Stereo Vision-based Semantic 3D Object and Ego-motion Tracking for Autonomous Driving

Peiliang Li[0000−0001−5839−8777], Tong Qin[0000−0002−0994−9816], and
Shaojie Shen[0000−0002−5573−2909]

Hong Kong University of Science and Technology
{pliap,tong.qin,eeshaojie}@ust.hk

Abstract. We propose a stereo vision-based approach for tracking the camera ego-motion and 3D semantic objects in dynamic autonomous driving scenarios. Instead of directly regressing the 3D bounding box using end-to-end approaches, we propose to use the easy-to-labeled 2D detection and discrete viewpoint classification together with a light-weight semantic inference method to obtain rough 3D object measurements. Based on the object-aware-aided camera pose tracking which is robust in dynamic environments, in combination with our novel dynamic object bundle adjustment (BA) approach to fuse temporal sparse feature correspondences and the semantic 3D measurement model, we obtain 3D object pose, velocity and anchored dynamic point cloud estimation with instance accuracy and temporal consistency. The performance of our proposed method is demonstrated in diverse scenarios. Both the ego-motion estimation and object localization are compared with the state-of-of-the-art solutions.

Keywords: Semantic SLAM, 3D Object Localization, Visual Odometry

1 Introduction

Localizing dynamic objects and estimating the camera ego-motion in 3D space are crucial tasks for autonomous driving. Currently, these objectives are separately explored by end-to-end 3D object detection methods [1, 2] and traditional visual SLAM approaches [3–5]. However, it is hard to directly employ these approaches for autonomous driving scenarios. For 3D object detection, there are two main problems: 1. end-to-end 3D regression approaches need lots of training data and require heavy workload to precisely label all the object boxes in 3D space and 2. the instance 3D detection produces frame-independent results, which are not consistent enough for continuous perception in autonomous driving. To overcome this, we propose a light-weight semantic 3D box inference method depending only on 2D object detection and discrete viewpoint classification (Sect. 4). Comparing with directly 3D regression, the 2D detection and classification task are easy to train, and the training data can be easily labeled with only 2D images. However, the proposed 3D box inference is also frame-independent and conditional on the instance 2D detection accuracy. In

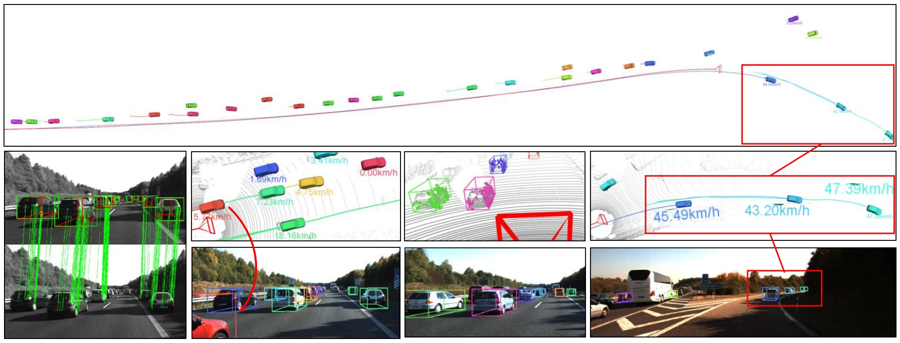


Fig. 1. Overview of our semantic 3D object and ego-motion tracking system. Top: 3D trajectories of ego-camera and all objects in the long travel history. Bottom: From left to right: Stereo feature matching for each object (Sect. 5). An extreme car-truncated case where our system can still track the moving car accurately. Dynamic 3D sparse feature recovered by our object BA. Consistent movement and orientation estimation.

another aspect, the well-known SLAM approaches can track the camera motion accurately due to precise feature geometry constraints. Inspired by this, we can similarly utilize the sparse feature correspondences for object relative motion constraining to enforce temporal consistency. However, the object instance pose cannot be obtained with pure feature measurement without semantic prior. To this end, due to the complementary nature of semantic and feature information, we integrate our instance semantic inference model and the temporal feature correlation into a tightly-coupled optimization framework which can continuously track the 3D objects and recover the dynamic sparse point cloud with instance accuracy and temporal consistency, which can be overviewed in Fig. 1. Benifitting from object-region-aware property, our system is able to estimate camera pose robustly without being affected by dynamic objects. Thanks to the temporal geometry constraints, we can track the objects continuously even for the extremely truncated case (see Fig. 1), where the object pose is hard for instance inference. Additionally, we employ a kinematics motion model for detected cars to ensure consistent orientation and motion estimation; it also serves as important smoothing for distant cars which have few feature observation. Depending only on a mature 2D detection and classification network [6], our system is capable of performing robust ego-motion estimation and 3D object tracking in diverse scenarios. The main contributions are summarized as follows:

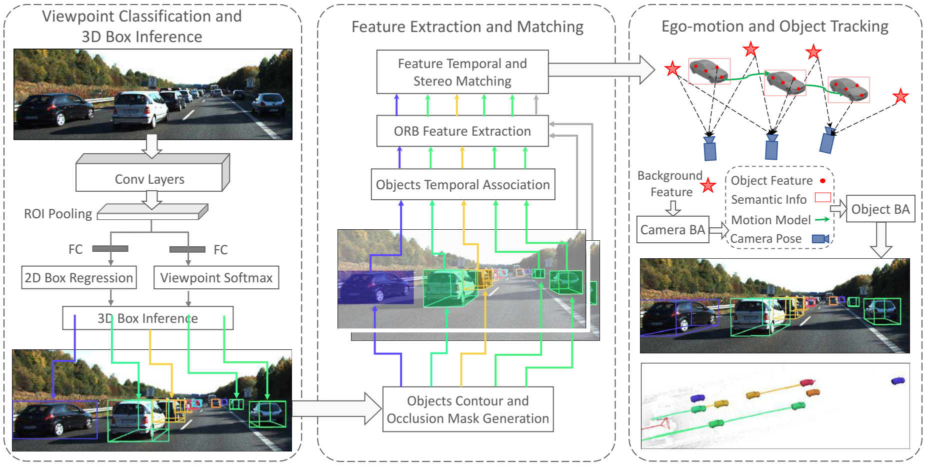
- A light-weight 3D box inference method using only 2D object detection and the proposed viewpoint classification, which provides the object reprojection contour and occlusion mask for object feature extraction. It also serves as the semantic measurement model for the follow-up optimization.
- A novel dynamic object bundle adjustment approach which tightly couples the semantic and feature measurements to continuously track the object states with instance accuracy and temporal consistency.
- Demonstration over diverse scenarios to show the practicability of the proposed system.

2 Related Work

We review the related works in the context of semantic SLAM and learning-based 3D object detection from images.

Semantic SLAM With the development of 2D object detection, several joint SLAM with semantic understanding works have sprung up, which we discuss in three categories. The first is semantic-aided localization: [7, 8] focus on correcting the global scale of monocular Visual Odometry (VO) by incorporating object metric size of only one dimension into the estimation framework. Indoor with small objects and outdoor experiments are conducted respectively in these two works. [9] proposes an object data association method in a probabilistic formulation and shows its drift correcting ability when re-observing the previous objects. However, it omits the orientation of objects by treating the 2D bounding boxes as points. And in [10], the authors address the localization task from only object observation in a prior semantic map by computing a matrix permanent. The second is SLAM-aided object detection [11, 12] and reconstruction [13, 14]: [11] develops an 2D object recognition system which is robust to viewpoint changing with the assistance of camera localization, while [12] performs confidence-growing 3D objects detection using visual-inertial measurements. [13, 14] reconstruct the dense surface of 3D object by fusing the point cloud from monocular and RGBD SLAM respectively. Finally, the third category is joint estimation for both camera and object poses: With pre-built bags of binary words, [15] localizes the objects in the datasets and correct the map scale in turns. In [16, 17], the authors propose a semantic structure from motion (SfM) approach to jointly estimate camera, object with considering scene components interaction. However, neither of these methods shows the ability to solve dynamic objects, nor makes full use of 2D bounding box data (center, width, and height) and 3-dimensions object size. There are also some existing works [18–21] building the dense map and segmenting it with semantic labels. These works are beyond the scope of this paper, so we will not discuss them in details.

3D Object Detection Inferring object pose from images by deep learning approaches provides an alternative way to localize 3D objects. [22, 23] use the shape prior to reason 3D object pose, where the dense shape and wireframe models are used respectively. In [24], a voxel pattern is employed to detect 3D pose of objects with specific visibility patterns. Similarly to object proposal approaches in 2D detection [6], [1] generates 3D proposals by utilizing depth information calculated from stereo images, while [2] exploits the ground plane assumption and additional segmentation features to produce 3D candidates; R-CNN is then used for candidates scoring and object recognition. Such high-dimension features used for proposal generating or model fitting are computationally complex for both training and deploying. Instead of directly generating 3D boxes, [25] regresses object orientation and dimensions in separate stages; then the 2D-3D


Viewpoint Classification and 3D Box Inference
Conv Layers
ROI Pooling
FC
FC
2D Box Regression
Viewpoint Softmax
3D Box Inference
Feature Extraction and Matching
Feature Temporal and Stereo Matching
ORB Feature Extraction
Objects Temporal Association
Objects Contour and Occlusion Mask Generation
Ego-motion and Object Tracking
Background Feature
Object Feature
Semantic Info
Motion Model
Camera Pose
Camera BA
Object BA

(a) Sect. 4 (b) Sect. 5 (c) Sect. 6

Fig. 2. Our whole semantic tracking system architecture.

box geometry constraints are used to calculate the 3D object pose, while purely depending on instance 2D box limits its performance in object-truncated cases.

In this work, we study the pros and cons of existing works and propose an integrated perception solution for autonomous driving that makes full use of the instance semantic prior and precise feature spatial-temporal correspondences to achieve robust and continuous state estimation for both the ego-camera and static or dynamic objects in diverse environments.

3 Overview

Our semantic tracking system has three main modules, as illustrated in Fig. 2. The first module performs 2D object detection and viewpoint classification (Sect. 4), where the objects poses are roughly inferred based on the constraints between 2D box edges and 3D box vertexes. The second module is feature extraction and matching (Sect. 5). It projects all the inferred 3D boxes to the 2D image to get the objects contour and occlusion masks. Guided feature matching is then applied to get robust feature associations for both stereo and temporal images. In the third module (Sect. 6), we integrate all the semantic information, feature measurements into a tightly-coupled optimization approach. A kinematics model is additionally applied to cars to get consistent motion estimation.

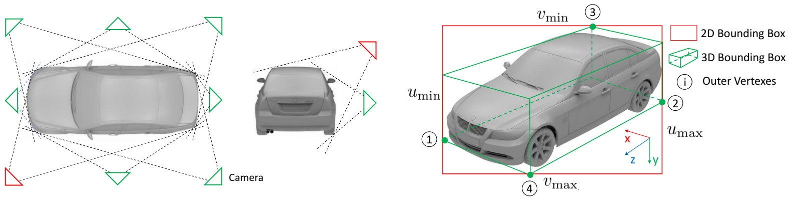
4 Viewpoint Classification and 3D Box Inference

Our semantic measurement includes the 2D object box and classified viewpoints. Based on this, the object pose can be roughly inferred instantly in close-form.

4.1 Viewpoint Classification

2D Object detection can be implemented by the state-of-the-art object detectors such as Faster R-CNN [6], YOLO [26], etc. We use Faster R-CNN in our system since it performs well on small objects. Only left images are used for object detection due to real-time requirement. The network architecture is illustrated in Fig. 2 (a). Instead of the pure object classification in the original implementation of [6], we add sub-categories classification in the final FC layers, which denotes object horizontal and vertical discrete viewpoints. As Fig 3(a) shown, We divide the continuous object observing angle into eight horizontal and two vertical viewpoints. With total 16 combinations of horizontal and vertical viewpoint classification, we can generate associations between edges in the 2D box and vertexes in the 3D box based on the assumption that the reprojection of the 3D bounding box will tightly fit the 2D bounding box. These associations provide essential condition to build the four edge-vertex constraints for 3D box inference (Sect. 4.2) and formulate our semantic measurement model (Sect. 6.2).

Comparing with direct 3D regression, the well-developed 2D detection and classification networks are more robust over diverse scenarios. The proposed viewpoint classification task is easy to train and have high accuracy, even for small and extreme occluded objects.



(a) (b)

Fig. 3. (a) presents all the horizontal and vertical viewpoints for our classification, their combinations are enough to cover all the observation cases in autonomous scenarios. (b) illustrates the 3D car in a specific viewpoint, where the object frame, four vertexes corresponding to four 2D box edges are denoted respectively.

4.2 3D Box Inference Based on Viewpoint

Given the 2D box described by four edges in normalized image plane $[u_{\min}, v_{\min}, u_{\max}, v_{\max}]$ and classified viewpoint, we aim to infer the object pose based on four constriants between 2D box edges and 3D box vertexes, which is inspired by [25]. A 3D bounding box can be represented by its center position $\mathbf{p} = [p_x, p_y, p_z]^T$ and horizontal orientation θ respecting to camera frame and the dimensions prior $\mathbf{d} = [d_x, d_y, d_z]^T$. For example, in such a viewpoint presented in Fig. 3(b) from one of 16 combinations in Fig. 3(a) (denoted as red), four vertexes are projected to the 2D edges, the corresponding constraints can be formulated as:

$$\begin{cases} u_{\min} = \pi\left(\mathbf{p} + \mathbf{R}_\theta \mathbf{C}_1 \mathbf{d}\right)_u, \; u_{\max} = \pi\left(\mathbf{p} + \mathbf{R}_\theta \mathbf{C}_2 \mathbf{d}\right)_u, \\ v_{\min} = \pi\left(\mathbf{p} + \mathbf{R}_\theta \mathbf{C}_3 \mathbf{d}\right)_v, \; v_{\max} = \pi\left(\mathbf{p} + \mathbf{R}_\theta \mathbf{C}_4 \mathbf{d}\right)_v, \end{cases} \tag{1}$$

where π is a 3D projection warp function which defined as $\pi(\mathbf{p}) = [p_x/p_z, p_y/p_z]^T$, and $(\cdot)_u$ represents the u coordinate in the normalized image plane. We use $\mathbf{R}_\theta$ to denote the rotation parameterized by horizontal orientation θ from the object frame to the camera frame. $\mathbf{C}_{1:4}$ are four diagonal selection matrixes to describe the relations between the object center to the four selected vertexes, which can be determined after we get the classified viewpoint without ambiguous. From the object frame defined in Fig. 3(b), it's easy to see that:

$$\mathbf{C}_{1:4} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}, \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}, \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}. \quad (2)$$

With these four equations, the 4 DoF object pose can be solved intuitively given the dimensions prior. This solving process has very trivial time consuming comparing with [25] which exhaustive tests all the valid edge-vertex configurations.

We convert the complex 3D object detection problem into 2D detection, viewpoint classification, and straightforward closed-form calculation. Admittedly, the solved pose is an approximated estimation which is conditioned on the instance "tightness" of the 2D bounding box and the object dimension prior. Also for some top view cases, the reprojection of the 3D box does not strictly fit the 2D box, which can be observed from the top edge in Fig. 3(b). However, for the almost horizontal or slight looking-down viewpoints in autonomous driving scenarios, this assumption can be held reasonably. Note that our instance pose inference is only for generating object projection contour and occlusion mask for the feature extraction (Sect. 5) and serves as an initial value for the follow-up maximum-a-posteriori (MAP) estimation, where the 3D object trajectory will be further optimized by sliding window based feature correlation and object point cloud alignment.

5 Feature Extraction and Matching

We project the inferred 3D object boxes (Sect. 4.2) to the stereo images to generate a valid 2D contour. As Fig. 2 (b) illustrates, we use different colors mask to represent visible part of each object (gray for the background). For occlusion objects, we mask the occluded part as invisible according to objects 2D overlap and 3D depth relations. For truncated objects which have less than four valid edges measurements thus cannot be inferred by the method in Sec. 4.2, we directly project the 2D box detected in the left image to the right image. We extract ORB features [27] for both the left and right image in the visible area for each object and the background.

Stereo matching is performed by epipolar line searching. The depth range of object features are known from the inferred object pose, so we limit the search area to a small range to achieve robust feature matching. For temporal matching, we first associate objects for successive frames by 2D box similarity score voting. The similarity score is weighted by the center distance and shape similarity of the 2D boxes between successive images after compensating the camera rotation.

The object is treated as lost if its maximum similarity score with all the objects in the previous frame is less than a threshold. We note that there are more sophisticated association schemes such as probabilistic data association [9], but it is more suitable for avoiding the hard decision when re-visiting the static object scene than for the highly dynamic and no-repetitive scene for autonomous driving. We subsequently match ORB features for the associated objects and background with the previous frame. Outliers are rejected by RANSAC with local fundamental matrix test for each object and background independently.

6 Ego-motion and Object Tracking

Beginning with the notation definition, we use $\mathbf{s}_{\mathrm{k}}^t = \{\mathbf{b}_{kl}^t, \mathbf{b}_{kr}^t, l_k, \mathbf{C}_{k1:4}^t\}$ to denote the semantic measurement of the k^{th} object at time t, where $\mathbf{b}_{kl}^t, \mathbf{b}_{kr}^t$ are the observations of the left-top and the right-bottom coordinates of the 2D bounding box respectively, l_k is the object class label and $\mathbf{C}_{k1:4}^t$ are four selection matrixes defined in Sec. 4.2. For measurements of sparse feature which is anchored to one object or the background, we use ${}^n\mathbf{z}_k^t = \{{}^n\mathbf{z}_{kl}^t, {}^n\mathbf{z}_{kr}^t\}$ to denote the stereo observations of the n^{th} feature on the k^{th} object at time t ($k = 0$ for the static background), where ${}^n\mathbf{z}_{kl}^t, {}^n\mathbf{z}_{kr}^t$ are feature coordinates in the normalized left and right image plane respectively. The states of the ego-camera and the k^{th} object are represented as ${}^w\mathbf{x}_c^t = \{{}^w\mathbf{p}_c^t, {}^w\mathbf{R}_c^t\}$, ${}^w\mathbf{x}_{ok}^t = \{{}^w\mathbf{p}_{ok}^t, \mathbf{d}_k, {}^w\theta_{ok}^t, v_{ok}^t, \delta_{ok}^t\}$ respectively, where we use ${}^w(\cdot)$, $(\cdot)_c$ and $(\cdot)_o$ to denote the world, camera and object frame. ${}^w\mathbf{p}$ represents the position in the world frame. For objects orientation, we only model the horizontal rotation ${}^w\theta_{ok}^t$ instead of $\mathbb{SO}(3)$ rotation ${}^w\mathbf{R}_c^t$ for the ego-camera. $\mathbf{d}_k$ is the time-invariant dimensions of the k^{th} object, and v_{ok}^t, δ_{ok}^t are the speed and steering angle, which are only estimated for cars. For conciseness, we visualize the measurements and states in Fig. 4 at the time t.

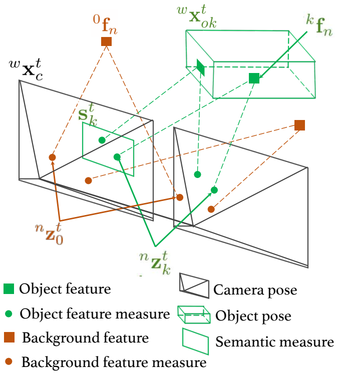


Fig. 4. Notation visualization.

Considering a general autonomous driving scene, we aim to continuously estimate the ego-motion of the onboard camera from time 0 to T: ${}^w\mathcal{X}_c = \{{}^w\mathbf{x}_c^t\}_{t=0:T}$, and track the K_t number of 3D objects: ${}^w\mathcal{X}_o = \{{}^w\mathbf{x}_{ok}\}_{k=1:K_t}$, ${}^w\mathbf{x}_{ok} = \{{}^w\mathbf{x}_{ok}^t\}_{t=0:T}$, and recover the 3D position of the dynamic sparse features: $\mathcal{F} = \{{}^k\mathbf{f}\}_{k=0:K_t}$, ${}^k\mathbf{f} = \{{}^k\mathbf{f}_n\}_{n=0:N_k}$, (note that here we use ${}^k(\cdot)$ to denote the k^{th} object frame, in which the features are relatively static, $k = 0$ for background world, in which the features are globally static), given the semantic measurements: $\mathcal{S} = \{\mathbf{s}_k\}_{k=1:K_t}$, $\mathbf{s}_k = \{\mathbf{s}_k^t\}_{t=0:T}$ and sparse feature observations anchored to the k^{th} object: $\mathcal{Z} = \{\mathbf{z}_k\}_{k=0:K_t}$, $\mathbf{z}_k = \{{}^n\mathbf{z}_k\}_{n=0:N_k}$.${}^n\mathbf{z}_k = \{{}^n\mathbf{z}_k^t\}_{t=0:T}$. We formulate our semantic objects and camera ego-motion tracking from the probabilistic model to a nonlinear optimization problem.

6.1 Ego-motion Tracking

Given the static background feature observation, the ego-motion can be solved via maximum likelihood estimation (MLE):

$$
\begin{aligned}
{}^w\mathcal{X}_c, {}^0\mathbf{f} &= \underset{{}^w\mathcal{X}_c, {}^0\mathbf{f}}{\arg\max} \prod_{n=0}^{N_0}\prod_{t=0}^{T} p({}^n\mathbf{z}_0^t | {}^w\mathbf{x}_c^t, {}^0\mathbf{f}_n, {}^w\mathbf{x}_c^0) && (3)\\
&= \underset{{}^w\mathcal{X}_c, {}^0\mathbf{f}}{\arg\max} \sum_{n=0}^{N_0}\sum_{t=0}^{T} \log p({}^n\mathbf{z}_0^t | {}^w\mathbf{x}_c^t, {}^0\mathbf{f}_n, {}^w\mathbf{x}_c^0) && (4)\\
&= \underset{{}^w\mathcal{X}_c, {}^0\mathbf{f}}{\arg\min} \sum_{n=0}^{N_0}\sum_{t=0}^{T} \left\| r_{\mathcal{Z}}({}^n\mathbf{z}_0^t,\ {}^w\mathbf{x}_c^t,\ {}^0\mathbf{f}_n) \right\|^2_{{}^0\sum_n^t}. && (5)
\end{aligned}
$$

This is the typical SLAM or SfM approach. The camera pose and background point cloud are estimated conditionally on the first state. As Eq. 3 shows, the log probability of measurement residual is proportional to its Mahalanobis norm $\|\mathbf{r}\|^2_{\sum} = \mathbf{r}^T\sum^{-1}\mathbf{r}$. Then the MLE is converted to a nonlinear least square problem, this process is also known as Bundle Adjustment(BA).

6.2 Semantic Object Tracking

After we solve the camera pose, the object state at each time t can be solved based on the dimension prior and the instance semantic measurements (Sect 4.2). We assume the object is a rigid body, which means the feature anchored to it is fixed respecting to the object frame. Therefore, the temporal states of the object are correlated if we have continuous object feature observations. States of different objects are conditionally independent given the camera pose, so we can track all the objects in parallel and independently. For the k^{th} object, we have the dimension prior distribution $p(\mathbf{d}_k)$ for each class label. We assume the detection results and feature measurements for each object at each time are independent and Gaussian distributed. According to Bayes' rule, we have the following maximum-a-posteriori (MAP) estimation:

$$
\begin{aligned}
{}^w\mathbf{x}_{ok}, {}^k\mathbf{f} &= \underset{{}^w\mathbf{x}_{ok}, {}^k\mathbf{f}}{\arg\max}\, p({}^w\mathbf{x}_{ok}, {}^k\mathbf{f} \mid {}^w\mathbf{x}_c, \mathbf{z}_k, \mathbf{s}_k) && (6)\\
&= \underset{{}^w\mathbf{x}_{ok}, {}^k\mathbf{f}}{\arg\max}\, p(\mathbf{z}_k, \mathbf{s}_k | {}^w\mathbf{x}_c, {}^w\mathbf{x}_{ok}, {}^k\mathbf{f}) p(\mathbf{d}_k) && (7)\\
&= \underset{{}^w\mathbf{x}_{ok}, {}^k\mathbf{f}}{\arg\max}\, p(\mathbf{z}_k | {}^w\mathbf{x}_c, {}^w\mathbf{x}_{ok}, {}^k\mathbf{f}) p(\mathbf{s}_k | {}^w\mathbf{x}_c, {}^w\mathbf{x}_{ok}) p(\mathbf{d}_k) && (8)
\end{aligned}
$$

$$
= \underset{{}^w\mathbf{x}_{ok}, {}^k\mathbf{f}}{\arg\max} \prod_{t=0}^{T}\prod_{n=0}^{N_k} p({}^n\mathbf{z}_k^t | {}^w\mathbf{x}_c^t, {}^w\mathbf{x}_{ok}^t, {}^k\mathbf{f}_n) p(\mathbf{s}_k^t | {}^w\mathbf{x}_c^t, {}^w\mathbf{x}_{ok}^t) p({}^w\mathbf{x}_{ok}^{t-1} | {}^w\mathbf{x}_{ok}^t) p(\mathbf{d}_k). \tag{9}
$$

Similar to Eq. 3, we convert the MAP to a nonlinear optimization problem:

$$
\begin{aligned}
{}^{w}\mathbf{x}_{ok}, {}^{k}\mathbf{f} = &\mathop{\arg\min}_{{}^{w}\mathbf{x}_{ok}, {}^{k}\mathbf{f}} \left\{ \sum_{t=0}^{T} \sum_{n=0}^{N_k} \left\| r_{\mathcal{Z}}({}^{n}\mathbf{z}_{k}^{t},\ {}^{w}\mathbf{x}_{c}^{t}, {}^{w}\mathbf{x}_{ok}^{t}, {}^{k}\mathbf{f}_{n}) \right\|_{{}^{k}\sum_{n}^{t}}^{2} + \left\| r_{\mathcal{P}}(d_{k}^{l}, \mathbf{d}_{k}) \right\|_{\sum^{l}}^{2} \right. \\
&\left. + \sum_{t=1}^{T} \left\| r_{\mathcal{M}}({}^{w}\mathbf{x}_{ok}^{t},\ {}^{w}\mathbf{x}_{ok}^{t-1}) \right\|_{\sum_{k}^{t}}^{2} + \sum_{t=0}^{T} \left\| r_{\mathcal{S}}(\mathbf{s}_{k}^{t},\ {}^{w}\mathbf{x}_{c}^{t}, {}^{w}\mathbf{x}_{ok}^{t}) \right\|_{\sum_{k}^{t}}^{2} \right\}, \qquad (10)
\end{aligned}
$$

where we use $r_{\mathcal{Z}}$, $r_{\mathcal{P}}$, $r_{\mathcal{M}}$, and $r_{\mathcal{S}}$ to denote the residual of the feature reprojection, dimension prior, object motion model, and semantic bounding box reprojection respectively. $\sum$ is the corresponding covariance matrix for each measurement. We formulate our 3D object tracking problem into a dynamic object BA approach which makes fully exploit object dimension and motion prior and enforces temporal consistency. Maximum a posteriori estimation can be achieved by minimizing the sum of the Mahalanobis norm of all the residuals.

Sparse Feature Observation We extend the projective geometry between static features and camera pose to dynamic features and object pose. Based on anchored relative static features respecting to the object frame, the object poses which share feature observations can be connected by a factor graph. For each feature observation, the residual can be represented by the reprojection error of predicted feature position and the actual feature observations on the left and right image:

$$
\begin{aligned}
&r_{\mathcal{Z}}({}^{n}\mathbf{z}_{k}^{t},\ {}^{w}\mathbf{x}_{c}^{t}, {}^{w}\mathbf{x}_{ok}^{t}, {}^{k}\mathbf{f}_{n}) && (11)\\
&= \begin{bmatrix} \pi\left(h^{-1}({}^{w}\mathbf{x}_{c}^{t},\ h({}^{w}\mathbf{x}_{ok}^{t}, {}^{k}\mathbf{f}_{n}))\right) - {}^{n}\mathbf{z}_{kl}^{t} \\ \pi\left(h({}^{r}\mathbf{x}_{l}, h^{-1}({}^{w}\mathbf{x}_{c}^{t},\ h({}^{w}\mathbf{x}_{ok}^{t}, {}^{k}\mathbf{f}_{n})))\right) - {}^{n}\mathbf{z}_{kr}^{t} \end{bmatrix}, && (12)
\end{aligned}
$$

where we use $h(\mathbf{x}, p)$ to denote applying a 3D rigid body transform $\mathbf{x}$ to a point p. For example, $h\left({}^{w}\mathbf{x}_{ok}^{t}, {}^{k}\mathbf{f}_{n}\right)$ transforms the n^{th} feature point ${}^{k}\mathbf{f}_{n}$ from the object frame to the world frame, $h^{-1}(\mathbf{x}, p)$ is the corresponding inverse transform. ${}^{r}\mathbf{x}_{l}$ denotes the extrinsic transform of the stereo camera, which is calibrated offline.

Semantic 3D Object Measurement Benefiting from the viewpoint classification, we can know the relations between edges of the 2D bounding box and vertexes of the 3D bounding box. Assume the 2D bounding box is tightly fitted to the object boundary, then each edge is intersected with a reprojected 3D vertex. These relations can be determined as four selection matrixes for each 2D edge. The semantic residual can be represented by the reprojection error of the predicted 3D box vertexes with the detected 2D box edges:

$$
\begin{aligned}
&r_{\mathcal{S}}(\mathbf{s}_{k}^{t},\ {}^{w}\mathbf{x}_{c}^{t}, {}^{w}\mathbf{x}_{ok}^{t}, \mathbf{d}_{k}) = \begin{bmatrix} \pi\left(h_{\mathbf{C}_1}\right)_{u} - (\mathbf{b}_{kl}^{t})_{u} \\ \pi\left(h_{\mathbf{C}_2}\right)_{u} - (\mathbf{b}_{kr}^{t})_{u} \\ \pi\left(h_{\mathbf{C}_3}\right)_{v} - (\mathbf{b}_{kl}^{t})_{v} \\ \pi\left(h_{\mathbf{C}_4}\right)_{v} - (\mathbf{b}_{kr}^{t})_{v} \end{bmatrix}, && (13)\\
&h_{\mathbf{C}_i} = h^{-1}({}^{w}\mathbf{x}_{c}^{t},\ h({}^{w}\mathbf{x}_{ok}^{t}, \mathbf{C}_{i}\mathbf{d}_{k}^{l})), && (14)
\end{aligned}
$$

where we use $h_{\mathbf{C}_i}$ to project a vertex specified by the corresponding selection matrix $\mathbf{C}_i$ of the 3D bounding box to the camera plane. This factor builds the connection between the object pose and its dimensions instantly. Note that we only perform 2D detection on the left image due to the real-time requirement.

Vehicle Motion Model To achieve consistent estimation of motion and orientation for the vehicle class, we employ the kinematics model introduced in [28]. The vehicle state at time t can be predicted with the state at $t-1$:

$$ {}^w\hat{\mathbf{x}}^t_{ok} = \begin{bmatrix} {}^w\mathbf{p}^t_{ok} \\ {}^w\theta^t_{ok} \\ \delta^t_{ok} \\ v^t_{ok} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3\mathrm{x}3}, & \mathbf{0}, & \mathbf{0}, & \boldsymbol{\Lambda} \\ \mathbf{0}, & 1, & 0, & \frac{\tan(\delta)\Delta \mathrm{t}}{L} \\ \mathbf{0}, & 0, & 1, & 0 \\ \mathbf{0}, & 0, & 0, & 1 \end{bmatrix} \begin{bmatrix} {}^w\mathbf{p}^{t-1}_{ok} \\ {}^w\theta^{t-1}_{ok} \\ \delta^{t-1}_{ok} \\ v^{t-1}_{ok} \end{bmatrix}, \boldsymbol{\Lambda} = \begin{bmatrix} \cos(\theta)\Delta \mathrm{t} \\ \sin(\theta)\Delta \mathrm{t} \\ 0 \end{bmatrix}, \qquad (15)$$

$$ r_{\mathcal{M}}({}^w\mathbf{x}^t_{ok}, {}^w\mathbf{x}^{t-1}_{ok}) = {}^w\mathbf{x}^t_{ok} - {}^w\hat{\mathbf{x}}^t_{ok}, \qquad (16)$$

where L is the length of the wheelbase, which can be parameterized by the dimensions. The orientation of the car is always parallel to the moving direction. We refer readers to [28] for more derivations. Thanks to this kinematics model, we can track the vehicle velocity and orientation continuously, which provides rich information for behavior and path planning for autonomous driving. For other class such as pedestrians, we directly use a simple constant-velocity model to enhance smoothness.

Point Cloud Alignment After minimizing all the residuals, we obtain the MAP estimation of the object pose based on the dimension prior. However, the pose might be biased estimated due to object size difference (See Fig. 5). We therefore align the 3D box to the recovered point cloud, which is unbiased because of accurate stereo extrinsic calibration. We minimize the distance of all 3D points with their anchored 3D box surfaces:

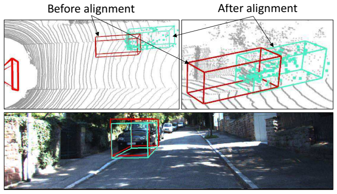


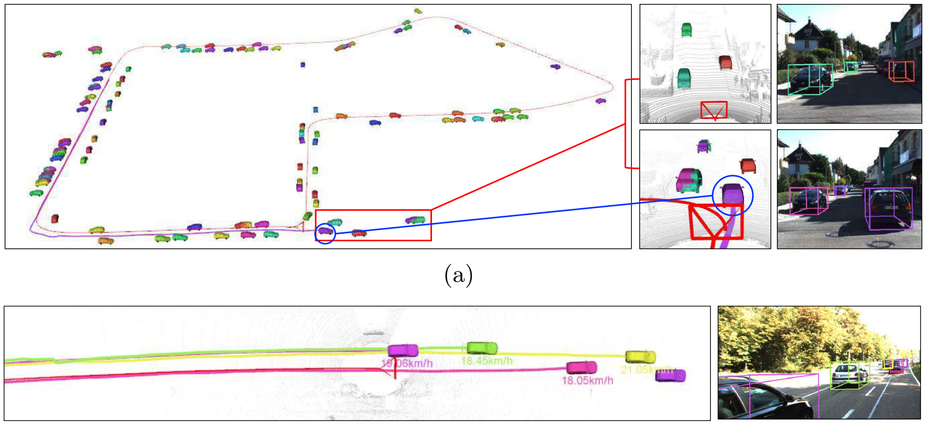
Fig. 5. Point cloud alignment.

$$ {}^w\mathbf{x}^t_{ok} = \underset{{}^w\mathbf{x}_{ok}}{\arg\min} \sum_{n=0}^{N_k} d({}^w\mathbf{x}^t_{ok}, {}^k\mathbf{f}_n), \qquad (17)$$

where $d({}^w\mathbf{x}^t_{ok}, {}^k\mathbf{f}_n)$ denotes the distance of the k^{th} feature with its corresponding observed surface. After all the above information is tightly fused together, we get consistent and accurate pose estimation for both the static and dynamic objects.

7 Experimental Results

We evaluate the performance of the proposed system on both KITTI [29, 30] and Cityscapes [31] dataset over diverse scenarios. The mature 2D detection and classification module has good generalization ability to run on unseen scenes, and



(a)

(b)

Fig. 6. Continuous tracking results over long trajectories. (a) shows a roughly 700 m close-loop trajectory including both static and dynamic cars. The right top and right bottom are enlarged start and end views respectively. The car in the blue circle is tracked over 200 meters, the trajectory of which can be found in the left top view. (b) shows a scenario which mainly contains dynamic and truncated cars. The estimated trajectory, velocity and reprojected 2D image are presented in left and right respectively. Note that the LiDAR point cloud is only for reference in all the top views.

the follow-up nonlinear optimization is data-independent. Our system is therefore able to perform consistent results on different datasets. The quantitative evaluation shows our semantic 3D object and ego-motion tracking system has better performance than the isolated state-of-the-art solutions.

7.1 Qualitative Results Over Diverse Scenarios

Firstly, we test the system on long challenging trajectories in KITTI dataset which contains 1240 × 376 stereo color and grayscale images captured at 10 Hz. We perform 2D detection on left color images and extract 500 (for the background) and 100 (for the object) ORB features [27] on both left and right grayscale images. Fig. 6(a) shows a 700 m close-loop trajectory which includes both static and dynamic cars. We use red cone and line to represent the camera pose and trajectory, and various color CAD models and lines to represent different cars and their trajectories, all the observed cars are visualized in the top view. Currently, our system performs object tracking in a memoryless manner, so the re-observed object will be treated as a new one, which can also be found in the enlarged start and end views in Fig. 6(a). In Fig. 6(b), the black car is continuously truncated over a long time, which is an unobservable case for instance 3D box inference (Sect. 4.2). However, we can still track its pose accurately due to temporal feature constraints and dynamic point cloud alignment.

We also demonstrate the system performance on different datasets over more scenarios which include concentrated cars, crossroads, and dynamic roads. All the reprojected images and the corresponding top views are shown in Fig. 7.

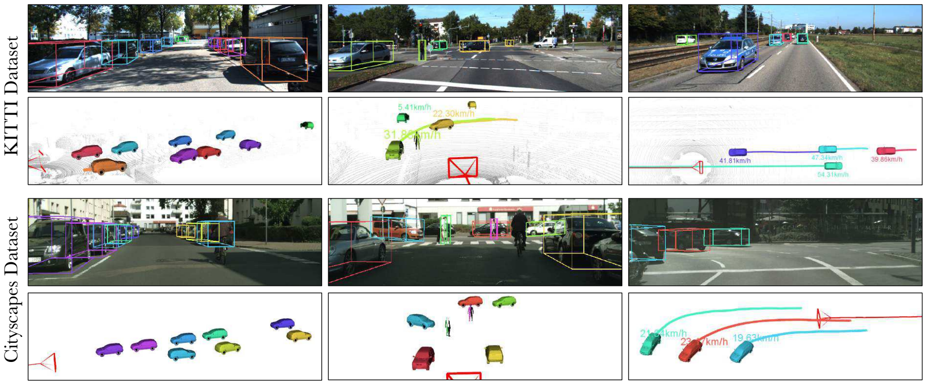
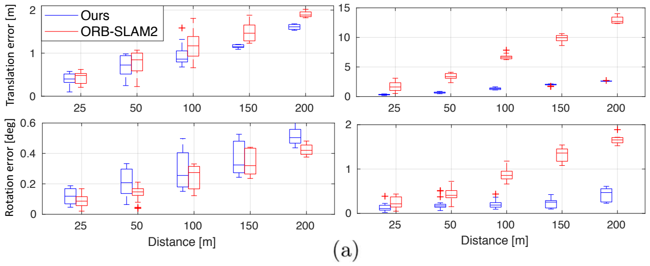


Fig. 7. Qualitative examples over diverse scenarios. From left to right: Concentrated cars. Crossroads which include both cars and pedestrians (note that we do not solve orientation for pedestrians), Dynamic cars. The top two rows are the results on the KITTI dataset, and the bottom two rows show the results on the Cityscapes dataset.

7.2 Quantitative Evaluation

Since there are no available integrated academic solutions for both ego-motion and dynamic objects tracking currently, we conduct quantitative evaluations for the camera and objects poses by comparing with the isolated state-of-the-art works: ORB-SLAM2 [4] and 3DOP [1].



(a)

Sequence	ORB-SLAM2(m)	Ours(m)
1003_0047	19.35	**1.03**
0929_0004	1.07	**0.40**
0926_0009	1.31	**1.14**
0926_0013	0.51	**0.35**
0926_0014	0.74	**0.51**
0926_0051	**0.72**	0.76
0926_0101	6.32	**5.30**
0926_0009	1.31	**1.14**

(b)

Fig. 8. (a) RPEs comparison. Left and right are the results of 0929_0004 and 1003_0047 sequences from the KITTI raw dataset respectively. (b) RMSEs of ATE comparisons on ten long KITTI raw sequences.

Camera Pose Evaluation Benefiting from the semantic prior, our system can perform robust camera estimation in dynamic environments. We evaluate the accuracy of camera odometry by comparing the relative pose error (RPE) [29] and RMSE of ATE (Absolute Trajectory Error) [32] with the ORB-SLAM2 [4] with stereo settings. Two sequences in KITTI raw dataset: 0929_0004 and 1003_0047 which include dynamic objects are used for RPEs comparison. The relative translation and rotation errors are presented in Fig. 8 (a). Ten long sequences of KITTI raw dataset are additionally used to evaluate RMSEs of ATE, as detailed in Fig. 8 (b). It can be seen that our estimation shows almost same accuracy with [4] in less dynamic scenarios due to the similar Bundle Adjustment approaches (0926_0051, etc.). However, our system still works well in high dynamic environments while ORB-SLAM2 shows non-trivial errors due to introducing many outliers (1003_0047, 0929_0004, etc.). This experiment shows that the semantic-aided object-aware property is essential for camera pose estimation, especially for dynamic autonomous driving scenarios.

Object Localization Evaluation We evaluate the car localization performance on KITTI tracking dataset since it provides sequential stereo images with labeled objects 3D boxes. According to occlusion level and 2D box height, we divide all the detected objects into three regimes: easy, moderate and hard then evaluate them separately. To evaluate the localization accuracy of the proposed estimator, we collect objects average position error statistics. By setting series of Intersection-over-Unions (IoU) thresholds from 0 to 1, we calculate the true positive (TP) rate and the average error between estimated position of TPs and ground truth at each instance frame for each threshold. The average position error (in %) vs TP rate curves are shown in Fig. 9, where we use blue, red, yellow lines to represent statistics for easy, moderate and hard objects. It can be seen that the average error for half tuth positive objects is below 5%. For all the estimated results, the average position errors are 5.9%, 6.1% and 6.3% for easy, moderate and hard objects respectively.

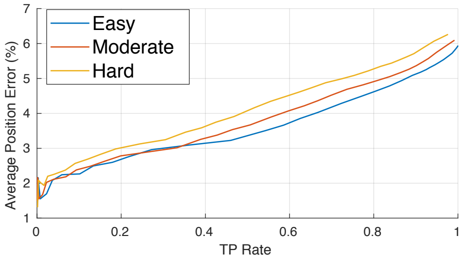


Fig. 9. Average position error vs TP rate results. We set 40 discrete IoU thresholds from 0 to 1, then count the TP rate and the average position error for the true positives for each IoU threshold.

To compare with baselines, we evaluate the Average Precision (AP) for bird's eye view boxes and 3D boxes by comparing with 3DOP [1], the state-of-the-art stereo based 3D object detection method. We set IoU thresholds to 0.25 and 0.5 for both bird's eye view and 3D boxes. Note that we use the oriented box overlap, so the object orientation is also implied evaluated in these two metrics. We use S, M, F, P to represent semantic measurement, motion model, feature observation, and point cloud alignment respectively. As listed in Table.1, the semantic measurement serves as the basis of the 3D object estimation. Adding feature observation increases the AP for easy (near) objects obviously due to large feature extraction area (same case for adding point clout alignment), while adding motion model helps the hard (far) objects since it ”smooths” the non-trivial 2D detection noise for small objects. After integrating all these cues together, we obtain accurate 3D box estimation for both near and far objects. It can be seen that our integrated method shows more accurate results for all the AP in bird's eye view and 3D box with 0.25 IoU threshold. Due to the unregressed object size, our performance slightly worse than 3DOP in 3D box comparison of 0.5 IoU. However, we stress our method can efficiently track both static and dynamic 3D objects with temporal smoothness and motion consistency, which is essential for continuous perception and planning in autonomous driving.

Table 1. Average precision (in %) of bird's eye view and 3D boxes comparison.

Method	AP_{bv}(IoU=0.25)			AP_{bv}(IoU=0.5)			AP_{3d}(IoU=0.25)			AP_{3d}(IoU=0.5)			Time (ms)
	Easy	Mode	Hard	Easy	Mode	Hard	Easy	Mode	Hard	Easy	Mode	Hard	
S	63.12	56.37	53.18	33.12	28.91	27.77	58.78	52.42	48.82	25.68	21.70	21.02	120
S+M	66.27	63.81	58.84	41.08	38.90	34.84	62.97	60.70	55.28	34.18	30.98	27.32	121
S+F	76.23	70.18	66.18	48.82	43.07	39.80	73.35	66.86	62.66	38.93	33.43	30.46	170
S+F+M	77.87	74.48	70.85	46.96	44.39	42.23	73.32	71.06	67.30	40.50	36.28	34.59	171
S+F+M+P	**88.07**	**77.83**	**72.73**	**58.52**	**46.17**	**43.97**	**86.57**	**74.13**	**68.96**	48.51	37.13	34.54	173
3DOP	81.34	70.70	66.32	54.83	43.36	37.15	80.62	70.01	65.76	**53.73**	**42.27**	**35.87**	1200

8 Conclusions and Future work

In this paper, we propose a 3D objects and ego-motion tracking system for autonomous driving. We integrate the instance semantic prior, sparse feature measurement and kinematics motion model into a tightly-coupled optimization framework. Our system can robustly estimate the camera pose without being affected by the dynamic objects and track the states and recover dynamic sparse features for each observed object continuously. Demonstrations over diverse scenarios and different datasets illustrate the practicability of the proposed system. Quantitative comparisons with state-of-the-art approaches show our accuracy for both camera estimation and objects localization.

In the future, we plan to improve the object temporal correlation by fully exploiting the dense visual information. Currently, the camera and objects tracking are implemented successively in our system. We are also going to model them into a fully-integrated optimization framework such that the estimation for both camera and dynamic objects can benefit from each other.

9 Acknowledgment

This work was supported by the Hong Kong Research Grants Council Early Career Scheme under project no. 26201616. The authors also thank Xiaozhi Chen for providing the 3DOP [1] results on the KITTI tracking dataset.

References

1. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems. (2015) 424–432
2. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2147–2156
3. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. arXiv preprint arXiv:1708.03852 (2017)
4. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics **33**(5) (2017) 1255–1262
5. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European Conference on Computer Vision, Springer (2014) 834–849
6. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
7. Frost, D.P., Kähler, O., Murray, D.W.: Object-aware bundle adjustment for correcting monocular scale drift. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE (2016) 4770–4776
8. Sucar, E., Hayet, J.B.: Probabilistic global scale estimation for monoslam based on generic object detection. In: Computer Vision and Pattern Recognition Workshops (CVPRW). (2017)
9. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic slam. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE (2017) 1722–1729
10. Atanasov, N., Zhu, M., Daniilidis, K., Pappas, G.J.: Semantic localization via the matrix permanent. In: Proceedings of Robotics: Science and Systems. Volume 2. (2014)
11. Pillai, S., Leonard, J.J.: Monocular slam supported object recognition. In: Proceedings of Robotics: Science and Systems. Volume 2. (2015)
12. Dong, J., Fei, X., Soatto, S.: Visual-inertial-semantic scene representation for 3-d object detection. arXiv preprint arXiv:1606.03968 (2016)
13. Civera, J., Gálvez-López, D., Riazuelo, L., Tardós, J.D., Montiel, J.: Towards semantic slam using a monocular camera. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE (2011) 1277–1284
14. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: Slam++: Simultaneous localisation and mapping at the level of objects. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 1352–1359
15. Gálvez-López, D., Salas, M., Tardós, J.D., Montiel, J.: Real-time monocular object slam. Robotics and Autonomous Systems **75** (2016) 435–449
16. Bao, S.Y., Savarese, S.: Semantic structure from motion. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 2025–2032
17. Bao, S.Y., Bagra, M., Chao, Y.W., Savarese, S.: Semantic structure from motion with points, regions, and objects. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2703–2710
18. Kundu, A., Li, Y., Dellaert, F., Li, F., Rehg, J.M.: Joint semantic segmentation and 3d reconstruction from monocular video. In: European Conference on Computer Vision, Springer (2014) 703–718

19. Vineet, V., Miksik, O., Lidegaard, M., Nießner, M., Golodetz, S., Prisacariu, V.A., Kähler, O., Murray, D.W., Izadi, S., Pérez, P., et al.: Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE (2015) 75–82
20. Li, X., Belaroussi, R.: Semi-dense 3d semantic mapping from monocular slam. arXiv preprint arXiv:1611.04144 (2016)
21. McCormac, J., Handa, A., Davison, A., Leutenegger, S.: Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE (2017) 4628–4635
22. Bao, S.Y., Chandraker, M., Lin, Y., Savarese, S.: Dense object reconstruction with semantic priors. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 1264–1271
23. Zia, M.Z., Stark, M., Schiele, B., Schindler, K.: Detailed 3d representations for object recognition and modeling. IEEE transactions on pattern analysis and machine intelligence **35**(11) (2013) 2608–2623
24. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Data-driven 3d voxel patterns for object category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1903–1911
25. Mousavian, A., Anguelov, D., Flynn, J., Košecká, J.: 3d bounding box estimation using deep learning and geometry. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE (2017) 5632–5640
26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 779–788
27. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE international conference on, IEEE (2011) 2564–2571
28. Gu, T.: Improved Trajectory Planning for On-Road Self-Driving Vehicles Via Combined Graph Search, Optimization & Topology Analysis. PhD thesis, Carnegie Mellon University (2017)
29. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012)
30. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
31. Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset. In: CVPR Workshop on the Future of Datasets in Vision. Volume 1. (2015) 3
32. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE (2012) 573–580