# Modeling of Drivable Free Space with Fused Camera Data for Autonomous Driving

Oliver Speidel, Fabian Gies and Klaus Dietmayer*

**Abstract:** The detection and modeling of drivable free space is a major challenge for autonomous driving. A novel approach is to use high resolution cameras especially in order to get semantic information of the environment. However, most approaches fail to use a suitable representation of the free space which is indispensable for subsequent processes like the behavior and motion planning. This paper presents a generic framework for detecting and compactly modeling free space based on fused camera data. For the detection, the disparity image and pixel classification of a stereo-camera are used. Based on that, a new continuous semantic free space model including temporal filtering is introduced. First evaluations show impressive results even in challenging scenarios.

**Keywords:** B-Splines, Camera Processing, Free Space Modeling, Temporal Filtering

## 1 Introduction

In the context of automated driving, the estimation of drivable areas is an essential and intensively explored topic. A common strategy is to use a high-precision digital map in which road areas are stored. By localizing the vehicle with a high-precision Global Positioning System (GPS), a safe trajectory can be estimated based on the map and an object tracking algorithm [4, 15]. Nevertheless, these concepts, which heavily depend on digital maps, are not flexible enough to overcome complex scenarios or situations with poor GPS reception, for example in dense urban areas. So for autonomous driving it is inevitable to detect the dynamic drivable space while driving.

In order to do so, a novel approach is to use camera based sensors in combination with Convolutional Neural Networks (CNNs) as a pixel classification of the scene can be obtained [13]. Beside this, to overcome the missing 3D information in a mono-camera setup, there are concepts utilizing depth information yielded by a stereo-camera [14]. However, an appropriate representation of the free space is required in order to plan the vehicles motion for drivable areas. A common approach is to use grid maps where each cell has a probability to be occupied [4]. An according extension are Digital Elevation Maps (DEM) which additionally store height information of a cell [12]. Another strategy is to use so called Stixels, which model objects by vertical structures with according height extensions [8]. To further reduce the complexity for motion planning algorithms there are more compact and efficient mathematical models that explicitly represent the free space

*The authors are with the Institute of Measurement, Control, and Microtechnology at Ulm University, Albert-Einstein-Allee 41, 89081 Ulm (e-mail: firstname.lastname@uni-ulm.de).
The research leading to these results was conducted within the Tech Center a-drive. Responsibility for the information and views set out in this publication lies entirely with the authors.

boundary. Some novel approaches use polynomials, piecewise polynomials or B-splines to represent the boundary. In their fundamental form, these models also allow to use common temporal filtering methods as, for example, the Information or Kalman Filter (KF) for B-splines [9, 5].

To improve the accuracy and robustness of the obtained information, a multi sensor setup in combination with information fusion algorithms can be utilized. A generic approach based on a grid map is presented in [4], where radar and lidar information are fused, however, only free space and occupied cells are distinguished. A sensor specific fusion based on Stixels is used in [6]. By that, a pixel classification of a Neural Network (NN) is fused with the result of an object detection algorithm applied on a disparity image in order to detect unexpected objects.

In contrast, this paper presents a generic concept based on fused camera data in order to model free space. Therefore, the information obtained by a disparity image of a stereo-camera and a pixel classification is fused to estimate the drivable free space in vehicle coordinates. Additionally, a suitable spline model is developed which is not only able to smoothly represent the position of a free space boundary but also holds semantic information. Furthermore, a method for temporally filtering the spline model is introduced. The remainder of this paper is structured as follows. In Section 2 the developed framework is presented. Section 3 shows results based on real world data. Finally, a short summary and outlook is given in Section 4.

# 2 Free Space Modeling

## 2.1 Data Acquisition

As input data, the disparity image $\mathcal{C}$ of a stereo-camera is utilized to get depth information of the scene. Further semantic information is obtained by a NN which yields a pixel classification $\mathcal{S}$ of the image. Exemplary input data is depicted in Figure 1.

**Generation of a Digital Elevation Map**
The depth information is used to generate a DEM and to estimate labels for the single cells by fitting adjacent planes and a street plane into the DEM. The according algorithms are described in [12].In general, each cell receives unary and binary probabilities for belonging to a street or adjacent surface. The unary probability is based on the height of the cell and the binary probability is based on the relative height of neighboring cells. This concept was extended by the label *obstacles*. The according unary *obstacle* probability is formulated as,

$$\begin{aligned} P(z, z_{min}, z_{max}, \lambda_o, P_{min}, P_{max}) &= (P_{max} - P_{min})(s(z - z_{min}, \lambda_o) \\ &\quad -s(z - z_{max}, \lambda_o)) + P_{min}\,, \end{aligned} \tag{1}$$

where

$$s(\Delta z, \lambda_o) = \frac{1}{(1 + exp(-\Delta z/\lambda_o))}\ . \tag{2}$$

Thereby, $P_{min}$ and $P_{max}$ are the minimum and maximum probability for the label *obstacle* and $z_{min}$ and $z_{max}$ denote the minimum and maximum height of an obstacle. Thus, the

*obstacle* probability increases for DEM cells with a height between $z_{min}$ and $z_{max}$ above the street plane. The steepness of this gating function is controlled with $\lambda_o$. The binary probability is equally distributed.

As a result, each DEM cell is described as $\Omega = [\boldsymbol{x}, \boldsymbol{P}_x, h, \boldsymbol{p}]$, where $\boldsymbol{x} = [x_v, y_v]$ is the position of the cell center, $\boldsymbol{P}_x$ represents the according covariances, $h$ is the height and $\boldsymbol{p}$ contains the label probabilities. In general the label set $\mathcal{L} = \{\textit{street}, \textit{obstacle}, \textit{background}\}$ is applied.

**Processing of Scene Labeling**

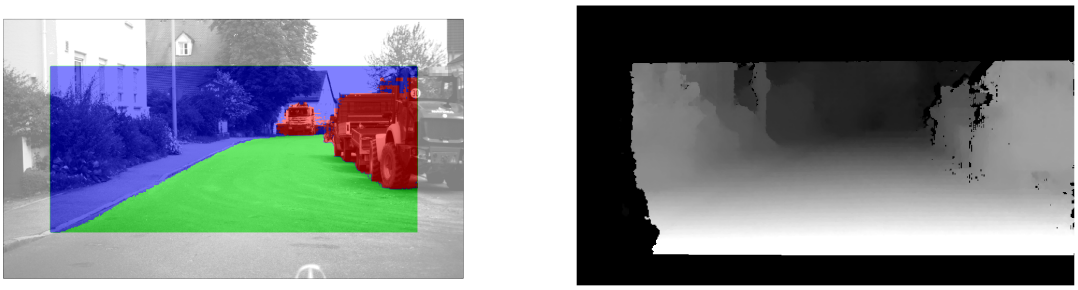
Basically, any pixel classification that provides at minimal the labels in $\mathcal{L}$ can be utilized as input. The algorithm used in this work is described in [13]. In order to reduce noise in $\mathcal{S}$, a morphological opening on each of the probability sets is done followed by a mapping of the labels to $\mathcal{L}$. In the end, the pixel-wise probabilities can be associated to the DEM cells using the stereo depth information. Thereby multiple pixels in the pixel classification image possibly belong to a single DEM cell. Therefore, the mean of all pixel-wise probabilities is used.

**Data Fusion**

Subsequently, the label probabilities can be fused for each DEM cell independently. Therefore, the single label probabilities for a cell yielded by $\mathcal{C}$ and $\mathcal{S}$ are weighted with a confidence weight $w_i$. Consequently, the fusion can be calculated by

$$p_i^{\mathcal{F}}(l) = (w_i^{\mathcal{C}}(l) \cdot p_i^{\mathcal{C}}(l) + w_i^{\mathcal{S}}(l) \cdot p_i^{\mathcal{S}}(l)) \cdot w_{norm}\,, \tag{3}$$

where $p_i^{\mathcal{C}}(l)$ and $p_i^{\mathcal{S}}(l)$ represent the probabilities for the label $l \in \mathcal{L}$ yielded by the according sensor and $p_i^{\mathcal{F}}(l)$ is the label probability of the fused DEM which is normalized with $w_{norm}$. This generic fusion also allows the integration of other information sources as only the weighted probabilities have to be added before normalizing.



(a) Output Scene Labeling

(b) Disparity Picture

Figure 1: Input Data Acquired by Existing Data Preprocessing Modules. On the left, the image of a grey-scale camera overlayed by the pixel classification result, where blue denotes the label *background*, green the label *street* and red the label *obstacle*. Thereby, the grey areas are not labeled for performance reasons. On the right, the disparity image is shown, where white pixels encode close data points and black pixels data points far away. Unknown pixels are also encoded with black.

## 2.2 Free Space Model

In order to be able to model the free space boundary, associated boundary points $\boldsymbol{B} = \boldsymbol{b}_{1...m} = [x_v, y_v, \boldsymbol{p}^t]^{\mathrm{T}}$ have to be extracted of the DEM, where $x_v$ and $y_v$ describe the position and $\boldsymbol{p}^t = [p_s^t, p_{bg}^t, p_{ob}^t]^{\mathrm{T}}$ describes the probability for the transition labels, i.e. the label of the objects beyond the free space boundary. The boundary point estimation is done column wise in the DEM which itself is aligned to the $u$-columns of the disparity image. Thereby, the last free cell in each column represents a boundary point $\boldsymbol{b}_j$, where $\boldsymbol{p}^t$ is calculated by means of the label probabilities of surrounding cells of $\boldsymbol{b}_j$. This is done by using a mean filter on the fused DEM $\mathcal{M}$ according to

$$p_j^t(l^t = c) = mean(\mathcal{M}_{p(l=c)} \notin street, \boldsymbol{H}_{\boldsymbol{b}_j}) \,, \tag{4}$$

with the boundary point $\boldsymbol{b}_j$ in the center of the mask $\boldsymbol{H}$ and $c \in \mathcal{L}$. Where the size of the mask defines the range of considered neighboring cells. While filtering, the cells labeled as *street* are ignored since only information about the occupied space is required. A schematic illustration of $u$-columns and boundary points can be seen in Figure 2. For the mathematical model of the boundary, B-splines are utilized. Therefore, the control points $\boldsymbol{C} = [\boldsymbol{c}_1, ..., \boldsymbol{c}_n]$ are estimated based on $\boldsymbol{B}$. In addition, a knot sequence $\boldsymbol{\tau} = [\tau_1, ..., \tau_{n+d}]$ is utilized which is equidistant in the $xy$-plane and where $d$ represents the order of the spline. In addition to the position, the probabilities for transition labels are approximated by the spline. This is done by adding additional dimensions to the spline for the labels in $\mathcal{L}$. The resulting boundary state model is given as

$$\hat{\boldsymbol{b}}(\tau) = [\hat{x}_v(\tau), \hat{y}_v(\tau), \hat{p}_s^t(\tau), \hat{p}_{bg}^t(\tau), \hat{p}_{ob}^t(\tau)]^{\mathrm{T}} = \hat{\boldsymbol{C}} \cdot \boldsymbol{N}_d(\tau)^{\mathrm{T}} \,, \tag{5}$$

where $\boldsymbol{N}_d(\tau)$ contains the values of the basis functions of the B-spline.

In general, the control points $\boldsymbol{C}$ can be estimated by using a Least Squares Estimator (LSE). With the concept of P-Splines [2] the curvature of the spline can be penalized which enables the generation of smooth boundaries. The according LSE is given as

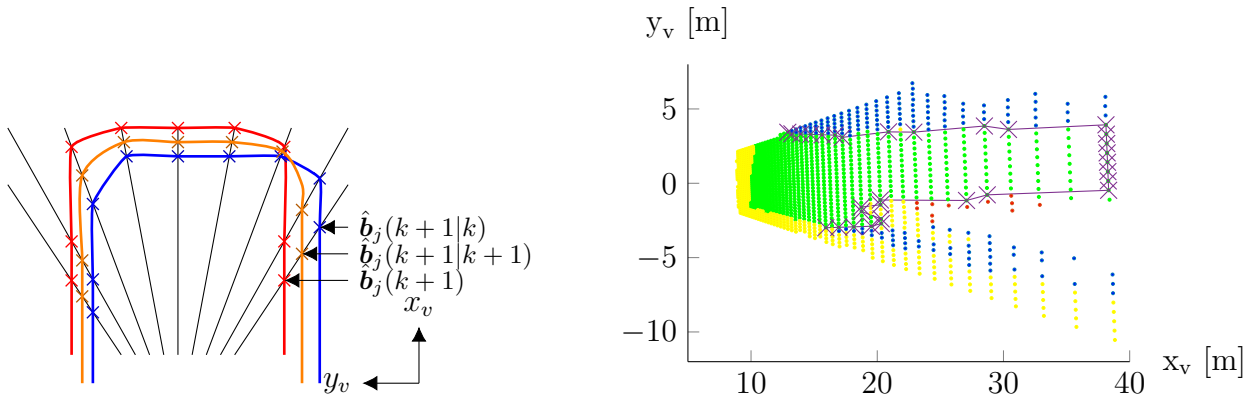
$$\hat{\boldsymbol{C}}(\boldsymbol{B}) = [(\boldsymbol{N_d}^{\mathrm{T}} \cdot \boldsymbol{N_d} + \lambda \cdot \boldsymbol{D_k}^{\mathrm{T}} \cdot \boldsymbol{D_k})^{-1} \cdot \boldsymbol{N_d}^{\mathrm{T}} \cdot \boldsymbol{B}^{\mathrm{T}}]^{\mathrm{T}} \,, \tag{6}$$

where $k$ defines the order of the penalty and $\lambda$ the intensity.

Thereby, a crucial problem is the association of the boundary points to the corresponding positions $\tau_i$. In [11] and [10] the measurements are assumed to be equidistantly distributed on the spline. In case of camera based sensors this is true for the image plane or respectively the $uv$-plane. However, this assumption does not hold for complex boundaries in the $xy$-plane as large distances might be covered by only a few boundary points and thus leads to overshooting behavior of the spline. Therefore, in this work, the positions of the boundary points on the spline are estimated by their Euclidean distance to each other. Thus,

$$\tau_j = \frac{||\boldsymbol{b}_{1,..,j}||}{||\boldsymbol{b}_{1,..,m}||} \cdot \tau_{n+d} \,, \tag{7}$$

where $||\boldsymbol{b}_{1,..,j}||$ denotes the accumulative Euclidean distances from $\boldsymbol{b}_1$ to $\boldsymbol{b}_j$ and $\tau_{n+d}$ is the total length of the spline. In areas where the density of measurements is low, pseudo measurements have to be generated in order to prevent ambiguous estimations. Considering the sensor model, the boundary between two boundary points has to be between




(a) Schematic Boundary Points

(b) Boundary Points in Downscaled DEM

Figure 2: Exemplary Boundary Points. On the right, exemplary position of boundary points in a downscaled DEM, where blue denotes the label *background*, green the label *street*, red the label *obstacle* and yellow the label *unassigned*. The crosses represent the position of boundary points. On the left, schematic spline models, where the measured, predicted and filtered spline as well as the according boundary points of $u$-columns are depicted. In addition, 3 corresponding boundary points $\boldsymbol{b}_j$ are labeled.

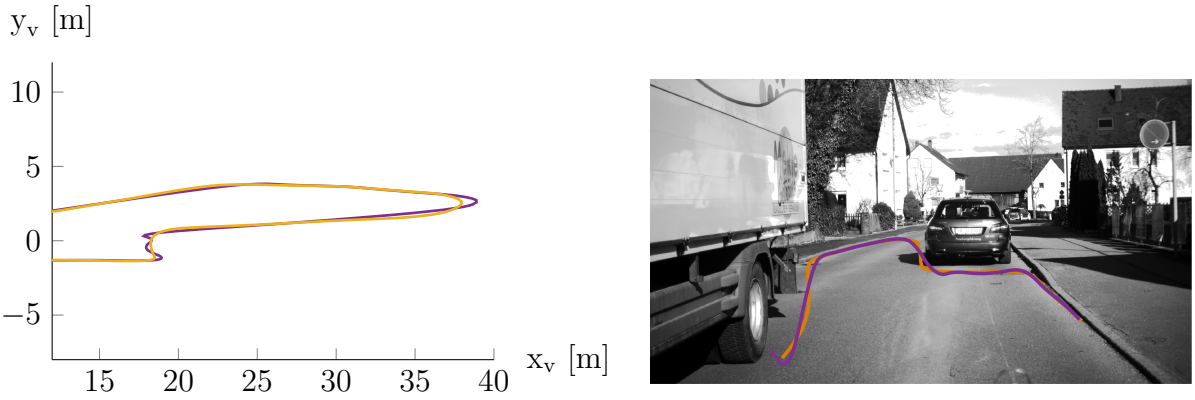
the according DEM columns. Therefore, pseudo boundary points can be estimated by linear interpolation. If the distance between two boundary points exceeds the required minimum distance, an additional boundary point $\boldsymbol{b}_j = \frac{1}{2}(\boldsymbol{b}_{j-1} + \boldsymbol{b}_{j+1})$ is generated. As a result, a smooth boundary spline can be estimated. Examples for both spline models are depicted in Figure 3.

## 2.3 Temporal Filtering

In order to estimate the boundary up to the margins of the DEM and to improve the robustness of the free space model, a temporal filtering based on a Extended Kalman Filter (EKF) [1] is implemented. Thereby, $\boldsymbol{C}$ is treated as the state and $\boldsymbol{B}$ contains the measurements. Therefore, the association of a predicted boundary point $\hat{\boldsymbol{b}}_j(k+1|k)$ with a measurement $\boldsymbol{b}_j(k+1)$ is done by calculating the intersections of the predicted spline with the DEM columns. As a result, each $\boldsymbol{b}_j(k+1)$ can be easily associated with a $\hat{\boldsymbol{b}}_j(k+1|k)$. Finally, the single filtered boundary points $\hat{\boldsymbol{b}}_{1,\dots,m}(k+1|k+1)$ are calculated independently and consequently the filtered control points $\hat{\boldsymbol{c}}_{1,\dots,n}(k+1|k+1)$ can be estimated. The different boundary points are depicted in Figure 2 and the according prediction of measurements and the subsequent innovation is described in the following.

**Prediction**

The prediction of the control points is done with the assumption of constant turn rate and velocity (CTRV) which results in a standard prediction step of a EKF. However, the transformation of the position covariances into the measurement space is approximated. This is done by treating the elements of the covariance matrices as additional dimensions of $\hat{\boldsymbol{c}}_i$ and as a result an approximation of the covariances for the position of corresponding boundary points is obtained.




(a) Top View (b) Image Plane

Figure 3: Exemplary scenes for a qualitative comparison between the proposed spline model in the $xy$-plane (orange) and the state of the art model with equidistant knots (purple) or respectively in the $uv$-plane. In the top row, the splines transformed into the image plane. In the bottom row, the splines in top view.

**Innovation**

Before calculating the filter gain, further adoptions are made. The model used for the prediction assumes a static boundary which, obviously, does not hold for dynamic objects. Therefore, the covariances of the boundary points are adapted according to their label. Finally, a Kalman update for the positions can be performed.

The filtering of the labels has to be done separately. For this reason, fixed weightings $w^{\boldsymbol{c}}_{\mathcal{L},j}$ and $w^{\boldsymbol{b}}_{\mathcal{L},j}$ for the previous and measured label are used with $\sum w_{\mathcal{L},j} = 1$. Consequently, the filtered label probabilities are given as

$$\hat{\boldsymbol{p}}^t_j(k+1|k+1) = w^{\boldsymbol{b}}_{\mathcal{L},j} \cdot \boldsymbol{p}^t_j(k+1) + w^{\boldsymbol{c}}_{\mathcal{L},j} \cdot \hat{\boldsymbol{p}}^t_j(k|k)\,. \tag{8}$$

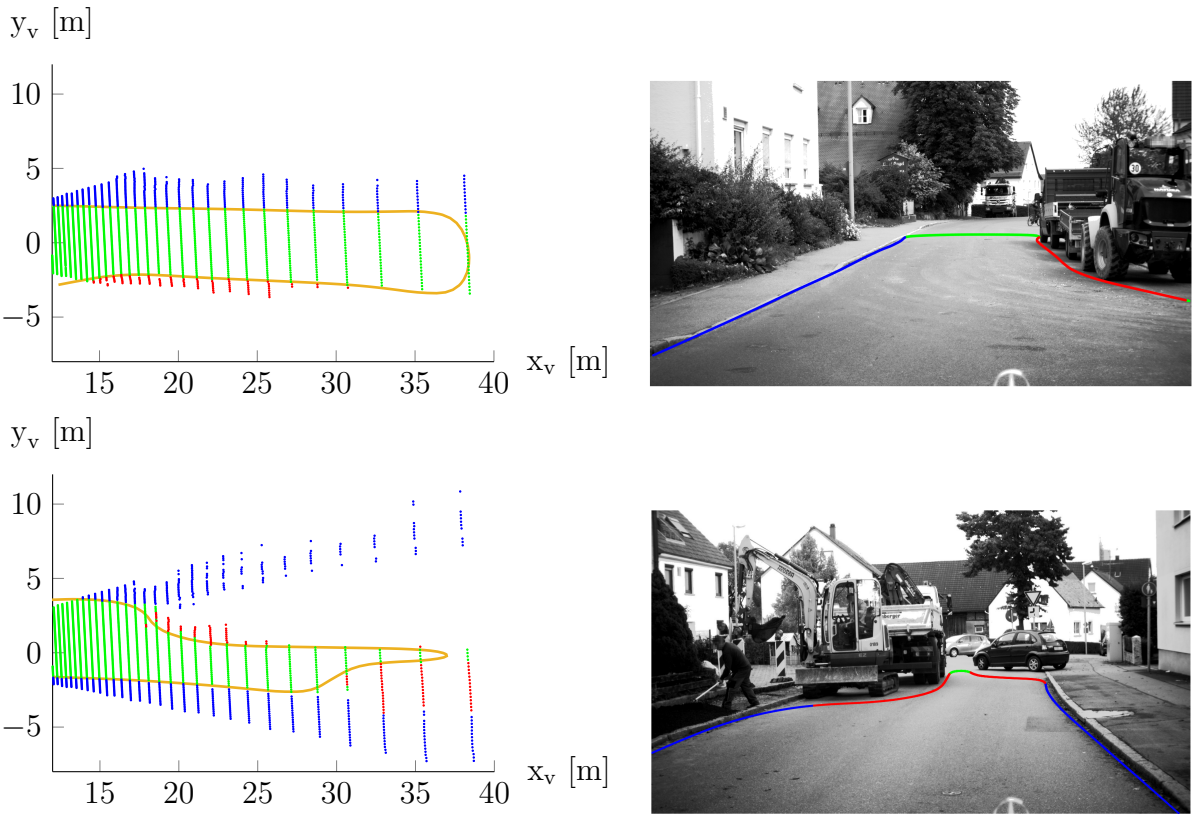
Based on that, the filtered boundary points $\hat{\boldsymbol{b}}_{1,\dots,m}(k+1|k+1)$ can be obtained by combining the corresponding position and labels. Exemplary results are shown in Figure 4. Thus, a temporal filtering for the spline is available which avoids association problems and handles semantic information.

# 3 Evaluation

For the evaluation, real world data is used which was acquired with an experimental vehicle of Ulm University [4] and labeled manually in order to get ground truth data. Thereby, four recorded sequences are used comprising rural roads and urban streets as well as different light conditions and complex scenarios in construction zones.

In general, the labeling in the image plane is done using MATLAB with the Annotation Tool presented in [3] and the labeling toolbox shown in [7]. As all sequences together contain more than 3000 frames, a subset of 80 frames are labeled with a high variety in the scenes. Consequently, input data and according ground truth is available to evaluate the detected free space.

Evaluating the accuracy of the spline position, boundary points of the labeled ground




(a) Top View (b) Image Plane

Figure 4: Boundary Spline Examples, where the label *background* is encoded in blue, *obstacle* in red and *street* is shown in green. On the left, the resulting free space boundary spline in vehicle coordinates, where the DEM cell centers with according labels are shown. On the right, the resulting free space boundary in the image plane, where the transition labels of the spline are shown.

truth are used and a Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{m_{gt}} \sum_{j=1}^{m_{gt}} ||\boldsymbol{b}_j - \hat{\boldsymbol{b}}(\tau_j)||} \tag{9}$$

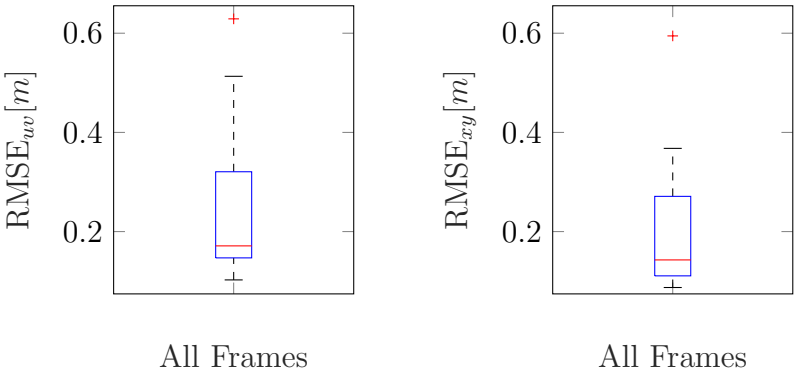
is calculated. Where $\boldsymbol{b}_j$ describes the ground truth boundary point that has the minimal Euclidean distance to the spline boundary point $\hat{\boldsymbol{b}}(\tau_j)$ and $m_{gt}$ is the number of evaluated boundary points.

The label quality is evaluated separately by the percentage of false labels, where the labels $\boldsymbol{p}_j^t$ and $\hat{\boldsymbol{p}}^t(\tau_j)$ are compared. By that, the false label ratio is defined as

$$\text{false label ratio} = \frac{\sum_{l \in \mathcal{L}} FP}{m_{gt}}, \tag{10}$$

where $FP$ represents the false positives for the label $l$.

In order to analyze the accuracy of the spline model itself, the RMSE is measured for the presented approach and the spline based free space model introduced in [9]. The




| Symbol | Value |
|---|---|
| $d$ | 3 |
| $n$ | 30 |
| $m$ | 54 |
| $\lambda$ | 0.3 |
| $k$ | 2 |
| range | $40m$ |

Figure 5: Comparison between the RMSE yielded by the proposed spline model in the $xy$-plane and the state of the art model with equidistant knots or respectively in the $uv$-plane.

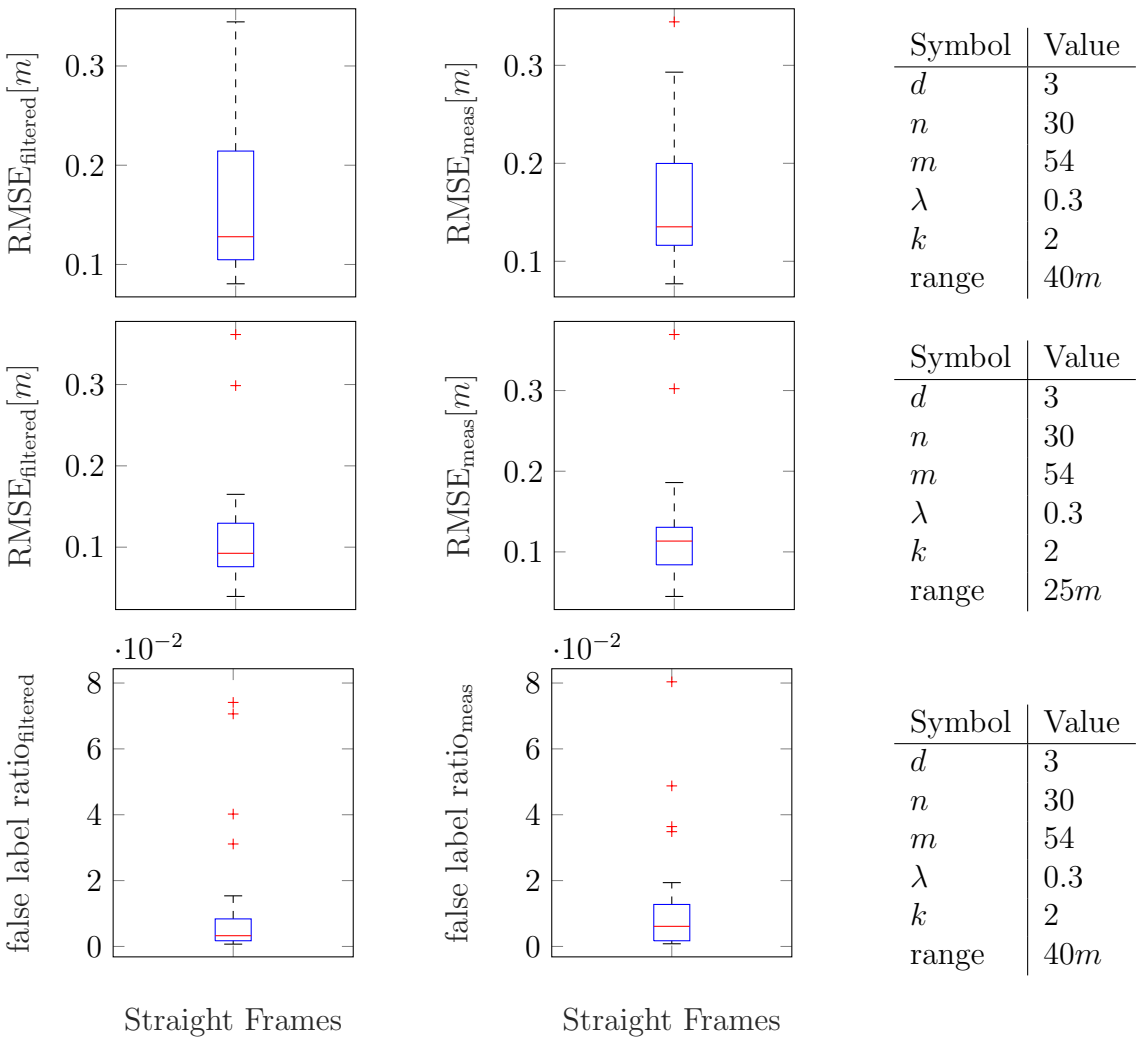
corresponding results are depicted in Figure 5 using all 80 annotated frames. Accordingly, the median is reduced from $0.17m$ to $0.14m$. Further, the standard deviation of the spline resulting from the presented model is with $0.088m$ lower than $0.095m$ in the sate of the art model. These findings are also qualitatively shown in Figure 3. Thereby, the $xy$-plane spline is much smoother as there is no overshooting behavior. When using camera based sensors this is often the case in transition areas between near and far objects.

Further, the temporal filtering of the spline is evaluated in Figure 6. Thereby, the advantages of the filtering can be illustrated best for straight road parts as the utilized ego motion module yields worse yaw angle estimations in curvy scenarios. This leads to errors of the spline prediction but due to sensor input and not the concept itself. The median of the label error evaluated on a range of 40 meters can be reduced from 0.61% to 0.36% by the filtering, as well as the upper quartile and whisker.

As the filtering leads to stronger smoothing at object transitions, *background* boundary parts are used to calculate the RMSE in order to keep comparability by using the same parameter set. The median of the RMSE is reduced from $0.135m$ to $0.127m$ however the upper quartile increases form $0.199m$ to $0.214m$ in the filtered result. This is due to some outlier frames in which the yaw angle estimation is still inaccurate. By regarding the spline boundary up to 25 meters and therefore reduce the influence of errors in the estimated yaw angle, it can be seen that the median with $0.099m$ and $0.113m$ as well as the upper quartile of the RMSE with $0.129m$ and $0.131m$ is lower in the filtered result. The remaining outliers occur due to systematic errors of the input data in multiple subsequent frames and therefore can not be compensated with filtering algorithms. By regarding the qualitative results[1], it can be observed that the boundary is smoothly estimated up to the DEM margins. In addition, the filtered boundary appears much more stable.

Consequently, the developed spline model successfully represents the free space boundary even better than state of the art concepts. Given a accurate yaw angle estimation, the boundary can be additionally tracked and improved by the presented temporal filtering approach.

[1]https://youtu.be/P18miHm0chE




| Symbol | Value |
|---|---|
| $d$ | 3 |
| $n$ | 30 |
| $m$ | 54 |
| $\lambda$ | 0.3 |
| $k$ | 2 |
| range | $40m$ |

| Symbol | Value |
|---|---|
| $d$ | 3 |
| $n$ | 30 |
| $m$ | 54 |
| $\lambda$ | 0.3 |
| $k$ | 2 |
| range | $25m$ |

| Symbol | Value |
|---|---|
| $d$ | 3 |
| $n$ | 30 |
| $m$ | 54 |
| $\lambda$ | 0.3 |
| $k$ | 2 |
| range | $40m$ |

Figure 6: Evaluation of the Temporal Filtering, where *background* boundaries on straight sequences are regarded. First row, evaluation of the RMSE with a range of 40 meters. Second row, evaluation of the RMSE with a range of 25 meters. Third row, evaluation of the label error ratio on a range of 40 meters.

# 4 Conclusion and Outlook

The main contribution of this paper is a new continuous semantic free space model with an according temporal filtering. In addition, a generic concept for the integration and fusion of semantic camera based data is proposed. As the evaluation shows, the proposed concept is capable of representing even complex scenarios and the spline model even outperforms state of the art concepts.

To further improve the results, future work includes the incorporation of lidar data, as well as an improved estimation of the yaw angle by utilizing the sequence of stereo images. In addition, it is reasonable to replace the currently used pixel classification with a state of the art concept. Furthermore, a real time implementation is aspired in order to use the free space representation for motion planning.

# References


[1] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[2] Paul H. C. Eilers and Brian D. Marx. "Flexible Smoothing with $B$-splines and Penalties". In: *Statistical Science* 11.2 (1996), pp. 89–102.

[3] F. Korč and D. Schneider. *Annotation Tool*. Tech. rep. TR-IGG-P-2007-01. June 2007.

[4] F. Kunz et al. "Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 666–673.

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

[6] Sebastian Ramos et al. "Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling". In: *CoRR* abs/1612.06573 (2016).

[7] Bryan C. Russell et al. "LabelMe: A Database and Web-Based Tool for Image Annotation". In: *International Journal of Computer Vision* 77.1 (May 2008), pp. 157–173.

[8] L. Schneider et al. "Semantic Stixels: Depth is not enough". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. June 2016, pp. 110–117.

[9] M. Schreier, V. Willert, and J. Adamy. "From grid maps to Parametric Free Space maps; A highly compact, generic environment representation for ADAS". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 938–944.

[10] Matthias Schreier. *Bayesian environment representation, prediction, and criticality assessment for driver assistance systems*. Technische Universität Darmstadt, 2016.

[11] Jan Siegemund. "Street Surfaces and Boundaries from Depth Image Sequences Using Probabilistic Models". PhD thesis. Universitäts-und Landesbibliothek Bonn, 2013.

[12] J. Siegemund et al. "Curb reconstruction using Conditional Random Fields". In: *2010 IEEE Intelligent Vehicles Symposium*. June 2010, pp. 203–210.

[13] M. Thom and F. Gritschneder. "Rapid Exact Signal Scanning With Deep Convolutional Neural Networks". In: *IEEE Transactions on Signal Processing* 65.5 (Mar. 2017), pp. 1235–1250.

[14] Q. Yang et al. "Road detection by RANSAC on randomly sampled patches with slanted plane prior". In: *2016 IEEE 13th International Conference on Signal Processing (ICSP)*. Nov. 2016, pp. 929–933.

[15] J. Ziegler et al. "Making Bertha Drive; An Autonomous Journey on a Historic Route". In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (Summer 2014), pp. 8–20.