

MIT Open Access Articles

Towards lifelong feature-based mapping in semi-static environments

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Rosen, David M., Julian Mason, and John J. Leonard. "Towards Lifelong Feature-Based Mapping in Semi-Static Environments." IEEE, 2016. 1063–1070.

As Published: <http://dx.doi.org/10.1109/ICRA.2016.7487237>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <http://hdl.handle.net/1721.1/107620>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Towards Lifelong Feature-Based Mapping in Semi-Static Environments

David M. Rosen, Julian Mason, and John J. Leonard

Abstract—The feature-based graphical approach to robotic mapping provides a representationally rich and computationally efficient framework for an autonomous agent to learn a model of its environment. However, this formulation does not naturally support *long-term* autonomy because it lacks a notion of environmental change; in reality, “everything changes and nothing stands still,” and any mapping and localization system that aims to support truly persistent autonomy must be similarly adaptive. To that end, in this paper we propose a novel feature-based model of environmental evolution over time. Our approach is based upon the development of an expressive probabilistic generative *feature persistence model* that describes the survival of abstract semi-static environmental features over time. We show that this model admits a recursive Bayesian estimator, the *persistence filter*, that provides an exact online method for computing, at each moment in time, an explicit Bayesian belief over the persistence of each feature in the environment. By incorporating this feature persistence estimation into current state-of-the-art graphical mapping techniques, we obtain a flexible, computationally efficient, and information-theoretically rigorous framework for *lifelong* environmental modeling in an ever-changing world.

I. INTRODUCTION

The ability to learn a map of an initially unknown environment through exploration, a procedure known as *simultaneous localization and mapping* (SLAM), is a fundamental competency in robotics [1]. Current state-of-the-art SLAM techniques are based upon the graphical formulation of the *full* or *smoothing* problem [2] proposed in the seminal work of Lu and Milios [3]. In this approach, the mapping problem is cast as an instance of maximum-likelihood estimation over a probabilistic graphical model [4] (most commonly a Markov random field [5] or a factor graph [6]) whose latent variables are the sequence of poses in the robot’s trajectory together with the states of any interesting features in the environment.

This formulation of SLAM enjoys several attractive properties. First, it provides an information-theoretically rigorous framework for explicitly modeling and reasoning about uncertainty in the inference problem. Second, because it is abstracted at the level of probability distributions, it provides a convenient modular architecture that enables the straightforward integration of heterogeneous data sources, including both sensor observations *and* prior knowledge (specified in the form of prior probability distributions). This “plug-and-play” framework provides a formal model that greatly simplifies the design and implementation of complex systems while ensuring

D.M. Rosen and J.J. Leonard are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. {dmrosen, jleonard}@mit.edu

J. Mason is with Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043 USA. jmason@google.com

This work was partially supported by the Office of Naval Research under grant N00014-11-1-0688 and by the National Science Foundation under grant IIS-1318392, which we gratefully acknowledge.

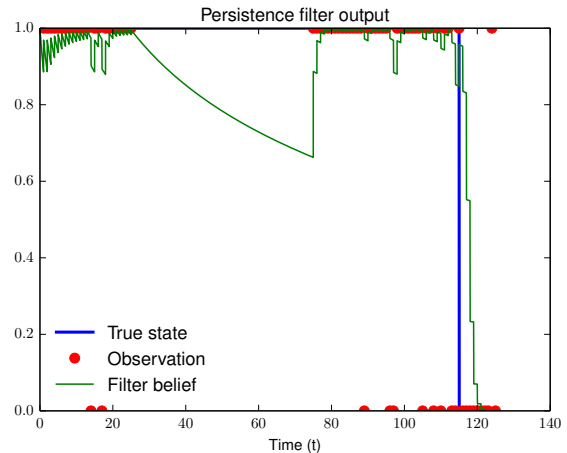


Fig. 1. Recursive Bayesian estimation of feature persistence. This plot shows the (continuous-time) temporal evolution of the persistence filter belief $p(X_t = 1 | \mathcal{Y}_{1:N})$ (the belief over the continued existence of a given environmental feature) given a sequence $\mathcal{Y}_{1:N} \triangleq \{y_{t_i}\}_{i=1}^N$ of intermittent, error-prone outputs from a feature detector with missed detection and false alarm probabilities $P_M = .2$, $P_F = .2$, respectively. Note the decay of the belief in the absence of any signal from the detector (for $t \in (25, 75)$) and the rapid adaptation of the filter belief in response to both newly-available data (at $t = 75$) and change in the hidden state (at $t = 115$).

the correctness of the associated inference procedures. Finally, this formulation admits computationally efficient inference; there exist mature algorithms and software libraries that can solve graphical SLAM problems involving tens of thousands of variables in real time on a single processor [7–10].

However, while graphical SLAM approaches have proven very successful at reconstructing the geometry of environments explored over relatively short timescales, they do not naturally support *long-term* robotic autonomy because their underlying models assume a static world. In reality, “everything changes and nothing stands still,”¹ and any SLAM system that aims to support truly persistent autonomy must be similarly adaptive.

To that end, in this paper we propose a novel feature-based model of environmental change over time. Our approach is grounded in the fields of survival analysis and information theory, and is based upon the development of a probabilistic generative model to describe the temporal persistence of abstract semi-static environmental features. We show that this *feature persistence model* admits a recursive Bayesian estimator, the *persistence filter*, that provides an exact real-time online method for computing, at each moment in time, an explicit Bayesian belief over the persistence of each feature in the environment (Fig. 1). By incorporating this feature persistence estimation into existing graphical SLAM

¹Heraclitus of Ephesus (c. 535 – c. 475 BCE)

techniques, we obtain a flexible, computationally efficient, and information-theoretically sound framework for *lifelong* environmental modeling in an ever-changing world.

II. RELATED WORK

The principal technical challenge that distinguishes semi-static environmental modeling from the well-studied static case is the need for an additional mechanism to *update* an existing map in response to changes in the world. Several novel map representations have been proposed to facilitate this procedure.

In *sample-based* representations, the map consists of a set of *places*, each of which is modeled by a collection of raw sensor measurements; updating a place in response to environmental change is then achieved by simply updating its associated observations. This approach was originally proposed by Biber and Duckett [11, 12], who considered maps comprised of 2D laser scans, and performed updates by replacing a randomly-selected fixed fraction of the scans at each place every time it is revisited. More recently, Churchill and Newman [13] have proposed *plastic maps*, in which a place is modeled as a collection of *experiences* (temporal sequences of sensor observations associated with a small physical area). In contrast to our approach, these methods are intended only to support reliable long-term *localization* in semi-static environments, and do not actually attempt to produce a temporally coherent or geometrically consistent environmental reconstruction.

Konolige and Bowman [14] proposed an extension of Konolige et al.’s *view-based maps* [15] for semi-static environments. In this approach, a place is modeled as a collection of stereocamera frames captured near a single physical location, and the global map is modeled as a graph of places whose edges encode six-degree-of-freedom transforms between camera views. Similarly, Walcott-Bryant et al. [16] extended the pose graph model to support 2D semi-static environment modeling using laser scanners. While these approaches *do* enable temporally and geometrically consistent environmental reconstruction, they are algorithmically tailored to specific classes of sensors and map representations, and lack a unified underlying information-theoretic model for reasoning about environmental change.

As an alternative to sample-based representations, Meyer-Delius et al. [17] and Saarinen et al. [18] independently proposed *dynamic occupancy grids*, which generalize occupancy grid maps [19] to the case of semi-static environments by modeling the occupancy of each cell as a stationary two-state Markov process. Tipaldi et al. [20] subsequently incorporated this model into a generalization of the Rao-Blackwellized particle filtering framework for grid mapping [21], thereby producing a fully Bayesian model-based mapping technique. However, this restriction to the occupancy grid model admits capturing only the volumetric geometry of the environment, which is only one of many potential properties of interest (for example, visual appearance is a highly informative environmental signal for mapping and navigation systems [22]).

Most recently, Krajník et al. [23] have proposed a method for predicting future states of semi-static environments using

Fourier analysis; this prior work is perhaps the most similar to our own, as it combines a feature-abstracted environmental representation with an explicit model of environmental change. However, it is designed to capture only long-term *periodic patterns of activity* in the environment, and appears to lack an information-theoretically grounded approach for addressing sensing errors or quantifying the uncertainty of prediction.

In contrast to this prior work, our formulation provides a unified, feature-abstracted, model-based, information-theoretically grounded framework for temporally coherent and geometrically consistent modeling of generic unstructured semi-static environments.

III. MODELING SEMI-STATIC ENVIRONMENTAL FEATURES

In feature-based mapping, the environment is modeled as a collection of abstract entities called *features*, which encode whatever environmental information is considered relevant to the mapping task (examples of such features include points, lines, planes, objects, and visual interest points). The mapping problem then consists of *identifying* the set of features that are present in the environment and *estimating* their states (e.g. position, orientation, or color), given a collection of noisy observations of the environment.

In static environments, the set of environmental features is fixed for all time by hypothesis, and therefore constructing a map requires only that features be added to it as they are discovered. In contrast, in semi-static environments the feature set itself evolves in time due to environmental change, and therefore mapping requires *both* the ability to add newly-discovered features into the map *and* to remove features that are no longer present.

Ideally, it would suffice to remove a feature from the map if that feature’s location is reobserved and the feature is not detected. In reality, however, a “feature observation” is usually the output of a detector (e.g. an object detector), which may fail to detect features that are present, or generate false detections due to noise in the sensor data. The detector outputs alone are thus insufficient to unambiguously determine whether a feature is still present; the best we can hope for is to estimate a *belief* over feature persistence given this data.

Another complication arises from the passage of time: it seems intuitively clear that our belief about the state of a feature that has not been observed for five *minutes* should differ from our belief about the state of a feature that has not been observed for five *days*.

Given these considerations, it is natural to consider the feature persistence problem in terms of *survival analysis* [24], the branch of statistics that studies the waiting time until some event of interest occurs. In this application, we consider a feature that is first detected at time $t = 0$, and are interested in estimating its *survival time* $T \in [0, \infty)$ (the length of time that it exists in the environment before vanishing), given a sequence of Boolean random variables $\{Y_i\}_{i=1}^N \subseteq \{0, 1\}$ indicating the (possibly erroneous) output of a feature detector sampled at the times $\{t_i\}_{i=1}^N \subset [0, \infty)$. We formalize this scenario using

the following *feature persistence model*:

$$\begin{aligned} T &\sim p_T(\cdot), \\ X_t|T &= \begin{cases} 1, & t \leq T, \\ 0, & t > T, \end{cases} \\ Y_t|X_t &\sim p_{Y_t}(\cdot|X_t); \end{aligned} \quad (1)$$

here $p_T: [0, \infty) \rightarrow [0, \infty)$ is the probability density function for a prior distribution over the survival time T , $X_t \in \{0, 1\}$ is a variable indicating whether the feature is still present at time $t \in [0, \infty)$, and $p_{Y_t}(Y_t|X_t)$ is a conditional distribution modeling the feature detector's measurement process.

We observe that the detector measurement model $p_{Y_t}(\cdot|X_t)$ in (1) is a conditional distribution over two Boolean random variables $X_t, Y_t \in \{0, 1\}$, for which there is a single parametric class with two free parameters: the *probability of missed detection* P_M and the *probability of false alarm* P_F :

$$\begin{aligned} P_M &= p_{Y_t}(Y_t = 0 | X_t = 1; P_M, P_F) \\ P_F &= p_{Y_t}(Y_t = 1 | X_t = 0; P_M, P_F) \end{aligned} \quad (2)$$

with $P_M, P_F \in [0, 1]$.² Since P_M and P_F are innate characteristics of the feature detector, the only freedom in the design of the model (1) is in the selection of the survival time prior $p_T(\cdot)$, which encodes a prior belief about the dynamics of feature persistence in the environment. In Section V we will show how one can design appropriate priors $p_T(\cdot)$ in practice.

IV. ESTIMATING FEATURE PERSISTENCE

In this section we describe methods for performing inference within the feature persistence model (1), assuming access to a sequence of noisy observations³ $\mathcal{Y}_{1:N} \triangleq \{y_{t_i}\}_{i=1}^N$ sampled from the feature detector at times $\{t_i\}_{i=1}^N \subseteq [0, \infty)$ (with $t_i < t_j$ for all $i < j$) and a method for evaluating $p_T(\cdot)$'s cumulative distribution function $F_T(\cdot)$:

$$F_T(t) \triangleq p(T \leq t) = \int_0^t p_T(x) dx. \quad (3)$$

Specifically, we will be interested in computing the posterior probability of the feature's persistence at times in the *present* and *future* (i.e. the posterior probability $p(X_t = 1|\mathcal{Y}_{1:N})$ for times $t \in [t_N, \infty)$), as this is the relevant belief for determining whether to maintain a feature in the map.

A. Computing the posterior persistence probability

We begin by deriving a closed-form solution for the posterior probability $p(X_t = 1|\mathcal{Y}_{1:N})$ for $t \in [t_N, \infty)$.

Observe that the event $X_t = 1$ is equivalent to the event $t \leq T$ in (1). We apply Bayes' Rule to compute the posterior

²We use constant detector error probabilities (P_M, P_F) in equation (2) in order to avoid cluttering the notation with multiple subscripts; however, all of the equations in Section IV continue to hold if each instance of (P_M, P_F) is replaced with the appropriate member of a temporally-varying sequence of error probabilities $\{(P_M, P_F)_{t_i}\}_{i=1}^N$.

³Here we follow the lower-case convention for the y_{t_i} to emphasize that these are sampled *realizations* of the random variables Y_{t_i} .

persistence probability $p(X_t = 1|\mathcal{Y}_{1:N})$ in the equivalent form:

$$p(X_t = 1|\mathcal{Y}_{1:N}) = \frac{p(\mathcal{Y}_{1:N}|T \geq t)p(T \geq t)}{p(\mathcal{Y}_{1:N})}. \quad (4)$$

The prior probability $p(T \geq t)$ follows from (3):

$$p(T \geq t) = 1 - F_T(t). \quad (5)$$

The same equivalence of events also implies that the likelihood function $p(\mathcal{Y}_{1:N}|T)$ has the closed form:

$$p(\mathcal{Y}_{1:N}|T) = \prod_{t_i \leq T} P_M^{1-y_{t_i}} (1-P_M)^{y_{t_i}} \prod_{t_i > T} P_F^{y_{t_i}} (1-P_F)^{1-y_{t_i}}. \quad (6)$$

We observe that $p(\mathcal{Y}_{1:N}|T)$ as defined in (6) is right-continuous and constant on the intervals $[t_i, t_{i+1})$ for all $i = 0, \dots, N$ (where here we take $t_0 \triangleq 0$ and $t_{N+1} \triangleq \infty$) as a function of T . These properties can be exploited to derive a simple closed-form expression for the evidence $p(\mathcal{Y}_{1:N})$:

$$\begin{aligned} p(\mathcal{Y}_{1:N}) &= \int_0^\infty p(\mathcal{Y}_{1:N}|T) \cdot p(T) dT \\ &= \sum_{i=0}^N \int_{t_i}^{t_{i+1}} p(\mathcal{Y}_{1:N}|t_i) \cdot p(T) dT \\ &= \sum_{i=0}^N p(\mathcal{Y}_{1:N}|t_i) [F_T(t_{i+1}) - F_T(t_i)]. \end{aligned} \quad (7)$$

Finally, if $t \in [t_N, \infty)$, then equation (6) shows that:

$$p(\mathcal{Y}_{1:N}|T \geq t) = p(\mathcal{Y}_{1:N}|t_N) = \prod_{i=1}^N P_M^{1-y_{t_i}} (1 - P_M)^{y_{t_i}}. \quad (8)$$

By (4), we may therefore recover the posterior persistence probability $p(X_t = 1|\mathcal{Y}_{1:N})$ as:

$$p(X_t = 1|\mathcal{Y}_{1:N}) = \frac{p(\mathcal{Y}_{1:N}|t_N)}{p(\mathcal{Y}_{1:N})} (1 - F_T(t)), \quad t \in [t_N, \infty) \quad (9)$$

where the likelihood $p(\mathcal{Y}_{1:N}|t_N)$ and the evidence $p(\mathcal{Y}_{1:N})$ are computed using (8) and (6)–(7), respectively.

B. Recursive Bayesian estimation

In this subsection we show how to compute $p(X_t = 1|\mathcal{Y}_{1:N})$ online in *constant time* through recursive Bayesian estimation.

Observe that if we append an additional observation $y_{t_{N+1}}$ to the data $\mathcal{Y}_{1:N}$ (with $t_{N+1} > t_N$), then the likelihood functions $p(\mathcal{Y}_{1:N+1}|T)$ and $p(\mathcal{Y}_{1:N}|T)$ in (6) satisfy:

$$\begin{aligned} p(\mathcal{Y}_{1:N+1}|T) &= \\ &\begin{cases} P_F^{y_{t_{N+1}}} (1 - P_F)^{1-y_{t_{N+1}}} p(\mathcal{Y}_{1:N}|T), & T < t_{N+1}, \\ P_M^{1-y_{t_{N+1}}} (1 - P_M)^{y_{t_{N+1}}} p(\mathcal{Y}_{1:N}|T), & T \geq t_{N+1}. \end{cases} \end{aligned} \quad (10)$$

In particular, (8) and (10) imply that

$$p(\mathcal{Y}_{1:N+1}|t_{N+1}) = P_M^{1-y_{t_{N+1}}} (1 - P_M)^{y_{t_{N+1}}} p(\mathcal{Y}_{1:N}|t_N). \quad (11)$$

Algorithm 1 The Persistence Filter

Input: Feature detector error probabilities (P_M, P_F) , cumulative distribution function $F_T(\cdot)$, feature detector outputs $\{y_{t_i}\}$.

Output: Persistence beliefs $p(X_t = 1 | \mathcal{Y}_{1:N})$ for $t \in [t_N, \infty)$.

1: **Initialization:** Set $t_0 \leftarrow 0$, $N \leftarrow 0$, $p(\mathcal{Y}_{1:0}|t_0) \leftarrow 1$, $L(\mathcal{Y}_{1:0}) \leftarrow 0$, $p(\mathcal{Y}_{1:0}) \leftarrow 1$.

// Loop invariant: the likelihood $p(\mathcal{Y}_{1:N}|t_N)$, partial evidence $L(\mathcal{Y}_{1:N})$, and evidence $p(\mathcal{Y}_{1:N})$ are known at entry.

2: **while** \exists new data $y_{t_{N+1}}$ **do**

Update:

3: Compute the partial evidence $L(\mathcal{Y}_{1:N+1})$ using (14).

4: Compute the likelihood $p(\mathcal{Y}_{1:N+1}|t_{N+1})$ using (11).

5: Compute the evidence $p(\mathcal{Y}_{1:N+1})$ using (13).

6: $N \leftarrow (N + 1)$.

Predict:

7: Compute the posterior persistence probability $p(X_t = 1 | \mathcal{Y}_{1:N})$ for $t \in [t_N, \infty)$ using (9).

8: **end while**

Similarly, consider the following lower partial sum for the evidence $p(\mathcal{Y}_{1:N})$ given in (7):

$$L(\mathcal{Y}_{1:N}) \triangleq \sum_{i=0}^{N-1} p(\mathcal{Y}_{1:N}|t_i) [F_T(t_{i+1}) - F_T(t_i)]. \quad (12)$$

From (7) we have:

$$p(\mathcal{Y}_{1:N}) = L(\mathcal{Y}_{1:N}) + p(\mathcal{Y}_{1:N}|t_N) [1 - F_T(t_N)], \quad (13)$$

and $L(\mathcal{Y}_{1:N+1})$ and $L(\mathcal{Y}_{1:N})$ are related according to:

$$\begin{aligned} L(\mathcal{Y}_{1:N+1}) &= P_F^{y_{t_{N+1}}} (1 - P_F)^{1-y_{t_{N+1}}} \\ &\quad \times (L(\mathcal{Y}_{1:N}) + p(\mathcal{Y}_{1:N}|t_N) [F_T(t_{N+1}) - F_T(t_N)]) \end{aligned} \quad (14)$$

(this can be obtained by splitting the final term from the sum for $L(\mathcal{Y}_{1:N+1})$ in (12) and then using (10) to write the likelihoods $p(\mathcal{Y}_{1:N+1}|t_i)$ in terms of $p(\mathcal{Y}_{1:N}|t_i)$).

Equations (11)–(14) admit the implementation of a *recursive Bayesian estimator* for computing the posterior feature persistence probability $p(X_t = 1 | \mathcal{Y}_{1:N})$ with $t \in [t_N, \infty)$. The complete algorithm, which we refer to as the *persistence filter*, is summarized as Algorithm 1. An example application of this filter is shown in Fig. 1.

V. DESIGNING THE SURVIVAL TIME PRIOR

In this section we describe a practical framework for designing survival time priors $p_T(\cdot)$ in (1) using analytical techniques from survival analysis.

A. Survival analysis

Here we briefly introduce some analytical tools from survival analysis; interested readers may see [24] for more detail.

A distribution $p_T(\cdot)$ over survival time $T \in [0, \infty)$ is often characterized in terms of its *survival function* $S_T(\cdot)$:

$$S_T(t) \triangleq p(T > t) = \int_t^\infty p_T(x) dx = 1 - F_T(t); \quad (15)$$

this function directly reports the probability of survival beyond time t , and is thus a natural object to consider in the context of survival analysis. Another useful characterization of $p_T(\cdot)$ is its *hazard function* or *hazard rate* $\lambda_T(\cdot)$:

$$\lambda_T(t) \triangleq \lim_{\Delta t \rightarrow 0} \frac{p(T < t + \Delta t | T \geq t)}{\Delta t}; \quad (16)$$

this function reports the probability of failure (or “death”) per unit time at time t conditioned upon survival until time t (i.e. the *instantaneous rate of failure* at time t), and so provides an intuitive measure of how “hazardous” time t is. We also define the *cumulative hazard function* $\Lambda_T(\cdot)$:

$$\Lambda_T(t) \triangleq \int_0^t \lambda_T(x) dx. \quad (17)$$

By expanding the conditional probability in (16) and applying definition (15), we can derive a pair of useful expressions for the hazard rate $\lambda_T(\cdot)$ in terms of the survival function $S_T(\cdot)$ and the probability density function $p_T(\cdot)$:

$$\lambda_T(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left[\frac{p(t \leq T < t + \Delta t)}{p(T \geq t)} \right] = \frac{p_T(t)}{S_T(t)} = -\frac{S'_T(t)}{S_T(t)}. \quad (18)$$

We recognize the right-hand side of (18) as $\frac{d}{dt} [-\ln S_T(t)]$; by integrating both sides and enforcing the condition that $S_T(0) = 1$, we find that

$$S_T(t) = e^{-\Lambda_T(t)}. \quad (19)$$

Finally, differentiating both sides of (19) shows that

$$p_T(t) = \lambda_T(t) e^{-\Lambda_T(t)}, \quad (20)$$

which expresses the original probability density function $p_T(\cdot)$ in terms of its hazard function $\lambda_T(\cdot)$.

Equations (16) and (19) show that every hazard function satisfies $\lambda_T(t) \geq 0$ and $\lim_{t \rightarrow \infty} \int_0^t \lambda_T(x) dx = \infty$; conversely, it is straightforward to verify that any function $\lambda_T: [0, \infty) \rightarrow \mathbb{R}$ satisfying these two properties defines a valid probability density function $p_T(\cdot)$ on $[0, \infty)$ (by means of (20)) for which it is the hazard function. In the next subsection, we will use this result to design survival time priors $p_T(\cdot)$ for (1) by specifying their hazard functions $\lambda_T(\cdot)$.

B. Survival time belief shaping using the hazard function

While the probability density, survival, hazard, and cumulative hazard functions are all equivalent representations of a probability distribution over $[0, \infty)$, the hazard function provides a *dynamical* description of the survival process (by specifying the *instantaneous event rate*), and is thus a natural domain for designing priors $p_T(\cdot)$ that encode information about how the feature disappearance rate varies over time. This enables the construction of survival time priors that, for example, encode knowledge of patterns of activity in the environment, or that induce posterior beliefs about feature persistence that evolve in time in a desired way. By analogy with *loop shaping* (in which a control system is designed implicitly by specifying its transfer function), we can think

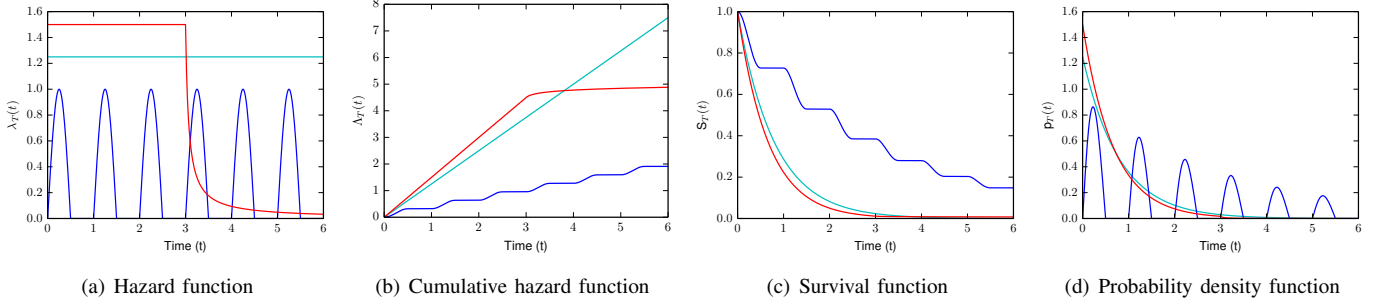


Fig. 2. Survival time belief shaping using the hazard function. (a)–(d) show the corresponding hazard, cumulative hazard, survival, and probability density functions, respectively, for three survival time distributions (colored red, blue, and cyan). Note that even visually quite similar probability density functions (compare the red and cyan functions in (a) and (d)), thus illustrating the utility of analyzing and constructing survival time priors using the hazard function.

of this process of designing the prior $p_T(\cdot)$ by specifying its hazard function $\lambda_T(\cdot)$ as survival time *belief shaping*.

As an illustrative concrete example, Fig. 2 shows several hazard functions and their corresponding cumulative hazard, survival, and probability density functions. Consider the red hazard function in Fig. 2(a), which has a high constant failure rate for $t \in [0, 3]$ and falls off rapidly for $t > 3$. Interpreting this in terms of feature persistence, this function encodes the idea that *any feature that persists until time $t = 3$ is likely static* (e.g. part of a wall). Similarly, the blue hazard function in Fig. 2(a) shows a periodic alternation between high failure rates over the intervals $(k, k + \frac{1}{2})$, $k = 0, 1, \dots$ and *zero risk* over the intervals $[k + \frac{1}{2}, k + 1]$ (i.e. any feature that survives until $k + \frac{1}{2}$ is guaranteed to survive until $k + 1$). This provides a plausible model of areas which are known to be static at certain times, e.g. an office environment that closes overnight.

This example also serves to illustrate the utility of the belief-shaping approach. It may not always be clear how to directly construct a survival-time prior that encodes a given set of environmental dynamics; for example, comparing the red and cyan curves in Figs. 2(a) and 2(d) reveals that even visually-similar probability density functions may encode radically different environmental properties. In contrast, the dynamical description provided by the hazard function is easily interpreted, and therefore easily designed.

C. General-purpose survival time priors

Survival time belief shaping provides an expressive modeling framework for constructing priors $p_T(\cdot)$ that incorporate knowledge of time-varying rates of feature disappearance; however, this approach is only useful when such information is *available*. In this subsection, we describe methods for selecting non-informative priors $p_T(\cdot)$ that are safe to use when little is known in advance about the environmental dynamics.

Our development is motivated by the *principle of maximum entropy* [25], which asserts that the epistemologically correct choice of prior $p_\theta(\cdot)$ for a Bayesian inference task over a latent state θ is the one that has *maximum entropy* (and is therefore the *maximally uncertain* or *minimally informative* prior), subject to the enforcement of any conditions on θ that are *actually known* to be true. The idea behind this principle

is to avoid the selection of a prior that “accidentally” encodes additional information about θ beyond what is *actually known*, and may therefore improperly bias the inference.

From the perspective of belief shaping using the hazard function, this principle suggests the use of a *constant* hazard rate (since the use of any non-constant function would encode the belief that there are certain distinguished times at which features are more likely to vanish). And indeed, setting $\lambda_T(t) \equiv \lambda \in (0, \infty)$ corresponds (by means of (20)) to selecting the exponential distribution:

$$p_T(t; \lambda) = \lambda e^{-\lambda t}, \quad t \in [0, \infty) \quad (21)$$

which is the maximum entropy distribution amongst all distributions on $[0, \infty)$ with a specified expectation $E[T]$. In the case of the exponential distribution (21), this expectation is:

$$E[T] = \lambda^{-1} \quad (22)$$

so we see that the selection of the (hazard) rate parameter λ in (21) encodes the expected feature persistence time.

We propose several potential methods for selecting this parameter in practice:

1) *Engineering the persistence filter belief*: As can be seen from (9), the survival function $S_T(t) = 1 - F_T(t)$ controls the rate of decay of the persistence probability $p(X_t = 1 | \mathcal{Y}_{1:N})$ in the absence of new observations (cf. Fig. 1). The survival function corresponding to the exponential distribution (21) is

$$S_T(t; \lambda) = e^{-\lambda t}, \quad (23)$$

so that λ has an additional interpretation as a parameter that controls the half-life $\tau_{1/2}$ of the decay process (23):

$$\tau_{1/2} = \ln(2)/\lambda. \quad (24)$$

Thus, one way of choosing λ is to view it as a design parameter that enables engineering the evolution of the persistence filter belief $p(X_t = 1 | \mathcal{Y}_{1:N})$ so as to enforce a desired decay or “forgetting” rate in the absence of new data.

2) *Class-conditional learning*: A second approach follows from the use of a feature detector to generate the observations in the feature persistence model. For cases in which features are objects (e.g. chairs or coffee cups) and their detectors are

trained using machine learning techniques, one could use the training corpus to learn class-conditional hazard rates λ that characterize the empirical “volatility” of a given class.

3) *Bayesian representation of parameter uncertainty*: Finally, in some cases it may not be possible to determine the “correct” value of the rate parameter λ , or it may be possible to do so only imprecisely. In these instances, the correct Bayesian procedure is to introduce an additional prior $p_\lambda(\cdot)$ to model the uncertainty in λ explicitly; one can then recover the marginal distribution $p_T(\cdot)$ for the survival time (accounting for the uncertainty in λ) by marginalizing over λ according to:

$$p_T(t) = \int_0^\infty p_T(t; \lambda) \cdot p_\lambda(\lambda) d\lambda. \quad (25)$$

Once again we are faced with the problem of selecting a prior, this time over λ . However, in this instance we know that λ parameterizes the sampling distribution (21) as a *rate parameter*, i.e. we have $p_T(t; \lambda) = \lambda f(\lambda t)$ for some fixed normalized density $f(\cdot)$ (in this case, $f(x) = e^{-x}$). Since t enters the distribution $p_T(t; \lambda)$ in (21) only through the product λt , we might suspect that $p_T(t; \lambda)$ possesses a symmetry under the multiplicative group $\mathbb{R}_+ \triangleq (0, \infty)$ with action $(\lambda, t) \mapsto (q\lambda, t/q)$ for $q \in \mathbb{R}_+$. Letting $\lambda' \triangleq q\lambda$ and $t' \triangleq t/q$, we find that $p_T(t'; \lambda') dt' = p_T(t; \lambda) dt$, which proves that the probability measure encoded by the density $p_T(t; \lambda)$ is indeed invariant under this action.

The *principle of transformation groups* [25] asserts that any prior $p_\theta(\cdot)$ expressing ignorance of the parameter of a sampling distribution $p_y(y|\theta)$ that is invariant under the action of a group G should likewise be invariant under the same action. (This principle is closely related to the *principle of indifference*, and can be thought of as a generalization of this principle to the case of priors over continuous parameter spaces that are equipped with a specified symmetry group.)

Enforcing this requirement in the case of our rate parameter λ , it can be shown (cf. [25, Sec. 7]) that the prior we seek satisfies $p_\lambda(\lambda) \propto \lambda^{-1}$. While this is formally an improper prior, if we incorporate the additional (weak) prior information that the feature survival processes we wish to model evolve over timescales (24) on the order of, say, seconds to decades, we may restrict $\lambda \in [\lambda_l, \lambda_u]$ using (conservative) lower and upper bounds λ_l, λ_u , and thereby recover:

$$p_\lambda(\lambda; \lambda_l, \lambda_u) = \frac{1}{\lambda \ln(\lambda_u/\lambda_l)}, \quad \lambda \in [\lambda_l, \lambda_u]. \quad (26)$$

The marginal density $p_T(\cdot)$ obtained from (25) using (21) and (26) is then:

$$p_T(t; \lambda_l, \lambda_u) = \frac{e^{-\lambda_l t} - e^{-\lambda_u t}}{t \ln(\lambda_u/\lambda_l)} \quad (27)$$

with corresponding survival function:

$$S_T(t; \lambda_l, \lambda_u) = \frac{E_1(\lambda_l t) - E_1(\lambda_u t)}{\ln(\lambda_u/\lambda_l)}, \quad (28)$$

where $E_1(x)$ is the *first generalized exponential integral*:

$$E_1(x) \triangleq \int_1^\infty t^{-1} e^{-xt} dt. \quad (29)$$

Equations (27)–(29) define a class of minimally-informative “general-purpose” survival time priors that can be safely applied when little information is available in advance about rates of feature disappearance in a given environment.

VI. GRAPHICAL SLAM IN SEMI-STATIC ENVIRONMENTS

Finally, we describe how to incorporate the feature persistence model (1) into the probabilistic graphical model formulation of SLAM to produce a framework for feature-based mapping in semi-static environments.

The approach is straightforward. Our operationalization of features as having a fixed geometric configuration but finite persistence time enforces a clean separation between the estimation of environmental geometry and feature persistence. Existing graphical SLAM techniques provide a mature solution for geometric estimation, and the persistence filter provides a solution to persistence estimation. Thus, to extend the graphical SLAM framework to semi-static environments, we simply associate a persistence filter with each feature. In practice, newly-discovered environmental features are initialized and added to the graph in the usual way, while features that have “vanished” (those for which $p(X_t = 1 | \mathcal{Y}_{1:N}) < P_V$ for some threshold probability P_V) can be removed from the graph by marginalization [26, 27].

VII. EXPERIMENTS

One of the principal motivations for developing the persistence filter is to support long-term autonomy by enabling a robot to update its map in response to environmental change. To that end, consider a robot operating in some environment for an extended period of time. As it runs, it will from time to time revisit each location within that environment (although not necessarily at regular intervals or equally often), and during each revisitation, it will collect some number of observations of the features that are visible from that location. We are interested in characterizing the persistence filter’s estimation performance under these operating conditions.⁴

A. Experimental setup

We simulate a sequence of observations of a single environmental feature using the following generative procedure:

- 1) For a simulation of length L , begin by sampling a feature survival time T uniformly randomly: $T \sim \mathcal{U}([0, L])$.
- 2) Sample the time R until revisitation from an exponential distribution with rate parameter λ_R : $R \sim \text{Exp}(\lambda_R)$.
- 3) Sample the number $N \in \{1, 2, \dots\}$ of observations collected during this revisitation from a geometric distribution with success probability p_N : $N \sim \text{Geo}(p_N)$.
- 4) For each $i \in \{1, \dots, N\}$, sample an observation y_i according to the detector model (2), and a waiting time O_i until the next observation y_{i+1} from an exponential distribution with rate parameter λ_O : $O_i \sim \text{Exp}(\lambda_O)$.

⁴A C++ implementation (with Python bindings) of the persistence filter, together with the code used to perform the following experiments, is available through http://groups.csail.mit.edu/marine/persistence_filter/.

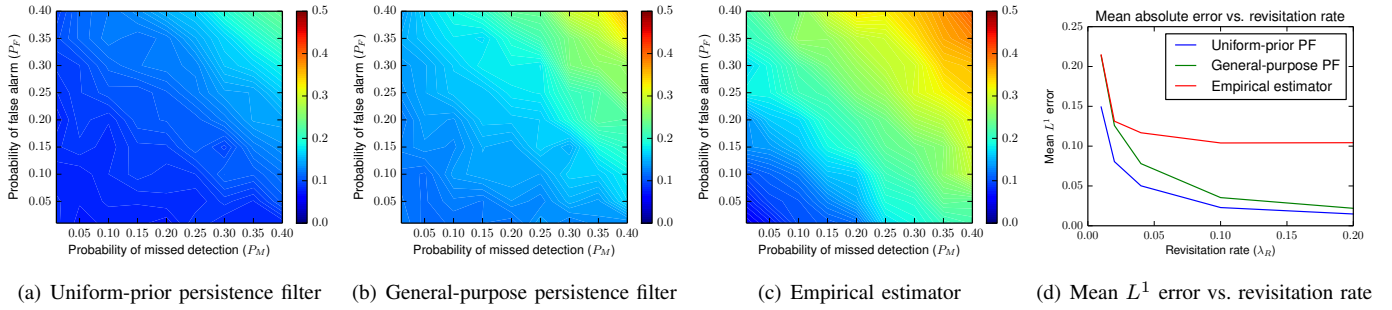


Fig. 3. Feature persistence estimation performance under the average L^1 error metric. (a)–(c) plot the mean value of $E(X_t, p(X_t = 1 | \mathcal{Y}_{1:N}))$ as defined in (30) over 100 observation sequences sampled from the generative model of Section VII-A as a function of the detector error rates $(P_M, P_F) \in \{.01, .05, .1, .15, .20, .25, .30, .35, .40\}^2$ (holding $L = 1000$, $\lambda_R = 1/50$, $\lambda_O = 1$, and $P_N = .2$ constant) for the uniform-prior persistence filter, general-purpose prior persistence filter, and empirical estimator, respectively. (d) plots the the mean value of the error $E(X_t, p(X_t = 1 | \mathcal{Y}_{1:N}))$ over 100 observation sequences sampled from the generative model of Section VII-A as a function of the revisitation rate $\lambda_R \in \{1/100, 1/50, 1/25, 1/10, 1/5\}$ (holding $L = 1000$, $\lambda_O = 1$, $P_N = .2$, and $(P_M, P_F) = (.1, .1)$ constant).

5) Repeat steps 2) through 4) to generate a sequence of observations $\{y_{t_i}\}$ of temporal duration at least L .

We consider the performance of three feature persistence estimators on sequences drawn from the above model:

- The persistence filter constructed using the true (uniform) survival time prior $\mathcal{U}([0, L])$;
- The persistence filter constructed using the general-purpose survival time prior defined in (27)–(29) with $(\lambda_l, \lambda_u) = (.001, 1)$; and
- The *empirical estimator*, which reports the most recent observation y_{t_N} as the true state X_t with certainty.

B. Evaluation

We consider two metrics for evaluating the performance of our estimators: average L^1 error and classification accuracy.

1) *Average L^1 error*: In our first evaluation, we consider both the ground-truth state indicator variable X_t and the posterior persistence belief $p(X_t = 1 | \mathcal{Y}_{1:N})$ as functions of time $t \in [0, L]$ taking values in $[0, 1]$, and adopt as our error metric $E(\cdot, \cdot)$ the mean L^1 distance between them:

$$E(f, g) \triangleq \frac{1}{L} \int_0^L |f(t) - g(t)| dt, \quad f, g \in L^1([0, L]). \quad (30)$$

Intuitively, the error $E(X_t, p(X_t = 1 | \mathcal{Y}_{1:N}))$ measures how closely an estimator’s persistence belief $p(X_t = 1 | \mathcal{Y}_{1:N})$ tracks the ground truth state X_t over time (graphically, this corresponds to the average vertical distance between the green and blue curves in Fig. 1 over the length of the simulation.)

Fig. 3 plots the mean of the error $E(X_t, p(X_t = 1 | \mathcal{Y}_{1:N}))$ over 100 observation sequences sampled from the generative model of Section VII-A as a function of both the feature detector error rates (P_M, P_F) and the revisitation rate λ_R for each of the three persistence estimators. We can see that the uniform-prior persistence filter strictly dominates the other two estimators. The general-purpose persistence filter also outperforms the empirical estimator across the majority of the test scenarios; the exceptions are in the low-detector-error-rate and infrequent-revisitation regimes, in which the L^1 error metric penalizes the decay in the persistence filter’s belief over time (cf. Fig. 1), while the empirical estimator always reports

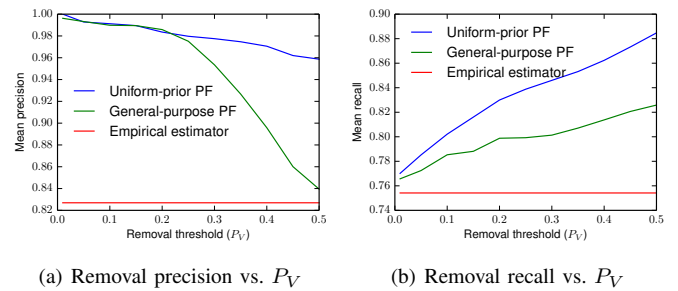


Fig. 4. Feature removal classification performance. (a) and (b) show the mean removal precision and recall over 100 observation sequences sampled from the generative model of Section VII-A (with $L = 1000$, $(P_M, P_F) = (.1, .1)$, $\lambda_R = 1/50$ and $\lambda_O = 1$) as a function of the removal threshold $P_V \in \{.01, .05, .1, .15, .2, .25, .3, .35, .4, .45, .5\}$. For each observation sequence, the removal precision and recall are computed by querying the persistence belief $p(X_t = 1 | \mathcal{Y}_{1:N})$ at $.1$ -time intervals, thresholding the belief at P_V to obtain a $\{0, 1\}$ -valued prediction, and computing the precision and recall of the sequence of “removal” decisions (0-valued predictions).

its predictions with complete certainty. We will see in the next evaluation that the empirical estimator’s gross overconfidence in its beliefs actually leads to much poorer performance on feature removal decisions versus the persistence filter.

2) *Classification accuracy*: In practice, we would use the persistence belief to decide when a feature should be removed from a map. While the belief is in $[0, 1]$, the decision to remove a feature is binary, which means that we must select a belief threshold P_V for feature removal (see Section VI). The threshold P_V thus parameterizes a family of binary classifiers for feature removal, and we would like to characterize the performance of this family as a function of P_V .

Fig. 4 plots the mean precision and recall of these feature removal decisions over 100 observation sequences sampled from the generative model of Section VII-A as a function of P_V . We can see that both persistence filter implementations significantly outperform the empirical estimator in precision and recall for feature removal decisions. Furthermore, the loss in performance introduced by using the general-purpose prior in place of the true (uniform) prior is relatively modest, validating our earlier development in Section V-C3.

VIII. CONCLUSION

In this paper we proposed the *feature persistence model*, a novel feature-based model of environmental temporal evolution, and derived its associated inference algorithm, the *persistence filter*. Our formulation improves upon prior work on modeling semi-static environments by providing a unified, feature-abstracted, information-theoretically rigorous model of environmental change that can be used in conjunction with any feature-based environmental representation. Our approach also provides a remarkably rich modeling framework versus prior techniques, as it supports the use of *any* valid survival time prior distribution (including non-Markovian priors), while still admitting exact, constant-time online inference.

REFERENCES

- [1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2008.
- [2] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.
- [3] F. Lu and E. Milius. Globally consistent range scan alignment for environmental mapping. *Autonomous Robots*, 4:333–349, April 1997.
- [4] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA, 2009.
- [5] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *Intl. J. of Robotics Research*, 25(5–6):403–429, 2006.
- [6] F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, September 2012.
- [7] G. Grisetti, C. Stachniss, and W. Burgard. Nonlinear constraint network optimization for efficient map learning. *IEEE Trans. on Intelligent Transportation Systems*, 10(3):428–439, September 2009.
- [8] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3607–3613, May 2011.
- [9] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31(2):216–235, February 2012.
- [10] D.M. Rosen, M. Kaess, and J.J. Leonard. RISE: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Trans. Robotics*, 30(5): 1091–1108, October 2014.
- [11] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, June 2005.
- [12] P. Biber and T. Duckett. Experimental analysis of sample-based maps for long-term SLAM. *Intl. J. of Robotics Research*, 28(1):20–33, January 2009.
- [13] W. Churchill and P. Newman. Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4525–4532, St. Paul, MN, May 2012.
- [14] K. Konolige and J. Bowman. Towards lifelong visual maps. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1156–1163.
- [15] K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *Intl. J. of Robotics Research*, 29(8):941–957, July 2010.
- [16] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J.J. Leonard. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1871–1878, Vilamoura, Algarve, Portugal, October 2012.
- [17] D. Meyer-Delius, M. Beinhofer, and W. Burgard. Occupancy grid models for robot mapping in changing environments. In *Proc. AAAI Conf. on Artificial Intelligence*, pages 2024–2030, Toronto, Canada, July 2012.
- [18] J. Saarinen, H. Andreasson, and A.J. Lilienthal. Independent Markov chain occupancy grid maps for representation of dynamic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3489–3495, Vilamoura, Algarve, Portugal, October 2012.
- [19] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 116–121, St. Louis, MO, USA, March 1985.
- [20] G.D. Tipaldi, D. Meyer-Delius, and W. Burgard. Lifelong localization in changing environments. *Intl. J. of Robotics Research*, 32(14):1662–1678, December 2013.
- [21] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. Robotics*, 23(1):34–46, February 2007.
- [22] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *Intl. J. of Robotics Research*, 30(9):1100–1123, August 2011.
- [23] T. Krajník, J.P. Fentanes, G. Cielniak, C. Dondrup, and T. Duckett. Spectral analysis for long-term robotic mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3706–3711, May 2014.
- [24] J.G. Ibrahim, M.-H. Chen, and D. Sinha. *Bayesian Survival Analysis*. Springer Science+Business Media, 2001.
- [25] E.T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227–241, September 1968.
- [26] G. Huang, M. Kaess, and J.J. Leonard. Consistent sparsification for graph optimization. In *European Conf. on Mobile Robotics (ECMR)*, pages 150–157, Barcelona, Spain, September 2013.
- [27] M. Mazuran, G.D. Tipaldi, L. Spinello, and W. Burgard. Nonlinear graph sparsification for SLAM. In *Robotics: Science and Systems (RSS)*, Berkeley, CA, USA, 2014.