Stixmantics: A Medium-Level Model for Real-Time Semantic Scene Understanding

Timo Scharwächter[1,2], Markus Enzweiler[1], Uwe Franke[1], and Stefan Roth[2]

[1] Environment Perception, Daimler R&D, Sindelfingen, Germany
[2] Department of Computer Science, TU Darmstadt, Germany

Abstract. In this paper we present *Stixmantics*, a novel medium-level scene representation for real-time visual semantic scene understanding. Relevant scene structure, motion and object class information is encoded using so-called *Stixels* as primitive elements. Sparse feature-point trajectories are used to estimate the 3D motion field and to enforce temporal consistency of semantic labels. Spatial label coherency is obtained by using a CRF framework.

The proposed model abstracts and aggregates low-level pixel information to gain robustness and efficiency. Yet, enough flexibility is retained to adequately model complex scenes, such as urban traffic. Our experimental evaluation focuses on semantic scene segmentation using a recently introduced dataset for urban traffic scenes. In comparison to our best baseline approach, we demonstrate state-of-the-art performance but reduce inference time by a factor of more than 2,000, requiring only 50 ms per image.

Keywords: semantic scene understanding, bag-of-features, region classification, real-time, stereo vision, stixels.

1 Introduction

Robust visual scene understanding is one of the fundamental requirements for artificial systems to interpret and act within a dynamic environment. Essentially, two main levels of scene and object representation have been proposed, with contradicting benefits and drawbacks.

Object-centric approaches, *i.e.* sliding window detectors [9,15], have shown remarkable recognition performance due to strong scene and geometric model constraints (holistic or deformable bounding-boxes), easy cue integration and strong temporal tracking-by-detection models. The scene content is represented very concisely as a set of individual detected objects. However, the generalization to partial occlusion cases, object groups or geometrically not well-defined classes, such as road surface or building, is difficult.

Region-centric models, *i.e.* (semantic) segmentation approaches, such as [6,20,28,37,44] among many others, operate in a bottom-up fashion and usually do not recover an object-level scene description. They are rather generic in terms of the geometry and the number of object classes involved. However, grouping

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 533–548, 2014.

Fig. 1. Different scene representation levels trading-off specificity (object-centric) and generality (region-centric). We advocate the use of a medium-level *scene-centric* model to balance this trade-off and gain efficiency.

is based on pixel-level intensity, depth or motion discontinuities with only few geometry or scene constraints leading to noise in the recovered scene models. Furthermore, segmentation approaches are often computationally expensive and the final representation, a pixel-wise image labeling, is overly redundant for many real-world applications, *e.g.* mobile robotics or intelligent vehicles.

To balance this trade-off between specificity (objects) and generality (regions), as shown in Fig. 1, we consider the ideal model for visual semantic scene understanding to be a medium-level *scene-centric* representation that builds upon the strengths of both object-centric and region-centric models. The framework we propose in this paper, called Stixmantics, is based on the *Stixel World* [35], a compact environment representation computed from dense disparity maps. The key aspect that qualifies Stixels as a good medium-level representation is simply the fact that it is based on depth information and that it maps the observed scene to a well-defined model of ground surface and upright standing objects. This makes it adhere more to boundaries of actual objects in the scene than standard superpixels. Yet, in contrast to object-centric approaches, the separation into thin stick-like elements (the Stixels) retains enough flexibility to handle complex geometry and partial occlusions. More precisely, a Stixel models a part of an elevated (upright standing) object in the scene and is defined by its 3D foot point, height, width and distance to the camera.

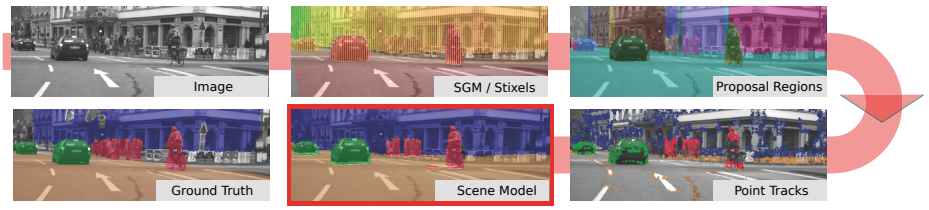

Fig. 2. System overview. A spatio-temporally regularized medium-level scene model (bottom center) is estimated in real-time. This model represents the scene in terms of 3D scene structure, 3D velocity and semantic class labels at each Stixel. Dense stereo and Stixel visualization (top center) is color-encoded depending on distance. Stixel-based proposal regions for classification are shown in false-color (top right). In all other images, colors represent semantic object classes.

In this work, we augment this Stixel representation with spatio-temporally regularized semantic object category and motion information, so that our recovered scene model gives easy access to all relevant information about objects in the scene. To this end, we aggregate intensity and depth information through a semantic bag-of-features classification model that takes Stixel-based proposal regions as input. Reliable sparse point trajectories are used to estimate ego-motion corrected 3D velocity and to enforce temporal coherence of semantic labels in a recursive fashion. Finally, spatial consistency is imposed by a conditional random field using individual Stixels as nodes. See Fig. 2 for an overview.

For unparalleled real-time inference in semantic scene understanding, our framework operates on a few hundred Stixels as opposed to millions of pixels without loss of scene representation quality. Although specialized real-time implementations of pixel dense semantic segmentation exist [7,37], we believe only a medium-level aggregation will enable unrivaled computational efficiency.

2 Related Work

Among the wealth of existing literature about semantic segmentation [6,20,28,37] and scene understanding [13,22,45], we focus on models imposing temporal coherency upon the segmentation result. On a broad level, existing work can be distinguished into *offline* and *online* methods. Offline or batch methods, *e.g.* [24], have the potential to yield best possible segmentation performance, as they can access all available information from all time steps during inference. However, being not causal, they cannot be applied to streaming video analysis, which is a requirement for many applications, such as mobile robotics.

In contrast, online methods only require observations from previous time steps. A closer look reveals fundamental differences, mainly separating *recursive* models [12,44,45] from models considering longer time history [10,16,33]. The latter also include models performing inference in a 3D graphical model (space and time) over a stack of several frames.

Furthermore, the position in the processing pipeline and the level of abstraction on which temporal consistency is enforced has several important implications. For example, low-level motion segmentation (detection-by-tracking) methods, such as [10] or [34], can provide temporally stable proposal regions as input for semantic labeling but require prominent motion of an object in the image for proper detection. In [31], a post-processing algorithm for causal temporal smoothing of frame-by-frame segmentation results is proposed. Requiring dense optical flow, temporal contributions are weighted according to a pixel-wise similarity measure.

Over the last years, prevalent consensus has emerged that increasing the level of abstraction from pixels to superpixels or larger image regions allows for richer models and more efficient inference. In fact, most state-of-the-art methods rely on superpixels, *e.g.* [3,6]. However, as superpixels are typically built in a bottom-up fashion, their boundaries often fluctuate when applied to consecutive frames in a video sequence. The difficulty of registering and aligning superpixels over

time has recently been addressed by [8] and [43]. Alternatively, using the warped previous segmentation as initialization for superpixels in the current frame is exploited in [1] and [30]. In [40], spatio-temporal segments are extracted from video data and are subsequently ranked according to weakly labeled categories provided with the video. However, even with perfect temporal registration of superpixels and object shapes, the semantic label decision can still be incorrect, mainly due to temporal noise in the classification results [36,41]. We provide a method to encourage temporal label coherence which is independent of the used superpixel approach.

Further related work addresses the recovery of a rough 3D scene layout from single images [26] or from video data using structure-from-motion point clouds [5]. None of these methods exploits dense stereo as well as motion information at the same time.

We consider the main contribution of this paper to be the spatio-temporally regularized Stixmantics scene model, where structure, motion and semantics are aggregated on local scene elements, the so-called Stixels. Most closely related to this work is [36], which also combines Stixels and a bag-of-features classification scheme for semantic segmentation. However, their model does not use motion information at all and includes neither spatial nor temporal regularization. Another related Stixel-based approach is [12], where Stixels are grouped into objects, based on discrete motion directions only. This method does not involve any semantic information. Furthermore, we rely on long-term point trajectories for temporal label coherence by applying a recursive Bayesian filtering scheme on trajectory-level. This effectively combines the efficiency of recursive methods [12,13,44,45] with the robustness of considering a longer temporal window [16,33].

Our Stixmantics approach delivers state-of-the-art performance on the public Daimler Urban Segmentation Dataset but at a fraction of the computational cost of previously reported methods. In the same way as Stixels have dramatically improved processing speeds for object detection [4,11] and motion segmentation [12], we demonstrate a real-time method for semantic 3D scene understanding.

3 Generation and Classification of Proposal Regions

One of the foundations of our framework is a bag-of-features model to classify free-form proposal regions. In this context, meaningful initial regions are crucial for semantic segmentation performance [3,6,36]. To rapidly obtain good regions, we follow the method of [36], who first showed how to leverage medium-level depth information of the Stixel World for semantic scene understanding. In this section, we briefly summarize their work, which is the foundation for our Stixmantics model.

Semi-global matching (SGM) [25] is used to obtain dense disparity maps and from that the Stixel representation is computed as described in [35]. To efficiently obtain larger proposal regions R_k, the authors group Stixels according to their 3D spatial proximity, leveraging the fact that Stixels model upright standing

objects on the ground surface at different distances. This approach comes with a bias towards objects on the ground, which however holds true for outdoor traffic scenes and in fact strongly helps to regularize the resulting scene representation. For more details, the reader is referred to [36].

To describe the appearance information within the resulting free-form proposal regions, dense multi-scale SIFT descriptors [29] are encoded with extremely randomized clustering forests and finally pooled over each region R_k. Height information, another medium-level cue, is incorporated in terms of *height pooling*, where visual words are pooled into different locations in the histogram according to their height above the ground plane to introduce a vertical geometric ordering into the descriptor [36]. The resulting bag-of-features region histogram $\mathbf{h}(R_k)$ is subsequently classified by means of a multi-class SVM with histogram-intersection kernel. We refer to the per-label classification confidence for region R_k as $\Gamma_i(\,\mathbf{h}(R_k)\,) \in (0, 1)$, with $i = 1 \ldots L$ and L denoting the number of labels.

This model, as introduced in [36], does not contain any spatial or temporal integration of semantic class labels. Our proposed Stixmantics model fills this gap. Note that the ideas presented in the next sections are conceptually independent from the particular proposal generation, feature descriptor, codebook method or classifier.

4 Temporal Filtering on Trajectory-Level

In order to obtain a temporally consistent representation, we propose a method to efficiently incorporate knowledge from previous time steps into our medium-level scene model. In contrast to existing methods, which aim to filter the semantic label information either densely on pixel-level or focus explicitly on registering superpixels over time, we aggregate information over time locally on sparse long-term point trajectories, where correspondence is very reliable and can be computed efficiently. For this we rely on the well-established KLT feature tracker [42]. We deliberately do not use pixel-level dense optical flow, as it can be spatially redundant, is often overly smooth at object boundaries due to the required regularization term and is, despite modern parallel GPU implementations, computationally expensive. This choice is supported by [10], where the combination of sparse point tracks with superpixels results in more efficient models with adequate performance.

In the following, we describe our methods to filter the discrete semantic label decision (Sec. 4.1), as well as continuous velocity information (Sec. 4.2) for each KLT trajectory over time.

4.1 Label Integration

To integrate the semantic region classification output $\Gamma_i(\,\mathbf{h}(R_k)\,)$ from Sec. 3 over time, we opt for a Hidden Markov Model (HMM) approach. We model label transitions as a Markov chain for each trajectory and perform label filtering in

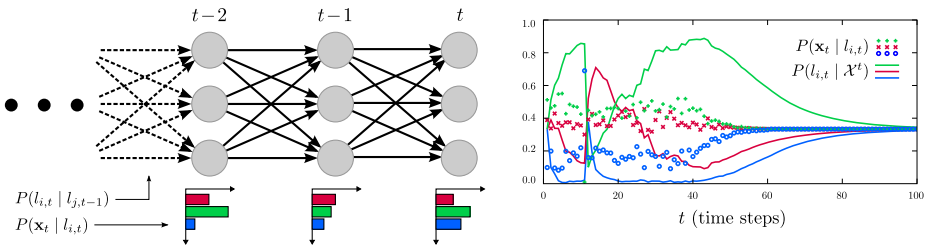

Fig. 3. Illustration of the recursive label filtering scheme, which is applied to each point trajectory. Unfolded directed Markov model with three labels (left). Arrows indicate possible causal transitions. Simulated result for a trajectory of length 100 (time steps), where dots represent noisy observations and solid lines show the resulting filtered posterior for the labels in each time step, given all previously observed data (right). In time step 10, we place a strong outlier in the observation and starting from time step 40, we quickly shift all observations towards a uniform value of $P(\mathbf{x}_t \mid l_{i,t}) = \frac{1}{3}\,\forall i$ to demonstrate the filtering effect of the model. Note that in practice we never observed such strong outliers as simulated in time step 10. The weight α is set to $\alpha = 0.95$, as in all our experiments.

the Bayesian sense, as shown in Fig. 3. For a trajectory with an age of t time steps, we estimate the posterior $P(l_{i,t} \mid \mathcal{X}^t)$ of label $l_{i,t}$, given the set of all previous and current observations $\mathcal{X}^t = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\}$ up to time t. Prediction is performed using forward inference, which involves recursive application of predict (1) and update (2) steps [32]:

$$P\big(l_{i,t} \mid \mathcal{X}^{t-1}\big) = \sum_j P(l_{i,t} \mid l_{j,t-1})\, P\big(l_{j,t-1} \mid \mathcal{X}^{t-1}\big) \tag{1}$$

$$P\big(l_{i,t} \mid \mathcal{X}^{t}\big) = \frac{P(\mathbf{x}_t \mid l_{i,t})\, P\big(l_{i,t} \mid \mathcal{X}^{t-1}\big)}{\sum_j P(\mathbf{x}_t \mid l_{j,t})\, P(l_{j,t} \mid \mathcal{X}^{t-1})}\,. \tag{2}$$

The term $P(l_{i,t} \mid l_{j,t-1})$ in (1) corresponds to the transition model of labels between two subsequent time steps and acts as the temporal regularizer in our setup. Note that ideally objects do not change their label over time, especially not on trajectory-level. The only two causes for a label change are errors in the observation model $P(\mathbf{x}_t \mid l_{i,t})$ or measurement errors in the trajectory, *i.e.* the tracked point is accidentally assigned to another object. Thus, we assign a relatively large weight $\alpha \in (0, 1)$ to the diagonal entries of the transition matrix (self loops) and a small value to the remaining entries, such that we obtain a proper probability distribution:

$$P(l_{i,t} \mid l_{j,t-1}) = \begin{cases} \alpha & i = j \\ \dfrac{1-\alpha}{L-1} & i \neq j\,. \end{cases} \tag{3}$$

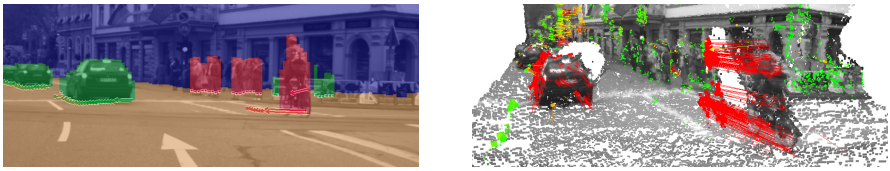

Fig. 4. Scene labeling result with averaged velocity information, indicated by the arrow at the bottom of each Stixel (left). 3D scene reconstruction showing the corresponding Kalman filtered velocity information for each tracked feature point (right). The arrows indicate the predicted position in 500 ms and the color encodes velocity from slow (green) to fast moving (red).

We empirically choose $\alpha = 0.95$ in our experiments. Alternatively, the transition probabilities could be learned from training data. However, we point out again that the resulting transitions would only reflect erroneous correspondences in the trajectory and do not correspond to actual semantic object relations.

Following Sec. 3, we model the observation $\mathbf{x}_t$ as the bag-of-features histogram $\mathbf{h}(R_k)$ of the region R_k covering the tracked feature point in time step t. We relate the per-label classification output $\Gamma_i(\,\mathbf{h}(R_k)\,)$ to the observation model given uniform label priors as $P(\mathbf{x}_t \mid l_{i,t}) \propto \Gamma_i(\,\mathbf{h}(R_k)\,)$.

To maintain a constant number of tracks, new trajectories are instantiated in every time step to account for lost tracks. The label prior of a new trajectory is chosen uniformly as $P(l_{i,t=0} \mid \emptyset) = \frac{1}{L}$. In Fig. 3, we illustrate the recursive label filtering process and provide a simulation for better insight into the model behavior. To assign the track-wise filtered label posteriors back to proposal regions, we compute the average posterior over all trajectories a within region R_k, where $A(R_k)$ is the total number of trajectories in the region, *i.e.*

$$\overline{P}(l_i) = \frac{1}{A(R_k)} \sum_{a=1}^{A(R_k)} P_a\big(l_{i,t} \mid \mathcal{X}^t\big) \ . \tag{4}$$

4.2 Kalman Filter Tracking

In addition to the recursive label filtering scheme, we apply a Kalman filter to each trajectory in order to estimate the 3D world position and velocity of the feature point using stereo information. To compensate for apparent motion induced by the moving observer, we obtain odometry information from the vehicle and incorporate ego velocity $v_{\text{ego},t}$ and yaw-rate $\dot{\psi}_t$ into the estimation process.

As with the filtered label decision, we assign the averaged Kalman filtered 3D velocity information to each Stixel using the corresponding proposal region R_k for averaging. Fig. 4 shows a 3D point cloud of the reconstructed scene and each tracked feature point is depicted with an arrow, indicating the predicted position in 500 ms. We adopt the system model and estimation process from [18]. For more details, please consult this paper.

5 Spatial Regularization

One side effect of most data-driven grouping schemes, *e.g.* superpixels or Stixels, is local spatial over-segmentation. To incorporate global smoothness properties, we formulate a conditional random field (CRF) in each time step, where Stixels are connected within a graphical model. More formally, a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consisting of vertices $\mathcal{V}$ and edges $\mathcal{E}$ is built, where $|\mathcal{V}| = N$ is the number of nodes (Stixels). We assign a set of discrete random variables $\mathcal{Y} = \{y_n \mid n = 1 \dots N\}$, where each y_n can take a value of the label set $\mathcal{L} = \{l_i \mid i = 1 \dots L\}$. A labeling $\mathbf{y} \in \mathcal{L}^N$ defines a joint configuration of all random variables assigned to a specific label. In a CRF, the labeling $\mathbf{y}$ is globally conditioned on all observed data $\mathbf{X}$ and follows a Gibbs distribution: $p(\mathbf{y} \mid \mathbf{X}) = \frac{1}{Z}\exp\big(-\sum_{c\in\mathcal{C}} \psi_c(\mathbf{y}_c)\big)$, where Z is a normalizing constant, $\mathcal{C}$ is the set of maximal cliques, $\mathbf{y}_c$ denotes all random variables in clique c and $\psi_c(\mathbf{y}_c)$ are potential functions for each clique [27], having the data $\mathbf{X}$ as implicit dependency.

Finding the maximum a posteriori (MAP) labeling $\hat{\mathbf{y}}$ is equivalent to finding the corresponding minimum Gibbs energy, *i.e.* $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{X}) = \arg\min_{\mathbf{y}} E(\mathbf{y})$. For semantic labeling problems, the energy function $E(\mathbf{y})$ typically consists of unary (ψ_n) and pairwise (ψ_{nm}) potentials and is defined as

$$E(\mathbf{y}) = \sum_{n\in\mathcal{V}} \psi_n(y_n) + \sum_{(n,m)\in\mathcal{E}} \psi_{nm}(y_n, y_m) \ . \tag{5}$$

Note that we do not model spatial and temporal consistency jointly to allow better pipelining of the process. Instead, we use the temporally smoothed results from Sec. 4.1 during inference to additionally facilitate spatial smoothness. Inference is performed using five sweeps of α-expansion. In the following, we discuss the potential functions in more detail.

5.1 Unary Potentials

For the unary potential of a vertex, we employ the filtered label posterior from Sec. 4.1, averaged over the corresponding proposal region as

$$\psi_n(y_n = l_i) = -\log\big(\,\overline{P}(l_i)\,\big) \ . \tag{6}$$

Note that this unary potential not only incorporates data from a single Stixel locally but from the larger proposal region. In CRFs (compared to MRFs) the labeling $\mathbf{y}$ is globally conditioned on the data $\mathbf{X}$, so this is a valid choice to increase the robustness of the unary term.

5.2 Pairwise Potentials

To encourage neighboring Stixels to adopt the same label, the pairwise potentials take the form of a modified Potts model. In pixel-level CRFs, the Potts model is often extended to be contrast-sensitive to disable smoothing when strong changes

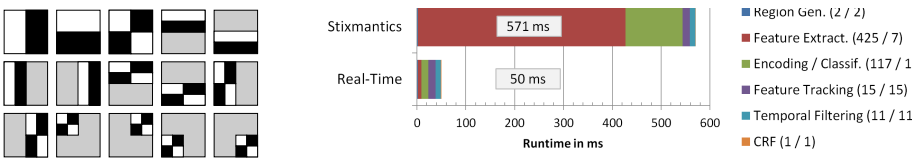

Fig. 5. 8×8 pixel Haar wavelet basis feature set (left). White, black and gray areas denote weights of +1, -1 and 0, respectively. Adapted from [38]. Average runtime in milliseconds of the proposed Stixmantics approach and the real-time variant (right).

in image intensities occur, *e.g.* [39]. In contrast, our Stixel-level model allows us to make smoothing sensitive to more concise and less noisy measures such as spatial proximity or different motion direction. Here, we propose a measure of common boundary length between Stixels as a proxy for spatial proximity. Hence, we define the pairwise potentials as

$$\psi_{nm}(y_n, y_m) = \begin{cases} 0 & y_n = y_m \\ \gamma\, \Omega(n,m) & y_n \neq y_m, \end{cases} \tag{7}$$

where γ is a smoothness factor. The term $\Omega(n,m)$ measures the affinity of adjacent Stixels. For two adjacent Stixels n and m, let b_n be the number of pixels on the boundary of Stixel n. Further let c_{nm} be the number of pixels on the boundary of Stixel n, which are direct neighbors of Stixel m. The terms b_m and c_{mn} are defined accordingly. The common boundary length is then defined as

$$\Omega(n,m) = \frac{c_{nm} + c_{mn}}{b_n - c_{nm} + b_m - c_{mn}}\,. \tag{8}$$

By definition, the measure is symmetric and limited to the range $[0, 1)$ for non-overlapping Stixels and the cost of a label change is reduced if two adjacent Stixels only have a small common boundary.

6 Going Real-Time

Our approach is implemented single-threaded in C++ using an Intel i7-3.33 GHz CPU and an NVIDIA GeForce GTX 770 GPU (for KLT feature tracking). We use a pipelining strategy where SGM stereo and Stixel computation are performed on dedicated FPGA hardware [17,21], which effectively increases the system latency by two frames (100 ms at a framerate of 20 Hz). For clarity of presentation, we set aside this additional delay in our reported timings as it is equally present in all systems except for the first baseline from [28].

Fig. 5 (right) shows, that SIFT feature extraction (red) and random forest encoding coupled with SVM classification (green) are the computational bottlenecks of the system. To address those bottlenecks, we replace the multi-scale SIFT features by simpler and faster descriptors. In particular, we use single-scale 8×8 pixel features derived from a 2D Haar wavelet basis resulting in a

15-dimensional descriptor [38], see Fig. 5 (left). Additionally, a weaker random forest encoder is applied using 5 trees with 50 leaves (instead of 500 leaves) each. Encoding is faster due to shallower trees and the shorter histograms also result in faster SVM classification. In doing so, the computation time for semantic class labeling is greatly reduced by an order of magnitude to 50 ms per image on average, see Fig. 5 (right). In total, our whole pipelined system is thus able to operate at a real-time framerate of 20 Hz. This includes the full estimation of 3D scene structure and motion (SGM, Stixels and KLT tracking) in additional to semantic class labeling.

7 Experiments

7.1 Dataset

For experimental evaluation we use the public Daimler Urban Segmentation Dataset[1] and the corresponding evaluation methodology as introduced in [36]. This dataset contains 5,000 stereo images captured in urban traffic with vehicle ego-motion data, where every 10-th image has exact pixel-wise semantic annotation, see Fig. 7. Note that other public datasets such as PASCAL VOC [14], MSRC [37] or KITTI [23] do not have all necessary data, *i.e.* stereo images, odometry and semantic labeling, available at the same time.

7.2 Discussion of Baselines

To provide adequate baselines, we use the publicly available ALE software with the framework proposed in [28] to exploit depth information. This method uses a bank of several local feature descriptors and performs joint optimization of class labels and dense disparity maps and arguably provides state-of-the-art performance in this domain (Joint-Optim. ALE in Table 1). For an additional external baseline, we apply Iterative Context Forests [19], adapted by adding the disparity image as additional feature channel and Stixel segments to regularize the per-pixel classification result (Depth-enabled ICF in Table 1). Note, that this baseline compared favorably to many other approaches on different datasets and in related applications [19]. As a final reference, we re-implemented the method of [36]. Here, Stixels are also used to create proposal regions but the whole system does neither incorporate any temporal analysis nor spatio-temporal regularization. We also provide numbers for the variant, where intensity-based SLIC superpixels [2] are used instead of Stixel-based proposal regions. The variants are called Stixbaseline and SLICbaseline in Table 1 respectively, where both correspond to the $\mathrm{ERC}_{\mathrm{G}}^{\mathrm{HP}}$ version in [36]. Note that with our re-implementation, we slightly improve over their originally reported results. In contrast to [36], we improve the segmentation of sky regions by including a prior based on location and intensity.

[1] The dataset is available at `http://www.6d-vision.com/scene-labeling/`

Table 1. Semantic segmentation results (PASCAL VOC IU measure) for all considered approaches. The best result per class is marked in **boldface**, the second best in *italics*. Besides the average performance over all classes, we additionally give the average for the most application-relevant dynamic object classes only, *i.e.* vehicle and pedestrian. We additionally report the computation time per frame for each method, where SLIC related timings assume a real-time implementation of SLIC superpixels.

	Baselines			
Class \ Method	Joint-Optim. ALE [28]	Depth-enabled ICF [19]	SLICbaseline [36]	Stixbaseline [36]
Ground	**89.9**	86.2	81.4	87.5
Vehicle	63.8	53.5	49.8	66.2
Pedestrian	**63.6**	34.9	40.4	53.4
Sky	**86.7**	35.1	27.1	51.4
Building	59.1	53.9	52.6	**61.1**
Average (all)	**72.6**	52.8	50.2	63.9
Average (dyn)	*63.7*	44.2	45.1	59.8
Runtime/frame	111 s	3.2 s	544 ms	544 ms

	This paper			
Class \ Method	Stixmantics (real-time)	Stixmantics (real-time/NR)	SLICmantics	Stixmantics
Ground	*87.6*	*87.6*	87.4	*87.6*
Vehicle	*67.4*	61.8	60.9	**68.9**
Pedestrian	57.8	51.5	47.4	*59.0*
Sky	*61.4*	55.2	48.0	57.6
Building	60.1	60.9	54.2	*60.2*
Average (all)	*66.9*	63.4	59.6	66.7
Average (dyn)	62.6	56.7	54.2	**64.0**
Runtime/frame	*50 ms*	**23 ms**	571 ms	571 ms

7.3 Results

We compare four of our own system variants against the four baselines discussed above using identical data and evaluation criteria. Although we estimate a full medium-level scene model including 3D structure, 3D motion and semantic labeling, we focus on evaluating the semantic segmentation performance at this point. Segmentation accuracy is evaluated using the standard PASCAL VOC intersection-over-union (IU) measure [14]. Temporal regularization is evaluated using an additional object-centric score. In Table 1 (bottom) we show our four system variants. In the first and second column, we provide results for the Stixmantics real-time version, and the real-time version without spatio-temporal

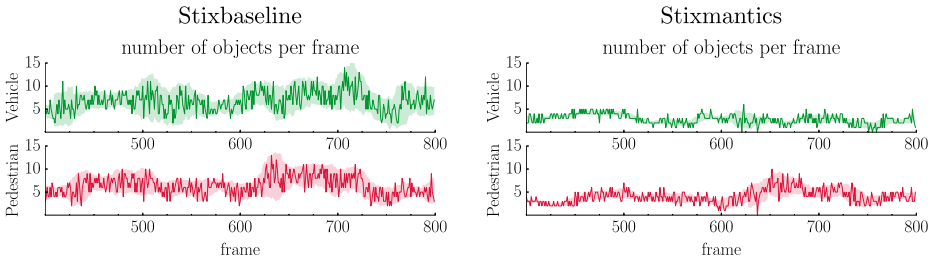

Fig. 6. Number of detected objects per frame, for the Stixbaseline (left) and our Stixmantics approach (right). We show an excerpt from frame $400 - 800$ of the `test_2` sequence for the dynamic object classes, *i.e.* vehicle and pedestrian. The TF_i score is shown as solid band in the background, illustrating the strength of local temporal fluctuations in the labeling decision. Our Stixmantics model clearly provides stronger temporal consistency.

regularization (NR). Furthermore, we show results when Stixel-based proposal regions are replaced with SLIC superpixels computed on the intensity images (SLICmantics). Finally, we show results for our full Stixmantics model.

One problem with the IU measure is its pixel-wise nature. It is strongly biased towards close objects that take up large portions of the image and small objects of the same class barely contribute to the final score. The same holds true for small temporal fluctuations in the result. To account for this fact, we provide an additional object centric evaluation to support the spatio-temporal regularization proposed in this paper. Fig. 6 shows the number of detected objects o_i over the frames i, for the Stixbaseline approach (left) and our Stixmantics approach (right). To approximate the number of objects, we count each closed image region with identical class label as one instance. Although the absolute number of objects is non-informative without ground-truth data, a lower temporal fluctuation in the number of objects indicates stronger temporal consistency. We define a temporal fluctuation measure TF_i at frame i as sliding mean squared deviation to the sliding average $\overline{o}_j$:

$$\mathrm{TF}_i = \frac{1}{2w+1} \sum_{j=i-w}^{i+w} (o_j - \overline{o}_j)^2 \quad \text{with } \overline{o}_j = \frac{1}{2w+1} \sum_{k=j-w}^{j+w} o_k \,, \tag{9}$$

where w is the temporal window size and is set to 10 frames in our evaluation. We show the TF_i score as $\overline{o}_i \pm \mathrm{TF}_i$ for each frame i as solid band in the background of Fig. 6. In Table 2, we show the averaged TF score over all 2,000 frames of the test sequences. Qualitative results of our Stixmantics framework are shown in Fig. 7.

From the reported results, we draw several conclusions. First, our proposed Stixmantics approach delivers state-of-the-art performance but requires only a fraction of the computational costs compared to all baseline methods, with the real-time variant being several orders of magnitude faster than Joint-Optim. ALE [28]. For the vehicle class, it even outperforms the method of [28] on this

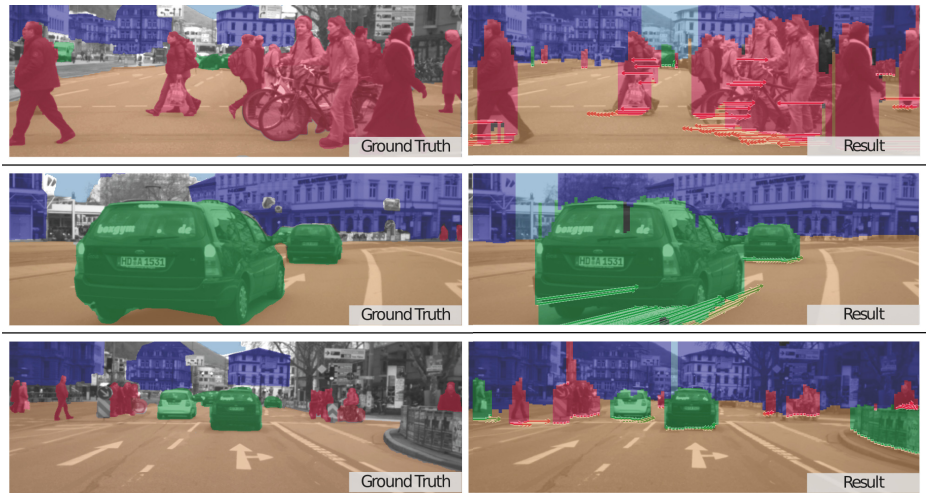

Fig. 7. Comparison of results obtained with our Stixmantics model against ground-truth labels. Colors denote the recovered semantic object classes. Arrows at the bottom of each underlying Stixel depict the ego-motion corrected 3D velocities of other traffic participants. Images are taken from the real-time version of our approach.

dataset. As apparent from Fig. 6 and Table 2, our regularized Stixmantics model consistently outperforms Stixbaseline throughout all object classes except for the Sky class w.r.t. the TF measure. The average numbers in Table 1 also support this.

We also observe that for dynamic objects the benefit of regularization is more pronounced when using simpler features, *cf*. Stixmantics (real-time) *vs*. Stixmantics (real-time/NR) with Stixmantics *vs*. Stixbaseline. We take this as evidence for the strength of our integrated medium-level model, where weaker classification performance can be compensated for by stronger constraints on the model. Non-surprisingly, the gain of temporal integration is also stronger when SLIC superpixels are utilized instead of Stixel-based proposal regions, given that they are inherently less temporally consistent than Stixels, *cf*. SLICmantics *vs*. SLICbaseline with Stixmantics *vs*. Stixbaseline. In general, results improve significantly when Stixel-based proposal regions are used.

Table 2. Average TF measure per class for the Stixbaseline and our Stixmantics approach. Lower scores indicate less temporal fluctuation. The best result per class is marked in **boldface**.

Method ╲ Class	Ground	Vehicle	Pedestrian	Sky	Building
Stixbaseline [36]	0.25	1.92	1.98	**2.43**	0.67
Stixmantics	**0.18**	**0.67**	**0.53**	3.10	**0.25**

8 Conclusion

In this paper, we presented a novel comprehensive scene understanding model that can be computed in 50 ms from stereo image pairs. At the same time, we achieve close to state-of-the-art performance in semantic segmentation of urban traffic scenes. Our spatio-temporally coherent model extracts application-relevant scene content and encodes it in terms of the medium-level Stixel representation with 3D position, height, 3D velocity and semantic object class information available at each Stixel. From a mobile vision and robotics application point-of-view, the richness, flexibility, compactness and efficiency of the proposed scene description make it an ideal candidate to serve as a generic interface layer between raw pixel values and higher level reasoning algorithms, such as for path planning and localization.

References

1. Abramov, A., Pauwels, K., Papon, J., Worgotter, F., Dellen, B.: Real-Time Segmentation of Stereo Videos on a Portable System With a Mobile GPU. IEEE Transactions on Circuits and Systems for Video Technology 22(9) (2012)
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC Superpixels Compared to State-of-the-art Superpixel Methods. Trans. PAMI 34(11) (2012)
3. Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic Segmentation using Regions and Parts. In: CVPR (2012)
4. Benenson, R., Mathias, M., Timofte, R., Gool, L.V.: Fast Stixel Computation for Fast Pedestrian Detection. In: CVVT Workshop, ECCV (2012)
5. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and Recognition Using Structure from Motion Point Clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
6. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic Segmentation with Second-Order Pooling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 430–443. Springer, Heidelberg (2012)
7. Costea, A., Nedevschi, S.: Multi-Class Segmentation for Traffic Scenarios at Over 50 FPS. In: IV Symposium. pp. 1–6 (2014)
8. Couprie, C., Farabet, C., LeCun, Y.: Causal Graph-based Video Segmentation. In: ICIP (2013)
9. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian Detection: An Evaluation of the State of the Art. Trans. PAMI 34(4) (2012)
10. Ellis, L., Zografos, V.: Online Learning for Fast Segmentation of Moving Objects. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part II. LNCS, vol. 7725, pp. 52–65. Springer, Heidelberg (2013)
11. Enzweiler, M., Hummel, M., Pfeiffer, D., Franke, U.: Efficient Stixel-Based Object Recognition. In: IV Symposium (2012)
12. Erbs, F., Schwarz, B., Franke, U.: Stixmentation - Probabilistic Stixel based Traffic Scene Labeling. In: BMVC (2012)

13. Ess, A., Mueller, T., Grabner, H., Gool, L.V.: Segmentation-Based Urban Traffic Scene Understanding. In: BMVC (2009)
14. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. IJCV 88(2) (2010)
15. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. Trans. PAMI 32(9) (2010)
16. Floros, G., Leibe, B.: Joint 2D-3D Temporally Consistent Semantic Segmentation of Street Scenes. In: CVPR (2012)
17. Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R.G.: Making Bertha See. In: CVAD Workshop, ICCV (2013)
18. Franke, U., Rabe, C., Badino, H., Gehrig, S.K.: 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 216–223. Springer, Heidelberg (2005)
19. Fröhlich, B., Rodner, E., Denzler, J.: Semantic Segmentation with Millions of Features: Integrating Multiple Cues in a Combined Random Forest Approach. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 218–231. Springer, Heidelberg (2013)
20. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: ICCV (2009)
21. Gehrig, S.K., Eberli, F., Meyer, T.: A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 134–143. Springer, Heidelberg (2009)
22. Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R.: 3D Traffic Scene Understanding from Movable Platforms. Trans. PAMI (2013)
23. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CVPR (2012)
24. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient Hierarchical Graph-based Video Segmentation. In: CVPR (2010)
25. Hirschmüller, H.: Stereo Processing by Semiglobal Matching and Mutual Information. Trans. PAMI 30(2) (2008)
26. Hoiem, D., Efros, A.A., Hebert, M.: Closing the Loop in Scene Interpretation. In: CVPR (2008)
27. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. The MIT Press (2009)
28. Ladický, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.H.S.: Joint Optimisation for Object Class Segmentation and Dense Stereo Reconstruction. In: BMVC (2010)
29. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV 60(2) (2004)
30. Mester, R., Conrad, C., Guevara, A.: Multichannel Segmentation Using Contour Relaxation: Fast Super-Pixels and Temporal Propagation. In: Heyden, A., Kahl, F. (eds.) SCIA 2011. LNCS, vol. 6688, pp. 250–261. Springer, Heidelberg (2011)
31. Miksik, O., Munoz, D., Bagnell, J.A., Hebert, M.: Efficient Temporal Consistency for Streaming Video Scene Analysis. In: ICRA (2013)
32. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. The MIT Press (2012)
33. de Nijs, R., Ramos, S., Roig, G., Boix, X., Gool, L.V., Kühnlenz, K.: On-line Semantic Perception using Uncertainty. In: IROS (2012)
34. Ochs, P., Brox, T.: Object Segmentation in Video: A Hierarchical Variational Approach for Turning Point Trajectories into Dense Regions. In: ICCV (2011)

35. Pfeiffer, D., Franke, U.: Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In: BMVC (2011)
36. Scharwächter, T., Enzweiler, M., Franke, U., Roth, S.: Efficient Multi-cue Scene Segmentation. In: Weickert, J., Hein, M., Schiele, B. (eds.) GCPR 2013. LNCS, vol. 8142, pp. 435–445. Springer, Heidelberg (2013)
37. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. IJCV (2009)
38. Stollnitz, E.J., DeRose, T.D., Salesin, D.H.: Wavelets for Computer Graphics: A Primer. IEEE Computer Graphics and Applications 15 (1995)
39. Sturgess, P., Alahari, K., Ladický, L., Torr, P.H.S.: Combining Appearance and Structure from Motion Features for Road Scene Understanding. In: BMVC (2009)
40. Tang, K., Sukthankar, R., Yagnik, J., Fei-Fei, L.: Discriminative Segment Annotation in Weakly Labeled Video. In: CVPR. IEEE (2013)
41. Tighe, J., Lazebnik, S.: SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 352–365. Springer, Heidelberg (2010)
42. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University (1991)
43. Vazquez-Reina, A., Avidan, S., Pfister, H., Miller, E.: Multiple Hypothesis Video Segmentation from Superpixel Flows. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 268–281. Springer, Heidelberg (2010)
44. Wojek, C., Schiele, B.: A Dynamic Conditional Random Field Model for Joint Labeling of Object and Scene Classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 733–747. Springer, Heidelberg (2008)
45. Wojek, C., Walk, S., Roth, S., Schindler, K., Schiele, B.: Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes. Trans. PAMI 35(4) (2013)