Dynamic SLAM: The Need For Speed

Mina Henein[1], Jun Zhang[1], Robert Mahony[1] and Viorela Ila[2]

***Abstract*— The static world assumption is standard in most simultaneous localisation and mapping (SLAM) algorithms. Increased deployment of autonomous systems to unstructured dynamic environments is driving a need to identify moving objects and estimate their velocity in real-time. Most existing SLAM based approaches rely on a database of 3D models of objects or impose significant motion constraints. In this paper, we propose a new feature-based, model-free, object-aware dynamic SLAM algorithm that exploits semantic segmentation to allow estimation of motion of rigid objects in a scene without the need to estimate the object poses or have any prior knowledge of their 3D models. The algorithm generates a map of dynamic and static structure and has the ability to extract velocities of rigid moving objects in the scene. Its performance is demonstrated on simulated, synthetic and real-world datasets.**

I. Introduction

SLAM is an established research field in robotics. While many accurate and efficient solutions to the problem exist, most of the existing techniques heavily rely on the static world assumption [1]. This assumption limits the deployment of existing algorithms to a wide range of increasingly important real world scenarios involving dynamic and unstructured environments. Advances in deep learning have provided algorithms that can reliably detect and segment classes of objects at almost real time [2], [3]. To incorporate such information in a geometric SLAM formulation then either a 3D-model of the object must be available [4], [5] or the front end must explicitly provide pose information in addition to detection and segmentation [6], [7], [8]. The requirement for accurate 3D-models severely limits the potential domains of application, while to the best of our knowledge, multiple object tracking and 3D pose estimation remain a challenge to learning techniques. There is a clear need for an algorithm that can exploit the powerful detection and segmentation capabilities of modern deep learning algorithms without relying on additional pose estimation or motion model priors.

In this paper, we propose a novel model-free, object-aware point-based dynamic SLAM approach that leverages image-based semantic information to simultaneously localise the robot, map the static structure, estimate a full SE(3) pose change of moving objects and build a dynamic representation of the world. We also fully exploit the rigid object motion to extract velocity information of objects in the scene (Fig. 1), an emerging task in autonomous driving which has not yet been thoroughly explored [9]. Such information is crucial to aid autonomous driving algorithms for tasks such as collision avoidance [10] and adaptive cruise control [11]. The key innovation in the paper is a novel *pose change representation* used to model the motion of a collection of points pertaining to a given rigid body and the integration of this model into a SLAM optimisation framework. The resulting algorithm is agnostic to the underlying 3D-model of the object as long as the semantic detection and segmentation of the object can be tracked. To the best of our knowledge, this is the first work able to estimate, along with the camera poses, the static and dynamic structure, the full SE(3) pose change of every rigid object in the scene, extract object velocities and be demonstrable on a real-world outdoor dataset.

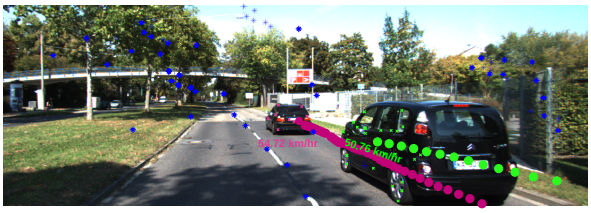


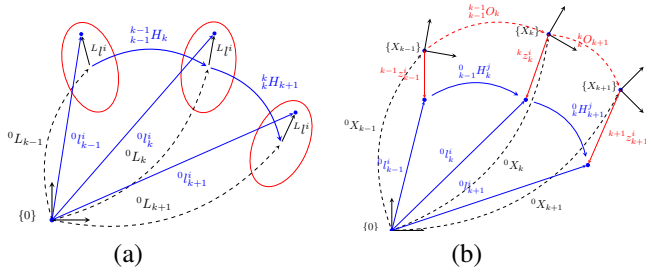
Fig. 1. Results of our object-aware dynamic SLAM on KITTI Sequence0003. Centroids of each object are obtained by applying our motion estimates to the first ground-truth object centroid. Speed estimates are also extracted for each object.

II. Related Work

Establishing the spatial and temporal relationships between a robot, stationary and moving objects in a scene serves as a basis for scene understanding [12] and the problems of simultaneous localisation, mapping and moving object tracking are mutually beneficial. In the SLAM community, information associated with stationary objects is considered positive, while information drawn from moving objects is seen as degrading the algorithm performance. SLAM systems either treat data from moving objects as outliers [13], [14], [15], [16], [17] or they track them separately using multi-target tracking [18], [19], [20], [21]. Bibby and Reid's SLAMIDE [22] estimates the state of 3D features (stationary or dynamic) with a generalised EM algorithm where they use reversible data association to include dynamic objects in a single framework SLAM. Wang *et al.* [12] developed a theory for performing SLAM with Moving Objects Tracking (SLAMMOT). In the latest version of their SLAM with detection and tracking of moving objects, the estimation problem is decomposed into two separate estimators (moving and stationary objects) to make it feasible to update both filters in real time. Kundu *et al.* [21] tackle the SLAM problem with dynamic objects by solving the problems of Structure from Motion (SfM) and tracking of

[1]Mina Henein, Jun Zhang, and Robert Mahony are with the College of Engineering and Computer Science, Australian National University, 0200 Canberra, Australia. firstname.lastname@anu.edu.au

[2]Viorela Ila is with the School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, 2006 Sydney, Australia viorela.ila@sydney.edu.au



(a) (b)

Fig. 2. (a): Coordinates of the rigid body in motion. The points ${}^{L}l^{i}$ are represented relative to the rigid body pose $\{L\}$ at each step. (b): Robot poses, moving landmark positions (black) and the measurements (red) at three different time steps.

moving objects in parallel. Reddy *et al.* [23] uses optical flow and depth to compute semantic motion segmentation. They isolate static objects from moving objects and reconstruct them independently, before using semantic constraints to improve the the 3D reconstruction. Dewan *et al.*[24] presents a model-free approach for detecting and tracking dynamic objects in 3D using LiDAR scans. Judd *et al.* [25] estimates the full SE(3) motion of both the camera and rigid objects in the scene by applying a multi-motion visual odometry (MVO) multimodel fitting technique. Although this approach does not require prior knowledge of the environment or object 3D models, they parameterise the motion transforms non-incrementally (with respect to the first observed frame) which might introduce severe linearisation errors and only show results on one lab-environment experiment, with no evaluation on any existing datasets. A very recent work by Yang and Scherer [26] presents a method for single image $3D$ cuboid detection, and multi-view object SLAM for both static and dynamic environments. Their main interest, however, is the camera pose and object detection accuracy and they provide no evaluation of the object pose estimation.

III. Accounting For Dynamic Objects In SLAM

The problem considered is one in which there are relatively large rigid objects moving within the sensing range of the robot that is undertaking the SLAM estimation. The SLAM front-end is able to identify and associate points from the same potentially moving object at different time steps. These points share an underlying motion constraint that can be exploited to improve the quality of the SLAM estimation.

A. Problem Formulation

The SLAM with dynamic objects estimation problem is modelled using factor graphs [27], and the goal is to obtain the static and dynamic $3D$ structure and the robot poses that maximally satisfy a set of measurements and pose change constraints. Assuming Gaussian noise, this problem becomes a non-linear least squares (NLS) optimisation over a set of variables [28]: the robot poses $\mathbf{x} = \{x_0 \ldots x_{n_x}\}$, with $x_k \in SE(3)$ where $k \in 0 \ldots n_x$ and n_x is the number of steps, and the 3D point features in the environment seen at different time steps: $\mathbf{l} = \{l_0^1 \ldots l_{n_x}^{n_l}\}$ where $l_k^i \in \mathbb{R}^3$ and $i \in 1 \ldots n_l$ is the unique landmark index and n_l is the total number of detected landmarks. The set of landmarks, $\mathbf{l} = \mathbf{l_s} \cup \mathbf{l_d}$, contains a set of static landmarks $\mathbf{l_s}$ and a set of moving object landmarks $\mathbf{l_d}$. The same point on a moving object is represented using a different variable at each time step, i.e. l_{k-1}^i and l_k^i are the same physical i^{th} point seen at times $k-1$ and k, respectively.

B. Motion Model Of A Point On A Rigid Body

Let $\{0\}$ denote the *reference coordinate frame*, and $\{L\}$ the coordinate frame associated to a moving rigid body. We write the pose ${}^{0}L_k \in SE(3)$ of the rigid-body with respect to the reference frame $\{0\}$. For a feature observed on an object, let ${}^{L}l^{i} \in \mathbb{R}^3$ denote the coordinates of this point in the object frame. We write ${}^{0}l_k^i$ for the coordinates of the same point expressed in the reference frame $\{0\}$ at time k. Note that for rigid bodies in motion, ${}^{L}l^{i}$ is constant for all the object instances, while both ${}^{0}L_k$ and ${}^{0}l_k^i$ are time varying. The point coordinates are related by the expression:

$$ {}^{L}\bar{l}^{i} = {}^{0}L_k^{-1}\, {}^{0}\bar{l}_k^i \tag{1} $$

where the bar indicates homogeneous coordinates.

The relative motion of the object L from $k-1$ to k is represented by a rigid-body transformation ${}^{L_{k-1}}_{k-1}H_k \in SE(3)$ called the *body-fixed frame pose change*. The indices indicate that the transformation maps a base pose ${}^{0}L_{k-1}$ (lower left index) to a target pose ${}^{0}L_k$ (lower right index), expressed in coordinates of the frame ${}^{0}L_{k-1}$ (upper left index):

$$ {}^{L_{k-1}}_{k-1}H_k = {}^{0}L_{k-1}^{-1}\, {}^{0}L_k \tag{2} $$

Fig. 2a shows this transformations for three consecutive object poses. The new rigid body coordinates are given by the incremental pose transformation:

$$ {}^{0}L_k = {}^{0}L_{k-1}\, {}^{L_{k-1}}_{k-1}H_k \tag{3} $$

Consider a point ${}^{L}l^{i}$ in the object frame $\{L\}$. Writing the expression (1) for two consecutive poses of the object at time $k-1$ and k and using the relative motion of the object in (3), the motion of this point can written as:

$$ {}^{0}\bar{l}_k^i = {}^{0}L_{k-1}\, {}^{L_{k-1}}_{k-1}H_k\, {}^{0}L_{k-1}^{-1}\, {}^{0}\bar{l}_{k-1}^i \,. \tag{4} $$

We observe that (4) relates the same point on the rigid body in motion at different time steps by a transformation ${}^{0}_{k-1}H_k = {}^{0}L_{k-1}\, {}^{L_{k-1}}_{k-1}H_k\, {}^{0}L_{k-1}^{-1}$, where ${}^{0}_{k-1}H_k \in SE(3)$. According to [29], this equation represents a frame change of a pose transformation, and shows how the body-fixed frame pose change in (2) relates to the *reference frame pose change*. The point motion in the reference frame becomes:

$$ \boxed{{}^{0}\bar{l}_k^i = {}^{0}_{k-1}H_k\, {}^{0}\bar{l}_{k-1}^i \,.} \tag{5} $$

This formulation is key to the proposed approach since it eliminates the need to estimate the object pose ${}^{0}L_k$ and allows us to work directly with points ${}^{0}\bar{l}_k^i$ in the reference frame.

Linear velocity extraction: Other vehicle velocity is a crucial piece of information in autonomous driving applications. Given vehicle's rigid body pose change in inertial frame ${}^{0}_{k-1}H_k$, its linear velocity vector in $(\frac{1}{\text{fps}}.\frac{\text{m}}{\text{s}})$ can be computed:

$$ v = {}^{0}_{k-1}t_k - (I_3 - {}^{0}_{k-1}R_k)\, c_{k-1} \tag{6} $$

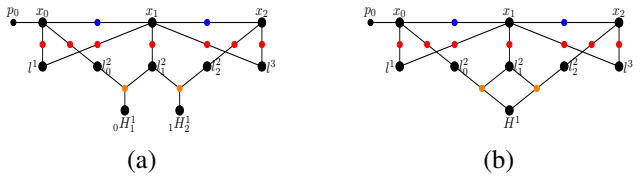


Fig. 3. Back-end (a): Factor graph representation of a problem with multiple pose change vertices for the same object. (b): Factor graph representation of a problem with a unique pose change vertex for the same object.

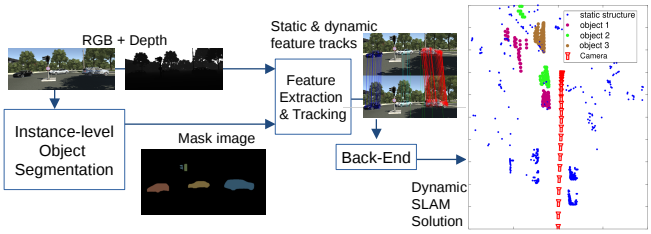


Fig. 4. System overview. Input images are used into an instance level object segmentation algorithm to provide objects masks. The algorithm then detects and tracks features on potentially moving objects. Potentially dynamic features along with tracked static features are used to build the graph, that is then fed into a back-end optimisation.

and its speed is the magnitude of this vector, where ${}^{0}_{k-1}R_k \in SO(3)$ and ${}^{0}_{k-1}t_k \in \mathbb{R}^3$ the rotation and translation components of the vehicle's pose change in inertial frame ${}^{0}_{k-1}H_k$ respectively, I_3 is the identity matrix, and c_{k-1} is the object's centroid position at time $k-1$. As our algorithm is sparse, we do not have access to the object's centroid but rather approximate it by the 3D centroid of features detected on the object. The derivation of (6) is detailed in the appendix and for more explanation, we refer the reader to [29].

C. Motion factors in dynamic SLAM

The proposed approach estimates the camera poses, the static and dynamic structure and the motion of the dynamic structure. To achieve this, motion factors along with the odometry obtained from the robot's proprioceptive sensors, and the landmarks observations are optimised jointly:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \Big\{ \sum_{k=1}^{m_k} \rho_h((h(x_k,l_k^i)-z_k^i)^\top \Sigma_{w_k}^{-1}(h(x_k,l_k^i)-z_k^i)) + \sum_{i=1}^{m_i} \rho_h((f(x_{k-1},x_k)-o_k)^\top \Sigma_{v_k}^{-1}(f(x_{k-1},x_k)-o_k)) + \sum_{i,j}^{m_s} \rho_h((g(l_{k-1}^i,l_k^i,{}^{0}_{k-1}H_k^j)^\top \Sigma_q^{-1}(g(l_{k-1}^i,l_k^i,{}^{0}_{k-1}H_k^j)\Big\} \tag{7}$$

where ρ_h is the Huber function, $h(x_k,l_k^i)$ is the 3D point measurement model with Σ_{w_k} the point measurement covariance matrix; $\mathbf{z} = \{z_1 ... z_{m_k}\} | z_k \in \mathbb{R}^3$ is the set of all m_k 3D point measurements at all time steps, $f(x_{k-1},x_k)$ is the odometry model with Σ_{v_k} odometry covariance matrix and $\mathbf{o} = \{o_1 ... o_{m_i}\}$ the set of m_i odometric measurements. Fig. 2b shows the measurements in red. $g(l_{k-1}^i, l_k^i, {}^{0}_{k-1}H_k^j)$ is the motion model of points on dynamic objects with Σ_q motion covariance matrix and m_s is the total number of motion factors. The motion of any point on a detected rigid object j can be characterised by the same pose transformation ${}^{0}_{k-1}H_k^j \in SE(3)$ given by (5) and the corresponding factor is:

$$g(l_{k-1}^i, l_k^i, {}^{0}_{k-1}H_k^j) = {}^{0}l_{k-1}^i - {}^{0}_{k-1}R_k^j \, {}^{0}l_{k-1}^i - {}^{0}_{k-1}t_k^j + q_{s_j} \tag{8}$$

where $q_s \sim \mathcal{N}(0,\Sigma_q)$ is the normally distributed zero-mean Gaussian noise. The factor in (8) is a ternary factor which we call the *motion model of a point on a rigid body* (orange factors in Fig.3). All the variables are grouped in $\boldsymbol{\theta} = \mathbf{x} \cup \mathbf{l} \cup \mathbf{H}$, where $\mathbf{H}$ is the set of all the variables characterising the objects' motions.

D. The factor graph

We model the dynamic SLAM problem as a factor graph. The factor graph formulation is highly intuitive and has the advantage that it allows for efficient implementations of batch [27] [30] and incremental [31], [32], [33] solvers. It has been shown that in dynamic SLAM, knowing the type of motion of the objects in the environment is highly valuable [12]. In this work we evaluate two scenarios without and with constant motion model :

- In city scenarios, where the objects motions are subject to changes (acceleration, deceleration, etc.) modelling the motion is challenging. Therefore, we allow for the estimation of a new pose change at every time step. Fig. 3a shows a factor graph representation of such scenario where the motion of the same object is estimated using two motions vertices for two different time transitions. A possible constraint is to minimise the change between these motion estimates.
- A highway scenario, where every vehicle maintains a constant motion. Fig. 3b shows the factor graph representation where a single motion is estimated per object.

Further we show that if the body-fixed frame pose change is constant then the reference frame pose change is constant too. For any $k-1, k'-1$ time indices, the constant motion in the body-fixed pose change is:

$${}^{L_{k-1}}_{k-1}H_k = C = {}^{L_{k'-1}}_{k'-1}H_{k'} \in SE(3)\,. \tag{9}$$

We rescale (3) and use (9) to obtain: ${}^{0}L_k = {}^{0}L_{k-1} C$ which we replace in ${}^{0}_{k-1}H_k = {}^{0}L_{k-1} C \, {}^{0}L_{k-1}^{-1}$ to obtain:

$${}^{0}_{k-1}H_k = {}^{0}L_k C \, {}^{0}L_k^{-1} = {}^{0}_{k}H_{k+1} \tag{10}$$

It follows that the reference frame pose change for a specific object j: ${}^{0}_{k-1}H_k^j = {}^{0}H^j = {}^{0}_{k'}H_{k'+1}^j \in SE(3)$ holds for any k, k' time indices and the factor in (8) is changed accordingly. The constant motion assumption can be used to handle occlusions by keeping hypothesis of previously detected objects and reviving those based on re-observations of occluded objects.

IV. System Overview

The system pipeline shown in Fig. 4 assumes a robot equipped with an RGB-D camera and proprioceptive sensors (e.g. odometers, IMU). Our feature-based object-aware dynamic SLAM back-end estimates the robot poses, the static

and dynamic structure and pose transformations for every detected object in the scene. To ensure features are being detected on moving objects, we employ an instance-level object segmentation algorithm to produce objects masks. Object segmentation constitutes an important prior in static/dynamic object classification and tracking of dynamic objects. The front-end then makes use of object masks to detect features on *potentially-moving* objects and on static background. Feature tracking is a crucial module for the success of our approach. Through object segmentation and feature tracking, the SLAM front-end is able to identify and associate points on the same rigid-body object at different time steps. These points share an underlying motion model that we exploit to achieve simultaneous localisation, mapping and moving object tracking. The algorithm does not require the front-end to estimate the objects' pose or use any geometric model of the objects. The static and dynamic 3D measurements along with the measurements from proprioceptive sensors are integrated into the back-end to simultaneously estimate the camera motion, the static and dynamic structure and the SE(3) pose transformations of detected objects in the scene.

V. Experiments And Results

A. Error Metrics

The accuracy of the solution is evaluated vs ground-truth (GT) by comparing the Relative Translational Error (RTE) in %, that is the translational component of the error between the estimated and GT robot pose changes. Similarly, the Relative Rotational Error (RRE) in $^\circ/m$ is the rotational component of the same error. We also evaluate the Relative Structure Error (RSE) in % for all static and dynamic landmarks, as the error between the corresponding relative positions of the estimated and GT structure points in the simulated experiments. We also provide an evaluation of the object pose change estimates; the Object Motion Translation Error (OMTE) in %, the Object Motion Rotational Error (OMRE) in $^\circ/m$ and for driving scenarios, the Object Motion Speed Error (OMSE) in %.

B. Virtual KITTI Dataset

Description : Virtual KITTI [34] is a photo-realistic synthetic dataset that provides RGB-D videos from a vehicle driving in an urban environment. Frames are fully annotated at the pixel level with unique object tracking identifiers (needed for errors calculations). GT information about camera and object poses is also provided which makes it a perfect dataset to test and evaluate the proposed technique.

Goal : We make use of the GT data to test the effect of each component in the front-end on the performance of the algorithm and the accuracy of the pose change estimation for the camera and moving objects in the scene. The three aspects studied are errors in: a) depth/3D point measurement, b) object segmentation, and c) feature tracking.

Implementation : Due to the fact that our algorithm is sparse-based, object pose change estimation is affected by the distribution of the extracted features on moving objects. Another important aspect is the percentage of the object mask

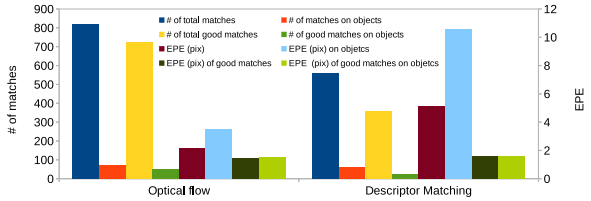


Fig. 5. Comparison of optical flow and descriptor matching for feature tracking.

in the image. In the experiments reported in this paper, we only estimate for objects whose segmentation masks amount to a certain percentage of the total image. This threshold ensures to exclude far-away and partially observed objects that are entering/exiting the camera field of view and which makes their motion estimates inaccurate. This threshold is set to 6% for vKITTI, and 2% for KITTI.

a) Depth error: We evaluate the performance of our algorithm using GT object segmentation, and feature tracking with odometry and varied point measurement noise. The noise levels added are 5% for translational odometry in each axis, and 10% for rotational odometry around each axis. Three different noise levels, drawn from a normal distribution with zero mean and standard deviation σ_1=0.02, σ_2=0.04, and σ_3=0.06 m in each axis per observation, were tested. These noise levels correspond to commercially available LiDAR system, and a stereo-camera rig respectively and a third higher value. This is conceptually the same as replacing the depth input in the front-end with a stereo depth estimation algorithm e.g. SPSS [35] or a single image depth estimation for a monocular system e.g. [36]. Point measurement noise is kept at σ_1 for further tests in this subsection.

b) Object segmentation error: This test aims at evaluating the effect of the object segmentation while using GT feature tracking with added odometry and point measurement noise. We employ MASK-RCNN [3], learning based model, for instance-level object segmentation. We perform evaluation tests of MASK-RCNN on all sequences of vKITTI and KITTI. Results for mean average precision (mAP) and mean intersection over union for predictions only (mIOU_Pred) of the 'car' class are 0.513 and 0.557 for vKITTI and 0.413 and 0.632 for KITTI. Numbers show good performance, however, testing the effect on camera and object pose change estimation is crucial.

c) Feature tracking error: As our algorithm is sparse feature-based, feature tracking is an essential component of the front-end. In order to test the effect of feature tracking, we first conduct tests on the quantity and quality of feature matches using 1) PWC-Net [37] and 2) feature descriptor matching. Fig. 5 shows the number of total and object matches and their corresponding end-point error (EPE), and then extends this test to show these values for "good matches"; matches with less than 3 pixels EPE.

Discussions : As shown in Fig. 6, the feature tracking is the most crucial component of the front-end and dictates the performance of our feature-based algorithm. Fig. 5 and 6 show better performance of optical flow over descriptor

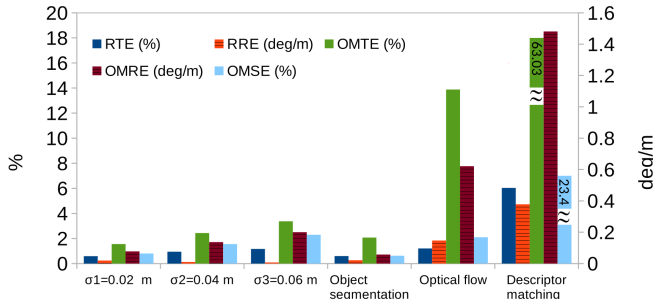


Fig. 6. Study evaluated on vKITTI of the effect of different front-end components on the camera/objects pose change estimation accuracy.

TABLE I

RESULTS OF APPLYING OUR OBJECT-AWARE DYNAMIC MOTION INTEGRATION ON A SIMULATED DATA

Error	SLAM+MOT	Ours
RTE (%)	4.426	**3.804**
RRE ($^\circ/m$)	1.34	**0.486**
RSE (%)	8.019	**4.177**
OMTE (%)	20.946	**4.018**
OMRE ($^\circ/m$)	0.349	**0.055**

matching in terms of quantity and quality of features. For the remainder of this paper, optical flow is used for feature tracking and we aim to look at improving the feature tracking in a future work by utilising the object motion estimates. Object segmentation appears to have the least effect on the estimation quality. Errors in the camera and object motion estimation due to the use of MASK-RCNN compared to GT segmentation appear to be minimal.

C. Simulated Data

Description : This experiment features a single simulated ellipsoid-shaped object tracked by a robot as it follows a circular motion in an environment with no static structure. The object is simulated to have a constant SE(3) pose change, and the estimation makes use of this piece of information to constraint the problem as explained in Subsection III-D. The simulation corresponds to a scenario where only moving structure is visible, e.g. a vehicle on a bridge or inside a tunnel occluded by other vehicles driving alongside and failing to track static structure.

Goal : This experiment is designed to show that our approach provides good solutions in cases where existing approaches to dynamic SLAM might fail. We compare our algorithm (one framework joint estimation) vs. parallel tracking and mapping, e.g. SLAM + Multiple Object Tracking (MOT) [18], [19], [20], [21]. This class of algorithms depend on the quality of the returned map, and will perform poorly in environments with insufficient number of reliable static structure such as the examples given above.

Discussions : Camera motion in the case of parallel SLAM+MOT is basically a direct integration of odometric measurements. Results in Table I show the clear advantage of our algorithm that jointly estimates the camera and rigid object pose transformations. Improvements are in the range

TABLE II

RESULTS OF APPLYING OUR OBJECT-AWARE DYNAMIC MOTION INTEGRATION ON KITTI

	Seq.007	Seq.006		Seq.0001		Seq.0003		Seq.0005		Seq.0000
Error	**Static Only**	**Static Only**	**Ours**	**Static Only**	**Ours**	**Static Only**	**Ours**	**SLAM + MOT**	**Ours**	**Ours**
RTE (%)	0.039	0.000	0.000	0.248	0.248	0.016	0.016	0.022	0.025	0.020
RRE ($^\circ/m$)	0.003	0.001	0.001	0.017	0.017	0.002	0.002	0.003	0.003	0.001
OMTE (%)	–	–	11.646	–	10.525	–	6.0/59.5	23.608	23.653	42.7/63
OMRE ($^\circ/m$)	–	–	0.254	–	0.555	–	0.2/1.0	0.472	0.473	1.2/5.0
OMSE (%)	–	–	10.587	–	5.561	–	2.5/2.7	11.809	11.809	20.7/22

of 80-85% in object pose change estimation. In an extreme case, where no static structure is observed, our algorithm not only improves the object motion estimates but also the camera pose estimation. However, in an environment with enough static structure, both algorithms yield very similar results as shown in the next section.

D. KITTI dataset

Description : KITTI [38] has been a standard benchmark suite for a number of challenging real-world computer vision tasks. We make use of the KITTI tracking dataset as it provides GT object poses in camera coordinate frame.

Goal : This experiment is designed to evaluate the performance of our algorithm on real-world challenging outdoor scenarios. In relevant dataset sequences, we compare our results with classical static only SLAM (where dynamic objects are considered outliers) and SLAM+MOT solutions.

To demonstrate the generality of the approach and the fact that the proposed framework performs well in any type of scenarios, we consider three different cases:

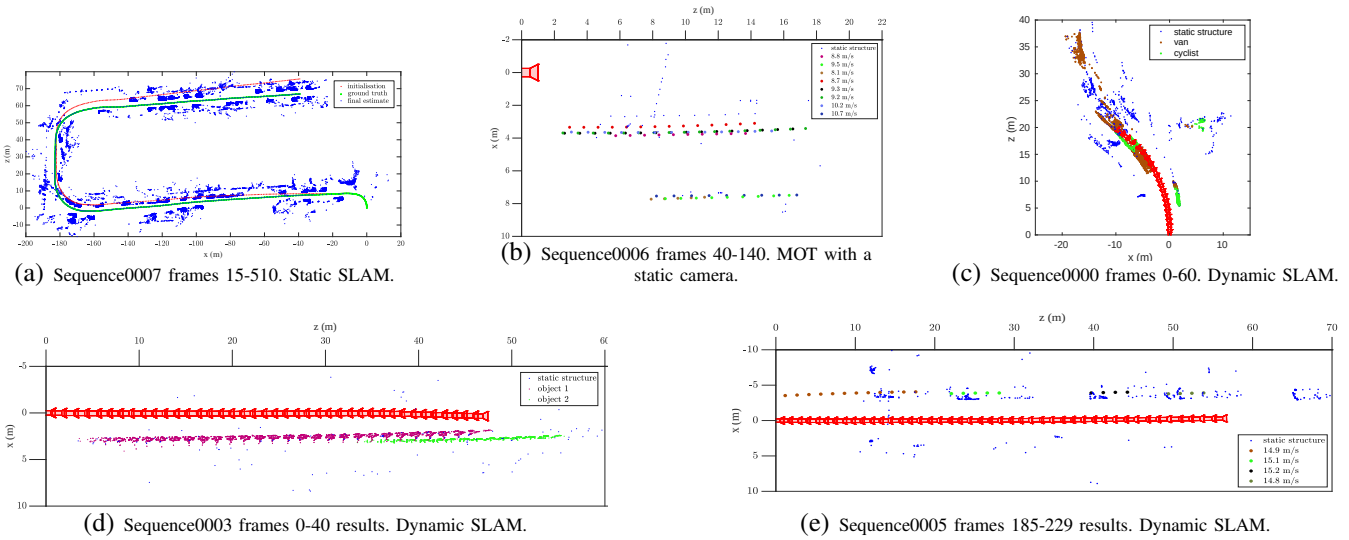
a) *Classical SLAM*: A moving robot equipped with an RGB-D camera in a static environment. Sequence0007 represents this case and shows that our algorithm performs equally well in a classical scenario with no dynamics and requires no prior knowledge or makes any prior assumptions of the scene.

b) *Multi-object tracking*: A static camera in a dynamic environment as shown in Sequence0006. For this specific data sequence, we consider a constant pose change assumption. Note that camera pose change errors for this sequence are reported in meters and degrees.

c) *Dynamic SLAM*: A moving robot equipped with an RGB-D camera in a dynamic environment. In here we do the distinction between two sub-scenarios:

• A highway scenario, represented by Sequence0005, where every vehicle is assumed to have constant motion. This allows us to constraint the problem by assuming a constant pose change model for each detected object in the scene.

• Sequence0001, Sequence0003 and Sequence0000 represent an intersection and other city driving scenarios, where motion models are difficult to impose. In here, the factor graph formulation allows for the estimation of a new pose change vertex every time step. Some insights on how to improve the estimation in such scenarios is provided in Section VI. Sequence0003 and Sequence000 contain two objects each, therefore the object motion error results are shown separated by a '/'. Sequence0000 consists of a "van" and a "cyclist"



(a) Sequence0007 frames 15-510. Static SLAM.

(b) Sequence0006 frames 40-140. MOT with a static camera.

(c) Sequence0000 frames 0-60. Dynamic SLAM.

(d) Sequence0003 frames 0-40 results. Dynamic SLAM.

(e) Sequence0005 frames 185-229 results. Dynamic SLAM.

Fig. 7. Sample results on KITTI various sequences.

which slightly violates the rigidity assumption, yet our approach still provides fairly good results.

Implementation : The three variants of SLAM: classical, SLAM+MOT, and dynamic are run using the front-end introduced in Section IV and implemented in GTSAM [39]. The tracking dataset is thought for camera-only based application, therefore GPS and IMU measurements are fused and further corrupted with noise to simulate odometric measurements available in a robotic (self-driving cars) scenario. The noise values are the same as the ones explained in Subsection V-B.0.a, except Sequence0007, where twice the noise is added. In autonomous driving, the literature normally distinguishes between different depth ranges for velocity estimation [9]: near (d $<$ 20 m), medium (20 m $\leq$ d $<$ 45 m) and far range (d $>$ 45 m). In all KITTI experiments presented here, we only consider objects $<$ 22 m of distance to the camera (near and early medium range).

Discussions : All results show high accuracy in the estimation of pose change transformations and speeds of moving objects. Results in Table II show a speed estimation accuracy in the range of 78-97.5%. The second objects in Sequence0003 and Sequence0000 are particularly hard to process. In Sequence0003, the second object only occupies a small part of the image, dominated by its wheels having a different motion than the vehicle, yet its speed estimate is reasonable. Sequence0000 consists of a van and a cyclist turning at very low speeds ($<$ 5.5 m/s). Their motion estimation is particularly hard because of association errors and the fact that a cyclist is a non-rigid object mostly formed by wheels not obeying the motion model of the object. Although speed errors seem high in percentage, they only account for an average speed error of 0.16 m/s for the van and 0.063 m/s for the cyclist.

VI. Conclusion And Future Work

In this paper we proposed a novel framework that exploits semantic information in the scene with no additional knowledge of the object pose or geometry, to achieve simultaneous localisation, mapping and tracking of dynamic objects. The algorithm shows consistent, robust and accurate results in various scenarios. Although the method presented here is applied to RGB-D/stereo images, we plan to explore semantic depth from single image or a purely monocular setup in the future. An important issue to be analysed, is the computational complexity of SLAM with dynamic objects. In long-term applications, different techniques can be applied to limit the growth of the graph [40], [41]. The estimation could be further enhanced by assuming a constant motion within a temporal window and use this assumption to handle occlusions and reduce the problem size. Another possible extension is to use the SLAM back-end estimates to improve the tracking accuracy of the front-end.

Appendix

To see that (6) is the same quantity in 3D as the translation vector from the origin of the object pose at time $\{k-1\}$ to the origin of the object pose at time $\{k\}$ as seen in $\{0\}$, we start by writing the object pose change in $\{0\}$ and substitute for ${}^{L_{k-1}}_{k-1}H_k$ by its definition in (2)

$$ {}^{0}_{k-1}H_k = {}^{0}L_{k-1}\, {}^{L_{k-1}}_{k-1}H_k\, {}^{0}L^{-1}_{k-1} = {}^{0}L_k\, {}^{0}L^{-1}_{k-1} \tag{11} $$

Assuming ${}^{0}R_{L_{k-1}} \in SO(3)$ and ${}^{0}t_{L_{k-1}} \in \mathbb{R}^3$ the rotation and translation components of ${}^{0}L_{k-1}$, and ${}^{0}R_{L_k}$, ${}^{0}t_{L_k}$ their corresponding at time k, the translation and rotation parts of ${}^{0}_{k-1}H_k$ can be expressed as ${}^{0}t_{L_k} - {}^{0}R_{L_k}\, {}^{0}R^{\top}_{L_{k-1}}\, {}^{0}t_{L_{k-1}}$ and ${}^{0}R_{L_k}\, {}^{0}R^{\top}_{L_{k-1}}$. Substituting these two quantities into (6), we get

$$ v = {}^{0}t_{L_k} - {}^{0}R_{L_k}\, {}^{0}R^{\top}_{L_{k-1}}\, {}^{0}t_{L_{k-1}} - (I_3 - {}^{0}R_{L_k}\, {}^{0}R^{\top}_{L_{k-1}})\, {}^{0}t_{L_{k-1}} \tag{12} $$

which reduces to $v = {}^{0}t_{L_k} - {}^{0}t_{L_{k-1}}$ which is the translation vector from the origin of the object pose at time $\{k-1\}$ to the origin of the object pose at time $\{k\}$ as seen in $\{0\}$.

ACKNOWLEDGMENT

This research was supported by the Australian Research Council through the "Australian Centre of Excellence for Robotic Vision" CE140100016.

REFERENCES

[1] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard, "Dynamic pose graph SLAM: Long-term mapping in low dynamic environments," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1871–1878.

[2] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV), 2017*. IEEE, 2017, pp. 2980–2988.

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013*. IEEE, 2013, pp. 1352–1359.

[5] D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel, "Real-time monocular object slam," *Robotics and Autonomous Systems*, vol. 75, pp. 435–449, 2016.

[6] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831 [cs]*, Mar. 2016, arXiv: 1603.00831. [Online]. Available: http://arxiv.org/abs/1603.00831

[7] A. Byravan and D. Fox, "Se3-nets: Learning rigid body motion using deep neural networks," in *IEEE International Conference on Robotics and Automation (ICRA), 2017*. IEEE, 2017, pp. 173–180.

[8] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3109–3118.

[9] M. Kampelmühler, M. G. Müller, and C. Feichtenhofer, "Camera-based vehicle velocity estimation from monocular video," 2018.

[10] R. Aufrère, J. Gowdy, C. Mertz, C. Thorpe, C.-C. Wang, and T. Yata, "Perception for collision avoidance and autonomous driving," *Mechatronics*, vol. 13, no. 10, pp. 1149–1161, 2003.

[11] R. K. Jurgen, "Adaptive cruise control," SAE Technical Paper, Tech. Rep., 2006.

[12] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.

[13] D. Hahnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, vol. 1. IEEE, 2002, pp. 496–501.

[14] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, vol. 2. IEEE, 2003, pp. 1557–1563.

[15] D. F. Wolf and G. S. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, 2005.

[16] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, and H. Zha, "SLAM in a dynamic large outdoor environment using a laser scanner," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.* IEEE, 2008, pp. 1455–1462.

[17] B. Bescós, J. M. Fácil, J. Civera, and J. Neira, "Dynslam: Tracking, mapping and inpainting in dynamic scenes," *arXiv preprint arXiv:1806.05620*, 2018.

[18] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, vol. 1. IEEE, 2003, pp. 842–849.

[19] I. Miller and M. Campbell, "Rao-blackwellized particle filtering for mapping dynamic environments," in *IEEE International Conference on Robotics and Automation, 2007*. IEEE, 2007, pp. 3862–3869.

[20] J. G. Rogers, A. J. Trevor, C. Nieto-Granda, and H. I. Christensen, "SLAM with expectation maximization for moveable object tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010*. IEEE, 2010, pp. 2077–2082.

[21] A. Kundu, K. M. Krishna, and C. Jawahar, "Realtime multibody visual slam with a smoothly moving monocular camera," in *IEEE International Conference on Computer Vision (ICCV), 2011*. IEEE, 2011, pp. 2080–2087.

[22] C. Bibby and I. Reid, "Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with reversible data association," in *Proceedings of Robotics: Science and Systems*, 2007.

[23] N. D. Reddy, P. Singhal, V. Chari, and K. M. Krishna, "Dynamic body vslam with semantic constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015*. IEEE, 2015, pp. 1897–1904.

[24] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3d lidar scans," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4508–4513.

[25] K. M. Judd, J. D. Gammell, and P. Newman, "Multimotion visual odometry (mvo): Simultaneous estimation of camera and third-party motions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3949–3956.

[26] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, 2019.

[27] F. Dellaert and M. Kaess, "Square Root Sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[28] L. Polok, M. Solony, V. Ila, P. Smrz, and P. Zemcik, "Efficient implementation for block matrix operations for nonlinear least squares problems in robotic applications," in *IEEE International Conference on Robotics and Automation (ICRA), 2013*. IEEE, 2013, pp. 2263–2269.

[29] G. S. Chirikjian, R. Mahony, S. Ruan, and J. Trumpf, "Pose changes from a different point of view," in *Proceedings of the ASME International Design Engineering Technical Conferences (IDETC) 2017*. ASME, 2017.

[30] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[31] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, p. 0278364911430419, 2011.

[32] L. Polok, V. Ila, M. Solony, P. Smrz, and P. Zemcik, "Incremental Block Cholesky Factorization for Nonlinear Least Squares in Robotics," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[33] V. Ila, L. Polok, M. Šolony, and P. Svoboda, "SLAM++-A highly efficient and temporally scalable incremental SLAM framework," *International Journal of Robotics Research*, vol. Online First, no. 0, pp. 1–21, 2017.

[34] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual Worlds as Proxy for Multi-Object Tracking Analysis," in *CVPR*, 2016.

[35] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *European Conference on Computer Vision*. Springer, 2014, pp. 756–771.

[36] H. Ren, M. El-khamy, and J. Lee, "Deep robust single image depth estimation neural network using scene understanding," *arXiv preprint arXiv:1906.03279*, 2019.

[37] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[39] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[40] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," in *IEEE International Conference on Computer Vision (ICCV), 2011*. IEEE, 2011, pp. 2352–2359.

[41] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact pose SLAM," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 78–93, 2010.