2011 IEEE Intelligent Vehicles Symposium (IV)
Baden-Baden, Germany, June 5-9, 2011

Moving Vehicle Detection by Optimal Segmentation of the Dynamic Stixel World

Friedrich Erbs, Alexander Barth and Uwe Franke

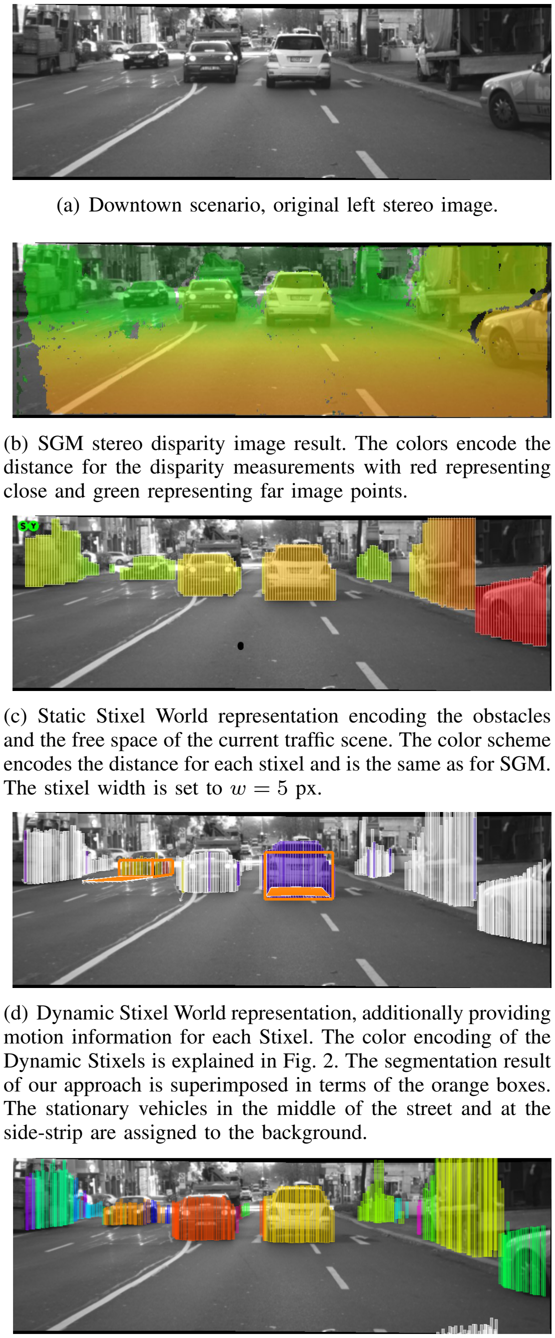
***Abstract*— The reliable detection of moving objects from a moving observer is one of the most challenging and important tasks for driver assistance and safety systems. Modern sensors such as Lidar, Imaging Radar or Stereo Vision deliver range data plus longitudinal motion (Radar) or even full 3D-motion (space-time vision). Based on this data, moving objects have to be separated from the static background to be able to determine their pose and motion state. Usually, heuristics are applied to cluster the data. In order to find the most probable segmentation, we formulate the task as a hypotheses testing problem that allows taking into account various constraints and assumptions simultaneously. We show that the optimal segmentation can be efficiently found by means of dynamic programming, for an arbitrary number of objects in the scene. In this paper we concentrate on the segmentation of space-time data obtained from stereo image sequences. The vision-based depth and motion information is transferred into so called *Stixels*, a very compact representation of 3D scenes that can also be applied to Lidar or Radar data. It turns out that our optimal segmentation is more robust w.r.t. noisy and erroneous data.**

I. INTRODUCTION

Detecting and tracking of other traffic participants such as vehicles and pedestrians from a moving vehicle is an essential task for driver assistance and autonomous navigation, and it provides an important basis for further traffic scene understanding [1], [2]. This problem is highly challenging especially due to strongly varying lighting conditions, highly dynamic background and the low viewpoint of the vehicle-mounted cameras.

In this contribution, we propose a robust probabilistic approach to detect and segment an arbitrary number of moving vehicle objects from space-time data obtained from stereo image sequences. Here we use the *Dynamic Stixel World* representation [3] as an efficient and compact one-dimensional modeling of real world 3d road scenes as input data to our approach. The Stixel world reconstructs 3D traffic scenes column-wise by a set of rectangular sticks, called Stixels, limiting the drivable freespace and approximating object boundaries. Furthermore, the dynamic Stixels provide motion information about the scene dynamics, see Fig. 1(d) for an example. However, our approach is not restricted to the Stixel representation, it can be applied analogously to other sensor data, e.g. obtained from Radar or Lidar measurements.

Many classical approaches for object segmentation such as K-means or Mean-Shift are data-driven and based on ideas


(a) Downtown scenario, original left stereo image.

(b) SGM stereo disparity image result. The colors encode the distance for the disparity measurements with red representing close and green representing far image points.

(c) Static Stixel World representation encoding the obstacles and the free space of the current traffic scene. The color scheme encodes the distance for each stixel and is the same as for SGM. The stixel width is set to $w = 5$ px.

(d) Dynamic Stixel World representation, additionally providing motion information for each Stixel. The color encoding of the Dynamic Stixels is explained in Fig. 2. The segmentation result of our approach is superimposed in terms of the orange boxes. The stationary vehicles in the middle of the street and at the side-strip are assigned to the background.

(e) Mean Shift segmentation results for comparison with our algorithm of this downtown scenario. Different clusters are colored differently.

Fig. 1. Main steps of our object detection algorithm.

F. Erbs and U. Franke are with the Environment Perception Group of Daimler Research, 71032 Boeblingen, Germany. [friedrich.erbs, uwe.franke]@daimler.com

A. Barth is with Mercedes-Benz Research & Development North America, Palo Alto, CA, USA. alexander.barth@daimler.com

978-1-4577-0891-6/11/$26.00 ©2011 IEEE

from the psychological Gestalt Laws of similarity, proximity and common fate which provide a clustering of the scene into coherent regions [4], [5], [6], [7]. However, without any regularization, the presence of erroneous input data may lead to severe misinterpretations of the scene. Common regularization schemes often include Markov neighborhoods, which are in general very flexible, but not sufficient in many situations to prevent that an object gets split into parts [8], [9], [10].

In this contribution we follow a model-driven hypotheses testing approach, exploiting our prior knowledge about the cuboid-like shape of most cars, in order to gain robustness with respect to noisy and erroneous input data. Several object hypotheses, representing the pose and dimension of a cuboidal object, are generated in the scene, see Fig. 3 for an example. Each object hypothesis is evaluated with respect to the measured Stixel data in a probabilistic sense. We use dynamic programming to efficiently search for the maximum likelihood object configuration that does not contain overlapping cuboids.

The remainder of this article is organized as follows. First, we briefly explain the input data to our algorithm and the necessary preprocessing steps in Sec. II. Sec. III-A gives a short overview on our approach: we show how our cuboid matching approach can be traced back to an interval scheduling problem and we further specify the object detection problem. Sec. III-B explains the hypothesis generation in our framework. Then, in Sec. III-C we analyze the concrete modeling of this cuboid matching process and Sec. III-D shows how to efficiently find the most probable segmentation in real-time by means of dynamic programming. Sec. III-E briefly introduces a Mean-Shift algorithm which is compared with our approach in the experimental results Sec. IV. Finally Sec. V concludes this contribution.

II. Input Data: Dynamic Stixels

A stereo camera system is mounted behind the windshield of our experimental vehicle. Input images from this camera system are rectified to standard geometry with epipolar lines parallel to the image rows and a dense stereo depth map is computed in real-time at 25 Hz on a custom-built FPGA platform using a semi-global matching algorithm, see Fig. 1(b) for an example [12], [13].

Based on this depth map, static Stixels are computed as a medium level representation of traffic scenes, cf. Fig. 1(c). This representation might be considered as a first step of image segmentation and bridges the gap between the pixel and the object level. The Stixel representation is based on the fact that traffic scenes typically consist of an approximately planar free space, which is limited by 3d obstacles that are nearly perpendicular to the ground. The Stixels partition the depth image column-wise into a set of rectangular sticks, having a fixed width (typically 5 pixels). Each Stixel limits the drivable freespace and forms the object boundaries. The base point of the Stixels is obtained from a stochastic occupancy grid using the method presented in [15]. The height of each Stixel is obtained by means of dynamic programming. Given the base point and the height of the Stixel, its distance estimate is refined by averaging the disparity values of all pixels inside the Stixel using a histogram based approach [14]. Moreover, the velocity for dynamic Stixels is estimated by Kalman filtering a representative velocity vector of this Stixel patch over time (Fig. 1(d)). This implies tracking the Stixels as described in [3]. In order to be able to derive absolute velocities, the motion of the ego-vehicle, extracted by the method as described in [16], is compensated.

Let $\mathcal{S} = \{S_n | \; n \in \{1...N\}\}$ denote the set of all N dynamic Stixels S_n, with each Stixel $S_n = [\boldsymbol{X}_n, \boldsymbol{v}_n, H_n]^\mathsf{T}$ containing the lateral and longitudinal position $\boldsymbol{X} = [X, Z]^\mathsf{T}$ of the Stixel's center, relative to the ego-vehicle, an absolute 2D velocity vector $\boldsymbol{v} = [v_x, v_z]^\mathsf{T}$, and the Stixel height over ground H. In the image plane, the Stixels can be described by their horizontal image coordinate u_n of their middle basepoint, their width w and an upper and lower image row coordinate v_n^{Bot} and v_n^{Top}. Note that the set of Stixels $\mathcal{S}$ can be represented as a one-dimensional list, ordered by their horizontal image coordinates. So a group of neighboring Stixels forms an $\mathrm{interval}\ I$.

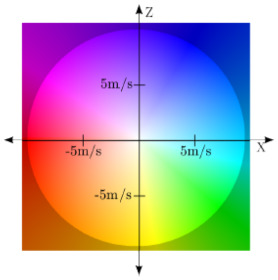


Fig. 2. Dynamic Stixel color encoding of speed and direction. Maximum saturation is reached at 10 m/s. The color value represents the moving direction with respect to the ego-vehicle.

III. APPROACH

A. Overview

The problem of Stixel segmentation can be specified as the most probable assignment of N Stixels to M moving objects $\mathcal{O}^m, \quad m \in \{1...M\}$, plus a background class $\mathtt{BG}$. Our primary objective is to detect and segment these M moving objects. A priori we do not know neither the number of objects M nor their motion states nor their orientations relative to the ego vehicle described by a yaw angle ψ_m (see Fig. 3). Even if we knew M, there would be $(M+1)^N$ possible assignments. Although the Stixel representation brings a significant data reduction from typically $N \approx 300.000$ pixels to $N \approx 200$ Stixels, the computation of $(M+1)^N$ possible assignments is still infeasible. However we can further reduce complexity significantly by exploiting prior knowledge about the segmentation task: it is reasonable to model most cars as a rectangular cuboid $\boldsymbol{D} = [w, l, h]^\mathsf{T}$ with a typical width w, length l and height h [17]. Furthermore we demand cars not to overlap, neither in the image plane nor in the real 3D world.

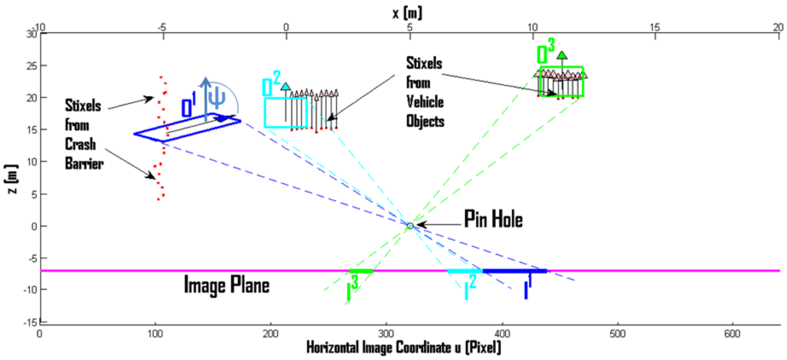


Fig. 3. Schematic illustration of the hypotheses generation procedure: three different vehicle cuboids hypotheses $\mathcal{O}^m$ are placed in the scene consisting of a crash barrier on the left side and two moving vehicle objects. The corresponding Stixels are plotted as red dots in world coordinates, their motion vector visualizes their predicted translation within the next time step. The object hypotheses can be described by their dimensions, orientation and their hypothetical velocity vector illustrated as a thick arrow in the same color as the hypothesis in the middle of the cuboid. These hypotheses are projected into the image plane. The Stixels S^m within these resulting intervals I^m are tested against the two possible labelings $\mathcal{O}^m$ and background. In this case $\mathcal{O}^1$ results in a low object likelihood $p\left(\mathcal{S}^1|\mathcal{L}^1=\texttt{OBJ}\right)$ and a high background likelihood $p\left(\mathcal{S}^1|\mathcal{L}^1=\texttt{BG}\right)$ whereas $\mathcal{O}^3$ yields a high object likelihood and a low background likelihood.

Assuming that we know the current position, orientation and dimensions of this cuboid related to $\mathcal{O}^m$, as well as the camera geometry, we can project it onto the image plane. The resulting leftmost and rightmost horizontal image coordinates u_{left} and u_{right} of the cuboid projection define an interval $I^m = \left[u^m_{\text{left}}, u^m_{\text{right}}\right]$, see Fig. 3.

Now, let S^m denote the Stixel subset containing all Stixels within this interval I^m, and $\mathcal{L}^m$ a binary labeling assigning S^m either the label $\texttt{OBJ}$, indicating the Stixels S^m belong to the object hypothesis $\mathcal{O}^m$, or $\texttt{BG}$ for background. Then, for each hypothesis $\mathcal{O}^m$, we compute the likelihood $p\left(\mathcal{S}^m|\mathcal{L}^m=\texttt{OBJ}\right)$ and $p\left(\mathcal{S}^m|\mathcal{L}^m=\texttt{BG}\right)$, indicating how likely is the given Stixel configuration in S^m, given the Stixels either belong to object hypothesis $\mathcal{O}^m$ or background. This formulation allows for learning the likelihood functions from sufficient training data. Since such data is not available yet, we alternatively approximate these functions based on model knowledge as will be further specified in Sec. III-C.

From all these object hypotheses $\mathcal{O}^m$ which can be transferred into intervals I^m, we find those which do not overlap and which give rise to the highest total likelihood of *all* labelings in the scene. This task of finding the most probable sequence of non-overlapping intervals can be solved efficiently by means of dynamic programming as will be addressed in Sec. III-D.

B. Hypothesis Generation

As explained in Sec. III-A and illustrated in Fig. 3, in each cycle of the algorithm we test several object hypotheses $\mathcal{O}^m$ with different parameters, e.g. several object cuboid orientations $\Delta\psi \in \{0...2\pi\}$ described by the objects's yaw angle ψ and different positions. We do not randomly place object cuboids in the image plane, but we assume that each Stixel S_n might be one of the visible cuboid corners. Depending on the orientation of the cuboid, there are two or three such visible object corners, for example there are two visible corners for $\mathcal{O}^1$ in Fig. 3 and three for $\mathcal{O}^2$. Then it is also possible to vary the object dimensions in discrete steps. The total number of object hypotheses M reflects a tradeoff between accuracy and time performance.

A priori we assume all of these object parameters related to these object hypotheses $\mathcal{O}^m$ to be equiprobable. This hypothesis testing can be scheduled massively in parallel.

C. Hypothesis Evaluation

Three different features are exploited to define the likelihood functions $p\left(\mathcal{S}^m|\mathcal{L}^m\right)$. These likelihood functions include the position $\boldsymbol{X}$ of the Stixels in S^m, the Stixels' velocities $\boldsymbol{v}$, and a so called *existence feature* E, as will be motivated below. The three features are assumed to be independent of each other, yielding

$$p\left(\mathcal{S}^m|\mathcal{L}^m\right) = p\left(\boldsymbol{X}^m \mid \mathcal{L}^m\right) p\left(\boldsymbol{v}^m \mid \mathcal{L}^m\right) p\left(E^m \mid \mathcal{L}^m\right). \tag{1}$$

1) Position Feature $p\left(\boldsymbol{X}^m \mid \mathcal{L}^m\right)$ *:* We compute the positional likelihood $p\left(\boldsymbol{X}^m \mid \mathcal{L}^m=\texttt{OBJ}\right)$ for the m-th hypothetical object by summing up the positional deviations of all Stixels S^m from the assumed cuboid bounding box, scaled by the given uncertainties and our model uncertainty (Mahalanobis distance), as illustrated in Fig. 4. The underlying probability distribution for $p\left(\boldsymbol{X}^m \mid \mathcal{L}^m=\texttt{OBJ}\right)$ is described by a chi-square distribution.

The probability distribution for the background hypothesis $p\left(\boldsymbol{X}^m \mid \mathcal{L}^m=\texttt{BG}\right)$ is formulated as a counter-hypothesis to the current object hypothesis $\mathcal{O}^m$. It is difficult to rate the background hypothesis with regard to its positional likelihood in a way similar to $p\left(\boldsymbol{X}^m \mid \mathcal{L}^m=\texttt{OBJ}\right)$ since there are many possible realizations of this class, e.g. side rails, parking vehicles or buildings. Furthermore erroneous Stixels, called phantom Stixels, which do not actually limit the drivable freespace, might be present and have to be included in the modeling of this background class. So we describe $p\left(\boldsymbol{X}^m \mid \texttt{BG}\right)$ by a broad uniform distribution which reflects this variety of possible background realizations. This

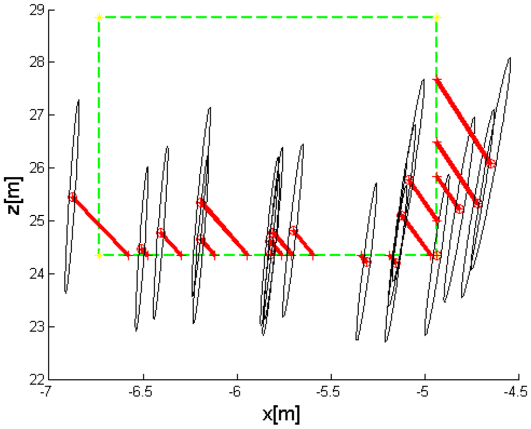


Fig. 4. Illustration of a typical case of the cuboid hypothesis testing: a cuboid box, plotted in green, is placed at a certain position in space with a predefined cuboid orientation. The positional probability $p\left(\boldsymbol{X}^m \mid \mathcal{O}^m\right)$ arises from the sum of distances of all Stixels from the hypothetical bounding box. These distances are plotted in red. Additionally, the covariance ellipses of the Stixels are shown in black. For the sake of easier viewing, the axes are scaled differently.

representation acts as a counter-hypothesis to the object hypothesis $p\left(\boldsymbol{X}^m \mid \mathcal{L}^m = \texttt{OBJ}\right)$. There might be further, more specific and discriminative features to rate the background hypothesis as discussed in Sec. V.

2) Velocity Feature $p\left(\boldsymbol{v}^m \mid \mathcal{L}^m\right)$: For the static background hypothesis, we also deduce $p\left(\boldsymbol{v}^m \mid \mathcal{L}^m = \texttt{BG}\right)$ from the sum of the Mahalanobis distances of all Stixels' velocities $\boldsymbol{v}^m$ to the static background hypothesis $v_{\text{bg}} = 0$. Again this probability distribution is modeled via a Chi-square distribution. The velocity uncertainty for all Stixels is obtained from the Stixel Kalman filter.

Since the velocity of an moving object v_{obj} is unknown in general, we approximate $v_{\text{obj}} \approx \max\left(v_{\text{ego}}, v_{\min}\right)$, where v_{ego} denotes the velocity of the ego-vehicle and $v_{\min}$ is a threshold for a minimum object velocity. Note that this assumption only affects the absolute value of the velocity and that the velocity components v_x and v_z are obtained by incorporating the yaw angle which is given as part of our object hypothesis $\mathcal{O}^m$, i.e. $v_x = v_{obj} \cdot \sin(\psi_m)$, $v_z = v_{obj} \cdot \cos(\psi_m)$. What is crucial is not this value for v_{obj}, but the statistical spread of the velocities of all Stixels S^m in this interval, since for a moving rigid body all Stixels are supposed to have the same velocity.

Then again $p\left(\boldsymbol{v}^m \mid \texttt{OBJ}\right)$ directly depends on the sum of the Mahalanobis distances between the Stixels' velocities $\boldsymbol{v}^m$ and v_{obj} via a Chi-squares distribution. We both incorporate the measurement uncertainty from the Kalman filter and our uncertainty about v_{obj} into these Mahalanobis distances.

3) Existence Feature $p\left(E^m \mid \mathcal{L}^m\right)$: Existent Stixel measurements are probably the most apparent and essential criterion for a true existing object. Vice versa, missing Stixel measurements reduce the object probability significantly. See Fig. 3 for an example: here the interval I^2 resulting from the object hypothesis $\mathcal{O}^2$ is only partly filled with Stixels. This fact should reduce the probability of this hypothesis. The actual number N^m of Stixels in each interval I^m can be compared with the expected number of Stixels N^{exp}, where N^{exp} is obtained by dividing the width of the interval I^m, $w^m = u^m_{\text{right}} - u^m_{\text{left}}$, through the constant width w of each Stixel. We model the existence probability $p\left(E^m \mid \mathcal{O}^m\right)$ for this interval by a gaussian distribution $\mathcal{N}\left(N^{\text{exp}} - N^m, 0, \sigma\right)$ if $N^m < N^{\text{exp}}$.

The background existence likelihood $p\left(E^m \mid \texttt{BG}\right)$ is set to a constant value.

D. Finding the Most Probable Configuration

We sort all intervals I^m according to their rightmost interval position u^m_{right}. For all intervals, we set its likelihood to the maximum value between its *object likelihood* $p\left(\mathcal{S}^m \mid \mathcal{O}^m\right)$ and its *background likelihood* $p\left(\mathcal{S}^m \mid \texttt{BG}\right)$. A function $f(m)$ is defined for all the interval indices m to yield the largest index $m' < m$ such that the intervals I^m and $I^{m'}$ do not overlap, cf. Fig. 5. If there is no such interval, m' is set to zero. Let $OPT(m)$ denote the optimal solution yielding the highest possible likelihood for all intervals $1 \ldots m$. Then $OPT(m)$ can be computed recursively as stated in equation (2). This dynamic programming step selects the best object hypotheses which do not overlap in space.

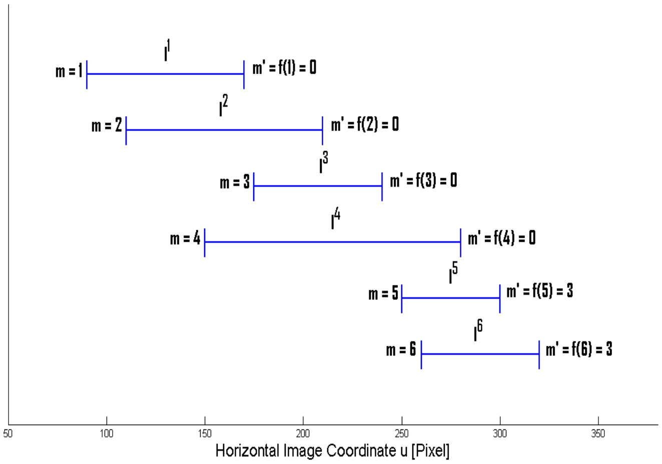


Fig. 5. Schematic illustration of the dynamic programming algorithm. All intervals I^m have the two likelihoods $p\left(\mathcal{S}^m \mid \mathcal{O}^m\right)$ and $p\left(\mathcal{S}^m \mid \texttt{BG}\right)$, respectively. The function $f(m)$ gives the largest interval index $m' < m$, such that the corresponding intervals I^m and $I^{m'}$ do not overlap.

E. Mean Shift Algorithm

In order to evaluate our results, we compare our segmentation results with a Mean Shift approach as introduced in [7]. Briefly, this algorithm randomly picks J Stixels from the initial Stixel set $\mathcal{S}$ containing N Stixels, typically $J \ll$ N, and it computes the mode for this small set of Stixels in a 4-dimensional data space, given by $\{v_x, v_z, x, z\}$, using the Mean-Shift procedure. Then all Stixels within a predefined radius r are assigned to this mode. All the assigned Stixels are separated from the initial Stixel data set and the algorithm starts again, excluding the already assigned Stixels. For details, please see [7].

$$OPT(m) = \begin{cases} 0 & \text{if } m = 0 \\ \max\left(p\left(\mathcal{S}^m \mid \mathcal{L}^m\right) + OPT(f(m)),\, OPT(m-1)\right) & \text{else.} \end{cases} \tag{2}$$

We vary the input parameters J and r, however as long as the radius r is set to a reasonable value for typical road vehicle dimensions, i.e. $1m < r < 5m$, in most cases the clustering results are quite insensitive to these parameters. This clustering into coherent regions serves as a starting point for further object detection, e.g. by thresholding the size of these resulting clusters.

IV. RESULTS

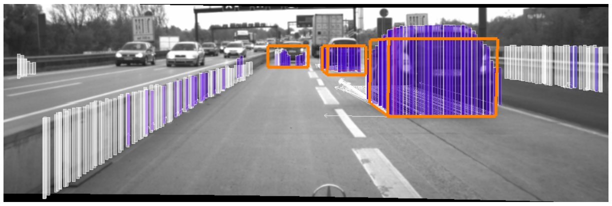
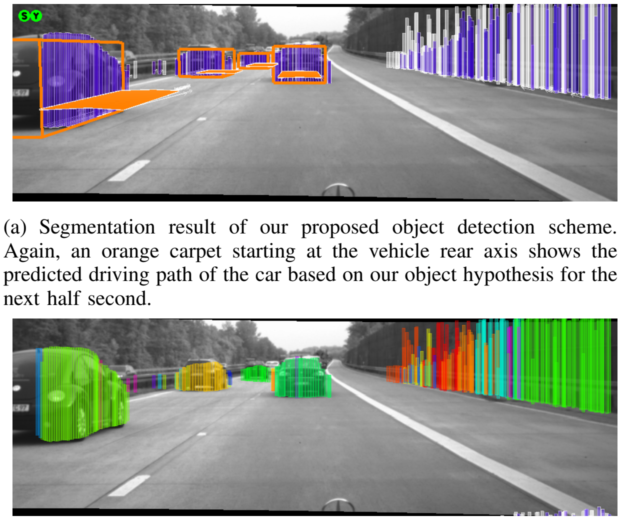


Fig. 6. Segmentation result for this highway scenario with missing Stixel measurements for the left car up-front.



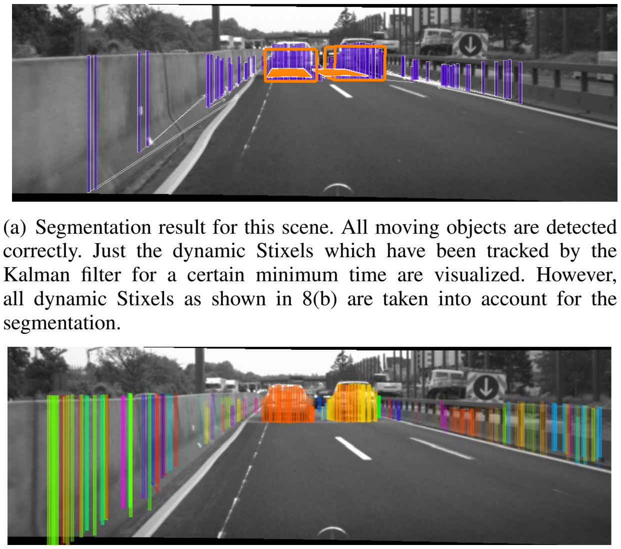
(a) Segmentation result of our proposed object detection scheme. Again, an orange carpet starting at the vehicle rear axis shows the predicted driving path of the car based on our object hypothesis for the next half second.

(b) Mean Shift segmentation result for this scene.

Fig. 7. Highway scenario with four moving vehicles.

The proposed object detection scheme has been tested on various challenging downtown and highway scenarios. The detection results of some example sequences are superimposed in Fig. 1, Fig. 6, Fig. 7, and Fig. 8. In these scenarios, we gradually increase the complexity to see where the two approaches differ.

Fig. 1 presents an interesting downtown scenario with two moving cars and some stationary or parking cars. Our approach in Fig. 1(d) detects both moving objects correctly. The car standing still in the middle and the cars on the side-strip are correctly segmented as background. The Mean Shift approach is able to separate these stationary objects, the segmentation is quite precise since there is little noise in this scene. Just the far distant oncoming car is split into parts because the motion information is noisy here.



(a) Segmentation result for this scene. All moving objects are detected correctly. Just the dynamic Stixels which have been tracked by the Kalman filter for a certain minimum time are visualized. However, all dynamic Stixels as shown in 8(b) are taken into account for the segmentation.

(b) Mean Shift segmentation result for this scene. Different clusters are colored differently. The mean cluster velocity is encoded by the saturation of the Stixels, full saturation corresponds to a relative velocities of $50\,m/s$.

Fig. 8. Comparison between our proposed algorithm and the Mean Shift algorithm.

Missing Stixels due to measurement errors pose another difficulty for object detection, since we seek to perceive an object as a whole. See Fig. 6 for an example. For the car up-front on the left side, some Stixel measurements in the middle are missing. However, again the algorithm has to choose the most probable object configuration among the possible interpretations. As can be seen, our approach can also cope with missing Stixel measurements if this interpretation is still the most likely.

Fig. 7 shows another highway scenario with four moving cars. The ego vehicle is driving with high speed. In this scene the involved cars are clearly separated from each other. The first car on the left side is only partly visible due to the restricted visual field of the stereo camera system. Note that our algorithm extrapolates this object outside the visible field of view using prior hypothesis knowledge about typical object dimensions. The other vehicles are segmented precisely and our algorithm is not distracted by the erroneous motion information on the right crash barrier since the motion information here is too noisy for a possible object.

The Mean Shift approach clusters this scene into 26 objects. There are several falsely segmented Stixels at the objects due to motion noise. The key problem in this scenario is the fact that the Mean Shift clustering detects several moving objects at the stationary side rails which can be traced back to the wrong motion information here. Since we are not modeling the background hypothesis explicitly, the Mean Shift is prone to fragment the background because we are searching for rather small vehicle objects. However, as long as these clusters are reported to be moving and they are not too small, they might be considered as potential moving objects.

Finally Fig. 8 shows a more complex highway scenario with two cars passing very close. Again, the motion information of the dynamic Stixels is noisy. Note that just the dynamic Stixels which have been tracked by the Kalman filter for a certain minimum time are visualized in Fig. 8(a). However, all dynamic Stixels as shown in Fig. 8(b) are incorporated in the segmentation process. Since the two vehicles are quite close to each other, their Stixels are not separated clearly any longer. Similarly, due to noise and erroneous depth measurements, the right car is not clearly separated from the right side rail. Although many Stixels on the side rails are reported to be moving and their motion vectors point into the same direction, our algorithm does not detect any moving objects here because these crash barrier Stixels are too sparse. On the other hand, the Mean Shift approach completely fails to describe this situation. The scene is clustered into 59 groups and the big majority of these clusters is detected to be moving. In Fig. 8(b) we have encoded the velocities of the resulting Mean Shift cluster centers by their saturation. Full saturation corresponds to an absolute velocity value of $50\,m/s$. The cars are partially merged in the Mean Shift segmentation. The spherical metric described by a radius r turns out to be insufficient in such crowded scenarios. Using a smaller radius yields split objects (results are not shown). Besides that, the fact that the Mean Shift segmentation needs not be an optimal solution causes this clearly inferior segmentation. These problems typically arise in more crowded scenarios which cannot be clustered trivially. For the Mean-Shift clustering, a complex postprocessing step would be necessary to interpret the resulting clustering results in order to extract true moving objects whereas our algorithm can do this in one step.

V. CONCLUSIONS AND FUTURE WORKS

A probabilistic framework for detecting moving objects using dynamic programming with geometric constraints has been presented. Dynamic Stixels have been proven to be a powerful and efficient representation of dynamic traffic scenes in a smart one-dimensional data structure. The proposed object detection approach changes the Stixel labeling decision between a moving object and the stationary background into a hypotheses testing problem. Doing so, one is able to incorporate local features as the velocity variance and use global object knowledge, geometric constraints plus global scene modeling to find the most probable interpretation of traffic scenarios. These steps greatly increase the stability with respect to outliers and eliminate unphysical small objects. The experimental results have shown the applicability of the algorithm for detecting moving objects in real traffic scenarios without any manual parameter tuning, even at large distances in the presence of noise and of Stixel modeling errors. We were especially interested in difficult, real world data sets which cannot be clustered trivially but need strong regularization and prior knowledge.

Further steps might extend the number of features. For example the Stixel height might be introduced as an object criterion and appearance-based features like color information might be learned online for moving objects to stabilize detection results. Besides that, one might revisit some problems arising at extreme dense traffic scenes such as large occlusions which have been excluded in most instances in this contribution. In addition, temporal integration and propagation of segmentation results allow to further specify and constrain the unknown moving object class.

REFERENCES

[1] C. Wojek, S. Roth, K. Schindler, B. Schiele, *Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes*, European Conference on Computer Vision, ECCV, 2010.

[2] D. M. Gavrila, S. Munder, *Multi-cue pedestrian detection and tracking from a moving vehicle*, International Journal of Computer Vision **73**, 2007.

[3] D. Pfeiffer and U. Franke, *Efficient Representation of Traffic Scenes by Means of Dynamic Stixels*, Intelligent Vehicles Symposium, IEEE, 2010.

[4] D. Hothersall, *History of Psychology*, McGraw-Hill, 2004.

[5] J. A. Hartigan, *A K-means clustering algorithm*, Journal of the Royal Statistical Society, 1979.

[6] Y. Cheng, *Mean Shift, Mode Seeking, and Clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. **17**, 1995.

[7] D. Comaniciu, P. Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. **24**, 2002.

[8] S. Z. Li, *Markov random field models in computer vision*, European Conference on Computer Vision, ECCV, 1994.

[9] C. Wojek, B. Schiele, *A Dynamic Conditional Random Field Model for Joint Labeling of Object and Scene Classes* , European Conference on Computer Vision, ECCV, 2008.

[10] P. Kohli, L. Ladicky, P. H. S. Torr, *Robust Higher Order Potentials for Enforcing Label Consistency*, International Journal of Computer Vision, 2009.

[11] H. Hirschmueller, S. Gehrig, *Stereo Matching in the Presence of Sub-Pixel Calibration Errors*, IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009.

[12] H. Hirschmueller, *Accurate and efficient stereo processing by semiglobal matching and mutual information*, IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2005.

[13] S. Gehrig, F. Eberli, and T. Meyer, *A real-time low-power stereo vision engine using semi-global matching*, International Conference on Computer Vision Systems, 2009.

[14] H. Badino, U. Franke, D. Pfeiffer, *The stixel world-a compact medium level representation of the 3d-world*, Pattern Recognition: 31st DAGM Symposium, Jena, Germany, 2009.

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

[16] H. Badino, *A robust approach for ego-motion estimation using a mobile stereo platform*, 1st International Workshop on Complex Motion, IWCM04, (Guenzburg, Germany), 2004.

[17] A. Barth, D. Pfeiffer, U. Franke, *Vehicle Tracking at Urban Intersections Using Dense Stereo*, 3rd Workshop on Behaviour Monitoring and Interpretation: Studying Moving Objects in a Three-Dimensional World, 2009.

[18] A. Barth, U. Franke, *Estimating the Driving State of Oncoming Vehicles From a Moving Platform Using Stereo Vision*, IEEE Transactions on Intelligent Transportation Systems, 2009.

[19] D. Pfeiffer, S. Morales, A. Barth, U. Franke, *Ground Truth Evaluation of the Stixel Representation Using Laser Scanners*, IEEE Intelligent Transportation System Conference 2010, Madeira, Portugal.