Curb Detection for Driving Assistance Systems: A Cubic Spline-Based Approach

Florin Oniga and Sergiu Nedevschi

Abstract—In this paper we present a real-time algorithm that detects curbs using a more general cubic spline model. A Digital Elevation Map (DEM) is used to represent the dense stereovision data. Curb measurements (cells) are detected on the current frame DEM. In order to compensate the small number of curb measurements for each frame we perform temporal integration. The result is a rich set of curb measurements that provides a good support for the least square cubic spline fitting. Thus, the curb cubic spline approximation is more stable and available on a much larger area, around the ego car. This compensates the limited field of view of typical stereo sensors. The detected curbs enrich the description of the ego car’s surrounding 3D environment and can be used for driving assistance applications.

I. INTRODUCTION

CURBS are delimiters of the drivable area that are usually not detected by standard obstacle detection algorithms and require dedicated algorithms for detection. Curb detection cannot be ignored in driving assistance applications: they must be taken into account for navigation, together with relevant obstacles.

The problem of curb detection has gained a certain interest in the recent years. Existing approaches for curb detection can be classified considering various criteria, the most relevant ones being the type of sensor used and the curb model.

Various sensors, passive or active, are used for data acquisition in the curb detection applications: video sensors (monocular or stereo), time-of-flight cameras (photonic mixer devices), laser line stripers, lidar sensors, or a mixture of these sensors.

Video sensors are one of the preferred sensors for the task of curb detection due to the richness of the data provided, whether they provide intensity (or color) data with monocular cameras, or intensity/color and depth data with stereovision cameras. Monocular cameras are usually combined with other sensors that provide range information, mainly lidar or laser line stripers [6], [11]. Stereo cameras provide both intensity and range information, and can be used as stand-alone sensors for curb detection [1], [2], [3], [4], [13], [14].

PMD cameras (Photonic-Mixer-Device or Time-Of-Flight camera) are used as stand-alone sensors for curb detection [9], providing range information for each pixel. Laser line stripers are composed of a video camera and a laser and are used for curb detection, stand-alone [10] or in combination with other sensors [11]. A light plane is generated with a cylindrical lens and intersects the 3D environment. The 3D locations associated with the bright pixels can be recovered by triangulation.

Lidars, or laser scanner, are active sensors that are based on the time-of-flight principle. There are several lidar sensors available on the market. An omnidirectional lidar with 64 beams is used for curb detection in [8]. Lidars with a limited field of view (less than 360 degrees) and fewer beams (up to four) are either used for curb detection as stand-alone sensors [5], [7], or combined with vision sensors [6].

Two types of curb models are encountered in the literature: parametric models and non-parametric models. Parametric models are represented mainly by linear/poly-linear curb models or, more recently, by cubic polynomial models. The linear curb model, regardless of the sensorial system, is the most frequently employed for modeling curbs [1], [5], [2], and [3]. A poly-linear (or linear spline) model is used in [4]. The cubic polynomial model was recently proposed in [13] and [14]. Non-parametric curb models are either free-form [8], or local linear models [10], [11].

In the next paragraphs we will briefly discuss the existing approaches that are based on stereovision.

One of the earliest approaches for curb detection is [1], based on stereovision. Candidate curbs are extracted as clusters of at least three parallel lines in the image space, by applying the Hough transform. Two planes are computed on the 3D data of the surrounding regions, and the height of the curb is estimated.

The approach presented in [2] is based on both range and photometry data. Image edge points are detected and the Hough accumulator is built. The scalar product of the elevation and brightness gradients of each edge point is used for voting in the Hough accumulator. One dominant straight curb (per scene) is extracted as the line with the maximum score in the accumulator. This method is improved later in [3]. The Hough accumulator is built from edge segments close to the ground plane, and the vote is weighted with the curvature index (based on the principal curvatures). Peaks of the Hough transform are considered candidate curb lines,
which are further refined: regression is performed to the 3D points of the candidate line, and the ends of the curb are searched.

The 3D points, from dense stereo, are transformed into a Digital Elevation Map (DEM) in [4]. Edges are detected on the DEM. Candidate curb points are those edges that are persistent along successive frames. The Hough accumulator is built from the curb points. Curb segments are located along linear curbs (Hough peaks): the largest segment with specific height variation (orthogonal to the curb) is selected for each curb line. A scheme is proposed to extend iteratively each curb segment with additional segments, in order to provide a better representation for curved curbs (as linear splines).

The problem of curb detection is solved using a global minimization approach in [13]. The dense stereo data are organized into a cartesian (image space) DEM. The world is modeled as consisting of a quadratic road surface and two adjacent (on the left and on the right sides) quadratic surfaces for the sidewalk. The boundary between the road and the adjacent sidewalk, where the curb lies, is modeled as a cubic polynomial. After the world model is fitted iteratively to the DEM, the cubic polynomial curb is extracted, with its adjacent elevation profiles provided by the road and the sidewalk surfaces. This allows robust detection of curbs with varying heights, and it extends the detection depth compared to local methods.

Another approach that is based on a cubic polynomial representation of curbs is [14]. Curbs are extracted without the assumption of a global model for the world, and, implicitly, of a certain predominant orientation for the detected curbs. The dense stereo data are organized into a DEM (Euclidian representation). Cell curbs are detected based on the DEM gradient and, using a RANdom SAmple Consensus (RANSAC) approach, relevant cubic polynomials are extracted. The curb patches are then located along the relevant cubic polynomials, and their average height is computed based on the local elevation data. Being a local method, this approach is faster and it can deal with curbs with any orientation relative to the ego. However, it is less stable if the 3D data is poor.

Even though the cubic polynomial model is quite general, allowing both curvature and curvature variation, there are some simple scenarios where it fails to model correctly the curbs. For example if a curved curb patch is continued with a straight curb patch (see Fig. 1) then the curvature variation term will cause an “overshooting” effect along the straight patch.

In the next section we will present an overview of our proposed general curb model/algorithm and contributions to the problem of curb detection.

II. OVERVIEW OF THE PROPOSED ALGORITHM AND CONTRIBUTIONS

In this paper we propose a more general model for curbs: cubic splines. An overview of the curb detection algorithm based on this general model is presented in Fig. 2.

First, dense stereo data are organized as a DEM. Curb measurements are selected (section III) as those DEM cells that are inliers of relevant cubic polynomials computed with a RANSAC approach.

The second step is to perform temporal integration (section IV) of the curb measurements: past curb measurements are transformed into the current system of coordinates. If multiple sets of curb measurements exist (several curbs) then data association is performed based on the 3D location of the curb measurements.

The final step is the least square fitting of the spline curb model (sections V and VI) to each set of global curb measurements. This fitting will provide the 3D curb, with its associated lower and upper elevation profiles.

III. CURB MEASUREMENTS EXTRACTION

This step is related to [14] regarding the representation of the dense stereo data and the way we select potential curb cells in the DEM.
A set of curb measurements is selected for each relevant polynomial extracted with the RANSAC approach.

IV. TEMPORAL INTEGRATION. DATA ASSOCIATION

The methodology of maintaining the curb measurements over time is similar to the tracking data association problem. A global set of curb measurements is a set of curb measurement that was built along several consecutive frames and represents a unique 3D curb.

First, let us define the area around the ego car outside of which past curb measurements are not stored anymore. Normally this area depends on the application needs. In order to provide enough support for the cubic spline curb model, we use an area that has the same width as the DEM and extends back three times the length of the ego car (see Fig. 4).

For the sake of readability, we briefly revisit the steps used to select curb cells [14]:

1. The DEM is built from the dense stereo data (see Fig. 3.b). A DEM cell has a size of 10 cm x 10 cm in the horizontal plane.
2. Edges having the DEM gradient within a curb specific interval are detected (see Fig. 3.c).
3. Temporally persistent edge points that are not occluded by other edge points are selected (see Fig. 3.d).
4. Relevant cubic polynomials are extracted using a RANSAC approach (see Fig. 3.e).
5. For each of the relevant polynomials the inliers are selected as curb cells for the current frame (see Fig. 3.f).

For each curb cell, a curb measurement is generated as a set of four features \((r, c, E_L, E_H)\). \(r\) and \(c\) represent the DEM row and column of the curb cell (depth and lateral position relative to the ego car). \(E_L\) and \(E_H\) are the lowest and the highest elevations in a small vicinity (3x3 or 5x5 cells) around the curb cell. These elevations are likely to represent the elevations of curb’s adjacent surfaces (road/sidewalk). A histogram-based approach can be applied to select these elevations more robustly (consequently by using a larger vicinity).

Past curb measurements are transformed into the current system of coordinates in two steps. First we use the ego car’s motion sensors (speed and yaw-rate) and frames timestamp to compute the translation and rotation in the horizontal plane (a circular motion model is used). The depth and lateral position \((r, c)\) of each past curb measurement are transformed from the previous to the current system of coordinates.

Now it is possible to associate the sets of current curb measurements with the global sets, based on their location. A set of \(K\) current curb measurements is linked to a global set if a percentage \(P\) of the \(K\) measurements overlaps the global set (with an accuracy of 1-2 DEM cells). Assuming a speed of 20m/s, a frame-rate of 20 fps, the depth displacement is 1 meter between two consecutive frames. Even if the curb would be only several meters long, a percentage \(P=50\%\) is more than conservative. If a global set is not found then this is the case of a new curb, and a new global set is created containing the current set. Otherwise, if a global set is found, the current set of curb measurements are added to the global set.
The second step involves the alignment of the elevation data associated with the curb measurements. Misalignment of the elevation data (past and current) does not appear under normal circumstances, while driving on a smooth surface road. However, road irregularities (such as speed bumps) can cause sudden vertical rotations of the coordinates system, resulting in a relative pitch angle between the past and current elevation data. This angle is computed between the elevation data \( E_L \) from the previous frame curb measurements (not from all the past measurements) and the elevation data \( E_L \) of the current frame curb measurements. Then, all of the past curb measurements from the global set are aligned to the current curb measurements. Future work will be performed in this direction, probably by using visual odometry, in order to cope with uphill or downhill scenarios.

V. THE CUBIC SPLINE CURB MODEL

The cubic spline \( S(x) \), defined on an interval \([t_0, t_m]\), consists of a set of cubic polynomials \( \{P_k(x), k=1...m\} \), such that:

\[
S(x) = \begin{cases} 
  P_1(x), & x \in [t_0, t_1) \\
  \vdots \\
  P_m(x), & x \in [t_{m-1}, t_m] 
\end{cases}
\]  

(1)

In order to have a smooth and differentiable (up to the second order derivative) function in each interval knot, additional constraints are imposed:

\[
\begin{align*}
  P_{k-1}(t_{k-1}) &= P_k(t_{k-1}) \\
  P_{k-1}(t_{k-1}) &= P_k'(t_{k-1}), \quad k = 2...m \\
  P_{k-1}'(t_{k-1}) &= P_k'(t_{k-1})
\end{align*}
\]  

(2)

In the case of spline interpolation (when the spline must pass through each data point), the polynomial coefficient are determined from the constraints (2) and by forcing the spline to be “straight” in the extreme knots (second order derivative in \( t_0 \) and \( t_m \) must be zero).

For the problem of curb detection, considering the large number of measurements obtained through temporal integration, we need an approximation in the least square sense. Performing the least square minimization globally involves either a predefined set of knots or multiple iterations until the optimal placement of knots is selected. Either way the computational complexity is high. This can be avoided by least square minimization on intervals:

1. The first polynomial \( P_1(x) \) will be fitted in a least square fashion to the points inside its interval.
2. Each polynomial \( P_k(x), k=2...m \), is fitted in a least square fashion after it fulfills the continuity constraints from (2).

Fitting the first cubic polynomial \( P_1(x) \) to the set of \( n_1 \) points \((x_i, y_i)\) inside \([t_0, t_1]\) using algebraic least square fitting can be formally reduced to (4), which is a convenient form for real-time computation.

\[
P_1(x) = a_1x^3 + b_1x^2 + c_1x + d_1,
\]

(3)

\[
\begin{bmatrix}
  \sum_{i=1}^{n_1} x_i^6 & \sum_{i=1}^{n_1} x_i^5 & \sum_{i=1}^{n_1} x_i^4 & \sum_{i=1}^{n_1} x_i^3 & \sum_{i=1}^{n_1} x_i^2 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} 1 \\
  \sum_{i=1}^{n_1} x_i^5 & \sum_{i=1}^{n_1} x_i^4 & \sum_{i=1}^{n_1} x_i^3 & \sum_{i=1}^{n_1} x_i^2 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} 1 & \sum_{i=1}^{n_1} x_i \\
  \sum_{i=1}^{n_1} x_i^4 & \sum_{i=1}^{n_1} x_i^3 & \sum_{i=1}^{n_1} x_i^2 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} 1 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i \\
  \sum_{i=1}^{n_1} x_i^3 & \sum_{i=1}^{n_1} x_i^2 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} 1 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i \\
  \sum_{i=1}^{n_1} x_i^2 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} 1 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i \\
  \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} 1 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i \\
  \sum_{i=1}^{n_1} 1 & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i & \sum_{i=1}^{n_1} x_i \\
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  b_1 \\
  c_1 \\
  d_1 \\
\end{bmatrix}
= \begin{bmatrix}
  \sum_{i=1}^{n_1} x_i y_i \\
  \sum_{i=1}^{n_1} x_i^2 y_i \\
  \sum_{i=1}^{n_1} x_i^3 y_i \\
  \sum_{i=1}^{n_1} x_i^4 y_i \\
\end{bmatrix}
\]  

(4)

The next step is to fit each polynomial \( P_k(x), k=2...m \) to its interval. At each step, the previous polynomial \( P_{k-1}(x) \) is already computed. Based on the three continuity constraints (2), we can express the coefficients \( b_k, c_k \) and \( d_k \) as functions of the curvature variation term \( a_k \). Thus, each polynomial \( P_k(x), k=2...m \) will have only one degree of freedom. The least square fitting based only on the curvature variation will not be able to approximate correctly the data set. Considering that for curbs it is less relevant if the curve is two times differentiable in the knot locations, we can relax the constraints from (2) by ignoring the third constraint. This will provide two freedom degrees (the curvature and curvature variation terms) for each polynomial \( P_k(x) \), increasing the chance of success for the least square fitting. Based on the first two continuity constraints from (2), we can express the coefficients \( c_k \) and \( d_k \) as functions of \( a_k \) and \( b_k \). Each polynomial \( P_k(x) \) can be written as:

\[
P_k(x) = a_k(x^3 - 3t_{k-1}^2x + 2t_{k-1}^3) + b_k(x-t_{k-1})^2 + P_{k-1}(t_{k-1}) + P_{k-1}'(t_{k-1})(x-t_{k-1})
\]

(5)

Under this form, \( P_k(x) \) is fitted using algebraic least squares to the set of \( n_k \) points \((x_i, y_i)\) within the interval \([t_{k-1}, t_k]\). The system that must be solved using algebraic least square fitting is:

\[
\begin{align*}
  P_k(x_i) &= y_i, \quad i = 1...n_k \\
\end{align*}
\]

(6)

The curvature terms \( a_k \) and \( b_k \) are obtained, and, consequently, all the coefficients of \( P_k(x) \).

VI. COMPUTING THE SPLINE-BASED CURBS

The spline curb model is fitted to each global set of curb measurements. The row/column data (or column/row – depending on the orientation of the ego vehicle relative to the curb) provide the set of \((x_i, y_i)\) points for the spline least square fitting.

The main problem is how to choose the knots. Too many intervals will result in instability, while too few will cause a
poor approximation, with increased residuals. Through various trials, we have reached the conclusion that the length of one interval should be around 5 meters (it is small enough to model strongly curved curbs, but in the same time large enough, with enough points, to provide a good support for a cubic polynomial).

Our first experiments with the proposed spline fitting scheme showed an unexpected behavior with curved curbs. If a polynomial $P_k(x)$ was not accurately fitted near the knot $t_k$, then this error was transmitted and amplified to the next polynomial (see Fig. 5). An erroneous change in the orientation near a knot will cause the next polynomial to start with a less accurate orientation, and the least square fitting cannot compensate it entirely. Such changes are visible as maximums in the first derivative of the polynomial, or, as zeros of the second order derivative.

In order to correct this undesired behavior, at each step $k$, after the polynomial $P_k(x)$ is fitted, the knot $t_k$ is relocated backwards (within a maximum range), to the closest zero-crossing of the second order derivative of $P_k(x)$. The success of this correction is visible in Fig. 6.

So far, the curb is located in the bird-eye view (depth and lateral position). This localization is the most important. Additionally, since we have elevation data for each curb measurement, the vertical profiles of the adjacent road and sidewalk are estimated. In [14] only the vertical profile of the road along the curb was estimated because of the small number of curb measurements per frame. A quadratic polynomial was fitted to the set of roadside elevations, using a RANSAC approach, and the average curb height was estimated. In our current work, the curb measurements are enriched through the temporal integration. The vertical profiles of the adjacent road and sidewalk are modeled as quadratic polynomials, computed using a RANSAC approach. Future investigation is required to see if a more complex model is needed for the vertical profile of the global curbs.
global curb (if hundreds of curb measurements are available). Implementation was done in C++, and the tests were carried out on a Pentium Dual-Core processor (the implementation is single core).

The benefits brought by the proposed curb detection method are visible compared to the “per frame” detection results from [14]. In what follows the curbs detected with the proposed algorithm will be called “global”. A typical roundabout scenario is shown in Fig. 7.

A difficult scenario for curb detection is presented in Fig. 8: due to poor illumination and the windshield wiper being active, the quality of the dense stereo data is very low (see the DEM in Fig. 8.b). The per-frame curb is short and imprecise, but the global curb provides a stable detection.

In Fig. 9 several intermediate frames from a sequence are shown with the detected curbs (global versus per-frame). The sequence was taken while driving along a straight curb, with strong shadows. The per-frame curb data was very poorly reconstructed in 3D, but the temporal integration managed to provide a good density of curb measurements.

A quantitative evaluation was performed regarding the curb’s height, which, similar to [14], has an accuracy of about 5%.

VIII. CONCLUSIONS AND FUTURE WORK

A new (more general) model for curb detection was introduced in this paper, as a cubic spline. A real-time algorithm for the detection of cubic spline curbs was presented.

The method shows promising results, but additional objective evaluation must be performed. For this, we intend to generate ground truth data for curbs using a lidar.

The stereo uncertainties should also be integrated in the curb model and used in the fitting process. A comparison with a global least square fitting method should be performed. This will help quantify the effects of the proposed spline fitting technique (which uses some simplifying assumptions) on the curbs accuracy.

REFERENCES