International Journal of Computer Vision (2019) 127:1643–1658
https://doi.org/10.1007/s11263-019-01226-9

Slanted Stixels: A Way to Represent Steep Streets

Daniel Hernandez-Juarez[1] · Lukas Schneider[2] · Pau Cebrian[1] · Antonio Espinosa[1] · David Vazquez[3] · Antonio M. López[1,4] · Uwe Franke[2] · Marc Pollefeys[5] · Juan C. Moure[1]

Received: 20 February 2018 / Accepted: 30 August 2019 / Published online: 14 September 2019

Abstract

This work presents and evaluates a novel compact scene representation based on Stixels that infers geometric and semantic information. Our approach overcomes the previous rather restrictive geometric assumptions for Stixels by introducing a novel depth model to account for non-flat roads and slanted objects. Both semantic and depth cues are used jointly to infer the scene representation in a sound global energy minimization formulation. Furthermore, a novel approximation scheme is introduced in order to significantly reduce the computational complexity of the Stixel algorithm, and then achieve real-time computation capabilities. The idea is to first perform an over-segmentation of the image, discarding the unlikely Stixel cuts, and apply the algorithm only on the remaining Stixel cuts. This work presents a novel over-segmentation strategy based on a fully convolutional network, which outperforms an approach based on using local extrema of the disparity map. We evaluate the proposed methods in terms of semantic and geometric accuracy as well as run-time on four publicly available benchmark datasets. Our approach maintains accuracy on flat road scene datasets while improving substantially on a novel non-flat road dataset.

Keywords Stereo vision · Stixel world · Self-driving cars · Scene understanding · Automotive vision · Intelligent vehicles

1 Introduction

Autonomous vehicles, advanced driver assistance systems, robots and other intelligent devices need to understand their environment. For this purpose, both geometric (distance) and semantic (classification) sources of information are useful. We want to represent these inputs in a very compact model and compute them in real-time to serve as a building block of higher-level modules, such as localization and planning.

This success has led to increased interest in the model from the intelligent vehicles community over the past years The Stixel world has been successfully used for representing traffic scenes, as introduced in Pfeiffer and Franke (2011). It has shown its potential particularly in the Bertha-Benz drive (Ziegler et al. 2014b), where it has been successfully applied for visual scene understanding in autonomous driving. This success has led to increased interest in the model from the intelligent vehicles community over the past years (Schneider et al. 2016; Hernandez-Juarez et al. 2017a; Benenson et al. 2011; Cordts et al. 2014, 2017; Ignat 2016; Levi et al. 2015; Carrillo and Sutherland 2016; Hernandez-Juarez et al. 2017b).

The Stixel world defines a compact medium-level representation of dense 3D disparity data obtained from stereo vision using rectangles, the so called *Stixels*, as elements. Stixels are classified either as *ground*-like planes, upright *objects* or *sky*, which are important geometric elements found in man-made environments. This representation transforms millions of disparity pixels to hundreds or thousands of Stixels. At the same time, most scene structures, such as free space and obstacles, which are relevant for autonomous driving tasks, are adequately represented.

Communicated by Tae-Kyun Kim, Stefanos Zafeiriou, Ben Glocker, Stefan Leutenegger.

Authors Daniel Hernandez-Juarez and Lukas Schneider have contributed equally to this work.

✉ Daniel Hernandez-Juarez
dhernandez0@gmail.com

[1] Universitat Autònoma de Barcelona (UAB), Barcelona, Spain

[2] Daimler AG R&D, Böblingen, Germany

[3] Element AI, Montreal, Canada

[4] Computer Vision Center (CVC), Barcelona, Spain

[5] ETH Zürich, Zürich, Switzerland

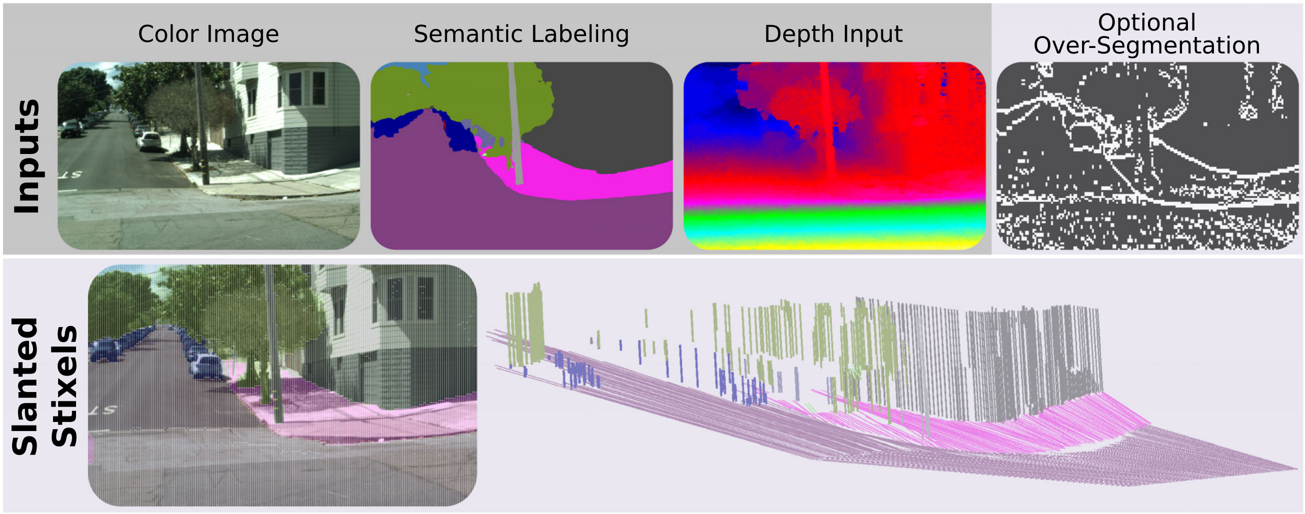


Fig. 1 The proposed approach: pixel-wise color, semantic and depth information serve as input to our Slanted Stixels model, which is a compact semantic representation of a 3D scene that accurately handles arbitrary scenarios such as San Francisco city. The optional over-segmentation in the top-right yields significant speed gains while nearly retaining the depth and semantic accuracy (Color figure online)

The idea behind the Stixel model is that planar surfaces are dominant in man-made environments and they can be modeled using this assumption. Scene structure found in urban environments can be modeled with certain constraints, e.g. the sky is above the horizon line and objects usually lie on the ground. Generally, the geometric constraints of a scene are tied to the vertical direction. Hence, the environment can be modeled as a column-wise segmentation of the image with a 3D stick-like shape, i.e. a set of Stixels, c.f. Fig. 1. The segmentation of the image is estimated by solving a column-wise energy minimization problem, taking depth and semantic cues as inputs as well as *a priori* information that is used to regularize the solution c.f. Fig. 1.

The Stixel model has been successfully used for automotive vision applications either to decrease parsing time, increase accuracy or both. We can find examples of works using the Stixel representation in different topics such as object recognition (Benenson et al. 2012; Li et al. 2016), building a grid map over time (Muffert et al. 2014) and for autonomous driving (Ziegler et al. 2014b). Specifically, for motion planning in the context of autonomous driving, the Stixel model has been used c.f. (Ziegler et al. 2014b, a) to model the geometric constraints of a given scene.

We propose a new depth model that is able to accurately represent arbitrary kinds of slanted objects and non-flat roads. The improved Stixel representation outperforms the original Stixel model in scenarios with non-flat roads, while keeping the same accuracy on flat road scenes. The induced extra computational complexity is reduced by incorporating an over-segmentation strategy that can be applied to any Stixel model proposed so far. An earlier version of our work (Hernandez-Juarez et al. 2017b) proposed a simple over-segmentation strategy that provided faster execution at the expense of decreasing the accuracy of the model. This paper introduces a novel over-segmentation approach based on a Fully Convolutional Network (FCN) that outperforms the previous strategy, and achieves similar speedup results but retaining most of the accuracy of the original version. An overview of our method is shown in Fig. 1.

Our main contributions are: (1) a novel depth model to accurately represent arbitrary kinds of slanted surfaces into the Stixel representation; (2) a novel over-segmentation prior designed to reduce the run-time of the method; (3) an effective over-segmentation strategy based on a shallow Fully Convolutional Network; (4) a new synthetic dataset with non-flat roads that includes pixel-level semantic and depth ground-truth, which is publicly available[1]; and (5) an in-depth evaluation in terms of run-time as well as semantic and depth accuracy carried out on this novel dataset and several real-world benchmarks. Compared to the existing state-of-the-art approaches, our method substantially improves the depth accuracy in non-flat road scenarios.

The remainder of this paper is structured as follows. Section 2 reviews the state of the art. Section 3 presents the new Stixel formulation. We present two over-segmentation methods in Sect. 4. Section 5 explains the experiments we carried on and discusses their results. Finally, we state our conclusions in Sect. 6.

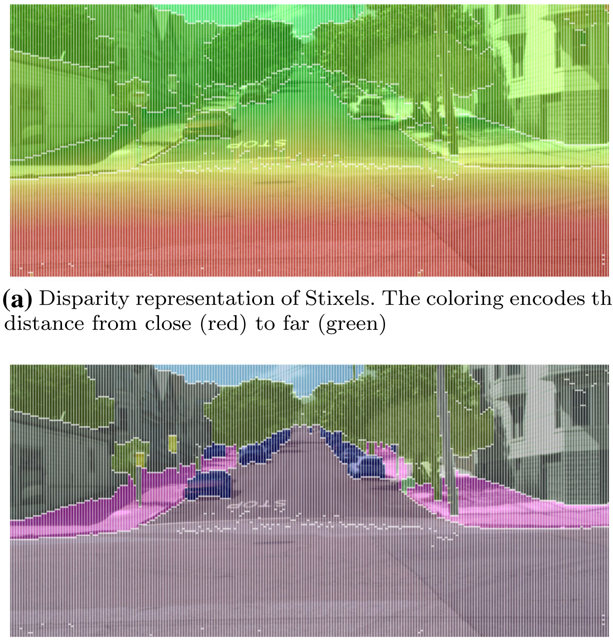
[1] http://synthia-dataset.net.

2 Related Work

Our proposed method introduces a novel Stixel-based scene representation that is able to account for non-flat roads, c.f. Fig. 2. We also devise an approximation to reduce the computational complexity of the underlying Dynamic Programming algorithm.

First, we will comment on works proposing different road scene models. Occupancy grid maps are models used to represent the surroundings of the vehicle (Dhiman et al. 2014; Muffert et al. 2014; Nuss et al. 2015; Thrun 2002). Typically, a grid in bird's eye perspective is defined and used to detect occupied grid cells and then, from this information, to extract the obstacles, drivable area, and unobservable areas from range data. These grids and the Stixel world both represent the 2D image in terms of column-wise stripes allowing to capture the camera data in a polar fashion. Also, the Stixel data model is similar to the forward step usually found in occupancy grid maps (Cordts et al. 2017). However, the Stixel inference method in the image domain presents important differences compared to classical grid-based approaches.

Our work builds upon the proposal from Schneider et al. (2016): they use semantic cues in addition to depth to extract a Stixel representation, which is able to provide a rich yet compact representation of the traffic scene. However, their model assumes a constant road slant and is therefore limited to flat road scenarios. In contrast, our proposal overcomes this drawback by incorporating a novel plane model together with effective priors on the plane parameters.



(a) Disparity representation of Stixels. The coloring encodes the distance from close (red) to far (green)

(b) Semantic representation of Stixels. The coloring encodes the semantic class following Cordts et al (2016)

Fig. 2 Scene representation obtained by our method of a challenging street environment with a slanted road. Both geometric (top) and semantic (bottom) representations are shown (Color figure online)

Our proposal of using Stixels cuts is related to Cordts et al. (2014): they use fast object detectors for different object classes, e.g. Viola-Jones cascade detector (Viola and Jones 2001), to produce top and bottom Stixel cuts that are used as prior information, which is then integrated into the Stixel algorithm. They prove that using object-level knowledge provides significant accuracy improvements. Instead, we leverage semantic information as pixel-level knowledge in our model for the same purpose of improving accuracy. Semantic segmentation identifies the objects and other elements of the image, e.g. walls or sidewalks, providing pixel-level information, instead of boxes around the objects. Also, semantic segmentation requires a single predictor, while the method proposed by Cordts et al. (2014) needs a detector trained for each object class. In contrast, we define a Stixel cut prior to generate an over-segmentation of the optimal Stixel cuts in order to speed up the execution of the algorithm.

There are some methods (Benenson et al. 2011; Ignat 2016; Levi et al. 2015), that represent simplified scene models with a single Stixel per column. The advantage of these approaches is that the computational complexity of the underlying algorithms is linear, but they cannot represent some complex scenarios found in the real world, e.g. a pedestrian and a building in the same column.

Recent work by Carrillo and Sutherland (2016) uses edge-based disparity maps to compute Stixels. Their method is fast but they show that it gives inferior accuracy compared to the original Stixel model (Pfeiffer et al. 2013).

Levi et al. (2015) firstly introduced the use of an FCN in Stixel-based methods. A single RGB image feeds the FCN to estimate the bottom of the first non-road Stixel, i.e. closest obstacle. We use an FCN for a entirely different objective: to extract a Stixel cut over-segmentation that accelerates the execution of the algorithm. Moreover, the input of our FCN is a disparity map obtained from a stereo camera.

Finally, there are some works proposing fast implementations for Stixel computation. The FPGA implementation from Muffert et al. (2014) runs at 25 Hz with a Stixel width of 5 pixels, but the authors do not indicate the image resolution. Hernandez-Juarez et al. (2017a) present a GPU-accelerated implementation that runs at 26 Hz for a Stixel width of 5 pixels and image resolution of 1024 × 440 pixels, computed using a Semi-Global Matching (SGM) (Hirschmüller 2008) stereo algorithm. We propose a novel approximation that accelerates the computation by reducing the algorithmic complexity. Accordingly, our proposal could benefit from the aforementioned FPGA- or GPU-accelerated implementations.

3 Stixel Model

The Stixel world is a compressed representation of a 3D scene that preserves its relevant structure. Since the structure in street environments is dominant in the vertical domain, the Stixel world leverages this idea to model a scene without taking into account the horizontal neighborhood. This assumption leads to an efficient inference method and also allows the inference to be performed on all columns in parallel.

The Stixel world is defined as a segmentation of image columns into stick-like super-pixels with class labels and a 3D planar depth model c.f. Fig. 3. We consider three structural classes: *object*, *ground* and *sky*. These classes have properties that are derived from an underlying 3D model: for *object* Stixels the distance is roughly constant and usually lie on the ground, for *sky* Stixels the distance is infinite and for *ground* Stixels we favor planes with accordance to the expected ground.

The Stixel world has several properties that are useful for higher-level processing stages: (1) it is a medium-level scene representation that significantly reduces the quantity of elements, e.g. from millions of pixels to hundreds of Stixels, while keeping an abstract representation of physical extent, depth and semantics; (2) the representation is based upon a street model; (3) the representation is not high-level because an object is represented by more than one Stixel horizontally and it can be split in more than one Stixel vertically too, e.g. occlusions and slanted objects such as cars viewed from behind.

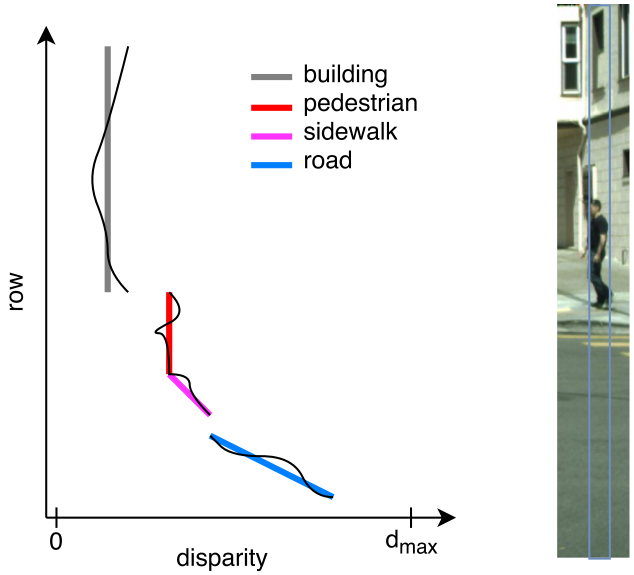


Fig. 3 Example of input disparity measurements (black lines) and output Stixels encoded with semantic colors (colored lines) for a typical scene column (right). Adapted from Cordts et al. (2017) (Color figure online)

The joint Stixel segmentation and labeling problem is carried out via optimization of the column-wise posterior distribution $P(\boldsymbol{S}_{:} \mid \boldsymbol{M}_{:})$ defined over a Stixel segmentation $\boldsymbol{S}_{:}$ given all measurements $\boldsymbol{M}_{:}$ from that particular image column. In the following, we drop the column indexes for ease of notation. We obtain Stixel width > 1 as illustrated e.g. in Fig. 1 by down-sampling of the inputs, this width is fixed and is chosen to reduce the computational complexity during inference, however heavy down-sampling leads to degradation in accuracy (Cordts et al. 2017).

A Stixel column segmentation consists of an arbitrary number N of Stixels $\boldsymbol{S}_i$, each representing four random variables: the Stixel extent via bottom V_i^{b} and top V_i^{t} row, as well as its class C_i and geometric depth model G_i. Thereby, the number of Stixels itself is a random variable that is optimized jointly during inference. To this end, the posterior probability is defined by means of the unnormalized prior and likelihood distributions

$$P(\boldsymbol{S} \mid \boldsymbol{M}) = \frac{1}{Z} \tilde{P}(\boldsymbol{M} \mid \boldsymbol{S})\, \tilde{P}(\boldsymbol{S}) \tag{1}$$

where Z is the normalizing partition function. Transformed to log-likelihoods via

$$P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{M} = \boldsymbol{m}) = -\log(e^{-E(\boldsymbol{s},\boldsymbol{m})}) \tag{2}$$

where $E(\cdot)$ is the energy function, $E_{data}(\cdot)$ is the **likelihood** term and $E_{prior}(\cdot)$ is the **prior** term.

$$E(\boldsymbol{s}, \boldsymbol{m}) = E_{data}(\boldsymbol{s}, \boldsymbol{m}) + E_{prior}(\boldsymbol{s}) \tag{3}$$

3.1 Data Term

The **likelihood** term $E_{data}(\cdot)$ thereby rates how well the measurements $\boldsymbol{m}_v$ at pixel v fit to the overlapping Stixel $\boldsymbol{s}_i$

$$\begin{aligned} E_{data}(\boldsymbol{s}, \boldsymbol{m}) &= \sum_{i=1}^{N} E_{stixel}(\boldsymbol{s}_i, \boldsymbol{m}) \\ &= \sum_{i=1}^{N} \sum_{v=v_i^{\mathrm{b}}}^{v_i^{\mathrm{t}}} E_{pixel}(\boldsymbol{s}_i, \boldsymbol{m}_v)\ . \end{aligned} \tag{4}$$

This pixel-wise energy is further split in a semantic and a depth term

$$E_{pixel}(\boldsymbol{s}_i, \boldsymbol{m}_v) = E_{disp}(\boldsymbol{s}_i, d_v) + w_l \cdot E_{sem}(\boldsymbol{s}_i, l_v)\,. \tag{5}$$

The parameter w_l controls the influence of the semantic data term. The input is provided by an FCN that delivers normalized semantic scores $l_v(c_i)$ with $\sum_{c_i} l_v(c_i) = 1$ for all classes c_i at pixels v. The semantic energy favors semantic classes of the Stixel that fit to the observed pixel-level semantic input (Schneider et al. 2016). The semantic likelihood term is

$$E_{sem}(\boldsymbol{s}_i, l_v) = -log(l_v(c_i)) \ . \tag{6}$$

The depth model is designed to represent the different characteristics of the different geometric classes, i.e. *object*, *ground* and *sky* Stixels. Furthermore, the model enforces multiple stacked Stixels in cases of objects with the same class but different depths.

Our depth input is a dense disparity map, each pixel is assigned a disparity value or is masked as invalid i.e. $d_v \in \{0...d_{max}, d_{invalid}\}$. The depth term is defined by means of a probabilistic and generative sensor model $P_v(\cdot)$ that considers the accordance of the depth measurement d_v at row v to the Stixel $\boldsymbol{s}_i$

$$E_{disp}(\boldsymbol{s}_i, d_v) = -\log(P_v(D_v = d_v \mid \boldsymbol{S}_i = \boldsymbol{s}_i)) \ . \tag{7}$$

Invalid d_{inv} disparity measurements have to be handled, therefore, a prior probability of a valid disparity value is defined as p_{val}

$$P_v(D_v \mid \boldsymbol{S}_i) = \begin{cases} p_{val} P_{v,val}(D_v \mid \boldsymbol{S}_i) & \text{if } d_v \neq d_{inv} \\ (1 - p_{val}) & \text{otherwise} \end{cases} \tag{8}$$

where $P_{v,val}(D_v \mid \boldsymbol{S}_i)$ is the measurement model of valid disparities only. It is comprised of a constant outlier probability p_{out} and a Gaussian sensor noise model for valid measurements with confidence c_v

$$P_{v,val}(D_v \mid \boldsymbol{S}_i) = \frac{p_{\text{out}}}{Z_U} + \frac{1 - p_{\text{out}}}{Z_G(\boldsymbol{s}_i)} e^{-\left(\frac{c_v(d_v - \mu(\boldsymbol{s}_i, v))}{\sigma(\boldsymbol{s}_i)}\right)^2} \tag{9}$$

that is centered at the expected disparity $\mu(\boldsymbol{s}_i, v)$ given by the depth model of the Stixel, where Z_U and $Z_G(\boldsymbol{s}_i)$ normalize the distributions. Similarly to Pfeiffer et al. (2013), we use the confidence of the depth estimates c_v to influence the shape of the distribution. The Gaussian models the typical disparity noise and the uniform distribution makes the model more robust to outliers, which is weighted by p_{out}. The standard deviation $\sigma(\boldsymbol{s}_i)$ models the noise of the stereo matching algorithm and depends on the class c_i.

3.1.1 New Depth Model

The depth model defines the 3D outline of a Stixel using very few parameters per Stixel and reflects our assumptions on the surrounding scene. Both, data term (c.f. eq. (9)) and priors (c.f. Sect. 3.2) have a significant impact on the inferred depth model. In previous formulations, the three different geometric classes were designed using restrictive constant height (ground Stixels) and constant depth (object and sky Stixels), assumptions per Stixel, e.g. for object Stixels: $\mu(\boldsymbol{s}_i, v) = constant$.

This paper introduces a new plane depth model that relaxes the previous assumptions in favor of a more accurate depth representation. The new model is formulated such that it nicely interacts with this well founded and experimentally validated depth sensor model. To this end, we formulate the depth model $\mu(\boldsymbol{s}_i, v)$ using two random variables defining a plane in the disparity space that evaluates to the disparity in row v via

$$\mu(\boldsymbol{s}_i, v) = b_i \cdot v + a_i \ . \tag{10}$$

Note that we assume narrow Stixels and thus can neglect one plane parameter, i.e. the roll.

This model is a generalization of the previous class-specific depth models used in previous works, allowing for a more flexible representation of the scene because of the extra free parameter c.f. Fig. 4. The way of modeling the different Stixel classes i.e. *object*, *ground* and *sky* is through priors, as explained in Sect. 3.2.5. Also, to completely understand the details about the inference, we suggest to read Sect. 3.3.

3.2 Prior Term

The **prior** captures knowledge about the segmentation independent from measurements, in this section we define the priors used for this model, they are based on Cordts et al. (2017). The Markov property is used so that the prior reduces to pair-wise relations between subsequent Stixels. Accordingly, the prior is computed as

$$E_{prior}(\boldsymbol{s}) = \sum_{i=2}^{N} E_{pair}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1}) + E_{first}(\boldsymbol{s}_1) \ . \tag{11}$$

In the next sections, where different priors are introduced, $E_{pair}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1})$ is the summation of all these priors. However, $E_{first}(\boldsymbol{s}_1)$ does not include pairwise terms, i.e.

$$\begin{aligned} E_{first}(\boldsymbol{s}_1) = E_{mc}(\boldsymbol{s}_1) + E_{segfirst}(\boldsymbol{s}_1) + E_{seglast}(\boldsymbol{s}_1) \\ + E_{top \geq bottom}(\boldsymbol{s}_i) + E_{plane}(\boldsymbol{s}_1) \end{aligned} \tag{12}$$

3.2.1 Model Complexity Prior

A model complexity term favors solutions composed of fewer Stixels and thus invokes costs for each Stixel in the column segmentation $\boldsymbol{S}$:

$$E_{mc}(\boldsymbol{s}_i) = C_{mc} \ . \tag{13}$$

There is a trade-off between compactness and accuracy. A high C_{mc} parameter would lead to a very compact segmentation i.e. few Stixels. However, a representation with few

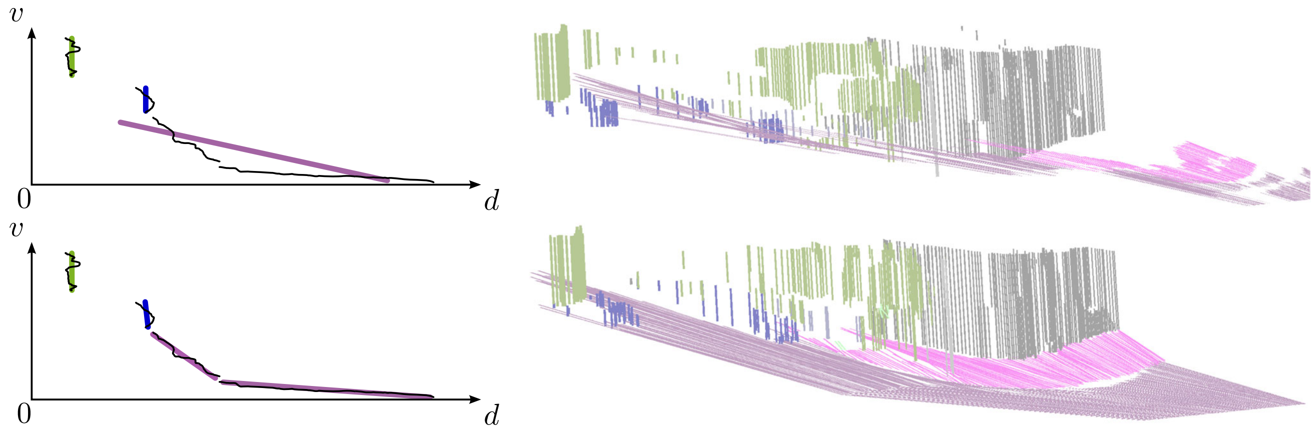


Fig. 4 Comparison of original (Schneider et al. 2016) (top) and our Slanted Stixels (bottom): due to the fixed slant in the original formulation, the road surface is not well represented as illustrated on the top-left figure. The novel model is capable of reconstructing the whole scene accurately

Stixels is more likely to have lower accuracy, e.g. a solution comprised of one Stixel the size of the whole column would result in a huge disparity and semantic error.

3.2.2 Segmentation Priors

The model has to enforce that all pixels are assigned to exactly one Stixel, i.e. non-overlapping Stixels, Stixels extend over all the column and Stixels are connected. Therefore, the first priors are defined to comply with the following rules: The first Stixel must begin in row 1 and the last Stixel must end in row h, i.e.

$$E_{segfirst}(\boldsymbol{s}_i) = \begin{cases} \infty & \text{if } v_i^b \neq 1, \text{i} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$E_{seglast}(\boldsymbol{s}_i) = \begin{cases} \infty & \text{if } v_i^t \neq h, \text{i} = n \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

Furthermore, every Stixel must be connected to the next one and the Stixel top row must be greater than the bottom row, i.e.

$$E_{connection}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1}) = \begin{cases} 0 & \text{if } v_i^b = v_{i-1}^t + 1 \\ \infty & \text{otherwise} \end{cases} \tag{16}$$

$$E_{top \geq bottom}(\boldsymbol{s}_i) = \begin{cases} 0 & \text{if } v_i^b \leq v_i^t \\ \infty & \text{otherwise} \end{cases}. \tag{17}$$

3.2.3 Structural Priors

The gravity prior penalizes a flying object i.e. an *object* Stixel not lying on top of the previous *ground* Stixel,

$$E_{gravity}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1}) = \begin{cases} \alpha^-_{gravity} + \beta^-_{gravity}\Delta_g & \text{if } \Delta_g < 0 \\ \alpha^+_{gravity} + \beta^+_{gravity}\Delta_g & \text{if } \Delta_g > 0 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where $\Delta_g = \mu_s(v_i^b, g_i) - \mu_s(v_{i-1}^t, g_{i-1})$ is the difference between the *object* Stixel disparity $\mu_s(v_i^b, g_i)$ at it's bottom pixel v_i^b and the disparity of the *ground* Stixel $\mu_s(v_{i-1}^t, g_{i-1})$ at the top row v_i^t. It only applies for s_i being an object and s_{i-1} being a ground Stixel.

The depth ordering prior penalizes a combination of two staggered *object* Stixels when the upper of the two is closer (in distance to the car) than the lower one.

$$E_{ord}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1}) = \begin{cases} \alpha_{ord} + \beta_{ord}(g_i - g_{i-1}) & \text{if } g_i > g_{i-1} \\ 0 & \text{otherwise} \end{cases}. \tag{19}$$

A novel prior is introduced in this paper: the ground gap prior penalizes two consecutive *ground* Stixels when the bottom disparity of the upper Stixel i.e. disparity at row v_i^b and the disparity of the lower Stixel at row v_i^b do not match.

$$E_{gap}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1}) = \begin{cases} \alpha^-_{gap} + \beta^-_{gap}\Delta_{gap} & \text{if } \Delta_{gap} < 0 \\ \alpha^+_{gap} + \beta^+_{gap}\Delta_{gap} & \text{if } \Delta_{gap} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where $\Delta_{gap} = g_s(v_i^b, g_i) - g_s(v_i^b, g_{i-1})$. These structural priors do not enforce their assumptions. Instead, they penalize unusual combinations, e.g. a flying object (gravity prior), traffic signs (ordering prior).

3.2.4 Transition Priors

These priors define the knowledge regarding the transition between a pair of Stixels.

$$E_{transition}(\boldsymbol{s}_i, \boldsymbol{s}_{i-1}) = \gamma_{c_i, c_{i-1}} \tag{21}$$

where $\gamma_{c_i, c_{i-1}}$ is the transition cost between previous Stixel class c_{i-1} to current Stixel class c_i. This is defined via a two-dimensional transition matrix for all combinations of classes $\gamma_{c_i, c_{i-1}}$. Only first order relations are modeled in order to infer efficiently.

3.2.5 Plane Prior

In this paper, we propose a new additional prior term that uses the specific properties of the three geometric classes. We expect the two random variables A, B representing the plane parameters of a Stixel to be Gaussian distributed, i.e.

$$E_{plane}(\boldsymbol{s}_i) = \left(\frac{a_i - \mu^a_{c_i}}{\sigma^a_{c_i}} \right)^2 + \left(\frac{b_i - \mu^b_{c_i}}{\sigma^b_{c_i}} \right)^2 - \log(Z) \ . \tag{22}$$

This prior favors planes in accordance to the expected 3D layout corresponding to the geometric class. For instance, *object* Stixels are expected to have an approximately constant disparity, i.e. $\mu^b_{object} = 0$. The expected road slant μ^b_{ground} can be set using prior knowledge or a preceding road surface detection. For *sky* Stixels we expect infinite distance i.e. 0 disparity, therefore, we set $\mu^a_{sky} = \mu^b_{sky} = 0$.

The standard deviations $\sigma^a_{c_i}$ and $\sigma^b_{c_i}$ are used in order to enforce the assumptions for each Stixel class, i.e. the more confident we are that *object* Stixels have constant distance, the closer to 0 we would set σ^b_{object}. The same applies for *ground* Stixels: if we know the road is not slanted, we can rely on the expected previous road model and set $\sigma^b_{ground} \rightarrow 0$. For *sky* Stixels, it does not make sense to have a disparity different to 0. Therefore, we set $\sigma^a_{sky} \rightarrow 0$ and $\sigma^b_{sky} \rightarrow 0$.

Note that the novel formulation is a strict generalization of the original method, since they are equivalent, e.g. if the slant is fixed, i.e. $\sigma^b_{object} \rightarrow 0$, $\mu^b_{object} = 0$.

3.3 Inference

The sophisticated energy function defined in Sect. 3 is optimized via Dynamic Programming as in Pfeiffer and Franke (2011). However, we must also optimize jointly for the novel depth model. When optimizing for the plane parameters a_i, b_i of a certain Stixel $\boldsymbol{s}_i$, it becomes apparent that all other optimization parameters are independent of the actual choice of the plane parameters. We can thus simplify

$$\underset{a_i, b_i}{\operatorname{argmin}}\ E(\boldsymbol{s}, \boldsymbol{m}) = \underset{a_i, b_i}{\operatorname{argmin}}\ E_{stixel}(\boldsymbol{s}_i, \boldsymbol{m}) + E_{plane}(\boldsymbol{s}_i) \ . \tag{23}$$

Thus, we minimize the global energy function with respect to the plane parameters of all Stixels and all geometric classes independently. We can find an optimal solution of the resulting weighted least squares problem in closed form. However, we still need to compare the Stixel measurements to our new plane depth model. Therefore, the complexity added to the original formulation is another quadratic term in the image height.

3.4 Stixel Cut Prior

The Stixel inference process described so far requires the estimation of the cost for each possible Stixel in an image. However, many Stixels can be trivially discarded, e.g. in image regions with homogeneous depth and semantic input, making it possible to avoid the computation steps associated to the calculation of these.

We propose a novel prior that exploits hypothesis generation to significantly reduce the computational burden of the inference task. To this end, we formulate a new prior similar to Cordts et al. (2014); however, instead of Stixel bottom and top probabilities, we incorporate generic likelihoods for pixels being the cut between two Stixels.

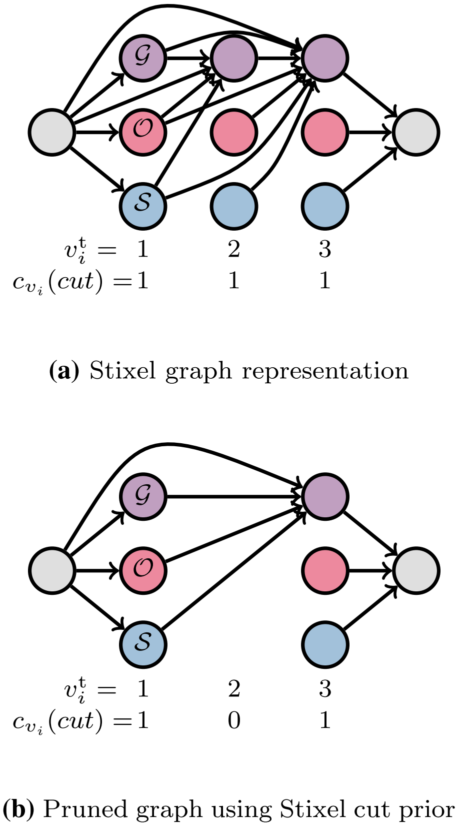
We leverage this additional information adding a novel prior term for a Stixel $\boldsymbol{s}_i$

$$E_{cut}(\boldsymbol{s}_i) = -\log\big(c_{v_i}(cut)\big) \tag{24}$$

where $c_{v_i}(cut)$ is the confidence for a cut at v_i, thus $c_{v_i}(cut) = 0$ implies that there is no cut between two Stixels at row v.

As described in Pfeiffer (2014), we can design a recursive definition of the optimization problem in order to solve the problem using a Dynamic Programming scheme. In order to simplify our description, we use a special notation to refer to Stixels: $ob^t_b = \{v^b, v^t, object\}$. Similarly, OB^k is defined as the minimum aggregated cost of the best segmentation from position 0 to k. The Stixel at the end of the segmentation associated with each minimum cost is denoted as ob^k. We next show a recursive definition of the problem:

$$OB^k = min \begin{cases} E_{data}(ob^k_0) & + E_{prior}(ob^k_0) \\ E_{data}(ob^k_x) & + E_{prior}(ob^k_x, ob^{x-1}) \\ & \quad + OB^{x-1} \forall x \in cuts, x \leq k \\ E_{data}(ob^k_x) & + E_{prior}(ob^k_x, gr^{x-1}) \\ & \quad + GR^{x-1} \forall x \in cuts, x \leq k \\ E_{data}(ob^k_x) & + E_{prior}(ob^k_x, sk^{x-1}) \\ & \quad + SK^{x-1} \forall x \in cuts, x \leq k \end{cases} . \tag{25}$$



(a) Stixel graph representation

(b) Pruned graph using Stixel cut prior

Fig. 5 Stixel inference illustrated as shortest path problem on a directed acyclic graph: the Stixel segmentation is computed by finding the shortest path from the source (left gray node) to the sink (right gray node). The vertices represent Stixels with colors encoding their geometric class, i.e. **g**round, **o**bject and **s**ky. Only the incoming edges of ground nodes are shown for simplicity. Adapted from Cordts et al. (2017) (Color figure online)

We only show the case for *object* Stixels, but the other cases are solved similarly. Also, GR^k and SK^k stand for *ground* and *sky* respectively. The base case problem, i.e. segmenting a column of the single pixel at the bottom, is defined: $OB^0 = E_{data}(ob_0^0) + E_{prior}(ob_0^0)$. Our method trusts that all the optimal cuts will be included in our over-segmentation [$cuts$ in Eq. (25)], therefore, only those positions are checked as Stixel bottom and top. This reduces the complexity of the Stixel estimation problem for a single column to $\mathcal{O}(h' \times h')$, where h' is the number of over-segmentation cuts computed for this column, h is image height and $h' \ll h$.

The computational complexity reduction becomes apparent in Fig. 5. As stated in Cordts et al. (2017), the inference problem can be interpreted as finding the shortest path in a directed acyclic graph. Our approach prunes all the vertices associated with the Stixel's top row not included according to the Stixel cut prior, c.f. Fig. 5b.

4 Generation of the Stixel Cut Prior

The previous section explained how to use a Stixel cut prior to reduce the computational complexity of the Stixel inference. The idea is that many Stixel cuts could be trivially discarded, e.g. in image regions with homogeneous depth and semantic input. We can save a lot of computation by not processing those unlikely Stixel cuts. The goal is to devise a fast method to generate an over-segmentation of the optimal Stixel cuts. And, if those optimal cuts are included in the generated hypothesis, then the Stixel algorithm will provide the same output as in the original case, but doing much fewer computation steps.

We propose two methods to generate Stixel cuts. The first method is a simple strategy that uses some mathematical concepts to identify points of interest c.f. Sect. 4.1. It is a very fast approach, but misses some of the optimal Stixel cuts and, therefore, the final accuracy of the Stixel inference is reduced. The second method uses a shallow Fully Convolutional Network (FCN) that is trained on the disparity map to infer likely Stixel cuts c.f. Sect. 4.2. This strategy is also very fast, since the FCN is small, and is able to provide almost all of the optimal Stixel cuts. For both methods, we leverage semantic segmentation information by including the edges of the semantic image into the set of the generated Stixel cuts.

4.1 Time Series Compression

The first method to generate Stixel cuts is based on the work of Ignat (2016), and has linear time complexity and linear memory requirements. In their work, each column of the disparity map is treated independently as a time series, i.e. a signal with measurements on equal intervals of time. They first perform an *extreme points detection* step that generates a list of possible Stixel cuts, and then apply subsequent filters to this list in order to generate the final Stixel segmentation. As we want to obtain an over-segmentation containing all the optimal Stixel cuts, we only use the first step of their proposal.

The detection of extreme points is based on techniques for time series compression (Fink and Gandhi 2011). A time series can be compressed by selecting local extreme points, i.e. maxima and minima of a function within a range. The assumption is that local extreme points are enough to find the important parts of the signal, and the rest would be unimportant points or noise.

In Ignat (2016) only left and right extrema are selected, while other kinds of extrema are discarded. Given a time series $\{t_1, t_2, \ldots, t_i, \ldots, t_{n-1}, t_n\}$ and point t_i with $1 < i < n$, the definition of left and right minimum is as follows (the definition of maxima is symmetric):

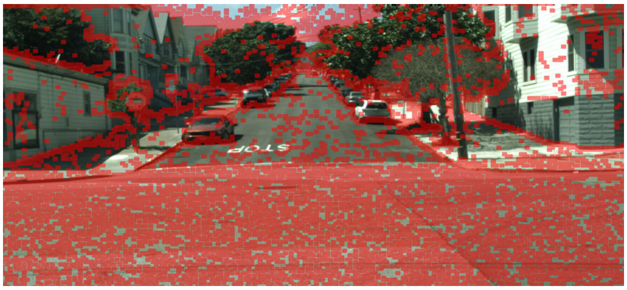


Fig. 6 Generated Stixel cuts (highlighted in red) using the left and right extrema as defined in Ignat (2016), and also cuts generated from semantic segmentation. Stixel cut density is 30%, equivalent to a 3.3× reduction in vertical resolution (Color figure online)

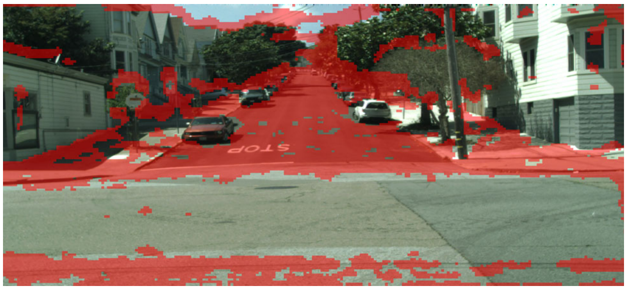


Fig. 7 Generated Stixel cuts (highlighted in red) for the FCN-based method. Stixel cut density is 31.5%, equivalent to a 3.2× reduction in vertical resolution (Color figure online)

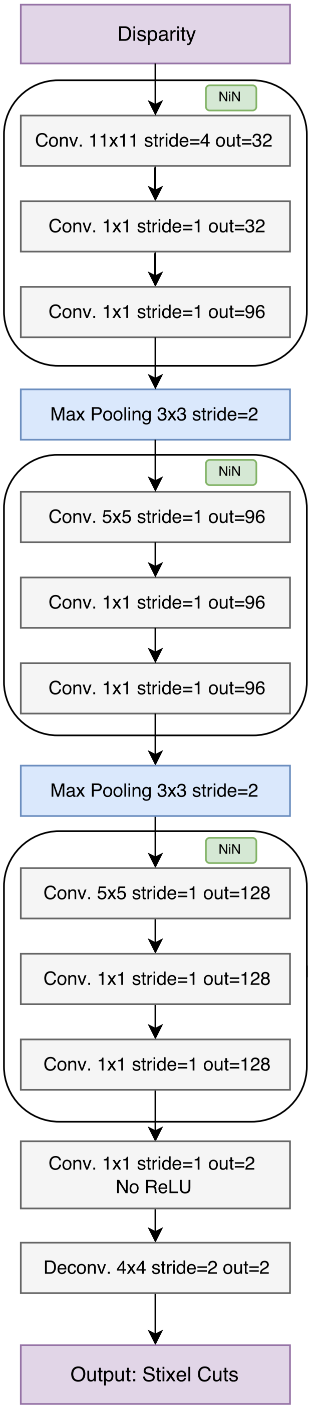
- t_i is left minimum if $t_i < t_{i-1}$ and there is t_j such that $j > i$ and $t_i = \ldots = t_j < t_{j+1}$.
- t_i is right minimum if $t_i < t_{i+1}$ and there is t_j such that $j < i$ and $t_{j-1} > t_j = \ldots = t_i$.

Similarly, we generate Stixel cuts by finding left and right extrema and the first and last points of the sequence of pixels in the column. The example in Fig. 6 illustrates the method. The predicted Stixel cuts are indicated in red color. In the example the vertical resolution is reduced around 3.3 times, which implies reduced computational work for the Stixel inference task.

4.2 FCN-Based Method

We propose a novel shallow deep neural network c.f. Fig. 8 that generates a set of promising Stixel cuts from depth images c.f. Fig. 7. We follow the proposal in Jasch et al. (2018): we use disparities instead of depth. We have experimentally found that adding the RGB image to the input of the neural network does not improve the accuracy of the method, compared to the simpler and faster strategy of directly adding the edges of the semantic image into the set of the generated Stixel cuts.

We design the network to provide an over-segmentation of the optimal Stixel cuts that should be significantly smaller than the total number of potential Stixel cuts (which is the height of the image). Also, the computational work required for the network inference must be small, ideally similar to the Time Series method proposed in Sect. 4.1. In the remainder of this section, we will first discuss the proposed network architecture, and then describe the data and training strategy.



Disparity
NiN
Conv. 11x11 stride=4 out=32
Conv. 1x1 stride=1 out=32
Conv. 1x1 stride=1 out=96
Max Pooling 3x3 stride=2
NiN
Conv. 5x5 stride=1 out=96
Conv. 1x1 stride=1 out=96
Conv. 1x1 stride=1 out=96
Max Pooling 3x3 stride=2
NiN
Conv. 5x5 stride=1 out=128
Conv. 1x1 stride=1 out=128
Conv. 1x1 stride=1 out=128
Conv. 1x1 stride=1 out=2
No ReLU
Deconv. 4x4 stride=2 out=2
Output: Stixel Cuts

Fig. 8 Definition of the proposed Fully Convolutional Network for generating Stixel cuts

4.2.1 Network Architecture

Our proposal is based on the architecture described by Schneider et al. (2017). They present a multi-modal FCN designed for semantic segmentation with a mid-level fusion architecture that exploits complementary input cues, i.e. RGB and disparity data. Their design includes the Network in Network (NiN) method proposed by Lin et al. (2013). Our proposal inherits the network branch that processes the disparity data and discards the branch on the RGB data, which is described in detail in Fig. 8. The proposed FCN is a very shallow network with three consecutive NiNs, and a final deconvolution that recovers the desired resolution of the Stixel cuts. The output of the FCN is a binary image indicating whether or not there is a Stixel cut for that pixel.

4.2.2 Training Data

We trained the proposed FCN using disparity maps generated from images in the Synthia synthetic dataset (Ros et al. 2016) and from images in a real-data sequence (6757 images) recorded in San Francisco, c.f. Fig. 9. In both cases, the disparity maps are generated from the left and right RGB images using a stereo matching algorithm (Hirschmüller 2008). This is the expected situation in a realistic scenario, where the SGM algorithm in the perception pipeline generates the disparity map and feeds the FCN that produces the Stixel over-segmentation.

The ground-truth for the training data (the expected Stixel cuts) is generated as a combination of methods. In the case of the annotated synthetic dataset, which contains both pixel-level semantic and instance-level annotations, the ground-truth includes, as desired Stixel cuts, the boundaries of the instances and the semantic classes in the image (as in Cordts et al. 2017). Finally, the Stixel cuts associated to disparity changes are obtained by running the Stixel inference method. In the real-data sequence, we only perform this last step because we lack ground-truth.

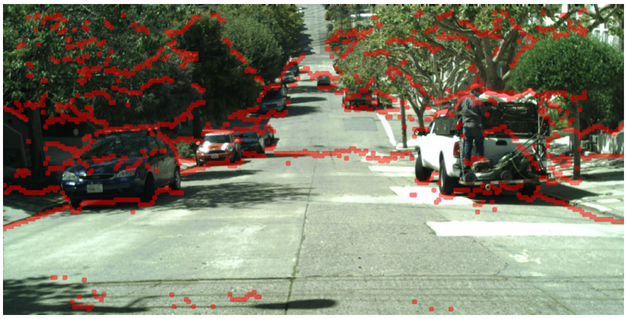


Fig. 9 Sample image from the real-data sequence used for Stixel cut generation. Stixel cut ground-truth is highlighted in red (Color figure online)

As discussed previously c.f. Sect. 3.2.1, the definition of the parameters of the Stixel model represent a trade-off between compactness and accuracy. Since we need an over-segmentation of the optimal Stixel cuts, we adjust the parameters of the model to be conservative and to favor accuracy versus compactness.

The idea of using the Stixel model as a way to train a fast and simple neural network to approximate the optimal Stixel segmentation is inspired by model distillation techniques (Bucila et al. 2006). The comparatively slow Dynamic Programming method to solve the probabilistic model is used to transfer the knowledge inside the complex model to a fast and compact FCN that approximates the optimal Stixel cuts.

4.2.3 Training Strategies

Since our problem is to classify each pixel of our input disparity map as *cut* or *not-cut*, we use cross-entropy as the loss function that must be minimized. The distribution of *cut*/*not-cut* is strongly biased in our input and, accordingly, we introduce a class-balancing weight in the loss function, similarly to Xie and Tu (2017). These weights cause the FCN to generate wider edges c.f. Fig. 7. This is useful, since the FCN roughly detects the Stixel cut positions, and the precise detection is left to the Stixel inference.

We set the learning rate to 10^{-8} and the batch size to five: four of those inputs are Synthia images and one of them is a real-data image. The missing disparities are encoded as -1. Input normalization is done by subtracting the mean value from the disparity map. We initialize the FCN with the weights used in Schneider et al. (2017), since semantic segmentation is a similar problem.

5 Experiments

This section assesses the accuracy and run-time of our proposal. A previous concern is to verify that our method not only improves the representation of scenes with non-flat roads, but also maintains the accuracy for scenes containing only flat roads. For that purpose, we present datasets of synthetic and real data to evaluate our proposal in Sect. 5.1. We introduce inputs, metrics, baselines, and other experimental details in Sect. 5.2. Finally, quantitative and qualitative results are reported in Sect. 5.3.

5.1 Datasets

As our Stixel model represents geometric and semantic information, we must evaluate the accuracy of our method for both. For that purpose, we select *Ladicky* (Ladicky et al. 2014), an annotated subset of *KITTI* (Geiger et al. 2012), which is, to the best of our knowledge, the only dataset con-

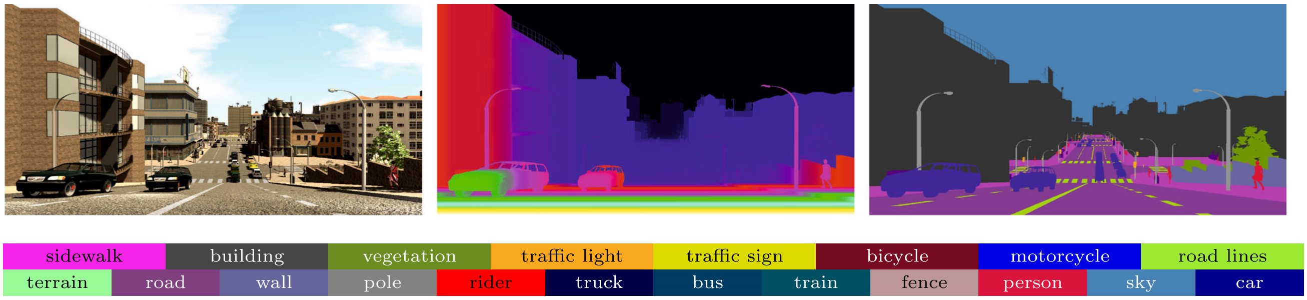


Fig. 10 The SYNTHIA-SF Dataset. A sample frame (left) with its depth (center) and semantic labels (right)

taining both dense semantic labels and depth ground-truth. It consists of a set of 60 images with 0.5 MP resolution that we use for evaluating Stixel semantic and depth accuracy. We follow the suggestion given by the author (Ladicky et al. 2014) to ignore the three rarest object classes, which leaves us with 8 classes.

Additionally, for training our semantic segmentation FCN, we use publicly available semantic annotations on other parts of *KITTY* (Kundu et al. 2014; He and Upcroft 2013; Sengupta et al. 2013; Xu et al. 2013; Zhang et al. 2015). Our total training set is composed of 676 images, where we harmonized the object classes used by the different authors to the previously mentioned set suggested by Ladicky et al. (2014). This harmonization and data processing is the same applied in the previous work (Schneider et al. 2016) to allow for fair comparison.

In order to further evaluate disparity accuracy we use the training data of the well-known stereo challenge *KITTI 2015* (Geiger et al. 2012). This dataset provides a set of 200 images with sparse disparity ground-truth obtained from a laser scanner. There is no suitable semantic ground-truth available for this dataset.

Furthermore, we also evaluate semantic accuracy using *Cityscapes* (Cordts et al. 2016), a highly complex dataset with dense annotations of 19 classes on $\sim$ 3000 images for training and 500 images for validation that we use for testing.

Unfortunately, all the above datasets were generated in flat road environments. Hence, they only help us validate that we are not decreasing our accuracy for this kind of environments. In order to compare the accuracy of competing algorithms on non-flat road scenarios, we need a new dataset.

Therefore, we introduce a new synthetic dataset inspired by Ros et al. (2016). This dataset has been generated with the purpose of evaluating our proposed model; however, it contains enough information to be useful in additional related tasks, such as object recognition, semantic and instance segmentation, among others.

SYNTHIA-San Francisco (*SYNTHIA-SF*) consists of photo-realistic frames rendered from a virtual city and comes with precise pixel-level depth and semantic annotations for 19 classes c.f. Fig. 10. This new dataset contains 2224 images that we use to evaluate both depth and semantic accuracy in non-flat roads.

5.2 Experiment Details

5.2.1 Metrics

We evaluate our proposed method in terms of semantic and depth accuracy using two metrics. The depth accuracy is obtained as the rate of outliers of the disparity estimates, the standard metric used to evaluate on KITTI benchmark (Geiger et al. 2012). An outlier is a disparity estimation with an absolute error larger than 3 px or a relative deviation larger than 5% compared to the ground-truth. The semantic accuracy is evaluated with the average Intersection-over-Union (IoU) over all classes, which is also a standard measure for semantic segmentation (Everingham et al. 2015). We measure the number of Stixels generated per image to quantify the complexity of the obtained representation. Finally, we evaluate the inference speed of the algorithm using the Frame-rate (Hz) metric, which helps us estimate if our system is capable of real-time performance. All the execution times of *Stixels* and *SGM* are obtained using a multi-threaded implementation running on standard consumer hardware: Intel i7-6800K. The semantic segmentation FCN frame-rate estimations are obtained using Maxwell NVidia Titan X. The Stixel frame-rate includes the over-segmentation approach. Note that Stixel frame-rate is variable if we use an over-segmentation method, i.e. it will depend on the number of Stixel cuts removed, therefore we provide a representative frame-rate. Similarly to Cordts et al. (2017), we can maximize the throughput of the system by computing SGM and Semantic Segmentation in parallel, then the system would run with one frame delay.

5.2.2 Baseline

Semantic Stixels (Schneider et al. 2016) serve as our comparison baseline, as they achieve state-of-the-art results in terms of Stixel accuracy. We provide the accuracy of our new disparity model, c.f. Sect. 3. Finally, we evaluate the complexity

Table 1 Accuracy of our methods compared to Semantic Stixels (Schneider et al. 2016), raw SGM and FCN

Metric	Dataset	Input		No over-segmentation		Fast: time series		Fast: FCN	
		SGM	FCN	Sem. Stixels	Ours	Sem. Stixels	Ours	Sem. Stixels	Ours
Disp Error (%)	Ladicky	16.66	–	17.38	16.84	17.60	17.01	17.44	16.84
	KITTI 15	11.01	–	11.05	11.21	11.9	11.9	11.21	11.24
	SYNTHIA-SF	11.06	–	29.33	**12.99**	30.60	14.20	31.12	14.19
IoU (%)	Ladicky	–	69.8	66.2	66.1	66.0	66.0	66.2	66.1
	Cityscapes	–	66.7	65.4	65.8	64.9	65.0	65.5	65.6
	SYNTHIA-SF	–	48.1	46.0	**48.5**	45.7	48.0	47.0	**48.6**

We evaluate on four datasets: Ladicky (Ladicky et al. 2014), KITTI 15 (Geiger et al. 2012), Cityscapes (Cordts et al. 2016) and SYNTHIA-SF using these metrics: Disparity Error (less is better) and Intersection over Union (more is better) c.f. Sects. 5.1 and 5.2.1. *Fast* versions are detailed in Sects. 4.1 and 4.2. Significantly best results are highlighted in bold

Table 2 Number of Stixels (10^3) generated by our methods compared to Semantic Stixels (Schneider et al. 2016) and raw input (total number of pixels)

Dataset	Input	No over-segmentation		Fast: time series		Fast: FCN	
	SGM/FCN	Sem. Stixels	Ours	Sem. Stixels	Ours	Sem. Stixels	Ours
Ladicky	454	0.6	0.6	0.6	0.6	0.6	0.6
KITTI 15	452	0.7	0.7	0.7	0.7	0.7	0.7
Cityscapes	2 k	1.4	1.5	1.3	1.4	1.4	1.5
SYNTHIA-SF	2 k	1.5	1.7	1.2	1.3	1.3	1.3

We evaluate on four datasets: Ladicky (Ladicky et al. 2014), KITTI 15 (Geiger et al. 2012), Cityscapes (Cordts et al. 2016) and SYNTHIA-SF c.f. Sect. 5.1. *Fast* versions are detailed in Sects. 4.1 and 4.2

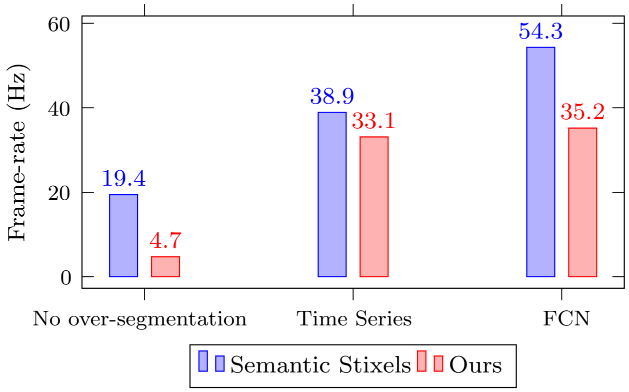


Fig. 11 Frame-rate of our method (only the Stixel computation step and the corresponding over-segmentation approach) compared to Semantic Stixels (Schneider et al. 2016) for SYNTHIA-SF (image resolution of 1920×1080) on a multi-threaded CPU implementation (Intel i7-6800K) computed with a Stixel width of 8 pixels and equivalent down-sampling in the v-direction. Different methods of over-segmentation are also compared, these are: Time Series c.f. Sect. 4.1, FCN c.f. Sect. 4.2

Table 3 Per-stage report of frame-rate of our pipeline for a stereo pair of resolution 1242 × 375. *OS* stands for *Over-segmentation*

Stage	Frame-rate (Hz)
SGM	55
Semantic segmentation	47.6
Our Stixels (OS: time series)	116
Our Stixels (OS: FCN)	130
Our Stixels (No OS)	61
Total (OS: time series)	20.92
Total (OS: FCN)	21.33
Total (No OS)	18

SGM run-time using a CPU Intel i7-6800K. For the *Semantic Segmentation* method, a Maxwell NVidia Titan X is used. Note that Stixel frame-rate is variable if we use an over-segmentation method, therefore we provide a representative run-time. The total frame-rate is reported as the sum of the stages

of the fast approach defined in Sect. 3.4, with the two over-segmentation techniques presented in Sects. 4.1 and 4.2.

5.2.3 Input

As input, we use disparity images obtained via SGM (Hirschmüller 2008) and pixel-level semantic labels computed by an FCN (Long et al. 2015). We use the same FCN model used in Schneider et al. (2016) without retraining, to allow for comparison. For the same reason, we set Stixel width to 8 px. The same down-sampling is applied in the vertical direction. The rest of the parameters used are taken from Schneider et al. (2016).

We use the camera parameters obtained after calibration to set the expected values of μ^a_{ground} and μ^b_{ground}. For *object* Stixels, we set $\sigma^b_{object} \rightarrow 0$, $\mu^b_{object} = 0$ because the disparity is too noisy for the slanted object model. Finally, since *sky* Stixels can not have slanted surfaces, we set: $\mu^a_{sky} = 0$, $\mu^a_{sky} = 0$, $\sigma^a_{sky} \to 0$, $\sigma^b_{sky} \to 0$.

In order to improve the computational efficiency of our approach, we use the two *Fast* Stixel over-segmentation methods presented in Sect. 4.1, labeled as *Time Series*, and Sect. 4.2, labeled as *FCN*.

5.3 Results

The quantitative results of our proposals and baselines, as described in Sect. 3, are shown in Tables 1 and 2 and Fig. 11.

The first observation is that our method achieves comparable or slightly better results on all datasets with flat roads c.f. compare *Semantic Stixels* to *Ours* for *Ladicky*, *KITTI 15* and *Cityscapes* datasets in Table 1. These results indicate that the novel and more flexible model does not harm the accuracy in such scenarios.

We also observe that our novel model is able to accurately represent non-flat scenarios in contrast to the original Stixel approach, yielding a substantially increased depth accuracy of more than 16% c.f. when comparing *Semantic Stixels* to *Ours* for the *SYNTHIA-SF* dataset in Table 1. Additionally, to verify that our method equally works also on real data, we provide a video of the Stixel 3D representation of a challenging non-flat road scene as supplementary material. Results also improve in terms of semantic accuracy, which we explain as a consequence of the joint semantic and depth inference that benefits from a better depth model.

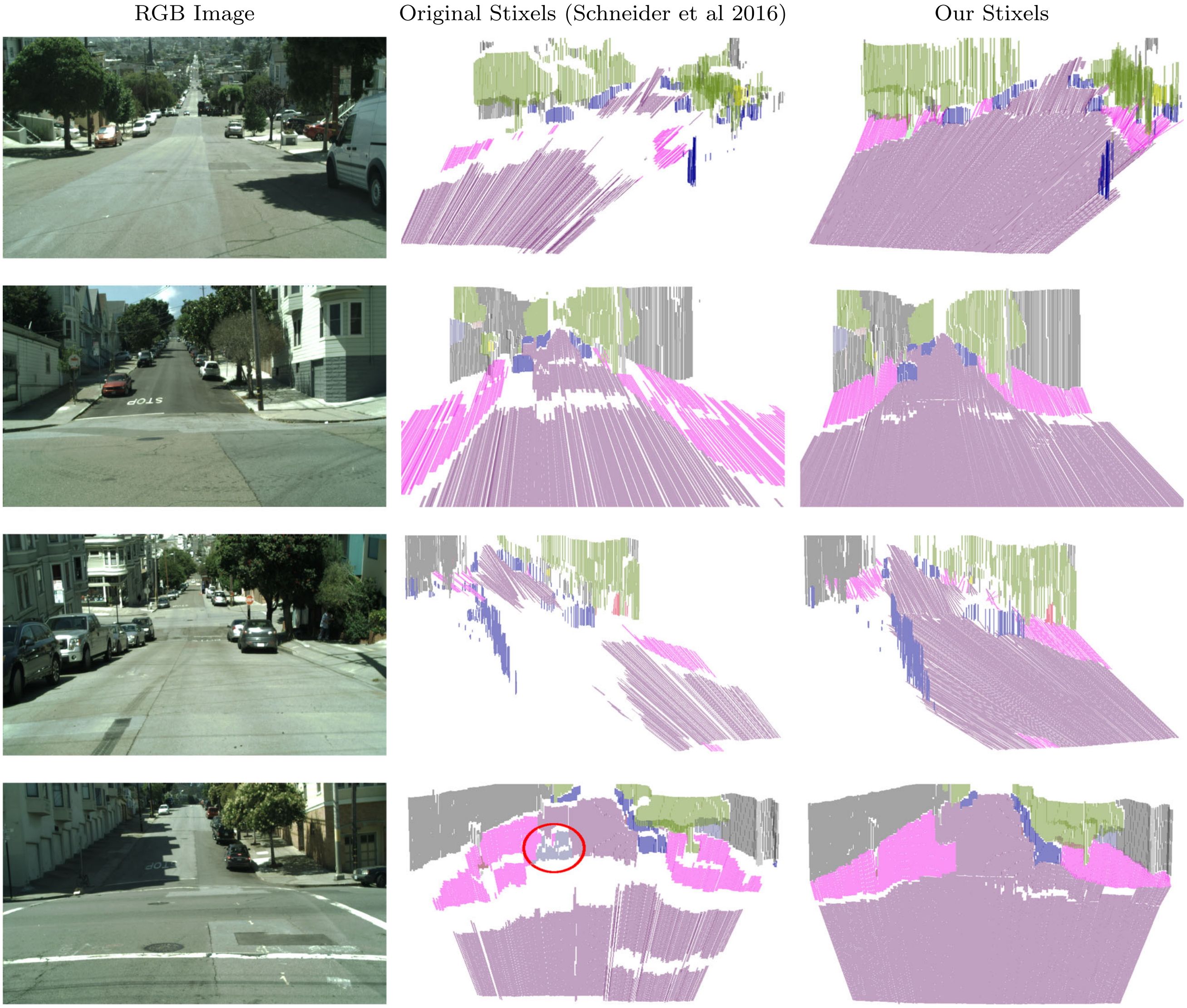


Fig. 12 Exemplary outputs on real data: in all cases with non-flat roads our model correctly represents the scene, while retaining accuracy on objects. The last example shows a failure case, where our approach classifies the road as sidewalk due to erroneous semantic input. However, the original approach reconstructs a wall in this case, highlighted by a red circle. This could lead to an emergency break (Color figure online)

A perfect over-segmentation method would find all optimal cuts, and consequently, it would have the same accuracy as not using any over-segmentation.

Our novel approach *Fast: FCN* has an accuracy almost equal to not using any over-segmentation method (in all cases but one). Note that, our proposed approach *Fast: FCN* is superior to *Fast: Time Series* method in all cases c.f. when comparing both methods for the *SYNTHIA-SF* dataset in Table 1.

Both over-segmentation methods increase the error for our challenging *SYNTHIA-SF* dataset; we think this is because of the difficult road Stixel cuts in these scenes, c.f. compare *No over-segmentation* to *Fast* methods in Table 1.

All variants are compact representations of the surrounding, since the complexity of the Stixel representation is small compared to the high resolution input images, c.f. Table 2.

Our last observation is that the proposed *Fast* variants improve the run-time of the original Stixel approach by up to 2×, and also improve the novel Slanted Stixel approach by up to 7×, with only a slight drop in depth accuracy c.f. Fig. 11. The benefit increases with higher resolution input images due to the quadratic and cubic computational complexity of the original and slanted Stixel inference methods, respectively. We also detail per-stage run-time c.f. Table 3 for completeness.

In addition to the quantitative evaluation presented before, we have visually inspected many of the obtained Stixel representations, to check the qualitative differences between our proposal and the previous work. Figure 12 illustrates some of these examples, in which the scenes with non-flat roads are correctly represented and all the objects in the scenario are identified by our proposal, while the previous model produces an incomplete road representation, or even generates non-existing objects at some road positions.

6 Conclusions

This paper presented a novel depth model for the Stixel world that is able to account for non-flat roads and slanted objects in a compact representation that overcomes the previous restrictive constant height and depth assumptions. This change in the way Stixels are represented is required for difficult environments that are found in many real-world scenarios. Moreover, in order to significantly reduce the computational complexity of the extended model, a novel approximation has been introduced that consists of checking only reasonable Stixel cuts inferred using fast methods. We showed in extensive experiments on several related datasets that our depth model is able to better represent slanted road scenes, and that our approximation is able to reduce the run-time drastically, with only a slight drop in accuracy.

As future work, we would like to focus on circumventing the limitations of our method. Namely, (1) the vertical/column independence assumed by the model is clearly not true. A more global representation, e.g. super-pixels that span vertically and horizontally, would be more compact and less prone to errors; (2) some surfaces are not well represented by a linear model, e.g. cars. A more complex depth model and specific models for each semantic class could represent more faithfully the scene. Nonetheless, a model with more free variables could also lead to a bad representation because of the noise; (3) the proposed over-segmentation algorithm has a non-predictable run-time. And this is a bad characteristic for a real-time system. The worst-case scenario, i.e. no Stixel cuts removed, is as slow as not using over-segmentation at all (although very unlikely); (4) in case of movement of the stereo rig during operation, there could be an offset in roll effectively breaking the vertical world assumption.

Acknowledgements This work has been partially supported by Spanish TIN2017-84553-C2-1-R (MINECO/AEI/FEDER, UE). We also thank the Generalitat de Catalunya CERCA Program, the 2017-SGR-1597 and 2017-SGR-313 projects, as well as the ACCIO agency. We also acknowledge SEBAP for the internship funding program. Antonio M. López acknowledges the financial support by the Spanish TIN2017-88709-R (MINECO/AEI/FEDER, UE), and by ICREA under the ICREA Academia Program. Finally, we thank Francisco Molero and Marc García at CVC/UAB for the generation of the SYNTHIA-SF dataset.

References

Benenson, R., Mathias, M., Timofte, R., & Gool, L. J. V. (2012). Fast stixel computation for fast pedestrian detection. In *Proceedings, Part III Computer vision—ECCV 2012. Workshops and demonstrations*—Florence, Italy, October 7–13, 2012 (pp 11–20). https://doi.org/10.1007/978-3-642-33885-4_2.

Benenson, R., Timofte, R., & Gool, L. J. V. (2011). Stixels estimation without depth map computation. In *IEEE international conference on computer vision workshops, ICCV 2011 Workshops*, Barcelona, Spain, November 6–13, 2011 (pp. 2010–2017). https://doi.org/10.1109/ICCVW.2011.6130495.

Bucila, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the twelfth ACM SIGKDD international conference on knowledge discovery and data mining*, Philadelphia, PA, USA, August 20–23, 2006 (pp. 535–541). https://doi.org/10.1145/1150402.1150464.

Carrillo, D. A. P., & Sutherland, A. (2016). Fast obstacle detection using sparse edge-based disparity maps. In *Fourth international conference on 3D Vision (3DV), 2016, IEEE* (pp. 66–72).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE con-*

ference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016 (pp. 3213–3223). https://doi.org/10.1109/CVPR.2016.350.

Cordts, M., Rehfeld, T., Schneider, L., Pfeiffer, D., Enzweiler, M., Roth, S., et al. (2017). The stixel world: A medium-level representation of traffic scenes. *Image and Vision Computing*,. https://doi.org/10.1016/j.imavis.2017.01.009.

Cordts, M., Schneider, L., Enzweiler, M., Franke, U., & Roth, S. (2014). Object-level priors for stixel generation. In *Proceedings pattern recognition - 36th German conference, GCPR 2014*, Münster, Germany, September 2–5, 2014 (pp. 172–183). https://doi.org/10.1007/978-3-319-11752-2_14.

Dhiman, V., Kundu, A., Dellaert, F., & Corso, J. J. (2014). Modern MAP inference methods for accurate and fast occupancy grid mapping on higher order factor graphs. In *ICRA*.

Everingham, M., Eslami, S. M. A., Gool, L. J. V., Williams, C. K. I., Winn, J. M., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136. https://doi.org/10.1007/s11263-014-0733-5.

Fink, E., & Gandhi, H. S. (2011). Compression of time series by extracting major extrema. *Journal of Experimental & Theoretical Artificial Intelligence*, *23*(2), 255–270.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on computer vision and pattern recognition*.

He, H., & Upcroft, B. (2013). Nonparametric semantic segmentation for 3D street scenes. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, November 3–7, 2013 (pp. 3697–3703). https://doi.org/10.1109/IROS.2013.6696884.

Hernandez-Juarez, D., Espinosa, A., Moure, J. C., Vázquez, D., & López, A. M. (2017). GPU-accelerated real-time stixel computation. In *2017 IEEE winter conference on applications of computer vision, WACV 2017*, Santa Rosa, CA, USA, March 24–31, 2017 (pp. 1054–1062). https://doi.org/10.1109/WACV.2017.122.

Hernandez-Juarez, D., Schneider, L., Espinosa, A., Vázquez, D., López, A. M., Franke, U., Pollefeys, M., & Moure, J. C. (2017). Slanted stixels: Representing san francisco's steepest streets. In *British Machine Vision Conference, BMVC 2017*, London, UK, September 4–7, 2017.

Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 328–341. https://doi.org/10.1109/TPAMI.2007.1166.

Ignat, O. (2016). Disparity image segmentation for free-space detection. In *2016 IEEE 12th international conference on intelligent computer communication and processing (ICCP)* (pp. 217–224). https://doi.org/10.1109/ICCP.2016.7737150.

Jasch, M., Weber, T., & Rätsch, M. (2018). Fast and robust RGB-D scene labeling for autonomous driving. *Journal of Computers*, *13*(4), 393–400.

Kundu, A., Li, Y., Dellaert, F., Li, F., & Rehg, J. M. (2014). Joint semantic segmentation and 3D reconstruction from monocular video. In *13th European conference computer vision—ECCV 2014*, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI (pp. 703–718). https://doi.org/10.1007/978-3-319-10599-4_45.

Ladicky, L., Shi, J., & Pollefeys, M. (2014). Pulling things out of perspective. In *2014 IEEE conference on computer vision and pattern recognition, CVPR 2014*, Columbus, OH, USA, June 23–28, 2014 (pp. 89–96). https://doi.org/10.1109/CVPR.2014.19.

Levi, D., Garnett, N., & Fetaya, E. (2015). Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *Proceedings of the British machine vision conference 2015, BMVC 2015*, Swansea, UK, September 7–10, 2015 (pp. 109.1–109.12). https://doi.org/10.5244/C.29.109.

Li, X., Flohr, F., Yang, Y., Xiong, H., Braun, M., Pan, S., Li, K., & Gavrila, D. M. (2016). A new benchmark for vision-based cyclist detection. In *2016 IEEE intelligent vehicles symposium, IV 2016*, Gotenburg, Sweden, June 19–22, 2016 (pp 1028–1033). https://doi.org/10.1109/IVS.2016.7535515.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. In *Proceedings of the international conference on learning representations (ICLR)*.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE conference on computer vision and pattern recognition, CVPR 2015*, Boston, MA, USA, June 7–12, 2015 (pp. 3431–3440). https://doi.org/10.1109/CVPR.2015.7298965.

Muffert, M., Schneider, N., & Franke, U. (2014). Stix-fusion: A probabilistic stixel integration technique. In *Canadian conference on computer and robot vision, CRV 2014*, Montreal, QC, Canada, May 6-9, 2014, pp 16–23. https://doi.org/10.1109/CRV.2014.11.

Nuss, D., Yuan, T., Krehl, G., Stuebler, M., Reuter, S., & Dietmayer, K. (2015). Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter. In *IEEE intelligent vehicles symposium (IV)*.

Pfeiffer, D. (2014). The stixel world—a compact medium-level representation for efficiently modeling three-dimensional environments. Ph.D. thesis, Hu Berlin.

Pfeiffer, D., & Franke, U. (2011). Towards a global optimal multi-layer stixel representation of dense 3D data. In *Proceedings of British machine vision conference, BMVC 2011*, Dundee, UK, August 29–September 2, 2011 (pp 1–12). https://doi.org/10.5244/C.25.51.

Pfeiffer, D., Gehrig, S., & Schneider, N. (2013). Exploiting the power of stereo confidences. In *2013 IEEE conference on computer vision and pattern recognition*, Portland, OR, USA, June 23–28, 2013 (pp 297–304). https://doi.org/10.1109/CVPR.2013.45.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. (2016). The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*.

Schneider, L., Cordts, M., Rehfeld, T., Pfeiffer, D., Enzweiler, M., Franke, U., Pollefeys, M., & Roth, S. (2016). Semantic stixels: Depth is not enough. In *2016 IEEE intelligent vehicles symposium, IV 2016*, Gotenburg, Sweden, June 19–22, 2016 (pp 110–117). https://doi.org/10.1109/IVS.2016.7535373.

Schneider, L., Jasch, M., Fröhlich, B., Weber, T., Franke, U., Pollefeys, M., & Rätsch, M. (2017). Multimodal neural networks: Rgb-d for semantic segmentation and object detection. In *Scandinavian conference on image analysis*, Berlin: Springer (pp 98–109).

Sengupta, S., Greveson, E., Shahrokni, A., & Torr, P. H. S. (2013). Urban 3D semantic modelling using stereo vision. In *2013 IEEE international conference on robotics and automation*, Karlsruhe, Germany, May 6–10, 2013 (pp 580–585). https://doi.org/10.1109/ICRA.2013.6630632.

Thrun, S. (2002). Robotic mapping: A survey. In G. Lakemeyer & B. Nebel (Eds.), *Exploring artificial intelligence in the new millenium*. Burlington: Morgan Kaufmann.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on Computer vision and pattern recognition, 2001. CVPR 2001*, IEEE, (vol. 1, pp. I–I).

Xie, S., & Tu, Z. (2017). Holistically-nested edge detection. *International Journal of Computer Vision*, *125*(1–3), 3–18. https://doi.org/10.1007/s11263-017-1004-z.

Xu, P., Davoine, F., Bordes, J., Zhao, H., & Denoeux, T. (2013). Information fusion on oversegmented images: An application for urban scene understanding. In *Proceedings of the 13. IAPR international conference on machine vision applications, MVA 2013*, Kyoto, Japan, May 20–23, 2013 (pp 189–193). URL http://www.mva-org.jp/Proceedings/2013USB/papers/08-04.pdf.

Zhang, R., Candra, S. A., Vetter, K., & Zakhor, A. (2015). Sensor fusion for semantic segmentation of urban scenes. In *IEEE international conference on robotics and automation, ICRA 2015*, Seattle, WA, USA, 26–30 May, 2015 (pp. 1850–1857). https://doi.org/10.1109/ICRA.2015.7139439.

Ziegler, J., Bender, P., Dang, T., & Stiller, C. (2014). Trajectory planning for bertha—A local, continuous method. In *2014 IEEE intelligent vehicles symposium proceedings*, Dearborn, MI, USA, June 8–11, 2014 (pp. 450–457).

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C. G., Kaus, E., Herrtwich, R. G., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., & Zeeb, E. (2014), Making bertha drive—an autonomous journey on a historic route. In *IEEE intelligent transportation systems magazine* (pp. 8–20). https://doi.org/10.1109/MITS.2014.2306552.