Fusing semantic labeled camera images and 3D LiDAR data for the detection of urban curbs

Selma Evelyn Catalina Goga
Computer Science Department
Technical University of Cluj-Napoca
selma.goga@cs.utcluj.ro

Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca
sergiu.nedevschi@cs.utcluj.ro

***Abstract*—This article presents a new approach for detecting curbs in urban environments. It is based on the fusion between semantic labeled images obtained using a convolutional neural network and a LiDAR point cloud. Semantic information will be used in order to exploit context for the detection of urban curbs. Using only the semantic labels associated to 3D points, we will define a set of 3D ROIs in which curbs are most likely to reside, thus reducing the search space for a curb. A traditional curb detection method for the LiDAR sensor is next used to correct the previously obtained ROIs. For this, spatial features are computed and filtered in each ROI using the LiDAR's high accuracy measurements. The proposed solution works in real time and requires few parameters tuning. It proved independent on the type of the urban road, being capable of providing good curb detection results in straight, curved and intersection shaped roads.**

Keywords—curb detection; LiDAR; semantic information; deep learning; traditional method

I. INTRODUCTION

Curbs detection is a very important subject for autonomous driving tasks. It is a vital step for navigable road detection, trajectory planning and parking.

The detection of the curb delimiters has been studied for over a decade. The proposed solutions found in literature vary according to the sensors used. Common sensors used for detecting curbs are Cameras, LiDARs, Stereos and radars. Each sensor has limitations based on the type of information it provides. A camera offers RGB information but is sensitive to illumination variations, shadows and so on. LiDARs provide very accurate depth and reflectance information of the environment, making the detection of objects in 3D space possible with high precision. They are independent of the illumination variation. However, they return erroneous results once laser beams are reflected of wet or black surfaces. Stereo sensors provide both RGB and depth, but still have the limitation of the camera sensor and their provided depth information has low precision, just as in the case of radars. Among all the used sensors, Cameras and LiDARs gained the highest popularity, and a fusion between the two is in many cases preferred.

Traditional curb detection methods exist for both camera and LiDAR sensors. They usually imply finding and analyzing features or fitting a model onto data. The main advantage of such method is its fast processing speed. However, they require expensive manually parameter tuning and do not offer the greatest results in terms of accuracy and precision.

Deep learning semantic segmentation of car's camera images has become nowadays an almost mandatory task for autonomous driving perception module. This is because recent research has managed to optimize the testing stage of the semantic segmentation network, making it run in real time and obtain outstanding results [1]. Camera images semantic segmentation do provide a good classification but they are not capable of offering a precise position of an object in 3D space. Also, erroneous segmentation still exists and a perfect objects classification is for now impossible to obtain.

The proposed solution tries to overcome the problems stated earlier by using a fused sensors approach which combines both traditional curb detection techniques and deep learning techniques. For this, we propose fusing semantic segmented information obtained from cameras with LiDAR data. The points which were labeled as curb or are found to be road delimiters will be considered as ROIs were curbs are likely to be found. Curb ROIs are defined per each scanline of a LiDAR. A traditional curb detection technique for LiDAR sensor is applied for each ROI and a correction of the curbs proposals is thus made. In the end, a polyline representation will be used to output the final found curbs.

The main contributions of the proposed algorithms are:

- The usage of semantic information for obtaining ROI of curb.
- The extraction of the curbs using traditional LiDAR methods is done only inside the previously found ROIs, thus increasing the time performance.
- Algorithm is flexible: behaves well in many types of urban road scenarios: straight roads, curb roads, fork roads, intersections.

II. RELATED WORK

Traditional camera based methods for the boundary of the road detection use the vanishing point as reference [7], [3]. In [7] authors considered the edges which converge to vanishing point

978-1-5386-8445-0/18/ $31.00 © 2018 IEEE

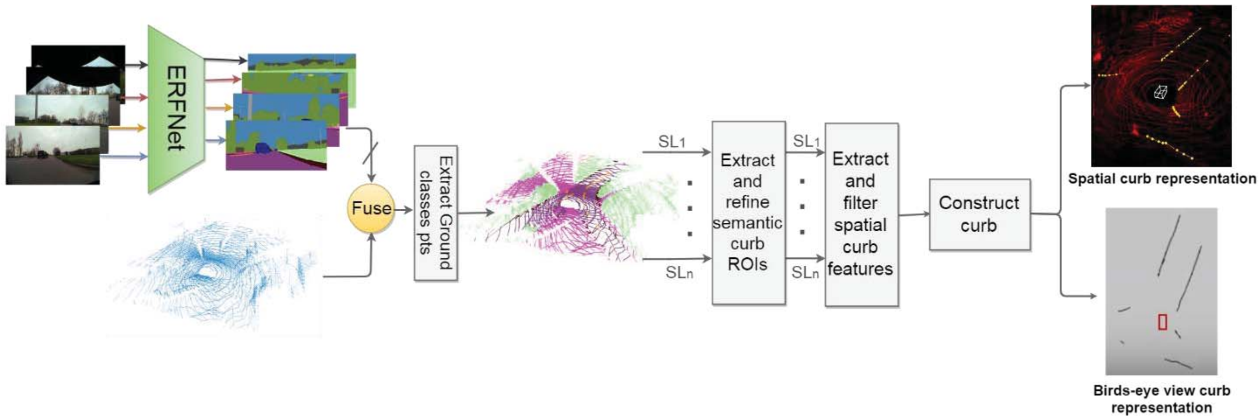

Fig. 1. The processing pipeline of the proposed system.

as potential lanes or curbs edges found in urban scenarios. Authors in [3] also included the color distribution in order to detect the navigable road regions contoured by any type of road delimiter.

In [24] authors proposed a digital elevation map constructed onto stereo data. They found potential curb cells applying a Canny edge detector to the map and fitted with RANSAC a 3rd degree polynomial to data the curb cells. Same authors in [23] use a spline interpolation method to represent curb segments which appeared to increase curb detection accuracy. Although in the final fitting stage many outlier points were filtered, the usage of camera sensors for road boundary detection still has limitations.

More accurate results for road boundary detection have been obtained using a 2D or a 3D LiDAR sensor. The difference between a 2D and a 3D LiDAR sensor is that 2D sensors have only one laser scan line and a limited FoV, while a 3D LiDAR has a FoV of 360 degrees and more than one scan line. 2D LiDAR are sometimes preferred because they are less expensive than a 3D one. In [13], authors used two 2D Lidar sensor which had the point cloud data projected onto the vehicle's front ground plane in order to detect curbstones and lanes. They applied a sliding window technique from which a convex angle detection criterion was used to select the curb edges. In the end a path planning is made on local lane fitting and prediction. Although 2D Lidar methods are fast, they are hard to use because of their need to perform temporal fusion between consecutive frames in order to obtain a complete representation of the environment. To cope with this problem, 3D Lidar sensors are preferred for autonomous driving tasks.

An approach for detecting curbs from 3D LiDAR data includes representing the data under the form of a grid map (as in [4], [6], [10], [19], [7], [22] and [8]). The spatial features used in order to find curb proposals are elevation difference (as in [19], [10], [6] and [7]), curvature (as in [4], [8], [15]), density (as in [4] and [6]), slope [19]. Temporal persistency is also taken into consideration for further curb cells filtering in [4] and [19]. In the end, potential curb segments are generated by fitting a curve model onto them. Authors in [19] used a multi-model RANSAC in order to find the best fitting polynomial to the proposed curb points. In [10], authors fitted 4th order Splines to curb points in order to obtain the final curbs representation. Curb detection using the grid based representation method however does not offer the same accuracy for sparse LiDAR point clouds as in the case of a scan line based method. To address the sparsity problem authors in [8] proposed combining stereo sensor's dense non-accurate data with laser scanners sparse accurate data. Another problem when using a grid based approach is the possibility of the presence of multiple objects at different elevations in the same curb cell. A previous solution for addressing this problem was proposed in [12] which implied using a modified grid representation called a multi-volumeric grid structure. The structure was capable of capturing multiple objects intervals found at different elevations in the same cell.

In [2], [21], [20], [14], [16] and [11] authors detected urban road boundaries using a scan line technique. At basis, scan-line based methods for boundary extraction include sliding window sweeping of each LiDAR ring and the extraction and filtering of each spatial cue. In the end, the temporal persistency of curbs is used and a final curve is fitted onto data. The most used spatial feature is the elevation difference (as in [2], [21], [11] and [20]), followed by the normal orientation [14], slope [20] or angle [2]. These features however do not provide context information which can decrease the false positive rate from the proposed potential curbs. Many of these false positives come from the vehicles wheel tires or even stairways which have similar aspect to that of a curb. Adding temporal persistency is a viable solution [18] which solves part of the problem but is inefficient for forked roads. A better solution to these problems would be using context information as [2] stated. A possible use of context information for curbstones using only LiDAR data is checking the height smoothness of a region of points around the proposed curb edges as in [11]. This method filters out part of the false positives but will increase the number of false negatives as in regions where the curb is followed by a grass area this statement will no longer be valid.

In order to add context information, an enrichment of the LiDARs data is obtained by fusing it with camera information as in [17], [5], [26]. In [5], authors used a 2D Lidar combined with a camera and proposed a curb detection and tracking algorithm. Authors in [17], used the normal map of the scene surfaces created by combining image features with 3D LiDAR depth information. In the end, in [17], dynamic programming was used in order to link curb points and to obtain the final curb's position proposals. More recently, in [26] an approach for road detection was proposed by combining road proposals from preprocessed camera images using a DNN and from a scan-line-based Lidar road detection. A high level fusion was done in the end using a CRF.

III. PROPOSED SOLUTION

We propose a top down approach for detecting the curb edge points. The stages of our method can be seen in figure 1, where *SL* stands for Scan Line and *n* is the maximum number of scan lines coming from one or more LiDAR sensors. First, a multi-camera semantic segmentation is performed using an ERFNet[1] convolutional network. The resulted images are fused with LiDAR's point cloud in order to assign a semantic class to the 3D points. This association will reduce the search area for a potential curb in the 3D space. Next, curb ROIs are selected and refined in each LiDAR scan line using only the semantic information from images. We search for the potential curb edge points inside each ROI. Using 3D features computed from the high accuracy LiDAR's measurements, we are able to find the final curb proposals and to increase the precision of the semantic curb detection. A polyline representation is used in the end to represent the curb's lower and upper edges.

The proposed method is split in five main steps:

A) *3D points semantic enhancement* - where the low level fusion between LiDAR data and semantic images is performed.

B) *Curb ROIs selection and extraction* – in which curb ROIs are defined based on the semantic labels of the points on each LiDAR scan line.

C) Curb ROIs refinement – where each ROI is expanded in order to increase the chance of enclosing inside it a curb's upper and lower edge points.

D) Spatial features extraction and filtering – in which the final curb edge points for each ROI are identified.

E) Curb reconstruction – where the final polyline representation is built.

A. 3D points semantic enhancement

A fisheye multi-camera network capturing the entire surrounding view of the environment is available. The entire process of obtaining the semantically enhanced 3D point cloud is described in more detail in [9].

We used ERFNet[1] to segment the images coming from the intelligent vehicle's cameras. ERFNet is an efficient semantic segmentation architecture specially designed for autonomous driving tasks. It runs in real-time and offers high quality output.

For the training stage, the network was fed with annotated images from the car's cameras which captured various urban scenarios. The color coding of the semantic classes matches the one from the Cityscapes dataset [25] to which we added two new classes: CURB class, LANE_MARKINGS class. In the end, two basic class categories can be depicted: GROUND class category (i.e. ROAD, TERRAIN, SIDEWALK, GROUND, PARKING, LANE_MARKINGS, CURBS) and OBJECT class category (i.e. the remaining semantic classes).

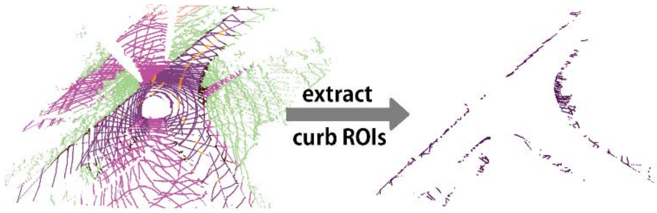

Fig. 2. Obtaining the unrefined curb ROIs.

After obtaining the semantic images for a frame in the testing stage, we fuse them with the 3D point cloud. For this, we project the LiDARs point cloud onto them. The color of the semantic image pixel a 3D point has fallen onto will give its corresponding semantic class. Because the field of view from the car's cameras was overlapping at sides, LiDAR points might receive in some cases two distinct semantic classes. In order to solve this issue, we chose the class with the maximum number of appearances from the images neighborhoods.

Because the LiDAR might perceive objects which are not visible from the camera's view, false associations might appear. An occlusion handling is performed to solve this problem and points which belong to an occluded object will not be labeled.

B. Curb ROIs selection and extraction

Using only the 3D points semantic information we are able to extract regions of interest from the point cloud where real curbstones are very likely to reside. The result of this step is illustrated in figure 2.

A point cloud is a set of 3D points denoted by *P*. A scan line is an ordered subset of points from *P* denoted with SL_i, where *i* represents the number of a LiDAR sensor's layer such that $i = [\ 1,\ maxNrLayers\ ]$. The order is given by the column number of the points inside it. A point from SL_i is denoted with $sl\_p_{i,j}$, where $j = [1, size(SL_i)]$ denotes the order of the point in the layer.

A ROI is defined as a pair of points from a SL_i and is denoted with $R_{i,k} = (R_{in}^{i,k}, R_{out}^{i,k})$, where *i* is the index of the scan line the ROI is found on and *k* is the index of the ROI found on that scan line SL_i. R represents the set of all $R_{i,k}$. The following equations summarize the previously explained concepts:

$$R_{i,k} \subset SL_i \subset P \quad (1)$$

$$\cup_{i=1}^{maxNrLayers} SL_i \equiv P \quad (2)$$

Each ROI point represents one of the endpoints of a consecutive region of 3D points. $R_{in}^{i,k}$ expresses the endpoint found near the ROAD side and $R_{out}^{i,k}$ is the endpoint found in the neighborhood of a SIDEWALK or TERRAIN class. In order to find a consecutive region and select a ROI $R_{i,k}$ we sweep each LiDAR scan line SL_i in an ordered manner and we analyze the semantic labels of the points encountered. There are two cases which can be encountered when selecting a curb ROI:

1. when finding a region composed of consecutive points labeled as CURB class. Such a region is valid only if it has at least 2 points.
2. when a semantic class transition between a ROAD labeled region and a TERRAIN or SIDEWALK labeled region is present.

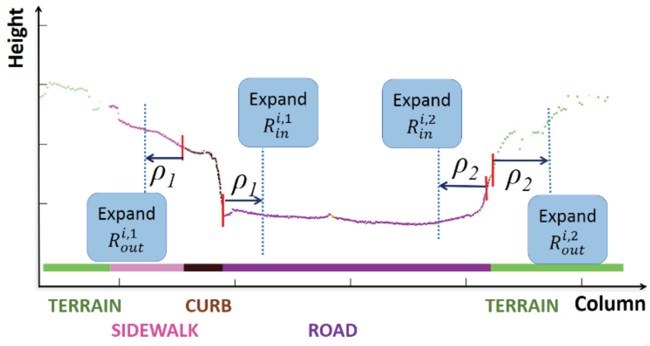

Fig. 3. Expanding curb ROIs for a LiDAR scan line.

The first case corresponds to the most frequent method of extracting a ROI using semantic information. A ROI *R* of a sequence of CURB labeled points from SL_i, scan line will be constructed as follows:

- $R_{in}^{i,k}$ will be set to the point $sl\_p_{i,j}$ when $sl\_p_{i,j}.label$ = *CURB class* and either $sl\_p_{i,j+1}.label$ = *ROAD class* or $sl\_p_{i,j-1}.label$ = *ROAD*
- $R_{out}^{i,k}$ will be set to the point $sl\_p_{i,j}$ when $sl\_p_{i,j}$ *.label = CURB class* and either $sl\_p_{i,j+1}.label$ = *SIDEWALK | TERRAIN class* or $sl\_p_{i,j-1}.label$ = *SIDEWALK | TERRAIN class*.

Notice that we restricted the ROI for the detection of curbs by forcing it to be around a ROAD and a SIDEWALK or TERRAIN labeled neighborhoods. This is because the output of the semantic segmentation may include false positives or even other curbstones which do not delimit the road area.

The second case for curb ROI selection appeared as a necessity because small or far distance curbs are sometimes omitted by the semantic segmentation while the LiDAR can still detect them accurately. For this case, the presence of a potential curb is noticed as a transition between a ROAD labeled region and a SIDEWALK or TERRAIN labeled region. Given a scan line SL_i such a transition is identified when: $sl\_p_{i,j}.label$ = *ROAD class* and $sl\_p_{i,j-1}.label$ or $sl\_p_{i,j+1}.label$ has *SIDEWALK* or *TERRAIN class* assigned to. In this case, a ROI $R_{i,k}$ will be built as follows:

- $R_{in}^{i,k}$ will be set to the point $sl\_p_{i,j}$ from the scan line SL_i, where $sl\_p_{i,j}$ = *ROAD class.*
- $R_{out}^{i,k}$ will be set to the point $sl\_p_{i,j-1}$ or $sl\_p_{i,j+1}$ the scan line SL_i which has either a TERRAIN or SIDEWALK label assigned to.

C. *Curb ROIs refinement*

Because the search for a curb's edge points will be done only inside the regions of interest, a ROI should contain sufficient details in order to capture the entire structure of a curb. The lower and upper edges of the curb along with a part from its neighboring regions should be present inside the ROI for a precise curb detection.

We noticed that the CURB class semantic information assigned to 3D points lacks precision and a ROI may not capture the entire information needed for further processing. Also, all the ROIs detected in the transition ROAD-SIDEWALK or TERRAIN case will have only two points which are not capable of encapsulating all the curb structure as it can be seen in figure 3. In order to solve this issue, we expand the previously detected ROIs. For this, we use two window of size ρ_1 *(used to expand ROIs found using the first previously mentioned case)* and ρ_2 (*used to expand ROIs found using the second previously mentioned case*) expressed in meters. This is because the region of points covered by the ROIs which were created based using the transition criterion have only two points inside them which cover a very small 3D space area and a greater expansion is needed for them. In general, for a ROI $R_{i,k}$, $R_{in}^{i,k}$ will be expanded with size ρ_1 or ρ_2 towards the ROAD side and $R_{out}^{i,k}$ will be expanded with size ρ_1 or ρ_2 towards the TERRAIN or SIDEWALK side.

The refinement of the curb ROI proposals is necessary for obtaining the position of the curb in 3D space with higher accuracy. Figure 3 illustrates this process. The red lines delimit the primary unrefined curb ROIs and the dotted blue lines show the new ROI limits defined after the expansion with an assumed fix window size. The horizontal line above the Column axis is the projection of the points semantic labels onto the Column axis. $R_{in}^{i,k}$'s will move towards the ROAD labeled region while $R_{out}^{i,k}$'s will move towards the TERRAIN or SIDEWALK labeled region.

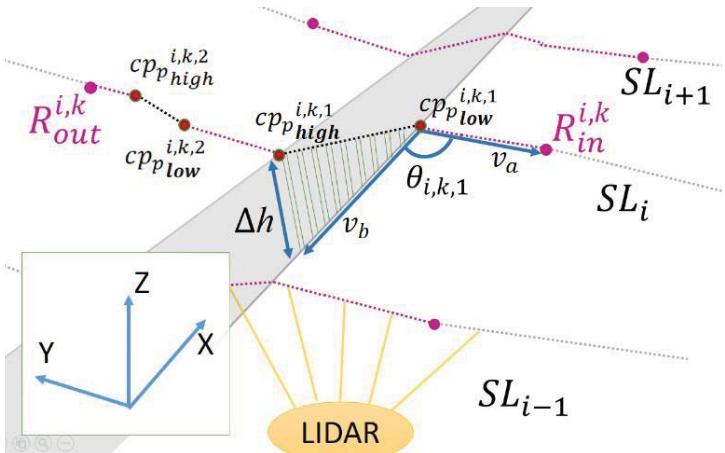

Fig 4. Candidate curb transversal regions illustration extracted using spatial features.

D. Spatial features extraction and filtering

Inside each previously expanded ROI we next search for the curb's lower and upper edge points.

We define a curb as a monotonically ascending region of points which has a standard height interval (represented by the elevation difference feature) and a vertical structure relative to the ground plane (represented by the angle feature). To encapsulate this definitions, we first search for monotonically ascending point regions inside each ROI $R_{i,k}$. A sweep starting from $R_{in}^{i,k}$ is performed. A region where the first derivative of the points height maintains positive will be considered as a potential curb's transversal region. We denote with $CP_{i,k,m}$ the monotonically ascending points region, where i represents the index of the scan line SL_i, k is the index of the ROI found on that scan line SL_i, and m the index of the CP such that $m = [1, maxNrCPsInAROI]$. We denote $cp\_p_{low}^{i,k,m}$ as the point with the lowest elevation from the monotonically ascending region and $cp\_p_{high}^{i,k,m}$ as the highest elevated point, such that:

$$\{cp\_p_{low}^{i,k,m},\ cp\_p_{high}^{i,k,m}\} \subseteq CP_{i,k,m} \subset R_{i,k} \qquad (3)$$

$cp\_p_{low}^{i,k,m}$ corresponds to the potential curbs' low edge point while $cp\_p_{high}^{i,k,m}$ to a curb's high edge point (see Figure 4).

For each $CP_{i,k,m}$ we compute and filter two features which were described in the previously stated curb definition:

a. The elevation difference

In order to find the height Δh of the potential curb, the absolute difference between the endpoints $cp\_p_{low}^{i,k,m}$ and $cp\_p_{high}^{i,k,m}$ is computed. The resulted value should be in a predefined curb height range as shown in formula (4). The range is a fixed interval $[\Delta H_{min}, \Delta H_{max}]$ representing the typical height of an urban curb. ΔH_{min} refers to the minimum height of a small urban curb. ΔH_{max} is considered to be the maximum elevation of a regular sized curb encountered in an urban environment.

The filtering of Δh is done as:

$$\Delta H_{min} \leq abs(cp\_p_{high}^{i,k,m}.z - cp\_p_{low}^{i,k,m}.z) \leq \Delta H_{max} \quad (4)$$

b. The angle

Because of the way a LiDAR scan line is projected onto a curb region, the vertical structure of a curb can be considered by looking from a bird's eye view perspective. In [20] the authors described $\theta_{i,k,m}$ as the angle between two vectors originating from the same point. The angle was used to search for the lower curb edge point. By sweeping a laser scan line and computing the $\theta_{i,k,m}$ for each point encountered, the lower curb edge point was found checking the condition $\theta_{i,k,m} < \theta_{TH}$, where θ_{TH} is a threshold expressed in degrees. In this article, this feature will be computed only for the lowest endpoint $cp\_p_{low}^{i,k,m}$ from each candidate curb region $CP_{i,k,m}$. The formulas from [20] are used and adapted for our case:

$$\theta_{i,k,m} = \cos^{-1}\frac{v_a \cdot v_b}{|v_a| \cdot |v_b|}$$

$$v_a = [R_{in}^{i,k}.x - cp_{p_{low}}^{i,k,m}.x, R_{in}^{i,k}.y - cp_{p_{low}}^{i,k,m}.y] \qquad (5)$$

$$v_b = [cp\_p_{high}^{i,k,m}.x - cp_{p_{low}}^{i,k,m}.x, cp_{p_{high}}^{i,k,m}.y - cp_{p_{low}}^{i,k,m}.y]$$

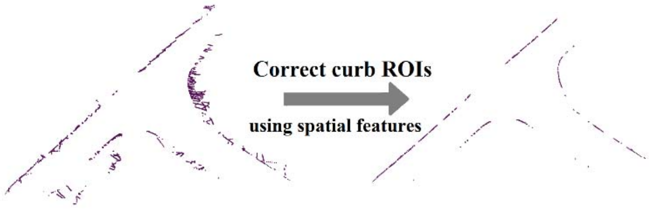

Fig 5. Obtaining the final curb points.

Both tests should be passed in order for a potential curb's transversal region $CP_{i,k,m}$ to become a valid curb. In the case more than one $CP_{i,k,m}$ satisfy the previous conditions, the candidate curb region found closest to $R_{in}^{i,k}$ is selected in the end as final curb region. By using these simple heuristics on each candidate set of points we are capable of defining the final curb edges points. A result after performing step D can be seen in figure 5.

When we find the final valid $CP_{i,k,m}$ for a ROI $R_{i,k}$, the endpoints $R_{in}^{i,k}$ and $R_{out}^{i,k}$ of $R_{i,k}$ will migrate towards the low and high endpoints of $CP_{i,k,m}$. This is expressed as: $R_{in}^{i,k} \leftarrow cp\_p_{low}^{i,k,m}$ and $R_{out}^{i,k} \leftarrow cp\_p_{high}^{i,k,m}$.

There are cases when a set $CP_{i,k}$ for a ROI $R_{i,k}$ might be empty or when no $CP_{i,k,m}$ has passed the two features tests. These cases appear mostly when a false positive curb is encountered in the semantic image. Using the values of the spatial features computed from the high accuracy LiDAR measurements for each $CP_{i,k,m}$, we are capable of eliminating most of the false positives, thus correcting the camera semantic information. In the case of no valid $CP_{i,k}$ element, the region $R_{i,k}$ will be discarded.

E. Curb reconstruction

Given a set R of previously found pairs of curb lower and upper edges, we are interested in connecting its components based on the distance between them. For this we try to construct a graphs forest F such that a graph G, $G \subset F$, represents an entire curb segment. A node of G is represented by an element from the set R. An edge from G is a connection between two nodes which are most likely from the same curb segment.

A graph G must have at least one two points connected with an edge in order to be considered a real curb segment. For each curb region node, we search for two nearest nodes which are found in opposite directions. Consider $R_{i,1}$ and $R_{i,3}$ two close neighbors of the node $R_{i,2}$, where either $R_{i,1}$ or $R_{i,3}$ is the closest neighbor of $R_{i,2}$. v_1 and v_3 are two vectors originating both in $R_{i,2}$ and constructed as follows:

$$v_1 = [R_{i,1}.x - R_{i,2}.x\ , R_{i,1}.y - R_{i,2}.y] \qquad (6)$$

$$v_3 = [R_{i,1}.x - R_{i,3}.x\ , R_{i,1}.y - R_{i,3}.y]$$

In order to consider edges between $R_{i,1}R_{i,2}$ and $R_{i,2}R_{i,3}$, the following conditions should be met:

$$v_1 \cdot v_3 < 0 \qquad (7)$$

$$||R_{i,1} - R_{i,2}||_2 < \delta(x) \quad and \quad ||R_{i,3} - R_{i,2}||_2 < \delta(x) \quad (8)$$

$\delta(x)$ is a threshold function which makes sure the neighbors of the middle point $R_{i,2}$ are in a valid range. X is the range of the point $R_{i,2}$ computed as the Euclidean distance between the sensor and $R_{i,2}$. $\delta(x)$ was computed from observations taking into account the distance between consecutive scan lines at various ranges:

$$\delta(x) = \begin{cases} 2\ , x < 7 \\ 0.1 \cdot x + 2, x \geq 7 \end{cases} \qquad (9)$$

The output of $\delta(x)$ is expressed in meters.
A polyline is drawn for each edge of the graphs G.

IV. EXPERIMENT RESULTS

For evaluating the feasibility of the proposed, we ran it on a video sequence with real urban scenarios which contained 3037 frames. The video was captured using an intelligent vehicle, who's sensors infrastructure is presented in subsection A.
Because the sparsity between points from the same scan line for a LiDAR sensor is dependent on the distance from those points to the sensor, we have limited the curb detection range to 60m front, 40 m rear and 30m left and right. In our experiments we set the parameter thresholds to: ρ_1 =0.7 m, ρ_2 = 1.5 m, ΔH_{min} = 4 cm, ΔH_{max} = 25 cm, θ_{TH} = 150°.
The shape of a road is identified from a bird's eye view perspective and can be either: straight, curved, intersection.

A. Vehicle sensors infrastructure

The video was acquired by driving an intelligent vehicle in an urban European environment and by recording the data received from the following car sensors:

- a multi-camera network consisting of four fish-eye cameras which capture the surrounding view of the vehicle
- five 3D Lidar sensors, three of them having 32 layers each (32 scan lines) and the rest of them, 16 layers each (16 scan lines)

The position of the sensors onto the test vehicle can be seen in figure 5.

	Straight Road	Curved Road	Intersected Road
PPV	87.53%	86.22%	81.31%
TPR	71.19%	63.09%	50.98%

Table 1. Curb detection results evaluation for different types of urban road scenarios.

B. Runtime evaluation

The proposed algorithm was written and optimized in C++.
Each video frame was processed on a normal PC which had an *Intel i7-3770K CPU* with a frequency of 3.50GHZ processor.
The semantic algorithm runtime was almost 12ms per camera image. The fusion between the LiDAR data and the sensor was $\approx 2\ ms$ for a 16 layered LiDAR sensor.
The average processing time for a frame using our curb detection method(which starts from the *Extract Ground Classes pts* (see Figure 1) pipeline stage) was $\approx 4\ ms$.

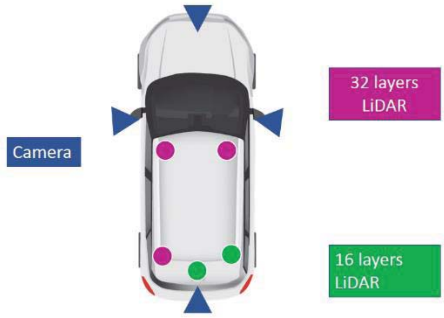

Fig 5. Vehicle's sensors infrastructure.

C. Precision and recall evaluation

The testing video contained several intersections and many straight and curved roads.
In order to evaluate the systems performance, we compute the following coefficients:

- *The positive predictive value (PPV)* – also called the *precision* - represents the proportion of curbs detected correctly from all the curbs detected using the proposed solution in one frame and is expressed as:

$$PPV = \frac{TP}{TP + FP}$$

- *The true positive rate (TPR)* – also called the *recall* - represents the proportion of curbs detected correctly from all curbs which have been or should have been detected in one frame by the system and is expressed as:

$$TPR = \frac{TP}{TP + FN}$$

TP represents the True Positive number which tells us the number of correctly detected curb regions. FP represents the False Positive Number which indicates the number of wrongly

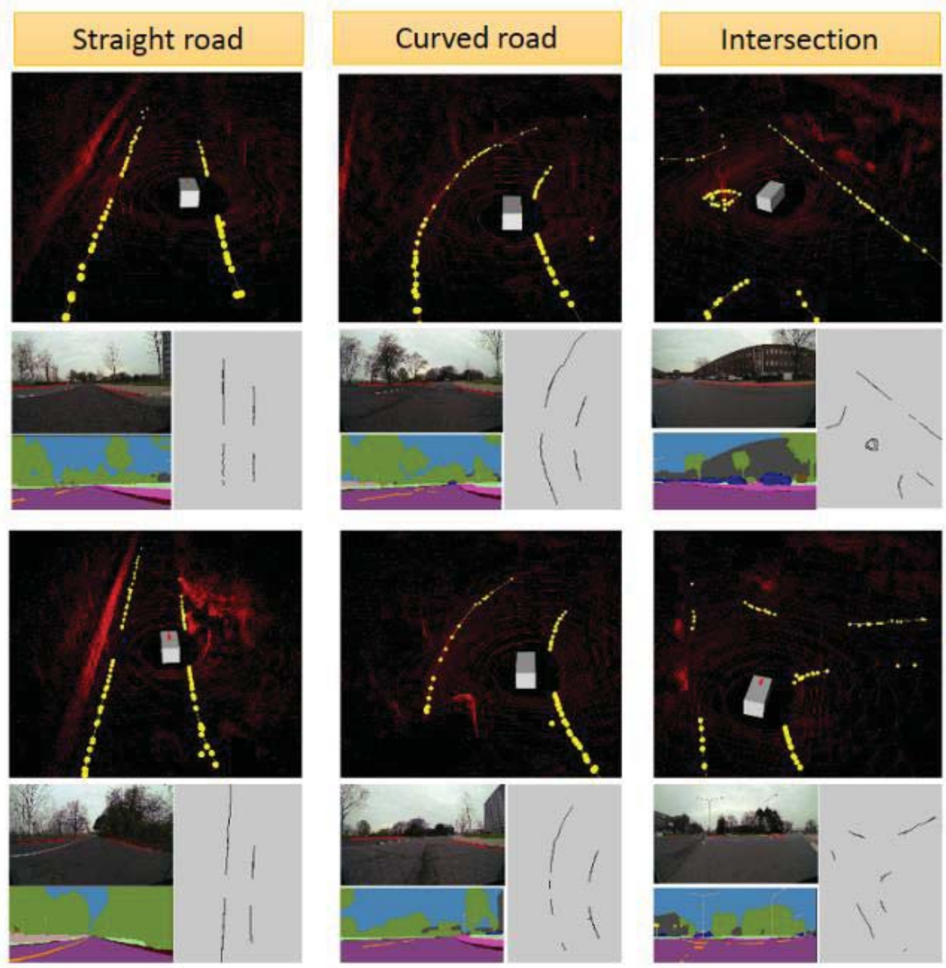

Fig. 6. Results of the proposed method for different types of urban road scenarios.

detected curb regions. FN is the False negative number which identifies the number of true curbs which have been omitted from the detection. The results for computing these metrics for each curb category encountered in the video sequence can be seen in Table 1.

The precision ratios were over 80% for all road shapes. In order to increase this coefficient, the number of False Positives should be reduced as much as possible. We noticed that sometimes the semantic information offers erroneous curb classifications of regions and if a structure resembling to that of a curb is still present, the used spatial features are not able to filter it out (e.g. railways). A solution would be to increase the accuracy of the semantic segmentation. Another would be to add extra features (e.g. curb continuity constraint, temporal persistency) in order to make the spatial filtering more assertive.

The recall ratios for the road shapes gave lower results compared to the precision coefficient value. This is because curb points found at a distance farther than 30m, tend to be omitted because of the sparsity of the LiDARs sparse data. Solutions would be to either increase the expansion threshold or to increase the precision of the semantic segmentation.

Figure 6 captures the practical results of the proposed methods for various road types. Two examples for each road category is presented in each column. In the 3D space, where the LiDAR point cloud is represented, the points detected and the polylines between them are drawn with yellow on the black background. The car is visible as a 3D white box. We associated to each 3D scene the front camera images which had the curb polylines projected with red. Under each front camera image the result of the semantic segmentation is seen. Next to the two images, a bird's eye view showing the curb segments is presented for a better understanding of the scene.

V. CONCLUSIONS

In this article, a new approach for detecting urban curbs was proposed. The main contribution of the article is the usage of semantic information from images to provide context in order to reduce the search space for 3D curb proposals. The semantic labels were obtained applying a deep learning approach to cameras images while the 3D features were captured using traditional curb detection methods for a LiDAR sensor. By combining the high accuracy of the LiDAR and the camera's semantic information, the proposed algorithm was able to reduce the search in the 3D space for curbs detection and to offer a more precise classification of curb region.

Possible refinements can still be made in order to increase its performance. Future improvements include adding a curb continuity constraint between consecutive LiDAR scan lines or making use of the temporal persistency of curbs in consecutive frames. Other improvements include creating adaptive parameters in order to escape current parameter tuning.

ACKNOWLEDGEMENTS

This work has been supported by the UP-Drive project (Automated Urban Parking and Driving), Horizon 2020 EU funded, Grant Agreement Number 688652.

REFERENCES

[1] E. Romera, J.M. Alarez, L.M. Bergasa, R.Arroyo, "Efficient convnet for real-time semantic segmentation." Intelligent Vehicles Symposium (IV), pp. 1789-1794, June 11-14 2017.

[2] 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, pp. 1-11, 2018.

[3] N. John, B. Anusha, K. Kutty, "A reliable method for detecting road regions from a single image based on color distribution and vanishing point location.", Procedia Computer Science, vol. 58, pp. 2-9, 2015.

[4] 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, p. 93, 2017.

[5] 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), pp. 4645–4672, 2017.

[6] B. Rodriguez-Cuenca, S. Garcia-Cortes, C. Ordonez, and M. A. Alonso, "Morphological operations to extract urban curbs in 3D MLS point clouds." ISPRS International Journal of GeoInformation, vol. 5, no. 6, p. 93, 2016.

[7] N. Simond, P. Rives, "Homography from a vanishing point in urban scenes." in Proceedings Intelligent Robots and Systems, 2003.

[8] C. Fernandez, D. Llorca, C. Stiller, M. Sotelo, "Curvature-based curb detection method in urban environments using stereo and laser." Intelligent Vehicles Symposium (IV), pages 579–584, 2015.

[9] R. Varga, A. Costea, H. Florea, I. Giosan, S. Nedevschi, “Supersensor for 360-degree environment perception: Point cloud segmentation using image features,” in IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017.

[10] A. Miraliakbari, M. Hahn, S. Sok. "Automatic extraction of road surface and curbstone edges from mobile laser scanning data." The International Archives of Photogrammetry, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-4/W5, pp.119-124, 2015.

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

[12] S. Goga, S. Nedevschi. "An approach for segmenting 3D LiDAR data using multi-volume grid structures", 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2017.

[13] S. Zhiwei, H. Weiwei, W. Ning, W. Xiaojun, W.C.Y. Anthony, V.B. Saputra, B.C.H. Quan, C.J. Simon, Z. Qun, Y. Susu, H.B. Siew, "Map free lane following based on low-cost laser scanner for near future autonomous service vehicle." Intelligent Vehicles Symposium (IV), pp. 706 - 711, 2015.

[14] R. Miyazaki, M. Yamamoto, E. Hanamoto, H. Izumi, K. Harada,"A line-based approach for precise extraction of road and curb region from mobile mapping data." ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 2.5, pp 243, 2014.

[15] C. Fernandez, R. Izquierdo, D. Llorca, M. Sotelo "Road curb and lanes detection for autonomous driving on urban scenarios ” in Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, Oct 2014, pp. 1964–1969

[16] 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.

[17] J. Tan, J. Li, X. An, H. He, "Robust curb detection with fusion of 3d-lidar and camera data." Sensors, vol. 14.5, 2014.

[18] A. Hervieu, B. Soheilian. "Semi-automatic road/pavement modeling using mobile laser scanning." ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol 2 -3/W3, pp. 31-36, 2013.

[19] Z. Liu, J. Wang, D. Liu, "A new curb detection method for unmanned ground vehicles using 2D sequential laser data." Sensors, vol. 13.1, pp. 1102-1120, 2013.

[20] B. Yang, L. Fang, J. Li. "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds", ISPRS Journal of Photogrammetry and Remote Sensing, vol. 79, pp. 80-93, 2013.

[21] W. Yao, Z. Deng, L. Zhou. "Road curb detection using 3D lidar and integral laser points for intelligent vehicles”, In Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), pp. 100-105, 2012.

[22] G. Zhao, J. Yuan, "Curb detection and tracking using 3D-LIDAR scanner", 19th IEEE International Conference on Image Processing (ICIP), pp. 437–440, 2012.

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

[24] F. Oniga, S. Nedevschi, “Polynomial curb detection based on dense stereovision for driving assistance”, in 13th International Conference on Intelligent Transportation Systems (ITSC), pp. 1110–1115, 2010.

[25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, "The cityscapes dataset for semantic urban scene understanding", In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[26] S. Gu, T. Lu, Y. Zhang, J. M. Alvarez, J. Yang, H. Kong, “3D lidar + monocular camera: an inverse-depth induced fusion framework for urban road detection”, IEEE Transactions on Intelligent Vehicles, 2018.