Episodic Non-Markov Localization: Reasoning About Short-Term and Long-Term Features

Joydeep Biswas[1] and Manuela Veloso[2]

***Abstract*— Markov localization and its variants are widely used for localization of mobile robots. These methods assume Markov independence of observations, implying that observations made by a robot correspond to a static map. However, in real human environments, observations include occlusions due to unmapped objects like chairs and tables, and dynamic objects like humans. We introduce an episodic non-Markov localization algorithm that maintains estimates of the belief over the trajectory of the robot while explicitly reasoning about observations and their correlations arising from unmapped static objects, moving objects, as well as objects from the static map. Observations are classified as arising from long-term features, short-term features, or dynamic features, which correspond to mapped objects, unmapped static objects, and unmapped dynamic objects respectively. By detecting time steps along the robot's trajectory where unmapped observations prior to such time steps are unrelated to those afterwards, non-Markov localization limits the history of observations and pose estimates to "episodes" over which the belief is computed. We demonstrate non-Markov localization in challenging real world indoor and outdoor environments over multiple datasets, comparing it with alternative state-of-the-art approaches, showing it to be robust as well as accurate.**

I. Introduction

Human environments, particularly indoor ones, have elements of permanence - the architectural structure of a building is unlikely to change over time. However, such environments also include movable and moving objects. Movable objects like furniture and doors appear static, but are likely to be moved around frequently.

Mobile robots in human environments commonly use some variant of Markov localization [1] with a static map, and localize in such environments by ignoring observations that are hypothesized by means of various filters [1] to be of unmapped objects. In environments with frequent local changes like cafe seating areas, a large portion of the observations will be of unmapped movable objects that occlude features from the static map. The robot could use these observations to refine its localization estimate and lower its uncertainty, even if the objects are not part of the robot's static map.

Recognizing the importance of having a robot adapt to changes in the environment while localizing itself, a number of approaches have been proposed to either maintain up-to-date maps of the environment [2] or to model the dynamics of the environment [3], [4], [5], [6]. However, to maintain an up-to-date estimate or an accurate dynamic model of large environments, a robot would have to frequently visit every area of the environment periodically, which might not be feasible in a large deployment environment.

In this paper, we introduce a localization algorithm that explicitly reasons about observations of non-mapped objects without having to maintain up-to-date estimates of the environment or even having to estimate the dynamics of the environment. We call this non-Markov localization because it relaxes the Markov assumption that observations are independent given the map, which is the central assumption of Markov localization. We represent the non-Markov nature of the problem in a graphical representation called "Varying Graphical Networks" (VGNs). Episodic non-Markov localization maintains a belief of the history of pose estimates of the robot over "episodes" of observations of unmapped objects. For every timestep, it classifies observations into those arising from Long Term Features (LTFs), Short Term Features (STFs), or from Dynamic Features (DFs), corresponding to mapped permanent features, unmapped static objects, and unmapped moving objects, respectively. Observations made from LTFs are matched to a static map, while observations from STFs are matched to observations from STFs at different time steps. The belief is framed in terms of a cost function over odometry observations, observations of LTFs, and correlations between observations of STFs from different time steps. The maximum likelihood estimate of the belief is incrementally computed over successive steps as a non-linear functional optimization over the cost function.

We present results from running non-Markov localization on logs collected from our robot deployed in an indoor office environment, as well as on two other standard datasets. We compare the results from the standard datasets to three alternative approaches to robot localization in varying environments, including Dynamic Maps [3], Temporary Maps [4], and Rao-Blackwellized Particle Filters with Dynamic Occupancy Grid Maps [7]. We demonstrate that episodic non-Markov localization is more accurate than alternative approaches, localizes the robot reliably in varying environments without having to remap the environment over time.

II. Background and Related Work

Suppose a robot makes observations of the environment, $s_{1:n}$ and accumulates robot odometry $u_{1:n}$ over n time-steps $t_{1:n}$. We follow the convention that at time t_i, observation s_i is made, and odometry u_i indicates relative motion of the robot from timestep t_{i-1} to t_i. Given a prior map of the

[1]Joydeep Biswas is with the Robotics Institute, and [2]Manuela Veloso with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA {joydeepb, mmv}@cs.cmu.edu

This research was partially supported by the National Science Foundation award number NSF IIS-1012733, and by DARPA award number FA8750-12-2-0291. The views in this document are those of the authors only.

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,

$$Bel(x_n) = P(x_n|s_{1:n}, u_{1:n}, x_{0:n-1}, M).$$

Markov Localization [1] 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 $x_{0:n-2}$ and odometry $u_{1:n-1}$.
2) **Markov independence of observations**: Given the estimate of robot pose x_n, observation s_n is independent of past pose estimates $x_{0:n-1}$, observations $s_{1:n-1}$ and odometry $u_{1:n}$.

Using the Markov assumptions and applying Bayes rule, the recursive update of the belief simplifies to (see [8] for a complete derivation),

$$Bel(x_n) = \eta P(s_n|x_n, M) \int P(x_n|x_{n-1}, u_n) Bel(x_{n-1}) dx_{n-1},$$

where η is a normalizing constant.

In reality, neither the odometry nor the observations of a robot deployed in a real environment are truly Markovian in nature. In a human environment, 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 are penalized at every step when the static objects are observed. Alternatively, if unexpected observations are explicitly removed from the computation of the belief [1], then in areas with limited visibility of map features, the uncertainty in the belief will grow over time due to the absence of sensory feedback.

In recognition of these limitations, there has been some work on localization and long-term mapping in dynamic environments. The approach of "Dynamic Maps" [3] maintains a number of local maps over a number of time scales, and during the localization step selects those local maps that best agree with the observations. Dynamic Pose Graph SLAM [2] maintains the latest estimate of the map of the environment by saving a history of observations and selectively replacing older observations with newer ones when they are contradictory. Mapping with independent Markov chain occupancy grids[6] models a dynamic environment as an occupancy grid with associated Markov chains with every cell on the grid to model the transition probabilities of transitioning from unoccupied to occupied, and vice-versa. Rao-Blackwellized Particle Filters with dynamic occupancy grids [7] model a varying environment as an occupancy grid with an associated Hidden Markov Model for every cell to model its transitions between unoccupied and occupied.

As an alternative to estimating the complete state of a varying environment, there have been a number of proposed approaches to model each movable object separately in the environment. The approach of hierarchical object maps [9] assumes certain classes of shapes of objects, and matches unmapped objects to them. Patch maps [5] maintains multiple maps of the different possible states of the environment, and as the robot localizes in the environment it reasons about the probability of each local patch map as best representing the state of the environment. For the specific problem of estimating the state of doors in the environment, one proposed approach [10] relies on a hybrid vector - occupancy grid map with the occupancy grid representing the static map and the vector map representing the dynamic door states.

The approach of "Temporary Maps" [4] is similar to our approach in that it considers environments where observations differ from a static map. Temporary Maps model the effect of STFs by performing local SLAM, using the LTFs as an initial estimate for the local map. Using these locally static maps, the robot then localizes using a particle filter. In contrast, our approach partitions the observations into those caused by STFs and LTFs, and finds correlations between STFs from different time steps. By considering LTF observations separately from STF observations at every step, our approach provides global location corrections based on the LTF observations and the long-term map while simultaneously benefiting from the local corrections based on the STF observations.

III. Non-Markov Localization

In a human environment, observations are dependent not only on the static map (LTFs) but also on unmapped static objects (STFs) and moving objects (DFs). The STFs and DFs introduce additional correlations between observations from different timesteps. The classifications of observations into LTFs, STFs and DFs, and the resulting correlations need to be updated every time the belief is updated.

To represent the varying nature of this problem, we introduce a new graphical model, the "Varying Graphical Network" (VGN). As in a dynamic Bayesian network, a 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 nodes. The presence of STFs and DFs is encoded by the presence of associated varying nodes. The correlations between STFs observed at different timesteps

is encoded by the varying edges. Fig. 1 shows an example instance of a VGN for non-Markov localization.

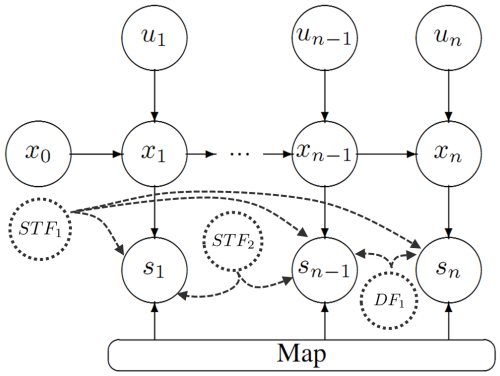


Fig. 1. An example instance of a Varying Graphical Network (VGN) for non-Markov localization. The non-varying nodes and edges are denoted with solid lines, and the varying nodes and edges with dashed lines. Due to the presence of short term features (STFs) and dynamic features (DFs), the structure is no longer periodic in nature. The exact structure of the graph will depend on the STFs and DFs present.

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.

In order to explicitly account for the effect of the past robot pose estimates $x_{0:n}$ on future observations and pose estimates, we solve for the belief over the complete history of robot poses, $Bel(x_{0:n})$. The dimension of the state space of the belief over the complete history of robot poses $x_{0:n}$ is $d(n+1)$, 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 $x^*_{0:n}$ such that

$$x^*_{0:n} = \arg\max_{x_{0:n}} (Bel(x_{0:n})).$$

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.

A. 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:

$$P(x_{0:n}|s_{1:n}, u_{1:n}, M) =$$
$$P(x_{0:i}|s_{1:i}, u_{1:i}, M) \times P(x_{i+1:n}|s_{i+1:n}, u_{i+1:n}, x_i, M).$$

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 assume such episode-boundary timesteps like t_i exist, allowing real-time non-Markov localization with limited computational resources. Episode-boundary timesteps frequently occur in practice when a robot either does not observe any STFs for one or more timesteps, 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.

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

For the rest of this paper, to simplify notation, we will refer to observations of LTFs, STFs, and DFs directly as LTFs, STFs, and DFs, respectively, with the understanding that henceforth they refer to the *observations*, not the actual *objects* that cause them.

B. Classification of Observations

For every timestep, the structure of the VGN, based on the classification of the observations into LTFs, STFs and DFs, is re-evaluated prior to updating the MLE of the belief. In this work the sensor we use is a laser rangefinder, so 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. We represent the pose x_i of the robot on the map at timestep i as an affine transform T_i that consists of a 2D rotation followed by a 2D translation. Thus, for every 2D point p^i_j observed by the robot in its own reference frame, the corresponding location of the point in the global reference frame of the map is given by $T_i p^i_j$. Each point p^i_j has to be classified as an LTF or as an STF or as a DF, and depending on its classification, results in addition of varying edges to the VGN for non-Markov localization.

1) Classification of LTFs: We use a vector map [11] representation $M = \{l_k\}_{k=1:m}$ for the permanent map, consisting of a set of m line segments l_k. To evaluate which of the observed points p^i_k are LTFs, an analytic ray cast [11] is performed from the latest MLE of x_i. The result of the analytic ray cast is a mapping from $p^i_j \to l_j \in M$, indicating that the line segment l_j from map M is the most line in the map to be observed by the point p^i_j. Let $\mathrm{dist}(p, l)$ 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 $P(p^i_j|T_i, M)$ of the point p^i_j is then given by

$$P(p^i_j|x_i, M) = \exp\left(-\frac{\mathrm{dist}(T_i p^i_j, l_j)^2}{\Sigma_s}\right),$$

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, $P(p_j^i|x_i, M) > \epsilon_{\text{LTF}}$. Observed points p_j^i 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

$$\text{LTF}_i = \left\{p_j^i \in s_i | P(p_j^i|x_i, M) > \epsilon_{LTF}\right\},$$
$$\overline{\text{LTF}}_i = s_i \setminus \text{LTF}_i.$$

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

$$P(p_j^i, p_k^l|x_i, x_l) = \exp\left(-\frac{||T_i p_j^i - T_l p_k^l||^2}{\Sigma_s}\right)$$

where Σ_s is the scalar variance of observations. Therefore, a non-LTF point $p_j^i \in \overline{\text{LTF}}_i$ is classified as an STF if there exists a point $p_k^l \in \overline{\text{LTF}}_l$ from a timestep $l, l < i$ such that $P(p_j^i, p_k^l|x_i, x_l) > \epsilon_{\text{STF}}$:

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

To speed up the correspondence check, points $p_j^i \in s_i$ from every observation s_i are stored in KD-trees [12]. The STF observation likelihood function for a pair of matching STF observations p_j^i and p_k^l introduces a correlation between observations s_i and s_l since $p_j^i \in s_i$ and $p_k^l \in s_l$. For every such pair of matching STF observations, the VGN of non-Markov localization thus includes a varying node representing the unmapped static object observed as STFs at different timesteps, and a pair of varying edges joining the varying node and each of the observations s_i and s_l.

3) Classification of DFs: Observations that are classified as neither LTFs nor STFs correspond to objects that were observed at one particular location for only one particular timestep, and were not observed at any other timestep 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, so the correlations between observations due to DFs are not used to further refine the belief.

IV. Cost Function Representation of Belief

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

$$\begin{aligned} Bel(x_{0:n}) &= P(x_{0:n}|s_{1:n}, u_{1:n}, M) \\ &= \exp(-C(x_{0:n}|s_{1:n}, u_{1:n}, M)). \end{aligned}$$

The computation of a single cost function over all states and observations is still intractable, so we split up the cost function C into multiple "sub-cost functions" $C = \sum_{i=0}^{i=n} c_i$ where each sub-cost function c_i depends only on a subset d_i of the set of states and observations, $d_i \subseteq \{x_{0:i}, s_{1:i}, u_{1:i}, M\}$. The subset d_i includes observations (and the corresponding pose estimates) from previous time steps that are estimated to be related to s_i due to STF observations. d_i also includes s_i, x_i and the map, required for computation of the cost due to LTF observations LTF_i. The form of the total cost function in terms of each sub-cost function c_i and their data subsets d_i is given by, $C(x_{0:n}) = \sum_{i=0}^{i=n} c_i(x_i|d_i)$. This expansion is equivalent to the chain rule expansion of the expression of the belief, where each sub-cost function c_i corresponds to a conditionally independent factor $P(x_i|d_i)$ in the belief. Using this expansion, the MLE can be estimated by non-linear optimization of the cost function:

$$x_{0:n}^* = \arg\max_{x_{0:n}}(Bel(x_{0:n})) = \arg\min_{x_{0:n}}\left[\sum_{i=0}^{i=n} c_i(x_i|d_i)\right]$$

With this formalization, we now have a framework for the non-Markov localization algorithm, outlined by Algorithm 1. When new data becomes available at t_n, for each timestep $i \in [1, n]$, set LTF_i is found by comparing observations to the map M. Set STF_i is computed from the remaining observed points by finding matching points R_i from prior observations S_i, transformed by their respective poses X_i' in X_i. The data subset d_i is thus expressed as the union of the observations and pose estimates from timestep i, and observations and pose estimates from prior timesteps that are matched to STF_i. The sub-cost function c_i is computed based on d_i and added to C. Non-linear optimization is run over cost function C to yield the latest MLE, until convergence.

Algorithm 1 Non-Markov Localization

procedure NonMarkovLocalization(X, S, U, M)
  while X not converged **do**
    $C \leftarrow 0$
    for $i = 1$ to n **do**
      $X_i \leftarrow \{x_{0:i} \subset X\}$
      $S_i \leftarrow \{s_{1:i} \subset S\}$
      Compute $\text{LTF}_i \subseteq s_i$ given x_i, M
      Compute $\text{STF}_i \subseteq \overline{\text{LTF}}_i, R_i \subseteq S_i$ and $X_i' \subseteq X_i$
      $d_i \leftarrow \text{LTF}_i \bigcup \text{STF}_i \bigcup R_i \bigcup X_i' \bigcup \{x_{i-1}, x_i, u_i\}$
      $C \leftarrow C + c_i(d_i)$
    end for
    $X \leftarrow \arg\min_X [C]$
  end while
  return X
end procedure

The sub-cost functions c_i are computed and stored as *function objects* [13], 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 [14]. Since the optimizer performs algorithmic differentiation [15] of the function objects, it provides the same accuracy as symbolic optimization, with much lower computational requirements.

The computation of the sub-cost function c_i is based on sub-cost functions due to odometry c_i^o as well as perceptual observations, c_i^s such that $c_i = c_i^o + c_i^s$. The odometry sub-cost function c_i^o is given by

$$c_i^o = (\mathbf{x}_i - \mathbf{x}_{i-1} - \mathbf{u}_i)^T \Sigma_o^{-1} (\mathbf{x}_i - \mathbf{x}_{i-1} - \mathbf{u}_i)$$

where $\mathbf{x}_i$ denotes the vector representation of pose x_i, and Σ_o is the covariance on odometry based on the motion model of the robot.

The sub-cost function based on perceptual observations, c_i^s depends on the observation s_i made at timestep t_i, and its relations to the LTFs from map M as well as relations to previous observations $s_{1:i-1}$ based on STF observations. The sub-cost function c_i^j for an observation point p_j^i that is classified as an LTF is given by taking the negative of the logarithm of the LTF observation likelihood function:

$$c_i^j = -\log(P(p_j^i | x_i, M)) = \frac{\mathrm{dist}(T_i p_j^i, l_j)^2}{\Sigma_s}.$$

Similarly, the sub-cost function c_i^j for an observation point p_j^i that is classified as an STF, with an associated STF observation point p_k^l from a previous timestep l is given by taking the negative of the logarithm of the STF observation likelihood function:

$$c_i^j = -\log(P(p_j^i, p_k^l | x_i, x_l)) = \frac{||T_i p_j^i - T_l p_k^l||^2}{\Sigma_s}$$

Once the cost sub-functions $C = \sum_{i=1}^{i=n} c_i$ have been computed, the pose estimates X are optimized by non-linear optimization of C to compute the MLE of the belief.

V. Results

We present two sets of experiments to evaluate the performance of episodic non-Markov localization and compare it with alternative approaches to robot localization in varying environments.

A. Comparison With Temporary Maps and RBPFs with Dynamic Occupancy Grid Maps

We compared localization using our approach to localization using Temporary Maps [4] and RBPFs with dynamic occupancy grids [7] 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 [7]. 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.

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.

Run	Length	MCL-GT	MCL-S	MCL-TM	RBPF	EnML
01	503.33	0.04	0.18	0.25	0.09	0.015
02	497.74	0.03	0.18	0.16	0.08	0.018
03	496.25	0.04	0.09	0.63	0.05	0.016
04	487.80	0.02	0.08	0.63	0.04	0.018
05	494.78	0.02	0.06	0.51	0.03	0.024
06	489.89	0.02	0.09	0.21	0.02	0.024
07	488.10	0.02	0.07	0.44	0.03	0.023
08	488.39	0.02	0.09	0.59	0.02	0.022
09	479.84	0.02	0.07	0.49	0.03	0.026
10	484.06	0.02	0.09	0.32	0.03	0.036
11	484.88	0.03	0.10	0.47	0.05	0.037
12	479.97	0.03	0.15	0.23	0.06	0.048
Total	5875.0	0.03	0.10	0.41	0.04	0.025

TABLE I
Localization Squared Errors (m^2) for the Freiburg-Parkinglot Dataset

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 I 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 [4] (MCL-TM), RBPFs with dynamic occupancy grids [7] (RBPF), and episodic non-Markov localization (EnML). The entries for MCL-GT, MCL-S, MCL-TM and RBPF are reproduced from the results of [7]. Episodic non-Markov localization has smaller localization errors for every run compared to the other online algorithm including MCL-S, MCL-TM and RBPF and even outperforms the Monte-Carlo Localization baseline (MCL-GT) for some of the runs. Furthermore, as a deterministic algorithm, our approach does not have any variance in localization over different runs with the same log, unlike the stochastic algorithms(MCL-S, MCL-TM, RBPF) which exhibit variance across trials [7]. Fig. 2 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. 3 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.

B. Comparison With Dynamic Maps

We ran episodic non-Markov localization on the Örebro-Longterm dataset, which has previously been used to em-

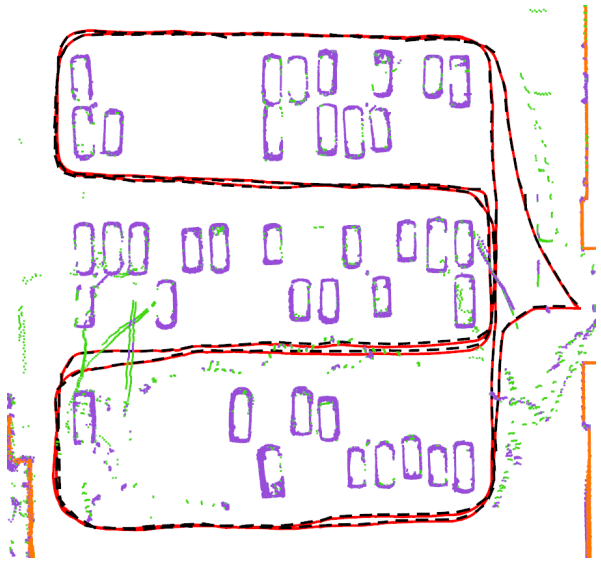
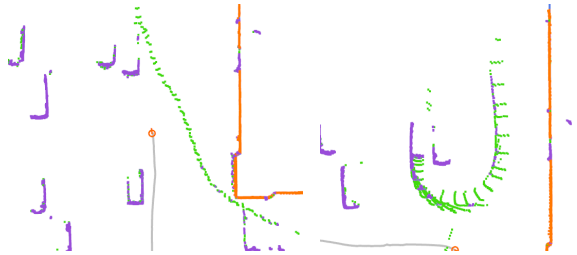


Fig. 2. The trajectory of the robot during run 11 of the Freiburg-Parkinglot dataset, as estimated by episodic non-Markov localization (red trace), and compared to ground truth (dashed black trace). The LTFs, STFs and DFs classified by episodic non-Markov localization are drawn as orange, purple, and green points, respectively.



(a) (b)

Fig. 3. 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 grey lines. Both images are $35m$ wide.

pirically evaluate localization using Dynamic Maps [3], and Dynamic Pose Graph SLAM [2]. 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. 4 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.

VI. Conclusion

We introduced episodic non-Markov localization, that classifies observations into long-term, short-term and dynamic features. The short-term features provide local corrections to the belief, while the long-term features provide global corrections. We presented experimental results from running episodic non-Markov localization, including comparisons on two standard datasets demonstrating its accuracy and robustness.

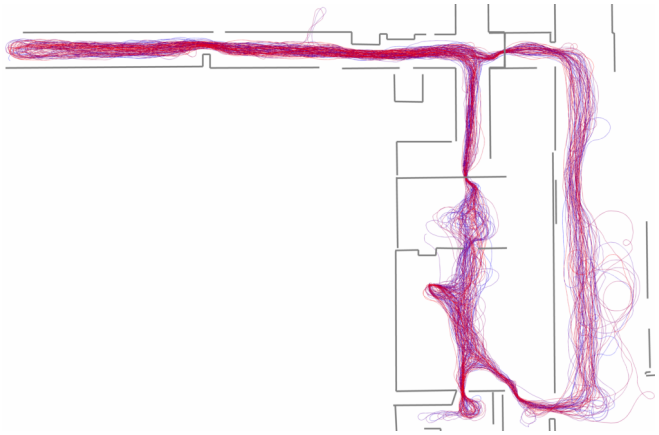


Fig. 4. Combined traces of localization using episodic non-Markov localization on all runs of the Orëbro-Longterm dataset. The traces are color-coded by time, from blue (oldest) to red (newest).

References

[1] D. Fox. *Markov localization: A probabilistic framework for mobile robot localization and navigation*. PhD thesis, Dept. of Computer Science, University of Bonn, Germany, 1998.

[2] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J.J. Leonard. Dynamic pose graph slam: Long-term mapping in low dynamic environments. In *IROS*, pages 1871 –1878, 2012.

[3] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *RSS 2005*, pages 17–24.

[4] D. Meyer-Delius, J. Hess, G. Grisetti, and W. Burgard. Temporary maps for robust localization in semi-static environments. In *IROS 2012*, pages 5750–5755.

[5] C. Stachniss and W. Burgard. Mobile robot mapping and localization in non-static environments. In *AAAI 2005*, pages 1324–1329.

[6] J. Saarinen, H. Andreasson, and A.J. Lilienthal. Independent markov chain occupancy grid maps for representation of dynamic environment. In *IROS 2012*, pages 3489 –3495.

[7] G. D. Tipaldi, D. Meyer-Delius, M. Beinhofer, and W. Burgard. Lifelong localization and dynamic map estimation in changing environments. In *RSS Workshop on Robots in Clutter*, 2012.

[8] D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. *Sequential Monte Carlo methods in practice*, pages 499–516, 2001.

[9] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *UAI 2002*, pages 10–17.

[10] A. Petrovskaya and A. Y. Ng. Probabilistic mobile manipulation in dynamic environments, with application to opening doors. In *IJCAI 2007*, pages 2178–2184.

[11] J. Biswas, B. Coltin, and M. Veloso. Corrective gradient refinement for mobile robot localization. In *IROS 2011*, pages 73–78.

[12] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[13] J.J. Barton and L.R. Nackman. *Scientific and Engineering C++: an introduction with advanced techniques and examples*. Addison-Wesley Longman Publishing Co.

[14] S. Agarwal and K. Mierle. *Ceres Solver: Tutorial & Reference*. Google Inc.

[15] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.