# 3D Traffic Scene Understanding
# from Movable Platforms

Andreas Geiger      Martin Lauer      Christian Wojek      Christoph Stiller      Raquel Urtasun

**Abstract**—In this paper, we present a novel probabilistic generative model for multi-object traffic scene understanding from movable platforms which reasons jointly about the 3D scene layout as well as the location and orientation of objects in the scene. In particular, the scene topology, geometry and traffic activities are inferred from short video sequences. Inspired by the impressive driving capabilities of humans, our model does not rely on GPS, lidar or map knowledge. Instead, it takes advantage of a diverse set of visual cues in the form of vehicle tracklets, vanishing points, semantic scene labels, scene flow and occupancy grids. For each of these cues we propose likelihood functions that are integrated into a probabilistic generative model. We learn all model parameters from training data using contrastive divergence. Experiments conducted on videos of 113 representative intersections show that our approach successfully infers the correct layout in a variety of very challenging scenarios. To evaluate the importance of each feature cue, experiments using different feature combinations are conducted. Furthermore, we show how by employing context derived from the proposed method we are able to improve over the state-of-the-art in terms of object detection and object orientation estimation in challenging and cluttered urban environments.

**Index Terms**—3D Scene Understanding, Autonomous Driving, 3D Scene Layout Estimation

✦

## 1 INTRODUCTION

RECENT scientific and engineering progress in advanced driver assistance systems and autonomous mobile robots make us believe that cars will be able to drive without human intervention in the near future [31]. While autonomous driving on highways has been successfully demonstrated for several decades [21], navigating urban environments remains an unsolved problem. This is mainly due to the complexity of urban traffic situations and the limited capabilities of current onboard sensors and image understanding algorithms. In recent years, the need for solving the onboard scene understanding problem has been bypassed thanks to the use of manually labeled maps and GPS localization systems, e.g., competitors at the DARPA Urban Challenge [13] or Google's autonomous car. While leading to impressive solutions, this approach suffers from the fact that up-to-date sub-meter accurate maps will most likely be impossible to acquire in practice. Furthermore, GPS signals are not always available, and localization can become imprecise in the presence of skyscrapers, tunnels or jammed signals. We believe that safe operation of an autonomous car can only be guaranteed if the car can interpret its environment reliably from its own sensory

- A. Geiger is with the MPI for Intelligent Systems in Tübingen and with the Institute of Measurement and Control, KIT, Germany.
  E-mail: andreas.geiger@tue.mpg.de
- M. Lauer and C. Stiller are with the Institute of Measurement and Control, Karlsruhe Institute of Technology, Germany.
  E-mail: {lauer,stiller}@kit.edu
- C. Wojek is with Carl Zeiss Corporate Research, Germany.
  E-mail: christian.wojek@zeiss.com
- R. Urtasun is with the Toyota Technological Institute at Chicago, USA.
  E-mail: rurtasun@ttic.edu

input just like a human driver can navigate unknown environments even without maps [12]. Video cameras are particularly appealing sensors in this context as they are cheap and often readily available in modern vehicles. But what makes the interpretation of urban traffic scenarios so challenging compared to highways and rural roads?

First, the traffic situations are very complex. Many different traffic participants may be present and the geometric layout of roads and crossroads is more variable than the geometry of highways. Unfortunately, the availability of unambiguous visual features is limited since road markings and curb stones are often missing or occluded. Difficult illumination conditions such as cast shadows caused by vegetation or infrastructure easily confuse image processing algorithms. Furthermore, the limited aperture angle of onboard cameras, their low mount point and the limited depth perception of stereo make the inference problem very difficult. As a consequence, only objects close to the observer can be located reliably.

In this paper, we present a method that is able to robustly deal with the inherent difficulties in onboard recognition of urban intersections. Our approach reasons jointly about the 3D scene layout of intersections as well as the location and orientation of objects in the scene. We refer the reader to Fig. 1 for an illustration. Towards this goal we exploit a larger number of visual cues than existing approaches, which typically rely on road markings that are often unavailable. In particular, we propose five different visual cues, which describe the static environment encoded in terms of occupancy grids, semantic labels and vanishing points, as well as the dynamic information of vehicles encoded by scene
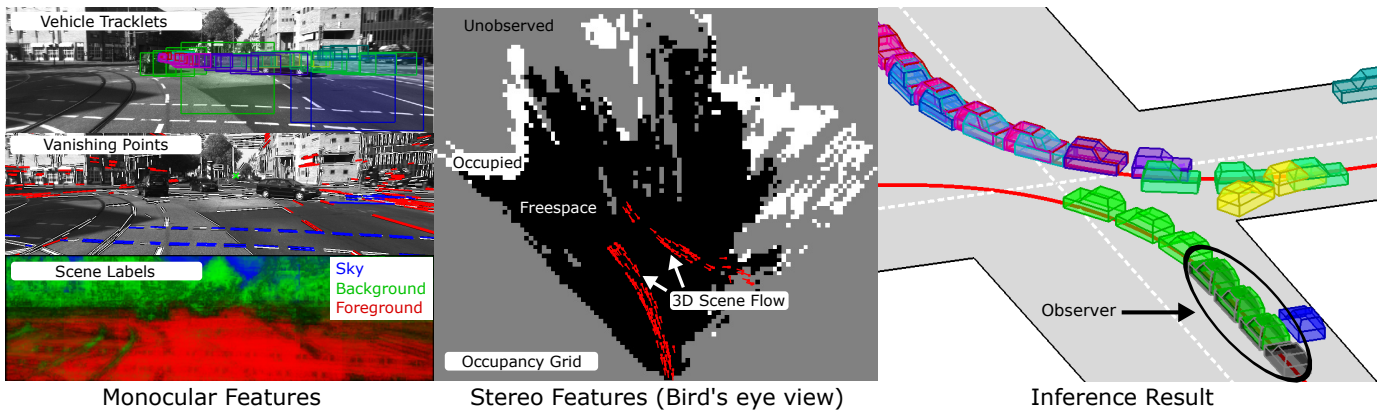
Fig. 1: **3D Intersection Understanding.** Our system makes use of monocular (left) and stereo (middle) feature cues to infer the road layout and the location of traffic participants in the scene (right) from short video sequences. The observer is depicted in black.

flow and vehicle tracklets. As each of these features might be ambiguous and lead to wrong interpretations of the scene, we introduce a probabilistic model that integrates all cues in a principled manner. Our approach is composed of a scalable geometry model for intersections and a proabilistic model that relates the geometry of an intersection to the visual features. We derive learning and inference procedures for this model and demonstrate its performance on video sequences from real-world intersections. Our dataset and a MATLAB/C++ reference implementation is made publicly available.[1]

A preliminary version of this paper appeared in [30], [33]. Herein, we unify both models and perform a greatly enriched analysis on a variety of different feature combinations, yielding novel insights into the importance of the individual feature cues. Moreover, we propose a principled way to learn the parameters of our model. As the partition function of our joint distribution is intractable, we re-formulate the problem in terms of a Gibbs random field and derive all required potentials. We show how the gradients of this function can be approximated by drawing samples from the model distribution, and learning can be simply performed by gradient ascent on the log-likelihood.

This paper is organized as follows. Section 2 gives an overview over related work. In Section 3 we present our probabilistic intersection model as well as our learning and inference procedures. Section 4 gives details on the computation of the visual features. Finally, Section 5 describes the data collection process as well as our experiments. Conclusions are drawn in Section 6.

## 2 RELATED WORK

In 1986, Ernst Dickmanns and his team equipped a van with cameras and demonstrated the first self-driving car on well-marked streets (i.e., highways) without traffic [21]. Motivated by this success, several efforts have been conducted towards autonomous driving in real traffic [10], [29], [51] triggering research on lane estimation [3],

[18], [21], [41], [49] and road detection [1], [2], [17] algorithms. However, simplifying assumptions were made such as low-traffic environments, manual longitudinal control or human intervention for lane changes and highway exits. While the DARPA Urban Challenge [13] came closer to challenging urban traffic situations, the streets were wider than usual, the field of view was unobstructed and only a very limited number of traffic participants were present. Furthermore, GPS in combination with manually annotated maps was neccessary for localization and navigation. Subsequently, Google gathered a team of engineers to equip a Toyota Prius with self-driving capabilities. Similarly to the participants of the Urban Challenge, Google's driver-less car is equipped with a Velodyne 3D laser scanner for perception and requires manually annotated maps at lane-level accuracy for path planning. In contrast, the approach presented in this paper aims at analyzing complex and cluttered scenes in the absence of maps or 3D point clouds.

For a long time intersection understanding has been recognized as a difficult problem [16], [22], [24], [35], [53]. For instance, Luetzeler and Dickmanns [48] extract local image features and match these to a T-shaped intersection model that involves several parameters. They use a multifocal active camera system that allows to detect missing lane boundary segments as an indicator for intersections. Subsequently, the segments are matched with a road map and tracked over time using a Kalman Filter. Zhu et al. [64] use a laser scanner for intersection recognition by searching for positions in front of the vehicle that may serve as a source for long virtual beams that stay collision free in a 2D static grid map. While all existing approaches focus on a small number of simple features such as road markings or 2D grids, in this paper we argue that a bigger picture of the scene that integrates several sources of information in a robust way is able to handle challenging real-world scenarios. In contrast to [64], our method is also able to deal with stereo information, which is much noisier than laser-based measurements.

In computer vision, a large body of work has focused

1. http://www.cvlibs.net/projects/intersection

on estimating 3D from single images [36], [39], [50], [54], [55]. Often, a Manhattan world [7], [43] is assumed to infer vanishing points from line segments. In addition, several methods try to infer the 3D locations of objects in outdoor scenarios [6], [20], [38]. In [14], [23], [60] the camera pose and the location of objects are inferred jointly. Structure-from-motion point clouds are leveraged in [11] to segment and semantically classify the environment. Recently, Singh and Kosecka [56] have proposed a semantic model for urban scene recognition segmenting buildings, road, sky, cars and trees. They employ a trained classifier for pixelwise scene classification in a panoramic image. Unfortunately, most of the existing 3D layout estimation techniques are mainly qualitative, do not model object dynamics, suffer from clutter and lack the level of accuracy neccessary for real-world applications such as autonomous driving. Existing methods that take objects into account usually model the scene in terms of a simple ground plane and thus are not able to draw conclusions from the complex interplay of the objects with the larger scene layout. In contrast, we propose a method that is able to extract accurate geometric information by reasoning jointly about static and dynamic elements as well as their interplay.

For a long time, dynamic objects have been considered either in isolation [5], [26], [27], [52] or using simple motion [9], [23], [42], [47], [63] or social interaction [15], [62] models. For example, Choi et al. [15] introduce a hierarchy of activities, modeling the behavior of groups. Methods for unsupervised activity recognition and abnormality detection [44], [59] are able to recover spatio-temporal dependencies using video sequences from a static camera mounted on top of a building. While promising results have been shown, the interplay of objects with their environment is neglected and the focus is put on surveillance scenarios with a fixed camera viewpoint, limiting their applicability. In contrast, the method developed in this paper infers semantics at a higher level such as multi-object traffic patterns at intersections, in order to improve layout and object estimation. Importantly, we do inference at scenes that we have never seen before and our viewpoint is substantially lower compared to the surveillance scenario, which renders the problem very challenging.

# 3 URBAN SCENE UNDERSTANDING

In this paper, we tackle the problem of understanding complex 3D traffic scenes from short video sequences. In particular, we are interested in inferring the scene layout (e.g., number and location of streets) as well as the 3D location and orientation of traffic participants (e.g., cars) from short video sequences captured onboard a driving vehicle. We assume a flat road surface and model the scene layout and all objects in the road coordinate system, which is located directly below the left camera in the last frame of the video using the yaw angle and coordinate axis illustrated in Fig. 2 (right) and Fig. 7.
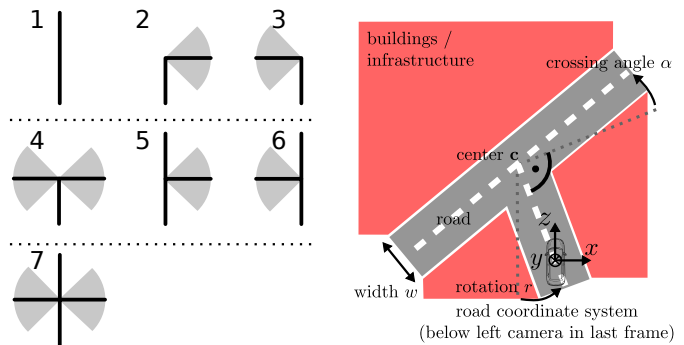


Fig. 2: **Topology Model** (left) and **Geometry Model** (right) in bird's eye perspective. The gray shaded areas (left) illustrate the flexibility of the crossing street ($\alpha$).

The number and location of streets and vehicles are related to the image observations through our probabilistic model. In this section, we introduce all relevant notation, develop the geometric and probabilistic models and describe our inference and parameter learning procedure.

## 3.1 Geometric Model

Our geometric model is inspired by typical traffic scenes. We assume that the layout of the scene is dominated by up to four roads intersecting at a single location, the intersection center. All vehicles are either parked at the side of the road or drive on lanes and respect some basic rules such as right-handed traffic. Lanes are modeled using B-splines. All inbound and outbound streets are connected with a lane, except for U-turns. Road boundaries determine the border between drivable regions and areas that are likely to contain buildings and infrastructure.

Let $\mathcal{R} = \{\kappa, \mathbf{c}, w, r, \alpha\}$ be the set of random variables describing the road layout, where $\kappa \in \{1, \ldots, 7\}$ denotes the **topology** of the intersection (see Fig. 2) and $\mathbf{c} = (x, z)^\mathsf{T} \in \mathbb{R}^2$ is the intersection **center**. Let $w \in \mathbb{R}^+$ be the street width and $r \in [-\frac{\pi}{4}, +\frac{\pi}{4}]$ the global **rotation**, i.e., the observer's yaw orientation with respect to the incoming street. We define $\alpha \in [-\frac{\pi}{4}, +\frac{\pi}{4}]$ as the **crossing angle**, i.e., the relative orientation of the crossing street with respect to the incoming street. For simplicity, we assume that all opposing intersection arms are collinear and all streets share the same width. All variables are illustrated in Fig. 2. Note that our model is able to express straight roads, turns, three-armed and four-armed intersections.

For simplicity, we restrict our focus to two lanes per street, one incoming and one outgoing lane for each intersection arm. As vehicles are allowed to cross the intersection in any possible direction, we have $K(K-1)$ lanes for a $K$-armed intersection. For each street we model two parking areas, one at the left side and one at the right side, yielding $2K$ parking areas in total. Two (out of six) lanes of a 3-armed intersection as well as one parking area are illustrated in Fig. 3 (left).

Lane centerlines are modeled using quadratic B-splines [19] governed by five control points $\{\mathbf{q}_1, \ldots, \mathbf{q}_5\}$,
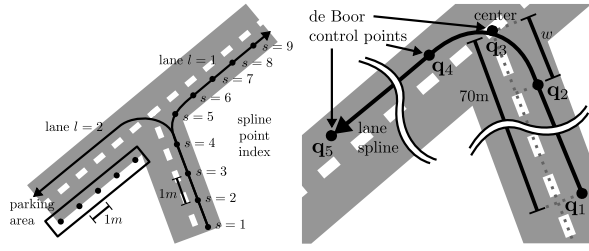
Fig. 3: **Lane/Parking Areas** (left) and **Lane B-splines** (right). Lane centerlines are defined via B-splines and parking areas are located at each road side. For tractability of our tracklet likelihood, all lanes/parking areas are discretized at 1 meter intervals (see Section 3.3.2).



Fig. 4: **Probabilistic Graphical Model** corresponding to the joint distribution over image evidence $\mathcal{E}$ and road layout $\mathcal{R}$ in Eq. (1).

which are located at the center of the lane as illustrated in Fig. 3 (right). The knot vector controlling the shape of the B-splines was chosen to be $(0\ 0\ 0\ 0.1\ 0.9\ 1\ 1\ 1)^\mathsf{T}$, forcing the spline to interpolate all but the central control point. Empirically this resulted in realistic curvatures. Given all lane splines and all parking areas, we equidistantly define discrete vehicle locations $s$ at 1m intervals as illustrated in Fig. 3 (left) to allow for efficient inference of the scene layout and the vehicle locations.

### 3.2 Image Evidence

Besides the geometric model, we define a probabilistic model to explain the image evidence $\mathcal{E} = \{\mathcal{T}, \mathcal{V}, \mathcal{S}, \mathcal{F}, \mathcal{O}\}$ with vehicle tracklets $\mathcal{T}$, vanishing points $\mathcal{V}$, semantic scene labels $\mathcal{S}$, scene flow $\mathcal{F}$ and occupancy grid $\mathcal{O}$. All features are mapped into the last frame of each sequence using visual odometry.

Let $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}\}$ be the set of **vehicle tracklets** that have been detected in the sequence. A vehicle tracklet $\mathbf{t}$ is defined as a sequence of object detections projected into bird's eye perspective $\mathbf{t} = \{\mathbf{d}_1, \ldots, \mathbf{d}_{M_d}\}$. Here, $\mathbf{d} = (f_d, \mathbf{m}_d, \mathbf{S}_d, \mathbf{o}_d)$ denotes a single object detection with $f_d \in \mathbb{N}$ the frame number, $\mathbf{m}_d \in \mathbb{R}^2, \mathbf{S}_d \in \mathbb{R}^{2\times2}$ the mean and covariance describing the object location in road coordinates and $\mathbf{o}_d \in \mathbb{R}^8$ the probability of the object facing into each of eight possible directions.

Let $\mathcal{V} = \{v_1, \ldots, v_{N_v}\}$ be the set of **vanishing points**. We detect up to two vanishing points ($N_v \in \{0, 1, 2\}$) and represent each vanishing point by a single rotation angle around the yaw axis of the road coordinate system, i.e. $v_i \in [0, \pi)$. The vertical vanishing point is non-informative for our task and not considered here. Let $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_s}\}$ be the set of **semantic labels**. Here, $N_s$ is the number of image patches of size $n_s \times n_s$ pixels on a regular grid and $\mathbf{s}_i \in \Delta^2$ is a discrete probability distribution over the semantic categories *road*, *background* and *sky*. Both, vanishing points and scene flow, are computed in the last frame of each sequence as this yielded a good compromise between the quality of the results and computation time. Further, denote $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_{N_f}\}$ the set of **scene flow** vectors capturing the 3D motion of the scene not explained by the observer's egomotion. Each flow vector $\mathbf{f} = (\mathbf{p}_f, \mathbf{q}_f)$ is defined by its location $\mathbf{p}_f \in \mathbb{R}^2$ and velocity $\mathbf{q}_f \in \mathbb{R}^2$ on the road plane. All velocity vectors are normalized to $\|\mathbf{q}_f\|_2 = 1$ as our scene flow
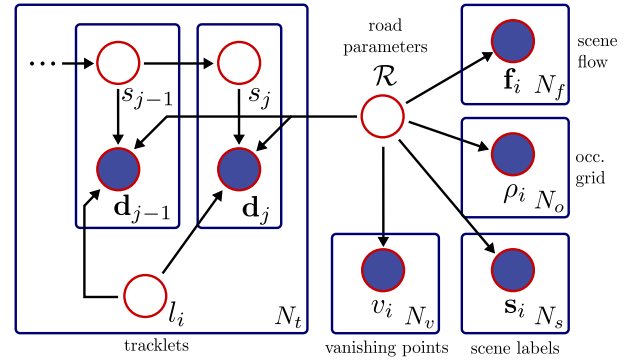
model does not explicitly reason about vehicle velocities. The **occupancy grid** $\mathcal{O} = \{\rho_1, \ldots, \rho_{N_o}\}$ is represented by $N_o$ cells of size $n_o \times n_o$ meters. Each cell $\rho_i \in \{-1, 0, +1\}$ can be either *free* $(-1)$, *occupied* $(+1)$ or *unobserved* $(0)$. We postpone the discussion on feature extraction to Section 4 and describe our probabilistic model in this section.

### 3.3 Probabilistic Model

We assume that all observations $\mathcal{E} = \{\mathcal{T}, \mathcal{V}, \mathcal{S}, \mathcal{F}, \mathcal{O}\}$ are conditionally independent given the road layout $\mathcal{R}$. As a consequence, the joint distribution over the image evidence $\mathcal{E}$ and the road parameters $\mathcal{R}$ factorizes as

$$p(\mathcal{E}, \mathcal{R}|\Theta) = p(\mathcal{R}|\Theta) \underbrace{\prod_{i=1}^{N_t} p(\mathbf{t}_i|\mathcal{R}, \Theta)}_{\text{Vehicle Tracklets}} \underbrace{\prod_{i=1}^{N_v} p(v_i|\mathcal{R}, \Theta)}_{\text{Vanishing Points}}$$

$$\times \underbrace{\prod_{i=1}^{N_s} p(\mathbf{s}_i|\mathcal{R}, \Theta)}_{\text{Scene Labels}} \underbrace{\prod_{i=1}^{N_f} p(\mathbf{f}_i|\mathcal{R}, \Theta)}_{\text{Scene Flow}} \underbrace{\prod_{i=1}^{N_o} p(\rho_i|\mathcal{R}, \Theta)}_{\text{Occupancy Grid}} \quad (1)$$

where the Prior term $p(\mathcal{R}|\Theta)$ is underbraced as "Prior".

where $\Theta$ denotes the set of all parameters in our model. This is illustrated in the graphical model of Fig. 4.

#### 3.3.1 Prior

We define the prior on road parameters $\mathcal{R}$ as

$$p(\mathcal{R}|\Theta) = p(\kappa|\Theta)\, p(\mathbf{c}, r, w|\kappa, \Theta)\, p(\alpha|\kappa, \Theta) \quad (2)$$

with

$$\kappa \sim \mathcal{C}at(\boldsymbol{\xi}_p) \quad (3)$$

$$(\mathbf{c}, r, \log w)^\mathsf{T}|\kappa \sim \mathcal{N}\left(\boldsymbol{\mu}_p^{(\kappa)}, \boldsymbol{\Lambda}_p^{(\kappa)^{-1}}\right) \quad (4)$$

$$\alpha|\kappa \sim f_\kappa(\alpha, \sigma_\alpha)^{\lambda_p} \quad (5)$$

where $\mathcal{C}at(\cdot)$ denotes the categorical distribution and $\mathbf{c}, r$ and $w$ are modeled jointly to capture correlations between the variables. The width $w$ is modeled using a log-Normal distribution to enforce positivity. Empirically we found $\alpha$ to be highly multi-modal. We thus model $\alpha$ using kernel density estimation with a Gaussian kernel and bandwidth $\sigma_\alpha$. The scalar $\lambda_p$ controls the importance of the crossing street prior. All parameters are learned from training data as detailed in Section 3.5.

### 3.3.2 Vehicle Tracklets

Assuming a uniform prior on all lanes and parking areas we model vehicle tracklets as

$$p(\mathbf{t}|\mathcal{R}, \Theta) = \sum_{l=1}^{L} p(\mathbf{t}, l|\mathcal{R}) \tag{6}$$

$$p(\mathbf{t}, l|\mathcal{R}) = p(l|\mathcal{R}) \, p(\mathbf{t}|l, \mathcal{R}) \propto p(\mathbf{t}|l, \mathcal{R}) \tag{7}$$

where the tracklet index $i$ and the dependency on the parameters $\Theta$ have been dropped for clarity and the latent lane (or parking spot) index $l$ has been marginalized. To evaluate the tracklet posterior for lanes $p_l(\mathbf{t}|l, \mathcal{R})$, we discretize the lane spline at 1 meter intervals and augment the observation model with an additional discrete latent variable $s$ per object detection $\mathbf{d}$, which indexes the location on the lane, as illustrated in Fig. 3. We employ a simple left-to-right Hidden Markov Model to model the dynamics. Marginalizing over all hidden states $s_1, \ldots, s_{M_d}$ yields

$$p_l(\mathbf{t}|l, \mathcal{R}) = \sum_{s_1, \ldots, s_{M_d}} p(s_1) \, p_l(\mathbf{d}_1|s_1, l, \mathcal{R})$$
$$\times \prod_{j=2}^{M_d} p(s_j|s_{j-1}) \, p_l(\mathbf{d}_j|s_j, l, \mathcal{R}) \tag{8}$$

where $M_d$ denotes the number of object detections in the tracklet. We allow tracklets to start anywhere on the lane with equal probability. Our motion model is simple, yet effective. By constraining all tracklets to move forward with uniform probability, i.e.,

$$p(s_j|s_{j-1}) = \begin{cases} \frac{1}{M_l - s_{j-1} + 1} & \text{if } s_j \geq s_{j-1} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

lanes on crossing streets can be distinguished purely based on the vehicle's motion. Here, $M_l$ is the number of spline points on lane $l$. The emission probability for lanes $p_l(\mathbf{d}|s, l, \mathcal{R})$ is factorized into the object's location and the object's orientation

$$p_l(\mathbf{d}|s, l, \mathcal{R}) = p(\mathbf{m}_d|s, l, \mathcal{R}, \mathbf{S}_d) \, p(\mathbf{o}_d|s, l, \mathcal{R}) \tag{10}$$

The object location is modeled as a Gaussian mixture

$$p(\mathbf{m}_d|s, l, \mathcal{R}, \mathbf{S}_d) = (1 - \zeta_t) \, p_{in}(\mathbf{m}_d|s, l, \mathcal{R}, \mathbf{S}_d)$$
$$+ \zeta_t \, p_{out}(\mathbf{m}_d|s, l, \mathcal{R}) \tag{11}$$

with inlier and outlier distributions defined by

$$p_{in}(\mathbf{m}_d|s, l, \mathcal{R}, \mathbf{S}_d) \propto \exp\left(-\frac{(\boldsymbol{\phi}_t - \mathbf{m}_d)^\mathsf{T} \mathbf{S}_d^{-1}(\boldsymbol{\phi}_t - \mathbf{m}_d)}{2}\right)$$

$$p_{out}(\mathbf{m}_d|s, l, \mathcal{R}) \propto \exp\left(-\frac{\mathbf{m}_d^\mathsf{T} \mathbf{m}_d}{2\sigma_{out}^2}\right) \tag{12}$$

respectively. Here, $\boldsymbol{\phi}_t(s, l, \mathcal{R}) \in \mathbb{R}^2$ denotes the 2D location of spline point $s$ on lane $l$ according to the B-spline model presented in Section 3.1, $\zeta_t$ is the outlier probability and $\sigma_{out}$ is a parameter controlling the 'spread' of the outlier distribution.
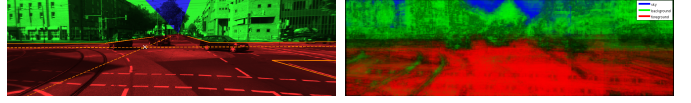


Fig. 5: **Semantic Scene Label Likelihood.** Intuitively, the semantic likelihood measures the 'overlap' of the rendered model hypothesis (left) with the semantic labels estimated by a classifier (right).

For the object orientation likelihood, we impose a categorical distribution over 8 object orientations $\mathbf{o}_d$

$$p(\mathbf{o}_d|s, l, \mathcal{R}) = \prod_{i=1}^{8} o_{d,i}^{[\varphi_t(s, l, \mathcal{R})=i]} \tag{13}$$

where $\varphi_t(s, l, \mathcal{R}) \in \{1, \ldots, 8\}$ selects the orientation bin corresponding to the viewpoint relative to the observer: The relative viewing direction is computed from the tangent of lane $l$ at spline point $s$. Intuitively, Eq. (13) encourages lane associations such that the estimated vehicle orientation and the direction of the lane coincide.

As parking cars are static, the tracklet likelihood for parking areas reduces to

$$p_p(\mathbf{t}|l, \mathcal{R}) = \sum_s \prod_{j=1}^{M_d} p(s) \, p_p(\mathbf{d}_j|s, l, \mathcal{R}) \tag{14}$$

assuming a uniform prior on the location $s$. Note that we do not make any assumptions about the orientation of a parked car. Thus the emission probability becomes

$$p_p(\mathbf{d}|s, l, \mathcal{R}) = \frac{1}{8} \, p(\boldsymbol{\mu}_d|s, l, \mathcal{R}, \mathbf{S}_d) \tag{15}$$

with $p(\boldsymbol{\mu}_d|s, l, \mathcal{R}, \mathbf{S}_d)$ defined in Eq. (11).

### 3.3.3 Vanishing Points

We model the vanishing likelihood over $v \in [0, \pi)$ as

$$p(v|\mathcal{R}, \Theta) \propto \zeta_v + (1 - \zeta_v) \exp\left(-\lambda_v \phi_v(v, \mathcal{R}, \Theta)\right) \tag{16}$$

with orientation error

$$\phi_v(v, \mathcal{R}, \Theta) = 1 - \cos(2v - 2\varphi_v(\mathcal{R})) \tag{17}$$

derived from the von Mises distribution, which is a continuous probability distribution on the circle. Here, $\zeta_v$ is a small constant capturing outlier detections, $\varphi_v(\mathcal{R})$ is the orientation of the closest street, based on the current road model configuration $\mathcal{R}$, and $\lambda_v$ is a precision parameter that controls the importance of this term.

### 3.3.4 Semantic Scene Labels

The semantic scene label likelihood [33] for each $n_s \times n_s$ pixels image patch $\mathbf{s}$ is modeled as

$$p(\mathbf{s}|\mathcal{R}, \Theta) \propto \exp\left(\frac{\lambda_s}{N_s} \, w_{s, \phi_s(\mathcal{R})} \cdot s_{\phi_s(\mathcal{R})}\right) \tag{18}$$

where $N_s$ is the total number of image patches (see Section 4.3) and $\lambda_s$ is a parameter controlling the importance of this cue. $\phi_s(\mathcal{R}) \in \{1, 2, 3\}$ selects the class label of $\mathbf{s}$ according to a 'virtual' segmentation of the scene which depends on the reprojection of the road layout
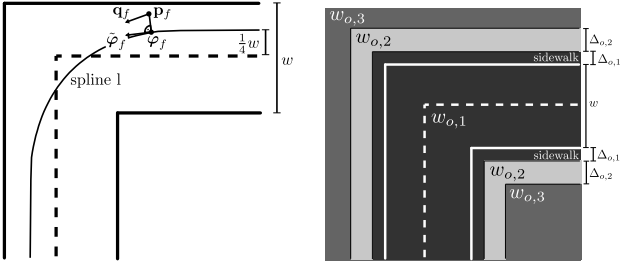
Fig. 6: **Scene Flow** (left) and **Occupancy Grid Likelihood** (right). The proposed scene flow likelihood encourages flow vectors to agree with the lane geometry. The geometric occupancy prior (right) can be envisioned as a 'template' of freespace and occupied areas.

$\mathcal{R}$ such that $s_{\phi_s(\mathcal{R})}$ returns the probability of the class label given a semantic classifier. This is illustrated in Fig. 5. Additionally, the weight vector $\mathbf{w}_s \in \mathbb{R}^3$ controls the importance of each semantic class. Our model uses three classes: foreground (e.g., road), background (e.g., buildings, vegetation) and sky. To simplify matters, we assume that the background (i.e., buildings, trees) starts directly behind the curb of the road and buildings reach a height of four stories on average, thereby defining the background area which separates the sky from the road region. Facades adjacent to the observer's own street are not considered. Despite these approximations, we have observed that many inner-city scenes in our dataset follow this scheme closely.

### 3.3.5 Scene Flow

In contrast to the tracklet observations, the 3D scene flow likelihood directly explains all moving objects in the scene given the road layout $\mathcal{R}$. Thus, moving objects that do not fit the appearance model of the car detector (e.g., trucks, tractors, quad bikes, motorbikes) are considered here as well. The probability of a scene flow vector depends on its proximity to the closest lane and on how well its velocity vector aligns with the tangent of the respective B-spline at the corresponding foot point

$$p(\mathbf{f}|\mathcal{R},\Theta) \propto \phi_f(\mathbf{f},\mathcal{R},\Theta)^{\frac{1}{N_f}} \qquad (19)$$

where

$$\phi_f(\cdot) = \zeta_f \exp\left(-\frac{\|\mathbf{p}_f\|_2^2}{2\sigma_{out}^2}\right) + (1-\zeta_f)\exp\left(-\tilde{\phi}_f(\mathbf{f},\mathcal{R},\Theta)\right)$$

$$\tilde{\phi}_f(\cdot) = -\lambda_{f1}\|\mathbf{p}_f - \boldsymbol{\varphi}_f(\mathbf{p}_f,\mathcal{R})\|_2^2 - \lambda_{f2}(1 - \mathbf{q}_f^\mathsf{T}\tilde{\boldsymbol{\varphi}}_f(\mathbf{p}_f,\mathcal{R}))$$

with parameters $\zeta_f, \lambda_{f1}, \lambda_{f2}, \sigma_{out}$. Here, $\zeta_f$ accounts for outliers and $\lambda_{f1}$ and $\lambda_{f2}$ control the importance of the location and the orientation agreement. $N_f$ normalizes for the number of scene flow vectors and, similar to the vehicle tracklet model from Section 3.3.2, $\sigma_{out}$ denotes the width of the outlier distribution. The functions $\boldsymbol{\varphi}_f(\mathbf{p}_f,\mathcal{R}) \in \mathbb{R}^2$ and $\tilde{\boldsymbol{\varphi}}_f(\mathbf{p}_f,\mathcal{R}) \in \mathbb{R}^2$ return the spline foot point and tangent vector at the location closest to $\mathbf{p}_f$, respectively. This is illustrated in Fig. 6. The dependencies are modeled as a hard mixture, i.e. for each flow vector we select the spline $l$ that maximizes Eq. (19).

### 3.3.6 Occupancy Grid

We assume that the road area should coincide with free space while non-road areas may be covered by buildings or vegetation. Thus, we use stereo information to compute an occupancy grid, which represents occupied and free-space in bird's eye perspective, see Fig. 1 (middle) for an illustration. The occupancy likelihood of each cell $\rho$ in the grid is modeled as follows

$$p(\rho|\mathcal{R},\Theta) \propto \exp\left(\frac{\lambda_o}{N_o}\rho \cdot \phi_o(\rho,\mathcal{R})\right) \qquad (20)$$

where $\phi_o(\rho,\mathcal{R}) \in \{w_{o,1}, w_{o,2}, w_{o,3}\}$ is a mapping that for any cell $\rho \in \{-1, 0, +1\}$ returns the value (or weight) of a model-dependent geometric prior expressing the belief on the location of free space (i.e., road) and buildings alongside the road. The geometric prior is illustrated in Fig. 6 for the case of a right turn. Intuitively, it encourages free space where the road is located and obstacles elsewhere, with a preference towards the roadside region. $\lambda_o$ controls the strength of this term.

## 3.4 Inference

As inference is a key component for learning we start with a discussion on how to obtain samples from the joint distribution using Markov Chain Monte Carlo techniques. Given the image evidence $\mathcal{E}$, we are interested in determining the underlying road layout $\mathcal{R}$ and the location of cars $\mathcal{C} = \{(l, \mathbf{s})\}$ in the scene, where $l$ denotes the lane index and $\mathbf{s}$ contains the spline points of all detections in a tracklet. Unfortunately, the posteriors involved in this computation have no analytical solution and cannot be computed in closed form. Thus we approximate them using Metropolis-Hastings sampling [4]. To keep computations tractable, we split the problem into two sub-problems: We first estimate $\mathcal{R}$ while marginalizing $\mathcal{C}$ as

$$\hat{\mathcal{R}} = \underset{\mathcal{R}}{\mathrm{argmax}}\ p(\mathcal{R}|\mathcal{E},\Theta) = \underset{\mathcal{R}}{\mathrm{argmax}}\ \sum_{\mathcal{C}} p(\mathcal{R},\mathcal{C}|\mathcal{E},\Theta) \quad (21)$$

through Eq. (1), Eq. (8) and Eq. (14). Given an estimate of $\mathcal{R}$, we infer the object locations $\mathcal{C}$ as

$$\hat{\mathcal{C}} = \underset{\mathcal{C}}{\mathrm{argmax}}\ p(\mathcal{C}|\mathcal{E},\mathcal{R},\Theta) \qquad (22)$$

Both steps are detailed in the following.

### 3.4.1 Inferring the Road Layout

For maximizing Eq. (21), we draw $n_{inf}$ samples using Markov Chain Monte Carlo and select the one with the highest probability as our MAP estimate $\hat{\mathcal{R}}$. We exploit a combination of *local*, *inter-topology* and *global* moves to obtain a well-mixing Markov chain. While local moves modify $\mathcal{R}$ only slightly, global moves sample $\mathcal{R}$ directly from the prior. This ensures a quick traversal of the search space, while still exploring local modes. For local moves we choose symmetric proposals in the form of Gaussians centered on the previous state. Table

| Local Metropolis Proposals (33%) |
|---|
| 1. Vary center of crossroads $\mathbf{c}$ ($\sigma_{\mathbf{c}}$) |
| 2. Vary width of all roads $\mathbf{w}$ ($\sigma_{\mathbf{w}}$) |
| 3. Vary angle of crossing street $\alpha$ ($\sigma_{\alpha}$) |
| 4. Vary overall orientation $r$ ($\sigma_r$) |
| 5. Vary center $\mathbf{c}$ and width $\mathbf{w}$ jointly |
| 6. Vary center $\mathbf{c}$, width $\mathbf{w}$, angle $\alpha$ and rotation $r$ jointly |
| **Inter-Topology Metropolis Proposals (33%)** |
| 7. Re-sample $\kappa$ uniformly |
| **Global Metropolis-Hastings Proposals (33%)** |
| 8. Re-sample all parameters $\mathcal{R} = \{\kappa, \mathbf{c}, w, r, \alpha\}$ from the prior |

TABLE 1: **Metropolis-Hastings-Proposals for Inference.** We randomly chose a local, inter-topology or global move with uniform probability.

1 gives an overview of the move categories which are selected at random. Each sample requires the evaluation of $p(\mathcal{R}|\mathcal{E}, \Theta)$ up to a normalizing constant. The marginalization in Eq. (8) can be carried out efficiently using the forward algorithm for hidden Markov models. To avoid trans-dimensional moves, we include $\alpha$ in all models.

### 3.4.2 Inferring the Location of Objects

Given the road model $\mathcal{R}$, we are now interested in recovering the location of cars $\mathcal{C} = \{(l_1, \mathbf{s}_1), \ldots, (l_{N_t}, \mathbf{s}_{N_t})\}$. Conditioned on $\mathcal{R}$, all tracklets become independent and the inference problem decomposes into sub-problems. Neglecting the tracklet index $i$ and the dependency on $\Theta$ for clarity, and observing that $p(\mathbf{t})$ is constant, $\hat{l}$ can be inferred by computing the marginal over object locations

$$\hat{l} = \underset{l}{\arg\max}\, p(l|\mathbf{t}, \mathcal{R}) = \underset{l}{\arg\max}\, p(\mathbf{t}, l|\mathcal{R}) \quad (23)$$

with $p(\mathbf{t}, l|\mathcal{R})$ defined by Eq. (7). Given $l$, we have

$$\hat{s}_1, \ldots, \hat{s}_{M_d} = \underset{s_1, \ldots, s_{M_d}}{\arg\max}\, p_l(\mathbf{t}, s_1, \ldots, s_{M_d}|l, \mathcal{R}) \quad (24)$$

which can be easily inferred using Viterbi decoding for hidden Markov models [8].

### 3.5 Learning

The parameters $\Theta = \{\lambda_p, \boldsymbol{\xi}_p, \lambda_t, \lambda_v, \lambda_s, \lambda_{f1}, \lambda_{f2}, \lambda_o\}$ of our model are learned from held-out training data using maximum likelihood and contrastive divergence [37]. Let $(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{R}})$ be a training set with $\boldsymbol{\mathcal{E}} = \{\mathcal{E}_1, \ldots, \mathcal{E}_D\}$ denoting the image evidence and $\boldsymbol{\mathcal{R}} = \{\mathcal{R}_1, \ldots, \mathcal{R}_D\}$ the annotated road layout of each sequence. The parameter set $\hat{\Theta}$ maximizing the likelihood of the data is given by

$$\hat{\Theta} = \underset{\Theta}{\arg\max}\, p(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{R}}|\Theta) \quad (25)$$

with

$$p(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{R}}|\Theta) = \prod_{d=1}^{D} p(\mathcal{E}_d, \mathcal{R}_d|\Theta) \quad (26)$$

Unfortunately, maximizing Eq. (26) directly for $\Theta$ is intractable due to the partition function. Instead, we rewrite $p(\mathcal{E}_d, \mathcal{R}_d|\Theta)$ in terms of a Gibbs random field

$$p(\mathcal{E}_d, \mathcal{R}_d|\Theta) = \frac{1}{Z_d(\Theta)} \exp\left(-\Psi(\mathcal{E}_d, \mathcal{R}_d, \Theta)\right) \quad (27)$$

where $\Psi(\mathcal{E}_d, \mathcal{R}_d, \Theta)$ is the sum of a set of potential functions $\{\psi_i\}$ corresponding to the probability distributions in Section 3.3 which are described at the end of this section. The log-likelihood function is given by

$$\mathcal{L}(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{R}}|\Theta) = \sum_{d=1}^{D} \log p(\mathcal{E}_d, \mathcal{R}_d|\Theta)$$
$$= -\sum_{d=1}^{D} \left(\Psi(\mathcal{E}_d, \mathcal{R}_d|\Theta) + \log Z_d(\Theta)\right) \quad (28)$$

Taking the partial derivative of $\mathcal{L}(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{R}}|\Theta)$ with respect to a parameter $\theta_i \in \Theta$, we obtain

$$\frac{\partial \mathcal{L}(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{R}}|\Theta)}{\partial \theta_i} = -\sum_{d=1}^{D} \left(\frac{\partial}{\partial \theta_i}\Psi(\mathcal{E}_d, \mathcal{R}_d, \Theta) + \frac{\partial}{\partial \theta_i}\log Z_d(\Theta)\right) \quad (29)$$

While the first term is easy to evaluate, the second term can be rewritten as

$$\frac{\partial}{\partial \theta_i}\log Z_d(\Theta) = \frac{1}{Z_d(\Theta)}\int \frac{\partial}{\partial \theta_i}\exp(-\Psi(\mathcal{E}_d, \mathcal{R}, \Theta))d\mathcal{R}$$
$$= -\left\langle \frac{\partial}{\partial \theta_i}\Psi(\mathcal{E}_d, \mathcal{R}, \Theta)\right\rangle_{p(\mathcal{E}_d, \mathcal{R}|\Theta)} \quad (30)$$

where $\langle\cdot\rangle_{p(\cdot)}$ denotes the expected value with respect to $p(\cdot)$. Note that in contrast to [37] the potentials $\Psi$ additionally depend on $\mathcal{E}_d$ in our case. While it is impossible to evaluate this expression exactly, it can be approximated by drawing samples using MCMC as described in Section 3.4. Sampling exhaustively from the model distribution is computationally prohibitive. We therefore make use of contrastive divergence [37], and instead of running the Markov chain until convergence, we only run it for a small number of $n_{learn}$ iterations per gradient update. By starting the chain at the training data we obtain samples in all places where we want the model distribution to be accurate. Running the MCMC sampler for a couple of iterations already draws the samples closer to the model distribution. For more details and derivations we refer the interested reader to [28], [37].

**Energy Potentials:** The joint potential $\Psi(\mathcal{E}, \mathcal{R}, \Theta)$ in Eq. (27) decomposes as

$$\Psi(\mathcal{E}, \mathcal{R}, \Theta) = \psi_p(\mathcal{R}, \Theta) + \psi_t(\mathcal{T}, \mathcal{R}, \Theta) + \psi_v(\mathcal{V}, \mathcal{R}, \Theta)$$
$$+ \psi_s(\mathcal{S}, \mathcal{R}, \Theta) + \psi_f(\mathcal{F}, \mathcal{R}, \Theta) + \psi_o(\mathcal{O}, \mathcal{R}, \Theta)$$

using the same subscript notation as in Eq. (1). Taking the logarithm of Eq. (2) we obtain

$$\psi_p(\mathcal{R}, \Theta) = -\lambda_p \log f_\kappa(\alpha) - \sum_{i=1}^{7}[\kappa = i]\log \xi_{p,i}$$
$$+ \frac{1}{2}\phi_p(\mathcal{R}, \boldsymbol{\mu}_p^{(\kappa)})^\mathsf{T}\boldsymbol{\Lambda}_p^{(\kappa)}\phi_p(\mathcal{R}, \boldsymbol{\mu}_p^{(\kappa)}) \quad (31)$$

with $\phi_p(\mathcal{R}, \boldsymbol{\mu}_p^{(\kappa)}) = (\mathbf{c}, r, \log w)^\mathsf{T} - \boldsymbol{\mu}_p^{(\kappa)}$. We set $\boldsymbol{\mu}_p \in \mathbb{R}^4$ and $\boldsymbol{\Lambda}_p \in \mathbb{R}^{4\times4}$ to their empiric marginals and learn $\boldsymbol{\xi}_p, \lambda_p \in \Theta$ using the approach described in Section 3.5.

The derivatives are readily derived from Eq. (31). Taking the logarithm of Eq. (6) - (20) we obtain:

$$\psi_t = -\frac{\lambda_t}{N_t} \sum_{i=1}^{N_t} \log \sum_{l=1}^{L} p(\mathbf{t}_i, l|\mathcal{R})$$

$$\psi_v = -\sum_{i=1}^{N_v} \log \left[\zeta_v + (1 - \zeta_v) \exp\left(-\lambda_v \phi_v(v_i, \mathcal{R}, \Theta)\right)\right]$$

$$\psi_s = -\frac{\lambda_s}{N_s} \sum_{i=1}^{N_s} w_{s,\phi_s(\mathcal{R})} \; s_{i,\phi_s(\mathcal{R})}$$

$$\psi_f = -\frac{1}{N_f} \sum_{i=1}^{N_f} \log \phi_f(\mathbf{f}_i, \mathcal{R}, \Theta), \quad \psi_o = -\frac{\lambda_o}{N_o} \sum_{i=1}^{N_o} \rho_i \, \phi_o(\mathcal{R})$$

Here, we have added an additional degree of freedom $\lambda_t$ to the tracklet potential $\psi_t$, which accommodates for violations of the naïve Bayesian observation model and controls the relative strength of the tracklet feature with respect to the prior and all other features.

## 4 IMAGE EVIDENCE

This section describes the feature cues used by our probabilistic model described in Section 3.3. Using motion information from visual odometry, we represent all features in the reference coordinate system which is located on the road surface below the left camera coordinate system in the last frame of each sequence, as illustrated in Fig. 7. While vehicle tracklets, vanishing points and semantic scene labels can be computed from a monocular video stream, the scene flow and occupancy grid features require a stereo setup. The importance of each of these cues is investigated in our experiments.

### 4.1 Vehicle Tracklets

We define a tracklet as a set of object detections, projected into bird's eye perspective $\mathbf{t} = \{\mathbf{d}_1, \ldots, \mathbf{d}_{M_d}\}$ with $\mathbf{d} = (f_d, \mathbf{m}_d, \mathbf{S}_d, \mathbf{o}_d)$. Here, $f_d \in \mathbb{N}$ is the frame number and $\mathbf{m}_d \in \mathbb{R}^2, \mathbf{S}_d \in \mathbb{R}^{2 \times 2}$ are the mean and covariance matrix of the object location. $\mathbf{o}_d \in \mathbb{R}^8$ describes the probability of the vehicle being viewed from any of $8$ possible orientations. Our goal now is to associate object detections to tracklets and project them into 3D using cues such as the object size or the bounding box ground contact point in combination with the height and the pitch angle of the camera. Association of detections to tracklets is performed in image-scale space to better account for uncertainties of the object detector.

For detection, we train the part-based object detector of [25] on a large set of manually annotated images. As our training set comprises ground truth orientation labels, we modify the latent SVM [25] such that the latent component variables are fixed to the orientations. We normalize $\mathbf{o}$ to one using the softmax transformation, applied to the scores of all detections that overlap the non-maximum-suppressed ones.

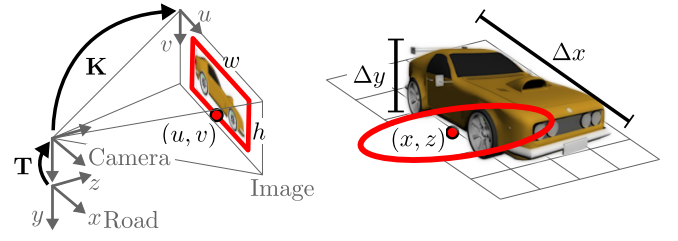Tracking is performed in two simple stages. First, we associate all object detections frame-by-frame using the



Fig. 7: **Projection of 2D Object Detections into 3D.** By learning the statstics of the vehicle dimensions, we are able to relate the 2D object bounding box to the location of the object in 3D.

Hungarian method [45]. The affinity matrix is computed using geometry and appearance cues of the object. As geometry cue we employ the bounding box intersection-over-union score. The appearance cue is computed by correlating the bounding box region in the previous frame with the bounding box region in the current frame, using a small margin (20%) to account for the localization uncertainty of the object detector. In a second stage we associate tracklets to each other. We allow for bridging occlusions of up to 20 frames and make use of the Hungarian algorithm for optimal tracklet association. Towards this goal, we employ a geometry cue [28] that predicts the object location and size from the bounding boxes in the other tracklet. Similar to above, object appearance is compared via normalized cross-correlation.

Given the associated bounding boxes, we estimate the 3D location of the vehicles relative to the camera. Let $(u, v)^\mathsf{T}$ be the image coordinates of the object's bounding box bottom-center and let $w, h$ be its width and height. Let $(x, 0, z)^\mathsf{T}$ be the 3D location of an object in ground plane coordinates ($y = 0$) as illustrated in Fig. 7. Further, let $\Delta x, \Delta y$ be the object width and height in meters, measured via parallel-projection to the $z = 0$ plane. Finally, let $o$ denote the MAP orientation of the vehicle as returned by the object detector. Assuming a uniform prior over $x$ and $z$, the posterior on the object's 3D location can be factorized as $p(x, z|u, v)p(z|w, \Delta x, o)p(z|h, \Delta y)$. Here, the first term relates the bounding box position $(u, v)^\mathsf{T}$ to the 3D location $(x, 0, z)^\mathsf{T}$ and the second and last term model the relationship between the distance $z$ and the width $w$ and height $h$ of the bounding box. As the bounding box width varies with the orientation of a vehicle, $p(z|w, \Delta x, o)$ depends on $o$. We learn a separate set of parameters for each $o$, but will drop this dependency in the following for clarity of presentation. Let $x, z|u, v \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1^{-1})$, $z|w, \Delta x \sim \mathcal{N}(\mu_2, \lambda_2^{-2})$ and $z|h, \Delta y \sim \mathcal{N}(\mu_3, \lambda_3^{-2})$. Then, $x, z|u, v, w, h, \Delta x, \Delta y \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}\boldsymbol{\Lambda}_2 \left[0 \; \mu_2\right]^\mathsf{T} + \boldsymbol{\Sigma}\boldsymbol{\Lambda}_3 \left[0 \; \mu_3\right]^\mathsf{T}$$
$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2 + \boldsymbol{\Lambda}_3)^{-1} \tag{32}$$

where $\boldsymbol{\Lambda}_1$ has full rank and $\boldsymbol{\Lambda}_2, \boldsymbol{\Lambda}_3$ are singular matrices of the form $\boldsymbol{\Lambda}_2 = \text{diag}(0, \lambda_2)$ and $\boldsymbol{\Lambda}_3 = \text{diag}(0, \lambda_3)$. Assuming a standard pinhole camera model we have

$$\left[u \; v \; 1\right]^\mathsf{T} = \mathbf{P}^{3 \times 4} \left[x \; 0 \; z \; 1\right]^\mathsf{T} \tag{33}$$

| | $\Delta x_1$ | $\Delta x_2$ | $\Delta x_3$ | $\Delta y$ |
|---|---|---|---|---|
| $\mu$ | 1.86 | 4.37 | 2.82 | 1.59 |
| $\sigma$ | 0.28 | 0.64 | 0.63 | 0.22 |

Fig. 8: **Object Size Statistics.** $\Delta x$ and $\Delta y$ are the width and height after parallel projection to $z = 0$. Cars in our dataset are $\sim 1.9$m wide and $\sim 4.4$m long.
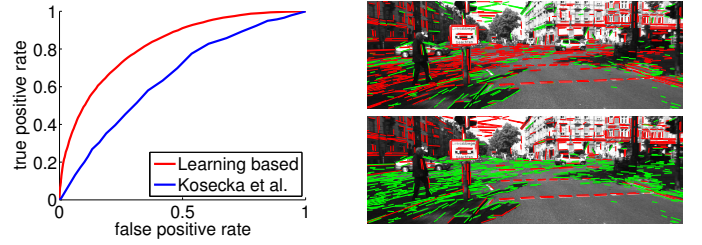


Fig. 9: **Structured Line Segments.** ROC classification results into *structure* and *clutter* (left). Qualitative results from Kosecka et al. [43] (top-right) vs. our method (bottom-right). Red corresponds to *structure*.

where $\mathbf{P} = \mathbf{K}\,\mathbf{T}\,\mathbf{R}$ is the product of a calibration matrix $\mathbf{K}^{3\times3}$, the transformation from ground plane coordinates to camera coordinates $\mathbf{T}^{3\times4}$ and an additional pitch error $\theta$, parameterized by the rotation matrix $\mathbf{R}^{4\times4}(\theta)$. Given $\begin{bmatrix} u & v \end{bmatrix}^\top$ we obtain $\begin{bmatrix} x & z \end{bmatrix}^\top$ by solving the linear system $\mathbf{A}\,\begin{bmatrix} x & z \end{bmatrix}^\top = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} uP_{31} - P_{11} & uP_{33} - P_{13} \\ vP_{31} - P_{21} & vP_{33} - P_{23} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} P_{14} - uP_{34} \\ P_{24} - vP_{34} \end{bmatrix}$$

where we have made use of $\langle\theta\rangle_{p(\theta)} = 0$. The covariance of $\begin{bmatrix} x & z \end{bmatrix}^\top$ can be approximated using error propagation

$$\boldsymbol{\Lambda}_1 = \boldsymbol{\Sigma}_1^{-1}, \quad \boldsymbol{\Sigma}_1 = \mathbf{J}\,\mathrm{diag}(\sigma_u^2, \sigma_v^2, \sigma_\theta^2)\,\mathbf{J}^\top \tag{34}$$

where the Jacobian $\mathbf{J} \in \mathbb{R}^{3\times3}$ is given by

$$\mathbf{J} = \left( \tfrac{\partial}{\partial u}\mathbf{A}^{-1}\mathbf{b} \quad \tfrac{\partial}{\partial v}\mathbf{A}^{-1}\mathbf{b} \quad \tfrac{\partial}{\partial \theta}\mathbf{A}^{-1}\mathbf{b} \right) \tag{35}$$

with $\partial(\mathbf{A}^{-1}\mathbf{b}) = \mathbf{A}^{-1}\left(\partial\mathbf{b} - \partial\mathbf{A}\mathbf{A}^{-1}\mathbf{b}\right)$. Furthermore, for a pinhole camera with focal length $f$ we have $\mu_2 = z = \frac{f\Delta x}{w}$. We obtain the precision in $z$ as

$$\lambda_2 = \sigma_2^{-2}, \quad \sigma_2^2 = \mathbf{J}\begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_{\Delta x}^2 \end{bmatrix}\mathbf{J}^\top, \quad \mathbf{J} = \begin{bmatrix} -\frac{f\Delta x}{w^2} & \frac{f}{w} \end{bmatrix} \tag{36}$$

Similarly, we have $\mu_3 = z = \frac{f\Delta y}{h}$ with precision

$$\lambda_3 = \sigma_3^{-2}, \quad \sigma_3^2 = \mathbf{J}\begin{bmatrix} \sigma_h^2 & 0 \\ 0 & \sigma_{\Delta y}^2 \end{bmatrix}\mathbf{J}^\top, \quad \mathbf{J} = \begin{bmatrix} -\frac{f\Delta y}{h^2} & \frac{f}{h} \end{bmatrix} \tag{37}$$

The unknown parameters of the proposed projection model $\sigma_u$, $\sigma_v$, $\sigma_w$, $\sigma_h$, $\Delta x$, $\Delta y$, $\sigma_{\Delta x}$ and $\sigma_{\Delta y}$ are determined empirically from a held out training set composed of 1020 stereo images with 3634 annotated vehicle 2D bounding boxes, including orientation labels (8 bins). The object depth is computed from the median disparity within the respective bounding box. and computed the corresponding disparity maps. Due to the characteristics of sliding-window detectors, we expect the noise to be dependent on the object scale. We model this relationship using linear regression with respect to the bounding box height $h$. We learn a separate $\Delta x$ for each of three car orientation classes illustrated in Fig. 8.

As the raw 3D location estimates $\{(\mathbf{m}, \mathbf{S})\}$ are noisy due to the low camera viewpoint, the uncertainties in the object detector and the ground plane estimation process, we temporally integrate detections within a tracklet $\mathbf{t}$ using a Kalman smoother [40] assuming constant velocity.

### 4.2 Vanishing Points

We detect up to two vanishing points $\mathcal{V} = \{v_1, \ldots, v_{N_v}\}$ in the last frame of each sequence, where a vanishing point is defined by a rotation angle around the $y$-axis in road coordinates, i.e. we assume that all vanishing lines are collinear with the ground plane and $v_i$ represents the yaw angle. All 3D lines that are collinear with a vanishing line intersect at the same vanishing point. For typical scenarios, two vanishing points are dominant: One which is collinear with the forward facing street and one which is collinear with the crossing street. This is illustrated in Fig. 1.

In order to detect vanishing points we make use of the method described by Kosecka et al. [43], which we have modified slightly for taking into account the known camera calibration information. We restrict the search space such that all vanishing lines are collinear with the ground plane. Additionally, we relax the model to also allow for non-orthogonal vanishing points as this is required by the intersection types encountered in our dataset. Unfortunately, traditional vanishing point detection methods [7], [43] require relatively clean scenarios and tend to fail in the presence of clutter such as cast shadows or defects in the road surface. We learn a k-nearest-neighbor classifier based on a held-out annotated set of 185 images, in which all detected line segments have been manually labeled as either *structure* (e.g. road markings, building facades) or *clutter*. The classifier's confidence on *structure* is used as a weight in the vanishing point voting process. Details on the feature set which we use can be found in [33].

Fig. 9 (left) shows the ROC curve for classifying lines into *structure* and *clutter*. The curves have been obtained by adjusting $k$ for the k-nn classifier in the learning based method and by varying the inlier threshold for [43]. Fig. 9 (right) compares the classification results for a particular scene: While the cast shadows in the lower-left part of the image causes wrong evidence for traditional detectors (top), the proposed classification step is able to reject most of these line segments (bottom).

### 4.3 Semantic Scene Labels

For extracting semantic information in the form of scene labels we use the joint boosting framework proposed in [58] to learn a strong classifier. Following [61], we

divide the last image of each sequence into patches of size $n_s \times n_s$ pixels and classify them into the categories *road*, *background* and *sky*. Details on the feature set can be found in [33]. In order to avoid hard decisions and to interpret the boosting confidences as probabilities we apply the softmax transformation [46] to the resulting scores. The semantic label of a single patch is defined as $\mathbf{s} \in \Delta^2$, where $\Delta^2$ is the unit 2-simplex, i.e., the space of possible categorical distributions. For training, we use a held out dataset of 200 hand-labeled images. After softmax normalization we obtain a label distribution $\mathbf{s}$ for each image patch $\mathbf{s} \in \mathcal{S}$, which is used in our semantic scene label likelihood described in Section 3.3.4.

### 4.4 Scene Flow and Egomotion

We compute the vehicle's egomotion and 3D scene flow using the method presented in [34]. Towards this goal we accumulate all vectors in the reference (road) coordinate system, compensate the egomotion and threshold them by their length, i.e., we remove short vectors that are likely to belong to the static environment. As the 3D scene flow likelihood doesn't account for object velocities, we normalize all flow vectors to unit length and project them onto the road plane as illustrated in Fig. 1 (red arrows in occupancy grid).

### 4.5 Occupancy Grid

Buildings represent obstacles in the scene and thus should never coincide with drivable regions (road). This assumption is incorporated into the occupancy grid feature. We construct a 2D grid in road plane coordinates from disparity measurements which classifies the area in front of the vehicle into the categories *obstacle*, *free space* and *unobserved* cells as illustrated in Fig. 1.

We make use of the efficient large-scale stereo matching algorithm ELAS [32] to compute disparity maps for each frame in each sequence. Given the disparity maps and the motion estimates from Section 4.4, we compute a 2D occupancy grid of the environment in reference road coordinates, representing obstacles and drivable (road) areas. The occupancy grid is obtained by assuming spatial independency amongst cells, applying the discrete Bayes filter [57] and ignoring moving objects (tracklets, scene flow).

## 5 EXPERIMENTAL EVALUATION

We recorded a dataset of 113 video sequences of real traffic with a duration of 5 to 30 seconds each. This corresponds to a total of 9438 frames or $\sim 944$ seconds. All sequences end when entering the intersection, i.e., when the autonomous system would need to take a decision. Fig. 11 shows the large variability in appearance. We annotate the data using GoogleMaps aerial images. For each intersection in the database we labeled the center of the intersection as well as the number, orientation and width of the intersecting streets in bird's eye perspective.

We then map the annotated geometry into the road coordinate system using the GPS coordinates of the vehicle. Note that the GPS is only used for annotation. Additionally, for all vehicle tracklets that have been detected by the approach described in Section 4.1, we annotate their lane index $l$, including parking. For vehicles that have been associated to a lane, the tangent at the closest foot point of lane $l$ is used as object orientation ground truth. Furthermore, we manually annotate all lanes in each scenario with a binary label indicating whether the lane is active, i.e., whether vehicles move on that lane.

### 5.1 Settings

Our experiments evaluate the performance of the proposed system as well as the importance of each individual feature cue, abbreviated as

| | | |
|---|---|---|
| P | = | **P**rior (see Section 3.3.1) |
| T | = | **T**racklets (see Section 3.3.2) |
| V | = | **V**anishing Lines (see Section 3.3.3) |
| S | = | **S**emantic labels (see Section 3.3.4) |
| F | = | Scene **F**low (see Section 3.3.5) |
| O | = | **O**ccupancy Grid (see Section 3.3.6) |

To gain insights into the strengths and weaknesses of each cue, we evaluate the prior alone, all terms individually (PT, PV, PS, PF, PO), all terms but one (PVSFO, PTSFO, PTVFO, PTVSO, PTVSF) and the full model (PTVSFO). For each setting we learn a separate set of parameters.

For evaluating the proposed model, we leverage 10-fold cross-validation and learn the model parameters $\Theta = \{\lambda_p, \boldsymbol{\xi}_p, \lambda_t, \lambda_v, \lambda_s, \lambda_{f1}, \lambda_{f2}, \lambda_o\}$ in each fold using contrastive divergence, as described in Section 3.5. All parameters are initialized to $\theta_i = 1$. Convergence typically occurred after 150-250 gradient ascent steps. We excluded $\zeta_t$, $\zeta_v$ and $\zeta_f$ from the maximum likelihood estimation process described in Section 3.5 as we found them hard to optimize. Instead, we estimated them using cross-validation. All parameters which are not part of $\Theta$ have been determined empirically and are summarized in Table 2. Inference is performed by drawing $n_{inf} = 10,000$ samples and computing the MAP solution. Our mixed C++/MATLAB implementation requires $\sim 8$ seconds per frame for inference and about an hour for learning the model parameters on a standard PC using 8 CPU cores. Most time is spent on feature extraction, especially object detection.

### 5.2 Topology and Geometry

To judge the performance of the proposed model, we evaluate the estimation results of each setting against several metrics. First, we measure the accuracy in topology estimation, which is the percentage of all 113 cases in which the correct topology $\kappa$ has been recovered. We propose three geometric metrics: average Euclidean error when estimating the intersection center, average street orientation error and the road area overlap. Regarding the street orientation, we assign each street to

| | |
|---|---|
| **Prior:** (Section 3.3.1) | |
| $\sigma_\alpha = 0.1$ rad | KDE kernel bandwidth |
| | |
| **Vehicle Tracklets:** (Section 3.3.2 and Section 4.1) | |
| $\tau_d = 0.2$ | NMS overlap threshold (object detection) |
| $\tau_{t1} = 0.5$ | Gating threshold (tracking stage 1) |
| $\tau_{t2} = 0.3$ | Gating threshold (tracking stage 2) |
| $\zeta_t = 10^{-20}$ | Outlier threshold |
| $\sigma_{out} = 70$ m | Std. deviation of outlier distribution |
| | |
| **Vanishing Points:** (Section 3.3.3) | |
| $\zeta_v = 10^{-10}$ | Outlier threshold |
| | |
| **Semantic Scene Labels:** (Section 3.3.4 and Section 4.3) | |
| $n_s = 4$ Px | Image patch (superpixel) size |
| $\mathbf{w}_s = (1, 1, 4)$ | Scene label weights |
| | |
| **Scene Flow:** (Section 3.3.5 and Section 4.4) | |
| $n_f = 50$ | Number of RANSAC samples |
| $\zeta_f = 10^{-15}$ | Outlier threshold |
| $\sigma_{out} = 70$ m | Std. deviation of outlier distribution |
| | |
| **Occupancy Grid:** (Section 3.3.6 and Section 4.5) | |
| $n_o = 1$ m | Occupancy grid cell size |
| $\mathbf{w}_o = (-1, 4, 1)$ | Weights of geometric prior |
| $\Delta_o = (2, 20)$ m | Margins of geometric prior |
| | |
| **Inference and Learning:** (Section 3.4 and Section 3.5) | |
| $n_{inf} = 10,000$ | Number of samples at inference |
| $n_{learn} = 10$ | Number of samples per learning iteration |
| $n_{iter} = 500$ | Number of learning iterations |

TABLE 2: **Setting of Constants in the Model.** For reproducibility of our results, we specify all constants in our model. On acceptance of this paper, we will also release the data and MATLAB/C++ code.

its (rotationally) closest counterpart in the ground truth layout in order to decouple the orientation measure from the estimated topology $\kappa$. More precisely, we take the layout with the smaller number of streets and assign all streets to their closest counterparts in the layout with the larger number of streets. Finally, the road area overlap measures to which extend the estimated road layout overlaps with the ground truth by computing the intersection-over-union of both road areas. Here, the road area is defined as the concave hull enclosing all streets up to a distance of three times the average ground truth street width.

As evidenced by Table 3, each feature is able to improve results compared to the prior (column 1-6). The strongest cues in our framework are tracklets, 3D scene flow and the occupancy grid features. This indicates that despite its noisy nature, depth information is very important. The smallest gain in performance is observed for the vanishing point feature as it only improves performance in combination with other cues.

Performance improves further when combining features. In terms of topology estimation, the best results have been obtained by making use of all information. Regarding the geometric error measures all settings that include the occupancy grid feature perform comparably well. This leads us to the conclusion that occupancy information is complementary information, while other cues such as 3D scene flow and vehicle tracklets can partly replace each other.

## 5.3 Tracklet Associations and Semantic Activities

Besides the geometric reasoning discussed so far, an important aspect in real-world applications is to understand the scene at a higher level. This includes the association of vehicle tracklets to lanes ('Tracklet Accuracy' in Table 3) as well as the detection of active lanes ('Lane Accuracy' in Table 3), where we assume a single lane connecting each inbound with each outbound street, as described in Section 3.1. With active we refer to lanes on which at least one vehicle is moving. For evaluating the above mentioned metrics we extract all *unique* tracklets, defined by a minimum length of 10 meters. We define a lane as active if at least one tracklet has been uniquely assigned to it. The tracklet and lane accuracies for all settings are depicted in Table 3 (rows 5 and 6). As expected, the best results are obtained when either the 3D scene flow or the vehicle tracklet features are present, with $80\%$ accuracy in tracklet associations and $90\%$ accuracy in detecting active lanes.

## 5.4 Object Detection and Orientation Estimation

As we have shown in Section 5.2, objects help in estimating the layout and geometry of the scene. On the other hand, knowledge about the road layout should also help in improving the performance of object detectors, both in accuracy and object orientation estimation.

First, we re-estimate the orientations of all objects that are used as input to our model. For associating the tracklets to lanes and the detections to lane spline points, we employ the inference procedure described in Section 3.4.2. We select the tangent angle at the associated spline's foot point $s$ on the inferred lane $l$ as our novel orientation estimate. Table 3 (row 7) shows that we are able to significantly reduce the orientation error of moving vehicles from 32.6 degrees, corresponding to the orientation error of the raw detections (not depicted in the table), down to 14.0 degrees when using our model in combination with vehicle tracklets or 3D scene flow.

We also manually annotated all cars in the last frame of each sequence using 2D bounding boxes, yielding 355 labeled car instances in total. Given the object detections and the inferred road geometry from Section 5.2, we re-score each object detection by adding the following term to the scores of [25]

$$0.5 \left[ \max_l \ \exp\left(-\frac{\Delta_l^2}{2w^2}\right) + \sum_{i=1}^{3} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) \right] - 1$$

Here $\Delta_l$ is the distance of a car detection to lane spline $l$, $w$ is the estimated street width and $\{\mu_i, \sigma_i\}$ are mean and standard deviation of the object width, height and position, respectively. These estimates are obtained from a held-out training set using maximum likelihood estimation. Fig. 10 depicts the precision-recall curves for the L-SVM baseline [25] and our approach. Note that our geometric and topological constraints increase detection performance significantly, improving average precision

| | P | PT | PV | PS | PF | PO | PVS FO | PTS FO | PTV FO | PTV SO | PTV SF | PTV SFO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Topology Accuracy (%,↑) | 19.5 | 64.6 | 20.3 | 55.8 | 66.4 | 83.2 | 91.1 | 89.4 | 91.1 | 90.3 | 72.6 | **92.0** |
| Location Error (m,↓) | 6.8 | 5.1 | 6.7 | 7.9 | 4.5 | 4.0 | 3.0 | 2.9 | **2.7** | 3.1 | 4.8 | 3.0 |
| Street Orient. Err. (deg,↓) | 8.7 | 5.4 | 8.7 | 7.3 | 5.2 | 5.1 | 3.6 | 3.6 | **3.6** | 3.6 | 4.7 | 3.6 |
| Road Area Overlap (%,↑) | 33.1 | 51.0 | 32.9 | 41.2 | 53.6 | 63.4 | 69.3 | 69.7 | **71.2** | 68.9 | 57.5 | 69.9 |
| Tracklet Accuracy (%,↑) | 28.1 | 78.7 | 30.0 | 35.8 | 79.2 | 65.1 | 80.7 | 80.3 | 79.8 | 77.6 | 81.6 | **82.0** |
| Lane Accuracy (%,↑) | 77.3 | 90.3 | 78.0 | 80.2 | **90.5** | 85.2 | 89.7 | 89.4 | 89.8 | 88.3 | 90.2 | 89.7 |
| Object Orient. Err. (deg,↓) | 53.0 | 17.7 | 54.4 | 45.1 | 17.5 | 24.2 | **14.0** | 15.1 | 14.9 | 15.4 | 15.1 | 14.3 |
| Object Detection AP (%,↑) | 73.6 | 73.7 | 73.7 | 73.1 | 73.6 | 74.1 | 74.1 | 74.1 | **74.2** | 74.1 | 73.7 | 74.0 |

| | | |
|---|---|---|
| P | = | **P**rior |
| T | = | **T**racklets |
| V | = | **V**anishing Lines |
| S | = | **S**emantic labels |
| F | = | Scene **F**low |
| O | = | **O**ccupancy Grid |

TABLE 3: **Quantitative Results.** ↑ (↓) means higher (lower) is better. All numbers represent averages over all 113 sequences.
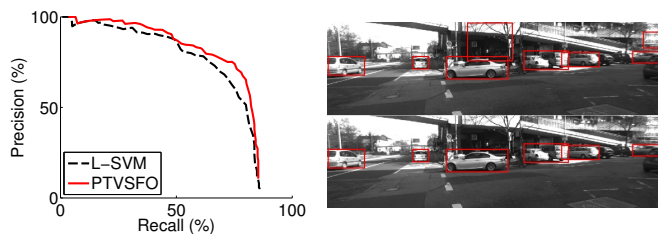


Fig. 10: **Improving Object Detection.** Precision-recall curves (left) and an example where our algorithm (bottom-right) is able to eliminate false positives of a state-of-the-art object detector [25] (top-right).

from 69.9% to 74.2%. The benefits of including this knowledge into the detection process are also illustrated in Fig. 10 (right). In order to include the partly occluded car to the right into the detection result, the threshold of the baseline has to be lowered to a value which produces two false positives (top). In contrast, our re-scored ranking is able to handle this case (bottom).

## 5.5 Qualitative Results

Fig. 11 illustrates our inference results for the setting 'PTVSFO', with the most likely lanes for each unique tracklet, indicated by an arrow. The ego-vehicle (observer) is depicted in black. For most sequences the road layout has been estimated correctly and the vehicles have been assigned to the correct lanes. Only vehicles that are very far away or visible only for a couple of frames pose problems in terms of their lane associations. However, note that in many cases this didn't affect the estimated layout. More results can be found in our supplementary video which is available from our project website.[2]

## 6 CONCLUSIONS AND FUTURE DIRECTIONS

We have developed an approach that allows autonomous vehicles to estimate the layout of urban intersections based on onboard stereo imagery alone. Our model does not rely on strong prior knowledge such as intersection maps, but infers all information from different types of visual features that describe the static environment of the crossroads (i.e., facades of houses) and the motions of objects (i.e., cars) in the scene. Different from previous approaches, our method does not rely on traditional features like lane markings or curb stones which are

often unavailable in urban scenarios. While we found performance improvements for all features, occupancy grids as well as vehicle tracklets and 3D scene flow have been identified as the strongest and most important cues.

To accommodate for the ambiguities in the visual information we developed a probabilistic model to describe the appearance of the proposed features relative to the intersection geometry. We utilized Markov Chain Monte Carlo sampling for inference to cope with the complex relationship between the crossroad layout and the appearance of the features. As evidenced by the experiments, our approach is able to recognize urban intersections reliably with an accuracy of up to 90% on a data set of 113 realistic real-world intersections. For most of these intersections, traditional approaches based on lane markings and curb stones would have failed due to the absence of these feature cues. Moreover, we found that context from our model helps to improve the performance of state-of-the-art object detectors in terms of detecting objects as well as estimating their orientation.

Our approach serves as basis for future improvements. Currently, the assumption that tracklets are independent can lead to implausible configurations such as cars colliding with each other. Including higher-level background knowledge on how cars can pass an intersection will help to reduce ambiguities and increase robustness. Furthermore, improved sensor observations and more computing power will allow for more accurate motion models and lane representations. Another interesting direction of research will be to integrate information from other traffic participants (e.g., pedestrians) into the model to perform collaborative inference.

**Andreas Geiger** received his Diploma in computer science and his Ph.D. degree from Karlsruhe Institute of Technology in 2008 and 2013. Currently, he is a research scientist in the Perceiving Systems group at the Max Planck Institute for Intelligent Systems in Tübingen. His research interests include computer vision, machine learning and scene understanding.

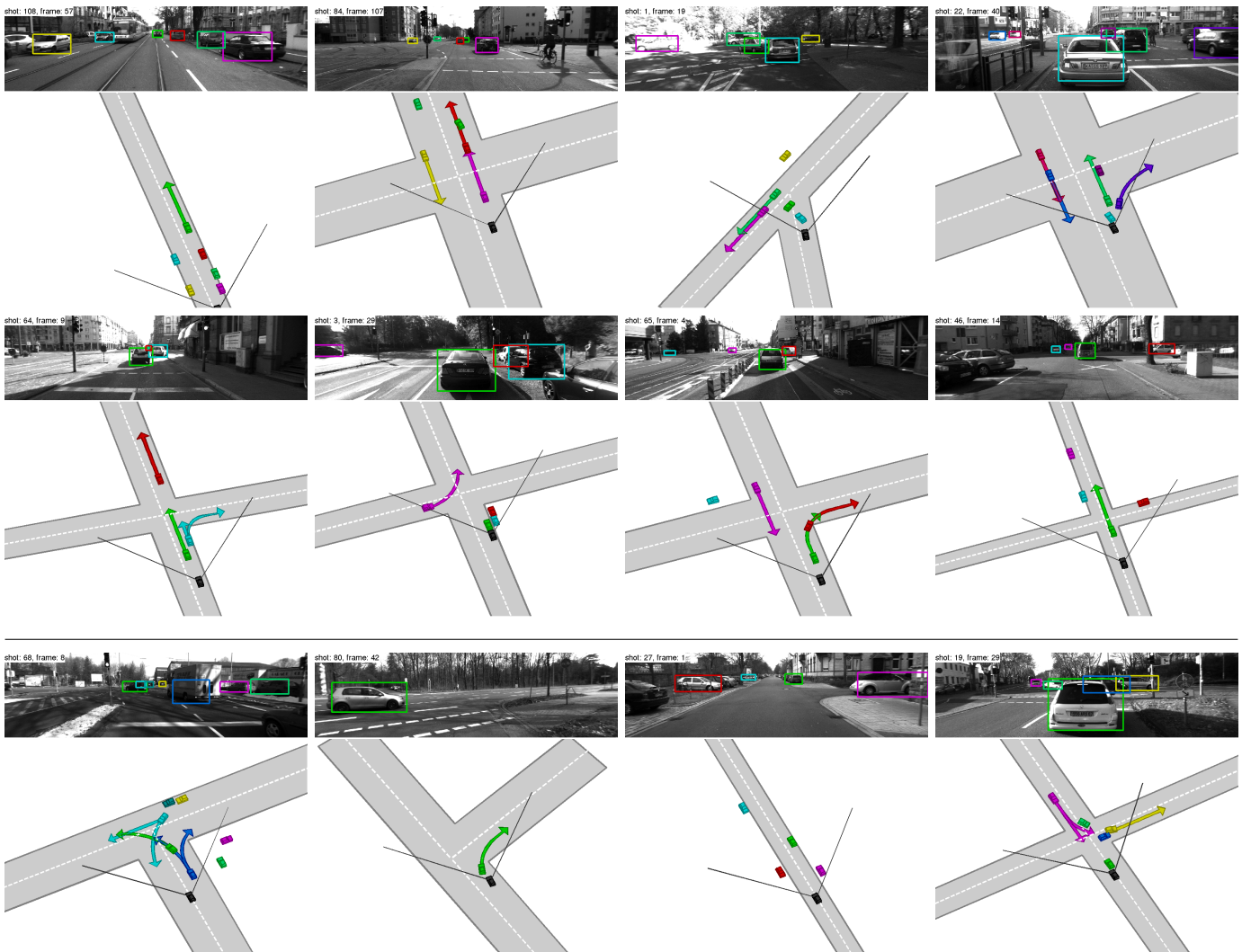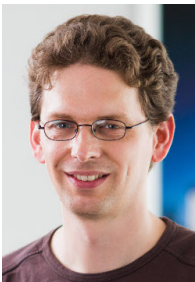2. http://www.cvlibs.net/projects/intersection

Fig. 11: **Inference Results.** For each sequence we show the input image (top) and the inference result (bottom). Arrows indicate the predicted driving direction(s). The last row shows four cases where topology estimation has failed. However, note that in many cases the lanes are still associated correctly.



**Martin Lauer** studied computer science at Karlsruhe University and received his Ph.D. in computer science from University of Osnabrück. As research group leader he works in the fields of machine vision, robotics, and machine learning at Karlsruhe Institute of Technology.



**Christoph Stiller** is professor at the Karlsruhe Institute of Technology and head of the Institute of Measurement and Control. At the same time he is director at the FZI Research Center for Information Technology heading the mobile perception research group. His research focus is in stereo vision, camera based scene understanding, driver assistance systems, and autonomous vehicles.



**Christian Wojek** received his Diplom degree in Computer Science from the University of Karlsruhe in 2006 and his PhD from TU Darmstadt in 2010. He was awarded a DAAD scholarship to visit McGill University from 2004 to 2005. He was with MPI Informatics Saarbruecken as a postdoctoral fellow from 2010 to 2011 and in 2011 joined Carl Zeiss Corporate Research. His research interests are object detection, scene understanding and activity recognition in particular with application to real world scenarios.



**Raquel Urtasun** Raquel Urtasun is an Assistant Professor at TTI-Chicago. Previously, she was a postdoctoral research scientist at UC Berkeley and ICSI and a postdoctoral associate at the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. Raquel Urtasun completed her PhD at the Computer Vision Laboratory at EPFL, Switzerland. Her major interests are statistical machine learning and computer vision.

# REFERENCES

[1] Y. Alon, A. Ferencz, and A. Shashua. Off-road path following using region classification and geometric projection constraints. In *CVPR*, 2006.

[2] J. M. Alvarez, T. Gevers, and A. M. Lopez. 3d scene priors for road detection. In *CVPR*, 2010.

[3] M. Aly. Real time detection of lane markers in urban streets. In *IV*, 2008.

[4] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[5] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, 2010.

[6] S. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. In *CVPR*, 2010.

[7] O. Barinova, V. Lempitsky, E. Tretyak, and P. Kohli. Geometric image parsing in man-made environments. In *ECCV*, 2010.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st ed. 2006 edition, October 2006.

[9] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009.

[10] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli. Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge. *ARC*, 36(1):161–171, 2012.

[11] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using SfM point clouds. In *ECCV*, 2008.

[12] M. A. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *CVPR*, 2013.

[13] M. Buehler, K. Iagnemma, and S. Singh, editors. *The DARPA Urban Challenge*, volume 56 of *Advanced Robotics*, 2009.

[14] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *ECCV*, 2010.

[15] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[16] J. Crisman and C. Thorpe. Scarf: A color vision system that tracks roads and intersections. *TRA*, 9(1):49–58, February 1993.

[17] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski. Self-supervised monocular road detection in desert terrain. In *RSS*, 2006.

[18] R. Danescu and S. Nedevschi. Probabilistic lane tracking in difficult road scenarios using stereovision. *TITS*, 10(2):272–282, 2009.

[19] C. De Boor. *A Practical Guide to Splines*. Number 27 in Applied Mathematical Sciences. Springer-Verlag, 1978.

[20] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[21] E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *PAMI*, 14(2):199–213, Feb 1992.

[22] W. Enkelmann, G. Struck, and J. Geisler. Roma - a system for model-based analysis of road markings. In *IV*, 1995.

[23] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 31:1831–1846, 2009.

[24] A. Ess, T. Mueller, H. Grabner, and L. van Gool. Segmentation-based urban traffic scene understanding. In *BMVC*, 2009.

[25] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32:1627–1645, 2010.

[26] D. M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *PAMI*, 29:1408–1421, 2007.

[27] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV*, 73:41–59, 2007.

[28] A. Geiger. *Probabilistic Models for 3D Urban Scene Understanding from Movable Platforms*. PhD thesis, KIT, 2013.

[29] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler. Team annieway's entry to the grand cooperative driving challenge 2011. *TITS*, 2012.

[30] A. Geiger, M. Lauer, and R. Urtasun. A generative model for 3d urban scene understanding from movable platforms. In *CVPR*, 2011.

[31] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

[32] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010.

[33] A. Geiger, C. Wojek, and R. Urtasun. Joint 3d estimation of objects and scene layout. In *NIPS*, 2011.

[34] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3d reconstruction in real-time. In *IV*, 2011.

[35] V. Gengenbach, H. H. Nagel, F. Heimes, G. Struck, and H. Kollnig. Model-based recognition of intersections and lane structures. In *IV*, 1995.

[36] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[37] G. Hinton. Training products of experts by minimizing contrastive divergence. *NC*, 14:2002, 2000.

[38] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80:3–15, 2008.

[39] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, October 2007.

[40] R. E. Kalman. A new approach to linear filtering and prediction problems. *JBE*, 1(82):35–45, 1960.

[41] V. Kastrinaki. A survey of video processing techniques for traffic applications. *IVC*, 21(4):359, 2003.

[42] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 27(11):1805–1819, 2005.

[43] J. Kosecka and W. Zhang. Video compass. In *ECCV*, 2002.

[44] D. Kuettel, M. D. Breitenstein, L. V. Gool, and V. Ferrari. What's going on?: Discovering spatio-temporal dependencies in dynamic scenes. In *CVPR*, 2010.

[45] H. W. Kuhn. The Hungarian method for the assignment problem. *NRLQ*, 2:83–97, 1955.

[46] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, 2005.

[47] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled detection and tracking from static cameras and moving vehicles. *PAMI*, 30(10):1683–1698, 2008.

[48] M. Lutzeler and E. D. Dickmanns. Ems-vision: recognition of intersections on unmarked road networks. In *IV*, 2000.

[49] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *TITS*, 7(1):20–37, March 2006.

[50] V. Nedovic, A. W. M. Smeulders, A. Redert, and J. M. Geusebroek. Stages as models of scene geometry. *PAMI*, 32:1673–1687, 2010.

[51] D. Pomerleau. Ralph: Rapidly adapting lateral position handler. In *IV*, pages 506–511, 1995.

[52] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *CVPR*, 2003.

[53] C. Rasmussen. Road shape classification for detecting and negotiating intersections. In *IV*, 2003.

[54] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D depth reconstruction from a single still image. *IJCV*, 76:53–69, 2008.

[55] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 31:824–840, 2009.

[56] G. Singh and J. Kosecka. Acquiring semantics induced topology in urban environments. In *ICRA*, 2012.

[57] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[58] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

[59] X. Wang, X. Ma, and W. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *PAMI*, 31:539–555, 2009.

[60] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *PAMI*, 2012.

[61] C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV*, 2008.

[62] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR*, 2011.

[63] Q. Yu and G. Medioni. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *PAMI*, 31(12):2196–2210, 2009.

[64] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nuchter, and J. Wang. 3d lidar point cloud based intersection recognition for autonomous driving. In *IV*, 2012.