2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

The Importance of Coordinate Frames in Dynamic SLAM

Jesse Morris, Yiduo Wang and Viorela Ila

***Abstract*— Most Simultaneous localisation and mapping (SLAM) systems have traditionally assumed a static world, which does not align with real-world scenarios. To enable robots to safely navigate and plan in dynamic environments, it is essential to employ representations capable of handling moving objects. Dynamic SLAM is an emerging field in SLAM research as it improves the overall system accuracy while providing additional estimation of object motions. State-of-the-art literature informs two main formulations for Dynamic SLAM, representing dynamic object points in either the world or object coordinate frame. While expressing object points in their local reference frame may seem intuitive, it does not necessarily lead to the most accurate and robust solutions. This paper conducts and presents a thorough analysis of various Dynamic SLAM formulations, identifying the best approach to address the problem. To this end, we introduce a front-end agnostic framework using GTSAM [1] that can be used to evaluate various Dynamic SLAM formulations.[1]**

I. Introduction

Simultaneous localisation and mapping (SLAM) is a problem that has been studied for more than three decades [2]. SLAM systems enable robots to create representations of the environment while simultaneously localising themselves within it. Many current SLAM solutions operate with the assumption that the environment consists mostly of stationary elements [3], [4], [5], which may not hold true in real-world situations where dynamic objects are abundant.

Conventionally, SLAM systems treat sensor data associated with moving objects as outliers and reject them from the estimation process [6], [7], disregarding any useful information pertaining to dynamic objects. Integrating objects into the SLAM framework has the advantage that the resulting map can directly inform navigation and task planning systems [8], [9] of the estimated object motion and scene structure, improving robotic system robustness in complex dynamic environments [10], [11]. As such, an emerging theme in SLAM is to incorporate observations of the dynamic components of the scene and estimate their motions [2] – in this paper we refer to such a system as Dynamic SLAM.

Recently, multi-object visual odometry techniques [12], [13] and graph-based optimisation Dynamic SLAM systems [7], [14], [15] have been explored to jointly estimate the robot pose, the static structure and the motion/trajectory of rigid-body objects in the scene based on static and dynamic point observations. The literature proposes a variety of ways to formulate this optimisation problem, each of which optimises for different sets of variables and objective functions. This defines different underlying graph structures of the optimisation. The choice of formulation significantly affects the robustness, accuracy and efficiency of SLAM systems [2]. Therefore it is paramount to conduct formal analysis on different formulations to clearly delineate the circumstance that leads to the best performance.

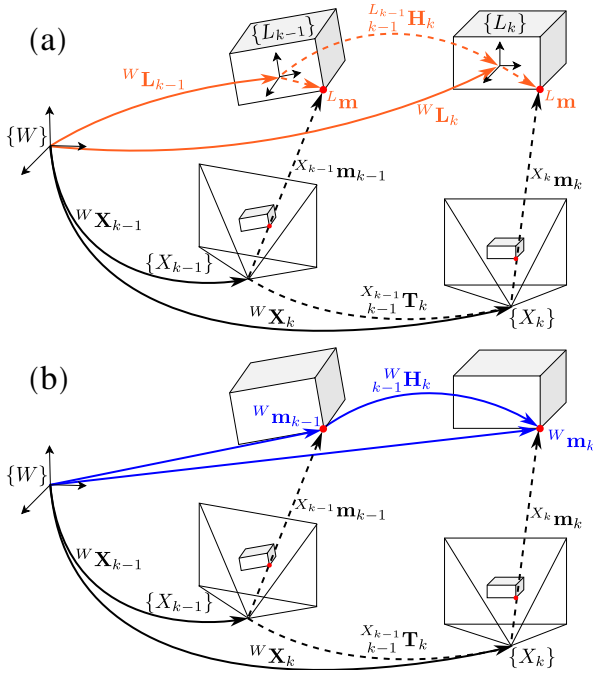

Fig. 1: **Object-centric vs world-centric**. A comparison of two Dynamic SLAM formulations viewing the same scene from time-step $k-1$ to k. Three reference frames are included, namely world $\{W\}$, object $\{L\}$ and camera $\{X\}$. **(a)** The object-centric formulation expresses dynamic points $^L\mathbf{m}$ in the local object frame $\{L\}$ defined by an estimate for object pose $^W\mathbf{L}$ at each time-step. **(b)** The world-centric representation describes the rigid body motion $^W_{k-1}\mathbf{H}_k$ directly with dynamic points $^W\mathbf{m}$.

In the context of Dynamic SLAM systems, the optimisation problem can be formulated by representing variables in different frames of reference. A common approach, presented in Fig. 1 (a), expresses observed dynamic points in the local frame of their corresponding objects, which this paper refers to as *object-centric*. This method appears intuitive as for rigid bodies, points expressed locally are static with respect to the object frame, allowing each dynamic point to be modelled as a single state variable [14]. As a consequence, the pose of each object per time-step must be a variable in the optimisation problem.

This research is funded with the support of ARIA Research and the Australian Government via the Department of Industry, Science, and Resources CRC-P program (CRCPXI000007).

Jesse Morris, Yiduo Wang and Viorela Ila are with the University of Sydney (USyd), 2006 Sydney, Australia. {jesse.morris,yiduo.wang,viorela.ila}@sydney.edu.au

[1]Open-source: https://github.com/ACFR-RPG/dynamic_slam_coordinates

An alternative approach expresses dynamic points in a known reference frame, such as camera [13] or map/world frame [15]. Our previous work, VDO-SLAM [11], [15], [16], demonstrates that an $\mathrm{SE}(3)$ motion can be expressed in any reference frame including the world frame. With that and by representing dynamic object points in the world frame, [15] produces accurate object motion estimates. Structurally, this approach results in more state variables as each dynamic point is modelled per time-step, but avoids the need to estimate the pose of each object. This paper refers to this formulation as *world-centric*, which is visualised in Fig. 1 (b).

This paper explores the impact of the formulations resulting from different representations on the underlying optimisation problem, so as to understand how to better represent objects in Dynamic SLAM systems. To this end, we introduce a graph-based optimisation framework for developing and testing different Dynamic SLAM approaches. Intrigued by the state-of-the-art literature, we implemented world and object-centric formulations, rigorously analysing the accuracy and robustness of the resulting optimisation problem. Based on this analysis, we propose the Dynamic SLAM formulation that most accurately and robustly estimates camera poses and object motions.

The contributions of this paper are as follows:

- introduces a collection of detailed mathematical formulations and graph structures for estimating egomotion and tracking dynamic objects in SLAM problems,
- rigorously analyses, evaluates and tests each formulation using real-world datasets
- provides a Dynamic SLAM optimisation framework using GTSAM [1] that implements a variety of formulations as presented in this paper.

II. Related Work

Dynamic SLAM is an active area of research in robotics, with several efficient solutions being proposed in recent years [14], [13], [15], [7], [17], [18]. Conventional solutions like ORB-SLAM 3 [5] reject dynamic objects as outliers using methods such as RANSAC [19]. Semantic information from deep learning methods are also used to detect and remove dynamic objects [7], [17], [18] to create a global map from which only camera pose and the static structure are estimated. These methods can accurately estimate camera pose in dynamic environments; however, relevant information about objects moving in the environment is discarded.

To overcome this problem, recent approaches tightly couple object tracking with SLAM, directly integrating observations of dynamic objects into the SLAM formulation and use joint optimisation methods to provide accurate estimates of the dynamic scene. These systems rely on separating dynamic points from the static background using either kinematics [20], [13] or semantics [21], [14], [15] to model each object individually, and optimise the pose or motion of these objects together with the camera/robot locations and the map (e.g. dynamic and static points). State-of-the-art literature presents two different solutions to represent the dynamic points, categorised by the reference frames in which these points are expressed.

The most common and intuitive approach is an object-centric formulation [14], [21], [22]. The immediate advantage is that each object point can be associated with only one variable in the optimisation problem, reducing the number of variables in the system. However, one of the challenges posed by such a formulation is that object poses used as dynamic points' reference frames are not directly observable. Among object-centric representations, DynaSLAM II [14] reports the best egomotion estimation when compared with other Dynamic SLAM approaches. Their experimental results present poor object motion estimations and the authors consider their use of sparse features to be the main reason behind such performance [14].

An alternative formulation is to represent dynamic objects and estimate their motions directly in a known reference frame [23], [24], [13], [15]. In this context, a known frame can either be a camera frame that moves with a sliding window, or be a well-defined reference frame, such as the world frame which commonly coincides with the first camera/robot pose. MVO [13] employs a sliding window to track dynamic objects and reports accurate camera and object motion estimates. Their formulation represents the dynamic points in the camera frame at the start of each sliding window; though it models object motions in the object frame, similar to [14]. MVO uses the object observation at the start of the sliding window as reference. Our previous work, VDO-SLAM [11], [15], [16], proposes a model-free formulation to represent and estimate object motions in any desired reference frame based on the rigid-body assumption. VDO-SLAM expresses both dynamic points and object motions in the world frame. While this formulation may appear less efficient because it introduces new variables associated with observed dynamic points at each step, our intuition suggests that a world-centric approach could significantly enhance the performance of the nonlinear solver.

III. Background

A. Reference Frames and Notations

The particular formulations discussed in this paper are concerned with a robot in motion equipped with an RGB-D camera observing and tracking static and dynamic points in the environment. Robot and camera coordinate frames are assumed to coincide. Fig. 1 presents the basic notations employed by this paper. The world frame $\{W\}$ defines the fixed global reference frame. Let ${}^{W}\mathbf{X}_k, {}^{W}\mathbf{L}_k \in \mathrm{SE}(3)$ be the camera and object poses in $\{W\}$ at time-step k, respectively. Each ${}^{W}\mathbf{L}_k$ is associated with an object frame $\{L_k\}$ and each ${}^{W}\mathbf{X}_k$ with a camera frame $\{X_k\}$.

Let $\mathbf{m}^i = \left[\tilde{\mathbf{m}}^i, 1\right]^\top$ define the homogeneous coordinates of a 3D point $\tilde{\mathbf{m}}^i \in \mathbb{R}^3$, where i is the unique tracklet index, indicating correspondences between observations. A point in the camera frame is denoted as ${}^{X_k}\mathbf{m}^i_k$. The coordinates of a dynamic point in the world frame $\{W\}$ observed at time k is ${}^{W}\mathbf{m}^i_k$, and a static point in the world frame is

${}^{W}\mathbf{m}^{i} = {}^{W}\mathbf{X}_k \, {}^{X_k}\mathbf{m}^{i}_k$. The time-step k is omitted when the variable is time-independent, i.e. static, within the represented reference frame. A point in object frame $\{L^j_k\}$ is ${}^{L^j}\mathbf{m}^i$ where j is a unique object identifier. The same point can be expressed in $\{W\}$ as ${}^{W}\mathbf{m}^i_k = {}^{W}\mathbf{L}^j_k \, {}^{L^j}\mathbf{m}^i$, where the j becomes implicit.

Fig. 1 further highlights how this notation extends to homogeneous transformations. ${}^{X_{k-1}}_{k-1}\mathbf{T}_k \in \mathrm{SE}(3)$ describes the relative camera transformation from time-step $k-1$ to k, expressed in the camera frame $\{X_{k-1}\}$, and ${}^{L_{k-1}}_{k-1}\mathbf{H}^j_k \in \mathrm{SE}(3)$ describes the motion for object j in the object frame $\{L^j_{k-1}\}$:

$$ {}^{X_{k-1}}_{k-1}\mathbf{T}_k = {}^{W}\mathbf{X}^{-1}_{k-1} \, {}^{W}\mathbf{X}_k \tag{1} $$

$$ {}^{L_{k-1}}_{k-1}\mathbf{H}^j_k = {}^{W}\mathbf{L}^{j\ -1}_{k-1} \, {}^{W}\mathbf{L}^j_k, \tag{2} $$

defining the kinematic models for camera and object.

B. Pose Transformation and Frame Change

Our previous work [15] demonstrates that, for a rigid-body object j with motion ${}^{L_{k-1}}_{k-1}\mathbf{H}^j_k$, there exists a single SE(3) transformation from time-step $k-1$ to k for all points on this object in the world frame $\{W\}$:

$$ \begin{aligned} {}^{W}\mathbf{m}^i_k &= {}^{W}\mathbf{L}^j_{k-1} \, {}^{L_{k-1}}_{k-1}\mathbf{H}^j_k \, {}^{W}\mathbf{L}^{j\ -1}_{k-1} \, {}^{W}\mathbf{m}^i_{k-1} \\ {}^{W}\mathbf{m}^i_k &= {}^{W}_{k-1}\mathbf{H}^j_k \, {}^{W}\mathbf{m}^i_{k-1}, \end{aligned} \tag{3} $$

where ${}^{W}_{k-1}\mathbf{H}^j_k$ describes the motion of a point on a rigid-body.

$$ {}^{W}_{k-1}\mathbf{H}^j_k := {}^{W}\mathbf{L}^j_{k-1} \, {}^{L_{k-1}}_{k-1}\mathbf{H}^j_k \, {}^{W}\mathbf{L}^{j\ -1}_{k-1} \in \mathrm{SE}(3) \tag{4} $$

Equation (4) represents a *frame change of a pose transformation* [25], relating ${}^{L_{k-1}}_{k-1}\mathbf{H}^j_k$, the motion in the object (or body) frame, to that in a world (inertial) reference frame ${}^{W}_{k-1}\mathbf{H}^j_k$. Using a world reference frame allows the object motion to be described in a model-free manner, eliminating the need to consider the object pose in the formulation. Based on (2) and (3), the kinematic model that describes the object motion in the world frame $\{W\}$ is as follows:

$$ {}^{W}_{k-1}\mathbf{H}^j_k = {}^{W}\mathbf{L}^j_k \, {}^{W}\mathbf{L}^{j\ -1}_{k-1} \in \mathrm{SE}(3). \tag{5} $$

IV. Formulations

This section introduces several formulations to define variables and model relations (factors) between those variables in a factor-graph-based Dynamic SLAM estimation framework similar to state-of-the-art approaches [14], [13], [15]. We categorise these formulation as either world-centric (Section IV-B) or object-centric (Section IV-C).

A. SLAM front-end

This paper focuses on the factor-graph-based Dynamic SLAM optimisation (e.g. back-end or local batch) and the proposed framework is intended to be front-end-agnostic. The discussion on a complete Dynamic SLAM pipeline or its front-end is not within the scope of this paper. The interface between the front-end and the back-end is streamlined so that the front-end can be easily replaced.

The front-end is expected to provide frame-to-frame tracking for all (static and dynamic) 3D points ${}^{X_k}\mathbf{m}^i$ and to be able to associate/cluster dynamic points by the corresponding objects. Furthermore, it can also provide initial estimates for camera poses ${}^{W}\mathbf{X}_k$ and object motion ${}^{W}_{k}\mathbf{H}^j_{k-1}$ used to track the static and dynamic points.

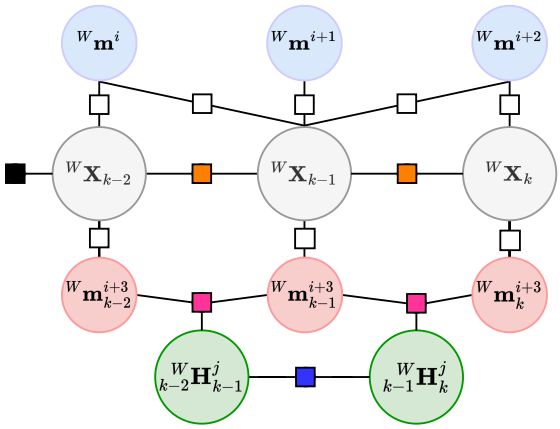

Fig. 2: **World-centric formulation factor graph**. Example factor graph including three static points ${}^{W}\mathbf{m}^{i:i+2}$ and one dynamic point ${}^{W}\mathbf{m}^{i+3}$ on object j seen at three consecutive time-steps $k-2:k$. Point measurement factors are shown as white squares, odometry factors as orange squares and world-centric motion factors as magenta squares. The motion smoothing factor is shown in blue and the prior factor is in black.

B. World-centric Formulation

The world-centric formulation jointly estimates for camera pose, object motion, static and dynamic points, all expressed in the world frame $\{W\}$ [15]. A conceptually similar variant would represent both in the first camera of a sliding window optimisation problem. Fig. 2 shows a simple example of the corresponding factor graph.

Given an observation of a 3D point ${}^{X_k}\mathbf{m}^i_k$, the *point measurement factor* models the camera pose ${}^{W}\mathbf{X}_k$ with a map point ${}^{W}\mathbf{m}^i_k$ and is given by:

$$ r({}^{W}\mathbf{X}_k, {}^{W}\mathbf{m}^i) = {}^{X_k}\mathbf{m}^i_k - {}^{W}\mathbf{X}^{-1}_k \, {}^{W}\mathbf{m}^i, \tag{6} $$

where ${}^{W}\mathbf{X}_k$ and ${}^{W}\mathbf{m}^i$ are vertices in the factor graph and require initialisation. The initial value for ${}^{W}\mathbf{X}_k$ is provided by the front-end, and ${}^{W}\mathbf{m}^i$ is initialised as ${}^{W}\mathbf{m}^i = {}^{W}\mathbf{X}_k \, {}^{X_k}\mathbf{m}^i_k$. As shown in Fig. 2, the same factor is used to refine dynamic points ${}^{W}\mathbf{m}^i_k$ as well.

The *camera odometry factor* between consecutive camera poses in the graph is formulated as:

$$ r({}^{W}\mathbf{X}_{k-1}, {}^{W}\mathbf{X}_k) = \left[\log \left({}^{W}\mathbf{X}^{-1}_k \, {}^{W}\mathbf{X}_{k-1} \, {}^{X_{k-1}}_{k-1}\mathbf{T}_k \right) \right]^{\vee}, \tag{7} $$

where the relative pose change ${}^{X_{k-1}}_{k-1}\mathbf{T}_k$ is given by the front-end. The operation $[\log(\cdot)]^{\vee}$ maps an SE(3) transformation to an $\mathbb{R}^6$ vector as per the notations of Chirikjian [26].

Based on (3), the motion of a point on a rigid body is described by a ternary motion factor, relating a pair of tracked points with their motion:

$$ r({}^{W}\mathbf{m}^i_k, {}^{W}\mathbf{m}^i_{k-1}, {}^{W}_{k-1}\mathbf{H}^j_k) = {}^{W}\mathbf{m}^i_k - {}^{W}_{k-1}\mathbf{H}^j_k \, {}^{W}\mathbf{m}^i_{k-1}. \tag{8} $$

In (8), the points from tracklet i are on the j-th object and observed at time-step $k-1$ and k, forming the *world-centric*

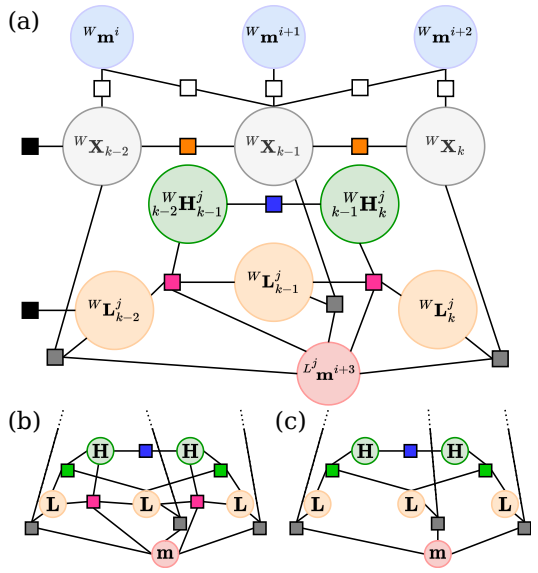

Fig. 3: **Object-centric formulation factor graph**. Example factor graphs showing three static points ${}^{W}\mathbf{m}^{i:i+2}$ and a tracked point ${}^{L^j}\mathbf{m}^{i+3}$ on a dynamic object $\mathbf{L}^j$, seen at consecutive time-steps $k-2:k$. Dynamic point measurement factors are shown as gray squares, object-centric motion factors are magenta squares and green squares represent object kinematic factors. The motion smoothing factor is blue and priors factors are black. (b) and (c) show variations of the graph in (a).

motion factors in Fig. 2.

Finally, the *smoothing factor* is introduced between consecutive object motions:

$$r({}^{W}_{k-2}\mathbf{H}^j_{k-1}, {}^{W}_{k-1}\mathbf{H}^j_k) = \left[\log\left({}^{W}_{k-2}\mathbf{H}^{j\ -1}_{k-1}\ {}^{W}_{k-1}\mathbf{H}^j_k\right)\right]^{\vee}. \quad (9)$$

This factor helps prevent abrupt, drastic and unrealistic changes in object motions between consecutive frames.

C. Object-centric Formulation

The object-centric approach estimates camera pose, static points, object motion and pose in $\{W\}$, and object points in $\{L\}$. The corresponding factor graph is visualised in Fig. 3 which shows different object-centric variations used for experiments. Variation (a) shows the basic object-centric structure, and (b) modifies the graph structure to include the *object kinematic factor* (Section IV-D), while (c) retains this factor and removes the *object-centric motion factor*. To ensure a fair comparison with the world-centric formulation, we retain common factors where possible, i.e. point measurement factors for static points and odometry factors, as indicated by identical connections between Fig. 2 and Fig. 3. These variations are introduced so that we can explore the effect that different object-centric factors have on the underlying optimisation structure, behaviour and performance.

Equation (6) is extended to express dynamic points in object frame, additionally constraining the object pose:

$$r({}^{W}\mathbf{X}_k, {}^{W}\mathbf{L}^j_k, {}^{L}\mathbf{m}^i) = {}^{X_k}\mathbf{m}^i_k - {}^{W}\mathbf{X}^{-1}_k\ {}^{W}\mathbf{L}^j_k\ {}^{L}\mathbf{m}^i. \quad (10)$$

Following the rigid body assumption we consider points in object frame ${}^{L}\mathbf{m}$ to be time independent – they are static relative to the object frame $\{L\}$.

At each time-step, the translation component of the object pose ${}^{W}\mathbf{L}^j_k$ is initialised using the centroid of the tracked object points and the rotation component is initialised with identity matrix [14]. This initial object pose is used to initialise each new dynamic point first seen at that time step: ${}^{L^j}\mathbf{m}^i = {}^{W}\mathbf{L}^{j\ -1}_k\ {}^{W}\mathbf{X}_k\ {}^{X_k}\mathbf{m}^i_k$.

The *object-centric motion factor* now connects points ${}^{L^j}\mathbf{m}$, consecutive object poses ${}^{W}\mathbf{L}^j$, and object motion ${}^{W}\mathbf{H}^j$:

$$\begin{aligned} & r({}^{W}\mathbf{L}^j_k, {}^{W}\mathbf{L}^j_{k-1}, {}^{W}_{k-1}\mathbf{H}^j_k, {}^{L^j}\mathbf{m}^i) \\ &= {}^{W}\mathbf{L}^j_k\ {}^{L^j}\mathbf{m}^i - {}^{W}_{k-1}\mathbf{H}^j_k\ {}^{W}\mathbf{L}^j_{k-1}\ {}^{L^j}\mathbf{m}^i \\ &= \left({}^{W}\mathbf{L}^j_k - {}^{W}_{k-1}\mathbf{H}^j_k\ {}^{W}\mathbf{L}^j_{k-1}\right){}^{L^j}\mathbf{m}^i. \end{aligned} \quad (11)$$

As this residual is the only factor containing $\mathbf{H}$, it should encode the kinematic model that uses object pose to define the object motion and the motion of a point on rigid body as expressed in (5) and (3), respectively. However, this factor does not actually reflect the kinematic model established in (5) as ${}^{W}\mathbf{L}^j_k - {}^{W}_{k-1}\mathbf{H}^j_k\ {}^{W}\mathbf{L}^j_{k-1} \notin \mathrm{SE}(3)$.

D. Object Kinematic Factor

We therefore propose adding an additional factor to directly model the kinematic relationship between consecutive object poses:

$$r({}^{W}\mathbf{L}^j_k, {}^{W}\mathbf{L}^j_{k-1}, {}^{W}_{k-1}\mathbf{H}^j_k) = \left[\log\left({}^{W}\mathbf{L}^{j\ -1}_k\ {}^{W}_{k-1}\mathbf{H}^j_k\ {}^{W}\mathbf{L}^j_{k-1}\right)\right]^{\vee} \quad (12)$$

We refer to it as the *object kinematic factor* and is shown as green squares in Fig. 3 (b) and (c). It explicitly describes the change in object pose ${}^{W}\mathbf{L}$ between time-step $k-1$ and k with an object motion ${}^{W}_{k-1}\mathbf{H}_k$ derived from (5).

V. Experiments

The formulations presented in Section IV are implemented and optimised using GTSAM [1]. The multi-motion visual odometry component of our previous work [15] is used as the front-end to provide frame-to-frame tracking and initial estimations for points, camera poses and object motions.

For each experiment, the front-end output is saved as a graph file [28] to provide identical data association, measurements and initial estimates, eliminating variation and randomness to ensure consistent input for each test. The same input measurements and initial estimates are used to construct each system, depending on the desired graph structure and reference frame.

The KITTI Tracking Dataset [29] is used to assess the performance of each formulation. Sequences with a sufficient variety of dynamic objects are selected, as some sequences contain no moving objects, or objects with very short trajectories. For each selected sequence, we evaluate the solution accuracy and analyse the behaviour of the optimisation for all world and object-centric formulations. The effect of different object-centric factors is investigated by including the object-centric variations (Fig. 3) in our experiments.

For comparison, we further include the camera pose errors of DynaSLAM II [14], the state-of-the-art in egomotion estimation, as reported in their paper since DynaSLAM II is not

TABLE I: Camera pose estimation (mean RPE, where $\mathbf{M} = {}^{W}\mathbf{X}_{k-1}^{-1}\,{}^{W}\mathbf{X}_{k}$) on KITTI sequences [27] comparing DynaSLAM II [14] with object and world-centric formulations. The number of variables in the factor graph (# var) for object-centric and world-centric formulations are additionally included, as well as the time to solve the full Dynamic SLAM system, which is averaged over 10 runs.

Seq	DynaSLAM II		object-centric		object-centric with OKF		world-centric		object-centric		world-centric	
	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	# var	time(s)	# var	time(s)
00	0.06	**0.04**	0.06	0.05	0.09	0.05	**0.05**	**0.04**	26335	365.3	153155	**72.0**
01	**0.04**	0.05	0.05	0.04	0.07	0.04	**0.04**	**0.03**	51426	38.4	117923	**34.2**
02	**0.02**	0.04	**0.02**	0.03	0.06	0.03	**0.02**	**0.03**	19034	102.0	38450	**22.4**
03	0.04	**0.06**	0.05	0.07	0.09	0.07	**0.03**	**0.06**	20410	215.4	92264	**24.8**
04	0.06	0.07	**0.04**	0.07	0.06	0.06	**0.04**	**0.06**	36486	39.8	80352	**22.3**
05	0.03	**0.06**	0.03	**0.06**	0.07	**0.06**	**0.02**	**0.06**	31369	63.9	99990	**41.8**
06	**0.04**	0.02	0.06	0.02	0.07	0.02	0.05	**0.01**	21353	225.7	91187	**73.9**
18	**0.02**	0.05	0.03	**0.04**	0.1	**0.04**	**0.02**	**0.04**	53680	**269.2**	340844	381.5
20	0.04	0.07	**0.03**	**0.05**	0.1	**0.05**	**0.03**	**0.05**	129316	**422.6**	711804	656.9

open-source. However, we omit their object motion error in our comparison because their object motion estimation [14] performs comparably to our object-centric formulation, and their error metrics are not specified. MVO [13], another closed-source system, is formulated similarly to the world-centric approach presented in this paper. However, their system uses a sliding window optimisation; therefore, their results are not comparable.

A. Error Metrics

The paper reports Relative Pose Error (RPE) [30] for both camera and objects computed as follows. Given a ground truth transformation $\mathbf{M}_{\text{gt}} \in \text{SE}(3)$ and a corresponding estimate $\mathbf{M} \in \text{SE}(3)$, we compute the error as $\mathbf{E} = \mathbf{M}^{-1}\mathbf{M}_{\text{gt}}$ for all $\text{SE}(3)$ estimates. The translational error E_t is the L_2 norm of the translational component of $\mathbf{E}$, and the rotational error E_r is the angle of its rotational component. Each table will indicate what transformation $\mathbf{M}$ represents.

B. Camera Pose Error & Factor Graph

Table I shows the evaluation results of estimated camera poses from the different formulations. The world-centric formulation provides the best results among all methods in the most sequences, consistently performing better than, or at least on a par with, the state-of-the-art benchmark. However, all formulations present similar accuracy for camera pose estimations with minor differences. We believe that it is because there are many static background features throughout KITTI sequences to enable accurate camera tracking.

The number of variables in each formulation and associated optimisation time are presented in Table I. Despite a greater number of variables, the world-centric approach on average takes substantially less time to provide a solution while producing more accurate object motion and pose estimates, as shown in Table II and III. This highlights how the graph structures resulting from different representation choices have a clear impact on the optimisation efficiency.

C. Object Motion Error

Object motion errors are shown in Table II. The world-centric formulation is the most accurate overall, outperforming the state-of-the-art, i.e. the object-centric formulation, for ∼95 % of dynamic objects. Our proposed object-centric variations improve the accuracy of the state-of-the-art, but

TABLE II: Errors of object motion, $\mathbf{H}$, on sequences with well-tracked dynamic objects, with $\mathbf{M} = {}^{W}_{k-1}\mathbf{H}_{k}$. The average error for dynamic objects tracked for the most frames are included, as is the mean error across all objects in the sequence. The object-centric variations correspond to the factor graphs visualised in Fig. 3 (a), (b) and (c) respectively. Blue entries represent the best object-centric estimations, and the bold ones are the best results overall.

Seq|obj	object-centric		object-centric with OKF		object-centric only OKF		world-centric	
	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)
00|01	3.54	1.7	1.27	0.49	1.01	**0.4**	**0.9**	0.51
mean	4.73	3.85	1.67	1.35	1.05	1.11	**0.78**	**0.52**
03|01	2.14	1.07	3.87	1.26	0.68	0.4	**0.22**	**0.17**
mean	0.93	5.39	2.38	1.12	0.46	0.34	**0.24**	**0.23**
04|03	5.22	2.47	2.7	1.08	1.82	0.73	**0.64**	**0.37**
04|04	1.13	4.63	1.9	0.85	1.19	0.59	**0.38**	**0.23**
04|05	3.57	6.55	2.56	1.21	1.13	0.62	**0.5**	**0.33**
mean	5.48	5.08	2.13	2.11	1.81	1.35	**0.77**	**0.51**
05|20	1.91	6.54	4.79	30.77	1.94	5.38	**0.53**	**1.37**
05|24	4.74	18.28	1.19	5.48	3.45	15.04	**0.51**	**2.28**
mean	1.66	13.68	**0.18**	12.43	2.01	7.73	0.7	**5.19**
18|04	6.1	14.50	1.83	4.68	0.89	4.07	**0.25**	**1.82**
mean	**0.25**	11.26	0.97	25.67	1.17	4.67	0.52	**1.95**
20|32	3.60	2.17	0.42	0.23	0.32	0.17	**0.08**	**0.15**
mean	4.90	14.70	1.15	5.96	1.21	7.17	**0.69**	**5.46**

the world-centric method still produces the best motion estimations in ∼80 % of all objects. The poor results of the base object-centric approach suggests that the object-centric motion factor is unable to effectively contribute to the optimisation as the kinematic model is not correctly encoded.

To further probe this observation, the object kinematic factor (OKF) described in (12) is subsequently included in the optimisation. These results are denoted in all tables as 'object-centric with OKF'. This factor explicitly enforces the kinematic model in (5), and improves the motion estimate substantially, particularly in translation.

Our results indicate that object-centric motion factor alone is detrimental to the optimisation problem as it 'pulls' the optimisation in several directions, resulting in sub-optimal solutions. We further investigated and analysed the evolution of the chi-squared errors of the world and object-centric formulations during nonlinear least squares optimisation using Levenberg-Marquardt (LM) solver, as per Fig. 4. The x-axis denotes the number of steps that the LM solver requires for convergence and indicates the amount of times a linear sys-

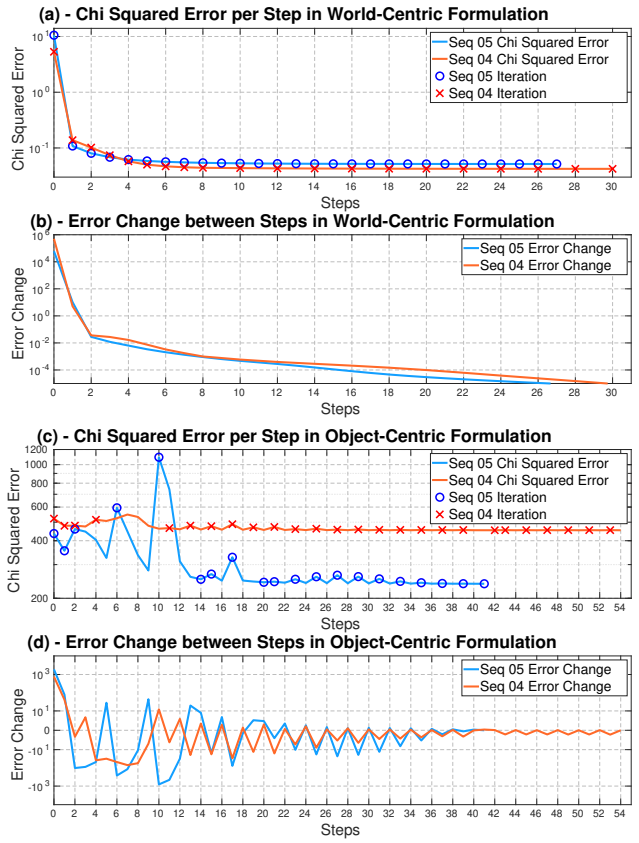

Fig. 4: **Chi-squared errors χ_n for world and object-centric formulations per-step during optimisation for sequences 04 and 05**. The Error Change is computed as $\chi^2_{n-1} - \chi^2_n$ where n represents the step. The linear system is solved at every step but only re-linearised every iteration, which is marked in plots (a) and (c).

tem is solved. The world-centric system in Fig. 4 (a) exhibits a consistent downward trend in the chi-squared error, while the per-step error change of the object-centric formulation, shown in Fig. 4 (c), oscillates between positive and negative, requiring more steps than the former. We believe this is the reason behind the poor efficiency reported in Table I. While not presented due to limited space, the convergence trends displayed in Fig. 4 are common to all sequences and will be included in the supplementary material.

Removing the object-centric motion factor improves the overall estimation accuracy, as shown in the 'object-centric only OKF' columns of Table II and III, further emphasising the detrimental effect of this factor. While this formulation exhibits better performance than the other object-centric variations, only retaining the object kinematic factor limits the effectiveness of the SLAM problem as the point measurements are no longer used to model the rigid-body point motion expressed in (3). We additionally noted that the object centric formulations require an extra prior on the first pose of each object trajectory to avoid an indeterminate linear system — this was also observed in [14].

In contrast, the world-centric formulation does not require an object prior, and explicitly models the rigid-body motion only using variables in a known reference frame $\{W\}$. The optimisation problem arising from this formulation results in a more stable optimisation process, as shown in Fig. 4 (a) and (b), and the most accurate estimation in our experiments.

TABLE III: Relative errors for object pose, $\mathbf{L}$, with $\mathbf{M} = {}^{W}\mathbf{L}_{k-1}^{-1}\,{}^{W}\mathbf{L}_{k}$. Blue entries are the best object-centric estimations, and the bold ones are the best results overall.

	object-centric		object-centric with OKF		object-centric only OKF		world-centric	
Seq|obj	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)
00|01	3.63	0.38	1.33	0.26	1.07	0.35	**0.92**	**0.22**
mean	3.72	0.38	1.51	0.36	1.11	0.33	**0.79**	**0.15**
03|01	3.79	0.68	3.85	1.94	0.67	0.83	**0.2**	**0.15**
mean	3.73	0.72	2.37	1.4	0.46	0.46	**0.24**	**0.15**
04|03	3.28	0.9	2.63	1.18	1.79	1.11	**0.64**	**0.12**
04|04	4.13	0.88	2.1	0.89	1.39	0.97	**0.38**	**0.12**
04|05	0.51	1.11	2.58	1.03	1.09	1.09	**0.43**	**0.15**
mean	3.74	0.9	2.23	0.87	1.92	0.93	**0.76**	**0.1**
05|20	2.64	3.02	4.58	2.73	2.05	3.19	**0.54**	**0.23**
05|24	5.29	2.74	1.34	2.79	3.37	3.1	**0.53**	**0.15**
mean	2.25	2.05	**0.29**	1.94	2.07	2.33	0.63	**0.55**
18|04	0.78	0.20	1.92	**0.09**	0.94	0.16	**0.26**	0.19
mean	1.69	1.34	1.08	1.00	1.21	1.48	**0.53**	**0.27**
20|32	0.77	0.21	0.45	0.08	0.35	0.12	**0.08**	**0.03**
mean	1.90	0.31	1.28	**0.20**	1.33	0.58	**0.68**	0.53

D. Object Pose Error

Table III presents relative object pose errors which follow a similar trend to the object motion errors. The world-centric approach does not estimate object pose; instead, using an initial pose, the estimated per-frame motion can be used to propagate the pose of a corresponding object, constructing its full trajectory. The ground truth ${}^{W}\mathbf{L}^{j}_{0,\,\mathrm{gt}}$ is used as the starting pose. To ensure a fair comparison, the same ground truth is used to initialise object poses ${}^{W}\mathbf{L}^{j}_{0}$ in object-centric approaches. Despite not estimating for ${}^{W}\mathbf{L}$, the world-centric formulation is more accurate for $\sim$95 % of all objects when compared to the state-of-the-art, and $\sim$80 % compared to our proposed object-centric variations.

VI. Conclusion and Future Work

This paper has undertaken a comprehensive analysis of multiple solutions for Dynamic SLAM and evaluated the proposed formulations on existing real-world datasets. For that, we developed a front-end agnostic optimisation framework using GTSAM [1] that can easily implement and test different configurations. These formulations are categorised as *object-centric* and *world-centric* according to how dynamic objects and their corresponding point observations are represented in the factor graph. The object-centric formulation is more intuitive, but our analysis shows that a world-centric approach produces much more accurate object motion estimations while displaying better stability during optimisation. Our results highlight that employing different representations, as well as their subsequent graph structures, has a significant impact on the definition and performance of the underlying optimisation problem. In the future, we plan to derive a formal characterisation of our findings that can also be used to provide clear guidelines in advance for determining the circumstances under which specific formulations will outperform others.

REFERENCES

[1] F. Dellaert and GTSAM Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam

[2] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, "Advances in inference and representation for simultaneous localization and mapping," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 215–242, 2021.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE/ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, 2011, pp. 127–136.

[4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Trans. Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[6] 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 *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2008, pp. 1455–1462.

[7] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[8] M. N. Finean, W. Merkt, and I. Havoutis, "Simultaneous scene reconstruction and whole-body motion planning for safe operation in dynamic environments," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 3710–3717.

[9] A. Hermann, J. Bauer, S. Klemm, and R. Dillmann, "Mobile manipulation planning optimized for gpgpu voxel-collision detection in high resolution live 3d-maps," in *Intl. Symp. on Robotics/Robotik*, 2014, pp. 1–8.

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

[11] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic SLAM: The Need for Speed," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 2123–2129.

[12] K. M. Judd and J. D. Gammell, "Occlusion-robust mvo: Multimotion estimation through occlusion via motion closure," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 5855–5862.

[13] ——, "Multimotion Visual Odometry (MVO)," *Intl. J. of Robotics Research*, 2021, submitted, Manuscript #IJR-21-4311, arXiv:2110.15169 [cs.RO].

[14] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "Dynaslam ii: Tightly-coupled multi-object tracking and slam," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5191–5198, 2021.

[15] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: A Visual Dynamic Object-aware SLAM System," *arXiv preprint arXiv:2005.11052*, 2020.

[16] ——, "Robust Ego and Object 6-DoF Motion Estimation and Tracking," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 5017–5023.

[17] R. Hachiuma, C. Pirchheim, D. Schmalstieg, and H. Saito, "Detectfusion: Detecting and segmenting both known and unknown dynamic objects in real-time slam," *arXiv preprint arXiv:1907.09127*, 2019.

[18] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "Flowfusion: Dynamic dense rgb-d slam based on optical flow," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7322–7328.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] J. Huang, S. Yang, Z. Zhao, Y. Lai, and S. Hu, "Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation," in *Proc. of the Intl. Conf. on Computer Vision (ICCV)*, 2019, pp. 5874–5883.

[21] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, "Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings," in *Proc. of the IEEE/CVF Intl. Conf. Computer Vision and Pattern Recognition*, 2020, pp. 2168–2177.

[22] I. Ballester, A. Fontán, J. Civera, K. H. Strobl, and R. Triebel, "Dot: Dynamic object tracking for visual slam," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021, pp. 11 705–11 711.

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

[24] K. M. Judd and J. D. Gammell, "The Oxford Multimotion Dataset: Multiple SE(3) Motions with Ground Truth," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 800–807, 2019.

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

[26] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups: Classical Results and Geometic Methods*. Birkhäuser, 1994.

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition*, 2012.

[28] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3607–3613.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Intl. J. of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[30] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 573–580.