Multi-Scale Object Candidates for Generic Object Tracking in Street Scenes

Aljoša Ošep, Alexander Hermans, Francis Engelmann, Dirk Klostermann, Markus Mathias and Bastian Leibe

***Abstract*— Most vision based systems for object tracking in urban environments focus on a limited number of important object categories such as cars or pedestrians, for which powerful detectors are available. However, practical driving scenarios contain many additional objects of interest, for which suitable detectors either do not yet exist or would be cumbersome to obtain. In this paper we propose a more general tracking approach which does not follow the often used tracking-by-detection principle. Instead, we investigate how far we can get by tracking unknown, generic objects in challenging street scenes. As such, we do not restrict ourselves to only tracking the most common categories, but are able to handle a large variety of static and moving objects. We evaluate our approach on the KITTI dataset and show competitive results for the annotated classes, even though we are not restricted to them.**

I. INTRODUCTION

Outdoor visual scene understanding is a key component for autonomous mobile systems. Specifically, detection and tracking of other traffic participants are essential steps towards safe navigation and path planning through populated urban areas. Recent results on standard benchmarks [1] show that some object categories, such as cars or pedestrians, can already be tracked rather reliably by state-of-the-art tracking-by-detection approaches [2], [3], [4], [5]. In practical driving scenarios, however, there are numerous other objects that could pose potential safety hazards and it quickly becomes infeasible to train specific detectors for all possible classes.

In this paper, we therefore investigate the problem of *generic* object tracking in street scenes. Rather than starting from the output of a class-specific detector, we try to extract a set of object candidate regions purely from low-level cues and to track them over time. This approach has the advantage that it is not a priori restricted in the types of objects that can be tracked. However, the tracking task becomes much more challenging, since it requires solving a complex figure-ground segmentation problem in every frame to decide which scene regions contain valid objects and at what spatial extent those objects should be represented.

In order to address this segmentation problem, we make use of scene information from stereo depth to generate Generic Object Proposals (GOPs) in 3D and keep only those proposals that can consistently be tracked over a sequence of frames. In our tracking step, we link these object proposals into trajectories and integrate the individual 3D measurements into a 3D shape model for each tracked object. We jointly reason about valid object proposals and their corresponding trajectories via a model selection based multi-object tracking procedure.

All authors are with Visual Computing Institute, RWTH Aachen University. Email: {lastname}@vision.rwth-aachen.de

Fig. 1. We propose an approach to track generic objects in street scenes that goes beyond the capabilities of pre-trained object detectors. Our approach can handle a wide variety of static and moving objects of different sizes and robustly track them. Blue areas indicate potential object regions.

For such an approach to work, the generation of good object proposals is a key requirement. This is a very challenging problem, since the unknown objects may originate from vastly different scales (see Fig. 1), measurements from nearby objects tend to merge, and objects close to scene structures are difficult to segment due to often noisy stereo data. To reach acceptable recall values, state-of-the-art appearance-based object proposal generation approaches [6] typically need several thousand object proposals per frame, two orders of magnitude more than what would be tractable to use in a tracking framework.

We propose a novel robust *multi-scale object proposal extraction* procedure that uses a two-stage segmentation approach. First, a *coarse supervised segmentation* removes non-object regions corresponding to known background categories such as *road*, *building*, or *vegetation*. Next, we perform a *fine unsupervised multi-scale segmentation* to extract scale-stable object proposals from the remaining scene regions. As many of these proposals may overlap and the correct object scale often cannot be determined on a single-frame basis, we perform multi-hypothesis tracking at the level of object proposals.

In summary, our main contributions are: (1) We present a novel, scalable approach that successfully tracks a large variety of generic objects in challenging street scenes. (2) As a key component of this approach, we propose a robust multi-scale 3D object proposal extraction procedure based on a two-stage segmentation and scale-stable clustering. (3) We demonstrate the validity of our approach quantitatively and

qualitatively on the KITTI dataset [1]. We show that our approach can compete with state-of-the-art detector-based methods in close and medium camera distance.

Object definition. In the remainder of this paper we refer to an "object" as an entity that appears in urban street scenes, sticks out of the ground plane, has a well-defined closed boundary in space [7], and is surrounded by a certain band of free-space. In addition, objects need to appear consistently in a sequence of frames, either moving or not, and maintain a roughly consistent appearance. We also assume a size range for objects of interest between 0.5m and 5m. This definition includes other traffic participants, as well as static/parked vehicles and items of street furniture. We explicitly exclude only items that are better explained by stuff categories such as *vegetation* or *building facade*.

II. RELATED WORK

Many approaches have been proposed for object tracking in street scenarios [8], [2], [9], [3], [5]. Most of those follow a tracking-by-detection strategy by first applying detectors trained for specific categories on each frame and then linking the detections into trajectories. The KITTI tracking benchmark [1] gives a good overview of such tracking methods. Zhang *et al.* [5] pose tracking as a maximum-a-posteriori data association problem with non-overlap constraints. Pirsiavash *et al.* [3] consider tracking as a spatio-temporal grouping problem and propose greedy global optimization approach. Milan *et al.* [2] use a continuous energy minimization approach that takes into account physical constraints and track persistence. While these approaches obtain impressive results, they have the drawback that they assume that all interesting object categories are known beforehand and that detectors can be trained for each category.

Recently, the problem of tracking generic objects has received more attention. For automotive scenarios several approaches address this problem using highly precise LIDAR data as input [10], [8], [11], [12]. Petrovskaya *et al.* [10] use a model-based approach to detect car-sized objects in laser point clouds. Held *et al.* [12] utilize 3D shape and color information to obtain precise velocity estimates of generic objects in LIDAR data. In contrast, we use depth information obtained from a stereo camera pair, which is far less accurate and requires a more robust processing pipeline.

Other approaches try to find and track generic objects based on motion segmentation. For example, Bewley *et al.* [13] use a self-supervised framework to detect dynamic object clusters extracted from a monocular camera stream.

In contrast to those approaches, we are also interested in tracking static instances of interesting objects.

To the best of our knowledge, only few approaches deal with generic object tracking from stereo depth. Nguyen *et al.* [14] also target generic objects of several sizes, but they only track moving objects, with the purpose of generating improved occupancy grids of the scene for a driver assistance system. While our pipeline is similar to the approaches of [9], [15], they only track pedestrian sized objects, whereas we aim to also track larger objects such as cars and vans.

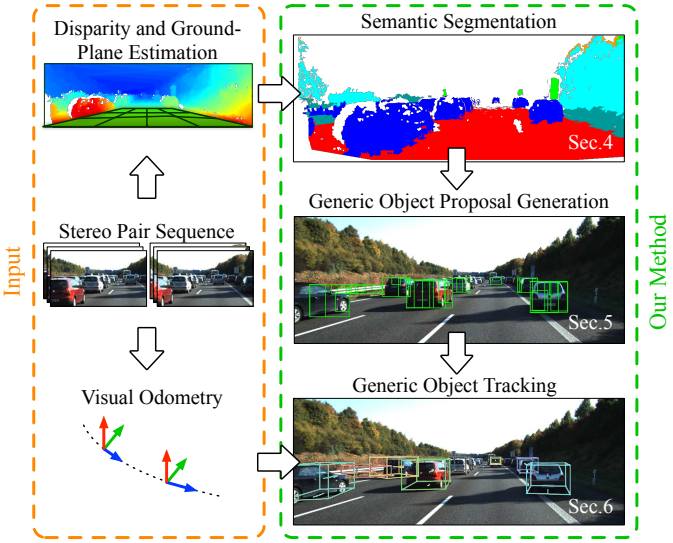

Fig. 2. High-level overview of our pipeline.

A key part of our pipeline is the generation of good Generic Object Proposals (GOPs). Several previous methods have been proposed for this step, often based on LIDAR data. Wang *et al.* [16] use a minimal-spanning-tree clustering approach to extract 3D object proposals from LIDAR data and then classify them into background, bicyclist, car, or pedestrian. Ioanneu *et al.* [17] propose a Difference-of-Normals operator to extract scale-stable object proposal regions from LIDAR data. We compare against this approach in Sec. VII. Bansal *et al.* [18] propose a semantic structure labeling approach based on stereo data in order to create proposal regions for a pedestrian detector. The resulting regions are too coarse and would not generalize well to all interesting objects. There is also a large set of approaches that try to find good object proposals in the form of bounding boxes from color images [7], [6], [19]. While state-of-the-art methods such as EdgeBoxes [6] obtain a very high recall, their precision is too low to be applicable for our approach.

Our approach builds upon a semantic segmentation to reject scene parts that can be well explained by background categories. Several other approaches have already demonstrated semantic segmentation on different subsets of KITTI [1]. Xu *et al.* [20] fuse information from several sensors to classify superpixels, while we only rely on stereo data. Ladický *et al.* [21] jointly infer disparity maps and dense semantic segmentations based on a monocular image using a combination of depth and semantic classifiers. Ros *et al.* [22] pre-compute a high-quality semantic map of the static parts of a scene in order to later on label the environment based on the current location within that map. Objects that appeared in the scene can then automatically be labeled. While this is fast and gives good results, our approach also generalizes to unknown scenes.

III. METHOD OVERVIEW

Fig. 2 gives an overview of our approach. Given a sequence of stereo image pairs, we compute disparity maps using ELAS [23]. From a disparity map we generate point cloud and fit a ground plane using RANSAC. To narrow

down the 3D search space for potential objects, we perform supervised coarse semantic segmentation on the point cloud (Sec. IV). Based on the idea of things and stuff [24], we remove all points that belong to stuff regions such as road, sky, or building points. This gives us a coarse idea where potential objects could be located. On the remaining point cloud we perform multi-scale search for generic object proposals (Sec. V). Each proposal is defined by a set of 3D points. The result is an over-complete set of possible 3D objects.

In the last step we perform object tracking. First, we transform the 3D object proposals to the common coordinate frame using visual odometry of [25] and link them across frames. Next, we identify the best set of objects and their corresponding tracks (Sec. VI). We perform this selection jointly in a model selection based multi-hypothesis tracking framework, which searches for the subset of object trajectory hypotheses that together best explains the observed data.

IV. SEMANTIC SEGMENTATION

We use a supervised semantic segmentation approach to classify parts of the scene that do not resemble objects and can therefore be removed for further processing. In contrast to the classical semantic segmentation tasks, we are interested in correctly recognizing the known background categories while generalizing to potentially unseen object categories. To achieve that, we specifically use features that capture the background categories well. We treat cars and pedestrians as one single object class, such that after semantic segmentation we know that something *is* an object, but not of what kind. We follow the design of a typical segmentation pipeline: starting with an over-segmentation, features are extracted for each segment and are then used to classify the segments into semantic categories. A Conditional Random Field (CRF) is then applied to enforce spatial coherence. As our further approach operates in 3D, we use the VCCS algorithm [26] to partition the point cloud into segments. For each segment we compute several features which can be grouped into four categories:

Appearance. We compute *L*a*b** histograms over the points within a segment. We use three separate histograms for *L**, *a**, and *b**, each containing 10 bins (30 dimensions). Furthermore, we compute the mean and covariance of the *L*a*b** gradients within the segment (3+6 dimensions as the covariance is symmetric). Finally, we add histograms of textons, similar to those used in [27]. We textonize the whole image and create a histogram of textons within the segment (50 dimensions), giving a total of 89 dimensions. Only the appearance features are based on the color image, while all further features are based on the 3D point cloud.

Density. These features are largely inspired by Bansal *et al*. [18]. Based on the orientation of an estimated ground plane, we slice the 3D space into 3 height bands and project the points of each band onto a density map. This gives us densities for 3 height regions. The density maps are then discretized using 3 resolutions. By projecting a segment's centroid onto each density map we are able to select a cell in each layer and resolution ($3 \times 3 = 9$ grid cells). We also consider the 4-neighborhood of the selected cells in each layer ($4 \times 9 = 36$ grid cells), resulting in a total of 45 dimensions. Furthermore, we count the 3D points within a segment, which represents the density of the segment itself, summing up to a total of 46 feature dimensions.

Geometry. Based on the covariance matrix of the 3D points within one segment, we compute several spectral and directional features [28]. From the eigenvalues we compute the "point-ness", "linear-ness", "surface-ness" and curvature of the segment. From the eigenvectors we determine the segment normal and the cosines between both the normal and tangent vectors and the ground plane normal. Finally, we compute a tight bounding box of the segment along the principal axes. This results in a total of 12 feature dimensions.

Location. This feature represents a location prior with 3 dimensions. It consists of: the height of the segment centroid, the depth of the segment centroid and the horizontal angle between the camera's optical axis and the vector from the camera center to the segment centroid.

Thus, our resulting feature vector consists of a total of 150 dimensions. We then train a Random Forest classifier [29] with single-attribute tests, yielding class posteriors for every segment. A fully connected CRF [30], defined over the segment centers in 3D, further improves the results. We use a close-range smoothing kernel defined only over the 3D centroid locations and a larger-range appearance kernel defined over the 3D centroid and the average *L*a*b** color of a segment. From this semantic segmentation we only consider the segments labeled as *object* for our further steps.

V. GENERIC OBJECT PROPOSAL GENERATION

The multi-scale object proposal generation method produces a ranked set of object proposals (GOPs) from the remaining object regions within the point cloud. In addition to correct object proposals (targets for tracking), this set may still contain under- and over-segmentations (*e.g.*, car parts, groups of pedestrians, pedestrians merged with other objects). These overlapping and competing proposals are a major difference to previous single-scale approaches [9] and make the data association task more challenging.

In order to support efficient data association and tracking, the object proposal generation procedure should achieve a high recall with a very small set of object proposals. Current appearance-based object proposal methods are able to achieve good recall, but at the cost of very large proposal sets (for an overview see [19]).

The multi-scale search for object proposals is necessary for several reasons. Firstly, sizes of potentially interesting objects fundamentally differ (*e.g.*, pedestrians and vans). Secondly, the observed objects might be just partially visible. Noisy stereo point clouds typically contain severe depth artifacts and outliers. This makes our problem even harder and requires a robust approach, which we describe in detail in the next subsections. In a nutshell, we first project the 3D point cloud to the ground plane and compute a density

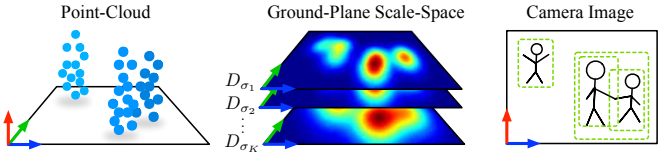

Fig. 3. Semantic segmentation allows us to only consider points labeled as object. Since object sizes are unknown, we consider different scales of the ground-plane density map.

map of the 3D points. Then we perform multi-scale search for object proposals as follows. We iteratively smooth the density map and identify blobs (clusters) around modes in the density map using Quick-Shift [31] at each scale. Our final proposals are clusters that persist in the scale space of the density map.

Scale-Space Representation of the Density Map. First, we discretize the ground plane of the point cloud into a regular grid and compute the point-density map $\mathbf{D}$ by projecting the 3D point cloud to the ground plane. Each grid cell stores the scalar value representing the density of points falling into the cell. In addition, cells store a list of associated 3D points. We create a ground-plane scale-space representation of the density map $\mathbf{D}_{\sigma_k},\ k = 1 \dots K$ by convolving $\mathbf{D}$ with a Gaussian kernel σ_k whose size increases in each iteration k (see Fig. 3).

Multi-Scale Clustering. In the next step, we apply Quick-Shift clustering [31] to obtain the modes of the scale-filtered density $\mathbf{D}_{\sigma_k}$: $\mathcal{C}_k = \{cluster_k(m)\}, m = 1 \dots \#clusters$. A cluster $cluster_k(m) = \left[\{cell_c\}_m, BB_m^{2D}, s_m\right]$ is defined by the set of cells $\{cell_c\}_m,\ c = 1 \dots \#cells$ that converged to its mode. BB_m^{2D} represents a 2D bounding box that is computed by projecting the corresponding 3D points to the image plane (see Fig. 3) and s_m is a scale-stability count.

Identification of Scale-Stable Clusters. In order to obtain a compact set of GOPs we identify the clusters that persist over scales. This step is motivated by results from scale-space filtering [32], namely that the most scale-stable proposals also tend to be the most salient ones. We identify scale-stable clusters by iterating through cluster sets $\mathcal{C}_k,\ k = 1 \dots K$ and search for *similar* clusters between sets $\mathcal{C}_k$ and $\mathcal{C}_{k+1}$. If two clusters $cluster_k(j)$, $cluster_{k+1}(l)$ are very similar according to our scale-stability criterion, then we merge them. This is done by removing $cluster_k(j)$ from $\mathcal{C}_k$ and merging it with $cluster_{k+1}(l)$ and incrementing the scale-stability count s_m of $cluster_{k+1}(l)$ by 1. In our scenario, two clusters should be declared as *similar* when they (roughly) correspond to the same object. This motivates the following scale-stability criterion: two clusters $cluster_k(j)$ and $cluster_{k+1}(l)$ are *similar* when their bounding boxes BB_j^{2D} and BB_l^{2D} have a very high overlap. To be specific, we compute the Jaccard Index $J(\cdot,\cdot)$ of the two bounding boxes and declare them as the same cluster if $J\left(BB_k^{2D}, BB_l^{2D}\right) > 0.9$.

Finally we obtain a set of GOPs $\{\mathbf{\Omega}_i^t\}$ for frame t where each GOP is defined as:

$$\mathbf{\Omega}_i^t = \left[\mathbf{p}_i^t, \mathbf{C}_{i,3D}^t, \mathbf{h}_i^t, S_i^t, r_i^t\right], \tag{1}$$

where $\mathbf{p}_i^t$ is the 3D position of the i^{th} GOP, projected onto the ground plane. $\mathbf{C}_{i,3D}^t$ is a 3×3 covariance matrix representing the uncertainty in 3D position $\mathbf{p}_i^t$, computed as [33]

$$\mathbf{C}_{i,3D}^t = \left(\mathbf{F}_{c_L}\mathbf{C}_{2D}^{-1}\mathbf{F}_{c_L} + \mathbf{F}_{c_R}\mathbf{C}_{2D}^{-1}\mathbf{F}_{c_R}\right)^{-1}, \tag{2}$$

where $\mathbf{F}_{c_L}, \mathbf{F}_{c_R}$ are Jacobians of the projection matrices of both cameras and $\mathbf{C}_{2D}$ is the covariance of pixel measurements. $\mathbf{h}_i^t$ denotes a color histogram, computed by dividing the bounding box of the GOP into 4×4 cells and stacking their RGB color histograms. $S_i^t \in \mathbb{R}^3$ denotes the set of 3D point measurements of the GOP (in the camera space) and the scalar $r_i^t \in [0,1]$ is the object stability score, computed as $r_i^t = \frac{s_i}{K}$, where s_i is scale-stability of the proposal.

VI. TRACKING

Starting with the previously introduced, possibly overlapping GOPs $\left\{\mathbf{\Omega}_i^{0:t}\right\}$, we now want to find a set of most likely objects and their trajectories $\{H_n\}$. Our basic assumption is that correct GOPs have a higher chance of producing stable trajectories with consistent appearance than GOPs caused by noise and incorrect segmentations.

We approach this problem by performing tracking and object selection jointly in a multi-hypothesis tracking framework. Other than classic tracking approaches we are not only looking for physically exclusive inlier detections (*i.e.* is the track continued by detection A or detection B?), but we also have an inlier hypothesis ambiguity on physically overlapping object proposals (see Fig. 4).

We tackle this challenging multi-hypothesis tracking problem on the object proposal level by maintaining a list of physically overlapping *object-trajectory hypotheses* that compete for the (potentially overlapping) GOPs. At each time step, our algorithm selects a subset of hypotheses, that best explains the observations. We formulate tracking as a model selection procedure and extend our previous work [34], [35], where trajectories with consistent motion and appearance are preferred. Additionally, our method takes temporal consistency of the 3D shape of the tracked object into account. Our method is also capable of keeping track of currently not selected track hypotheses. As a concrete example, this means that we may track a group of pedestrians as a single object over a sequence of frames[1], but we also keep hypotheses for the individual pedestrians. If at some point their motion starts diverging, the observed data can better be explained by individual pedestrian hypothesis.

Tracking is performed on the estimated ground plane and the camera pose computed for each frame using the Visual Odometry method of [25]. In order to obtain a stable 3D shape representations of the tracked objects, we integrate the noisy 3D measurements of the GOPs over time. In following, we will introduce the quadratic pseudo-Boolean optimization (QPBO) tracking method by Leibe *et al.* [34] and our extension of the approach, that enables us to perform

[1]Remember that we do not have pedestrian specific knowledge, such that a group of pedestrians is a valid object.

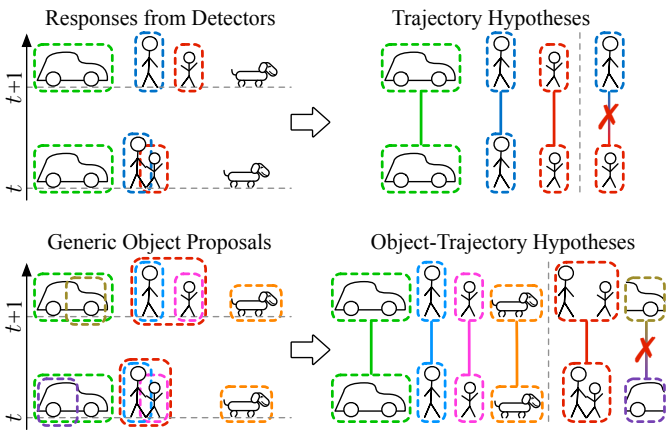

Fig. 4. Tracking-by-detection associates detections and rejects the incorrect tracks *(top)*. We associate GOPs and penalize incorrect associations (*e.g.*car parts) but associate both individual pedestrians and pedestrian groups *(bottom)*.

tracking without using a detector and track regions that likely correspond to the valid objects.

A. QPBO Tracking

The idea of [34] is to use a detector to generate an over-complete (possibly physically implausible) set of trajectory hypotheses. Then a (physically plausible) set of hypotheses is selected by solving a quadratic pseudo-Boolean optimization problem (QPBO):

$$\underset{\mathbf{m}}{\operatorname{argmax}} \quad \mathbf{m}^T \mathbf{Q} \mathbf{m}, \quad \mathbf{m} \in \{0,1\}, \tag{3}$$

where $\mathbf{m}$ is a binary indicator vector that indicates whether the model (hypothesis) was selected or not. The diagonal terms of the matrix $\mathbf{Q}$ represents the hypothesis likelihoods (cost benefits for specific hypothesis) reduced by a constant penalty ε_1 that enforces sparse solutions:

$$q_{nn} = -\varepsilon_1 + \sum_{D_i^t \in H_n^{0:k}} \left((1-\varepsilon_2) + \varepsilon_2 \cdot \mathcal{S}(D_i^t | H_n^{0:k}) \right). \tag{4}$$

Here, D_i^t represent the supporting detections of the hypothesis $H_n^{0:k}$ and $\mathcal{S}(\cdot)$ is the likelihood of the detection belonging to the hypothesis. With off-diagonal entries we model interactions between hypotheses:

$$q_{mn} = -0.5 \cdot \Bigg(\overbrace{\varepsilon_3 \cdot O(H_n^{0:k}, H_m^{0:k})}^{\text{Physical overlap penalty}} + \underbrace{\sum_{D_i^t \in H_n^{0:k} \cap H_m^{0:k}} \left((1-\varepsilon_2) + \varepsilon_2 \mathcal{S}\left(D_i^t | H^*\right) \right)}_{\text{Avoiding double-counting of inlier contributions}} \Bigg), \tag{5}$$

where $H^* \in \{H_m, H_n\}$ is the weaker hypothesis. $O(\cdot,\cdot)$ measures the physical overlap of the hypotheses and the second term corrects for double-counting detections that are consistent with both hypotheses. Model parameter ε_2 is the minimal score of the inlier detections and ε_3 weights penalization of the physical overlap.

In our formulation, we use the GOPs $\mathbf{\Omega}_i^t$ instead of the detections D_i^t and introduce a *shape model* of the unknown object to the tracking process. Physical overlap between the competing hypotheses is computed as a Bhattacharyya coefficient of the two 2D occupancy histograms of their shape representations. The histograms are computed by sampling 3D points from the hypotheses shape representations and projecting them to the ground plane.

B. Object-Trajectory Hypothesis Generation

The basic unit of our tracker is the *object-trajectory hypothesis* $H_n^{0:k}$, that spans over the frames $0 \dots k$:

$$H_n^{0:k} = \left[I_n^{0:k}, M_n^{0:k}, A_n^{0:k}, S_n^{0:k} \right], \tag{6}$$

where I_n represent the inlier GOP set of the n^{th} hypothesis, M_n is the motion model, A_n the appearance model and S_n is the 3D shape model. Note, that an *object-trajectory hypothesis* does not only hypothesize the trajectory but also the object's shape. This is a fundamental difference compared to the original QPBO tracking.

Hypothesis Generation. Following the QPBO tracking approach, the first step is to generate an over-complete set of hypotheses. In each frame, we extend the old hypothesis set using the new GOP set by running a forward Kalman filter. We start a new hypotheses from the new GOPs that were not used for extending old hypotheses by running the Kalman filter backwards. At each Kalman filter step we perform nearest neighbor data association within the validation volume of $\mathbf{C}_{i,3D}$, selecting inlier GOPs of past frames by evaluating the GOP association probability.

Data Association. We compute GOP $\mathbf{\Omega}_i^t$ association probability as (we omit the indices $0:k$ to reduce clutter):

$$p\left(\mathbf{\Omega}_i^t | H_n\right) = p\left(\mathbf{\Omega}_i^t | A_n\right) \cdot p\left(\mathbf{\Omega}_i^t | M_n\right) \cdot p\left(\mathbf{\Omega}_i^t | S_n\right). \tag{7}$$

As appearance model we compute the Bhattacharyya distance between the trajectory RGB color histogram A_n and GOP color histogram $\mathbf{h}_i^t$:

$$p\left(\mathbf{\Omega}_i^t | A_n\right) = \sum_{r,g,b} \sqrt{1 - \mathbf{h}_i^t(r,g,b) \cdot A_n(r,g,b)}. \tag{8}$$

For motion model we assume a constant-velocity Kalman filter with the following state vector:

$$\mathbf{x}^k = \left[x^k, y^k, \dot{x}^k, \dot{y}^k \right]^T, \tag{9}$$

where $\left[x^k, y^k\right]^T$ represent the 2D position on the ground plane and $\left[\dot{x}^k, \dot{y}^k\right]^T$ the velocity. Given the predicted state $\mathbf{x}^k$ and GOP $\mathbf{\Omega}_i^t$, we get the motion model probability as:

$$p\left(\mathbf{\Omega}_i^t | M_n\right) = e^{-\frac{1}{2}\left(\mathbf{p}_i^t - \left[x^k, 0, y^k\right]^T\right)\mathbf{C}^{-1}\left(\mathbf{p}_i^t - \left[x^k, 0, y^k\right]^T\right)}, \tag{10}$$

where $\mathbf{C} = \mathbf{C}_{i,3D} + \mathbf{C}_{sys}$, $\mathbf{C}_{sys}$ is the system uncertainty of the Kalman filter. The shape model is evaluated by:

$$p\left(\mathbf{\Omega}_i^t | S_n\right) = e^{-\alpha \cdot d_J BB^{2D} - \beta \cdot d_J BB^{3D}}, \tag{11}$$

where $d_J BB^{2D}$ is is the Jaccard distance (defined as $1 - J(\cdot,\cdot)$) between the 2D bounding boxes of the (integrated) hypothesis shape representation and the GOP. These bounding boxes are computed by projecting the associated 3D points to the camera image plane. $d_J BB^{3D}$ is the Jaccard

	Object	Road	Building	Tree	Bush	Sign	Pole	Sky	Grass	Dirt	Average
Jaccard	69.30	92.64	81.53	73.30		8.18		79.76	27.05		74.11
Acc.	91.52	95.21	89.17	79.98		9.14		89.30	64.39		61.68

TABLE I

JACCARD SCORE & CLASS-ACCURACY FOR OUR 7 CLASSES.

distance between their 3D bounding boxes and α, β are the weighting factors for both terms.

Finally, the fit of the GOP $\mathbf{\Omega}_i^t$ to the hypothesis $\mathcal{S}\left(H_n^{0:k}\right)$ is evaluated as:

$$\mathcal{S}\left(\mathbf{\Omega}_i^t | H_n^{0:k}\right) = e^{-\left(\frac{k-t}{\tau}\right)} \cdot p\left(\mathbf{\Omega}_i^t | H_n^{0:k}\right) \cdot p\left(\mathbf{\Omega}_i^t\right). \quad (12)$$

The term $p\left(\mathbf{\Omega}_i^t\right) = e^{-\gamma(1-r_i^t)}$ is the GOP prior computed from the GOP stability score r_i^t. The final score of the hypothesis $\mathcal{S}\left(H_n^{0:k}\right)$ is a summation over its inlier GOP scores, weighted by temporal decay. The parameter τ regulates the extent of temporal decay and γ regulates the influence of the GOP prior.

C. Shape Model Measurement Integration

Our tracker relies on raw 3D depth estimates for the computation of GOP associations and selection costs. Because individual stereo-based 3D measurements are very imprecise, we integrate 3D measurements of inlier GOPs $I_n^{0:k}$ over time to create a stable 3D representation of the hypotheses. We continuously build hypothesis shape representations $S_n^{0:k}$ by integrating the GOP measurements in a voxel grid and computing occupancy probabilities of the voxel grid cells. We perform integration in a two-step procedure: first, we reconstruct the point cloud representation of the integrated model, second, we register model points with associated inlier GOP points S_i^t and update the shape model $S_n^{0:k}$ with new measurements.

Model Initialization. We initialize the model by centering a fixed-size regular voxel grid at the center of mass of the first inlier GOP of the hypothesis $H_n^{0:k}$ and initialize each voxel grid cell $c_j \in S_n^{0:k}$ with $p\left(c_j^0\right)$, the probability that a measured point falls into the cell (normalized count of the points falling into the cell).

Model Update. To update the shape model $S_n^{0:k}$ with new GOP measurements S_i^t we center the voxel grid representation of the integrated model to the last position (world coordinates) of the hypothesis $H_n^{0:k}$ and reconstruct points with the highest occupancy probability along the camera ray. We align the shape model $S_n^{0:k}$ to the new measurement S_i^t using weighted Iterative Closest Point (ICP) algorithm. For efficient updates, we consider cells c_j independent and use a Binary Bayes Filter to update occupancy probabilities of each cell [36]. The state transition model applies an exponential decay towards the uniform distribution.

VII. EXPERIMENTAL EVALUATION

In this section we conduct a series of experiments to first evaluate the individual stages of our approach and then assess overall performance. As a test bed we use the well known KITTI dataset [1]. All experiments are performed on the KITTI tracking training set. As we perform general object tracking and do not single out specific classes, the standard evaluation pipeline on the KITTI test set is not suitable for our approach. All methods evaluated in the remaining of the paper do not use the training set as input. This enables us to use it as a valid test bed.

A. Semantic Segmentation

To show the validity of our segmentation algorithm itself, we compare our approach to three recent baselines [21], [22], [20] which each provide ground truth annotations for a different set of images and semantic categories within the KITTI [1] dataset. Only an approximate comparison can be provided, as the approaches use different depth maps and thus label slightly different parts of the image. Ladický et al. [21] even estimate a dense semantic map without depth information, whereas our method provides semantic labels only for image pixels with a corresponding depth estimate. However, even with this rough comparison, Table II shows that our semantic segmentation obtains competitive results.

Segmentation Dataset. For our complete pipeline, we trained our semantic segmentation classifier on a total of 203 annotated images extracted across the KITTI odometry dataset (we will publicly release this data upon publication). In our annotations we labeled the following classes: *building*, *car*, *curb*, *grass/dirt*, *person*, *pole*, *road*, *sky*, *sidewalk*, *sign*, *surface marking*, *tree/bush* and *wall*. For our approach we group *person* and *car* into a single *object* class.

For the remaining pipeline, the semantic segmentation is used as an initial step to filter out regions which are unlikely to belong to an object. Therefore, its main goal is to be able to distinguish between object and non-object regions, rather than separating (non-)object classes. While our annotated dataset contains a total 13 object categories, we merge them into object and non-object classes for evaluation. We qualitatively and quantitatively found that better results can be obtained by using more than only two classes for training. We believe that this is the result of reducing the intra class variance. In practice, *curb*, *sidewalk* and *surface marking* were merged into the *road* class. We also joined *wall* with *building*, and *pole* with *sign*. Table I shows both the class accuracy and Jaccard scores for these classes.

B. Object Proposal Generation

In Fig. 5 we compare our generic object proposal generation method with two relevant baselines. Difference-of-Normals (DoN) [17] demonstrated excellent results on KITTI 3D laser data [1]; EdgeBoxes [6] is a state-of-the-art appearance-based object proposal generation method (as shown in [19]). The code of both methods is publicly available. We use default parameters for EdgeBoxes [6] and their pre-trained edge detection model. For DoN we used the specified parameters from [17].

Fig. 5 *(left)* shows that our method requires 2 orders of magnitude fewer proposals than EdgeBoxes [6] to cover roughly $\sim 70\%$ of the relevant targets (annotated in KITTI).

	Building	Car	Fence	Grass	Obstacle	Pole	Road	Sidewalk	Sign	Sky	Tree	Global	Average
Ladický [21]	87.2	88.9	**39.4**	**69.9**	*N/A*	**28.5**	83.2	**76.5**	*N/A*	**91.6**	84.6	**82.4**	**72.2**
Our approach	**90.53**	**93.29**	20.66	62.90		0.00	**89.25**	48.03		70.10	**87.05**	82.07	62.4
Ros [22]	84.3	*N/A*	**62.9**	*N/A*	*N/A*	2.1	**96.8**	**75.2**	17.1	*N/A*	92.8	51.2	**61.6**
Our approach	**88.48**		6.83			**12.91**	88.23	53.53	**48.64**		**94.43**	**81.26**	56.15
Xu [20]	*N/A*	*N/A*	*N/A*	**81.1**	81.6	*N/A*	89.1	*N/A*	*N/A*	81.6	**94.4**	-	86.6
Our approach				75.99	**90.13**		**94.53**			**90.13**	89.87	91.57	**88.87**

TABLE II

CLASS-ACCURACY COMPARISON TO OTHER APPROACHES. WE TRAIN OUR APPROACH ON THE DIFFERENT SEMANTIC ANNOTATIONS. OUR RESULTS ARE AVERAGED OVER 5 RUNS AND GRAY CELLS REPRESENT CLASSES NOT REPRESENTED IN A DATASET.

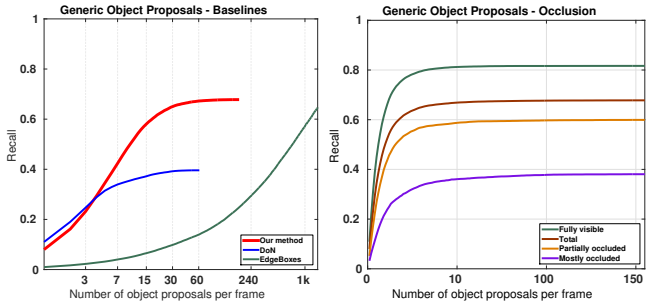

Fig. 5. GOP Recall. Left: Comparison of proposal generation method and two baselines, Difference-of-Normals [17] and EdgeBoxes [6]. Right: Recall per occlusion.

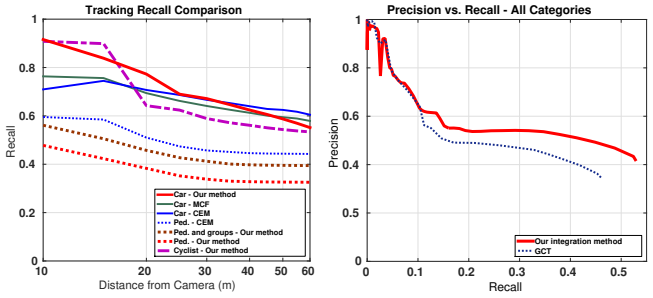

Fig. 7. Left: Tracking recall compared to two baselines [5], [2] on *pedestrian* and *car* categories. Right: Precision vs. recall of our method for all categories in KITTI, using our voxel-grid based and the GCT based integration [9].

With 30 object proposals per frame, DoN has a similar saturation point as our method, but achieves only $\sim 40\%$ recall. Fig. 5 *(right)* shows the recall of our method under varying amounts of occlusion. As can be seen, our approach achieves good recall for the mostly visible objects. For partially occluded objects, our method reports 2D bounding boxes only spanning the visible area, while the KITTI annotations cover the whole object (even if it is not actually visible). As our method is not aware of object categories, no class-specific size heuristics can be applied.

EdgeBoxes [6] does not require depth data, but needs too many proposals to be applicable to our problem. We observed that DoN [17] produces very relevant and compact proposals, but only in the close camera range.

C. Tracking

In this section we demonstrate competitive performance on *car* and *pedestrian* categories compared to other state-of-the-art detection-based approaches on the KITTI tracking dataset [1]. We will show that our proposed tracks include the categories annotated in KITTI.

Evaluation of tracking performance of our approach is non-trivial as we do not have category knowledge for the tracked objects. This means that we do not know if a trajectories represents, *e.g.*, a car or pedestrian; it is just a generic object. Especially the category-specific precision metrics become meaningless, as the confidence in a tracked object does not rely on its category!

We compare to two state-of-the-art tracking-by-detection methods [2], [5], for which we obtained tracking results from the authors. Fig. 7 *(left)* shows a frame-level recall evaluation for *cars* and *pedestrians* as a function of the distance from the cameras. In short camera-range (25m) we outperform the other methods in terms of recall, while they achieve a higher recall in the limit. In case of pedestrian tracking the state-of-the-art method [2] outperforms our method by about 13% points. We observed that this performance difference originates from the fact that we are simply not able to distinguish between individual pedestrians at the tracking level. Already at the proposal level, proposals for pedestrians walking close together are ranked higher, as the free-space surrounds the groups surpasses the free-space around individuals. Again, this is due to the fact that our tracker has no category-related knowledge. In order to validate this effect, we also plot the performance when changing the annotations, such that annotated pedestrians walking very close together are merged into a single hypothesis (See Fig. 7, left). To further show generalization to novel classes, we also report recall for the cyclist class in Fig. 7 (left).

In Fig. 7 *(right)* we show a full precision-recall curve for all annotated objects in KITTI based on the assumption that those annotations can be used as a proxy for all valid objects (in reality, not all objects are not annotated). Our approach can track about $\sim$50% of all annotated objects in a distance range of up to 30m. Experimentally the voxelgrid-based integration method turned out to be more robust for tracking than the GCT approach [9]. This experiment also demonstrates the importance of robust shape integration. Qualitative results are shown in Fig. 6.

VIII. CONCLUSIONS

In this paper, we investigated how far we can get with a generic object tracking approach. In particular, we proposed a novel tracking pipeline with the key feature of tracking multiple objects simultaneously without explicitly learning a classifier for each category. This is an important step towards better scene understanding, where it is impossible to learn class specific knowledge for everything interesting.

Fig. 6. Qualitative results on the KITTI tracking training set. Left: Semantic segmentation results. The label colors are shown in the color map at the bottom. Middle: Generic Object Proposals. Right: Tracking Results. The static objects are visualized with the gray bounding boxes.

We do not aim to replace detector-based tracking methods, but believe that an optimal tracking approach should combine the strengths of both paradigms, which we plan to address in future work. Towards our goal of general object tracking, we proposed a competitive semantic segmentation algorithm, a novel multi-scale object proposal generation stage, that reaches high recall with few proposals, and a 3D tracker that achieves competitive results for close-range objects.

Acknowledgments: This work was funded by ERC Starting Grant project CV-SUPER (ERC-2012-StG-307432). We would like to thank Dennis Mitzel for helpful discussions.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.
[2] A. Milan, S. Roth, and K. Schindler, "Continuous Energy Minimization for Multitarget Tracking," *PAMI*, vol. 36, no. 1, pp. 58–72, 2014.
[3] H. Pirsiavash, D. Ramanan, and C. C.Fowlkes, "Globally-optimal Greedy Algorithms for Tracking a Variable Number of Objects," in *CVPR*, 2011.
[4] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian Multi-object Tracking Using Motion Context from Multiple Objects," in *WACV*, 2015.
[5] L. Zhang, L. Yuan, and R. Nevatia, "Global Data Association for Multi-Object Tracking Using Network Flows," in *CVPR*, 2008.
[6] C. L. Zitnick and P. Dollár, "Edge Boxes: Locating Object Proposals from Edges," in *ECCV*, 2014.
[7] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," *PAMI*, vol. 34, no. 11, pp. 2189–2202, 2012.
[8] R. Kaestner, J. Maye, Y. Pilat, and R. Siegwart, "Generative Object Detection and Tracking in 3D Range Data," in *ICRA*, 2012.
[9] D. Mitzel and B. Leibe, "Taking Mobile Multi-Object Tracking to the Next Level: People, Unknown Objects, and Carried Items," in *ECCV*, 2012.
[10] A. Petrovskaya and S. Thrun, "Model Based Vehicle Detection and Tracking for Autonomous Urban Driving," *Autonomous Robots*, vol. 26, pp. 123–139, 2009.
[11] A. Teichman and S. Thrun, "Tracking-based semi-supervised learning," *IJRR*, vol. 31, no. 7, pp. 804–818, 2012.
[12] D. Held, J. Levinson, S. Thrun, and S. Savarese, "Combining 3D Shape, Color, and Motion for Robust Anytime Tracking," in *RSS*, 2014.
[13] A. Bewley, V. Guizilini, F. Ramos, and B. Upcroft, "Online Self-Supervised Multi-Instance Segmentation of Dynamic Objects," in *ICRA*, 2014.
[14] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M. Meinecke, "Stereo-Camera-Based Urban Environment Perception Using Occupancy Grid and Object Tracking," *TITS*, vol. 13, no. 1, pp. 154–165, 2012.
[15] D. Beymer and K. Kurt, "Real-time tracking of multiple people using continuous detection," in *IEEE Frame Rate Workshop*, 1999.
[16] D. Z. Wang, I. Posner, and P. Newman, "What Could Move? Finding Cars, Pedestrians and Bicyclists in 3D Laser Data," in *ICRA*, 2012.
[17] Y. Ioannou, B. Taati, R. Harrap, and M. A. Greenspan, "Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds," in *3DIMPVT*, 2012.
[18] M. Bansal, B. Matei, H. Sawhney, S.-H. Jung, and J. Eledath, "Pedestrian Detection with Depth-guided Structure Labeling," in *ICCV Workshops*, 2009.
[19] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" in *BMVC*, 2014.
[20] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denoeux, "Information Fusion on Oversegmented Images: An Application for Urban Scane Understanding," in *MVA*, 2013.
[21] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling Things out of Perspective," in *CVPR*, 2014.
[22] G. Ros, A. Bakhtiary, S. Ramos, D. Vazqueuez, M. Granados, and A. M. Lopez, "Vision-based Offline-Online Perception Paradigm for Autonomous Driving," in *WACV*, 2015.
[23] A. Geiger, M. Roser, and R. Urtasun, "Efficient Large-Scale Stereo Matching," in *ACCV*, 2010.
[24] G. Heitz and D. Koller, "Learning Spatial Context: Using Stuff to Find Things," in *ECCV*, 2008.
[25] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d Reconstruction in Real-time," in *Intel. Vehicles Symp.'11*, 2011.
[26] J. Papon, A. Abramov, M. Schoeler, and F. Wrgtter, "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds," in *CVPR*, 2013.
[27] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *IJCV*, vol. 81, no. 1, pp. 2–23, 2009.
[28] D. Munoz, N. Vandapel, and M. Hebert, "Onboard Contextual Classification of 3-D Point Clouds with Learned High-order Markov Random Fields," in *ICRA*, 2009.
[29] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[30] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials," in *NIPS*, 2011.
[31] A. Vedaldi and S. Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in *ECCV*, 2008.
[32] A. P. Witkin, "Scale-Space Filtering: A New Approach To Multi-Scale Description," in *ICASSP*, 1984.
[33] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
[34] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool, "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles," *PAMI*, vol. 30, no. 10, pp. 1683–1698, 2008.
[35] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, "Multi-person Tracking with Sparse Detection and Continuous Segmentation," in *ECCV*, 2010.
[36] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.