SLAM Backends with Objects in Motion: A Unifying Framework and Tutorial

Chih-Yuan Chiu*

Abstract—Simultaneous Localization and Mapping (SLAM) algorithms are frequently deployed to support a wide range of robotics applications, such as autonomous navigation in unknown environments, and scene mapping in virtual reality. Many of these applications require autonomous agents to perform SLAM in highly dynamic scenes. To this end, this tutorial extends a recently introduced, unifying optimization-based SLAM backend framework to environments with moving objects and features [1]. Using this framework, we consider a rapprochement of recent advances in dynamic SLAM. Moreover, we present dynamic EKF SLAM: a novel, filtering-based dynamic SLAM algorithm generated from our framework, and prove that it is mathematically equivalent to a direct extension of the classical EKF SLAM algorithm to the dynamic environment setting. Empirical results with simulated data indicate that dynamic EKF SLAM can achieve high localization and mobile object pose estimation accuracy, as well as high map precision, with high efficiency.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a well-studied robotics problem in which an autonomous agent attempts to locate itself in an uncharted environment while constructing a map of said environment [2, 3]. Most state-of-the-art SLAM algorithms operate under the static world setting, in which the locations of landmarks in the robot’s environment are assumed to be fixed. This greatly restricts the applicability of SLAM algorithms to robotics tasks such as autonomous navigation, in which SLAM-constructed maps must describe a wide variety of dynamic objects, such as moving obstacles, human-operated vehicles, or other autonomous agents.

To bridge this gap, the rapidly maturing dynamic SLAM community aims to design SLAM algorithms that track moving objects while performing SLAM on the underlying static scene. Specifically, dynamic SLAM algorithms simultaneously estimate ego robot states, static features, features on moving objects, and poses of moving objects. To this end, Wang et al. proposed the SLAMMOT algorithm, which separately performs motion tracking for dynamic objects and SLAM over an underlying, fixed background [4]. Yang et al. introduced CubeSLAM, which assigns each dynamic object a rectangular bounding box, and tracks the boxes’ trajectories across time [5]. Huang et al. proposed ClusterSLAM, which aggregates feature points corresponding to various dynamic objects in the scene, then performs bundle adjustment over each cluster [6]. Bescos et al. presented DynaSLAM and DynaSLAM II, which uses the ORB-SLAM algorithm to extract features of, and subsequently track, dynamic objects [7, 8]. Zhang et al. introduce VDO-SLAM, which fuses dense optical flow and image segmentation to perform joint inference over robot poses, static landmark positions, and the pose and feature positions of mobile objects [9]. Although these approaches obtain reasonable accuracy in tracking moving objects, they typically incur a computational burden that increases rapidly with the number of moving objects tracked, and the length of the time horizon over which inference is performed.

In this work, we extend the unifying, optimization-based SLAM formulation in [1] to the dynamic SLAM setting. We illustrate that the aforementioned dynamic SLAM algorithms employ back-ends corresponding to different design choices in the context of our framework. To address the computational limitations of existing methods, we use our framework to derive dynamic EKF-SLAM, a filtering-based algorithm that establishes a rapprochement between two classes of algorithms: efficient conventional filtering-based methods for static-world SLAM [10], and accurate but computationally costly bundle adjustment methods underlying existing dynamic SLAM algorithms. We prove that dynamic EKF SLAM is mathematically equivalent to a straightforward extension of the conventional EKF-SLAM algorithm to dynamic scenes. We then illustrate the empirical success of dynamic EKF-SLAM in performing inference over a simulated driving scenario, in which an ego autonomous vehicle travels down a highway in the presence of two other vehicles and a jaywalking pedestrian.

II. DYNAMIC SLAM: A UNIFYING FRAMEWORK

Suppose that, at time $t \geq 0$, the estimated variables of the ego robot describe its past and/or present poses, $n_f \in \mathbb{N}$ static features, and $n_o \in \mathbb{N}$ moving objects with $n_{of}(\alpha) \in \mathbb{N}$ features for each object index $\alpha \in \{1, \ldots, n_o\}$. Given $n, n_1, n_2 \in \mathbb{N}$, with $n_1 < n_2$, set $[n] := \{1, \ldots, n\}$ and $[n_1 : n_2] := \{n_1, \ldots, n_2\}$. We have:

- $\{x_t \in \mathbb{R}^{d_x} : t \in [T]\}$ denotes ego robot states, e.g., its poses and velocities, etc., relative to a global frame $\mathbf{G}$.
- $\{I_k^{(s)} \in \mathbb{R}^{d_f} : k \in [n_f]\}$ describes the current position estimate of each of the $n_f$ currently tracked static features relative to frame $\mathbf{G}$, with corresponding feature measurements $\{z_{t,k}^{(s)} \in \mathbb{R}^{d_z} : t \in \{0\} \cup [T], k \in [n_{of}(\alpha)]\}$ at each time $t \in \{0\} \cup [T]$.
- $\{I_t^{(m)} \in \mathbb{R}^{d_f} : t \in \{0\} \cup [T], \alpha \in [n_o], k \in [n_{of}(\alpha)]\}$ describes the feature position estimates, at each time $t \in \{0\} \cup [T]$, of each of the $k$ features on the $\alpha$-th moving object, maintained in the estimation window relative to $\mathbf{G}$.
to frame G, with corresponding feature measurements
\{z(m)_{t,\alpha,k} \in \mathbb{R}^d_z : t \in \{0\} \cup [T], \alpha \in [n_o], k \in [n_of(\alpha)]\}.
• This is a vigorous feature and pose augmentation scheme, with
little marginalization within the estimation window.

\[ \begin{align*}
& x_{t+1} = g(x_t) + w_t, \quad z_t \sim \mathcal{N}(0, \Sigma_w), \\
& z_{t,k} = h(x_t, f_k) + v_{t,k}, \quad v_{t,k} \sim \mathcal{N}(0, \Sigma_v), \\
& f_{t,\alpha,k}^{(m)} = g^\alpha(x_{t,\alpha}, f_{0,\alpha,k}) + u_{t,\alpha}, \quad u_{t,\alpha} \sim \mathcal{N}(0, \Sigma_u), \\
& \forall t \in \{0\} \cup [T], \alpha \in [n_o], k \in [n_of(\alpha)].
\end{align*} \]

where \( \mathcal{N}(\mu, \Sigma) \) denotes the Gaussian distribution with mean \( \mu \in \mathbb{R}^d \) and covariance matrix \( \Sigma \in \mathbb{R}^{d \times d} \), for some \( d \in \mathbb{N} \), and \( \Sigma_w, \Sigma_v \in \mathbb{R}^{d_z \times d_z}, \Sigma_u \in \mathbb{R}^{d \times d} \). The sections below, assume that \( \frac{\partial h}{\partial f}(x_t, f_k) \) is surjective at each \( (x_t, f_k) \in \mathbb{R}^{d_z} \times \mathbb{R}^d \), and that \( \frac{\partial^2 h}{\partial f^2} \) is injective at each \( (x_{t,\alpha}, f_{0,\alpha}^{(m)}) \in \mathbb{R}^{d_z} \times \mathbb{R}^d \).

Our optimization-based formulation of dynamic SLAM includes the following steps, each of which updates the running cost term (“cost → cost’”).

1) Feature Augmentation:
Let \( \{f_k : k \in I_f\} \subset \mathbb{R}^{d_f} \) denote feature measurements, taken with respect to previously untracked features \( \{f_k : k \in I_f\} \subset \mathbb{R}^{d_f} \). These may correspond to static or moving objects. The feature augmentation step updates the running cost to include residual terms concerning these newly observed features:

\[ \text{cost’} = \text{cost} + \sum_{k \in I_f} \|z_{t,k} - h(x_t, f_k)\|^2_{\Sigma_v^{-1}} \]

2) Moving Object Pose Augmentation:
Let \( \{f_{t,\alpha,k} : k \in I_{f,\alpha}, \alpha \in I_\alpha\} \) denote features of tracked moving objects that have been observed at times \( t_1 \) and \( t_2 \), with \( t_1 < t_2 \). For simplicity, define:

\[ f_{\tau,\alpha}^{(m)} := \begin{pmatrix} f_{\tau,\alpha,1}^{(m)} \\ \vdots \\ f_{\tau,\alpha,n_{of}(\alpha)}^{(m)} \end{pmatrix} \in \mathbb{R}^{n_{of}(\alpha)d_f} \]

for each \( \alpha \in [n_o], \tau \in \{0, t\} \). The moving object pose augmentation step appends the current pose estimates of tracked moving objects, i.e., \( \{\xi_{t,\alpha} : \alpha \in [n_o]\} \), to the running cost:

\[ \text{cost’} = \text{cost} + \sum_{\alpha \in I_\alpha} \sum_{k \in I_{f,\alpha}} \|f_{t,\alpha,k}^{(m)} - g^\alpha(\xi_{t,\alpha}, f_{0,\alpha,k}^{(m)})\|^2_{\Sigma_x^{-1}} \]

3) Static Feature Update:
Let \( \{z_k : k \in I_f\} \subset \mathbb{R}^{d_z} \) denote feature measurements, taken with respect to previously tracked static features \( \{f_k^{(s)} : k \in I_f\} \subset \mathbb{R}^{d_f} \). The feature update step updates the cost as follows:

\[ \text{cost’} = \text{cost} + \sum_{k \in I_f} \|z_k^{(s)} - h(x_t, f_k^{(s)})\|^2_{\Sigma_z^{-1}} \]

4) Smoothing Factor Augmentation:
The smoothing factor augmentation step constrains the most recent moving object pose transformation \( \xi_{t-1,\alpha} \) to \( \xi_{t,\alpha} \) from significantly differing from the second most recent moving object pose transformation \( \xi_{t-2,\alpha} \) to \( \xi_{t-1,\alpha} \), for each object indexed \( \alpha \in [n_o] \):

\[ \text{cost’} = \text{cost} + \sum_{\alpha \in I_\alpha} \|s(\xi_{t-2,\alpha}, \xi_{t-1,\alpha}, \xi_{t,\alpha})\|^2_{\Sigma_z^{-1}} \]

Here, \( s : \mathbb{R}^{d_z} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z} \) is a smoothing function, e.g., for \( d_z = 1 \), take \( s(\xi_{t-2,\alpha}, \xi_{t-1,\alpha}, \xi_{t,\alpha}) = (\xi_{t,\alpha} - \xi_{t-1,\alpha}) - (\xi_{t-1,\alpha} - \xi_{t-2,\alpha}) \).

5) State Propagation:
At each time \( t \), the state propagation step updates the cost to include residual terms involving the odometry measurements between \( x_t \), the pose at time \( t \), and \( x_{t+1} \), the pose at time \( t + 1 \):

\[ \text{cost’} = \text{cost} + \|x_{t+1} - g(x_t)\|^2_{\Sigma_x^{-1}} \]

Poles and features present in the optimization window may be dropped (instead of marginalized) to improve optimization accuracy, as is common in SLAM algorithms operating under the static world assumption [11, 12]. In addition, the above formulation naturally extends to scenarios in which dynamical quantities evolve on smooth manifolds, rather than on Euclidean spaces (see [1], Section 3 and Appendix A).

III. UNIFYING EXISTING ALGORITHMS

In this section, we interpret the back-ends of recently proposed dynamic SLAM algorithms as the selection of different design choices within the context of our framework, as presented in Section II. We focus in particular on design choices relevant to tracking moving objects.

• CubeSLAM [5]—In CubeSLAM, pose estimates of moving objects are obtained by forming and tracking rectangular bounding boxes across time. Feature augmentation of moving objects into the estimation window is avoided.

• ClusterSLAM [6]—ClusterSLAM models moving objects by aggregating and tracking feature clouds. The authors describe “fully-coupled”, “semi-decoupled”, and “decoupled” estimation schemes for static SLAM and moving object tracking, which correspond to increasingly aggressive marginalization schemes in our framework.

• VDO-SLAM [9]—The VDO-SLAM algorithm performs object segmentation, then samples dense feature clouds within each bounding box to track the associated moving object. In contrast with CubeSLAM, this is a vigorous feature and pose augmentation scheme, with little marginalization within the estimation window.
VDO-SLAM can enjoy considerable accuracy, but may also incur high computational burden \[8\].

- **DynaSLAM II** \[8\]— DynaSLAM II tracks moving objects across time, by repeatedly performing pose augmentation with pose estimates constructed from newly observed features. Unlike VDO-SLAM, the most recent feature position estimates of these moving objects are then quickly dropped or marginalized, to reduce the computation burden at the next timestep.

**IV. DYNAMIC EKF-SLAM**

Although the algorithms described in III can attain high estimation accuracy, their computation time often scales poorly with the number of moving objects or timesteps tracked. Inspired by the efficiency of filtering-based SLAM frameworks under the static world assumption, we use the unifying framework presented in Section II to construct the dynamic EKF algorithm, described below, to address this issue.

At each time \( t \), the dynamic EKF SLAM algorithm on Euclidean spaces maintains the full state vector:

\[
\tilde{x}_t = (x_t, f(s), f(m), \xi) \in \mathbb{R}^{d_x},
\]

where \( d_x := d_z + n_{df} + 2 \cdot \sum_{n=1}^{n_o} n_{af}(\alpha) d_f + (t-1)n_{od}d_x \). (For generality, we assume that all past moving object poses are maintained; in practice, these can be dropped). The components of \( \tilde{x}_t \) are as follows:

- **Ego robot pose:**
  \( x_t \in \mathbb{R}^{d_x} \) denotes the ego robot pose at the current time \( t \).

- **Static feature position estimates:**
  \( f(s) := (f'_1, \ldots, f'_{n_f}) \in \mathbb{R}^{n_f d_f} \) is the position estimates of the \( n_f \) in \( \mathbb{N} \) static features currently tracked.

- **Moving object feature position estimates:**
  \( f(m) \), defined below, is the feature positions of moving objects at the initial time \( 0 \) and the current time \( t \). Here, \( f_{\tau,\alpha,k} \in \mathbb{R}^{d_f} \) denotes the position estimate of the \( k \)-th feature of the moving object indexed \( \alpha \) at time \( \tau \), for each \( \tau \in \{0, t\} \), \( \alpha \in [n_o] \), and \( k \in [n_{af}(\alpha)] \), and \( N_f := \sum_{\alpha=1}^{n_o} n_{af}(\alpha) \) denotes the total number of features summed over all moving objects:

\[
f(m) \quad := \quad \left(f_{0,1,1}, \ldots, f_{0,1,n_{af}(1)}(1), \ldots, f_{0,1,1}, \ldots, f_{0,1,n_{af}(n_o)}(n_o), \right.
\]
\[
\left. f_{t,1,1}, \ldots, f_{t,1,n_{af}(1)}, \ldots, f_{t,1,1}, \ldots, f_{t,1,n_{af}(n_o)} \right) \in \mathbb{R}^{2 N_f d_f}.
\]

For notational simplicity, we assume all features on all moving objects have been observed since the start of the time horizon. (This assumption can easily be relaxed).

- **Moving object poses:**
  \( \xi := (\xi_1, \ldots, \xi_{n_x}, \ldots, \xi_{t,1}, \ldots, \xi_{t,n_x}) \in \mathbb{R}^{t n_x d_x} \) denotes the past and present poses of the \( n_o \) objects currently tracked. Here, \( \xi_{\tau,\alpha} \in \mathbb{R}^{d_x} \) denotes the pose, of the moving object indexed \( \alpha \in [n_o] \), at time \( \tau \).

---

**Algorithm 1: Dynamic EKF SLAM, as Iterative optimization.**

Data: Prior \( \mathcal{N}(\mu_0, \Sigma_0) \) on \( x_0 \in \mathbb{R}^{d_x} \), noise covariances \( \Sigma_w \in \mathbb{R}^{d_x \times d_x}, \Sigma \in \mathbb{R}^{d_x \times d_x}, \Sigma_0 \in \mathbb{R}^{d_x \times d_x} \), dynamics map \( g' : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x} \), measurement map \( h : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_f} \), inverse measurement map \( \ell : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_f} \), moving object dynamics map \( g'' : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_f} \) for each object indexed \( \alpha \in [n_o] \), inverse moving object dynamics map \( g'' : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_f} \) for each object indexed \( \alpha \in [n_o] \).

Result: Estimates \( \mu_t \) at \( t \in \{0, \ldots, T\} \).

1. cost_0 \( \leftarrow \left\| x_0 - \mu_0 \right\|_{\Sigma_0}^2 \\
2. n_f, n_o \leftarrow 0
3. for \( t = 0, \ldots, T - 1 \) do
4. \( n_f \leftarrow \) Number of tracked features on static features
5. \( N_f \leftarrow \) Total number of tracked features on moving objects
6. \( \{z_{t,k} : k \in [n_f + N_f + 1 : n_f + N_f + N'_f]\} \leftarrow \) Measurements of new features, corresponding to both static landmarks and moving objects.
7. cost \( \leftarrow \)
8. \( \mu_t' \leftarrow \left(\mu_t, \ell(x_t, f_{t,f(s)}(\alpha), z_{t,k})\right)_{\Sigma_t^{-1}} \)
9. \( \mu_t, \Sigma_t \leftarrow \text{Gauss-Newton, on cost}_t, \text{about } \mu_t \) [1].
10. Increment \( n_f, n_o \), \( n_{af}(\alpha) = \alpha \in [n_o] \) as appropriate, given the newly detected \( n'_f \) static features and \( N'_f \) features on moving objects.
11. if \( n_o \geq 1 \) then
12. cost \( \leftarrow \) cost + \( \sum_{\alpha=1}^{n_o} \sum_{k=0}^{n_{af}(\alpha)} \left\| f_{t,\alpha,k} - g''(\xi_{\alpha,\tau}, f_{t,\alpha,k}) \right\|_{\Sigma_t^{-1}}^2 \)
13. \( \mu_t \leftarrow \left(\mu_t, \gamma''(f_{t,\alpha,k}(\alpha), f_{t,\alpha,k}(\alpha), \gamma''(f_{t,\alpha,k}(\alpha), f_{t,\alpha,k}(\alpha)))) \right)_{\Sigma_t^{-1}} \)
14. \( \mu_t, \Sigma_t \leftarrow \text{Gauss-Newton, on cost}_t, \text{about } \mu_t \) [1], Alg. 3).
15. (Optional) Drop \( \{f_{t,\alpha,k} : \alpha \in [n_o]\} \) from the mean and covariance estimates.
16. end
17. cost \( \leftarrow \) cost + \( \sum_{\alpha=1}^{n_o} \left| s(\xi_{\alpha-2,0}, \xi_{\alpha,\tau}, \xi_{\alpha,\tau}) \right| \Sigma_t^{-1} \)
18. \( \mu_t, \Sigma_t \leftarrow \text{Gauss-Newton, on cost}_t, \text{about } \mu_t \) [1], Alg. 3).
19. \( \{z_{t,k} : k \in [n_f]\} \leftarrow \) Measurements of existing static features.
20. cost \( \leftarrow \) cost + \( \sum_{\alpha=1}^{n_o} \left\| z_{t,k} - h(x_t, f_{t,k}) \right\|_{\Sigma_t^{-1}}^2 \)
21. \( \mu_t, \Sigma_t \leftarrow \text{Gauss-Newton, on cost}_t, \text{about } \mu_t \) [1], Alg. 3).
22. cost \( \leftarrow \) cost + \( \left\| x_{t+1} - g(x_t) \right\|_{\Sigma_t^{-1}}^2 \)
23. \( \mu_{t+1}, \Sigma_{t+1} \leftarrow \text{Marginalization, on cost}_{t+1} \) with \( x_M = x_t, \text{about } (\mu, \Sigma) \) [1], Alg. 4.
24. cost \( \leftarrow \) cost + \( \left\| x_{t+1} - \mu_{t+1} \right\|_{\Sigma_{t+1}^{-1}}^2 \)
25. end
26. return \( \mu_0, \ldots, \mu_T \)
Algorithm 2: Dynamic EKF SLAM, Standard formulation.

Data: Prior $N(\mu_0, \Sigma_0)$ on $x_0 \in \mathbb{R}^{d_x}$, noise covariances $\Sigma_w \in \mathbb{R}^{d_w \times d_w}$, $\Sigma_u \in \mathbb{R}^{d_u \times d_u}$, $\Sigma_e \in \mathbb{R}^{d_e \times d_e}$, measurement map $h : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z}$, inverse measurement map $\ell : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z}$, object dynamics map $g^o : \mathbb{R}^{d_x + d_o} \rightarrow \mathbb{R}^{d_x + d_o}$, inverse moving object dynamics map $g^m : \mathbb{R}^{d_x + d_m} \rightarrow \mathbb{R}^{d_x + d_m}$, time horizon $T \in \mathbb{N}$, number of features $n_f \in \mathbb{N}$, number of moving objects $n_o \in \mathbb{N}$.

Result: Estimates $\mu_t$, $\forall t \in \{0, 1, \ldots, T\}$.

1. $c_{t,0} \leftarrow \|x_0 - \mu_0\|^2_{\Sigma_0}^{-1}$
2. $n_f, n_o \leftarrow 0$
3. for $t = 0, 1, \ldots, T - 1$
   4. $\{z^{(s)}_{t,k} : k \in [n_f + 1 : n_f + n_f']\} \leftarrow$ Measurements of new static features.
   5. $\mu_t, \Sigma_t, n_f \leftarrow$ Alg. 3, Dynamic EKF, (Static) Feature Augmentation
   6. if $n_o + 1 \geq n_f$ then
      7. $\{z^{(m)}_{t,k} : \alpha \in [n_o], k \in [n_f(\alpha) + n_f(\alpha')]\} \leftarrow$ Measurements of new objects' tracked and new objects'
         new features of previously moving objects indexed $\alpha \in [n_o]$.
      8. $\mu_t, \Sigma_t \leftarrow$ Alg. 3, Dynamic EKF, (Dynamic) Feature Augmentation
      9. $\mu_t, \Sigma_t \leftarrow$ Alg. 4, Dynamic EKF, (Dynamic) Object Pose Augmentation
   end
   10. if detect $n_o' \geq 1$ new moving objects then
       11. $\{z^{(m)}_{t,k} : \alpha \in [n_o + 1 : n_o + n_o']\} \leftarrow$ Measurements of features of new moving objects.
      12. $\mu_t, \Sigma_t \leftarrow$ Alg. 3, Dynamic EKF, (Dynamic) Feature Augmentation
      13. $n_o \leftarrow n_o + n_o'$
   end
   14. $\mu_t, \Sigma_t \leftarrow$ Alg. 6, Dynamic EKF, Smoothing Update
   15. $\{z^{(s)}_{t} : k \in [n_f]\} \leftarrow$ Measurements of existing static features.
   16. $\mu_{t+1}, \Sigma_{t+1} \leftarrow$ Alg. 7, Dynamic EKF, State Propagation
17. return $\mu_0, \cdots, \mu_T$

for each $\alpha \in [n_o]$ and $\tau \in [t]$. To ensure computational tractability, past pose estimates may be dropped.

Below, if unspecified, we assume the components in the full state $\hat{x}_t \in \mathbb{R}^{d_x}$ appear in the order given in (4), i.e., $\hat{x}_t = (x_t, f(s), f(m), \xi_t) \in \mathbb{R}^{d_x}$. At initialization ($t = 0$), no feature or object has been detected ($n_f = n_o = 0$, $d_u = d_e$), and the dynamic EKF full state is simply the initial state $\hat{x}_0 = x_0 \in \mathbb{R}^{d_x}$, with mean $\mu_0 \in \mathbb{R}^{d_x}$ and covariance $\Sigma_0 \in \mathbb{R}^{d_x \times d_x}$. Suppose, at some time $t$, the running cost $c_{dEKF,t,0} : \mathbb{R}^{d_{\mu'}} \rightarrow \mathbb{R}$ is:

$$c_{dEKF,t,0} = \|\hat{x}_t - \mu_t\|^2_{\Sigma_t}^{-1},$$

where $\hat{x}_t \in \mathbb{R}^{d_{\mu'}}$ denotes the EKF full state at time $t$, as described in the paragraphs above, with mean $\mu_t \in \mathbb{R}^{d_{\mu'}}$ and symmetric positive definite covariance matrix $\Sigma_t \in \mathbb{R}^{d_{\mu'} \times d_{\mu'}}$.

Let $N_f := n_f + \sum_{\alpha=1}^{n_o} n_o(\alpha)$ denote the total number of features (static and moving) tracked at time $t$. First, the feature augmentation step affixes new features’ maximum a posteriori position estimates, denoted $f_{n_f+N_f+n_f'+1}, \cdots, f_{n_f+N_f+n_f'+n_f'} \in \mathbb{R}^{d_f}$ to the EKF full state $\hat{x}_t$, and updates the mean and covariance of the full state. These new features may belong to static landmarks, previously detected moving objects, or new, previously undetected moving objects. Feature measurements $z_{t,n_f+N_f+1}, \cdots, z_{t,n_f+N_f+n_f'} \in \mathbb{R}^{d_z}$ are incorporated by adding measurement residuals to the current running cost $c_{dEKF,t,0}$, resulting in a new cost $c_{dEKF,t,1} : \mathbb{R}^{d_{\mu'}+(n_f+N_f')} \rightarrow \mathbb{R}$:

$$c_{dEKF,t,1}((\hat{x}_t, f_{t,n_f+N_f+1}, \cdots, f_{t,n_f+N_f+n_f'})) := \|\hat{x}_t - \mu_t\|^2_{\Sigma_t}^{-1} + \sum_{k=N_f+n_f'}^{N_f+n_f'} ||z_{t,k} - h(x_t, f_{t,k})||^2_{\Sigma_t}^{-1}. $$

Thus, $c_{dEKF,t,1}((\hat{x}_t, f_{t,n_f+N_f+1}, \cdots, f_{t,n_f+N_f+n_f'}))$ incorporates new feature positions to $\hat{x}_t$, and constrains it using feature measurements residuals. A Gauss-Newton step then updates the mean $\mu_t \in \mathbb{R}^{d_{\mu'}+n_f'd_{f}}$ and covariance $\Sigma_t \in \mathbb{R}^{(d_{\mu'}+n_f'd_{f}) \times (d_{\mu'}+n_f'd_{f})}$ for $\hat{x}_t$, resulting in a new cost:

$$c_{dEKF,t,2}((\hat{x}_t)) := \|\hat{x}_t - \mu_t\|^2_{\Sigma_t}^{-1}.$$
\[ \tilde{x}_t, \text{ resulting in a new cost } c_{dEKF,t,4}(\tilde{x}_t) : \mathbb{R}^{d_{\mu}+n_o n_d} \to \mathbb{R} : \\
\quad c_{dEKF,t,4}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1}. \]

We adjoin the new moving object poses \((\xi_{t,1}, \ldots, \xi_{t,n_o})\) to \(\tilde{x}_t\) (or record and drop them), then rearrange the components of the full state \(\tilde{x}_t\) so that each new moving object pose is stored alongside previously tracked poses for the same object. This restores the full state \(\tilde{x}_t\) to the form \((x_t, f(s), f(m), \xi_t) \in \mathbb{R}^{d_m}\), as introduced previously in (4).

Next, the static feature update step uses measurements of features contained in \(\tilde{x}_t\) to update the mean and covariance of \(\tilde{x}_t\). More precisely, measurements \(\tilde{x}_{t,1}, \ldots, \tilde{x}_{t,n_f} \in \mathbb{R}^{d_f}\), of the \(n_f\) static features \(f_{t,1}^{(s)}, \ldots, f_{t,n_f}^{(s)} \in \mathbb{R}^{d_f}\) currently tracked in \(\tilde{x}_t\), are introduced by incorporating associated measurement residuals to the running cost, resulting in a new cost \(c_{dEKF,t,5} : \mathbb{R}^{d_{\mu}+n_o d_f} \to \mathbb{R} : \\
\quad c_{dEKF,t,5}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1}. \]

A Gauss-Newton step then constructs an updated mean \(\mu_t \in \mathbb{R}^{d_{\mu}}\) and covariance \(\Sigma_t \in \mathbb{R}^{d_{\mu} \times d_{\mu}}\) for \(\tilde{x}_t\), resulting in a new cost \(c_{dEKF,t,6} : \mathbb{R}^{d_{\mu}} \to \mathbb{R} : \\
\quad c_{dEKF,t,6}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1}, \)

of the form of \(c_{dEKF,t,0}\).

The smoothing update step then updates the three most recent tracked dynamic object poses, denoted \(\{\xi_t, \alpha : \tau \in \{t-2, t-1, t\}, \alpha \in \{n_o\}\}\), by ensuring that the object’s motion from time \(t - 2\) to time \(t - 1\) does not deviate significantly from its motion from time \(t - 1\) to time \(t\). This regularization process ensures that the estimated trajectories of the moving objects are smooth enough to be physically feasible. To this end, we define a new cost \(c_{dEKF,t,7} : \mathbb{R}^{d_{\mu}} \to \mathbb{R} : \\
\quad c_{dEKF,t,7}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1}. \]

We then apply a Gauss-Newton step to update the mean \(\mu_t \in \mathbb{R}^{d_{\mu}}\) and covariance \(\Sigma_t \in \mathbb{R}^{d_{\mu} \times d_{\mu}}\), resulting in a new cost:

\[ c_{dEKF,t,8}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1}. \]

that has the form of the original cost \(c_{dEKF,t,0}\).

Finally, the state propagation step advances the EKF full state forward in time, via the EKF state propagation map \(g : \mathbb{R}^{d_k} \to \mathbb{R}^{d_k}\). To pass \(\tilde{x}_t\) forward to \(\tilde{x}_{t+1}\), we absorb the dynamics residual into the running cost, resulting in a new cost \(c_{dEKF,t,9} : \mathbb{R}^{d_{\mu}} \to \mathbb{R} : \\
\quad c_{dEKF,t,9}(\tilde{x}_t, x_{t+1}) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1} + \|x_{t+1} - g(x_t)\|_\Sigma^{-1}\Sigma^{-1}, \]

i.e., \(c_{dEKF,t,9}\) appends the new state \(x_{t+1} \in \mathbb{R}^{d_x}\) to \(\tilde{x}_t\), while adding a new cost encoded by the dynamics residuals.

The algorithm then applies a marginalization step, with \(\tilde{x}_{t,K} := (x_{t+1}, f(s), f(m), \xi_t) \in \mathbb{R}^{d_{\mu}}\), and \(\tilde{x}_{t,M} := x_t \in \mathbb{R}^{d_x}\), to remove the previous state \(x_t \in \mathbb{R}^{d_x}\) from the running cost. This step produces a mean \(\mu_{t+1} \in \mathbb{R}^{d_{\mu}}\) and a covariance \(\Sigma_{t+1} \in \mathbb{R}^{d_{\mu} \times d_{\mu}}\) for the new EKF full state, \(\tilde{x}_{t+1} := \tilde{x}_{t,K} = (x_{t+1}, f(s), f(m), \xi_t)\). The running cost is updated to \(c_{dEKF,t+1,0} : \mathbb{R}^{d_{\mu}} \to \mathbb{R}\), defined by:

\[ c_{dEKF,t+1,0}(\tilde{x}_{t+1}) := \|\tilde{x}_{t+1} - \mu_{t+1}\|_\Sigma^{-1}\Sigma^{-1}, \]

which assumes the form of \(c_{dEKF,t,0}\).

The theorems below establish the mathematical equivalence of the five steps of the dynamic EKF, as presented above in our optimization framework (Alg. 1), to those presented in the extension standard EKF SLAM algorithm to a dynamic setting (Alg. 2). Theorem statements and proofs concerning the equivalence of the feature augmentation, feature update, and state propagation steps are identical to those in the static EKF-SLAM case, and are omitted for brevity. For more details, please see Appendix A, or [1], Theorems 5.1-5.3.

**Theorem 4.1:** The dynamic object pose augmentation step of standard-formulation dynamic EKF SLAM (Alg. 4) is equivalent to applying a Gauss-Newton step to \(c_{dEKF,t,3} : \mathbb{R}^{d_{\mu}+n_o d_f} \to \mathbb{R}\), with:

\[ c_{dEKF,t,3}(\tilde{x}_t, \xi_{t,1}, \ldots, \xi_{t,n_o}) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1} + \sum_{\alpha=1}^{n_o} \sum_{k=1}^{n_f} \|f_{\alpha,k}^{(m)} - g^{(\alpha)}(\xi_{t,\alpha}, f_{0,\alpha,k})\|_\Sigma^{-1}\Sigma^{-1}. \]

when \(\Sigma_\xi\) is a diagonal matrix.

**Proof:** Please see Appendix A in the extended version of this paper [13].

**Theorem 4.2:** The smoothing update step of standard-formulation dynamic EKF SLAM (Alg. 6) is equivalent to applying a Gauss-Newton step to \(c_{dEKF,t,7} : \mathbb{R}^{d_{\mu}} \to \mathbb{R}\), with:

\[ c_{dEKF,t,7}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_\Sigma^{-1}\Sigma^{-1} + \sum_{\alpha=1}^{n_o} \|s(\xi_{t-2,\alpha}, \xi_{t-1,\alpha}, \xi_{t,\alpha})\|_\Sigma^{-1}\Sigma^{-1}. \]

**Proof:** Please see Appendix A in the extended version of this paper [13].

**V. EXPERIMENTS**

To illustrate the estimation accuracy and mapping precision of the dynamic EKF algorithm presented above, we constructed a simulated driving scenario (Figure 1). In the scenario, the ego vehicle navigates alongside two other vehicles (Agents 1, 2) and a pedestrian (Agent 3) on a highway with three lanes, while simultaneously tracking the positions of non-ego vehicles and fixed landmarks in its surroundings. As time progresses, the vehicles change lanes and adjust their velocities. Object motion is sampled every 0.5 s for 60 s to form a ground truth dataset.

To test our dynamic EKF algorithm, we performed Monte Carlo experiments on the simulated driving setting described above. For each combination of the three odometry and

1639
Fig. 1. Schematic for the ground truth trajectory of the driving example. The ego vehicle (red) navigates and runs dynamic EKF SLAM along a kilometer-long stretch of highway, from left to right, alongside two other vehicles (green, blue) and a jaywalking pedestrian (purple). Static landmarks (yellow) are scattered throughout the scene. Initial feature estimates of each moving object are plotted, but are not clearly visible due to the schematic scale.

| Noise Level | Data | Ego Poses | Static Features | Agent 1 Features | Agent 2 Features | Agent 3 Features | Agent 1 Poses | Agent 2 Poses | Agent 3 Poses |
|------------|------|-----------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| $\Sigma_{w,1}$, $\Sigma_{v,1}$ | $x$ (m) | 0.008 | 0.014 | 0.014 | 0.014 | 0.001 | 0.001 | 0.001 |
| | $y$ (m) | 0.032 | 0.028 | 0.003 | 0.001 | 0.025 | 0.028 | 0.035 | 0.006 |
| | $\theta$ (rad) | 0.000 | N/A | N/A | N/A | N/A | N/A | 0.001 | 0.001 |
| $\Sigma_{w,1}$, $\Sigma_{v,2}$ | $x$ (m) | 0.035 | 0.062 | 0.062 | 0.062 | 0.063 | 0.082 | 0.002 | 0.012 |
| | $y$ (m) | 0.061 | 0.059 | 0.007 | 0.003 | 0.052 | 0.054 | 0.066 | 0.012 |
| | $\theta$ (rad) | 0.000 | N/A | N/A | N/A | N/A | N/A | 0.004 | 0.015 |
| $\Sigma_{w,1}$, $\Sigma_{v,3}$ | $x$ (m) | 0.089 | 0.155 | 0.157 | 0.157 | 0.156 | 0.005 | 0.004 | 0.006 |
| | $y$ (m) | 0.152 | 0.157 | 0.168 | 0.009 | 0.138 | 0.137 | 0.166 | 0.032 |
| | $\theta$ (rad) | 0.000 | N/A | N/A | N/A | N/A | N/A | 0.011 | 0.011 |
| $\Sigma_{w,2}$, $\Sigma_{v,1}$ | $x$ (m) | 0.088 | 0.121 | 0.121 | 0.121 | 0.122 | 0.002 | 0.002 | 0.003 |
| | $y$ (m) | 0.072 | 0.053 | 0.004 | 0.004 | 0.046 | 0.065 | 0.080 | 0.014 |
| | $\theta$ (rad) | 0.000 | N/A | N/A | N/A | N/A | N/A | 0.001 | 0.001 |
| $\Sigma_{w,2}$, $\Sigma_{v,2}$ | $x$ (m) | 0.028 | 0.048 | 0.049 | 0.048 | 0.050 | 0.003 | 0.003 | 0.004 |
| | $y$ (m) | 0.088 | 0.0074 | 0.007 | 0.005 | 0.065 | 0.078 | 0.004 | 0.018 |
| | $\theta$ (rad) | 0.000 | N/A | N/A | N/A | N/A | N/A | 0.004 | 0.004 |
| $\Sigma_{w,2}$, $\Sigma_{v,3}$ | $x$ (m) | 0.103 | 0.181 | 0.182 | 0.182 | 0.182 | 0.005 | 0.005 | 0.007 |
| | $y$ (m) | 0.200 | 0.196 | 0.020 | 0.010 | 0.173 | 0.180 | 0.217 | 0.041 |
| | $\theta$ (rad) | 0.000 | N/A | N/A | N/A | N/A | N/A | 0.012 | 0.012 |
| $\Sigma_{w,3}$, $\Sigma_{v,1}$ | $x$ (m) | 0.019 | 0.017 | 0.019 | 0.016 | 0.021 | 0.009 | 0.008 | 0.012 |
| | $y$ (m) | 0.287 | 0.232 | 0.016 | 0.012 | 0.202 | 0.259 | 0.317 | 0.058 |
| | $\theta$ (rad) | 0.001 | N/A | N/A | N/A | N/A | N/A | 0.001 | 0.001 |
| $\Sigma_{w,3}$, $\Sigma_{v,2}$ | $x$ (m) | 0.029 | 0.043 | 0.044 | 0.044 | 0.045 | 0.010 | 0.009 | 0.013 |
| | $y$ (m) | 0.226 | 0.164 | 0.015 | 0.009 | 0.144 | 0.200 | 0.246 | 0.044 |
| | $\theta$ (rad) | 0.001 | N/A | N/A | N/A | N/A | N/A | 0.004 | 0.004 |
| $\Sigma_{w,3}$, $\Sigma_{v,3}$ | $x$ (m) | 0.095 | 0.161 | 0.161 | 0.010 | 0.014 | 0.010 | 0.010 | 0.014 |
| | $y$ (m) | 0.312 | 0.264 | 0.027 | 0.014 | 0.233 | 0.277 | 0.340 | 0.061 |
| | $\theta$ (rad) | 0.001 | N/A | N/A | N/A | N/A | N/A | 0.012 | 0.013 |

TABLE I

Root-mean-squared translation ($x$, $y$) and rotation ($\theta$) error on our simulated driving dataset. Noise level settings correspond to different choices of $\Sigma_w$ and $\Sigma_v$ (with $\Sigma_0 = \Sigma_w$), as defined in 5, 6, 7. Root-mean-squared errors are averaged over 25 experiments for each noise setting.

![Diagram](image_url)

image measurement noise covariance levels given below, we simulated the ground truth trajectory 25 times, each with independently generated errors:

$$\Sigma_{w,1} := \begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-8} \end{bmatrix}, \Sigma_{v,1} := 10^{-6} \cdot I_{2 \times 2}, \quad (5)$$

$$\Sigma_{w,2} := \begin{bmatrix} 10^{-5} & 0 & 0 \\ 0 & 10^{-5} & 0 \\ 0 & 0 & 10^{-7} \end{bmatrix}, \Sigma_{v,2} := 10^{-5} \cdot I_{2 \times 2}, \quad (6)$$

$$\Sigma_{w,3} := \begin{bmatrix} 10^{-4} & 0 & 0 \\ 0 & 10^{-4} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix}, \Sigma_{v,3} := 10^{-4} \cdot I_{2 \times 2}, \quad (7)$$

Here, $I_{2 \times 2}$ denotes the $2 \times 2$ identity matrix, and each entry of the above matrices has unit $m^2$ (meters squared). We then applied dynamic EKF SLAM to recover the ground truth trajectory, and computed the resulting root-mean-squared error for each noise level (Table I). By dropping past poses of all moving objects, each simulation can be run on a standard, single-threaded laptop in under 37 ms. We observed that our estimation accuracy decreases gracefully as the noise level
increased. For more details regarding the simulation setup and results, please see Appendix C.

Algorithm 3: Dynamic EKF, Feature Augmentation
Sub-block.

Data: Current EKF state \( \tilde{x}_t \in \mathbb{R}^{d_x} \), with mean \( \mu_t \), and covariance \( \Sigma_t \); number of static features \( n_f \in \mathbb{N} \); total number of features on moving objects \( N_f \in \mathbb{N} \); measurements of new features \( z_{k,t} \in \mathbb{R}^{d_z} \), \( k \in \{n_f + N_f + 1 : n_f + N_f + n_f' + N_f' \} \); measurement map \( h : \mathbb{R}^{d_z} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z} \); inverse measurement map \( \ell : \mathbb{R}^{d_z} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_f} \), with \( z_{k,t} = h(\tilde{x}_t, (x_{t,k}, z_{k,t})) \forall x_t \in \mathbb{R}^{d_x} \).

Result: Updated number of static features \( n_f \); updated total number of features on moving objects \( N_f \); updated EKF state dimension \( d_v \), updated EKF state mean \( \mu_t \in \mathbb{R}^{d_v} \), covariance \( \Sigma_t \in \mathbb{R}^{d_v \times d_v} \).

1. \((\mu_t, x, \mu_t, x) \leftarrow \mu_t, x \in \mathbb{R}^{d_x} \).
2. \( \ell(\mu_t, x, z_{k,t}, n_f + N_f + 1, \ldots, z_{k,t}, n_f + N_f + N_f') \leftarrow \ell(\mu_t, x, z_{k,t}, n_f + N_f + 1, \ldots, \ell(\mu_t, x, z_{k,t}, n_f + N_f + N_f')) \in \mathbb{R}^{d_v + N_f + N_f'} \), \( (x_{t,k}, z_{k,t}) \in \mathbb{R}^{d_x} \), \( (n_f + N_f) \) \( d_v \times d_v \) \( \mu_t \leftarrow \mu_t, x \in \mathbb{R}^{d_x} \).
3. \( \Sigma_t \leftarrow \Sigma_t, x \in \mathbb{R}^{d_v \times d_v} \).
4. \( L_k \leftarrow \frac{d \ell}{d x} \mu_t, \in \mathbb{R}^{(n_f + N_f') d_v \times d_x} \).
5. \( S \leftarrow \frac{d \ell}{d x} \mu_t, \in \mathbb{R}^{(n_f + N_f') d_v \times d_x} \).
6. \( \Sigma_t \leftarrow \Sigma_t, x \in \mathbb{R}^{d_v \times d_v} \).
7. \( L_k \leftarrow L_k, z_{k,t} \in \mathbb{R}^{(n_f + N_f') d_v \times (n_f + N_f') d_z} \).
8. \( \tilde{\Lambda} \leftarrow \left[ \begin{array}{c} \Sigma_{1,xx} \Sigma_{1,xe} \Sigma_{1,ex} \Sigma_{1,ee} \Sigma_{2,xx} \Sigma_{2,xe} \Sigma_{2,ex} \Sigma_{2,ee} \Sigma_{3,xx} \Sigma_{3,xe} \Sigma_{3,ex} \Sigma_{3,ee} \end{array} \right] \in \mathbb{R}^{(d_v + (n_f + N_f') d_f) \times (d_v + (n_f + N_f') d_f)} \).
9. \( \tilde{x}_t \leftarrow (\tilde{x}_t, f_{t,N_f + 1, \ldots, f_{t,N_f + N_f + N_f'}}) \in \mathbb{R}^{d_v + (n_f + N_f') d_f} \).
10. \( n_f \leftarrow n_f + N_f \).
11. \( N_f \leftarrow N_f + N_f' \).
12. \( d_v \leftarrow d_v + (n_f + N_f') d_f \).
13. Reorder variables in \( \tilde{x}_t \) to restore the variable ordering of (4).
14. return \( N_f, d_v, \mu_t, \Sigma_t \).

VI. CONCLUSION AND FUTURE WORK

In this tutorial, we extended the unifying optimization-based SLAM backend framework in [1] to environments with moving objects. We use this framework to describe the backends of recently proposed dynamic SLAM algorithms [5, 6, 8, 9, 12, 13]. To establish a rapprochement with filtering-based SLAM methods, we apply an aggressive marginalization scheme in our framework to derive the dynamic EKF-SLAM algorithm, which we prove to be mathematically identical to the straightforward extension of the conventional EKF-SLAM algorithm to environments with moving objects. Simulation results indicate that dynamic EKF-SLAM performs well in pose estimation, as well as static and dynamic feature tracking.

The formulation presented in this tutorial can be refined in several ways. First, we are eager to deploy our framework on real-world data, and explore the tradeoffs inherent in different design choices. Second, many robotics applications require topological and/or semantic maps of dynamic scenes, in addition to purely metric information. Thus, dynamic object features should be explicitly encoded in our framework as lower-dimensional semantic representations, e.g., bounding boxes, as is done in existing methods [5, 8, 9, 12]. Third, robust formulations of our framework must allow for the implementation of multi-hypothesis SLAM backends, to reliably safeguard against ambiguous data associations or high outlier densities [14, 15]. Finally, guidelines for selecting appropriate modeling choices within our dynamic SLAM framework depend heavily on downstream tasks, such as autonomous navigation in the presence of multiple agents [16, 17]. It is of interest to design complete autonomy stacks that fully harness the flexibility of our dynamic SLAM framework for estimation, prediction, and planning in challenging robotics tasks.

REFERENCES

[1] Amay Saxena, Chih-Yuan Chiu, Ritika Shrivastava, Joseph Menke, and Shankar Sastry. “Simultaneous Localization and Mapping: Through the Lens of Nonlinear Optimization”. In: IEEE Robotics and Automation Letters 7.3 (2022), pp. 7148–7155. DOI: 10.1109/LRA.2022.3183409.

[2] J.J. Leonard and H.F. Durrant-Whyte. “Simultaneous Map Building and Localization for an Autonomous Mobile Robot”. In: 3 (1991), pp. 1442–7.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”. In: IEEE T-RO 32.6 (2016), pp. 1309–1332.

[4] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. “Simultaneous Localization, Mapping and Moving Object Tracking”. In: The International Journal of Robotics Research 26.9 (2007), pp. 889–916. DOI: 10.1177/0278364907081229. eprint: https://doi.org/10.1177/0278364907081229.

[5] Shichao Yang and Sebastian Scherer. “CubeSLAM: Monocular 3-D Object SLAM”. In: IEEE Transactions on Robotics 35.4 (2019), pp. 925–938. DOI: 10.1109/TRO.2019.2909168.

[6] Jiahui Huang, Sheng Yang, Zishuo Zhao, Yu-Kun Lai, and Shi-Min Hu. “ClusterSLAM: A SLAM Backend for Simultaneous Rigid Body Clustering and Motion Estimation”. In: (2019).

[7] Berta Bescós, José M. Fábal, Javier Civera, and José Neira. “DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes”. In: Robotics and Automation Letters RA-L. (2018).

[8] Berta Bescós, Carlos Campos, Juan D. Tardós, and José Neira. “DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM”. In: IEEE Robotics and Automation Letters 6 (2021), pp. 5191–5198.

[9] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. “VDO-SLAM: A Visual Dynamic Object-aware SLAM System”. In: (2020). arXiv: 2005.11052 [cs.RO].

[10] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic Robotics. The MIT Press, 2005.

[11] Raúl Mur-Artal and Juan D. Tardós. “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”. In: IEEE Transactions on Robotics 33.5 (2017), pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103.
APPENDIX

The ArXiv version of this paper, which contains the appendix, is found here: http://arxiv.org/abs/2207.05043 [13]. The author will ensure that the link stays active.