

Better Together: Online Probabilistic Clique Change Detection in 3D Landmark-Based Maps

Samuel Bateman, Kyle Harlow, and Christoffer Heckman*

Abstract—Many modern simultaneous localization and mapping (SLAM) techniques rely on sparse landmark-based maps due to their real-time performance. However, these techniques frequently assert that these landmarks are fixed in position over time, known as the *static-world assumption*. This is rarely, if ever, the case in most real-world environments. Even worse, over long deployments, robots are bound to observe traditionally static landmarks change, for example when an autonomous vehicle encounters a construction zone. This work addresses this challenge, accounting for changes in complex three-dimensional environments with the creation of a probabilistic filter that operates on the features that give rise to landmarks. To accomplish this, landmarks are clustered into cliques and a filter is developed to estimate their persistence jointly among observations of the landmarks in a clique. This filter uses estimated spatial-temporal priors of geometric objects, allowing for dynamic and semi-static objects to be removed from a formally static map. The proposed algorithm is validated in a 3D simulated environment.

I. INTRODUCTION

As robots are deployed into real-world environments more often and for longer durations, it becomes increasingly important for them to be able to navigate in dynamic environments. Frequently, this is performed by providing spatial awareness through perception techniques such as simultaneous localization and mapping (SLAM), wherein map parameters are jointly estimated with robot pose in an online probabilistic framework. While SLAM is a very powerful tool for inference over core quantities for robotic perception, it can yield biased results due to core assumptions made, such as that the landmarks that compose the map are static in time. These assumptions are present at the level of the formalism employed to represent the problem. This work focuses on the limitations of this so-called *static-world assumption* and how to extend the formalisms and algorithms involved in solving the SLAM problem in dynamic environments.

In dynamic-world deployments, such as those in warehouses, buildings, and on streets, one of the challenges that can lead to catastrophic failure in SLAM is the problem of associating features derived from data to landmark quantities to be estimated, frequently termed the “data association problem.” Data association matches a measurement from a sensor to a state parameter to be estimated, e.g. a visual feature such as a corner in an image might be associated with the corner of an office chair [1]–[7]. However, if that



Fig. 1. Two images taken from the Oxford RobotCar dataset [9] at similar locations approximately one month apart have drastic differences in appearance. Landmarks a and c have both shifted to a' and c' respectively, with new vehicles parked in their previous locations. Shifts in these landmarks from associating new measurements improperly could cause a localization system to estimate that it is in a different position. In addition, landmarks b and d have disappeared, while landmark e has been added.

chair moves for any reason, the measurement of the same corner may be associated with the previous landmark position, rather than identifying that the observed feature should be used to generate a new landmark independent of any landmarks already in the map. Without properly identifying these associations, the SLAM solution will at best incur error, and at worst diverge entirely [8]. This is particularly problematic in long-term deployments, in which landmarks in the map are likely to vary in position and possibly even appearance over time. A pertinent example from the Oxford RobotCar Dataset [9] is shown in Figure I. While there are a number of potential data association techniques [5], [10], most do not account for dynamic environments, leading to almost all SLAM approaches making a static world assumption. While this assumption holds for short deployments in controlled environments, it is clear that long-term real-world deployments will violate this assumption.

This work attempts to account for dynamic maps by applying a Bayesian filter over the persistence of map parameters at the core of modern SLAM implementations, inspired by previous persistence filters developed by [6], [11]. Further, this work demonstrates the advantages of conditioning over arbitrary groupings of highly correlated landmarks as well as the objects they are associated to while accounting for differing sensing modalities and the degradation of said sensing modalities, which is unique to this work relative to the previous work in Bayesian landmark persistence filters such as [6], [11].

II. PRIOR WORK

The problem of deploying robots in dynamic environments has been explored since the formulation of SLAM [12]. The

All authors are with the Department of Computer Science, University of Colorado, Boulder, CO 80309, USA.

*Corresponding author. E-mail: christoffer.heckman@colorado.edu.

primary challenges of dynamic SLAM, detecting changes and updating the map, have been addressed several ways. Early approaches, using the extended Kalman filter [13] and particle filter SLAM [14], focused on separating the map into static and dynamic portions, thereby reducing the problem back to a static SLAM problem [15]. Other approaches focused on tracking the dynamic objects using a separate algorithm such as a human vision model [16]. In sparse landmark-based SLAM, early work focused on updating maps hierarchically [17]. Others focused on randomly selecting portions of the map to revisit throughout a run to update the prior map [18], [19]. Another set of proposed solutions involved increasing the storage of the map, such as in [20] where a multiple map approach was developed. The goal of these approaches was to store multiple maps and select the map that best fits the current sensor measurements. Another multi-map approach stored experiences tied to specific places, and then selected the best possible set of experiences for navigation [21]. While this work was robust to changes in lighting, weather, and seasonal changes, it suffered in areas where the experience was consistently changing, such as in parking lots.

Many other approaches focus on specific sensor modalities: [22] focused primarily on monocular SLAM for dynamic environments, while [23] used a similar approach including multiple cameras with dynamic priors assigned to features tracked by each camera. Additional studies have focused primarily on laser scanners, but relied on other sensors to help reduce pose error in the presence of dynamic objects [24]. Many of these sensor-based approaches lack the adaptability of an approach that is based purely on the parameters present in the SLAM formulation.

Some more recent approaches have begun to address these problems by tracking dynamic features explicitly or jointly estimating objects' motion relative to the current platform [25], [26]. In our work, we focus instead on filtering the features based on an initial prior accounting for the 3D semantic spatial relationships between collections of correlated features. To do this, we propose a new approach in the spirit of [6], [11] which tracks the joint persistence probability of one of these collections of landmarks and removes them from the factor graph when the posterior probability reaches a tunable threshold [27].

III. METHODOLOGY

Formulating map change detection in relation to a probabilistic SLAM problem formulation begins with introducing the SLAM problem. In its simplest form, SLAM gives the robots' pose $\mathbf{x}_p \in \mathbb{SE}(3)$ and map landmarks $\mathbf{x}_l \in \mathbb{R}^3$ based on measurements Z which can include odometry information from the robot as well as landmark detections, defined by a sensor model. With this information, the goal of maximum *a posteriori* (MAP) estimate SLAM is to maximize the posterior $p(\mathcal{X}|Z)$ where $\mathcal{X} = [\mathbf{x}_p, \mathbf{x}_l]$. While this is possible given a limited number of measurements, with any reasonably-sized map this would require summing over all possible measurement associations to obtain the full

probability distribution, leading to an intractable calculation [28]. As such, in most problems possible measurement associations are restricted through a data association step as mentioned in Section I. This is done by forming a vector of data association hypotheses J . Ideally, this vector is developed by considering the most likely data associations given our measurements Z and calculating $\text{argmax}_J P(J|Z)$. However, matching a sequence of measurements to a series of discrete landmarks, while also accounting for spurious measurements, is a combinatorially hard problem [5]. As such, most SLAM algorithms instead use a known data association technique such as joint compatibility branch-and-bound (JCBB) [5] or maximum likelihood data association [10] based on sensor model-derived landmark similarity metrics. For a survey of data association techniques, see [7]. Thus, the typical SLAM problem with the static-world assumption is $\text{argmax}_{\mathcal{X}} P(\mathcal{X}|J, Z)$ where J is now the vector of data associations assigning a single measurement to a single landmark. This formulation is extended by the introduction of landmark cliques, denoted τ , any one of which is a collection of correlated landmarks, e.g. those on the surface of a rigid body.

A similar approach was taken in [11], however the formulation is expanded to introduce the usage of arbitrary, rigid cliques to the previous methodology. The persistence of a clique is represented as a time-varying, binary random variable $\theta_\tau^t \in \{0, 1\}$ where $t \in [0, \infty)$ and $t = 0$ is the first time the clique was observed. Ideally, a joint calculation of the feature association hypothesis J and the existence of a landmark in a clique θ^t given a set of measurements Z would be possible, but the same problems of intractability and spurious measurements calculating $P(\theta^t, J|Z)$ exist as they would in calculating $P(J|Z)$. Instead, the landmark persistence is conditioned on the data associations, calculating $\text{argmax}_{\theta^t} P(\theta^t|J)$ by estimating J using a known data association technique and then estimating θ^t . From these results, the SLAM algorithm decides whether or not to continue integrating measurements of a particular element of J based on the probability of θ_τ , or to introduce a new feature into the map, while removing the old landmark, similar in spirit to the ideas of [11] and [6]. In this way, $\mathcal{X}$ is augmented as $\mathcal{X} = [\mathbf{x}_p, \mathbf{x}_l, \theta^t]$ and jointly estimated in $\text{argmax}_{\mathcal{X}} P(\mathcal{X}|J, Z)$ as before.

A. Clique Persistence

In [6], [11], it was assumed that the k^{th} feature in a map M has a survival time $T_k \in [0, \infty)$ such that the feature should be expected to no longer persist for $t > T_k$. This connection between survival time and persistence can be formalized as

$$P(\theta_k^t|J) = P(T_k \geq t_n | J_k^{1:N}), \quad (1)$$

where $j_k^{t_i}$ is the random variable describing the state of a detection of a feature $k \in M$ at possible observation time t_i , denoting the sequence of observations of k as

$$J_k^{1:N} \triangleq \{j_k^{t_i}\}_{i=1}^N. \quad (2)$$

However, the formulation presented in these works does not account for the fact that, in 3D, most features on a given object are not observable a large part of the time, even when they are within our perceptive field. The clique survival time

$$T_\tau \sim P_{T_\tau}(\cdot), \quad (3)$$

is used to address this, where $P_{T_\tau}(\cdot)$ is approximated by a prior distribution over the survival time T_τ , similar to the feature survival time in [6]. To account for the partial observability of landmarks, a similar relation is made to the clique survival time T_τ and the previously introduced clique persistence random variable θ_τ^t as was done in Eq. (1); that is,

$$P(\theta_\tau^t | J) = P(T_\tau \geq t_n | J_\tau^{1:N}), \quad (4)$$

where $J_\tau^{1:N}$ is the shorthand for $J_{k \in \tau}^{1:N}$. This carries with it the assumption that $\forall i \in \tau, T_i = T_\tau$, which simply states that all landmarks within a clique are likely to persist so long as the clique persists. This deceptively small change has a major implication: an individual landmark may be observed for only a few frames of observation, but some non-empty subset of the clique of landmarks will almost always be observable at every timestep of observation. This also assumes that clique persistence is independent of the observations of other cliques. This assumption of independence is justified by the observation that clique survival time is modeled to be independent of each other clique survival time. Since observability of any given landmark is strongly correlated with its persistence (precluding missed and false detections which are addressed later), it seems reasonable that the observation of landmarks in other cliques would provide little extra information about clique persistence then can be provided by only the sequence of observations of the component landmarks of a given clique. While this may not be strictly true in an analytical sense, it is a necessary assumption to maintain tractability of the presented problem, as also identified by [11].

Using Eq. (4), Bayes' Rule is then employed to estimate the persistence of a clique at a given time t

$$P(T_\tau \geq t | J_\tau^{1:N}) = \frac{P(J_\tau^{1:N} | T_\tau \geq t_n) P(T_\tau \geq t)}{P(J_\tau^{1:N})}. \quad (5)$$

From here, all that remains is to estimate the likelihood and the evidence of the clique to estimate persistence.

B. Joint Detection Likelihood

In order to calculate the joint likelihood of observations of a clique, an assumption that, for $a, b \in \tau$

$$P(J_a^{1:N}, J_b^{1:N} | T_\tau \geq t_n) = P(J_a^{1:N} | T_\tau \geq t_n) P(J_b^{1:N} | T_\tau \geq t_n) \quad (6)$$

is made, which is to say that a sequence of detections for one landmark a is conditionally independent from each other landmark e.g. b , in the clique, given the persistence of the clique. With this, the joint clique likelihood can be

broken into the product of detection likelihoods of individual landmarks

$$P(J_\tau^{1:N} | T_\tau \geq t_n) = \prod_{k \in \tau} P(J_k^{1:N} | T_\tau \geq t_n). \quad (7)$$

Another assumption is that, given the persistence of a feature $k \in \tau$, the sequence of detections will be fully Markovian, as we do not incorporate information about the projected trajectory of the sensor. The definition of landmark likelihood follows as:

$$P(J_k^{1:N} | T_\tau \geq t_n) = \prod_{i=1}^N P(j_k^{t_i} | T_\tau \geq t_n), \quad (8)$$

which then leads to the fully expanded joint detection likelihood

$$\begin{aligned} P(J_\tau^{1:N} | T_\tau \geq t_i) &= \prod_{k \in \tau} P(J_k^{1:N} | T_\tau \geq t_i) \\ &= \prod_{k \in \tau} \prod_{i=1}^N P(j_k^{t_i} | T_\tau \geq t). \end{aligned} \quad (9)$$

By identifying that the t_N likelihood is in the expression for the t_{N+1} likelihood, the t_N expression can be factored out. The clique likelihood can be defined recursively as

$$P(J_\tau^{1:N+1} | T_\tau \geq t) = P(J_\tau^{1:N} | T_\tau \geq t_i) \prod_{k \in \tau} P(j_k^{N+1} | T_\tau \geq t), \quad (10)$$

requiring only the new measurements acquired at time t_{N+1} to update the clique likelihood.

Finally, the individual detection likelihoods $P(j_k | T_\tau)$ must be calculated in order to fully define the clique likelihood in Eq. (5). With a perfect sensor, this would simply be a binary random variable such that

$$P(j_k | T_\tau) = \begin{cases} 1, & T_\tau \geq t \\ 0, & T_\tau < t, \end{cases} \quad (11)$$

in the case where τ is expected to be observable by the sensor. This does not however account for the probability of falsely detecting a non-existent landmark, denoted P_F , or the probability of missing detection for an existing landmark, denoted P_M , which is particularly troublesome in 3D due to occlusion and sensor degradation, e.g. sensor range limitations. Previous works utilized a fixed constant for P_F and P_M , but this does not capture the complexity of real sensors in general. To remedy this, sensor degradation is modeled by making P_M a function of minimum clique landmark distance from the sensor, denoted $P_M(s)$. It is common for perception systems to have a maximum detection range $s_{\max}$ but for measurements to only be incorporated into a model, such as a SLAM system, up to a smaller threshold s_{obs} where sensor data is more reliable. This notion is used to define a range-based sensor degradation model $P_M(s)$ as follows:

$$P_M(s) = 1 - \exp(-s \cdot s_{\max} / s_{\text{obs}}). \quad (12)$$

With P_M and P_F defined, the formulation of the feature detection likelihood from [6]

$$P(j_k|T_\tau) = \begin{cases} P_M^{1-j_k^t}(1-P_M)^{j_k^t}, & T_\tau \geq t \\ P_F^{j_k^t}(1-P_F)^{1-j_k^t}, & T_\tau < t \end{cases} \quad (13)$$

is used in this filter.

C. Joint Evidence

The derivation of the joint clique evidence is inspired by [6]. This work capitalizes on the conditional independence of landmark detection within a clique given clique persistence. Because the clique likelihood defined in Eq. (7) is only updated at discrete times $\{t_i\}_{i=1}^N$, the likelihood is constant over the intervals in the set

$$\{[t_i, t_{i+1}) | \forall i \in \mathbb{Z}, 0 \leq i \leq N\}. \quad (14)$$

As in [6], the integral is simplified by defining $t_0 \triangleq 0$ and $t_{N+1} \triangleq \infty$. Thus the evidence over cliques is calculated as

$$\begin{aligned} P(J_\tau^{1:N}) &= \int_0^\infty P(J_\tau|T_\tau)P(T_\tau)dT_\tau \\ &= \sum_{i=0}^N \int_{t_i}^{t_{i+1}} P(J_\tau|T_\tau)p(T_\tau)dT_\tau \\ &= \sum_{i=0}^N \prod_{k \in \tau} P(J_k^{1:N}|T_\tau)[F_{T_\tau}(t_{N+1}) - F_{T_\tau}(t_N)], \end{aligned} \quad (15)$$

where $F_{T_\tau}(t)$ is the cumulative distribution function of $P_{T_\tau}(t)$ which is described in Eq. (3). In the spirit of [6], the clique evidence $P(J_\tau^{1:N})$ is broken into a lower partial sum

$$L(J_\tau^{1:N}) \triangleq \sum_{i=0}^{N-1} \prod_{k \in \tau} P(J_k^{1:N}|T_\tau)[F_{T_\tau}(t_{N+1}) - F_{T_\tau}(t_N)], \quad (16)$$

so that the evidence is defined recursively as

$$P(J_\tau^{1:N}) = L(J_\tau^{1:N}) + \left(\prod_{k \in \tau} P(J_k^{1:N}|T_\tau) \right) [1 - F_{T_\tau}(t_N)]. \quad (17)$$

Then $L(J_\tau^{1:N})$ is related to $L(J_\tau^{1:N+1})$ as

$$\begin{aligned} L(J_\tau^{1:N+1}) &= \left(\prod_{k \in \tau} P_F^{j_k^{t_{N+1}}} (1 - P_F^{j_k^{t_{N+1}}}) \right) \\ &\quad (L(J_\tau^{1:N}) + P(J_\tau^{1:N}|t_N \geq T_\tau)[F_T(t_{N+1}) - F_T(t_N)]), \end{aligned} \quad (18)$$

admitting the full recursive Bayesian estimation procedure as outlined in Algorithm 1.

D. False Negative Suppression

A major shortcoming of Bayesian landmark filters in 3D is that cliques, and their constituent landmarks, are only partially observable due to sensor degradation, map geometry, and map occlusion. However, in a joint filter formulation such as in [11] and here, the filter must be updated for all landmarks in a clique if any one of the constituent landmarks

Algorithm 1 Algorithm for recursively calculating joint clique filter (JCF) posterior persistence probability

Input: Feature detector error probabilities (P_M, P_F), cumulative clique prior F_{T_τ} , feature detector outputs $J_\tau^{t_i}$.

Output: Persistence beliefs on clique $P(\theta_\tau = 1 | J_\tau^{1:N})$ for $t \in [t_N, \infty)$.

Initialization: Set $t_0 \leftarrow 0$, $N \leftarrow 0$, $P(J_\tau^{1:0}|t_0) \leftarrow 1$, $L(J_\tau^{1:N}) \leftarrow 0$, $P(J_\tau^{1:N}) \leftarrow 1$

- 1: **while** $\exists$ new data $J_\tau^{t_{N+1}}$ **do**
- 2: Compute the partial clique evidence $L(J_\tau^{1:N+1})$ using (18)
- 3: Compute the clique likelihood $P(J_\tau^{1:N+1}|T_\tau \geq t)$ using (10)
- 4: Compute the clique evidence $P(J_\tau^{1:N+1})$ using (17)
- 5: $N \leftarrow N + 1$
- 6: Compute the posterior probability $P(T_\tau \geq t | J_\tau^{1:N})$ using (5)
- 7: **end while**

is visible. This works well when the clique is well within sensor range, but becomes problematic when only a few of the landmarks within a clique are possibly observable, such as when the clique is on the boundary of the observable area. The filter would then update, including in a number of negative detections for which there is no corresponding real information, driving the persistence probability down prematurely.

To remedy this, for all landmark filters studied in this work, a technique to suppress false negatives is introduced: Let $d(\mathbf{x}_p, \mathbf{x}_{l_k})$ be defined as the Euclidean distance between the positional component of $\mathbf{x}_p$ and $\mathbf{x}_{l_k}$. Let

$$\tau^* = \{k \in \tau | k \in \text{positive detections at } t_n\} \quad (19)$$

and pick δ to be a positive threshold ratio of positive detections to total expected detections. This parameter would likely be tuned for the application, however for an example and purposes of simulations, our δ was chosen arbitrarily to be 0.03. If the following 3 statements are true

1. $\forall k \in \tau, s_{\text{obs}} < d(\mathbf{x}_p, \mathbf{x}_{l_k})$,
2. $\exists k \in \tau$ st. $d(\mathbf{x}_p, \mathbf{x}_{l_k}) < s_{\text{max}}$, and
3. $\frac{|\tau^*|}{|\tau|} < \delta$,

then reject detections for all of τ at this time step. While deceptively simple, this can significantly improve the performance of the joint clique filter in 3D, as well as the joint feature filter from [11] and even the independent feature filter from [6], which eschews any inter-feature correlation model.

IV. EXPERIMENTS AND RESULTS

A. Simulations

To accurately test the relevant algorithms on realistic 3D data, a 3D simulation of arbitrary, periodic 3D trajectories for a sensor moving past a collection of 3D objects with oriented

Sensor Model	Metric	No Negative Suppression				Negative Suppression			
		IFF	JFF	JCF	JCFR	IFF	JFF	JCF	JCFR
Lidar	Accuracy	0.715	0.621	0.850	0.942	0.762	0.618	0.978	0.987
	MES/S	0.144	0.099	0.099	0.308	0.211	0.012	0.479	0.788
Camera	Accuracy	0.672	0.587	0.793	0.844	0.731	0.603	0.936	0.916
	MES/S	0.254	0.025	0.179	0.170	0.336	0.037	0.326	0.300

TABLE I

IFF: Independent Feature Filter [6], JFF: Joint Feature Filter [11], JCF: Joint Clique Filter (Ours), JCFR: Joint Clique Filter with range sensor degradation (Ours). MES/S is the mean estimated survival time over true survival time which compares the rate at which landmarks were removed to when they should have been removed (Higher is Better). Each of the tested filters attempts to filter changed landmarks from the factor graph. When comparing the JCF/JCFR with the IFF, it is of note that the IFF can do arbitrarily badly depending on the environmental occlusion of landmarks, whereas the JCF and JCFR integrate a number of landmark measurements from a single clique at each time step and are therefore much more robust to occlusion in 3D environments. For the purposes of fair evaluation, this problem of the IFF is not explored in these results. The JFF can be directly compared to our methods.

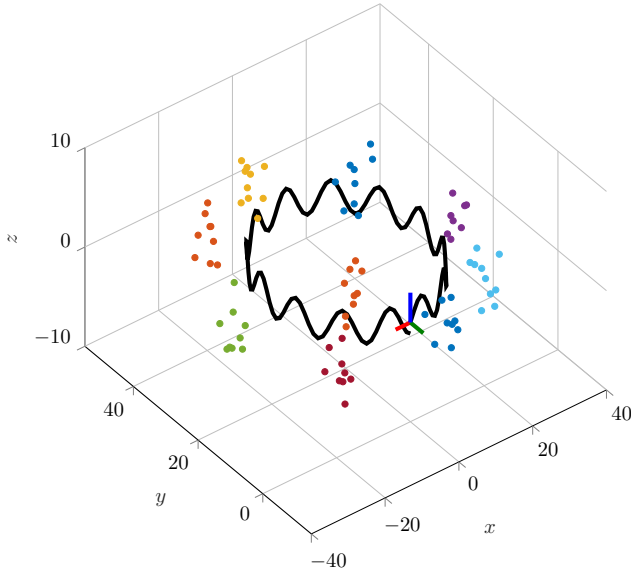


Fig. 2. Example single-period sensor trajectory simulation with all cliques visible. Most landmarks would not be visible to either of the sensor modalities at the shown time step due to their orientation and the limited field of view of the sensors.

features corresponding to landmarks, with normals equal to the vector from the center of the object to the feature point was developed. The simulation models two sensor types: a monocular camera and a 3D lidar. Both sensors have a canonical forward orientation but do not generally move in the forward direction. In simulation we first define $s_{\max}$, the range within which the sensor can detect features, and model detection probability as $P(k) = \exp(d(\mathbf{x}_p, \mathbf{x}_{l_k})/(\lambda))$ with $\lambda = 0.7 \cdot s_{\max}$. For the camera, we use a pinhole projection model where object detection candidates are decided by being within the field of view of the camera oriented along the local x -axis. We model the lidar returns along an annulus around the vertically-aligned coordinates of the sensor frame with a field of view angle above and below the xy plane of

the sensor frame.

In the simulations, each object has a uniformly-chosen random number $U \sim [15, 25]$ of features corresponding to landmarks on their surface. Like real landmarks in 3D, corresponding features are not observable from all angles, due to occlusion by the object on which the landmark lies or due to the maximum viewing angle at which a landmark can be detected, distorting their feature-based appearance [3]. This is modeled in simulation by having a maximum angle from the normal of the object at that landmark which is approximated as the vector from the center of the object to the point. If a sensor could detect a landmark that is outside this maximum angle of observation, it will only be detected with probability P_F . The objects stop persisting at a certain time in the simulation, similar to [6], to simulate the movement of a semi-static object or the change in a long term structure.

1000 simulations were run over arbitrary 3D periodic trajectories that pass by the same clique multiple times. An example trajectory and set of cliques are shown in Figure 2. All filters are run over the same trajectories for each run to make it as fair as possible. We denote true positives as TP , False Positives as FP , True Negatives as TN , False Negatives as FN and the total number of possible detection time steps as TR . Our accuracy metric is defined as is traditional as $(TP + TN)/TR$, where a positive is defined as above and negative is defined as below a persistence probability threshold ρ_h to allow a measurement to be incorporated to the factor graph.

B. Results

During an example run, we compare various filters by plotting the posterior probability of a given landmark within a clique. In the SLAM problem, a filter will remove that landmark from the map when the posterior drops below a certain threshold ρ_l , and reject new measurements when the posterior probability drops below ρ_h . As shown in Figure 3, the independent feature filter removes the given landmark

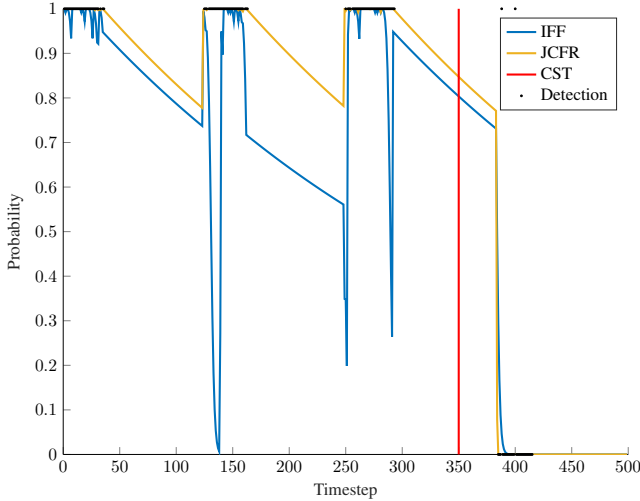


Fig. 3. Comparison of example simulation of the independent feature filter (IFF) [6] against the joint clique filter with range sensor degradation (JCFR), plotting clique survival time (CST) and at least 1 detection (1) or no detections (0) detections from the clique for each expected observation (D). Note that the independent feature filter is far less stable and has many discrepancies with the joint clique filter with range sensor degradation due to the inherent partial observability of landmarks in 3D.

several times, prior to its actual survival time. The joint clique filter with range sensor degradation on the other hand is able to reason over the various missed detections, only removing the landmark from the map after the first observations after the landmarks survival time passes at $t = 350$.

The full results of our simulations can be seen in Table I. The joint clique filter outperforms the others in accuracy, which is quite important as the Precision of all filters is almost always greater than 99%, meaning that nearly all of the error is coming from false negatives for each of the filters. A high false negative count will significantly decrease the number of measurements that can be incorporated into the SLAM algorithm over time, thereby limiting the accuracy of the map and pose estimation, potentially quite significantly. Based on the accuracy of our filters, even in a fully static environment, there should be little to no loss of accuracy while incorporating the ability to detect and remove semi-static and dynamic landmarks from a map.

We also define a metric called the mean estimated survival time over true survival time (MES/S) metric which, for a given clique τ , is defined as

$$\text{MES/S} = \frac{1}{R} \sum_{r=0}^R \frac{\hat{T}_{\tau}^r}{T_{\tau}^r}, \quad (20)$$

where R is the number of runs in a set of simulations, T_{τ}^r is the actual realization of T_{τ} in the r^{th} simulation and $\hat{T}_{\tau}^r$ is the time at which the filter removes the landmark from the map. This metric quantifies the efficiency of the landmark filters in a SLAM system as keeping landmarks in a map for the duration of their persistence in the real world. This is discussed in detail in Section V. For lidar, our results

represent an improvement over both the independent feature filter (IFF) [6] and the joint feature filter (JFF) [11], where the joint clique filter (JCF) and joint clique filter with range sensor degradation (JCFR) achieve a relatively high MES/S with negative suppression, nearing a perfect score of 1.0, and therefore imposing little to no overhead than necessary over the SLAM algorithm to provide feature persistence detection. For cameras, while our method still does well with negative suppression, it is much closer to the independent feature filter in terms of performance under ideal conditions for the IFF. Our algorithm is not particularly designed for camera sensing modalities, in particular, the range-based sensor degradation and negative suppression are particularly well-tailored to the problems that occur in lidar sensing. It is important to note, however, that in all of these sensing modalities the IFF can do arbitrarily badly on both the accuracy and MES/S metric given occlusions which commonly occur in 3D environments of interest. This makes a direct comparison of the methods difficult as the IFF lacks the inherent robustness of a joint filter utilizing all the information of the observed landmarks in the clique. None the less, the JCF and JCFR still both perform competitively against the IFF in the Camera sensing modality in the MES/S metric, and better in every other metric and sensing modality against both the IFF and the JFF.

V. DISCUSSION

When incorporated into a real SLAM pipeline, a negative detection would result in the removal of a landmark from the factor graph. Thus, a filter which underestimates the true survival time extensively of a landmark would cause excessive operations to be performed on the factor graph and measurement information to be lost. For this reason, we introduced MES/S defined in Eq. (20), which accounts for the inefficiency of a filter in prematurely removing landmarks from a graph before it has changed in reality. As the goal of these filters is to keep landmarks in the map for as long as possible without incurring error, this is a useful measure of how efficiently they perform the task they are designed to solve when combined with a high accuracy. Ideally, this metric should be as close to 1 as possible, indicating that landmarks are rarely removed from the map prematurely and then added back in afterwards. The higher the value, the less early removals from the map and thus the smaller penalty for performance and accuracy of the SLAM algorithm. A very low MES/S value would indicate constantly removing landmarks and subsequently all measurement constraints from the factor graph, significantly weakening the estimation.

This is a major weakness of the filters from the previous work [6], [11] we compare to here as they often will produce a false negative very early due to the geometric nature of the features which is not handled in either of these works, pushing this metric lower. To allow these previous filters to perform better on the MFS/S metric, we defined a removal threshold ρ_l which is less than our measurement filter threshold ρ_h as the persistence probability threshold to remove a landmark from the map.

In addition, the tracked landmark in the clique for simulations of the independent feature filter [6] is always observable from the trajectory so that it does not unfairly get suppressed by trajectories that could not detect their corresponding landmark, but can still be undermined by random missed detections. These modifications do give an unfair edge to the previous methods, however, we believe it is better to compare our filter in a more difficult scenario because it makes any positive results stand on their own. In reality, the problem of the observability for the independent feature filter is much more dire when applied to a real SLAM system. At every observable time step, nearly all occluded landmarks in a scene will be quickly removed from the map by the independent feature filter, vastly weakening the strength and reusability of the map for any trajectory except for the one being followed by the current sensor. This severely limits the independent feature filter's practical applicability to 3D SLAM workloads, which will not be able to perform loop closure if a large number of landmarks have been removed from the map upon revisiting a location. Our method addresses this major drawback, having the ability to keep landmarks that cannot be observed in the map while still removing dynamic and semi-static landmarks.

The addition of the sensor degradation range prior from the JCF to the JCFR on the LiDAR appears to be a strict improvement, modeling well the way that LiDAR detects features and observes objects. This does not necessarily seem to be the case in the camera modality though. This is likely because objects can get very close to the camera and then exit the frame, which would denote through the range prior that the filter should nearly always detect the landmarks, but the landmarks would suddenly disappear. This also gets around the negative suppression, explaining why the JCF is dominant with a much smaller margin in this sensing modality, particularly with the JCFR. It is likely, although we do not show, that a similar improvement in camera sensing modalities could be seen by adding a negative suppression based on viewing angle relative to the principle axis of the camera based on Field of View (FOV) to the negative suppression or sensor degradation model to handle the inherent boundary on the edge of the FOV which is similar to the range based boundary seen in lidar which was addressed with the presented negative suppression framework.

Finally, the false negative suppression improves the performance of all filters except for maybe the JFF from [11], which doesn't see much performance improvement from the method. It identifies a major problem of these filters: the fact that all potentially observable landmarks must be updated at once and landmarks are poorly observable at larger distances and angles of observation. False negative suppression provides a simple and satisfying solution to this problem.

VI. CONCLUSIONS

In this paper, we proposed a clique-based change detection model to account for non-static features in feature based

SLAM algorithms. This work expands on previous work developing probabilistic change detection filters, which primarily focused on individual [6] or joint features [11]. While these works provide robust frameworks in 2D environments, they both have limited applicability in 3D where determining individual feature survival times is more difficult due to constant occlusions, less direct sensor models, and reliance on highly robust features. By filtering over cliques we showed our algorithm is more robust to these types of errors achieving both better accuracy and comparable or better MES/S making this a more efficient and robust algorithm. In addition, it is able to account for correlations between features in a simpler manner than [11], by modelling entire cliques jointly.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation National Robotics Initiative 2.0 project #1830686 "Life-Long Learning for Motion Planning by Robots in Human Populated Environments" as well as the Defense Advanced Research Projects Agency Subterranean Challenge "MARBLE."

REFERENCES

- [1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008, ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014.
- [3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.
- [4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 487–495.
- [5] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, Dec. 2001, ISSN: 1042-296X. DOI: 10.1109/70.976019.
- [6] D. M. Rosen, J. Mason, and J. J. Leonard, "Towards lifelong feature-based mapping in semi-static environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1063–1070. DOI: 10.1109/ICRA.2016.7487237.
- [7] A. J. Cooper, "A comparison of data association techniques for Simultaneous Localization and Mapping," Master's thesis, Massachusetts Institute of Technology, United States, 2005.
- [8] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, "Tracking and data association," *The Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 918–919, 1990. DOI: 10.1121/1.398863.
- [9] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. DOI: 10.1177/0278364916679498.
- [10] D. Avitzour, "A maximum likelihood approach to data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 2, pp. 560–566, Apr. 1992, ISSN: 2371-9877. DOI: 10.1109/7.144581.
- [11] F. Nobre, C. Heckman, P. Ozog, R. W. Wolcott, and J. M. Walls, "Online probabilistic change detection in feature-based maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–9. DOI: 10.1109/ICRA.2018.8461111.

- [12] I. Cox and J. J. Leonard, "Modeling a dynamic environment using a Bayesian multiple hypothesis approach," *Artificial Intelligence*, pp. 311–344, 1994.
- [13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfong, Eds. New York, NY: Springer New York, 1990, pp. 167–193, ISBN: 978-1-4613-8997-2. DOI: 10.1007/978-1-4613-8997-2_14.
- [14] M. Montemerlo, "Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association," PhD thesis, Carnegie Mellon University, Pittsburgh, PA, Jul. 2003.
- [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, Jul. 2005, ISSN: 1573-7527. DOI: 10.1007/s10514-005-0606-4.
- [16] D. Weikersdorfer, R. Hoffmann, and J. Conradt, "Simultaneous localization and mapping for event-based vision systems," in *Computer Vision Systems*, M. Chen, B. Leibe, and B. Neumann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 133–142, ISBN: 978-3-642-39402-7.
- [17] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, and S. Thrun, "Learning hierarchical object maps of non-stationary environments with mobile robots," in *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'02, Alberta, Canada: Morgan Kaufmann Publishers Inc., 2002, pp. 10–17, ISBN: 1558608974.
- [18] P. Biber and T. Duckett, "Experimental analysis of sample-based maps for long-term SLAM," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 20–33, 2009. DOI: 10.1177/0278364908096286.
- [19] —, "Dynamic maps for long-term operation of mobile service robots," in *Robotics: Science and Systems*, 2005.
- [20] T. Morris, F. Dayoub, P. Corke, G. Wyeth, and B. Upcroft, "Multiple map hypotheses for planning and navigating in non-stationary environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2765–2770. DOI: 10.1109/ICRA.2014.6907255.
- [21] W. Churchill and P. Newman, "Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 4525–4532. DOI: 10.1109/ICRA.2012.6224596.
- [22] Wei Tan, Haomin Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct. 2013, pp. 209–218. DOI: 10.1109/ISMAR.2013.6671781.
- [23] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, Feb. 2013, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.104.
- [24] Huijing Zhao, Masaki Chiba, Ryosuke Shibasaki, Xiaowei Shao, Jinshi Cui, and Hongbin Zha, "SLAM in a dynamic large outdoor environment using a laser scanner," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1455–1462. DOI: 10.1109/ROBOT.2008.4543407.
- [25] F. Demim, A. Nemra, A. Boucheloukh, K. Louadj, M. Hamerlain, and A. Bazoula, "Robust SVSF-SLAM algorithm for unmanned vehicle in dynamic environment," in *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, Nov. 2018, pp. 1–5. DOI: 10.1109/SIVA.2018.8660984.
- [26] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–9. DOI: 10.1109/ICRA.2018.8460681.
- [27] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.
- [28] F. Dellaert, "Monte Carlo EM for data-association and its applications in computer vision," PhD thesis, Carnegie Mellon University, Pittsburgh, PA, Sep. 2001.