Robotics and Autonomous Systems 87 (2017) 162–176

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Episodic non-Markov localization

Joydeep Biswas *,[1], Manuela M. Veloso

Computer Science Department, Carnegie Mellon University, PA, USA

HIGHLIGHTS

- Reasoning about observations arising from permanent, temporary, or moving objects for mobile robot localization in changing environments.
- A new graphical representation that augments the Markov localization DBN to represent the presence of, and correlations between observations of unmapped objects.
- Derivation of the belief for Episodic non-Markov Localization (EnML).
- Analysis of its computational complexity.
- Experimental results showing the benefits of EnML.

ARTICLE INFO

Article history:
Received 20 November 2015
Accepted 2 September 2016
Available online 29 September 2016

Keywords:
Localization
Long-term autonomy
Mapping

ABSTRACT

Markov localization and its variants are widely used for mobile robot localization. These methods assume Markov independence of observations, implying that the observations can be entirely explained by a map. However, in real human environments, robots frequently make unexpected observations due to unmapped static objects like chairs and tables, and dynamic objects like humans. We therefore introduce Episodic non-Markov Localization (EnML), which reasons about the world as consisting of three classes of objects: long-term features corresponding to permanent mapped objects, short-term features corresponding to unmapped static objects, and dynamic features corresponding to unmapped moving objects. Long-term features are represented by a static map, while short-term features are detected and tracked in real-time. To reason about unexpected observations and their correlations across poses, we augment the Dynamic Bayesian Network for Markov localization to include varying edges and nodes, resulting in a novel Varying Graphical Network representation. The maximum likelihood estimate of the belief is incrementally computed by non-linear functional optimization. By detecting timesteps along the robot's trajectory where unmapped observations prior to such time steps are unrelated to those afterwards, EnML limits the history of observations and pose estimates to "episodes" over which the belief is computed. We demonstrate EnML using different types of sensors including laser rangefinders and depth cameras, and over multiple datasets, comparing it with alternative approaches. We further include results of a team of indoor autonomous service mobile robots traversing hundreds of kilometers using EnML.

1. Introduction

Human environments have elements that remain unchanged over time, for example, the architectural features of buildings. However, such environments also include many movable objects like tables and chairs, and moving objects like humans. Robots deployed in human environments therefore often encounter observations of movable and moving objects at locations that change over time. Markov localization and its many variants, widely used for mobile robot localization, are thus ill-suited for robots deployed over extended periods of time, since they assume that all observations of a robot can be explained by a static map. In recognition of the limitations of Markov Localization with a static map in changing human environments, two categories of alternate solutions have been proposed: (1) to re-map the environment when changes are detected, or (2) to model and learn the dynamics of the changes in the environment. Environments that change infrequently, and remain static between such reconfigurations, are suited to the approach of re-mapping. In environments where the

* Corresponding author.
E-mail addresses: joydeepb@cs.cmu.edu (J. Biswas), veloso@cs.cmu.edu (M.M. Veloso).
[1] Present address: College of Information and Computer Sciences, University of Massachusetts, Amherst, USA.

http://dx.doi.org/10.1016/j.robot.2016.09.005
0921-8890/

changes occur continually, and where robots may be deployed frequently to every location, it may also be possible to learn the dynamics of the changes in the environment.

However, in areas that are too large to observe repeatedly and frequently, or where there is no discernible pattern to the changes, it may be infeasible or intractable to either build updated maps, or to model the changes in the environment. We therefore propose a novel localization algorithm for a third alternative, where the robot reasons about the *nature* of its observations: whether they correspond to permanent objects which we term "Long-Term Features" (LTFs), movable objects termed "Short-Term Features" (STFs), or moving objects, termed "Dynamic Features" (DFs). We call this "Episodic Non-Markov Localization" (EnML) because it relaxes the central assumption of Markov localization that observations are independent given the map, and because it reasons about episodes of observations that relate the corresponding robot poses. EnML reasons about non-mapped objects in real time, at the time of deployments, without relying on up-to-date static maps, or dynamic maps.

We introduce a new graphical model, the Varying Graphical Network, to keep track of correlations between observations arising from observations of unmapped objects while simultaneously tracking correlations to a permanent map. EnML maintains a belief of the history of pose estimates of the robot over "episodes" of observations of unmapped objects. For every time-step, it classifies observations into LTFs, STFs, or DFs. Observations made from LTFs are matched to a permanent map, while correlations between observations from STFs at different time steps are computed. The belief is framed in terms of a cost function over odometry observations, observations of LTFs related to the static map, and correlations between observations of STFs from different time steps. The maximum likelihood estimate of the belief is incrementally computed over successive steps by functional non-linear least-squares optimization over the cost function. The cost function representation of the belief is distinct from other belief representations such as Particle Filters [1], Extended Kalman Filters (EKFs) and Unscented Kalman Filters (UKFs) [2], and Partially Observable Markov Decision Processes (POMDPs) [3]. As we shall show, the cost function representation allows EnML to tractably compute the maximum likelihood estimate of the belief even over long episodes, which would not have been possible using particle filters, EKFs, UKFs, or POMDPs.

We present experimental results using multiple datasets to quantitatively compare the accuracy of EnML to alternative approaches. EnML is agnostic to the specific type of sensor used — we present results over datasets using long-range and short-range laser rangefinders, as well as inexpensive depth cameras like the Microsoft Kinect sensor. Furthermore, we present results from extensive deployments of a set of autonomous service mobile robots, the CoBots [4], comparing the robustness of EnML to Corrective Gradient Refinement [5], a variant of Markov Localization.

2. Background and related work

Suppose a robot makes observations $S = \{s_1, \ldots, s_n\}$ of the environment, and accumulates robot odometry $U = \{u_1, \ldots, u_n\}$ over n time-steps $t_1 \ldots t_n$. At each time-step t_i, observation s_i is made, and odometry u_i is recorded, corresponding to the relative motion of the robot from timestep t_{i-1} to t_i. Given a prior map of the environment M, the problem of mobile robot localization is then stated as estimating the probability distribution over the robot pose at the latest time step x_n, or the "belief" *Bel*,

$$Bel(x_n) = P(x_n|s_1, \ldots, s_n, u_1, \ldots, u_n, x_0, \ldots, x_{n-1}, M). \quad (1)$$

This equation is general to all localization algorithms, assuming only that there exists a prior map of the environment M.

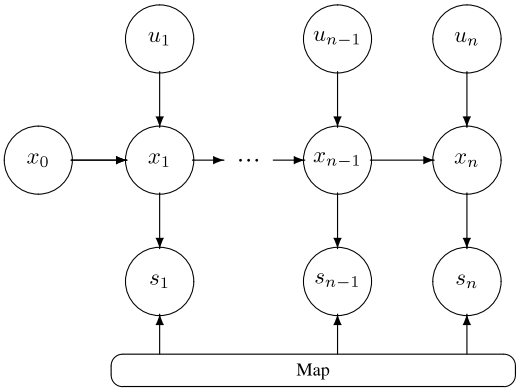

Fig. 1. Dynamic Bayesian network for Markov localization.

Markov Localization [6] makes two independence assumptions in order to simplify the computation of the belief:

1. **Markov independence of odometry**: Given the estimate of the robot pose x_{n-1} and the latest odometry u_n, the estimate of x_n is independent of past pose estimates $x_0, \ldots, x_{n-2}$ and odometry $u_1, \ldots, u_{n-1}$, so that

$$P(x_n|s_1, \ldots, s_{n-1}, u_1, \ldots, u_n, x_0, \ldots, x_{n-1}, M) = P(x_n|x_{n-1}, u_n). \quad (2)$$

2. **Markov independence of observations**: Given the estimate of robot pose x_n, observation s_n is independent of past pose estimates $x_0, \ldots, x_{n-1}$, observations $s_1, \ldots, s_{n-1}$ and odometry $u_1, \ldots, u_n$, so that

$$P(s_n|s_1, \ldots, s_{n-1}, u_1, \ldots, u_n, x_0, \ldots, x_n, M) = P(s_n|x_n, M). \quad (3)$$

Using the Markov assumptions and applying Bayes rule, the recursive update of the belief simplifies to (see [7] for a complete derivation),

$$Bel(x_n) \propto P(s_n|x_n, M) \int P(x_n|x_{n-1}, u_n) Bel(x_{n-1}) dx_{n-1}. \quad (4)$$

Fig. 1 shows the dynamic Bayesian network (DBN) for Markov localization. Based on this DBN, for the "prediction" step before s_n becomes available, the Markov blanket [8] of x_n consists of u_n, which is observed, and x_{n-1}, which is estimated by the belief $Bel(x_{n-1})$. For the "update" step, the Markov blanket of s_n consists of x_n, which is estimated by the predict step, u_n, which is observed, x_{n-1}, which is estimated by the belief $Bel(x_{n-1})$), and the map M, which is known. Thus the belief update for $Bel(x_n)$ does not require storing the history of observations and states prior to t_{n-1}.

2.1. Markov localization and variants

By varying the belief representation, the map representation, how the update steps are computed, and what sensors are used, numerous variants of Markov localization have been proposed. The belief could be represented by Position Probability Grids [6], Partially Observable Markov Decision Processes (POMDPs) [3], Normal distributions [9], or discrete samples called "particles" in Monte Carlo Localization [1].

Monte Carlo Localization (MCL) [1] approximates the belief distribution as a set of discrete samples, and has been shown to be effective at localizing a robot in a number of environments with few or no changes over time. However, in varying environments,

there are often observations that do not match the static map. Such observations are ignored by MCL, thus increasing the uncertainty of the belief. Several extensions to MCL have therefore been proposed to model the uncertainty of the belief by varying the distribution, and number of samples.

KLD-Sampling [10] adapts the number of samples based on the Kullback–Leibler Distance between the belief and the posterior, thus reducing the computational requirements when the belief accurately models the posterior, and increasing the number of particles when the posterior distribution is likely to spread out. Sensor Resetting Localization (SRL) [11] takes into account mismatch between observations and the belief by drawing samples directly from the observation model to incorporate into the belief. Corrective Gradient Refinement (CGR) [5] uses the analytically computed state space derivatives of location samples to refine the proposal distribution, thus reducing the number of required particles by sampling more densely along directions of high uncertainty.

Extensions to Markov Localization such as KLD-Sampling, SRL, and CGR allow a robot to model the large uncertainties in localization arising from observations that do not match the map in environments with frequent changes. However, their localization accuracy is still poor, due to insufficient corrections from the observation model resulting from the mismatch between observations and the map.

2.2. Simultaneous localization and mapping

So far we have assumed that a map M of the environment exists, and is used for robot localization. This map could be built by collecting sensory data and batch processing it offline, or (as is more common in recent work) online on the robot while simultaneously tracking the location of the robot on the map. "Simultaneous Localization And Mapping" (SLAM) is the problem of building a map M while concurrently maintaining a belief of the robot's trajectory $x_0, \ldots, x_n$.

The DBN for SLAM with a static map is similar in structure to that of Markov localization. The difference is that in Markov localization, the map M is known and only the latest pose estimate x_n is updated at every step, but in SLAM the map M is unknown a priori, and is updated at every step, along with the full history of pose estimates $x_0, \ldots, x_n$. Despite the updates to the history of pose estimates, the model is still Markovian: it is assumed that were x_{n-1} and u_n accurately known, x_n would be independent of all other prior states and observations, and that were x_n and M accurately known, observation s_n would be independent of all other prior states and observations. Bailey and Durant [12,13] review some of the popular approaches to SLAM. Pose graph SLAM [14] frames the SLAM problem as a graph optimization problem where robot poses are represented as nodes on a graph, and edges joining the nodes indicate relations between them based on odometry or observation matching. "Factor Graphs" [15] offer an alternative representation of the correlations between the pose history and the map in SLAM, replacing measurements by "factor nodes". Some approaches split up the SLAM problem into algorithms for building a pose graph (e.g., by scan matching [16,17]) and algorithms for efficiently updating and optimizing the pose graph by iterative non-linear optimization [18].

2.3. Varying environments

In reality, neither the odometry nor the observations of a robot deployed in a real environment are truly Markovian. In a human environment, as shown in Fig. 2, observations made by the robot include parts that do indeed correspond to the map, but also include moving objects (e.g., humans and other robots) and movable objects (e.g., tables, chairs, recycling bins) that change locations on a daily basis. Since Markov localization assumes that observations correspond only to features from a static map, unexpected observations of static objects will cause two potential problems:

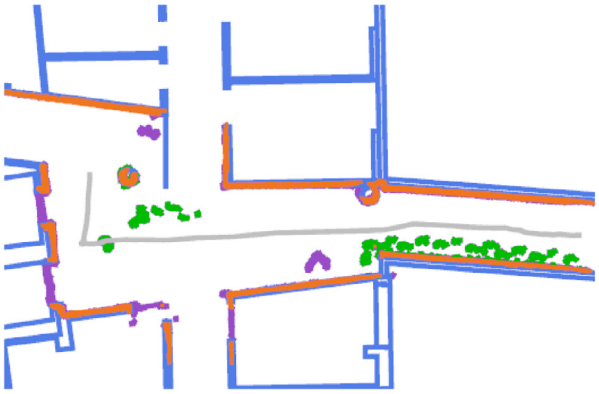

Fig. 2. Non-Markov nature of robot observations, shown by plotting laser rangefinder observations over multiple time steps, and registered by Episodic non-Markov localization to a static map (blue lines). The observations include observations that can be explained by the map (orange points), observations that can be explained by movable, but currently static objects like a potted plant and doors (purple points), and moving objects like humans (green points). The robot trajectory is plotted in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1. If there are no locations on the map that match the observations, the localization algorithm will be unable to track the location of the robot.
2. If the observations of movable objects coincidentally match a feature on the map (e.g., if the robot matches an observed plant pot to a pillar on the map), then the localization algorithm will estimate the pose of the robot incorrectly and with overconfidence.

If unexpected observations are explicitly removed from the computation of the belief by filtering the observations [6], then the incorrect and overconfident estimation of pose will be less likely, but the uncertainty of the robot's location will still grow.

Existing approaches to mobile robot localization in varying environments may be classified into three broad categories: Map-Update approaches, Local Static Map approaches, and Dynamic Map approaches.

Map-Update approaches.

Map-Update approaches model the environment as a static map, but update the static map when new observations conflict with the last map estimate. Dynamic Pose Graph SLAM (DPG-SLAM) [19] augments pose graph SLAM with indicators for every node to indicate parts of the observations from that node as being "static", "added", or "removed". The indicators for every node are updated after every pass of the robot through the environment, and the latest estimate of the map is recovered from the static and added observations from every pose node.

Local static map approaches.

Local Static Map approaches model changes to the environment by constructing local sub-maps that represent the observed change. The Patch Maps approach [20] maintains multiple sub-maps called "Patch Maps" of the different possible states of movable objects in the environment, like the different observed angles of a door. The resulting set of local maps represents observed configurations of the environment. For localization, the algorithm selects that local Patch Map that best represents the state of the environment at that moment. Temporary Maps [21] model the effect

of temporary objects by performing local SLAM, using the latest global map estimate as an initial estimate for the local map. Using these locally static maps, they perform localization with a particle filter. The Multiple Map Hypotheses approach [22] estimates the current state of the environment from a composite of prior maps, called the Long Term Memory (LTM), and the Short Term Memory (STM). Unexpected observations are initially added to the STM, and if they are seen persistently, later added to the LTM.

Dynamic map approaches.

Dynamic Map approaches attempt to capture the dynamics of the changes in the environment by incorporating them into the map. Dynamic Maps [23] extends SLAM and maintains estimates of the map over several timescales. In order to accommodate changes on the order of different timescales, the approach of Dynamic Maps maintains recency weighted samples of the map at several timescales simultaneously. The state of the environment is saved as multiple local maps at all predefined timescales. During deployments, the robot picks the local map for its current location from that timescale that best matches its present observations. Independent Markov Chain Maps [24] model a dynamic environment as an occupancy grid with associated independent Markov chains (iMac) with every cell on the grid. Each independent Markov Chain models the transition probabilities of the associated cell for transitioning from unoccupied to occupied, and vice-versa. The iMac occupancy grid thus models the environment by learning the probabilities of each cell remaining occupied *vs.* unoccupied. Rao-Blackwellized Particle Filters with Dynamic Occupancy Grids (RBPF-D) [25] simultaneously estimate the robot's pose in a varying environment, as well as the dynamics of the environment. The representation of the environment in RBPF-D is similar to iMac, where the dynamics of each cell in the occupancy grid map is modeled using a hidden Markov model.

2.4. Our approach

In contrast to the previous work, we introduce Episodic non-Markov Localization, which explicitly reasons about each observation as arising from three distinct classes (Section 3). The three classes correspond to permanent objects, termed Long-Term Features, or movable but temporarily static objects, termed Short-Term Features, or moving objects, termed Dynamic Features. The classifications of the observations into Long-Term, Short-Term, and Dynamic features are performed independently for each observation, and for each time step. Dynamic Features are not used for the computation of localization, but are reasoned about as being distinct from Long-Term or Short-Term Features, and are otherwise used for obstacle avoidance. We derive the expression for the localization belief for Episodic non-Markov Localization including the long-term features as well as the short-term features concurrently with the odometry (Section 4). To keep track of the correlations between the observations across different time steps, we introduce the Varying Graphical Network (Section 4.1), which extends the Dynamic Belief Network for Markov Localization to include correlations across timesteps due to observations of unmapped objects. The resulting localization belief is solved for the Maximum Likelihood Estimate by non-linear functional optimization of a cost function representation of the belief (Section 5). Thus, Episodic non-Markov Localization assumes neither that the world is static, nor that static objects will continue to remain static. The Long-Term Features are matched to a persistent map to provide global pose corrections, while Short-Term Features matched across time steps to provide relative pose corrections.

Map Update approaches attempt to estimate the latest state of all STFs and LTFs in the environment. Such approaches may be applicable for environments that change occasionally, so that updating the map might provide predictions of future observations. However, if the locations of movable objects in an area change between successive times that the robot visits that area, maintaining updated maps is of little use, since such maps would still not match future observations of the robot. A robot may be tasked to periodically re-visit all areas to mitigate such a limitation, but such a solution would require the robots to either periodically sacrifice task execution time in favor of mapping, or require separate specialized mapping robots.

Local Static Map approaches attempt to map the states of STFs and LTFs in small regions whenever changes are observed. Such approaches are effective when there is a finite set of observable configurations of movable objects. If, however, the set of configurations were intractably large or infinite (as is the case for objects like chairs in open spaces), it would be infeasible to maintain local static maps for *every* such possible configuration. Furthermore, local static maps assume that movable objects are static for the duration over which the local map is constructed, and thus do not address cases where static objects like chairs are moved while being observed.

Dynamic Map approaches attempt to model the dynamics of the STFs and DFs in the environment by augmenting the map representation to track the likelihood of STFs appearing or disappearing. Such approaches are effective when a robot is able to visit an environment often enough to capture the dynamics of the movable objects, and if there is a consistent trend to the dynamics. However, in large areas of deployments, it is infeasible for a robot to periodically visit all locations to build accurate dynamic maps. Furthermore, not all changes can be accurately modeled by their frequency, or transition probabilities: many changes in the environment are singular and unique, such as the occurrence of a crane in an atrium to repair a skylight.

Episodic non-Markov Localization does not rely on any map of the movable objects, and reasons about the Short-Term Features at runtime and on demand. It is thus still applicable in scenarios where the environment changes between successive deployments of the robot, or where the movable objects cannot be represented by locally static maps, or if the robot cannot observe and model the dynamics of the environment. It reasons about the unexpected observations being static or dynamic individually at every timestep, and is thus capable of handling cases where unmapped static objects start moving, or vice-versa.

3. Classification of observations

At each time step, Episodic non-Markov Localization classifies the observations from all poses as arising from Long-Term Features (LTFs), Short-Term Features (STFs), or Dynamic Features (DFs). Let x_i denote the pose of the robot, and s_i denote the observation, at time step t_i. Each observation s_i is a set of n_i 2D points $s_i = \{p_i^j\}_{j=1:n_i}$ observed by the robot. The points p_i^j may be acquired from a 2D laser rangefinder, or from an obstacle scan from a depth image [26]. The points p_i^j are in the robot's reference frame, and need to be transformed to the global reference frame to compare them with the map and other observation points from different timesteps. To aid with this transformation from the robot's reference frame to the global reference frame, we represent the pose x_i of the robot on the map at time-step i as an affine transform T_i that consists of the 2D rotation corresponding to the robot's angle at x_i, followed by a 2D translation corresponding to the robot's coordinates at x_i. Thus, for every 2D point p_i^j in the robot's reference frame, the corresponding location of the point in the global reference frame is given by $T_i p_i^j$. Each point p_i^j has to be classified as an LTF, STF, or DF. Fig. 3 demonstrates the steps in the classification of observations in an example with two poses.

3.1. Classification of long-term features

We use a vector map [5] representation $M = \{l_i\}_{1=1:s}$ for the permanent map, consisting of a set of s line segments l_i. To evaluate which of the observed points p_k^j are LTFs, an analytic ray cast [27] is performed from the latest estimate of x_i. The result of the analytic ray cast is a mapping from $p_i^j \rightarrow l_j, l_j \in M$, indicating that the observed point p_i^j is most likely to correspond to an observation of the line segment l_j. Let $dist(p, l)$ denote the perpendicular distance of point p from the line segment l where both p and l are in the reference frame of the map. The observation likelihood $P(p_i^j|x_i, M)$ of the point p_i^j is then given by

$$P(p_i^j|x_i, M) = \exp\left(-\frac{\text{dist}(T_i p_i^j, l_j)^2}{\Sigma_s}\right), \tag{5}$$

where Σ_s is the scalar variance of observations, which depends on the accuracy of the sensor used. Thus, given the location of the observed points in the reference frame of the map, observations are classified as LTFs if the observation likelihood of the point given the map is greater than a threshold, $P(p_i^j|x_i, M) > \epsilon_{\text{LTF}}$. Observed points p_i^j that satisfy this condition are classified as LTFs, and those that do not are classified as non-LTFs. The set $\text{LTF}_i \subseteq s_i$ denotes the set of points in s_i that have been classified as LTFs, and $\overline{\text{LTF}}_i$ the set of points that have been classified as non-LTFs. The sets LTF_i and $\overline{\text{LTF}}_i$ are thus given by

$$\text{LTF}_i = \left\{p_i^j \in s_i | P(p_i^j|x_i, M) > \epsilon_{\text{LTF}}\right\}, \tag{6}$$

$$\overline{\text{LTF}}_i = s_i \setminus \text{LTF}_i. \tag{7}$$

3.2. Classification of short-term features

Observed points that are classified as non-LTFs could potentially be STFs. To check if an observed point $p_i^j \in \overline{\text{LTF}}_i$ is an STF, it is compared to all non-LTF points observed prior to time-step i to check if they correspond to observations of the same point. Given a point $p_i^j \in \overline{\text{LTF}}_i$ observed at time-step i and another point $p_k^l \in \overline{\text{LTF}}_k$ observed at a previous time-step k, the probability that both the observations correspond to the same point is given by the STF observation likelihood function,

$$P(p_i^j, p_k^l|x_i, x_k) = \exp\left(-\frac{\|T_i p_i^j - T_k p_k^l\|^2}{\Sigma_s}\right), \tag{8}$$

where Σ_s is the scalar variance of observations. Let p_k^l be the closest neighboring point to p_i^j from the non-LTFs of other timesteps, defined as

$$p_k^l = \underset{p_k^l \in \overline{\text{LTF}}_l \forall l \neq i}{\arg\min} \|T_i p_i^j - T_l p_k^l\|. \tag{9}$$

Therefore, a non-LTF point $p_i^j \in \overline{\text{LTF}}_i$ is classified as an STF if the closest neighboring point $p_k^l \in \overline{\text{LTF}}_l$ from a time-step l, $l < i$ such that $P(p_i^j, p_k^l|x_i, x_l) > \epsilon_{\text{STF}}$:

$$\text{STF}_i = \left\{p_i^j \in \overline{\text{LTF}}_i | \exists p_k^l \in \overline{\text{LTF}}_l : P(p_i^j, p_k^l|x_i, x_l) > \epsilon_{\text{STF}}\right\}. \tag{10}$$

To speed up the correspondence check, points $p_i^j \in s_i$ from every observation s_i are stored in KD-trees [28]. The STF observation likelihood function for a pair of matching STF observations p_i^j and p_k^l introduces a correlation between observations s_i and s_l since $p_i^j \in s_i$ and $p_k^l \in s_l$. Note that at the first timestep that an STF is encountered, since it does not match any observations from other timesteps, it will be classified as a DF (as discussed in the next section). However, it will be classified as an STF from the next timestep on, as the robot makes additional observations that match the STF.

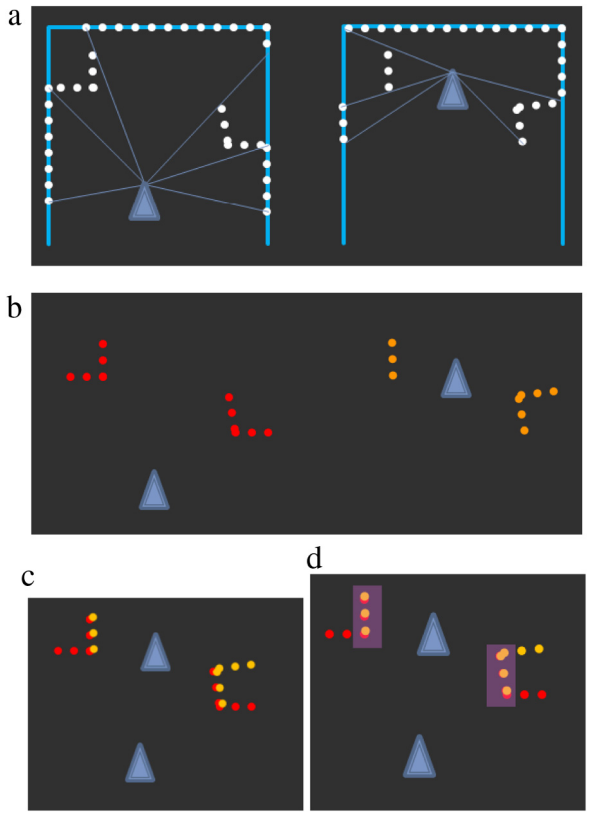

Fig. 3. An example of classification of observations by EnML from two poses. The robot poses are shown as blue triangles, and the observations are shown as dots. Observations from two poses (a) are first independently compared to the vector map (shown in blue) to find LTF observations. The remaining non-LTF observations (b) are aligned (c) with the last pose prior, and (d) matching observations are classified as STFs. The remaining observations are classified as DFs.

3.3. Classification of dynamic features

Observations that are classified as neither LTFs nor STFs correspond to objects that were observed at one particular location for only one particular time-step, and were not observed at any other time-step at the same location. This implies that these objects are not static, and hence their observations are classified as DFs, $\text{DF}_i = s_i \setminus \text{LTF}_i \setminus \text{STF}_i$. In this work, we do not actively track DFs or use them for localization, so the correlations between observations due to DFs are not used to further refine the belief. However, points that are classified as DFs in one pose, at one iteration of the EnML update may subsequently be matched with points observed at future poses, at which time they will become STFs.

3.4. Discussion

The classification of observations, as introduced, allows Episodic non-Markov Localization to treat observations differently based on the nature of the objects being observed. Such an approach is in contrast to that of SLAM, where all observations are treated equally, irrespective of whether they are of permanent objects like the walls, or whether they are of movable objects like tables, which are likely to move from their perceived locations. Furthermore, by keeping track of the observations over time, and reasoning about which spaces have been seen as being open in the

past, it would be possible to update the permanent map to include LTFs that have consistently been observed as STFs at the same location, and which the robot has never been able to see through. The observation classification thresholds ϵ_{LTF} and ϵ_{STF} are related to the sensor noise represented by the variance of the observations Σ_s, and the confidence interval of classifications. Tighter confidence bounds on ϵ_{LTF} and ϵ_{STF} result in fewer mis-classification of observations, but also results in more DF classifications due to sensor noise and pose uncertainties. For the deployments of the CoBots (Section 7.3), the laser rangefinder sensor had an experimentally calculated variance of $\Sigma_s = 0.0025\ \text{m}^2$, and the classification thresholds were set to the 99.7% acceptance confidence interval, corresponding to $\epsilon_{\text{LTF}} = \epsilon_{\text{STF}} = 0.09$.

4. Episodic non-Markov localization

We first derive the expression for the localization belief in Episodic non-Markov Localization, and introduce the non-Markov terms in the belief. Next, in Section 4.1 we introduce the Varying Graphical Network, a novel graphical representation to track the correlations between observations from different time-steps arising from the non-Markov terms in the belief. In Section 4.2 we introduce the concept of *episodes* to limit the history of poses, odometry and observations that must be tracked to maintain the belief in Episodic non-Markov Localization.

Given the initial pose of the robot x_0, observations $s_{1:n}$, odometry $u_{1:n}$, and a static map M, the belief over the robot's trajectory $x_{1:n}$ is given by

$$Bel(x_{1:n}) = P(x_{1:n}|s_{1:n}, x_0, u_{1:n}, M). \tag{11}$$

Applying Bayes' rule and separating out the terms involving sensing and odometry, we get

$$\begin{aligned} Bel(x_{1:n}) &= P(x_{1:n}|s_{1:n}, x_0, u_{1:n}, M) \\ &= \frac{P(x_{1:n}, s_{1:n}|x_0, u_{1:n}, M)}{P(s_{1:n}|x_0, u_{1:n}, M)} \\ &\propto P(s_{1:n}|x_{1:n}, M)P(x_{1:n}|x_0, u_{1:n}). \end{aligned} \tag{12}$$

Note that the odometry observations are independent at each step, so the pose update terms are independent of each other, resulting in

$$Bel(x_{1:n}) \propto P(s_{1:n}|x_{1:n}, M) \prod_{i=1}^{i=n} P(x_i|x_{i-1}, u_i). \tag{13}$$

Each observation s_i is a set of multiple points arising from a sensor like a laser range-finder or a depth sensor. As introduced in Section 3, each of these points are from either Long-Term Features (LTFs), or Short-Term Features (STFs), or Dynamic Features (DFs), so the observation s_i can be expressed as a union of the observations of these three types of features:

$$s_i = s_i^{\text{LTF}} \cup s_i^{\text{STF}} \cup s_i^{\text{DF}}. \tag{14}$$

Exploiting the fact that different classes of observations (LTFs, STFs and DFs) are independent of each other, the belief for Episodic non-Markov Localization thus becomes

$$\begin{aligned} &Bel(x_{1:n}) \\ &\propto P(s_{1:n}^{\text{LTF}}|x_{1:n}, M)P(s_{1:n}^{\text{STF}}|x_{1:n})P(s_{1:n}^{\text{DF}}|x_{1:n}) \prod_{i=1}^{i=n} P(x_i|x_{i-1}, u_i). \end{aligned} \tag{15}$$

We do not track DFs for localization (but use them for obstacle avoidance), so the terms involving DFs are ignored and assumed to be taken on a constant value. The observations of LTFs at each timestep, given the pose estimate of the same timestep and the map, are independent of observations from other timesteps. Furthermore, the observations of different STFs are independent of each other as well, resulting in,

$$Bel(x_{1:n}) \propto \prod_{i=1}^{i=n} \left[P(s_i^{\text{LTF}}|x_i, M)\right] P(s_{1:n}^{\text{STF}}|x_{1:n}) \prod_{i=1}^{i=n} P(x_i|x_{i-1}, u_i) \tag{16}$$

$$\propto P(s_{1:n}^{\text{STF}}|x_{1:n}) \prod_{i=1}^{i=n} \left[P(s_i^{\text{LTF}}|x_i, M)P(x_i|x_{i-1}, u_i)\right]. \tag{17}$$

Let the number of sets of STFs observed be m. Each set of STFs corresponds to observations of the same unmapped object, as seen from different poses. Each set of STFs introduces correlations between the poses from which that unmapped object was observed. The terms corresponding to the observations of sets of STFs can thus be further decoupled, since each set of STFs, corresponding to a different unmapped object, is independent of all the other sets of STFs. Note however, that observations in the same set of STFs are not independent, as they belong to the same object, and could potentially be related to each other, as determined by the nearest-neighbor search for classifying STFs (Section 3.2). The belief thus factorizes as,

$$Bel(x_{1:n}) \propto \prod_{j=1}^{i=m} \left[P(s_{1:n}^{\text{STF}_j}|x_{1:n})\right] \prod_{i=1}^{i=n} \left[P(s_i^{\text{LTF}}|x_i, M)P(x_i|x_{i-1}, u_i)\right]. \tag{18}$$

This expression is the foundation for episodic non-Markov localization. There are two parts to this expression: the first part consists of a product of m terms $P(s_{1:n}^{\text{STF}_j}|x_{1:n})$ corresponding to the observation of m STFs, and the second part consists of a product of n LTF observation likelihood terms $P(s_i^{\text{LTF}}|x_i, M)$ and n odometry terms $P(x_i|x_{i-1}, u_i)$. The LTF observation likelihood terms and the odometry terms in Episodic non-Markov Localization are identical to the observation likelihood and odometry terms, respectively, in Markov Localization. If the robot does not observe any unmapped observations ($m = 0$), the only terms remaining will be the n independent terms corresponding to the observations of LTFs, and odometry. Thus, *in the special case where there are no unexpected observations, Episodic non-Markov Localization reduces to Markov Localization*. The computation of the observation likelihoods for the STFs ($P(s_{1:n}^{\text{STF}_j}|x_{1:n})$) and the LTFs ($P(s_i^{\text{LTF}}|x_i, M)$), and the odometry likelihoods ($P(x_i|x_{i-1}, u_i)$) is explained in detail in Section 5.

The STFs introduce correlations between multiple poses. The exact set of such correlations are not known a-priori since they depend on the exact configurations of unmapped objects in the environment, and hence must be estimated at run time. To represent the varying nature of localization in the presence of unmapped observations, we introduce a new graphical model, the *Varying Graphical Network*.

4.1. The varying graphical network representation

As in the Dynamic Bayesian Network (Section 2) for Markov Localization, a Varying Graphical Network (VGN) includes certain periodically repeating nodes and edges that do not change with the belief. We term these the non-varying nodes and edges. A VGN includes two additional structural elements: varying nodes and varying edges. The presence and structure of the varying nodes and varying edges are not known a-priori, and are estimated jointly with the belief. Since the estimates of the structure may change with the belief, the structure is likely to change as new observations become available.

VGNs provide an accurate representation for non-Markov localization. The presence of LTFs and their relations to the map, and the correlations between successive poses of the robot due to odometry observations are encoded by the non-varying edges and

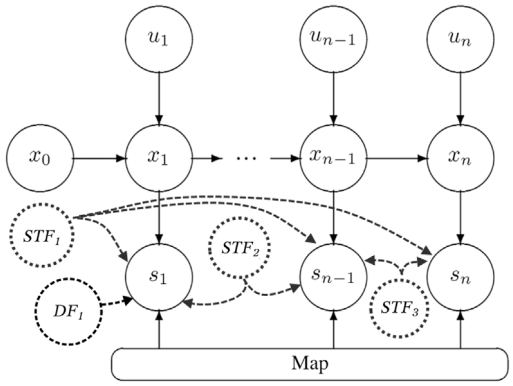

Fig. 4. An example instance of a Varying Graphical Network (VGN) for non-Markov localization. The non-varying nodes and edges are denoted with solid lines, and the varying nodes and edges with dashed lines. Due to the presence of short term features (STFs) and dynamic features (DFs), the structure is no longer periodic in nature, and unknown a priori.

nodes. The presence of STFs and DFs is encoded by the presence of associated varying nodes. STFs are represented as varying nodes, where the varying node for each STF is connected by multiple varying edges to the observations from those time-steps that the STF is observed. DFs are represented as varying nodes, where the varying node for each DF is connected by a single varying edge to the observation from that time-step that the DF is observed. Since EnML does not track the DFs, the correlations of observations of DFs across timesteps, and hence varying edges that connect varying nodes for DFs to observations of different timesteps, are not evaluated. Fig. 4 shows an example instance of a VGN for non-Markov localization.

Looking back at Eq. (18), the first part consists of the STF terms $P(s_{1:n}^{\text{STF}_j}|x_{1:n})$ that arise from the *varying nodes* and *varying edges* of the Varying Graphical Network, while the second part consists of the LTF terms $P(s_i^{\text{LTF}}|x_i, M)$ and odometry terms $P(x_i|x_{i-1}, u_i)$ that arise from the *repeating nodes* and *repeating edges* of the Varying Graphical Network. For every pair of matching STF observations as introduced in Section 3.2, the VGN of non-Markov localization includes a varying node representing the unmapped static object, and a pair of varying edges joining the varying node and each of the observations from the different timesteps.

The varying structure of the VGN, including the varying nodes consisting of the STFs and DFs and the varying edges indicating correlations between STFs and observations, is not enumerable a priori since there is no way of predicting beforehand the state of the unmapped static and dynamic objects in an environment. The structure of the VGN is dependent on the exact locations of the STFs, the trajectories of the DFs, and from which poses of the robot's trajectory the STFs and DFs are visible. The STFs and DFs cannot, in general, be added to the map since for any pose in the world, the robot will encounter different STFs and DFs at different times.

Since the VGN for non-Markov localization has no predefined structure, it might seem that computation of the belief would require storing the complete history of all states and observations since the robot was turned on. However, in practice this is not necessary, as we shall now show.

4.2. Episodes in non-Markov localization

Suppose there exists a time step t_i such that all observations and state estimates made after t_i, given x_i, are independent of all prior observations and state estimates:

$$P(x_{1:n}|x_0, s_{1:n}, u_{1:n}, M) = P(x_{1:i}|x_0, s_{1:i}, u_{1:i}, M) \times P(x_{i+1:n}|x_i, s_{i+1:n}, u_{i+1:n}, M). \quad (19)$$

This conditional independence implies that there are no STF observations after t_i that correspond to STF observations before t_i. In such a case, the history of states and observations prior to t_i, called the "episode" $t_{0:i-1}$, can be discarded when estimating $Bel(x_{i:n})$ over the episode $t_{i:n}$. We thus define an episode $t_{j:k}$ to be a consecutive sequence of time-steps from t_j to t_k such that the observations made between t_j and t_k are independent of all observations made before t_j, given the pose x_j, and the permanent map M. We assume such episode-boundary time-steps t_j exist, allowing real-time non-Markov localization with limited computational resources. Episode-boundary time-steps frequently occur in practice when a robot either does not observe any STFs for one or more time-steps, or if all the STFs prior to the episode-boundary are unrelated to the STFs after, for example when a robot leaves one room and enters another through a doorway. Fig. 5 shows an example VGN near an episode boundary, highlighting the absence of any varying edges crossing the episode boundary.

A special case of such a transition is when the robot does not encounter any STFs or DFs for a sequence of time steps from t_i to t_j, so that

$$P(x_{1:n}|x_0, s_{1:n}, u_{1:n}, M) = P(x_{1:i}|x_0, s_{1:i}, u_{1:i}, M) \times \prod_{k=i+1}^{k=j} P(x_k|x_{k-1}, u_k, s_k, M) P(x_{j+1:n}|x_j, s_{j+1:n}, u_{j+1:n}, M). \quad (20)$$

In this case, from t_i to t_j, the Markov assumptions hold.

Thus, the history of observations and state estimates of the robot can be divided into episodes such that only observations and state estimates of the latest episode need be considered to estimate the latest robot pose x_n.

5. Representation of the belief

The dimension of the state space of the belief for EnML $Bel(x_{1:n})$ over the complete history of robot poses $x_{1:n}$ is dn, where d is the dimension of the state space of each pose x_i. For ground robots, $d = 3$, corresponding to the Cartesian coordinates of the robot location, and the robot angle. Due to the prohibitively large state space, we concentrate on evaluating the belief in the neighborhood of its maximum likelihood estimate (MLE). The MLE of the belief is given by $x_{1:n}^*$ such that

$$x_{1:n}^* = \arg\max_{x_{1:n}} (Bel(x_{1:n})). \quad (21)$$

There are two challenges to maintaining the belief as thus described:

1. The dimensionality of the state space is *considerably larger* than for other localization algorithms. For example, for a ground robot, while the state space of other localization algorithms is 3 dimensional (x, y, rotation), for an episode length of 10 poses EnML has a state space of 30 dimensions.
2. The dimensionality of the state space is *variable* for EnML. The dimensions of the state space of EnML increases as new poses are added to the latest episode, and decreases when a new episode boundary is discovered, resulting in the latest episode being shortened.

Due to these challenges, commonly used belief representations such as Particle Filters, EKFs, UKFs, and POMDPs are ill-suited to being used for EnML. Particle filters and POMDPs are particularly poor choices for the belief representation of EnML since the number of particles required by particle filters, and the size of the POMDP, both scale exponentially with the dimensions of the state space. To tractably estimate the MLE of the belief in EnML, we therefore introduce a cost function representation of the belief. The

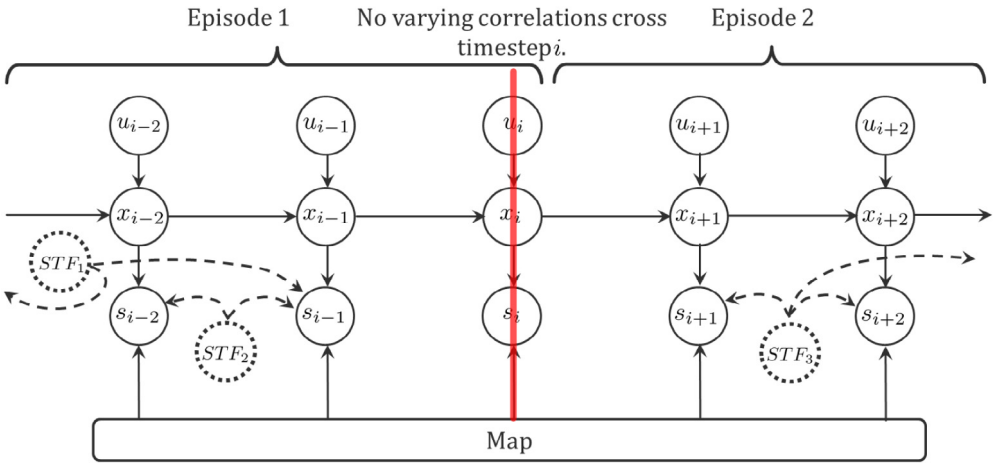

Fig. 5. An example VGN demonstrating the presence of an episode in non-Markov localization. Note the absence of any varying edges that cross the red line indicating the episode boundary. Hence the pose x_i is an episode boundary, where all previous poses up to x_{i-1} are in the previous episode, and poses x_i and later are in the latest episode. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

components of cost function representation, as we shall present, relate directly to the different terms in the expression of the belief, and the MLE can be tractably estimated by functional non-linear least-squares (NLLS) optimization of the cost function.

5.1. Belief as a cost function

Recall from Eq. (21) that we wish to compute the Maximum Likelihood Estimate (MLE) $x^*_{1:n}$ such that

$$x^*_{1:n} = \arg\max_{x_{1:n}}(Bel(x_{1:n})) \tag{22}$$

$$= \arg\max_{x_{1:n}} \left(\prod_{j=1}^{i=m} \left[P(s^{\text{STF}_j}_{1:n} | x_{1:n}) \right] \times \prod_{i=1}^{i=n} \left[P(s^{\text{LTF}}_i | x_i, M) P(x_i | x_{i-1}, u_i) \right] \right). \tag{23}$$

To estimate the MLE of the belief, we first convert the belief from a probability distribution representation to a cost function representation C such that

$$Bel(x_{1:n}) = P(x_{1:n} | x_0, s_{1:n}, u_{1:n}, M) \propto \exp(-C(x_{1:n} | x_0, s_{1:n}, u_{1:n}, M)). \tag{24}$$

The cost function C consists of a sum of m sub-cost functions c^{STF}_j corresponding to the STF terms $P(s^{\text{STF}_j}_{1:n} | x_{1:n})$, n sub-cost functions c^{LTF}_i corresponding to the LTF terms $P(s^{\text{LTF}}_i | x_i, M)$, and n sub-cost functions c^{odom}_i corresponding to the odometry terms $P(x_i | x_{i-1}, u_i)$. The complete expression for C is thus given by

$$C(x_{1:n} | x_0, s_{1:n}, u_{1:n}, M) = \sum_{j=1}^{j=m} \left(c^{\text{STF}}_j (s_{1:n} | x_{i:n}) \right) + \sum_{i=1}^{i=n} \left(c^{\text{LTF}}_i (s_i | x_i, M) + c^{\text{odom}}_i (x_i | x_{i-1}, u_i) \right). \tag{25}$$

The MLE is therefore computed by minimizing the cost function as

$$x^*_{1:n} = \arg\min_{x_{1:n}} (C(x_{1:n} | x_0, s_{1:n}, u_{1:n}, M)). \tag{26}$$

There are two important properties of the cost-function representation of the belief that make it a better representation for computing the MLE rather than the probability distribution representation.

1. **Quadratic Form:** An important property of the sub-cost functions that we exploit is that they are all *purely quadratic in form*. The quadratic forms of the sub-cost functions are introduced in the next section. Since all the sub-cost functions are purely quadratic, Eq. (26) can be solved by functional non-linear least-squares (NLLS) optimization.
2. **Numerical Stability:** The sub-cost functions scale as the log of the probability values of the corresponding terms in the belief, and are thus less affected by numerical precision errors. Furthermore, the sub-cost values are *added* instead of *multiplied*, as in the case of probability values, thus further avoiding numerical precision errors.

Before we proceed to present the forms of the sub-cost functions, we first summarize here the steps we took in deriving a cost-function form of the belief.

1. The belief is first factored into the odometry and sensing terms (Eq. (12)).
2. The sensing terms are further factorized into terms corresponding to the STFs and LTFs (Eq. (15)).
3. The LTF and odometry terms are decomposed into independent terms for each pose (Eq. (18)).
4. The belief is then represented in terms of a cost function, with each independent term in the belief corresponding to a sub-cost function (Eq. (25)).
5. The MLE is then computed by functional NLLS optimization of the cost function (Eq. (26)).

The computation of each individual sub-cost function is the subject of the next section.

5.2. Computation of sub-cost functions

The sub-cost functions are computed and stored as *function objects* [29], not as function values, thus allowing the arg min over C to be computed in real time by non-linear optimization of cost functions expressed as function objects [30]. Since the optimizer performs algorithmic differentiation [31] of the function objects, it provides the same accuracy as symbolic optimization, with much lower computational requirements.

The odometry sub-cost function c_i^{odom} relating poses x_i and x_{i-1} with observed odometry u_i is given by

$$c_i^{\text{odom}} = (x_i \ominus (x_{i-1} \oplus u_i))^T \Sigma_o^{-1} (x_i \ominus (x_{i-1} \oplus u_i)) \tag{27}$$

where Σ_o is the covariance on odometry based on the motion model of the robot, and $\oplus, \ominus$ denote the pose composition and decomposition operators for robot poses and odometry in the SE(2) group [32]. Pose composition $x = x_i \oplus x_j$ is equivalent to the multiplication of the corresponding affine transforms, $T = T_j T_i$, and pose decomposition $x = x_i \ominus x_j$ is equivalent to the inverse-multiplication of the corresponding affine transforms, $T = T_j^{-1} T_i$.

The sub-cost function c_i^{LTF} for LTFs for pose i depends on the observations of LTFs $s_i^{\text{LTF}} \in s_i$ made at time-step t_i. Let there be m LTF observations in s_i^{LTF}, and for each point $p_i^j \in s_i^{\text{LTF}}, j \in [1, m]$, let the corresponding line segment in the vector map M be $l_j \in M$. Let $dist(p, l)$ indicate the perpendicular distance of point p from line segment l. The LTF sub-cost function for pose i is then given by

$$c_i^{\text{LTF}} = \sum_{j=1}^{j=m} \left[\frac{dist(T_i p_i^j, l_j)^2}{\Sigma_s} \right], \tag{28}$$

where T_i denotes the affine transform for pose x_i as described in Section 3, and Σ_s is the variance of the range sensor. Note that the LTF sub-cost function is derived from the observation likelihood of LTFs, Eq. (5), introduced in Section 3.1.

Let STF j consist of a pair of points from poses i and k, with point $p_i \in s_i^{\text{STF}}$ being the point from pose i and point $p_k \in s_k^{\text{STF}}$ from pose k. The sub-cost function c_j^{STF} is then given by

$$c_j^{\text{STF}} = \frac{\|T_i p_i - T_k p_k\|^2}{\Sigma_s}. \tag{29}$$

The STF sub-cost function is derived from the pair-wise observation likelihood of STFs, Eq. (8), introduces in Section 3.2. Note that all the sub-cost functions (Eqs. (27)–(29)) are purely quadratic in form, as required for non-linear least-squares optimization.

6. Computational structure of the belief in episodic non-Markov localization

Thus far we have introduced the EnML algorithm as agnostic of the algorithm that performs the non-linear functional optimization of the cost function C. In principle, any general non-linear optimization algorithm may be used to solve Eq. (26) to estimate the MLE $x_{1:n}^*$. We use Ceres-Solver [30] to solve Eq. (26) using the Levenberg–Marquardt (LM) algorithm [33,34] with Sparse Normal Cholesky [35] as the linear solver. In this section, we examine the structure of EnML from the view of the non-linear solver solving Eq. (26). In particular, we investigate how the concept of episodes in the VGN translates to the solutions of the non-linear solver, and how the structure of the problem relates to SLAM and Markov localization.

The cost function C can be re-cast as

$$C(x_{1:n}) = \frac{1}{2} \|F(x_{1:n})\|^2. \tag{30}$$

This construction is possible because all of the sub-cost functions c_i (Eqs. (27)–(29)) that comprise C as $C = \sum_i (c_i)$ are of the purely quadratic form $c_i = \mathbf{a}^T \Sigma \mathbf{a}$, as described in Section 5.2. Therefore, the corresponding entries in F would be $\Sigma^{-\frac{1}{2}} \mathbf{a}$. In this form, Eq. (26) is rephrased as

$$x_{1:n}^* = \arg\min_{x_{1:n}} \frac{1}{2} \|F(x_{1:n})\|^2. \tag{31}$$

The solution to this equation is thus a non-linear least squares optimization. Global solutions to this equation are computationally intractable due to the presence of local minima arising from non-linearities in the arguments $\mathbf{a}$ for each sub-cost function c_i. However, the local optimum may be found in the neighborhood of an initial estimate. For EnML, the initial estimate for the solution of $x_{1:n}^*$ is taken from the last estimates of $x_{1:n-1}^*$ and the odometry-predicted estimates of x_n.

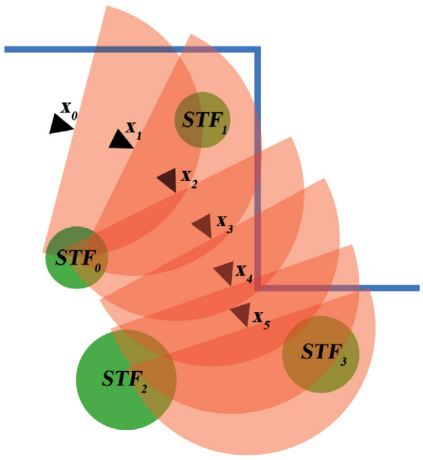

Fig. 6. An example robot scenario running EnML. There are 6 poses $x_0 - X_5$ denoted by the triangular robot markers, along with the maximum sensor range denoted by the translucent orange semicircles. There are four STFs, $STF_0 - STF_3$, shown by the green circles. The long-term static map is shown by the blue lines.

Non-linear least-squares solutions to Eq. (31) by LM are computed iteratively, with each iteration updating x^* proportional to Δx, where Δx is the solution of the equation

$$J^T J \Delta x = J^T F. \tag{32}$$

Here, J is the Jacobian of F, the elements of which are given by

$$J_{ij}(x) = \frac{\delta F_i}{\delta x_j}(x). \tag{33}$$

The matrix $J^T J$ is called the information matrix $\mathcal{I}$. The solution to Eq. (32) may be found by inverting the information matrix, although in practice alternative solutions like the QR decomposition or LU decomposition of $\mathcal{I}$ yield more numerically stable as well as computationally efficient solutions, in particular for sparse matrices. The structure of the information matrix $\mathcal{I}$, as we shall show, indicates the correlations between the poses of the robot in the EnML episode. We shall also show that the computational complexity of EnML scales from being comparable to Markov Localization in the absence of short-term features, to being comparable to pose graph SLAM when only short-term features are visible.

We present here a simple scenario of EnML running on a robot. Fig. 6 shows a robot navigating in an environment with four static unmapped objects. There are six poses of the robot over time, and at each pose the robot observes one or more of the unmapped objects, as well as part of the static map. Thus, in this scenario, there would be 15 sub-cost functions: 5 sub-cost functions $c_1^o - c_5^o$ for the odometry between successive poses, 4 sub-cost functions for the STFs $c_0^{\text{STF}} - c_3^{\text{STF}}$, and 6 sub-cost functions $c_0^{\text{LTF}} - c_5^{\text{LTF}}$ for the LTFs from each pose.

The structure of the Jacobian and the information matrix is thus as shown in Table 1. Note that in the Jacobian, the entries corresponding to cost functions span multiple rows for each cost function, since each cost function depends on multiple observed points. Furthermore, every entry in the information matrix corresponds to a 3×3 block, corresponding to the 3 degrees of freedom of the robot.

There are several interesting features about the structure of the information matrix. It is sparse and nearly block-diagonal with two

Table 1
The structure of the Jacobian (left) and the information matrix (right) for EnML in the example robot scenario. Non-zero entries are marked with a 'x', while zero entries are marked with a '.'.

	x_0	x_1	x_2	x_3	x_4	x_5
c_1^o	x	x	.	.	.	.
c_2^o	.	x	x	.	.	.
c_3^o	.	.	x	x	.	.
c_4^o	.	.	.	x	x	.
c_5^o	.	.	.	.	x	x
c_0^{STF}	x	x	x	.	.	.
c_1^{STF}	x	x	.	.	.	.
c_2^{STF}	.	.	.	x	x	x
c_3^{STF}	.	.	.	x	x	x
c_0^{LTF}	x	.	.	.	.	.
c_1^{LTF}	.	x	.	.	.	.
c_2^{LTF}	.	.	x	.	.	.
c_3^{LTF}	.	.	.	x	.	.
c_4^{LTF}	.	.	.	.	x	.
c_5^{LTF}	.	.	.	.	.	x

	x_0	x_1	x_2	x_3	x_4	x_5
x_0	x	x	x	.	.	.
x_1	x	x	x	.	.	.
x_2	x	x	x	x	.	.
x_3	.	.	x	x	x	x
x_4	.	.	.	x	x	x
x_5	.	.	.	x	x	x

major blocks $x_{0:2} \times x_{0:2}$ and $x_{3:5} \times x_{3:5}$. In fact, the only non-block diagonal elements arise from the relation between x_2 and x_3 due to the odometry sub-cost functions. Note also that in this scenario, x_3 would be an episode boundary: all the observations of STFs at and after pose x_3 are independent of the STFs observed before x_3. This is not just coincidence. The existence of episode boundaries enforce this almost block-diagonal structure to the information matrix by virtue of the fact that *except for odometry, correlations between poses arise only from* STFs, and by definition, no such correlations may cross the episode boundary. This nearly block-diagonal structure has a particularly simple sub-solution. Consider the information matrix $\mathcal{I}$ as composed of blocks A, B, C, D such that

$$\mathcal{I} = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}, \tag{34}$$

$$A = J_{0:14,0:2}^T J_{0:14,0:2}, \tag{35}$$

$$B = J_{0:14,0:2}^T J_{0:14,3:5}, \tag{36}$$

$$D = J_{0:14,3:5}^T J_{0:14,3:5}. \tag{37}$$

Here, the notation $J_{a:b,c:d}$ refers to the sub-matrix of J consisting of rows a to b and columns c to d. The information matrix $\mathcal{I}$ may then be inverted by the matrix block-inversion formula,

$$\mathcal{I}^{-1} = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^{-1} \tag{38}$$

$$= \begin{bmatrix} S_D^{-1} & -A^{-1}BS_A^{-1} \\ -D^{-1}B^T S_D^{-1} & S_A^{-1}, \end{bmatrix} \tag{39}$$

where S_A and S_D are the Schur complements of A and D respectively, given by

$$S_A = D - B^T A^{-1} B, \tag{40}$$

$$S_D = A - BD^{-1}B^T. \tag{41}$$

Let the only non-zero element in the lower left corner of B be given by b. We now investigate the structure of S_A.

$$S_A = D - \begin{bmatrix} 0 & 0 & b^T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} A^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b & 0 & 0 \end{bmatrix} \tag{42}$$

$$= D - \begin{bmatrix} 0 & 0 & b^T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_{00}^{-1} & A_{01}^{-1} & A_{02}^{-1} \\ A_{10}^{-1} & A_{11}^{-1} & A_{12}^{-1} \\ A_{20}^{-1} & A_{21}^{-1} & A_{22}^{-1} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b & 0 & 0 \end{bmatrix} \tag{43}$$

Table 2
The structure of the Jacobian and the information matrix for EnML in the no-map example robot scenario.

	x_0	x_1	x_2	x_3	x_4	x_5
c_1^o	x	x	.	.	.	.
c_2^o	.	x	x	.	.	.
c_3^o	.	.	x	x	.	.
c_4^o	.	.	.	x	x	.
c_5^o	.	.	.	.	x	x
c_0^{STF}	x	x	x	.	.	.
c_1^{STF}	x	x	.	.	.	.
c_2^{STF}	.	.	.	x	x	x
c_3^{STF}	.	.	.	x	x	x
c_4^{STF}	x	x	.	.	.	.
c_5^{STF}	.	x	x	x	x	.
c_6^{STF}	.	.	.	.	.	x
c_0^{LTF}	.	.	.	.	.	.
c_1^{LTF}	.	.	.	.	.	.
c_2^{LTF}	.	.	.	.	.	.
c_3^{LTF}	.	.	.	.	.	.
c_4^{LTF}	.	.	.	.	.	.
c_5^{LTF}	.	.	.	.	.	.

	x_0	x_1	x_2	x_3	x_4	x_5
x_0	x	x	x	.	.	.
x_1	x	x	x	x	x	.
x_2	x	x	x	x	x	.
x_3	.	x	x	x	x	x
x_4	.	x	x	x	x	x
x_5	.	.	.	x	x	x

$$= D - b^2 \begin{bmatrix} b^T A_{20}^{-1} & b^T A_{21}^{-1} & b^T A_{22}^{-1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{44}$$

$$= D - \begin{bmatrix} b^T A_{22}^{-1} b & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{45}$$

Hence the computation of the Schur complements S_A, S_D are trivial computations, given A^{-1} and D^{-1}. Furthermore, for cases where $b^T A_{22}^{-1} b \ll D_{00}$, the Schur complement may be approximated as $S_A \approx D$, further simplifying the inversion of $\mathcal{I}$. The most computationally expensive operations in the inversion of $\mathcal{I}$ are therefore the computations of A^{-1} and D^{-1}, which can be computed independently of each other. Thus, *across an episode boundary, the computations of the inverse of the dominant sub-matrices of* $\mathcal{I}$, $S_D^{-1} \approx A^{-1}$ *and* $S_A^{-1} \approx D^{-1}$, *are independent of each other.*

We next examine the example scenario if the robot did not have a static map. We call this the "no-map" example scenario. The three linear features that would otherwise have been considered LTFs would instead be considered by EnML to be STFs. The associated cost functions are denoted by c_4^{STF}, c_5^{STF}, c_6^{STF}. The Jacobian and information matrix for the no-map example scenario are thus listed in Table 2. Note that the LTF cost functions are zero for all the poses. Alternatively, if the robot were running pose graph SLAM (e.g., [18]), all the features, including what EnML considered to be STFs and LTFs would be considered landmark features, and give rise to correlations between the poses. The corresponding Jacobian and information matrix are thus shown in Table 3. Note that the structure of the information matrix is identical between EnML in the no-map scenario, and that of SLAM. Thus, *in the absence of a static map, the structure of the information matrix, and hence the computational complexity of the Maximum Likelihood Estimation of EnML is identical to that of pose graph SLAM.*

Finally, we consider the case where the STFs are not present in the example scenario. This is a scenario perfect for Markov localization. There would be no sub-cost functions of EnML for STFs, and the corresponding Jacobian and information matrix for EnML would thus be as given in Table 4. Note here that the information matrix is band-diagonal in nature, thus allowing efficient inversion

Table 3
The structure of the Jacobian matrix and the information matrix for pose graph SLAM in the example robot scenario.

	x_0	x_1	x_2	x_3	x_4	x_5
c_1^o	x	x	.	.	.	.
c_2^o	.	x	x	.	.	.
c_3^o	.	.	x	x	.	.
c_4^o	.	.	.	x	x	.
c_5^o	.	.	.	.	x	x
c_0^l	x	x	x	.	.	.
c_1^l	x	x	.	.	.	.
c_2^l	.	.	.	x	x	x
c_3^l	.	.	.	x	x	x
c_4^l	x	.	.	.	.	.
c_5^l	.	x	x	x	x	.
c_6^l	.	.	.	.	.	x

	x_0	x_1	x_2	x_3	x_4	x_5
x_0	x	x	x	.	.	.
x_1	x	x	x	x	x	.
x_2	x	x	x	x	x	.
x_3	.	x	x	x	x	x
x_4	.	x	x	x	x	x
x_5	.	.	.	x	x	x

Table 4
The structure of the Jacobian and the information matrix for EnML in the all-mapped example robot scenario.

	x_0	x_1	x_2	x_3	x_4	x_5
c_1^o	x	x	.	.	.	.
c_2^o	.	x	x	.	.	.
c_3^o	.	.	x	x	.	.
c_4^o	.	.	.	x	x	.
c_5^o	.	.	.	.	x	x
c_0^{LTF}	x	.	.	.	.	.
c_1^{LTF}	.	x	.	.	.	.
c_2^{LTF}	.	.	x	.	.	.
c_3^{LTF}	.	.	.	x	.	.
c_4^{LTF}	.	.	.	.	x	.
c_5^{LTF}	.	.	.	.	.	x

	x_0	x_1	x_2	x_3	x_4	x_5
x_0	x	x	.	.	.	.
x_1	x	x	x	.	.	.
x_2	.	x	x	x	.	.
x_3	.	.	x	x	x	.
x_4	.	.	.	x	x	x
x_5	.	.	.	.	x	x

of $\mathcal{I}$ [36] in $O(n)$ operations for n poses, similar to Markov Localization. Thus, *in the absence of short-term features, the computational complexity of EnML is similar to that of Markov Localization.*

Thus, the computational complexity of EnML scales from being comparable to Markov Localization in the absence of short-term features, to being comparable to pose graph SLAM when only short-term features are visible.

7. Results

We present three sets of experimental results to evaluate the performance of episodic non-Markov localization and compare it with alternative approaches to robot localization in varying environments. The first two sets of experiments were performed using data logs collected at the University of Freiburg, and Örebro University, respectively, and are used to perform quantitative comparisons.[2] The third set of experimental results are from the 1000 km Challenge [37], performed with four custom-built robots called CoBots [4] at Carnegie Mellon University.

7.1. Freiburg-Parkinglot dataset

We compared localization using our approach to localization using Temporary Maps [21] and Rao-Blackwellized Particle Filters with Dynamic Occupancy Grids (RBPF-D) [25] by running Episodic non-Markov Localization on the Freiburg-Parkinglot dataset collected on an outdoor robot driven around a parking lot at University of Freiburg [25]. This dataset consists of 12 runs in an outdoor parking lot over the course of the day totaling over 5.8 km traversed and presents a challenging environment where there are very few permanent features and many changes over the course of the day due to the arrival and departure of cars in the lot. For every run, a ground truth estimate was determined independently of the other runs by running static SLAM offline with manual corrections. To run episodic non-Markov localization on the Freiburg-Parkinglot dataset we extracted the dominant linear features from the results of running SLAM on run 01 of the dataset. This map was then used as the static map for all subsequent runs.

Table 5
Localization Squared Errors (m^2) for the Freiburg-Parkinglot Dataset.

Run	Length	MCL-GT	MCL-S	MCL-TM	RBPF-D	EnML
01	503.33	0.04	0.18	0.25	0.09	0.02
02	497.74	0.03	0.18	0.16	0.08	0.02
03	496.25	0.04	0.09	0.63	0.05	0.02
04	487.80	0.02	0.08	0.63	0.04	0.02
05	494.78	0.02	0.06	0.51	0.03	0.02
06	489.89	0.02	0.09	0.21	0.02	0.02
07	488.10	0.02	0.07	0.44	0.03	0.02
08	488.39	0.02	0.09	0.59	0.02	0.02
09	479.84	0.02	0.07	0.49	0.03	0.03
10	484.06	0.02	0.09	0.32	0.03	0.04
11	484.88	0.03	0.10	0.47	0.05	0.04
12	479.97	0.03	0.15	0.23	0.06	0.05
Total	5875.0	0.03	0.10	0.41	0.04	0.03

As a baseline, the ground truth maps for each of the runs were used along with Monte-Carlo Localization to localize the robot for every run. Table 5 lists the errors in localization with respect to ground truth for localization using the baseline (MCL-GT), Monte-Carlo Localization using a static map (MCL-S), Temporary Maps [21] (MCL-TM), Rao-Blackwellized Particle Filters with Dynamic Occupancy Grids [25] (RBPF-D), and Episodic non-Markov Localization (EnML). The entries for MCL-GT, MCL-S, MCL-TM and RBPF-D are reproduced from the results of [25]. Note that the baseline approach MCL-GT is intended for illustrative comparative purposes only: it is the result of performing SLAM from each run, manually correcting the errors in the map, and then using the same log for localization using MCL. Such an approach is neither realistic, nor feasible, but provides a "best-case" scenario for localization using MCL on as accurate a map as possible. Episodic non-Markov localization has smaller mean localization error over all the runs compared to the other online algorithms including MCL-S, MCL-TM and RBPF-D. Over individual runs, EnML has lower errors compared to the other algorithms except for runs 06 and 08 where EnML tied with RBPF-D; and run 10 where RBPF-D had a lower error. EnML also outperforms the Monte-Carlo Localization baseline (MCL-GT) for some of the runs, as a result of EnML using a fundamentally different belief representation and update algorithm than MCL-GT, which uses a particle filter. Furthermore, as a deterministic algorithm, our approach does not have any variance in localization over different runs with the same log, unlike the other algorithms, which are stochastic in nature as variants of particle filters, and exhibit variance across trials [25]. Fig. 7 shows the trajectory of the robot during run 11 of the Freiburg-Parkinglot dataset, as estimated by episodic non-Markov localization and compared to ground truth.

Fig. 8 shows two selected snapshots of episodic non-Markov localization running on the Freiburg-Parkinglot dataset. The snapshots show the classification of the observations as originating from LTFs, STFs and DFs. The snapshots demonstrate episodic non-Markov localization correctly identifying the parked cars as STFs and the pedestrian and moving car as DFs.

[2] The authors would like to thank Gian Diego Tipaldi and Daniel Meyer-Delius for sharing the Freiburg-Parkinglot dataset, and Tom Duckett for sharing the Örebro-Longterm dataset.

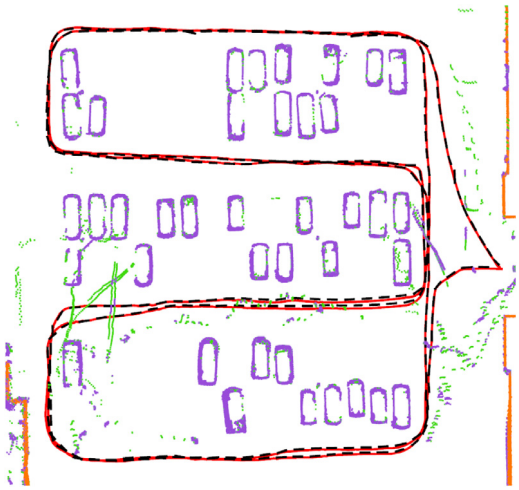

Fig. 7. The trajectory of the robot during run 11 of the Freiburg-Parkinglot dataset, as estimated by episodic non-Markov localization (red trace), and compared to ground truth (dashed black trace). The LTFs, STFs and DFs classified by episodic non-Markov localization are drawn as orange, purple, and green points, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.2. Örebro-Longterm dataset

We ran episodic non-Markov localization on the Örebro-Longterm dataset, which has previously been used to empirically evaluate localization using Dynamic Maps [23], and Dynamic Pose Graph SLAM [19]. This dataset was collected over a span of five weeks by driving a robot around an office-like environment, covering a total distance of 9.6 km.

As in the previous experiment, we estimated the static long-term map by running SLAM on the first run. Fig. 9 shows the traces of the robot as estimated using episodic non-Markov localization over all the runs. Despite the changes in the environment over the five-week period, our approach was successfully able to localize the robot without having to maintain up-to-date maps of the environment. This experiment demonstrates that even in a varying environment, using only a static map of LTFs, episodic non-Markov localization is successfully able to localize a robot without having to maintain maps of the exact state of the environment.

7.3. Deployments of the CoBots

The CoBots are custom-built robots[3] with four-wheel omni-directional drive bases and off-the-shelf tablet computers as the computational platforms to run all the algorithms necessary for them to operate autonomously. We have four CoBots in total, shown in Fig. 10, which were deployed over the 1000 km Challenge [37]. The different CoBots have slightly different sensing abilities, including a 4m laser rangefinder on CoBot 2 and CoBot 3, and 5m Microsoft Kinect depth cameras on CoBot 1 and CoBot 4. Episodic non-Markov Localization has been deployed on the CoBots over 755 km out of 1000 km in the 1000 km Challenge, and has been used to localize the robots in many different environments spanning multiple floors across multiple buildings. EnML used observations from the laser rangefinders on CoBot 2 and CoBot 3, and observations from the Kinects on CoBot 1 and CoBot 4. We present here some selected results from the 1000 km Challenge, and the impact of using EnML over the robot deployments.

[3] We thank Mike Licitra for designing and building the robots.

Table 6
Mean Distance Before Interventions (MDBI), in km, using CGR and EnML per map over the 1000 km Challenge.

	CGR	EnML
GHC4	0.62	4.42
GHC5	1.23	9.49
GHC6	8.61	9.48
GHC7	5.58	9.02
GHC8	6.04	19.36
GHC9	5.33	20.05
All	4.79	8.13

Table 7
Characteristics of EnML on the floors that the CoBots were deployed on, for the 1000 km Challenge: EP, the median number of poses in each episode; EL, the median pose length in meters; LTF%, the percentage of observations that were LTFs; STF%, the percentage of STFs; and DF%, the percentage of DFs.

	EP	EL	LTF%	STF%	DF%
GHC3	75	4.2	41.5	35.7	16.4
GHC4	80	11.2	12.0	67.7	15.6
GHC5	34	5.5	64.2	23.8	11.9
GHC6	21	3.6	69.6	22.8	6.7
GHC7	19	3.3	73.1	19.5	7.7
GHC8	23	3.4	70.5	22.0	7.0
GHC9	25	3.8	67.2	24.0	7.6
NSH1	59	3.2	56.1	35.2	8.9
NSH2	72	3.1	42.8	44.3	8.9
NSH3	80	3.4	53.2	33.8	9.8
NSH4	29	4.1	67.4	24.2	8.4
NYU19	80	8.3	44.3	44.4	11.2

Over the duration of the 1000 km Challenge, Corrective Gradient Refinement [5] was used for localization between September 2011 and January 2014 [26], while Episodic non-Markov Localization was used February 2014 onwards. As of July 1st 2015, the CoBots have autonomously traversed 755 km using Episodic non-Markov Localization, and continue to autonomously perform user-requested tasks in our environments on a daily basis. Table 6 compares the Mean Distance Before Interventions (MDBI), for each map in the Gates–Hillman Center building, when using CGR, and when using EnML for localization. The MDBI is the mean distance that the robot traversed before an operator intervention was required, and is thus a measure of the robustness of the autonomy of the robot. The MDBI for EnML is 8.13 km, which is significantly higher than the MDBI for CGR, 4.79 km, thus demonstrating the higher reliability of EnML for localization in real-world human environments.

The variations in the MDBI across the different floors are due to the differences in the number of movable objects and variations over time between the different floors. Table 7 highlights the different characteristics of each floor by enumerating, for each floor: (1) the median number of poses in each episode of EnML, (2) the median length of each pose in EnML, (3) the median fraction of observations that were LTFs, (4) the median fraction of observations that were STFs, and (5) the median fraction of observations that were DFs. Floors GHC6, GHC7, GHC8, and GHC9 were observed to have the fewest number of unmapped observations (LTFs and DFs), and hence had correspondingly shorter episodes, with median pose lengths between 3.3 m and 3.8 m. Floors NSH1, NSH2, NSH3, and NSH4 were in a building with a large number of shorter corridors with frequent intersections, resulting in episodes short in length (between 3.1 m and 4.1 m), but large number of poses (between 29 and 80 poses) due to frequent rotations. Compared to GHC6-9, floors NSH1-4 also had more DFs since they have more human traffic, and more STFs in the form of chairs, tables, and sofas in the hallways. Floor NYU19 had cubicle areas that were not included in the static map, and hence EnML discovered the cubicle walls as numerous STFs: 44.4% of the observations were of STFs,

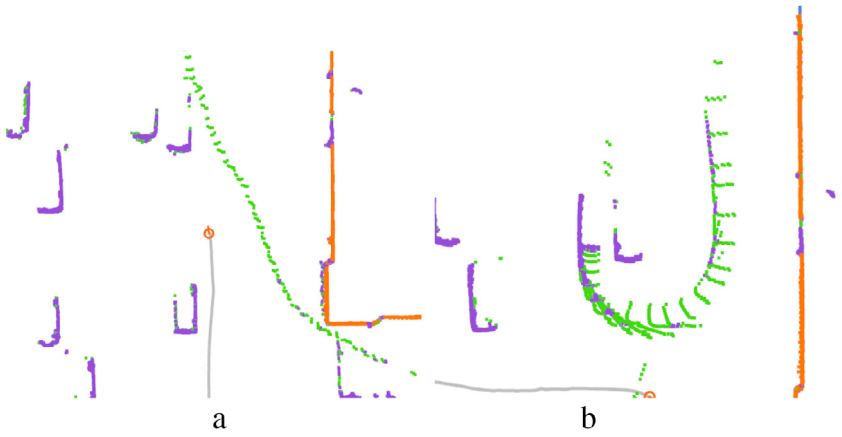

Fig. 8. Non-Markov localization on run 12 of the Freiburg-Parkinglot dataset showing (a) a pedestrian and (b) a moving car in the parking lot amidst static parked cars. LTFs are plotted in orange, STFs in purple, and DFs in green. The robot's location is shown by the orange marker and its trajectory as gray lines. Both images are 35 m wide. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

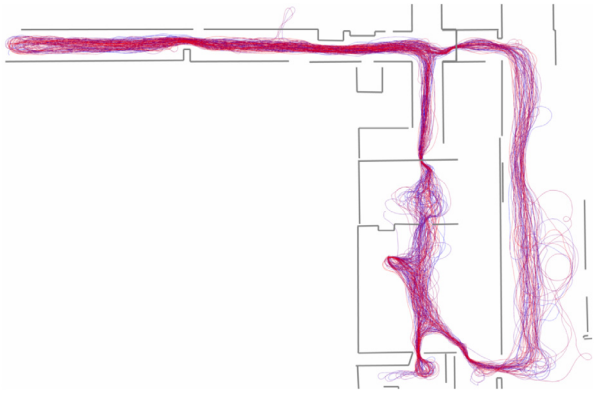

Fig. 9. Combined traces of localization using episodic non-Markov localization on all runs of the Orëbro-Longterm dataset. The traces are color-coded by time, from blue (oldest) to red (newest). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 10. CoBots 1, 2, 3 and 4, which were used for the 1000 km Challenge at Carnegie Mellon University.

resulting in long episodes. Floor GHC4 was the most challenging of all floors, with only 12% of the observations corresponding to LTFs, resulting in very long episodes. Floors GHC4 and GHC5 are particularly challenging for autonomous robots deployed over long periods of time. These floors have significant pedestrian traffic, being on the ground level and connecting different parts of the campus. GHC4 has an atrium with several unusual large movable lounge chairs which obstruct the robot's view of the walls. GHC5 has large areas with tiles which provide for very poor robot odometry due to wheel slippage. Fig. 11 shows some snapshots of EnML running on the CoBots while deployed on GHC4, at different times of the year. Note that the STFs are at different positions over the different deployments. Fig. 12 shows two examples of EnML accounting for large unmapped features, a wooden panel of lockers, and the walls of the helix in the Gates–Hillman Center.

Due to these challenges, prior to the introduction of EnML, the deployments of the CoBots in GHC4 and GHC5 were severely limited—the robots frequently required operator intervention as they got lost or uncertain of their localization. The low values of MDBI for CGR on GHC4 and GHC5, 0.62 km and 1.23 km respectively (tabulated in Table 6) attest to the frequent operator interventions on the floors. However, since January 2014 when EnML was deployed on the robots, they have been deployed with far greater success, as is shown by the significantly higher values of MDBI of 4.42 km and 9.49 km on GHC4 and GHC5, respectively. Table 8 shows the marked difference in the scope of deployments

Table 8
The contribution of EnML to the deployment of the CoBots on two challenging floors, GHC4 and GHC5. Over 93% of the distance traversed by the CoBots on GHC4 and GHC5 were with EnML, while only 70.2% of the distance traversed over the entire 1000 km Challenge over all floors were with EnML.

Floor	CGR (km)	EnML (km)	EnML fraction (%)
GHC4	6.4	88.1	93.2
GHC5	3.0	42.9	93.5
All	320.9	755.4	70.2

of the CoBots in these challenging areas using Corrective Gradient Refinement [5] and after deploying EnML. While EnML contributed to 70.2% of the total autonomous distance traversed by the CoBots over the 1000 km Challenge, on the GHC4 and GHC5 floors, EnML contributed over 93% of the autonomous distance traversed. The disproportionately large contributions of EnML to the distances traversed by the CoBots on GHC4 and GHC5 are entirely due to the increased reliability and robustness of localization of the robots in the presence of the abundant moving and movable objects on those floors.

8. Conclusion

In this article, we introduced a representation of the world that classifies observations as arising from long-term features, short-term features, or dynamic features. We introduced a graphical model called the Varying Graphical Network (VGN) to represent

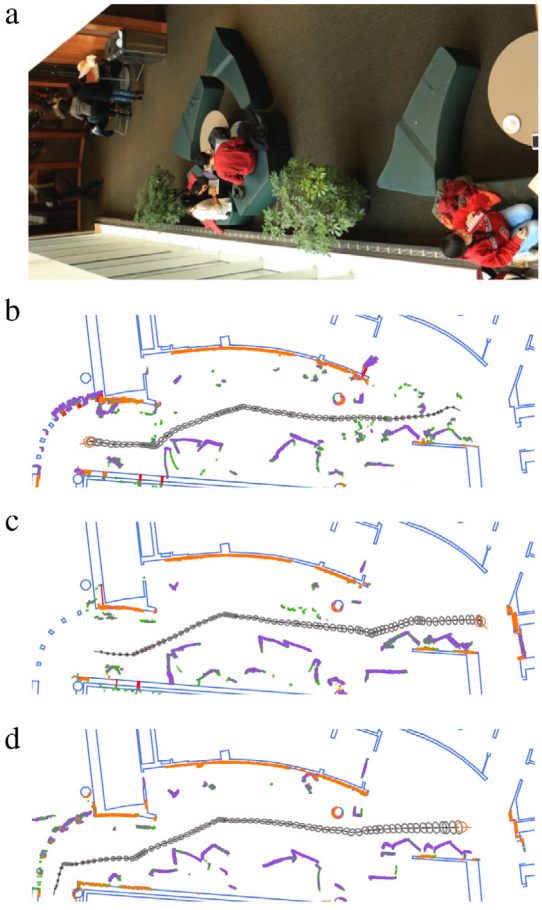

Fig. 11. Snapshots of EnML running in the GHC4 atrium on different days. The photograph (a) shows the atrium with large movable couches tables, plants, and humans. Snapshots (b–d) of Episodic non-Markov Localization show the observed state on different days. The trajectory of the robot over the episode is shown in gray, along with the covariance ellipses. LTF observations are shown as orange points, STF observations as purple points, and DF observations as green points. The long-term static map is shown as blue lines. Note the different placements of the STFs in the different runs, in particular the couches at the lower end. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

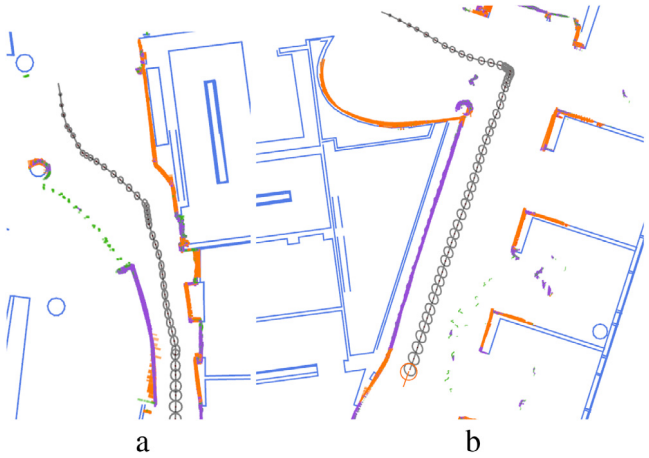

Fig. 12. Snapshots of EnML running on GHC5, showing unmapped features that are (a) the walls of the helix ramp, and (b) a wooden panel of lockers. LTF observations are shown as orange points, STF observations as purple points, and DF observations as green points. The long-term static map is shown as blue lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

correlations between observations from different time steps due to the different classes of observations. We further introduced the Episodic non-Markov Localization (EnML) algorithm that explicitly reasons about the different classes of observations to maintain accurate location estimates of the robot even in the presence of observations of unexpected objects and the absence of observations of the static map. We explored the structure of EnML, and thus compared its computational complexity to SLAM and Markov Localization. Finally, we presented results of running EnML on several datasets and comparing its results to alternate approaches.

Acknowledgments

This research was supported by NSF award IIS-1012733 and ONR grant number N00014-09-1-1031. The views and conclusions contained in this document are those of the authors only. We thank the members of the CORAL group, in particular Brian Coltin, Stephanie Rosenthal, and Richard Wang, for the underlying robot task scheduling, human-robot interaction planning, and data-collection deployments. We also thank Mike Licitra for the design and construction of the CoBot robots.

References

[1] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte carlo localization for mobile robots, in: ICRA, vol. 2, 1999, pp. 1322–1328.
[2] S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation, Proc. IEEE 92 (3) (2004) 401–422.
[3] S. Koenig, R. Simmons, Xavier: A robot navigation architecture based on partially observable markov decision process models, in: Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems, 1998, pp. 91–122.
[4] S. Rosenthal, J. Biswas, M. Veloso, An effective personal mobile robot agent through symbiotic human-robot interaction, in: AAMAS 2010, 2010, pp. 915–922.
[5] J. Biswas, B. Coltin, M. Veloso, Corrective Gradient Refinement for mobile robot localization, in: Intelligent Robots and Systems, IROS, 2011 IEEE/RSJ International Conference on, vol. 1, 2011, pp. 73–78.
[6] D. Fox, Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation, Dept. of Computer Science, University of Bonn, Germany, 1998.
[7] D. Fox, S. Thrun, W. Burgard, F. Dellaert, Particle filters for mobile robot localization, in: Sequential Monte Carlo methods in practice, 2001, pp. 499–516.
[8] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.
[9] J.J. Leonard, H.F. Durrant-Whyte, I.J. Cox, Dynamic map building for an autonomous mobile robot, Int. J. Robot. Res. 11 (4) (1992) 286–298.
[10] D. Fox, KLD-sampling: Adaptive particle filters and mobile robot localization, Adv. Neural Inf. Process. Syst. (2001).
[11] S. Lenser, M. Veloso, Sensor resetting localization for poorly modelled mobile robots, in: Int. Conf. on Robotics and Automation, ISBN: 0780358864, 2000.
[12] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): Part II, IEEE Robot. Autom. Mag. 13 (3) (2006) 108–117.
[13] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEE Robot. Autom. Mag. 13 (2) (2006) 99–110.
[14] S. Thrun, M. Montemerlo, The graph SLAM algorithm with applications to large-scale mapping of urban structures, Int. J. Robot. Res. 25 (5–6) (2006) 403–429.
[15] F. Dellaert, M. Kaess, Square root sam: Simultaneous localization and mapping via square root information smoothing, Int. J. Robot. Res. 25 (12) (2006) 1181–1203.
[16] E.B. Olson, Real-time correlative scan matching, in: ICRA, 2009, pp. 4387–4393.
[17] F. Lu, E. Milios, Robot pose estimation in unknown environments by matching 2d range scans, J. Intell. Robot. Syst. 18 (3) (1997) 249–275.
[18] E. Olson, J. Leonard, S. Teller, Fast iterative alignment of pose graphs with poor initial estimates, in: ICRA, 2006, pp. 2262–2269.
[19] A. Walcott-Bryant, M. Kaess, H. Johannsson, J.J. Leonard, Dynamic pose graph SLAM: Long-term mapping in low dynamic environments, in: IROS, 2012, pp. 1871–1878.
[20] C. Stachniss, W. Burgard, Mobile robot mapping and localization in non-static environments, in: AAAI, 2005, pp. 1324–1329.
[21] D. Meyer-Delius, J. Hess, G. Grisetti, W. Burgard, Temporary maps for robust localization in semi-static environments, in: IROS, 2010, pp. 5750–5755.

[22] T. Morris, F. Dayoub, P. Corke, G. Wyeth, B. Upcroft, Multiple map hypotheses for planning and navigating in non-stationary environments, in: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, 2014, pp. 2765–2770.
[23] P. Biber, T. Duckett, Dynamic maps for long-term operation of mobile service robots, in: Robotics: Science and Systems, 2005, pp. 17–24.
[24] J. Saarinen, H. Andreasson, A.J. Lilienthal, Independent Markov chain occupancy grid maps for representation of dynamic environment, in: IROS, 2012, pp. 3489–3495.
[25] G.D. Tipaldi, D. Meyer-Delius, M. Beinhofer, W. Burgard, Lifelong localization and dynamic map estimation in changing environments, in: RSS Workshop on Robots in Clutter, 2012.
[26] J. Biswas, M. Veloso, Localization and navigation of the CoBots over long-term deployments, Int. J. Robot. Res. 32 (14) (2013) 1679–1694.
[27] J. Biswas, M. Veloso, Depth camera based indoor mobile robot localization and navigation, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 1697–1702.
[28] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517.
[29] J.J. Barton, L.R. Nackman, Scientific and Engineering C++: an introduction with advanced techniques and examples, Addison-Wesley Longman Publishing Co., 1994.
[30] S. Agarwal, K. Mierle, Ceres Solver: Tutorial & Reference, 2012.
[31] A. Griewank, A. Walther, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Society for Industrial and Applied Mathematics (SIAM), 2008.
[32] F. Lu, E. Milios, Globally consistent range scan alignment for environment mapping, Auton. Robots 4 (4) (1997) 333–349.
[33] D.W. Marquardt, A method for the solution of certain nonlinear problems in least squares, Quart. Appl. Math 2 (2) (1944) 164–168.
[34] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math. 11 (2) (1963) 431–441.
[35] Y. Chen, T.A. Davis, W.W. Hager, S. Rajamanickam, Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate, ACM Trans. Math. Softw. 35 (3) (2008) 22.
[36] J. Du Croz, P. Mayes, G. Radicati, Factorizations of Band Matrices Using Level 3 BLAS, Springer, 1990.
[37] J. Biswas, Vector Map-Based, Non-Markov Localization for Long-Term Deployment of Autonomous Mobile Robots, The Robotics Institute, Carnegie Mellon University, USA, 2014.

Joydeep Biswas is an Assistant Professor in the College of Information and Computer Sciences at University of Massachusetts Amherst. Prior to joining UMass Amherst, Joydeep was a post-doctoral fellow in the Computer Science Department at Carnegie Mellon University, and before that he earned his Ph.D. in Robotics from the Robotics Institute at Carnegie Mellon University. As a Ph.D. student he was the recipient of the 2015 Siebel Scholarship. See www.cs.umass.edu/~joydeepb/ for further details related to his research on autonomous service mobile robots.

Manuela M. Veloso is the Herbert A. Simon University Professor in the Computer Science Department at Carnegie Mellon University. She is the Incoming Head of the Machine Learning Department, and she has courtesy appointments in the Robotics Institute and Electrical and Computer Engineering Department. She researches in Artificial Intelligence, Robotics, and Machine Learning. She founded and directs the CORAL research laboratory, for the study of autonomous agents that Collaborate, Observe, Reason, Act, and Learn, www.cs.cmu.edu/~coral. Professor Veloso is IEEE Fellow, AAAS Fellow, AAAI Fellow, and the past President of AAAI and RoboCup. Professor Veloso and her students research with a variety of autonomous robots, including mobile service robots and soccer robots. See www.cs.cmu.edu/~mmv for further information, including publications.