B-Spline Modeling of Road Surfaces with an Application to Free Space Estimation

Andreas Wedel, Hernán Badino, Clemens Rabe, Heidi Loose, Uwe Franke, and Daniel Cremers

***Abstract*—We propose a general technique to model the visible road surface in front of a vehicle. The common assumption of a planar road surface is often violated in reality. A workaround proposed in literature is the use of a piecewise linear or quadratic function to approximate the road surface. Our approach is based on representing the road surface as a general parametric B-spline curve. The surface parameters are tracked over time using a Kalman filter.**

The surface parameters are estimated from stereo measurements in the free space. To this end, we adopt a lately proposed road-obstacle segmentation algorithm to include disparity measurements and the B-spline road surface representation. Experimental results in planar and undulating terrain verify the increase in free space availability and accuracy using a flexible B-spline for road surface modeling.

***Index Terms*—Road surface, v-disparity, B-spline, free space.**

I. Introduction

MODELING a vehicle's environment is challenging but absolutely essential to maneuver autonomous vehicles. It includes the localization of moving objects as well as the modeling of the stationary infrastructure. In an ideal world, all the necessary information is available on demand from an omniscient oracle. In reality, only a small portion of the information is available on demand by making use of a database. This may include the location of traffic signs, the information about road curvature, or even a three dimensional model of the complete infrastructure.

In most environments with other traffic participants, such information has to be generated online using environment perception techniques. The ideal environment perception sensor generates a three dimensional model of the vehicle environment. In this paper we use rectified stereo camera images and focus on modeling the free space in front of the vehicle. The free space is the available space to maneuver a road vehicle so as to avoid collision with any object. It is described by the ground surface and is limited by other obstacles.

But what defines an obstacle? In general, obstacle refers to something that stands in the way. In vehicle environments it refers to a structure that blocks the path by sticking out of the ground surface. Common obstacle detection algorithms detect obstacles by evaluating the height above ground, where the ground is modeled as a planar surface. In situations such as shown in Fig. 1, the assumption of a planar ground surface is violated and such procedure fails.

A. Wedel, H. Loose, and U. Franke are with Daimler Research, Germany.
H. Badino is with the Robotics Institute, Carnegie Mellon University, USA
C. Rabe is with the Christian-Albrechts-University, Kiel, Germany
D. Cremers is with the Computer Vision Group, Friedrich-Wilhelms-University, Bonn, Germany
Manuscript received YYYY YY, 2008; revised XX, XXXX.

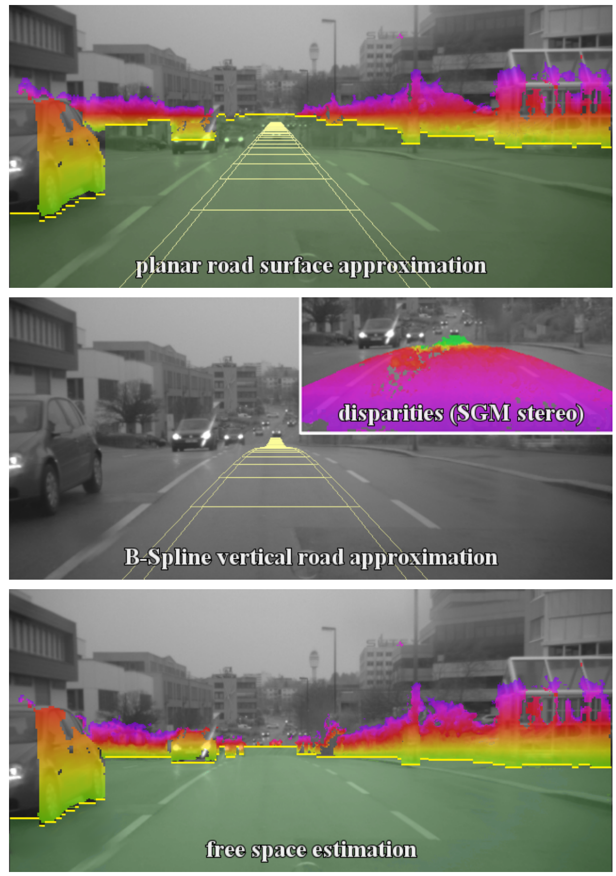


Fig. 1. The images illustrate the contributions in this paper. An example of a scene with undulating terrain in a city environment is shown. The road course ahead is planar in the vehicle vicinity; then it drops down before it starts rising. Color encodes the relative height of obstacles for the free space and the distance for the disparities. A planar ground assumption (*top image*) is invalid in the depicted scene and yields errors in the free space estimation. The better vertical road approximation using the flexible spline representation and the correct free space estimation is demonstrated in the *lower images*.

A robust free space estimation approach requires to model the road surface in order to distinguish between obstacles and a free driving corridor. Assuming a planar road surface, slope changes in the road course ahead due to approaching a hill or a dip are not modeled and can not be used to restrict the free space. In this paper we develop an algorithm to model non-planar road surfaces, which we represent as B-splines. The approximation via B-spline techniques, widely used in surface modeling, yields accurate results for the vertical road profile even in large distances up to 100 m. Section II introduces B-splines and describes the estimation of the spline parameters from stereo disparity measurements. We describe how to track the spline parameters over time using a Kalman filter to improve accuracy and gain robustness in Section III.

In Section IV, we adopt a lately published free space approach to use the obtained B-spline representation of the road surface. The original algorithm uses image edges to calculate a boundary between road and obstacle. In this paper, we extend this algorithm to use the disparity values of a stereo method as a second driving force for free space estimation. We fuse both approaches, edge directions and disparities, into a single framework for free space estimation, yielding better results in the novel combined approach. An evaluation section proving accurate free space estimation in situations where the planar ground estimation fails, demonstrates the practicability of ground surface modeling via B-splines.

II. Road Surface Modeling with B-splines

In this section we acquire the modeling of the road surface on the basis of B-splines. We first replicate the common v-disparity approach for modelling planar road surfaces and give an overview of existing approaches to extend this approach towards modelling non-planar surfaces. We will then discuss, how a subset of these approaches can be modeled using B-splines, which are more general. The embedding into a Kalman filter framework is discussed in Section III.

A. *Review of the v-disparity approach*

The v-disparity approach was first introduced as Helmholtz shear in [1], [2]. Under the assumption of a planar road without bank angle, the key idea is to fit a plane through 3D measurements obtained by triangulating corresponding image points in the left and right camera images. An image point (u, v) in the left image corresponds to an image point $(u - d, v)$ in the right image. The disparity offset d is zero at the horizon and increases linearly in v yielding $d(v) = a{\cdot}v{+}b$. v is the image row and a, b are parameters which depend on camera height and tilt angle. A linear fit is computed in the image space using the Hough transform on a row-disparity image, commonly known as *v-disparity* space (see Fig. 2). Labayrade et. al were the first to introduce the name *v-disparity approach* and proposed a real-time accumulation strategy in [3].

We will review the idea of this robust v-disparity approach (from [3]) in the simple case of fixed camera height h. Each point on the ground plane is then described by $Y(Z) = \tan\alpha \cdot Z - h$ with the camera tilt angle α, and the height Y and distance Z of world points, in the world coordinate system. Using the projection formulas for the pinhole camera and solving for the tilt angle α, we get

$$\tan\alpha = \frac{h}{c_b f_x} \cdot d + \frac{1}{f_y}\,(c_y - v) \tag{1}$$

for each v-disparity point (v, d). Here f_x, f_y are the focal lengths in pixel, c_b is the base line of the stereo camera system and c_y is the y-coordinate of the principal point in the image.

For a robust estimate of the tilt angle from a set of v-disparity points, the histogram of the $\tan\alpha$ values calculated by Equation (1) is analyzed and the tilt angle is found as the maximum in the histogram. In addition, the variance and the number of v-disparity points supporting the found tilt angle is used as a quality measure.

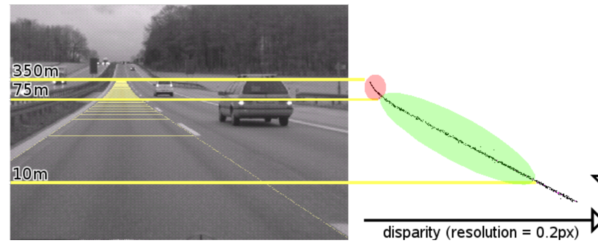
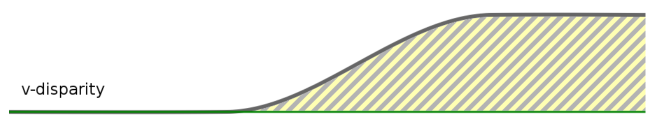


Fig. 2. v-disparity space of measurements on the road course ahead. The linear dependency between disparity and image row for the planar part of the road is visible. The plot also shows that the resolution of the v-disparity space (both v and d resolution) decreases with increasing distance.

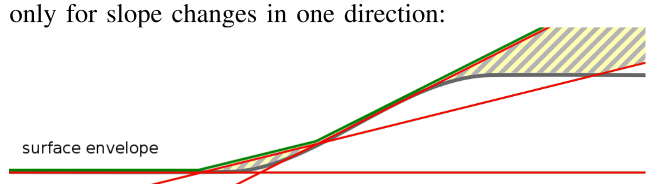
B. *Extending the v-disparity approach*

In the literature on intelligent vehicles, some approaches have been proposed to yield an approximation of non-planar road surfaces. We will describe the basic ideas of these approaches and point out their differences exemplary using an artificial ramp in the road course ahead (shown in *gray*). The camera is assumed on the left and measurements to be noise-free. The hatched regions show the approximation error for the different methods. As the focus is set on vertical road modeling, approaches which model the lateral surface change (such as [4], [5]) are excluded from this review.

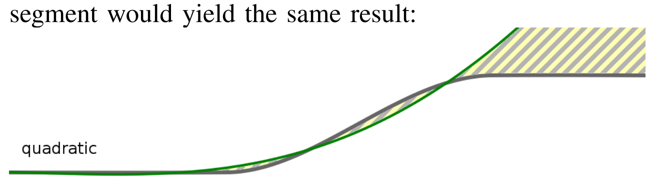
The Helmholtz shear approach as described in [2] approximates the ground as a planar surface. It uses the Hough transform to fit a planar surface (*green*) in the disparity space yielding accurate approximation if the planar vehicle vicinity but failing if the surface is rising or falling:



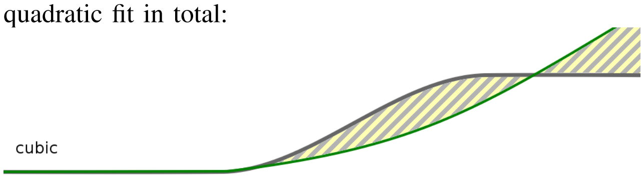
The envelope of surfaces approach described in [3] computes the k main v-disparity surfaces in the region, where k has to be chosen (3 in the example, shown in *red*). The resulting surface (*green*) is represented as inner or outer envelope (depending on the slope direction of the road). Surfaces are modeled as piecewise planar; hence slope changes are abrupt and not continuous. The approach incorporates the robust Hough transform techniques for the main surfaces but allows only for slope changes in one direction:



A quadratic approximation of the ground surface is proposed in [6]. It allows only slope changes in one direction and is not as stable in the vehicle vicinity as a Hough transform based approach. Using a B-spline of order two with one segment would yield the same result:

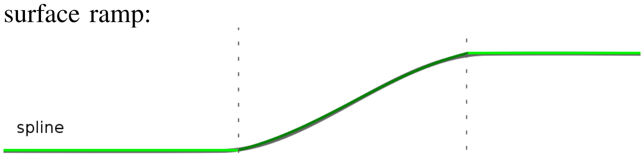


In [7] the authors propose a clothoid approximation of the ground surface. A clothoid is a higher order polynomial of degree three (cubic) widely used for lane recognition in the computer vision community. In their implementation however, the cubic parametrization is not evaluated. The authors use a Hough transform for the vehicle vicinity and a quadratic fit in large distances, yielding a better approximation than a quadratic fit in total:



Using a B-spline of order two with two segments, where the first segment is constrained by camera height and pitch would yield similar results.

Due to it's restricted parametrization all above mentioned techniques can only model slope changes into one direction; hence these approaches may fail to approximate the road surface if the road is undulating. The last example shows a surface approximation using a B-spline curve with three segments. Note the piecewise definition of the spline, shown in light and dark color and the good approximation of the surface ramp:



An example of undulating terrain in a real sequence with a B-spline surface reconstruction can be seen in Fig. 15. As we are interested in modeling the road course ahead (in large distances), fitting the approximated ground surface into the disparities, as done in the original a v-disparity approach, is not necessarily the best solution. If the modeling is done in image space, resolution decreases with increasing distance (illustrated in Fig. 2). To overcome this effect, we propose to fit measurements in world coordinates instead of image space to get a well defined depth resolution. The drawback of such an approach is that one has to account for the non-linear error propagation of the (noisy) disparity measurements.

In the next subsection we will review B-splines as a function vector space which embeds all polynomials, amongst them the piecewise linear and clothoid functions. Hence, our approach is a generalization of known surface approximations in literature and bridges the gap between these different approaches.

C. *B-splines*

B-splines are a basis for the vector space of piecewise polynomials of degree γ [8]. A B-spline curve $B(Z)$ of degree γ is defined by a $n+1$ dimensional coefficient vector $\mathbf{c}$:

$$B(Z) = \sum_{i=0}^{n} c_i N_{i,d}(Z) = \mathbf{N}_\gamma(Z)^\top \mathbf{c} \qquad (2)$$

$$\text{with} \quad \mathbf{N}_\gamma(Z) = \left[\begin{array}{ccc} N_{0,\gamma}(Z) & \dots & N_{n,\gamma}(Z) \end{array}\right]^\top .$$

$$\text{and} \quad \mathbf{c} = \left[\begin{array}{ccc} c_0 & \dots & c_n \end{array}\right]^\top .$$

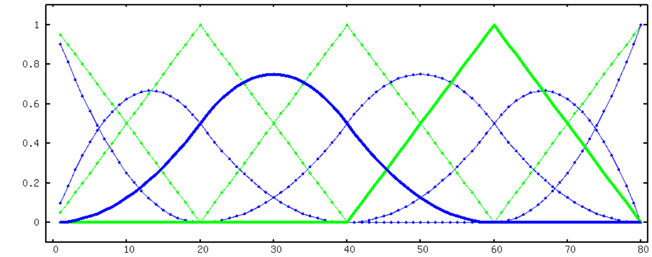


Fig. 3. Basis functions for an equidistant node vector. The linear basis functions are plotted in green, quadratic basis functions in blue.

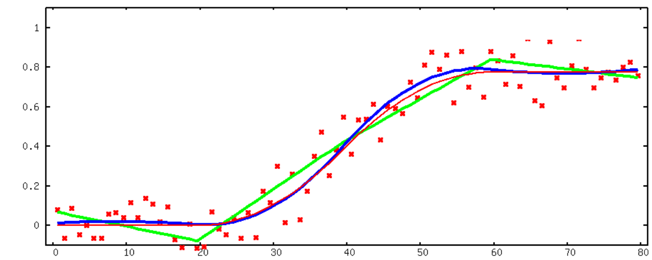


Fig. 4. The figure shows a surface fit through measurements with additive Gaussian noise (red) using piecewise linear splines (green) and piecewise quadratic splines (blue).

The polynomial basis functions $\mathbf{N}_\gamma(Z) = \{N_{i,\gamma}(Z)\}_i$ with local support are described by

$$N_{i,j}(Z) = \frac{Z - T_i}{T_{i+j} - T_i} N_{i,j-1}(Z) + \frac{T_{i+j+1} - Z}{T_{i+j+1} - T_{i+1}} N_{i+1,j-1}(Z)$$

$$\text{with} \quad N_{i,0}(Z) = \begin{cases} 1 & T_i \le Z < T_{i+1} \\ 0 & \text{otherwise} . \end{cases}$$

The number of nodes n defines the number of piecewise intervals where $1 \le \gamma \le n$. We use a node vector

$$\mathbf{T} = \{T_0, \dots, T_{n+d+1}\},$$

where the T_i denote distances in world coordinates and are in ascending order. Note that the following conditions have to be fulfilled:

$$T_0 = T_1 = \dots = T_\gamma , \quad T_\gamma < T_{\gamma+1} < \dots < T_n ,$$
$$\text{and } T_n = T_{n+1} = \dots = T_{n+\gamma+1} .$$

Fig. 3 shows the basis functions $\mathbf{N}_1(Z)$ for piecewise linear splines and $\mathbf{N}_2(Z)$ for piecewise quadratic splines.

To get the same parametrization as in the original v-disparity approach, d and n must both be set to 1. Note that only the parametrization is equivalent; the estimation technique via Hough transform is different to the least square approach we describe in the following. The quadratic and cubic ground surface approximation techniques are represented by setting $\gamma = 2, 3$ respectively.

In our implementation, we use equidistant nodes within the observed distance interval and cubic splines. Further details on B-spline construction and evaluation can be found in [8].

Fig. 4 shows a surface fit with linear and quadratic splines using the same node vector. Note the better approximation with quadratic splines.

If a fixed node vector is used, the basis functions remain constant and the spline function is altered only by the coefficient vector $\mathbf{c}$. This yields the common name *control vector* for the vector of spline coefficients. Due to the fact that the basis functions for the B-spline fit do not change, they can be calculated in a precomputing step yielding real time efficiency for the surface approximation via B-splines.

For the road course ahead, the B-spline $B(Z)$ encodes the relative height of the ground surface. If we are given M independent measurements

$$\{\text{distance, height}\}_{m=0}^{M} = \{Z_m, Y_m\}_{m=0}^{M}$$

and associated standard deviations σ_m (see [6] for the stereo triangulation error propagation), the goal is to find an optimal control vector $\mathbf{c}^*$ such that $B(Z)$ best fits to the measurements. The goodness of the fit can be expressed by a cost function evaluating the sum of deviations from the measurements:

$$\mathbf{c}^* = \min_{\mathbf{c}} \left\{ \sum_m \frac{1}{\sigma_m^2} \left(B(Z_m) - Y_m\right)^2 \right\} \ . \tag{3}$$

This boils down to finding the coefficient vector $\mathbf{c}^*$ which minimizes the weighted sum:

$$\mathbf{c}^* = \min_{\mathbf{c}} \left\{ \sum_m \frac{1}{\sigma_m^2} \left(\mathbf{N}_\gamma(Z_m)^\top \mathbf{c} - Y_m\right)^2 \right\} \tag{4}$$

$$\Leftrightarrow \quad \mathbf{c}^* = \min_{\mathbf{c}} \left\{ \underbrace{\begin{bmatrix} \frac{1}{\sigma_0^2}\mathbf{N}_\gamma(Z_0)^\top \\ \vdots \\ \frac{1}{\sigma_M^2}\mathbf{N}_\gamma(Z_M)^\top \end{bmatrix}}_{\mathbf{A}} \mathbf{c} - \underbrace{\begin{bmatrix} Y_0 \\ \vdots \\ Y_M \end{bmatrix}}_{\mathbf{h}} \right\} \tag{5}$$

and yields the familiar form of the least-squares problem

$$\mathbf{A}^\top \mathbf{A} \mathbf{c}^* = \mathbf{A}^\top \mathbf{h} \ . \tag{6}$$

Equation (6) can be solved either directly by matrix inversion or iteratively using any matrix-vector solver. Due to the embedding in a Kalman filter framework, these equations are fed as measurements to the Kalman Filter and are not directly solved. The state vector of the Kalman filter is the coefficient vector $\mathbf{c}$. Each row of $\{\mathbf{Ac} - \mathbf{h}\}$ is a Kalman filter measurement equation.

D. *Camera Parameters*

Until now the outer orientation of the camera is assumed to remain unchanged, therefore only changes in surface topology are accounted for. In vehicle applications, because of vehicle motion, one has to model for changes in the camera pitch angle α and for an offset in the camera height H_{off}. To account for these parameters, the surface equation is extended by these two parameters:

$$\text{Height}(Z) = \cos(\alpha)\mathbf{N}_\gamma(Z)^\top \mathbf{c} + \sin(\alpha) Z + H_{\text{off}} \ . \tag{7}$$

Recall, that distance Z and height are given in the world coordinate system of the moving observer. Because the camera height and the camera pitch angle could be modeled by translating and rotating the ground surface, additional boundary

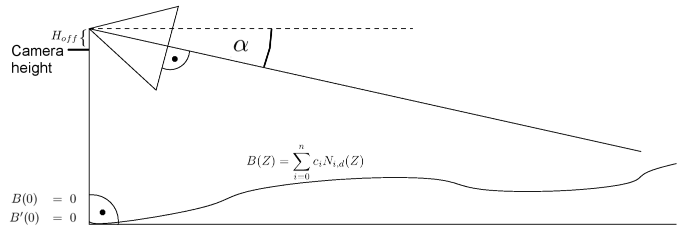


Fig. 5. The figure shows the camera parameters pitch angle α and height offset H_{off}. The touch and gradient constraints imposed on the B-spline can be seen as boundary conditions ensuring that the spline surface has height 0 and no slope at the camera foot print.

conditions have to be imposed. For moving platforms, these boundary conditions are straight-forward: The vehicle has to touch the ground surface and the surface gradient where the vehicle touches the ground has to vanish. Mathematically this can be formulated as:

$$\text{Touch constraint:} \quad B(0) = 0 \tag{8}$$

$$\text{Gradient constraint:} \quad B'(0) = 0 \ . \tag{9}$$

We include both equations in the Kalman filter framework as measurements $\{m_t = B(Z_0) - 0\}$ and $\{m_g = B'(Z_0) - 0\}$. Because the B-spline equation is linear in the coefficient vector $\mathbf{c}$, the measurement equation m_t for the Kalman filter is a linear measurement equation. The same linearity of the coefficients is true for the derivative measurement m_g; the derivative of a B-spline $B'(Z)$ computes as

$$B'(Z) = \sum_{i=0}^{n} c_i N'_{i,d}(Z)$$

with (see [8])

$$N'_{i,d}(Z) = \frac{d}{Z_{i+d} - Z_i} N_{i,d-1}(Z) - \frac{d}{Z_{i+d+1} - Z_{i+1}} N_{i+1,d-1}(Z) \ . \tag{10}$$

E. *Surface Smoothness*

For measurements corrupted by noise, a best fitted curve approximation is not necessarily the best continuous approximation. Two problems may arise: the number of measurements is too small to estimate all parameters and/or outliers may influence the result. One way to solve for such problems is prior knowledge in terms of smoothness. The curvature and the inclination of road surfaces are usually small. This knowledge can be introduced by penalizing high curvature and derivatives of the resulting B-spline. Since integration and differentiation are linear operators on the vector space of B-splines (compare Equation (10)), this can be formulated by additionally penalizing the quantities

$$\int \left(B'(Z)\right)^2 = \int \left(\mathbf{N}'_\gamma(Z)^\top \mathbf{c}\right)^\top \left(\mathbf{N}'_\gamma(Z)^\top \mathbf{c}\right) \tag{11}$$

and

$$\int \left(B''(Z)\right)^2 = \int \left(\mathbf{N}''_\gamma(Z)^\top \mathbf{c}\right)^\top \left(\mathbf{N}''_\gamma(Z)^\top \mathbf{c}\right) \ . \tag{12}$$

The integrals over the basis functions are computed efficiently using Gaussian quadrature. This step is computed offline because the basis functions remain constant.

The Gaussian quadrature with n_g being the number of (control) weights is exact for polynomials with degree $2n_g - 1$. For $n_g = 2$ the corresponding weights become $\omega_{1,2} = 1$ at position $x_1 = -\sqrt{\frac{1}{3}}$ and $x_2 = \sqrt{\frac{1}{3}}$ and the approximation is exact for polynomials of degree ≤ 3. The Gaussian quadrature is defined on the interval $[a, b]$ as

$$\int_a^b f(x)\, dx \approx \frac{b-a}{2} \sum_{i=1}^{n_g} \omega_i f\left(\frac{b-a}{2} x_i + \frac{a+b}{2}\right) . \quad (13)$$

For computing the smoothness constraints in (11) and (12) the function $f(x)$ becomes the multiplication of two basis functions,

$$f(x) = N'_{i,d}(x) N'_{j,d}(x) \quad \text{and} \quad f(x) = N''_{i,d}(x) N''_{j,d}(x) .$$

Because each of the first derivative basis function has degree $d-1$, the Gaussian quadrature for the integral of derivatives is exact for B-splines of degree $d = n_g$. The Gaussian quadrature for the integral of the second derivatives is exact for B-splines of degree $d = 2n_g$. Note that the local support of the basis polynomials can be used to gain real time processing.

III. Road Surface Kalman Filter

Since we work on image sequences, we impose regularity over time by applying a Kalman filter. This section describes the filter steps for the surface fit based on B-splines. Measurement equations for 3D world points are derived. A Kalman update is formulated and the Kalman prediction step is derived for the B-spline coefficients assuming a moving platform. For an introduction on Kalman filters we refer to [9].

The Kalman filter state vector $\mathbf{x}$ consists of the current parameter vector $\mathbf{c}$ describing the road profile, the camera pitch angle α and the camera height offset H_{off}

$$\mathbf{x} = \left[\begin{array}{ccc} \mathbf{c} & \alpha & H_{\text{off}} \end{array}\right]^\top . \quad (14)$$

This includes the physical actual movement of the vehicle in terms of height and pitch angle change and also implies optimization of the road surface in the coefficient space. In practice this is still acceptable because a one-to-one mapping from the coefficient space into world coordinates (height and distance) exists.

A. *Kalman Update Step*

Given an initial estimate $\mathbf{x}' = [\mathbf{c}', \alpha', H'_{\text{off}}]^\top$, the goal is to derive an equation for the update $\Delta\mathbf{x} = [\Delta\mathbf{c}, \Delta\alpha, \Delta H_{\text{off}}]^\top$ to get a better (updated) solution

$$\mathbf{x}^* = \mathbf{x}' + \Delta\mathbf{x} = [\mathbf{c}^*, \alpha^*, H^*_{\text{off}}]^\top .$$

Recall that B-splines form a function vector space and are linear in their coefficients. Assuming a small pitch angle we can set $\cos(\alpha) \approx 1$ and $\sin(\alpha) \approx \alpha$ and get:

$$\left[\begin{array}{ccc} \mathbf{N}_\gamma(Z_0)^\top & Z_0 & 1 \\ \vdots & & \\ \mathbf{N}_\gamma(Z_M)^\top & Z_M & 1 \end{array}\right]^\top \left[\begin{array}{c} \mathbf{c}' + \Delta\mathbf{c} \\ \alpha' + \Delta\alpha \\ H'_{\text{off}} + \Delta H_{\text{off}} \end{array}\right] = \left[\begin{array}{c} Y_0 \\ \vdots \\ Y_M \end{array}\right] .$$

The standard deviations for the single measurements are calculated via error propagation from stereo triangulation [10]. We assume a disparity standard deviation of 0.4 px. Our cameras used in the experiments have a base line of 35 cm and a focal length of 840 px.

The touch and gradient constraint boundary conditions in Equations (8) and (9) need to be formulated as measurements for the Kalman filter. This is done by introducing the deviation from zero for the B-spline and its derivative as additional measurements:

$$\underbrace{\left[\begin{array}{c} \mathbf{N}_\gamma(0)^\top \\ \mathbf{N}'_\gamma(0)^\top \end{array}\right]}_{\mathbf{P_C}} [\mathbf{c}' + \Delta\mathbf{c}] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right] . \quad (15)$$

We choose the standard deviations for these measurements in the order of 10^{-5}, allowing only small deviations to the constraints.

The smoothness constraints in Equations (11) and (12) are quadratic in the coefficients $\mathbf{c}$. We linearize the equations around the given estimate $\mathbf{c}'$ to solve for the update. Hence, we used lagged feedback for the smoothness constraints:

$$\underbrace{\left[\begin{array}{c} \mathbf{c}'^\top \int \mathbf{N}'_\gamma(Z) \mathbf{N}'_\gamma(Z)^\top \, dZ \\ \mathbf{c}'^\top \int \mathbf{N}''_\gamma(Z) \mathbf{N}''_\gamma(Z)^\top \, dZ \end{array}\right]}_{\mathbf{P_S}} [\mathbf{c}' + \Delta\mathbf{c}] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right] . \quad (16)$$

The summarized Kalman filter update equations for the measurement equations, the constraints and the smoothness equations read

$$\left[\begin{array}{ccc} \mathbf{N}_\gamma(Z_0)^\top & Z_0 & 1 \\ \vdots & & \\ \mathbf{N}_\gamma(Z_M)^\top & Z_M & 1 \\ \mathbf{P_C} & \mathbf{0} & \mathbf{0} \\ \mathbf{P_S} & \mathbf{0} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} c' + \Delta c \\ \alpha' + \Delta\alpha \\ H'_{\text{off}} + \Delta H_{\text{off}} \end{array}\right] = \left[\begin{array}{c} Y_0 \\ \vdots \\ Y_M \\ \mathbf{0} \\ \mathbf{0} \end{array}\right] .$$

Both matrices, P_C and P_S can be precomputed for a fixed node vector to save online computation time. Most of the measurement equations can also be precomputed using a simple trick (at the cost of some negligible inaccuracies): for each measurement equation, the basis functions N_γ need to be evaluated at the given distance Z_m. In our experiments we sum up all measurements within discrete equidistant distance intervals

$$\{I_{Z_i}, I_{Z_{i+1}}\} \text{ with } Z_{i+1} = Z_i + \Delta Z.$$

The basis functions can be precomputed for the middle point of every discrete interval and the corresponding row of the Kalman filter update equations is multiplied with the square root of the number of measurements per interval (because the solution is computed via least squares). For the sum of variances we use error propagation to weight the measurements of the Kalman filter with the correct variances [10].

B. Kalman Prediction Step

The Kalman filter prediction step models the dynamics of the system. For a moving vehicle, translation and rotation of the vehicle has to be modeled.

We model the road surface in the coordinate system of a moving observer. The node vector is kept fix in predefined distances. Keeping the node vector fix, old B-spline coefficients $\mathbf{c}'$ have to be projected onto the current coefficients $\mathbf{c}$. Minimizing the quadratic difference between last and current surface parametrization under the translation T yields

$$\min_{\mathbf{c}} \int \left(\mathbf{N}_\gamma(Z)^\top \mathbf{c} - \mathbf{N}_\gamma(Z+T)^\top \mathbf{c}'\right)^2 \; dZ \; . \tag{17}$$

This can be directly formulated into

$$\int \mathbf{N}_\gamma(Z) \; dZ \int \mathbf{N}_\gamma(Z)^\top \; dZ \; \mathbf{c} = \tag{18}$$

$$\int \mathbf{N}_\gamma(Z) \; dZ \int \mathbf{N}_\gamma(Z+T)^\top \; dZ \; \mathbf{c}' \; . \tag{19}$$

Again, the integrals can be computed using Gaussian quadrature. The Kalman filter hence acts as a low-pass filter on the coefficients. As the ground is assumed to be static, we set a low variance to the state coefficients and increase the variance with increasing distance.

Instead of projecting onto a static node vector, one can also shift the node vector with the ground plane. Then however, one has to deal with inserting new nodes and removing nodes at the endpoints.

Prior knowledge about change in camera pitch and camera height is also modeled in the prediction step. Such knowledge can either be estimated using ego-motion or applying the robust v-disparity approach in the vehicle vicinity. In our experiments we use the second approach and estimate the camera height and the camera pitch angle in the vehicle vicinity using the v-disparity approach. We update the state vector with the calculated height and tilt angle and allow only low variance as we assume these parameters to be accurate in the vehicle vicinity (up to 15 m).

IV. FREE SPACE COMPUTATION

In this section we describe our approach to compute the free space in front of a vehicle. The computation of the free space computation has two main goals:

- Find the distances to the closest objects.
- Find the road surface segmentation.

While finding the distance to objects aims at navigating the car or triggering safety systems, the second objective is probably of the same importance. It is crucial for the road surface estimation task described in the first part of this paper. The reason for this is, that measurements on vehicles and other objects in crowded scenarios influence the B-spline curve and the resulting curve estimation may become unstable in such scenarios. Therefore, only 3D measurements in the free space are used for the spline estimation, neglecting all stereo measurements on objects.

First, we present a literature overview describing different approaches to free space computation and motivate our choice of free space computation in Section IV-A. We describe the ideas presented in [11], which our algorithm is based on, in more detail in Section IV-B. Simultaneously, we describe how the non-planar road surface representation can be used for the free space computation and subsequently focus on our proposed changes and extensions.

A. Review of Free Space Algorithms

The computation of free space is an important issue in the autonomous robot domain. The motion planning problem implies the autonomous displacement of a robot from one place to another while avoiding collisions with obstacles on its way. For this purpose, occupancy grids are built [12] and the free space is obtained analyzing the occupancy likelihood of the grid cells (e.g. [13]). The literature on robot motion planning and occupancy grids is quite extensive and an overview of the state-of-the-art is given in [14].

Occupancy grids are also used in the automotive environment. The main difference is that robots usually maintain a global grid, while vehicles only build a local grid from the current ego-position. In [15] stereo measurement are used to build an occupancy grid. Free space is obtained applying a threshold to the occupancy likelihood of the cells.

In [16] free space is computed independently of evidence grids by applying inverse perspective mapping.

In [17] stochastic occupancy grids are computed based on stereo information. Stereo is integrated over time in order to reduce the disparity uncertainty and improve the accuracy of the occupancy grids. Free space is obtained by applying dynamic programming to a polar-like representation of the occupancy grid.

The algorithm presented in [11] computes a globally optimal solution to the road/obstacle boundary. The authors proposed an energy combining measurements on objects as well as measurements on the road surface. The maximum of this energy is found by dynamic programming. To our knowledge, this is the first time that measurements on the road surface in the free space directly contribute to the road/obstacle boundary estimation. We use the basic idea of this approach and extend it to use a B-spline representation of the road surface instead of the planar ground assumption. We also show how to integrate direct disparity measurements next to edge directions as originally proposed in [11].

B. Free Space using Dynamic Programming

The basic procedure to find the road-obstacle (free space) boundary via dynamic programming (see [11], [17]) is as follows:

1) Estimate the road surface orientation parameters.
2) Calculate a disparities matching score.
3) Find a consistent road-obstacles boundary.

In our case, the road surface orientation parameters are given by the B-spline $B(Z)$, where Z encodes distance and $B(Z)$ encodes the height of the road surface in the coordinate system of the moving observer. In order to distinguish between image pixels on the road surface and pixels on obstacles, the expected

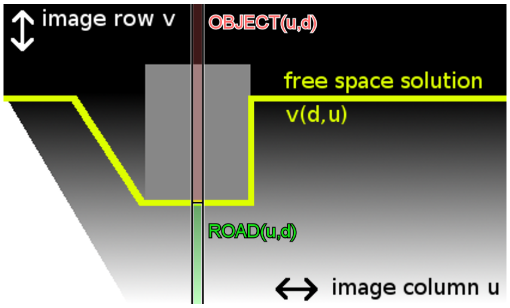


Fig. 6. Example of free space segmentation on planar ground. The gray value encodes disparity (white = near, black = far), the free space border is shown in yellow. The linear dependency (*v-disparity*) between disparity d and column v for the ground plane is depicted by the white triangle at the left.

disparity of the road surface for an image pixel has to be known. Under the common assumption of no road bank angle, the disparity of the road surface for an image row v is constant and depends on the relative height of the surface. We denote the disparity of the road surface for an image row v by $d(v)$:

$$d(v) = \text{disparity of mapped road at image row } v \; . \tag{20}$$

This equation is linear if and only if the ground plane is planar, yielding the v-disparity equation.

The first goal of a free space algorithm is to find the distance to the bounding obstacles. In image space this results in finding the disparity value d of the obstacles which bound the free space. The disparity of the obstacle becomes the same as the disparity value of the road surface in the image row, where the foot print of an obstacle touches the ground surface. A free space algorithm needs to know in which image row obstacles of disparity d touch the ground plane. We denote this image row $v(d)$:

$$v(d) = \begin{array}{c} \text{v-coordinate of the foot print} \\ \text{of objects with disparity } d \; . \end{array} \tag{21}$$

The free space can now be described by its boundary $v(d, u)$, respectively the distance $Z(u)$ or the disparity $d(u)$ of bounding obstacles, for every image column u (see Fig. 6).

But how is the *correct free space boundary* found? The key idea is to inspect every individual image column u (see Fig. 6 for an example). A matching score is obtained, summing up a score which evaluates the likelihood of pixels belonging to the road surface from the bottom of the image up to the free space boundary $v(d, u)$. A second matching score evaluates the fit of pixels belonging to objects with disparity d from the free space boundary in the image on upwards. The total score for an image row u and an an obstacle at disparity d becomes:

$$\text{SCORE}(u, d) = \text{ROAD}(u, d) + \text{OBJECT}(u, d) \; . \tag{22}$$

The best boundary match is given as the maximal score:

$$v(d, u) = \max_d \{\text{SCORE}(u, d)\}$$

If the maximal score for every image column is calculated independently (winner takes all strategy), the results will be noisy (differ from column to column). This is mainly due to low texture and low information content in some image regions. Another reason is stereo occlusion, shown in Fig. 7: *obstacle one* is visible in both, the left and the right camera; parts of *obstacle two* are only visible in the left camera and occluded by *obstacle one* in the right camera. A disparity measurement for this region is not possible. Hence, for the image columns within the occluded region, the free space boundary will be undefined.

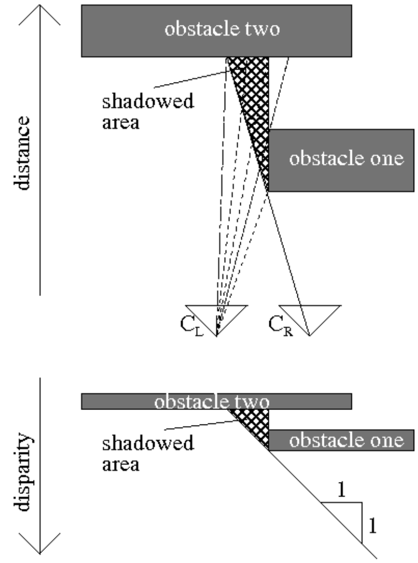


Fig. 7. Shadowed area, seen only by one camera. The upper draft shows distance encoded in world coordinates (X axis versus Z axis). The lower draft shows the same top view with distances encoded in disparities (X versus image disparity).

This is where the idea of dynamic programming comes in [11], [17]: deviations in the results between neighboring image columns are penalized in order to reduce the influence of outliers and to smooth the resulting free space boundary.

Fig. 7 also illustrates, why disparities are used instead of world distances to describe the free space boundary. If world distances were used, the shape of the *shadowed area* depends on object distances and camera setting. If on the other hand, distance is encoded in disparities, the incline has always an angle of 45° (change in u : change in d = 1 : 1). Hence, penalizing disparity differences is preferred over penalizing world distances due to the simple handling of shadowed regions and the direct inaccuracy treatment.

Algorithms which introduce smoothness in a global optimum manner via dynamic programming make use of a d-u (disparity-column) matching score table (see Fig. 10 in the results section). It has the dimensions image width by disparity range and encodes for every image column u and disparity d the likelihood that d is the disparity of the road-obstacle boundary $v(d)$. For details on finding the optimal boundary given the d-u table we refer to [11] or [17].

We will now describe two matching scores, image edges and disparity values, to construct a d-u table. We will discuss the advantages and disadvantages of both matching scores and combine both scores by adding the table entries for both approaches. The better accuracy in free space computation using the combined approach is verified in the experiments.

Fig. 8. The figure describes the idea of *plane sweep stereo*. The right image is translated according to a given disparity value and the consistency of gray values in the left image and the translated right image corresponds to the likelihood of a stereo match. In the figure, the superposition of the right and left image for different disparity values results in sharp structures in the background (for small disparities), mid plane, or foreground (for large disparities).

C. *Image Based Matching Score proposed in [11]*

In [11] a direct image based disparity score is proposed to find the free space boundary using stereo vision. It is based on the plane sweep idea [18]: for every disparity value d the pixel (u, v) in the left image is compared with the pixel $(u + d, v)$ in the right image. This corresponds to shifting the right image over the left image, or equivalent, projecting the left and right images onto a plane which is swept along the Z-axis of the camera coordinate system. The principle is illustrated in Fig. 8 using plain gray values. Only obstacles with the correct disparity value are in focus and the gray values in the right and left images coincide. All other regions of the image seem to be out of focus. [18] took this principle and applied an edge filter on the input images to provide a geometric reconstruction of the scene.

The same idea of matching image edges is adopted by [11] to construct the d-u table. Let $E_{L,R}(u, v)$ be the edge direction in the left and right image respectively at image position (u, v). The image based disparity score is then computed as

$$\mathrm{ROAD}(u,d) = \sum_{v=v(d)}^{v_{max}} w\left(E_L\left(u,v\right) - E_R\left(u, v + d\left(v\right)\right)\right) \quad (23)$$

$$\mathrm{OBJECT}(u,d) = \sum_{v=v_{min}}^{v(d)} w\left(E_L\left(u,v\right) - E_R\left(u, v + d\right)\right) \quad (24)$$

with $w(arg) = 1$ if $|arg| <$ threshold and 0 otherwise. v_{min} and v_{max} are the upper and lower bound of the region of interest in the images. The threshold for computing w is set to an angle of 10 degrees. A too large threshold yields an oversmooth free space boundary while a too small threshold does not accumulate enough edges. Equation (23) counts the number of matched edge directions on the road between the obstacle and the camera. Considering the homography of the road surface between the left and right images improves the results (see [19]). Equation (24) on the other hand counts the number of matches for the image column u on any potential obstacle with the disparity d.

D. *Disparity Based Matching Score*

If disparity maps are computed for pixels in the left image (we use the dense semi-global matching method [20]), the plane sweep approach can be replaced by direct disparity measurements. This has the advantage of implicit robustness because edge directions may be match for more than one disparity value. Furthermore, this speeds up the calculation of the disparity score table because no sweep step is necessary (here we assume, the disparity estimates are already available). Let (u, v) be an image position and $d_{u,v}$ the corresponding disparity value. The height $Y(v, d)$ and distance $Z(d)$ of the corresponding world point are computed by stereo triangulation. We define the disparity based score (with w as before and a threshold of $20\,\mathrm{cm}$ and $3\,\mathrm{px}$ resp.):

$$\mathrm{ROAD}(u,d) = \sum_{v=v(d)}^{v_{max}} w\left(\underbrace{Y(v, d_{u,v}) - Y(v, d(v))}_{\text{relative height}}\right) \quad (25)$$

$$\mathrm{OBJECT}(u,d) = \sum_{v=v_{min}}^{v(d)} w\left(d_{u,v} - d\right) \;. \quad (26)$$

Note that Equation (25) is equivalent to thresholding the height of measurements. Only measurements on the ground surface contribute to the ROAD score. In Equation (26) the distance to possible objects is thresholded. Again, measurements are summed up in the road area respectively on obstacles where the boundary is defined by the foot print of obstacles, $v(d)$.

E. *Discussion and Combined Approach*

The image based approach yields dense measurements and accurate boundaries. However, measurements may vote for many different disparity values which may yield instable results. An edge in the right image matches every edge with the same direction in the corresponding row of the left image. Hence, one can be sure that every possible match is found at the cost of many miss-matches. One way to reduce the number of miss-matches is a coarse-to-fine approach (see [11]). Such procedure however implies that one looses the global optimality of the solution.

The disparity based score relies on the disparity algorithm to dissolve multiple hypotheses. This implies, that the number of miss-matches is reduced. But if a false disparity measurement is present, the correct match is not in the set of disparity solutions. A disparity score is more robust (in terms that matches are unique) but less dense (no disparity available at some pixel positions) than the edge score.

We propose to combine both approaches, the edge based matching score and the disparity based matching score, by adding the single scores. This combines the robustness of the direct disparity measurements and the density of edge information. Experimental results showing the improved free space calculation are found in the next section.

Free space with edge based score. Free space with disparity based score. Free space using the combined approach.

Fig. 9. Comparison of free space computation using the different cost functions described in this paper. The edge based approach has problems in low contrast areas on the concrete wall. The disparity based approach does not pick up enough measurements at large distances, yielding to smoothing between the sign posts. The combined approach looks most convincing.

V. Experimental Results.

In our experiments we use stereo cameras with a base line of 35 cm and a focal length of 840 px. The image resolution is 640×480 px. Overall computation time (Intel CPU) is below 25 ms, using the hardware version of SGM stereo [20].

Fig. 9 compares the proposed cost functions. The edge based approach has some problems in low contrast areas of the image; the high contrast on the coming van for example influences the free space estimation and the free space is too large. The disparity based approach has not enough disparity measurements at large distances which yields to a smoothing of the free space boundary between objects (the sign posts). In the combined approach, the result looks most convincing.

In Fig. 10 the free space segmentation for a challenging scene in rainy conditions is shown. Note, that although only part of the vehicle on the left is visible the free space is determined correctly using the combination of edges and disparities. A smooth path from the left to the right with a large energy (depicted in gray value) is found in the d-u score table and describes the free space.

The following experiments evaluate the B-spline modeling of the road surface. In our experiments we use B-splines with degree 3 and five control points (equally distributed in the observed interval). The computational time is below 15 ms per frame on standard consumer hardware. Note, that for the ground approximation only the measurements within the free driving corridor are used. Using all stereo measurements would result in an unstable estimation. This can be seen in Fig. 11, where traffic blocks the view onto the road surface such that a reliable estimate is only possible up to 15 m.

Fig. 12 compares the pitch angle estimation using the proposed B-spline fit and the original v-disparity approach. In the scene, the road is mainly planar and we may assume that the v-disparity approach yields trustful results. The plot demonstrates similar results obtained by our algorithm when solving for the road surface parameters and the pitch angle within one Kalman filter (for the experiment we disabled the v-disparity prediction in the Kalman filter). It can also be seen, that the Kalman filter acts as a low-pass filter as long as no dynamic for the pitch angle is modeled. Experimentally, this demonstrates that the road surface and vehicle dynamics can be modeled in a Kalman filter.

Road surface approximations using the B-spline fit are

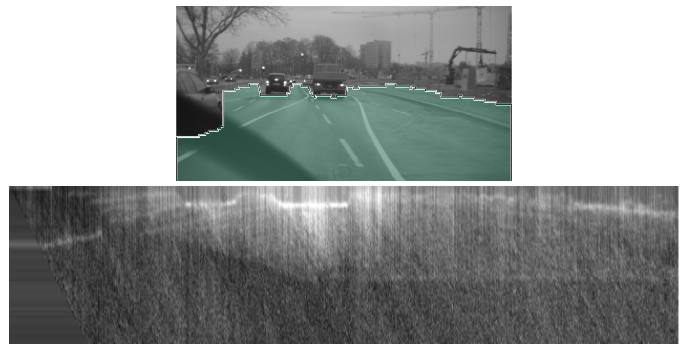


Fig. 10. The *top* image shows a typical free space segmentation in a challenging environment due to the partly visible windshield wiper and bad illumination in rainy weather. In the d-u table (*bottom*) brightness depicts the likelihood of the free space boundary; dynamic programming was used to find a smooth path from the left to the right with the largest likelihood.

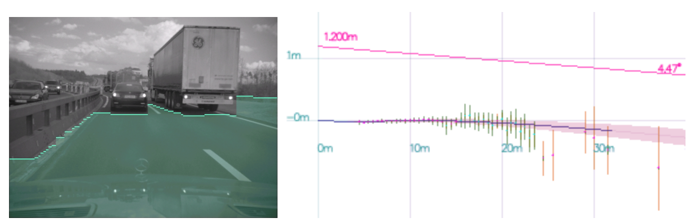


Fig. 11. Due to the free space segmentation (*left*) the height estimate (*right*) is not influenced by the vehicle 16 m ahead. The free space between the vehicle in front and the truck on the right allows for an B-spline estimate up to 30m, but reliable height measurements are found only up to 15 m.

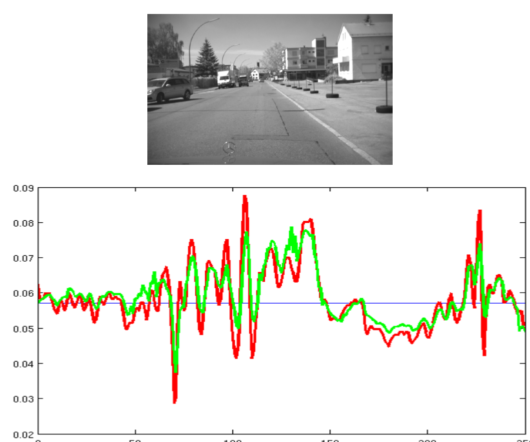


Fig. 12. Comparison of the pitch rate (rad/frame) obtained using the algorithm described in this paper (green) and using the v-disparity approach (red) on the scene shown at the top. Both plots show only small differences.

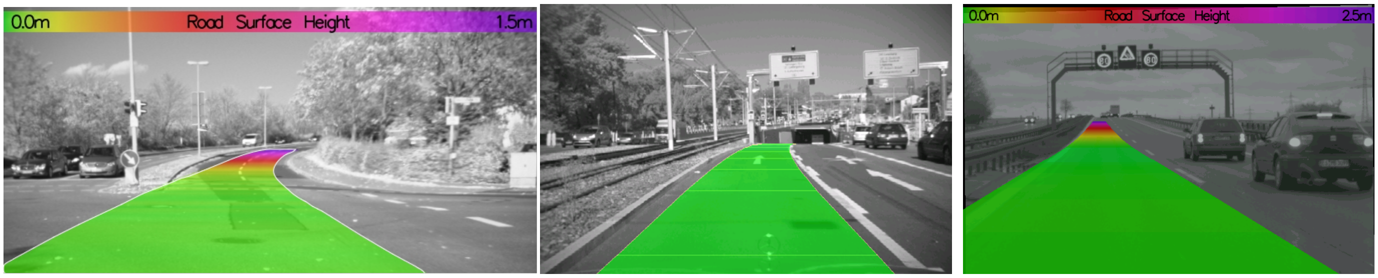


Fig. 13. Examples of the vertical road surface estimation for different scenarios. The proposed approach is able to estimate the height of the road surface up to 75 m, in uphill and downhill scenarios. The right image shows an Autobahn scene with road surface estimation up to 250 m.

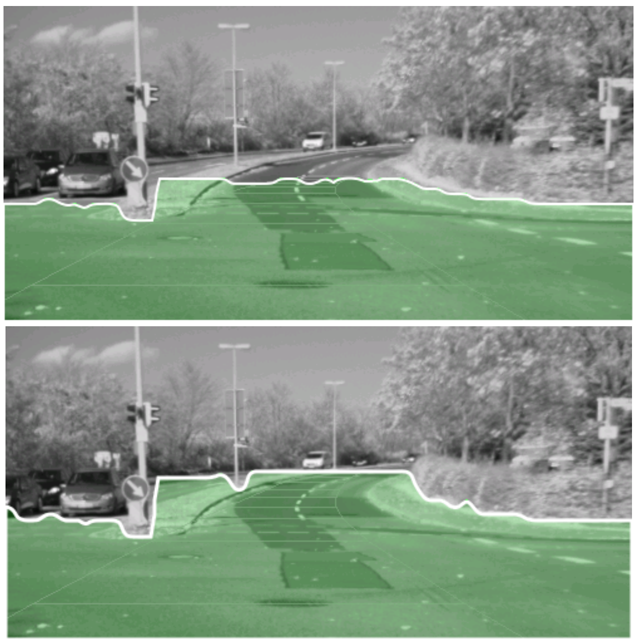


Fig. 14. Comparison of free space calculation using a planar road surface (*top*) and a B-spline representation of the road surface (*bottom*). The improvement compared to the flat ground assumption where the uprising road shows up as an obstacle becomes visible.

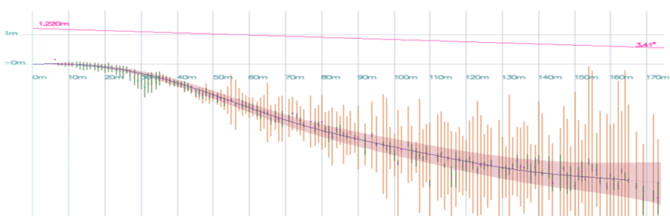


Fig. 15. The plot shows height measurements and variances used in the Kalman filter estimation of the B-spline road model for the example in Fig. 1. The camera is at the left. As one would expect, with increasing distance the measurement variance and the variance of the spline increase.

shown in Fig. 13. In the first two scenes the road surface is modeled up to 75 m in the Autobahn scene (camera focal length is 1400 px) up to 250 m. The qualitative experimental results demonstrate that modeling the road surface with B-splines is suitable and yields accurate results.

Fig. 14 now shows the result for the combination of B-spline road modelling and free space estimation. It also shows the result when modeling the road with a planar surface. This assumption holds for the close-by environment. Then the road rises to above one meter within the next 70 meters. Using a planar ground assumption, the image based free space computation fails because the assumed displacement of the road surface beyond 50m has an offset of several pixels; the disparity based approach fails because the height of the road surface beyond 50m is above any appropriate height threshold. Using the B-spline representation of the road surface, the free space is correctly determined.

An example of measurements and their approximating spline surface for the example from Fig. 1 can be seen in Fig. 15. The road profile is estimated correctly and as one expects, with increasing distance the measurement variance and the variance of the spline increase. The road course ahead falls before it rises again. In contrast to Fig. 14, the free space is estimated too large if the ground plane is assumed planar.

VI. Conclusions and Outlook.

We introduced an algorithm to robustly track smooth non-planar road surfaces. In contrast to existing approaches which are based on a piecewise planar or quadratic ground assumption, we allow for non-planar ground planes represented by a flexible B-spline curve. Thus our approach is a generalization of known surface approximations in literature and bridges the gap between these different approaches.

We experimentally demonstrated the accuracy of the B-spline representation for the application of free space estimation. To this end we modified a lately published free space algorithm to make use of the road surface approximation technique and directly use disparity values. Experimental results in planar and undulating terrain verify the gained availability of free space in everyday traffic.

Some open issues to be addresses in future work are

- Evaluating the visibility of surfaces to accumulate stereo measurements more intelligent.
- Integrating prior knowledge in terms of map data in the B-spline estimation.
- Robust M-estimator techniques to reduce the influence of outliers in the B-spline estimation.
- A quantitative evaluation of the B-spline modeling for the road profile.

The presented generalization of the v-disparity approach does not only offer more flexibility in road modelling from image sequences; it enables road modelling using range sensors such as the Velodyne 3-D laser scanner. The topic of sensor fusion is of broad interest. A possible research topic may be the fusion of stereo and laser scanner distance values within the presented approach for the road surface.

REFERENCES

[1] H. Helmholtz, "Treatise on physiological optics," 1925.

[2] J. Weber, D. Koller, Q.-T. Luong, and J. Malik, "An integrated stereo-based approach to automatic vehicle guidance," in *Fifth International Conference on Computer Vision*, vol. 2593, 1995.

[3] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection on non flat road geometry through 'v-disparity' representation," in *Proceedings of IEEE Intelligent Vehicle Symposium*, Versailles, France, 2002, http://perso.lcpc.fr/tarel.jean-philippe/iv02.html.

[4] R. Labayrade and D. Aubert, "A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision," in *IEEE Intelligent Vehicles Symposium*, Columbus, June 2003.

[5] N. Suganuma and N. Fujiwara, "An obstacle extraction method using virtual disparity image," in *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007, pp. 456–461.

[6] F. Oniga, S. Nedevschi, M. Meinecke, and T. Binh, "Road surface and obstacle detection based on elevation maps from dense stereo," in *IEEE Intelligent Transportation System*, Seattle, WA, USA, 2007.

[7] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, T. Graf, and R. Schmidt, "High accuracy stereovision approach for obstacle detection on non-planar roads," in *IEEE Intelligent Engineering Systems (INES)*, Romania, 2004, pp. 211–216.

[8] M. Unser, A. Aldroubi, and M. Eden, "B-Spline signal processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, February 1993.

[9] G. Welch and G. Bishop, "An introduction to the Kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 1995.

[10] L. Matthies and S. A. Shafer, "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 239–248, June 1987.

[11] S. Kubota, T. Nakano, and Y. Okamoto, "A global optimization algorithm for real-time on-board stereo obstacle detection systems," in *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007.

[12] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 249–265, June 1987.

[13] J. Miura, Y. Negishi, and Y. Shirai, "Mobile robot map generation by integrating omnidirectional stereo and laser range finder," in *Proc. of IROS*, 2002.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. The MIT Press, 2005.

[15] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[16] P. Cerri and P. Grisleri, "Free space detection on highways using time correlation between stabilized sub-pixel precision ipm images," in *Proc. of ICRA*, 2005.

[17] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV*, Rio de Janeiro, Brazil, October 2007.

[18] R. T. Collins, "A space-sweep approach to true multi-image matching," in *Proc. CVPR*, June 1996, pp. 358–363.

[19] T. Williamson and C. Thorpe, "Detection of small obstacles at long range using multibaseline stereo," in *Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles*, 1998.

[20] S. Gehrig and U. Franke, "Improving Stereo Sub-Pixel Accuracy for Long Range Stereo," in *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.*, Rio, Brasil, October 2007, pp. 1–7.

Uwe Franke received the Ph. D. degree in Electrical Engineering from the Technical University of Aachen, Germany, in 1988. Since 1989 he is with Daimler research working on vision based driver assistance systems. He developed lane keeping and Daimler's lane departure warning system and has been working on sensor fusion and truck platooning. His special interest is in image understanding in complex city scenarios and on real-time stereo analysis. Recent work is on optimal fusion of stereo and motion, called 6D-Vision and scene flow.

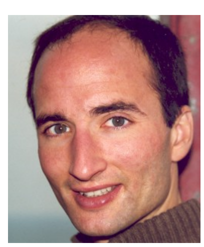
Since 2000 he is the head of the Daimler Environment Perception Group and concentrates on vision for increased traffic safety. Continuously working in the field of intelligent vehicles since 19 years, he is one of the most experienced experts all over the world. In 2002, he was program chair of the IEEE Intelligent Vehicles Conference in Versailles, France.

Andreas Wedel was born 1981 in Siegburg, Germany. He received B. Sc. degrees in Mathematics and Computer Science (2003), and a M. Sc. degree (2006) in Computer Vision from the University of Bonn, Germany. Since October 2005, he has been working with the Environment Perception Group at Daimler Research, Sindelfingen, and the Computer Vision Group at the University of Bonn; he is currently pursuing a Ph. D. degree in Computer Science. His research interests are in motion analysis, segmentation, and modeling of the vehicle environment.

Hernán Badino received his his M. Sc. degree in Software Engineering from the National Technological University, Córdoba, Argentina in 2002 and the Ph. D. degree from the Goethe University Frankfurt, Germany in 2008. He has worked with the Environment Perception Group at Daimler Research, Sindelfingen and is now with the Robotics Institute, Carnegie Mellon University, USA. His research interests include ego-motion from stereo imagery and the development of stereo vision algorithms for real-time detection and tracking of objects.

Clemens Rabe was born 1979 in Giessen, Germany. He received his diploma degree in Computer Engineering from the University of Applied Sciences Wuerzburg, Germany in 2005. He is currently pursuing a Ph. D. degree in Computer Science at the University of Kiel in cooperation with the Environment Perception Group at Daimler Research, Sindelfingen. His research interests are in stereo vision, motion analysis, and traffic scene understanding.

Heidi Loose received her diploma degree in Computer Science 2007 from the University of Tübingen, Germany. Since April 2007 she has been working at the Daimler Research Group for Environment Perception and since January 2008 she has been pursuing a Ph. D. degree in Engineering at the University of Karlsruhe. Her reasearch interests are in lane recognition and road modeling.



Daniel Cremers received Bachelor degrees in Mathematics (1994) and Physics (1994), and a Master's degree in Theoretical Physics (1997) from the University of Heidelberg. In 2002 he obtained a Ph. D. in Computer Science from the University of Mannheim, Germany. Subsequently he spent two years as a postdoctoral researcher at the University of California at Los Angeles (UCLA) and one year as a permanent researcher at Siemens Corporate Research in Princeton, NJ. Since October 2005 he heads the Research Group for Computer Vision, Image Processing and Pattern Recognition at the University of Bonn, Germany. His publications received several awards. Professor Cremers served on the program committee for ECCV, CVPR, and numerous other conferences and workshops. He regularly reviews for all major computer vision journals. He serves as a proposal reviewer for the German Science Foundation (DFG), the US National Science Foundation, the Israel Science Foundation, the Austrian Science Foundation and the Hungarian Science Foundation.