frac{\sum_{j=1}^{p} TP(C_j)}{p} \tag{15}$$

$$TP\left(C_j\right) = \begin{cases} 1\ (true\ positive), if\ \min\limits_{i=1,n}\left(dist\left(C_j, M_i\right)\right) \le T \\ 0\ (false\ positive), \qquad otherwise \end{cases} \tag{16}$$

V. EXPERIMENTAL RESULTS

The evaluation was performed on a recorded sequence obtained by driving approximately 3.5 km. Each frame of the sequence contains the odometry data of the car from the IMU and DGPS sensor, along with the complete 360 degrees' environment scans of each of the five LiDAR sensors and four RGB images coming from the monocular cameras.

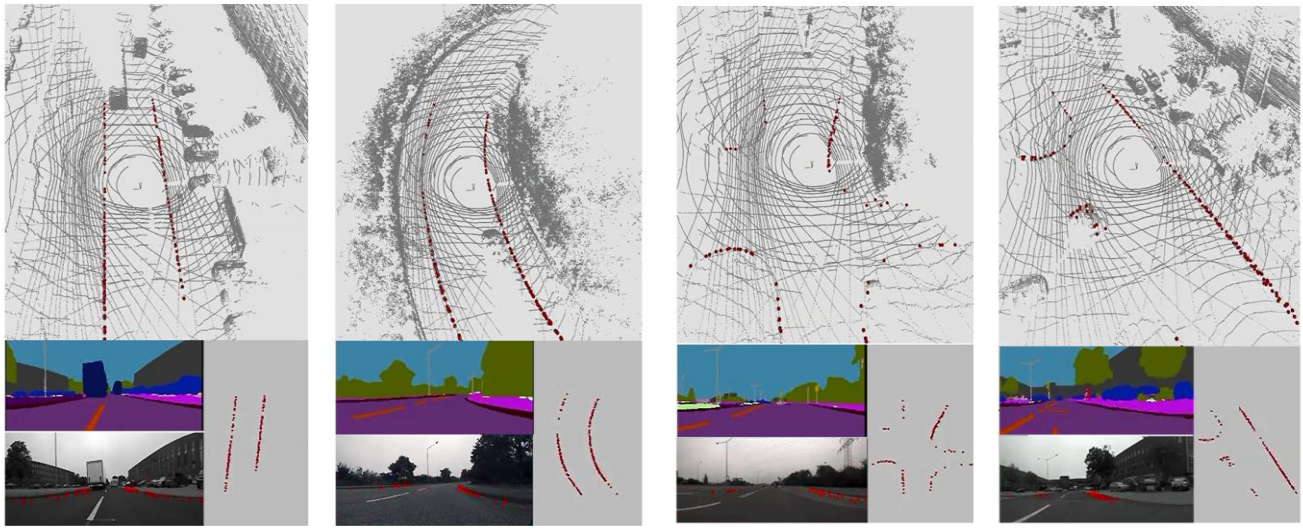

Figure 6. Results from the evaluation stage for different road shapes. The camera images are from the front monocular camera sensor. The car is moving forward (upward).

1. *Hardware infrastructure*

A robust detection of small objects like curbstones requires many LiDAR scanning lines. In order to increase the number of scanning lines and realize entire coverage of the surrounding environment, five LiDARs with 360 degree FOVs are mounted on top of the vehicle. Three of them perform scans along 32 lines (front-left, front-right, rear-left) covering 40 degrees in vertical direction and the other two perform scans along 16 lines (rear-center, rear-right), covering 30 degrees in vertical direction. With the given LiDARs setup, a coverage of at least one scanning line per degree is realized for each side of the vehicle.

Four cameras are placed to offer a 360 degrees' view of the scene. Fisheye lenses have a large field of view of 191 degrees' horizontal and 150 degrees vertical with possible overlapping regions between different cameras. The images are successfully used for the semantic segmentation process of all pixels.

An advanced inertial and navigation sensor DGPS/IMU is also mounted on the vehicle for measuring its motion, position and orientation. The position and heading angle are used for mapping the detected curbs in the Earth geographic coordinate system. The sensor measurement errors are at most 0.2m in position and 0.1 degrees in heading angle.

2. *Parameters tuning*

After continuous trial and error, we have set the following threshold parameters for the proposed algorithm.

The limits for the first two spatial filters parameters are: $\boldsymbol{v_h^{min}} = 0.013,\ \boldsymbol{v_h^{max}} = 0.13, H1 = 0.03m,\ H2 = 0.3m.$

Because each of the five LiDAR sensors had a different tilt angle relative to the ground plane we had to tune the distance feature thresholds for each sensor independently. For the Front-Left LiDAR we used $W1 = 0m$ and $W2 = 1.7m$; Front-Right: $W1 = 0m$ and $W2 = 2m$; Rear-Right: $W1 = 0m$ and $W2 = 1.7m$; Rear-Left: $W1 = 0.2m$ and $W2 = 1m$; Rear-Center: $W1 = 0.1m$ and $W2 = 1.5m$.

The parameters used for the outlier removal algorithm are: $r = 0.8m,\ K = 10,\ distTH = 0.05m,\ inlierPcTH = 30\%$, D = 5m.

3. *Experimental evaluation*

After optimizations, the supplementary time introduced by the proposed curb detection algorithm (stages B., C., D. and E. of the pipeline) is of 5.5ms on a PC with an Intel i7-3770K CPU and a frequency of 3.50GHz. The algorithm was implemented on one CPU core.

TABLE I. CURB DETECTION AND MAPPING RESULTS

Scenarios	**AVGD**	**PPV**
Straight road	0.20 m	79.4 %
Curved road	0.24 m	63.0 %
Complex intersection	0.32 m	84.2 %
T - intersection	0.24 m	81.0 %
Roundabout	0.34 m	80.6 %

A quantitative evaluation of the detected curbs according to the methodology presented in chapter IV was performed. Specific scenarios like straight roads, curved roads, intersections and roundabouts are selected out of the whole drive sequence for individual testing. The performance results achieved in these scenarios are presented in Table I. These obtained results are however affected by the ego vehicle's position and orientation errors from the DGPS/IMU sensors.

In figure 5, we present some plotted curb points onto satellite images coresponding to several scenarios. The displacement errors seen in the last two images are due to the measurement errors from the DGPS/IMU sensors. The robustness of the detected curb points can be seen in figure 6, where such displacements are no longer present.

VI. CONCLUSION

In this paper we proposed a curb detection method which makes use of the fusion between the semantically labeled camera images and the 3D LiDAR point clouds. A novel outlier removal algorithm which uses local linear models for finding the final curb points was proposed. The curb points sparsity issue was addressed by temporally integrating the detected curb points from consecutive frames.

The evaluation is performed using drive recorded sequences. The detected curb points are matched against a high resolution digital map containing curb points. Although the results are affected by the sensor's calibration and synchronization errors and DGPS/IMU errors, the average error distance is between 0.2m and 0.34m and the precision is between 63% and 84.2%.

Future improvements will be focused on the identification of the error sources and diminish their influence in order to increase the precision of the proposed system.

ACKNOWLEDGMENT

Research was partially supported by the UP-Drive project (Automated Urban Parking and Driving), Horizon 2020 EU funded, Grant Agreement Number 688652 and also by the MULTISPECT grant (Multispectral environment perception by fusion of 2D and 3D sensorial data from the visible and infrared spectrum) of the Romanian National Authority for Scientific Research and Innovation / UEFISCDI, project code PN-III-P4-ID-PCE-2016-0727.

We would like to thank the UP-Drive project partners: Volkswagen Group for providing the vehicle infrastructure and data sequences and Technical University of Prague for providing the calibration of the sensors.

REFERENCES

[1] S. E. C. Goga, S. Nedevschi, "Fusing semantic labeled camera images and 3D LiDAR data for the detection of urban curbs," *IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 301-308, Cluj-Napoca, Romania, 2018.

[2] R. Varga, A. Costea, H. Florea, I. Giosan, S. Nedevschi, "Supersensor for 360-degree environment perception: Point cloud segmentation using image features," *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1-8, Yokohama, Japan, 2017.

[3] A. Costea, A. Petrovai, S. Nedevschi, "Fusion Scheme for Semantic and Instance-level Segmentation," *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, USA, 2018.

[4] Y. Zhang, J. Wang, X. Wang, J. Dolan, et al. "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor." IEEE Transactions on Intelligent Transportation Systems, vol. 99, 2018, pp. 1-11.

[5] R. Huang, J. Chen, J. Liu, L. Liu, B. Yu, and Y. Wu, "A Practical Point Cloud Based Road Curb Detection Method for Autonomous Vehicle", Information, vol. 8, no. 3, 2017, p. 93.

[6] M. Yadav, A. K. Singh, B. Lohani, "Extraction of road surface from mobile LiDAR data of complex road environment", International Journal of Remote Sensing, vol. 38(16), 2017, pp. 4645–4672.

[7] G. Zhao and J. Yuan, "Curb detection and tracking using 3d-lidar scanner", in 2012 19th IEEE International Conference on Image Processing. IEEE, 2012, pp. 437[1]

[8] Smadja L., Ninot J., and Gavrilovic T. "Road extraction and environment interpretation from LiDAR sensors". IAPRS, 38.1,2010, pp. 281-286.

[9] T. Chen, B. Dai, D. Liu, J. Song, Z. Liu, "Velodyne-based curb detection up to 50 meters away",Intelligent Vehicles Symposium (IV), 2015, pp. 241-248.

[10] J. Tan, J. Li, X. An, H. He, "Robust curb detection with fusion of 3d-lidar and camera data." Sensors, vol. 14(5), 2014, pp. 9046–9073.

[11] A.Y. Hata, S.O. Fernando, D.F. Wolf. "Robust curb detection and vehicle localization in urban environments." Intelligent vehicles symposium proceedings, 8-11 June, 2014, pp. 1257-1262.

[12] Z. Liu, J. Wang, D. Liu, "A new curb detection method for unmanned ground vehicles using 2D sequential laser data." Sensors, vol. 13(1), 2013, pp. 1102-1120.

[13] A. Hervieu and B. Soheilian, "Semi-automatic road/pavement modeling using mobile laser scanning", ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 2, pp. 31-36, 2013.

[14] Zhang Yihuan, et al. "A real-time curb detection and tracking method for UGVs by using a 3D-LIDAR sensor." Control Applications (CCA), 2015 IEEE Conference on. IEEE, 2015, pp. 1020-1025.

[15] Wang Liang, Yihuan Zhang, and Jun Wang. "Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR." IFAC-PapersOnLine,50(1), 2017, pp. 276-281.

[16] R. Miyazaki, M. Yamamoto, E. Hanamoto, H. Izumi, and K. Harada, \A line-based approach for precise extraction of road and curb region from mobile mapping data." ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, vol. 2, no. 5, 2014.

[17] Siegemund J., Franke U., and Förstner W. "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields". In Intelligent Vehicles Symposium (IV), 2011 IEEE, 2011, pp. 637-642.

[18] F. Oniga, S. Nedevschi, "Curb detection for driving assistance systems: A cubic spline-based approach", Intelligent Vehicles Symposium (IV), pp. 945 –950, 2011