Semantic Stixels: Depth is Not Enough

Lukas Schneider*[,1,2], Marius Cordts*[,1,3], Timo Rehfeld[4,3], David Pfeiffer[1], Markus Enzweiler[1], Uwe Franke[1], Marc Pollefeys[2], and Stefan Roth[3]

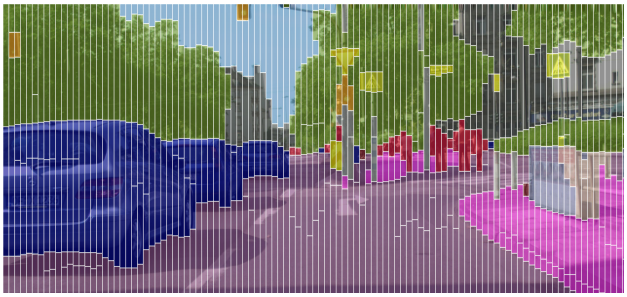
***Abstract*— In this paper we present *Semantic Stixels*, a novel vision-based scene model geared towards automated driving. Our model jointly infers the geometric and semantic layout of a scene and provides a compact yet rich abstraction of both cues using Stixels as primitive elements. Geometric information is incorporated into our model in terms of pixel-level disparity maps derived from stereo vision. For semantics, we leverage a modern deep learning-based scene labeling approach that provides an object class label for each pixel.**

Our experiments involve an in-depth analysis and a comprehensive assessment of the constituent parts of our approach using three public benchmark datasets. We evaluate the geometric and semantic accuracy of our model and analyze the underlying run-times and the complexity of the obtained representation. Our results indicate that the joint treatment of both cues on the Semantic Stixel level yields a highly compact environment representation while maintaining an accuracy comparable to the two individual pixel-level input data sources. Moreover, our framework compares favorably to related approaches in terms of computational costs and operates in real-time.

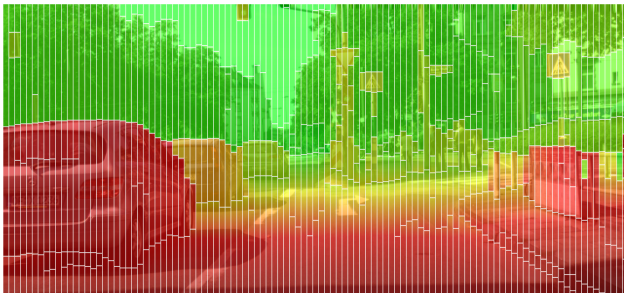
I. Introduction

Self-driving cars need to understand and effortlessly act within a traffic environment that has been specifically designed to be easily accessible for humans. On this account, many current prototypical implementations of automated driving heavily build upon recent advances made in the field of visual semantic scene understanding from camera sensors *e.g.* [1], [2], [3]. Such approaches are able to extract a rich model of traffic scenes that includes both a geometric and a semantic representation of traffic objects and infrastructural elements. This environment model then acts as the foundation for higher-level building blocks of automated vehicles, *i.e.* localization, planning, and vehicle actuation. Despite the demands for a fine-grained scene representation in terms of both 3D perception and semantic understanding, subsequent processing stages require the inferred interpretation to be compact to enable efficient data processing.

A particularly useful instance of an environment model is the Stixel representation, as introduced in [4], [5]. It has been widely adopted in the intelligent vehicles community over the last years *e.g.* [1], [6], [7], [8], [9]. The Stixel model defines a compact medium-level representation of dense 3D disparity data obtained from stereo vision using vertically oriented rectangles (Stixels) as primitive elements. As such, Stixels allow for an enormous reduction of the raw input data to a few hundred Stixels only. At the same time, most task-relevant scene structures such as free space and obstacles are adequately represented.



Semantic representation, where Stixel colors encode semantic classes following [10].



Depth representation, where Stixel colors encode disparities from close (red) to far (green).

Fig. 1: Scene representation obtained via Semantic Stixels. The scene is represented via its geometric layout (bottom) and semantic classes (top).

In this paper, we extend the original Stixel representation [5] by additionally incorporating semantic information in terms of object class information. We present Semantic Stixels, where the geometric and semantic layout of traffic scenes is jointly inferred from a dense disparity map and a pixel-level semantic scene labeling. Our framework yields a considerably improved Stixel representation that not only adds object class information as an attribute to each Stixel, but also outperforms the original Stixel model in terms of geometric accuracy. Subsequent system modules of a scene understanding pipeline are provided with a compact environment representation that accurately reflects the geometric and semantic structure of the scene. See Figures 1 and 2 for an overview.

As an input to our framework, we use semi-global matching [14] for stereo computation and deep fully convolutional networks [15] for scene-labeling with up to 19 object classes.

*Both authors contributed equally
[1]Daimler AG, R&D, Böblingen, Germany
[2]ETH Zurich, Zurich, Switzerland
[3]TU Darmstadt, Darmstadt, Germany
[4]MBRDNA, Sunnyvale, US
Primary contact: lukas.schneider@daimler.com

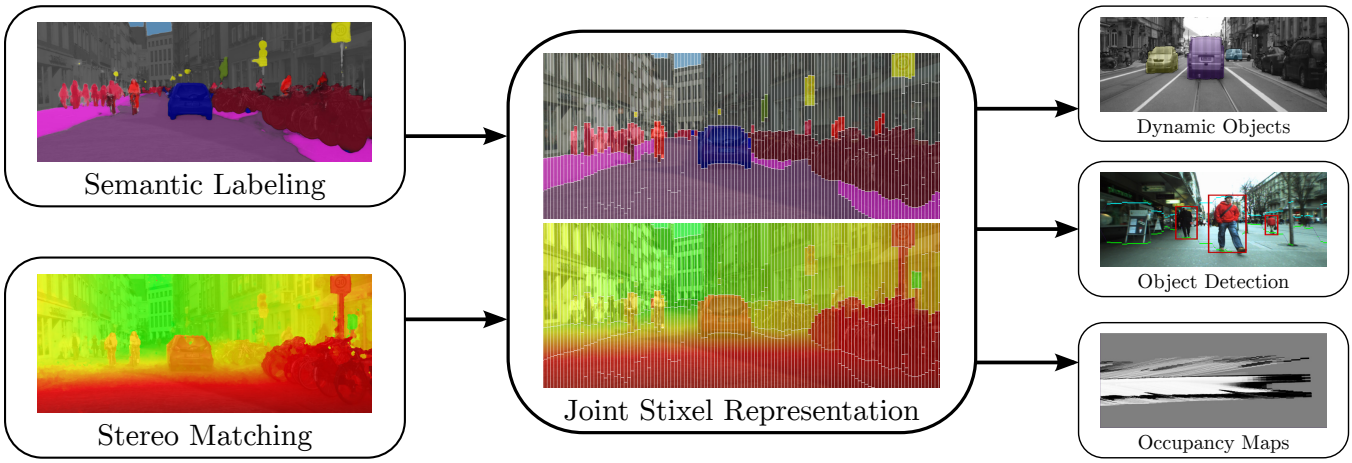


Fig. 2: System overview of our Semantic Stixels. Based on Semantic Labeling and Stereo Matching, we compute a compact scene model that in turn is the foundation for higher-level tasks such as object formation [11], object detection and classification [12], as well as mapping [13].

The proposed joint Stixel-level aggregation of low-level geometry and semantics not only boosts model compactness and robustness, it also maintains the overall scene representation quality of both low-level input components, despite the inevitable Stixel discretization artifacts.

We further demonstrate how the underlying joint optimization problem is efficiently solved in linear time complexity with regard to the number of object classes involved. This allows us to take a sufficiently large and detailed set of object classes into account, while at the same time keeping the overall runtime in an admissible range for a real-time application, i.e. 15 Hz on 2 MP images.

II. Related work

Our proposed Semantic Stixels leverage pixel-level semantic labels and dense depth maps to produce a compact scene representation, *c.f*. Fig. 2. Therefore, we see four categories of related publications.

The first category comprises semantic labeling, where all recent state-of-the-art methods rely on deep neural networks. For examples, refer to the PASCAL VOC benchmark [16]. One of the pioneering works for applying CNNs to semantic labeling are fully convolutional networks (FCNs) [15]. Several other methods build on top of FCNs and model statistical dependencies via conditional random fields (CRFs) [17], [18], [19], [20] or incorporate global scene context [21], [22], [23]. In this work, we opt for FCNs that we found to provide the best trade-off between classification and runtime performance.

The second category is formed by dense stereo matching algorithms that estimate a depth for almost all pixels in the image. Please refer to [2], [24], [25] for an excellent overview. Stereo methods can be characterized by leveraging either local or global optimization schemes, by the level of granularity, *i.e*. pixel or superpixel level, and by the type of incorporated scene prior, *e.g*. Manhattan world assumption. In this work, we use semi-global matching [14], a pixel-level globally optimizing dense stereo algorithm.

The third category includes approaches that leverage both, semantic and depth cues. Methods either perform a joint inference on pixel-level using monocular [26], [27] or stereoscopic [28], [29], [30] image data, or operate on 3D point clouds to leverage the mutual benefit of semantic and depth [31], [32], [33]. In contrast to such previous work, our model fuses pixel-level semantic and depth cues straight into a compact, robust and accurate scene representation with Stixels as the resulting level of abstraction.

The last category consists of Stixel-based methods, which we consider to be more closely related to our work. Stixels were originally used to represent the 3D scene as observed by stereoscopic [5], [6] or monocular imagery [34]. Later, an extension that adds semantic labels in a post-processing step was proposed [9]. Given that initially computed Stixels were grouped to obtain proposal regions for semantic classification, such a method can hardly recover from insufficiencies already present in the Stixel segmentation. These shortcomings can be either addressed via low-level appearance models in an on-line self-supervised framework [35] or via low-level fusion of depth and semantics in the Stixel generation process [7], [36]. Such semantic information is either obtained via object detectors for a suitable set of classes [7] or pixel classification with random decision forests [36]. In this work, we extend our previous method [36] by replacing the random forest with a state-of-the-art deep neural network to obtain strong semantic labels of many classes. Further, we propose a Stixel model that has a linear time complexity with respect to the number of semantic classes, compared to a quadratic complexity in [7] and [36]. Therefore, we can handle all relevant classes for our real-time application, as opposed to the other Stixel variants.

Our contributions are: (1) we present a geometrically and semantically consistent scene model geared towards the needs of autonomous driving applications, (2) an efficient

way to solve the underlying global optimization problem in linear time complexity regarding the number of semantic object classes involved and (3) an in-depth evaluation of our novel approach on several real-world benchmark datasets in terms of semantic and depth accuracy, as well as computational costs.

III. Method

Figure 2 provides an overview of our proposed system. We start by computing dense pixel-level semantic labels and depth estimates. These two channels are then used as unary data terms in our Stixel model yielding a Stixel segmentation that leverages the information in both modalities. The obtained Stixel world provides a consistent 3D semantic representation that typically serves as a compact interface to subsequent processing stages, such as object detectors or occupancy grids.

A. Semantic Labeling

Our proposed semantic Stixel model leverages pixel-level semantic label scores, *i.e.* we require the semantic labeling module to provide a probability estimate for each semantic class at each pixel in the image. In principle, many recent approaches deliver such probability maps, *c.f.* Sec. II. In this work, we opt for fully convolutional networks (FCNs) as presented in [15]. These deep neural networks are the main component in many other semantic labeling methods, *e.g.* [37], [19], [20], [17], and are relatively straightforward to train and use. In addition, FCNs yield excellent classification performance, while having a relatively low runtime on modern GPUs.

We use GoogLeNet [38] as the underlying network architecture. This network provides an excellent trade-off between classification performance, computational efficiency, and GPU memory demands. Analogous to [15], we replace the final fully connected classification layer with a convolutional layer and add "skip" layers to obtain an output stride of 8 pixels. Subsequent bilinear upscaling is performed to match the desired output resolution. In the last network layer, we apply softmax normalization to obtain probability scores for each class, as required by our Semantic Stixel model.

B. Stereo Matching

In addition to the semantic labeling, we leverage depth information. To that end, we require a stereo matching module that provides dense disparity estimates associated with a confidence score [39] for each pixel. While many such stereo methods exist, *c.f.* Sec. II, we rely on semi global matching (SGM) as originally proposed in [14]. SGM yields competitive results [25] and there exist real-time capable FPGA implementations that are successfully used in automotive environments [40].

C. Semantic Stixels

A Stixel as proposed in [5] is a narrow stick of width w, sufficiently described by only very few parameters: the column u and its bottom respectively top coordinates v_B and v_T. In the context of 3D environment perception, each Stixel reflects the geometric layout of the corresponding image segment. The underlying world model, which is specifically designed for outdoor man-made environments, thereby distinguishes three geometric classes: ground (lying), object (upright), and sky (infinitely far away). Therefore, such a geometric class g and accordingly either the distance to the camera (upright) or the vertical displacement to the reference ground plane (lying) d are two additional parameters attached to each Stixel. Thus, objects can be compactly described by a few Stixels only, *c.f.* Fig. 1. In this work, we add an additional semantic label l to each Stixel that is based on the semantic labeling module described in Sec. III-A.

In [5], the Stixel segmentation is formulated as the solution of an energy minimization problem. Since solving such an optimization problem in a discrete 2D label space is known to be a hard optimization problem, the energy function is formulated independently for each image column, rendering 1D minimization problems that can be solved optimally and efficiently via dynamic programming. Due to a high coupling between neighboring columns in the input data, the resulting Stixel segmentation is nevertheless based on wider context.

The energy function consists of unary terms $E_u(s_i)$ and pairwise terms $E_p(s_i, s_{i-1})$ defined over individual Stixel hypotheses $s_i = (u, v_B, v_T, g, d, l)$. The unary is composed of two data terms, *i.e.*

$$E_u(s_i) = E_d(s_i) + w_l E_l(s_i) \ . \tag{1}$$

The first is a generative model of the disparity estimates within a Stixel s_i and depends on its geometric configuration g and d. For details, please refer to [5]. The second data term $E_l(s_i)$ rates the consistency of the hypothesis s_i with semantic label l and the probability scores $\boldsymbol{\sigma}$ obtained by the semantic labeling module, *c.f.* Sec. III-A. Denoting $\mathcal{P}$ as the set of all pixels within the Stixel s_i, we define

$$E_l(s_i) = -\sum_{p \in \mathcal{P}} \log \sigma(p, l) \ . \tag{2}$$

Note that we neglect implausible hypotheses s_i, *e.g.* an upright Stixel with semantic label *road*. The parameter w_l in Eq. (1) controls the influence of the semantic data term with respect to the disparity model.

The pairwise terms $E_p(s_i, s_{i-1})$ capture prior knowledge on the Stixel segmentation, for example certain geometric configurations such as objects below of the ground surface are rated as rather unlikely. Again, the interested reader is referred to [5] for details. In principle, it is possible to model explicit transition likelihoods between any pairs of semantic classes, *c.f.* [10]. Unfortunately, the computational complexity of the resulting dynamic program grows quadratically with the number of classes, since all combinations of adjacent Stixels must be evaluated. However, we argue that such transition probabilities mainly depend on their geometric property and less on their semantic interpretation. For example, transition likelihoods between pedestrian and vehicle Stixels are similar in both directions. On the other

hand, object Stixels on top of ground ones are more likely than vice versa, independent of their actual semantic label.

Thus, we neglect the influence of the semantic labels l on the transition probabilities, *i.e.*

$$E_p(s_i, s_{i-1}) = E_p(\tilde{s}_i, \tilde{s}_{i-1}) \ , \tag{3}$$

where we use $s_i = \tilde{s}_i \cup l$, with $\tilde{s}_i = (u, v_B, v_T, g, d)$, for the ease of notation. Overall, the Stixel segmentation is defined as the solution of the energy minimization

$$\begin{aligned} \min_{\boldsymbol{s}} E(\boldsymbol{s}) &= \min_{\boldsymbol{s}} \left[\sum_i E_u(s_i) + \sum_i E_p(s_i, s_{i-1}) \right] \\ &= \min_{\tilde{\boldsymbol{s}}} \min_{\boldsymbol{l}} \left[\sum_i E_u(\tilde{s}_i, l_i) + \sum_i E_p(\tilde{s}_i, \tilde{s}_{i-1}) \right] \\ &= \min_{\tilde{\boldsymbol{s}}} \left[\sum_i \min_{l_i} E_u(\tilde{s}_i, l_i) + \sum_i E_p(\tilde{s}_i, \tilde{s}_{i-1}) \right] . \end{aligned} \tag{4}$$

Since the pairwise term is independent of the semantic labels and the unary term only depends on a single Stixel s_i, the minimization over the semantic labels can be pushed into the sums. Thus, the best semantic label for each Stixel hypothesis can be computed directly, *i.e.*

$$\min_{l_i} E_u(\tilde{s}_i, l_i) = E_d(s_i) + w_l \min_{l_i} E_l(\tilde{s}_i) \ . \tag{5}$$

Note that the latter term can be efficiently evaluated using integral tables of the probability scores along image columns.

IV. Experiments

In this section, we provide an in-depth analysis of our Stixel model. We introduce the metrics and baselines as well as the details of the training procedure and the parametrization. Finally, quantitative and qualitative results are reported on three different public datasets.

A. Datasets

From the small number of available realistic outdoor datasets in the area of autonomous driving, the subset of Kitti [2] annotated by Ladicky *et al*. [26] is, to the best of our knowledge, the only dataset containing dense semantic labels and depth ground truth. Therefore, this is the only dataset that allows to report performance metrics of both aspects of our Semantic Stixel representation on the same dataset. It consists of 60 images with a resolution of 0.5 MP that we all use for evaluation and none for training. We follow the suggestion of the original author to ignore the three rarest object classes, leaving a set of 8 classes. We use additional publicly available semantic annotations on other parts of Kitti [33], [41], [42], [43], [44], [45] for training. All in all, we have a training set of 676 images, where we harmonized the object classes used by the different authors to the previously mentioned set.

As a second dataset, we report disparity performances on the training data of the stereo challenge in Kitti'15 [25]. This dataset comprises a set of 200 images with sparse disparity ground truth obtained from a Velodyne HDL-64 laser scanner. However, there is no suitable semantic ground truth available for this dataset.

Third, we evaluate on Cityscapes [10], a highly complex and challenging dataset with dense annotations of 19 classes on 3500 images for training and 500 images for validation that we used for testing. While there are stereo views available, ground truth disparities do not exist.

B. Metrics

In our experiments, we use four different metrics that are designed to assess the viability of our Semantic Stixel model and several baselines in view of automated driving tasks.

The first metric evaluates the depth performance and is defined as the outlier rate of the disparity estimates [2]. A disparity estimation with an absolute deviation larger than 3 px or a relative deviation larger than 5 % compared to ground truth is considered as an outlier. The second metric assesses the semantic performance and is defined as the average Intersection-over-Union (IoU) over all classes [16]. Third and fourth, we report the framerates and use the number of Stixels per image as a proxy to assess the complexity of the obtained representation. Note that a system suitable for autonomous driving is expected to reach excellent performance in all four metrics simultaneously.

C. Baselines

We compare our results to three baselines. Following Sec. II, we consider [36] to be most related and included it as a baseline. Second, we report results using Stixels computed on the depth information only, *i.e.* as proposed in [5], and add semantic labels as the argmax label from our FCN for each Stixel ("Depth 1st"). Third, we leverage our proposed model, but disable the depth channel and compute the segmentation on the FCN output only. Subsequently, we freeze the segmentation and compute the disparity estimates via post-processing by computing the best disparity referring to the Stixel data term ("Semantic 1st").

D. FCN Training

A fully convolutional neural net (FCN) is the foundation of our semantic information, as described in Sec. III-A. To train this deep network, we follow the training procedure as outlined in [15]. We start with the coarsest output stride, train until convergence, add a skip layer to reach on output stride of 8 pixels, and continue training. We use identical learning parameters as in [15] and adapt the learning rate to our image resolution.

For training on Cityscapes, we initialize our network with an ImageNet [46] pre-trained model [47]. For Kitti, we trained a second model on Cityscapes with half resolution and use this model as initialization on Kitti. In doing so, we improve the performance of this raw FCN baseline by 6 % IoU over a model without Cityscapes initialization.

E. Stixel Parameterization

In our experiments, we report results for two variants of Stixels. The first produces Stixels with a width of 2 pixels

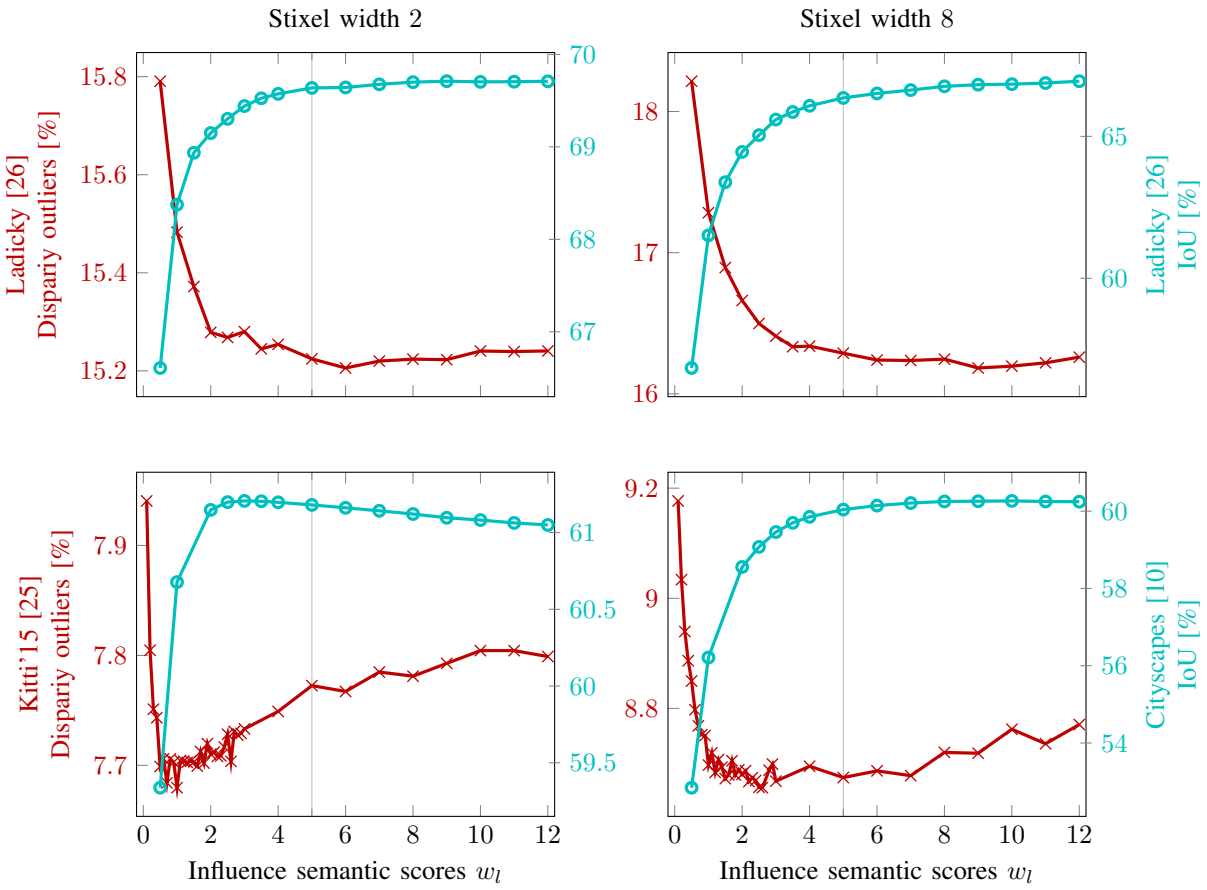


Fig. 3: Analysis of the influence of the semantic scores w_l. We evaluate Stixels with width 2 (left column) and width 8 (right column) regarding four metrics: (1) Disparity outliers on Ladicky [26] (top row, red) (2) IoU on Ladicky [26] (top row, blue) (3) Disparity outliers on Kitti'15 [25] (bottom row, red) (4) IoU on Cityscapes [10] (bottom row, blue).

and is designed to achieve maximum performance in terms of depth and semantic accuracy. The second parameterization, Stixels with a width of 8 pixels, is designed to be an excellent trade-off between accuracy, runtime, and representation complexity and hence more relevant for practical applications. To control the width, the input channels are downscaled by the desired value taking stereo confidences into account. We also apply the same downscaling in the v-direction to support the targeted trade-off between accuracy and efficiency. Note that in the case of eightfold downscaling, the FCN already produces matching score maps and we can skip the final upscaling layer. In doing so, the transfer time of the FCN results from GPU to CPU memory is significantly reduced.

The most important parameter in our experiments w_l controls the influence of our data terms based on the semantic versus the disparity channel. Therefore, we conducted experiments on all our datasets and measured the performances in terms of disparity and semantic accuracy for varying values of w_l, *c.f.* Fig. 3. With increasing w_l, the semantic data term gains influence and one would expect that the semantic performance increases, while the disparity performance decreases. However, as the results in Fig. 3 show, both metrics improve until a moderate value of w_l is reached and only then the disparity performance suffers, an effect that is consistent across all datasets and Stixel widths. This observation indicates that the semantic and disparity channel contain complementary information and support the performance in the other domain. Interestingly, on the challenging Cityscapes dataset, the semantic score decreases with high values w_l unveiling that also some regularization via the Stixel model helps in that domain. Referring to Fig. 3, we choose $w_l = 5$ for the remaining experiments, as this value represents the best compromise across all metrics for all datasets.

F. Results

The results of our model and the baselines described in Sec. IV-C are reported in Table I. We report four different metrics on three different datasets, *c.f.* Sec. IV-A and IV-B. Qualitative results can be found in Figs. 4 to 6.

Our method is clearly outperforming all other variants in terms of joint disparity and semantic accuracy for both Stixel resolutions and is competitive in terms of runtime and complexity. The results show that both, the depth first variant with semantic post-processing and the semantic first variant with depth post-processing, suffer from a substantial gap in the performance of the post-processed channel compared to

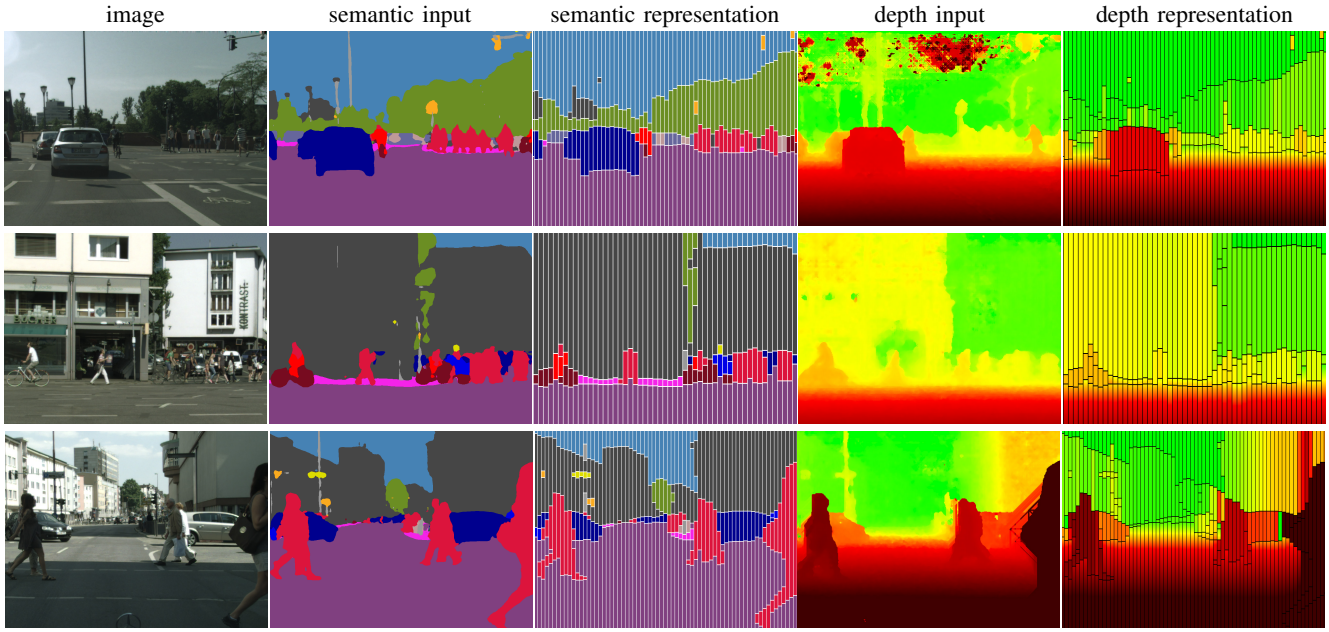


Fig. 4: Example output of our Semantic Stixels with color encodings as in Fig. 1. Even objects as small as traffic lights and signs are represented accurately.

TABLE I: Quantitative results of our Stixel model compared to three baselines and to raw SGM and FCN. We evaluate on three datasets using four metrics, *c.f.* Sec. IV-A and IV-B, and report results for a highly accurate setup (width 2) and a highly efficient one (width 8), *c.f.* Sec. IV-E.

		Raw		Stixel width 2				Stixel width 8			
Metric	Dataset	SGM	FCN	[36]	Depth 1st	Sem. 1st	Ours	[36]	Depth 1st	Sem. 1st	Ours
Disparity Error [%]	Ladicky [26]	20.4	—	18.2	18.7	25.1	**15.2**	18.9	19.4	27.3	**16.3**
	Kitti'15 [25]	8.9	—	9.8	8.6	16.5	**7.8**	10.5	9.6	18.3	**8.8**
IoU [%]	Ladicky [26]	—	69.8	34.0	47.1	69.1	**69.6**	33.5	45.9	62.5	**66.4**
	Cityscapes [10]	—	60.8	—[c]	44.3	60.7	**61.2**	—[c]	43.9	55.2	**60.0**
Frame-rate [Hz][a]	Kitti [2]	55.0	47.6	9.2	15.1	14.1	9.8	143	167	222	154
	Cityscapes [10]	22.0	15.4	0.12	1.9	1.4	1.1	6.5	40	47.2	30.3
No. of Stixels [10^3]	Kitti [2]	226[b]	226[b]	3.2	**2.0**	2.5	2.7	0.8	**0.5**	0.6	0.6
	Cityscapes [10]	1 k[b]	1 k[b]	—[c]	**4.5**	5.1	6.2	—[c]	1.1	**1.0**	1.4

[a] We report the frame-rates of the individual modules, *i.e.* SGM on FPGA, FCN on GPU (Nvidia Titan X), and Stixels on CPU (Intel Xeon, 10 cores, 3 GHz). The overall frame-rate is determined by the slowest component, since all modules use distinct processing hardware and can be perfectly pipelined. Note that [36] uses an RDF based pixel classifier instead of the FCN. This module is also implemented on GPU and its runtime is neglectable compared to the Stixel computation.

[b] We list half the number of pixels for SGM and FCN raw data to approximately compare the complexity to Stixels.

[c] We did not retrain an RDF pixel classifier [36] on Cityscapes and therefore do not report performances except for the runtime.

our method. This observation is consistent across all datasets and both Stixel widths and clearly shows the need for an early fusion of such information. On top of that, our model outperforms the post-processing methods in the domain of their stronger channel. Despite the inherent discretization, our Semantic Stixels even exceed the performance of the raw SGM input data and our width 2 variant also outperforms the FCN on Cityscapes. These observations clearly indicate the effectiveness of our model in terms of fusing different input channels. Qualitatively, the mutual benefit of the two channels can be seen in Fig. 4, top row. Note the noisy depth input in the sky region that is well suppressed in our joint representation due to the coupling with the semantic information. Another example can be found in Fig. 5, where errors in the semantic information channel are corrected via the joint reasoning in our Stixel model.

Compared to [36], we double the semantic performance, a fact that we account to our FCN classifier being superior to the RDFs in [36]. Surprisingly, [36] negatively affects the disparity performance on Kitti'15 [25], which we attribute to its weaknesses in image areas where many different objects are present. Thus, the disparity performance is harmed particularly in those parts of the image where Kitti'15 has disparity ground truth. Please refer to Fig. 6 for qualitative

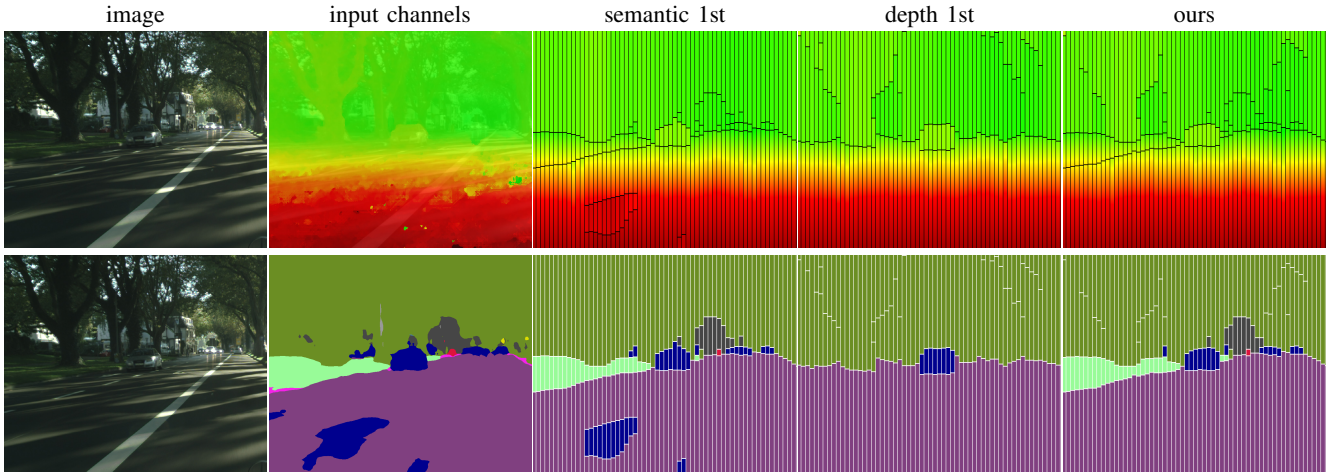


Fig. 5: Our method compared to the depth and semantic 1st baselines. Due to our joint optimization, the Stixel representation is able to recover from errors in the inputs like the erroneous car region on the road. Note that the semantic 1st baseline cannot recover from such errors, while the depth 1st baseline fails to represent many other elements in the scene.

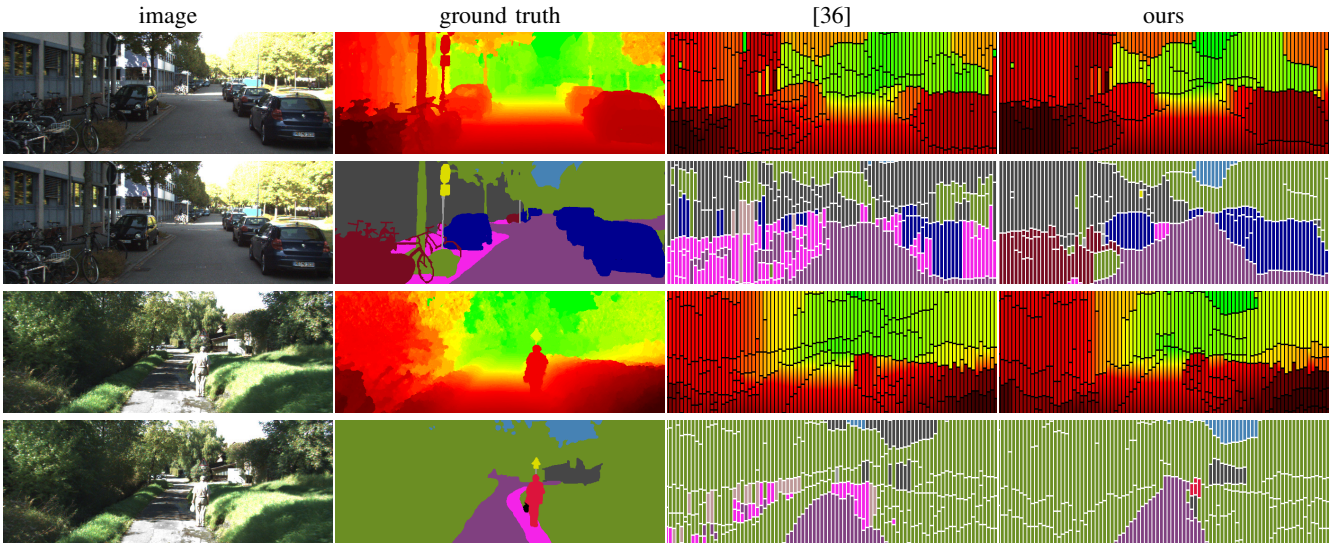


Fig. 6: Results of our method compared to [36] on Ladicky [26] including a failure case in the bottom row. The vegetation in the lower image half violates the planar word model, leading in turn to a high number of Stixels and a depth error above 30 %. Nevertheless, the drivable space is recognized correctly.

examples.

The superior disparity and semantic performance of our model comes with only a slight increase in complexity that we measure by the number of Stixels in the resulting representation. The fusion of two inputs requires some additional computations resulting in a slightly lower frame-rate of our Stixel generation compared to the post-processing implementations, especially for Cityscapes [10], where 19 classes need to be processed. However, in case of the practically more relevant variant with width 8, the overall system frame-rate is dominated by the FCN computation resulting in 15 Hz on 2 MP images in Cityscapes and 48 Hz on 0.5 MP in Kitti. Note that our Stixel module is significantly faster on Cityscapes than [36], due to our linear instead of quadratic dependency on the number of classes.

V. Conclusion

We presented the Semantic Stixels framework that leverages state-of-the-art pixel-level geometry and semantics, and integrates both in terms of a jointly optimized scene model with high accuracy. Our approach is one solution to the problem of translating the impressive scene labeling results obtained on pixel-level through deep learning into a semantically and geometrically consistent representation. This representation is closely aligned to the needs of the self-driving car application, where the scene model has to bridge the gap between the richness of detail available on pixel-level and the robustness and efficiency on object-level. We feel that Semantic Stixels provide a suitable abstraction level at the "sweet spot" between semantic and depth accuracy, compactness, and efficiency.

REFERENCES

[1] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. G. Herrtwich, "Making Bertha See," in *CVAD Workshop (ICCV)*, 2013.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.

[3] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D traffic scene understanding from movable platforms," *Trans. PAMI*, vol. 36, no. 5, pp. 1012–1025, 2014.

[4] H. Badino, U. Franke, and D. Pfeiffer, "The Stixel world - a compact medium level representation of the 3D-world," in *DAGM*, 2009.

[5] D. Pfeiffer and U. Franke, "Towards a global optimal multi-layer Stixel representation of dense 3D data," in *BMVC*, 2011.

[6] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Fast stixel computation for fast pedestrian detection," in *CVVT Workshop (ECCV)*, 2012.

[7] M. Cordts, L. Schneider, M. Enzweiler, U. Franke, and S. Roth, "Object-level priors for Stixel generation," in *GCPR*, 2014.

[8] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Extending the Stixel world with online self-supervised color modeling for road-versus-obstacle segmentation," in *ITSC*, 2014.

[9] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Stixmantics: a medium-level model for real-time semantic scene understanding," in *ECCV*, 2014.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for semantic urban scene understanding," in *CVPR*, 2016, to appear.

[11] F. Erbs, B. Schwarz, and U. Franke, "Stixmentation - Probabilistic Stixel based traffic scene labeling," in *BMVC*, 2012.

[12] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *CVPR*, 2012.

[13] M. Muffert, N. Schneider, and U. Franke, "Stix-Fusion: a probabilistic Stixel integration technique," in *CRV*, 2014.

[14] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *Trans. PAMI*, vol. 30, no. 2, pp. 328–341, 2008.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[16] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *IJCV*, vol. 111, no. 1, pp. 96–136, 2014.

[17] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015.

[18] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *ICCV*, 2015.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. P. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *ICLR*, 2015.

[20] G. Lin, C. Shen, I. Reid, and A. van den Hengel, "Efficient piecewise training of deep structured models for semantic segmentation," in *arXiv:1504.01013v2 [cs.CV]*, 2015.

[21] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv:1511.02680v1 [cs.CV]*, 2015.

[22] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *CVPR*, 2015.

[23] A. Sharma, O. Tuzel, and J. W. David, "Deep hierarchical parsing for semantic segmentation," in *CVPR*, 2015.

[24] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *IJCV*, vol. 47, no. 1, 2002, pp. 7–42.

[25] M. Menze and A. Geiger, "Object scene flow for Autonomous Vehicles," in *CVPR*, 2015.

[26] L. Ladický, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *CVPR*, 2014.

[27] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denoeux, "Multimodal information fusion for urban scene understanding," in *MVA*, 2014.

[28] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. S. Torr, "Joint optimization for object class segmentation and dense stereo reconstruction," in *IJCV*, vol. 100, no. 2, 2012, pp. 122–133.

[29] W. Chen, J. Hou, M. Zhang, Z. Xiong, and H. Gao, "Semantic stereo: integrating piecewise planar stereo with segmentation and classification," in *ICIST*, 2014.

[30] F. Gney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *CVPR*, 2015.

[31] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3D scene reconstruction and class segmentation," in *CVPR*, 2013.

[32] G. Floros and B. Leibe, "Joint 2D-3D temporally consistent semantic segmentation of street scenes," in *CVPR*, 2012.

[33] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. Rehg, "Joint semantic segmentation and 3D reconstruction from monocular video," in *ECCV*, 2014.

[34] D. Levi, N. Garnett, and E. Fetaya, "StixelNet: a deep convolutional network for obstacle detection and road segmentation," in *BMVC*, 2015.

[35] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Color-based free-space segmentation using online disparity-supervised learning," in *ITSC*, 2015.

[36] T. Scharwächter and U. Franke, "Low-level fusion of color, texture and depth for robust road scene understanding," in *IV Symposium*, 2015.

[37] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks," *arXiv:1503.02351 [cs.CV]*, 2015.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[39] D. Pfeiffer, S. K. Gehrig, and N. Schneider, "Exploiting the power of stereo confidences," in *CVPR*, 2013.

[40] S. K. Gehrig, R. Stalder, and N. Schneider, "A flexible high-resolution real-rime low-power stereo vision engine," in *ICVS*, 2015.

[41] H. He and B. Upcroft, "Nonparametric semantic segmentation for 3D street scenes," in *IROS*, 2013.

[42] G. Ros, S. Ramos, M. Granados, D. Vazquez, and A. M. Lopez, "Vision-based offline-online perception paradigm for autonomous driving," in *WACV*, 2015.

[43] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr, "Urban 3D semantic modelling using stereo vision," in *ICRA*, 2013.

[44] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denoeux, "Information fusion on oversegmented images: an application for urban scene understanding," in *MVA*, 2013.

[45] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor fusion for semantic segmentation of urban scenes," in *ICRA*, 2015.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[47] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," in *arXiv:1408.5093v1 [cs.CV]*, 2014.