Autonomous Robots manuscript No.
(will be inserted by the editor)

Topological local-metric framework for mobile robots navigation: A long term perspective

Li Tang[1] · Yue Wang[1,2,*] · Xiaqing Ding[1] · Huan Yin[1] · Rong Xiong[1] · Shoudong Huang[3]

the date of receipt and acceptance should be inserted later

Abstract Long term mapping and localization are the primary components for mobile robots in real world application deployment, of which the crucial challenge is the robustness and stability. In this paper, we introduce a topological local-metric framework (TLF), aiming at dealing with environmental changes, erroneous measurements and achieving constant complexity. TLF organizes the sensor data collected by the robot in a topological graph, of which the geometry is only encoded in the edge, i.e. the relative poses between adjacent nodes, relaxing the global consistency to local consistency. Therefore the TLF is more robust to unavoidable erroneous measurements from sensor information matching since the error is constrained in the local. Based on TLF, as there is no global coordinate, we further propose the localization and navigation algorithms by switching across multiple local metric coordinates. Besides, a lifelong memorizing mechanism is presented to memorize the environmental changes in the TLF with constant complexity, as no global optimization is required. In experiments, the framework and algorithms are evaluated on 21-session data collected by stereo cameras, which are sensitive to illumination, and compared with the State-of-art global consistent framework. The results demonstrate that TLF can achieve similar localization accuracy with that from global consistent framework, but brings higher robustness with lower cost. The localization performance can also be improved from sessions because of the memorizing mechanism. Finally, equipped with TLF, the robot navigates itself in a 1km session autonomously.

Keywords Mobile robot · localization · navigation · lifelong learning

Li Tang
E-mail: litang.cv@gmail.com

Yue Wang
E-mail: wangyue@iipc.zju.edu.cn

Xiaqing Ding
E-mail: dingxiaq@gmail.com

Huan Yin
E-mail: zjuyinhuan@gmail.com

Rong Xiong
E-mail: rxiong@zju.edu.cn

Shoudong Huang
E-mail: shoudong.huang@uts.edu.au

[1]State Key Laboratory of Industrial Control and Technology, and Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, P.R. China
[2]iPlusBot Hangzhou, P.R. China
[3]Center for Autonomous Systems (CAS), The University of Technology, Sydney, Australia
∗Corresponding author

1 Introduction

The reliable mobility of the robots is a research focus for decades, of which the primary difficulty is to localize the robot. Addressing this problem, the most important solution is to build a global consistent metric map, and then localize the robot by comparing the acquired sensor data to the map (Dissanayake et al (2000); Thrun and Montemerlo (2006); Fox et al (1999); Montemerlo et al (2002); Kurt Konolige (2008); Mur-Artal et al (2015)). This class of solutions pushed a significant step for mobile robots toward real world application with the satisfactory accuracy and reliability in those relatively stable applications. When extending these solutions to long-term operation, dynamic changes of environments make it hard to achieve global consistency, as can be seem in Fig. 2. Lots of efforts have been made to stitch multiple sessions of mapping into one global

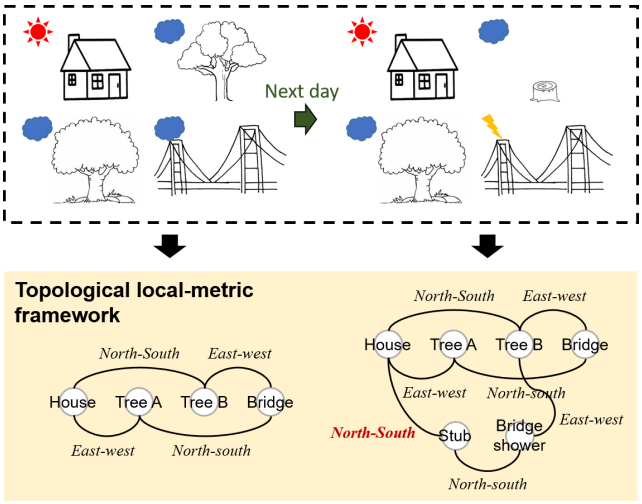
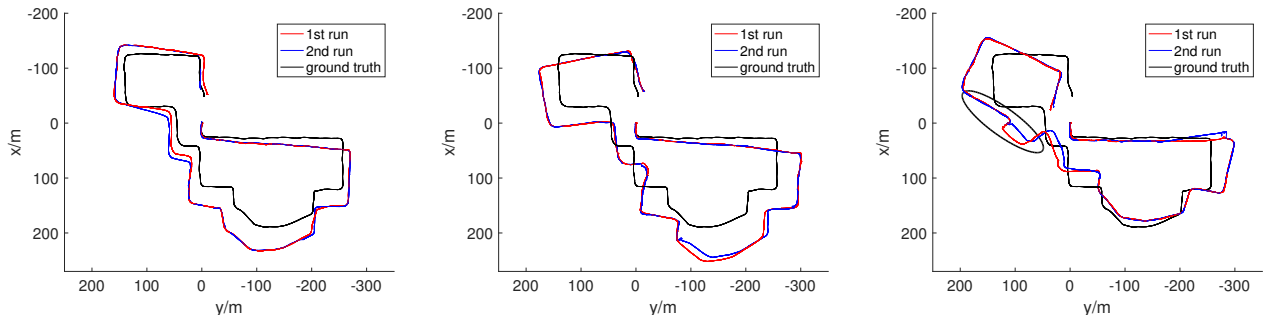


Fig. 1: The idea of the proposed topological local-metric framework (bottom) given the two-day evolving environment (top). There is a house, a bridge and two trees in the environment. The icons of sun and cloud indicate for the weather. Assume that we have a sensor that can measure the metric relation between two objects. Each node in the framework indicates for a submap describing the local environment with no global coordinates, while the edges between the nodes encode the geometric (geographic) information. During the mapping, error in relative measurement may happen caused by sensor error, or incorrect matching in front end, say, the relative observation in red. As the relative measurement is saved instead of global coordinates, the framework prevents the erroneous observation from affecting the other edges in the map, unlike the global consistency framework, in which the global poses of all nodes are affected by introducing only one erroneous observation. Besides, the evolution of the environment is recorded in the map, which is expected to be with stronger localization capability in the following days as the bridge in both cloudy and sunny condition are learned by the robot.

frame, so that the robot can localize itself across sessions (Mcdonald et al (2013); Newman et al (2009)). This solution, when correct, inherits all the advantages the global consistent framework have. However, as it calls for highly accurate alignment even between two sessions with unavoidable changes, and its complexity is related to the duration of operation, it cannot be easily achieved, especially in large facilities or outdoor environment. These two extremely difficult requirements prevent the solution from deployment in long term. On the other hand, topological mapping and localization are presented to reduce the requirement of accurate metric (Angeli et al (2009); Cummins and Newman (2008); Milford and Wyeth (2012); Lowry et al (2016); Churchill and Newman (2013)). In this solution, the sensor data are organized as a graph with nodes and edges encoding places and adjacencies. The localization thus becomes a problem of image retrieval, that only IDs of the matched images are given, without accurate metric information. When only the topological-level navigation is required, this solution is sufficient, while for the robot navigation calling for metric guidance, this solution is inappropriate. Churchill and Newman (2013) explores the organization of sessions in one graph to include more variations for higher success rate of localization, but in their work, metric only existed within one session, thus the successful localization in multiple sessions cannot be fused. In some studies, the metric information was inserted into the topological graph (Tully et al (2012); Konolige et al (2010)), but the solutions refer to the global consistent map to build the topological graph, failing to relax the requirements of accurate alignment and growing complexity.

In this paper, as shown in Fig. 1, our framework addresses the challenges by introducing the local metric into the topological graph, so that robot can localize and navigate itself in the environment without calling for the expensive global consistency, constrain the bad effect of getting erroneous observations in the local, and keep the complexity constant, raising the reliability of the system. Besides, on the top of this framework, a lifelong learning mechanism is proposed by letting the robot memorize the environmental changes to boost its performance of localization across multiple sessions, furthermore increasing the robustness against the uncertainty from the environment. In addition, the proposed methodology is independent of sensor types, which means laser scanner and vision are both supported by only modifying the way of alignment. Thus, the framework can be considered as the basis for more reliable robotic movement. The contributions of the paper can be summarized as follows:

- A topological local-metric framework (TLF) is proposed to organize and manage the sensor data collected by the robot across sessions. This framework combines the topological and metric maps in a unified graph.
- Algorithms of localization and navigation on the TLF are introduced for the robot to perform movement when global coordinates does not exist.
- A mechanism for robot to memorize the unvisited or changed places is presented to add and remove information in the TLF, so that the robot can have a memory of the changing environment across sessions.



(a) Dawn (b) Noon (c) Evening

Fig. 2: The ground truth and the global poses optimized by ORB-SLAM. With the robot keeping similar trajectory in two sessions, the global pose output by ORB-SLAM is referred to clarify the vulnerability of global consistency, also the advantages of relative configuration. Compared with the laser ground truth, the resultant trajectories are not accurate, which means the global coordinates defined by ORB-SLAM is not coincident with the global coordinates defined by laser and GPS. In addition, although the robot actually goes through the similar path in both sessions, the final trajectories maximally gives about 50m distance in the same physical place, indicating that the global coordinates is not coincident between the two sessions, even they are processed in one optimization. The main reason for this is the occurrence of unavoidable erroneous observations during the sessions. The circled region corresponds to the circled part in Fig. 7.

– Based on vision and laser sensors, we demonstrate the effectiveness and performance of the proposed framework in challenging outdoor navigation in long-term 21-session runs.

The remainder of the paper is organized as follows: In Section 2, the related works of robotic localization are reviewed and analyzed. In Section 3 and 4, the formulation of TLF and the algorithm of localization and navigation upon it are introduced, followed by the robot memorizing mechanism for lifelong learning based on the TLF in Section 5. Implementation details are illustrated in Section 6, while the experiments evaluating the performance of the proposed methodology is presented in Section 7. The conclusion is presented in Section 8, which completes the paper.

2 Related works

Localization and mapping of the robot has been studied for long time, named as simultaneous localization and mapping (SLAM), beginning from building a globally consistent map using filter based solutions, including extended Kalman filter, extended information filter and particle filter (Dissanayake et al (2000); Lauer and Stein (2015); Montemerlo et al (2002); Eustice et al (2006)). However, these filter based solutions are argued to be inconsistent as re-linearization cannot be conducted (Huang and Dissanayake (2007)). To avoid this problem, two branches of methods are developed. First, the graph based SLAM is proposed to State the trajectory of the robot and relative sensor alignment as nodes and edges, then the maximum likelihood estimation of the trajectory is solved by iterative non-linear optimization algorithms like Gauss-Newton or Levenberg-Marquardt, of which the solution is globally consistent as the re-linearization happens in each iteration (Thrun and Montemerlo (2006); Kummerle et al (2011); Mur-Artal et al (2015); Dellaert and Kaess (2006); Kaess et al (2008)). Another solution is to improve the consistency of the filter based solutions by fixing the linearization point when evaluating Jacobian at the first time, or finding the optimal linearization point by constrained optimization (Huang et al (2009, 2010)). These efforts significantly push the localization and mapping into general applications, but their growing complexity and vulnerability to erroneous observation exist as challenges. To deal with the first challenge, graph reduction is introduced into SLAM to control the complexity of SLAM with respect to the workspace area (Wang et al (2015, 2013); Carlevaris-Bianco et al (2014)). The second challenge is investigated by constructing robust backend to identify erroneous observations (Latif et al (2013); Lee et al (2013)). These modules improve the performance of the global consistent solutions, as a cost, the number of tunable parameters is large, and some of them do not have directive physical explanation. More importantly, the global optimization still exists, thus the challenges are relieved, but not eliminated.

To relax the large complexity of building a global consistent map, topology is considered by regarding each node being a submap to reduce the scaling problem, and the planning on the topological map is much more efficient (Konolige et al (2011); Angeli et al (2009); Rybski et al (2008)). In these solutions, each node in the topology denotes a metric pose, which means that the topological graph calls for global optimization, thus still experiences the same risk as the global consistent solutions. Based on such topological map, there are two ideas of localization. One is to keep the metric pose in the global coordinates using Bayesian filtering or local optimization (Tully et al (2012); Blaer and Allen (2002); Liu et al (2012)). The other is to find the topological localization through place recognition (Lowry et al (2016); Churchill and Newman (2013); Milford and Wyeth (2012); Cummins and Newman (2008)). The latter has stronger performance when the environment change happens, since the image level descriptors are more invariant than the feature point descriptors. The weakness is that this method cannot provide metric localization, thus cannot act as the feedback for robot navigation. Its application lies in the semi-automatic driving, rather than the fully autonomous robots. Simhon and Dudek (1998) used a similar map representation to ours, but it concentrated on map partition, while we focus on multiply sessions fusion for long-term localization. For more recent studies, some works find that the global consistency can be fully eliminated by only keeping the local metric, like Furgale and Barfoot (2010) and Krüsi et al (2015). Their methods are close to our TLF, but they aim at the teach session repetition. A similar work to our framework is Paton et al (2016), they used multiple sessions as bridging experience to fill the gap between the repeating session and the teaching session, which strongly improved the localization performance. The main difference between their work and ours is still that we have no teaching session in the framework, therefore sessions without any edges can be stored in our framework for localization, expanding the framework to a completed localization and mapping framework for long-term operation.

Dealing with the environmental changes during the robot's operation is also a topic for years. To resolve the transient dynamics, RANSAC based alignment methods are employed to remove outliers (Mur-Artal et al (2015)). Choi and Maurer (2016) tried to exploit more information from sensor input by integrating moving object tracking into localization module. Wolf and Sukhatme (2005) maintained two occupancy grids to distinguish static and dynamic objects in environments and used the former grid for localization. The more challenging situation is the low dynamic change, like structural or appearance changes. One solution is to track the environmental change so that the out-of-dated information can be pruned from the graph (Wang et al (2016); Walcott-Bryant et al (2012); Konolige et al (2010)). These methods are built upon the global consistent map so that the sensor data collected at different time from the same place can be compared to update the map. Another solution is to find more robust algorithm to align the two views under different disturbances, such as foreground segmentation, construction of the patch features, building illumination invariant color space or rendering a pre-surveyed map (Corcoran et al (2011); Mcmanus et al (2014); Pascoe et al (2015); McManus et al (2015); Paton et al (2017)). These alignment methods focus on providing better localization and thus can be a powerful localizer within our framework.

3 Map representation

We first introduce the map representation in TLF. The basic backbone of the TLF is a graph defined as $\mathcal{M} = \{\mathcal{N}, \mathcal{E}\}$ where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. For each $n_i \in \mathcal{N}$, the corresponding properties are defined as $n_i = \{s_i, l_i, N_i\}$:

- s_i is a submap built by the sensor data collected around this node. The submap is a concept, which can be either a real submap constructed using sensor data collected at multiple-steps, or a pure sensor measurement collected at one step. For example, an occupancy grid map can be defined as s_i, and a set of images can also be s_i. The coordinate frame of the submap is set at the first pose collecting the sensor data used for building this submap.
- l_i is a localizer responsible for data alignment which is a function $p_{t,i} = l_i(d_t, s_i)$ with current sensor measurement d_t and the submap s_i as the input, the pose $p_{t,i}$ of the robot in s_i as the output. As an instance, iterative closest point localizer can be applied as l_i to acquire $p_{t,i}$ by aligning the current captured point cloud d_t to the point cloud submap s_i Besl and Mckay (1992).
- N_i is a proxy for other properties corresponding to the node. In this paper, we assigned N_i with options including place descriptor for loop closure, differential GPS (DGPS) and the date/time (DT) when the submap is created. The place descriptor is employed for global localization, the DGPS for ground truth building, and DT of submap for navigation, which is introduced in sequel.

Then for each $e_{ij} \in \mathcal{E}$ connecting the neighboring nodes n_i and n_j, we define $e_{ij} = \{p_{i,j}, E_{i,j}\}$:

- $p_{i,j}$ is a rigid relative pose between node n_i and n_j, which can be obtained from the odometry like sensor measurement during the creation of the graph, including wheel odometry, inertial measurement unit, or visual odometry. These sensor measurements are all accurate in local short-time window, thus the $p_{i,j}$ is considered sufficient to encode the local geometry. Another source of the edge is the result of localizer or loop closure, which is determined by the sensor data alignment, thus is also accurately evaluated in local metric.
- $E_{i,j}$ is a proxy for other properties corresponding to the edge, which is also application dependent. In this paper, $v_{i,j}$ is assigned to $E_{i,j}$ as one property, which is the traversability between node i and node j. This value is important especially in vision based navigation. When the camera is front looking and the robot capturing the sensor data moves from node i and node j, $v_{i,j}$ tells the path from node i to j is unidirectional.

When there are multiple sessions, or multiple robots, their corresponding maps $\mathcal{M}$ could be merged together by finding the loop closures using global localizer (Cummins and Newman (2008)) or aided by external sensor, like GPS. We only store the relative measurements but no global poses, thus no unified global coordinates and no global optimization is required. The difference across different versions of the map only occurs in adding or removing of the nodes, no geometric information is modified, as they are raw alignment results. It leads to an additional advantage if merging and optimization are required for visualization, as in our framework, the utilization of the maps and the optimization of the maps are totally decoupled, thus the latter can run asynchronously using cloud computation. Even though multiple robots are using the different versions of maps, the positioning information are still communicable. By eliminating the real-time requirement for synchronization, the connections across multiple maps can thus be discovered and verified using slow but accurate algorithms, even the manual curation.

The core difference between our TLF and pose graph is that no global pose is required in TLF. When global poses are needed, global consistent optimization is necessary. The pose of the same physical place thus may change each time a new observation is acquired, making the older versions of maps unusable. Finally many problems occur: the erroneous alignment may ruin the whole map, the growing complexity leads to the growing computational time, and the necessary real-time synchronization of maps in all robots to avoid the out-of-dated positioning communication. While for TLF, the geometry are only encoded in the edge, which cannot affect the geometric information in other edges, constraining the erroneous alignment in local. Besides, a new node is added into TLF by simply connecting it with the existing nodes using the relative poses as edges, thus the complexity can be kept as constant. As the process of the mapping is much simpler than the global consistent framework, the implementation is more directive, also lowering the engineering risk. In a word, the core feature of TLF is that: it takes the front-end output directly to satisfy the navigation stage, skipping the intermediate stage of constructing global consistent pose graph, which we think is *over-qualified* for navigation.

4 Localization and navigation

TLF provides the map representation for the mobile robots to move around the workspace with improved robustness and less complexity. The real movement relies on the localization and navigation algorithms. Localization outputs the relative pose denoted as $p_{t,i}$, meaning that the pose of the robot at the tth timestep with respect to the ith submap. Obviously, this localization process consists of finding a node i and then calling the corresponding localizer l_i to estimate the relative pose. For navigation, as the metric is only contained in local coordinates, the general global metric planner cannot be employed. To navigate on TLF, we propose a double-layer planner with topological and metric planning being the top and bottom layer, respectively. This algorithm is inspired by the graph representation of a manifold Liao et al (2016), which defines Euclidean measure on the tangent plane at each point instead of a unified metric.

4.1 Relative Localization

As defined above, each node in $\mathcal{M}$ is assigned with a localizer, which gives the pose of the robot relative to this submap through aligning the current sensor data to the submap. Thus there can be multiple relative pose estimations by activating the localizers in a subset of nodes around the current position, denoted as $\{p_{t,k}|k \in Q(t)\}$, where $Q(t)$ is the nodes subset. These relative poses form an edge set $\mathcal{E}_{loc}$. Then we can formally define the process of our relative localization pipeline: first, a topological predictive selector finds the $Q(t)$, second each node n_k in $Q(t)$ apply its localizer l_k to generate $\mathcal{E}_{loc}$, and finally a small scale pose graph optimizer is applied to find relative localization. An illustration can be found in Fig. 3.

We introduce the localization from the last step. Given $Q(t)$, we build a subgraph G with nodes in-

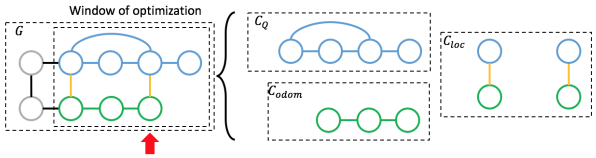


Fig. 3: Example formulation of the relative localization with $Q(t)$ indicated by blue subgraph (C_Q), which is the 3-nearest neighborhood of the current pose, pointed by red arrow. The odometry sliding window of length $l = 2$ is indicated by green subgraph (C_{odom}). The orange is the output of localizers (C_{loc}). These edges and nodes are included in the optimization. The gray node and black edges are not included.

cluding current timestep n_t and its previous l timestep $\{n_{t-1}, \ldots, n_{t-l}\}$, as well as all ones in $Q(t)$. The edges in G include the current odometry edges $\{\mathcal{E}_{odom}\}$, like $e_{t-1,t}$, the map edges $\mathcal{E}_Q$ between nodes in $Q(t)$, and the localizer edges $\mathcal{E}_{loc}$. On the subgraph G, we formulate the three cost functions for three kinds of edges:

$$C_{odom} = \sum_{e_{\tau-1,\tau} \in \mathcal{E}_{dom}} \|p_{\tau-1,\tau} - e_{\tau-1,\tau}(p_{\tau-1}, p_\tau)\|^2_{\Sigma_d} \quad (1)$$

$$C_Q = \sum_{e_{i,j} \in \mathcal{E}_Q} \rho(p_{i,j} - e_{i,j}(p_i, p_j))_{\Sigma_q} \quad (2)$$

$$C_{loc} = \sum_{e_{\tau,k} \in \mathcal{E}_{loc}} \rho(p_{\tau,k} - e_{\tau,k}(p_\tau, p_k))_{\Sigma_l} \quad (3)$$

where $\rho(\cdot)$ is the robust kernel, which is Huber kernel in this paper, p with single subscript is the unknown poses variables to be estimated anchored at a node among $Q(t)$. The subscript Σ_* indicates the weighted value, such as $\|e\|^2_\Sigma = e^T \Sigma^{-1} e$, and $\rho(e)_\Sigma = \Sigma^{-1} \cdot \rho(e)$. The weights Σ are also referred as information matrix in SLAM community. These variables are estimated by optimization as

$$\hat{p} = \arg\min C_{odom} + C_Q + C_{loc} \quad (4)$$

where $\hat{p}$ is the final estimates. Note that the edges in the submap and the localizers may be generated by loop closure, thus probably erroneous, so that the robust kernel is only assigned to the terms in C_Q and C_{loc}. Since the graph is built in local, the odometry constraints C_{odom} is a strong and relatively accurate prior to regularize the optimization compared to the global optimization, so the robust kernel under this context is prone to eliminate the bad effect of erroneous alignment with higher probability. Even the robust estimation does not identify the erroneous edge, there is no subsequent effect when this edge leaves the optimization subgraph G.

We then introduced the first step, selection of $Q(t)$. There are two scenarios for $Q(t)$, global topological localization and pose tracking. When robot has no prior on its pose, $Q(t)$ is determined by the loop closure algorithms, which is studied in the robotics community for long time, like Fabmap or DBoW (Cummins and Newman (2008); Cadena et al (2012)). Loop closure algorithms yield ranking of nodes based on the place similarity. $Q(t)$ is then assigned by selecting the top N nodes for the later geometric localization. Another way is to utilize the external aids like GPS, the current GPS reading can be compared to the GPS information in nodes properties, among which the nearest N nodes are set to $Q(t)$. In pose tracking, we combine two sources of information, the geometric information and semantic information. For the geometry, the odometry is first assigned to the last localized pose to predict the current pose. The anchor node of the final relative localization in (1) is defined as the nearest node to the predicted current pose in $Q(t)$. Then a K-nearest neighborhood of the anchor node is extracted from G to form $Q(t)$. For the semantics, we also consider the DT in N_i of the nodes in the geometric neighborhood, like morning and afternoon, or weather conditions, to further reduce the candidate set $Q(t)$, by excluding the nodes in the neighborhood but with unmatched semantic properties.

4.2 Manifold navigation

The map is Stated in the 3D space, but it is actually a 2D manifold which is locally smooth, and a curved ground surface in overall. This is the underlying reason that the robot pose is Stated in 3D, but cannot move to an arbitrary point in whole 3D space. Following this local property of manifold, we develop the navigation algorithm in two layers. The top layer only cares about the topological graph, it searches for a shortest path on the graph, which is a sequence of nodes the robot should go through, denoted as $H = \{h_i\}$, where h_i is the subscript of the ith node in the found path, the consecutive two nodes have an edge connecting them. In this step, the properties in the nodes and edges are considered as weights in the shortest path search. In this paper, by considering the path length in the edge, and the DT of the nodes, the resultant path is short and more probable for successful localization of the robot, since the cost encoding the DT drives the path to go through nodes with similar time in a day and weather to that of the current session.

Given the sequence H, the bottom layer is responsible for motion planning. When the robot passes a node in H, it is dropped from the H, so the first (starting) node in the sequence H is always the anchor node for

the current relative localization, i.e. the difference from the starting point of current path H, and current anchor node for the robot is 0 in metric. Then the bottom layer sets the next gth node, or the final node in H as the current metric goal h_g, which is computed by concatenating the relative poses $p_{i,j}$ along the path from node h_1 to node h_g

$$p_{h_1,h_g} = p_{h_1,h_2} p_{h_2,h_3} \cdots p_{h_{g-1},h_g} \tag{5}$$

With the relative localization $p_{h_1,t}$, we can derive the metric goal in robot coordinates as

$$p_{t,h_g} = p_{h_1,t}^{-1} p_{h_1,h_g} \tag{6}$$

By inserting this metric goal to the motion planner, the robot can go through the path and get the final goal.

The crucial idea in the navigation algorithm is to re-compute the metric goal in the bounded g-step ahead from current anchor submap each time when the submap for localization changes. This is different from the result that we pre-compute all the nodes in the beginning. Suppose the length of the whole path is N_H and the error in each edge is ϵ, then if the final node is computed in the beginning, the error is $N_H\epsilon$. If we compute the g-step ahead goal when the starting node is exactly the anchor of the current relative localization, the error of the metric goal is $g\epsilon$, $g \leq N_H$, which means the error can be bounded by setting g. Furthermore, when the robot approaches the goal, the error of the metric goal $< g\epsilon$, and ultimately ϵ when the robot is anchored in $n_{h_{g-1}}$. As the Euclidean metric exists in the tangent plane expanded at every points on the manifold, motivating our method to re-compute the metric goal at every anchor node, so that the metric is well defined, and the error is controlled. A simple example of the navigation is shown in Fig. 4.

The navigation algorithm proposed above limits the final goal of the robot, because it can only reach places with nodes. A small modification is made to deal with this problem. To go to a specific place, the robot firstly drives to the nearest node to this place, following the proposed pipeline. Then it drives to the goal directly, because the error of metric goal is small enough for navigation. The enhanced algorithm is listed in Algorithm 1.

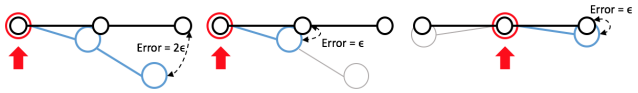


Fig. 4: A simple case of manifold navigation. The black graph indicate for the ground truth. The gray graph indicate for the TLF graph, while node and edge in blue are in the planned topological path. The node pointed by the red arrow is where the anchor of the robot relative localization is. The goal of the robot is to reach the right node in the graph. When a global metric planner is employed, the error is $N_H\epsilon$, namely 2ϵ, thus large (left). When we apply the manifold planner with 1-step ahead (middle and right), the error is ϵ, as we call the metric planner each time the anchor changes.

Algorithm 1 Mainifold Navigation

Require: Final goal: D, Topological local-metric map: M, Anchor node: R
1: $N_D \leftarrow$ find the nearest node to D in M
2: $P \leftarrow$ search a path from R to N_D on M
3: **for** $p \in P$ in order **do**
4: Robot moves to p
5: **end for**
6: Robot moves to D

5 Memorizing mechanism

The last component in the proposed framework is to build the map as the representation introduced in Section III. In long term, special mapping session is not preferred since the workspace environment is large and change occurs from sessions. In our framework, the environment is learned across sessions, each robot in each task session, and also the human involved session, are regarded as sessions. By sequentially feeding the sessions into the framework, the basic idea is to simply memorize the information which is never seen before. Intrinsically, both mapping and localization are processed simultaneously in TLF.

5.1 Mapping by localization

In each session, the localizer is try to localize the robot in $\mathcal{M}$. During a session, the anchor node is firstly localized by the global topological localizer when the robot has no prior on its position. Then the pose is tracked locally. The robot can sometimes lose the localization and rely on the odometry only, which outputs the pose on the submap where the robot lastly is localized before loss. At this phase, the robot again calls for the help of global topological localizer to find a submap for localization recovery. The loss of the localization generally consists of three reasons: First, the robot avoids the obstacles because of the transient high dynamic, like pedestrian or other moving objects, causing the serious perspective change compared to the recorded map information. Second, the environment experiences the low dynamics, like weather condition change, or the structural change like building repairing. Third, the robot

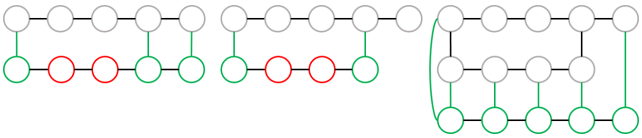


Fig. 5: A simple case of memorizing mechanism. The gray graph is $\mathcal{M}$, the nodes in red belong to $\mathcal{M}_{c,u=0}$ while green $\mathcal{M}_{c,u=1}$. The green edge are generated by the successful localization. At first, there are two nodes cannot be localized in the current $\mathcal{M}$ (left), then a subsequence $m_{c,u=0}$ of continuous nodes not localized ($u = 0$) and its precedent and subsequent nodes are inserted into $\mathcal{M}$ (middle). After that, the new session can be localized completely by the new map (right).

goes through a new place whose information is never recorded in $\mathcal{M}$, an example is the first session when $\mathcal{M}$ is empty.

Upon the analysis above, we have two findings: First, the chain graph of the current session $\mathcal{M}_c$ consists of two categories of nodes, localized nodes ($u = 1$), or nodes not localized ($u = 0$), i.e. $\mathcal{M}_c = \mathcal{M}_{c,u=1} \cup \mathcal{M}_{c,u=0}$. Second, change of environment, dynamic of traffic participants, and significant illumination variations should exist in the period $\mathcal{M}_{c,u=0}$. Therefore the memorizing mechanism is designed to add all these nodes $\mathcal{M}_{c,u=0}$ into $\mathcal{M}$. In our setting, we go over the nodes in $\mathcal{M}_{c,u=0}$ sequentially, leading to a session broken into several segments with each one being a continuous sequence of un-localized nodes $m_{c,u=0} \in \mathcal{M}_{c,u=0}$. If the length of $m_{c,u=0}$ is higher than a threshold, which means the loss of localization is not caused by transient dynamics, we add $m_{c,u=0}$, and its localized precedent one node and subsequent one into the map $\mathcal{M}$ for future utilization as shown in Fig. 5. This procedure is repeated for all segments. If there is no localized precedent or subsequent node, meaning that even global topological localizer cannot find a similar node, which occurs in the third situation, we insert $m_{c,u=0}$ into $\mathcal{M}$ with its existed precedent or subsequent localized node, or create a new separated graph, e.g. when the $\mathcal{M}$ is empty, the whole session belonging to $\mathcal{M}_{c,u=0}$, is entirely inserted into $\mathcal{M}$.

In summary, the memorizing mechanism can be understood as automatic switching between mapping when the robot does not know the environment, and localization when the robot knows it. With this mechanism, $\mathcal{M}$ is expected to include the periodic change of the environment, which gradually reduces the localization failure caused by this reason. The algorithm is summarized as Algorithm 2.

5.2 Maintenance

Each node actually indicates for a place with a specific appearance. Consequently, a physical place with different appearance, e.g. different illumination or dynamic obstacles, are regarded as different places in our framework. In long term, such mechanism causes the size of the map ever growing to unlimited size, as the out-of-dated information are all stored in the map $\mathcal{M}$. The maintenance of the map aims at filtering submaps indicating for places with low but aperiodic dynamics. To solve this problem, one way is to utilize the temporal statics. We record the number of the localizer being called N_{call}, and the number of the successful localization $N_{success}$ for each node n_i in a sliding time window, say a day. The first number indicates the frequency that a node is traversed during the daily tasks. If a localizer is hardly called, it means that the corresponding node is not in the regular daily task path, which may be included in the map by transient obstacle avoidance. Therefore, pruning such nodes from the map is reasonable. The ratio $\frac{N_{success}}{N_{call}}$ indicates that whether the difference between the submap and the current observation is large. When the ratio stably decreases, it means that the current environment is different from that in the submap at the similar geographic place, which can be caused by low dynamics. Therefore the stored submap is out-of-dated, which can also be pruned. Since we determine to call the localizer based on the node properties N_i, like weather and time of the day, such pruned nodes are mainly caused by the structural change. In addition, if the ratio decreases occasionally, it may be caused by the transient dynamics, thus no action is applied. As a result, the memorizing mechanism helps the robot to improve the localization performance, and also keep the size of the map stable at the same time.

6 Implementation

The proposed framework can be applied to different sensors, as long as implementing corresponding localizers, as mentioned in Section 3. To illustrate effectiveness of the proposed framework, an implementation for stereo camera is designed, namely, stereo localizer. It computes the relative transformation between two pairs of stereo images, one of which is from live frames and the other is from submaps. Thus each submap is a pair of stereo images. In this paper, the stereo localizer is an implementation of feature point based quad-match method (Geiger et al (2011)). As in Geiger et al (2011), if less than 6 matches are found, or RANSAC is not converged, it is assumed that matching is failed. If none of

Algorithm 2 Memorizing mechanism

Require: Localization successful mask of frame j in current session: s_j, Sensor measurement of frame j in current session: f_j, Map before current session: M_{in}
Ensure: Map after current session: M_{out}
$S \leftarrow null$ ▷ Segment
$M_{out} \leftarrow M_{in}$
for $j \in$ all frames in current session **do**
  if $s_j = true$ **then** ▷ Localization is successful
    if $S \neq null$ **then**
      Insert S to M_{out}
      $S \leftarrow null$
    end if
  else ▷ Localization is failed
    Append f_j to S
  end if
end for
$M_{out} \leftarrow$ MAINTENANCE(M_{out}) ▷ Pruning redundancy of the map as session 5.2

$Q(t)$ matches with the live images, localization is said to be failed, and new nodes are generated and cached, until the next successful localization. After re-localization, the cached images are added to the map, with each pair of images being one submap.

The topological predictive selector is very important, for that too much nodes in $Q(t)$ may bring unnecessary computation, while insufficient nodes may cause localization failure. In practice, K is adjusted dynamically, namely, $K = max(2 + L, 7)$, where L is number of lost frames since last successful localization. This ensures efficiency in places with high localization success rate and enough search range during localization failure. To achieve real-time, if there are more than 5 nodes in $Q(t)$, only 5 nodes of them are chosen randomly as final $Q(t)$.

In environment with large changes, localization may fail for a long time. To increase efficiency, localization is called every 1 meter. What's more, after failing more than 7 times, localization is stopped, and global localization is started. We use visual bag of words method (Gálvez-López and Tardos (2012)) for global localization. The live image is used to extract a descriptor given a precomputed vocabulary, to find the potential submap with closest descriptor. If live stereo pairs and the potential submap pass the geometry check of stereo localizer, it's regarded that a loop closure is found, and re-localization is reached.

7 Experiment

In the experiment, a four-wheeled mobile robot is employed as platform equipped with a ZED stereo camera[1] and a VLP-16 Velodyne LiDAR[2] as shown in Fig. 6. All algorithms including localization and navigation are deployed on a laptop with Intel i7-6700 CPU 2.6GHz and 8G memory. The data for experiments is collected from challenging outdoor environment in Hangzhou, China, hybrid with both unstructured and structured segments. The time for sunrise and sunset is around 7:30 and 17:30. The transient dynamics include moving cars, pedestrians and cyclist sharing the space with the robot. The low dynamics include lots of varying parking cars and the different time of the day. There are totally 21 sessions in the dataset resulting in more than 23km over 6.5 hours across 3 days. The stereo image pairs, 3D laser scans, wheel odometry and DGPS information are available in this dataset. More than half of the DGPS data are null due to the disturbance from occlusion of trees, also reflecting the difficulty of the dataset. The metadata of the dataset is shown in Tab. 1. The ground truth of the dataset is built by global consistent pose graph SLAM with laser scans registration and the available DGPS measurement as binary and unary edges since these two sources of sensor data are highly accurate. As shown in Fig. 7, by overlaying the ground truth on the satellite imagery, one can see that the trajectory are all within the physical roads, supporting the accuracy of the ground truth.

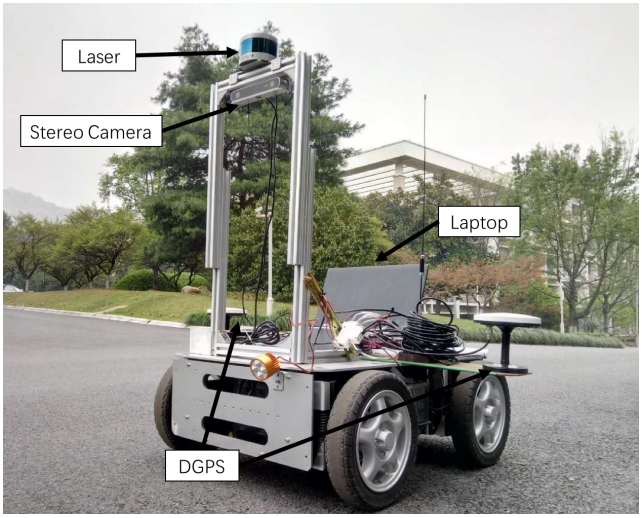


Fig. 6: Experimental platform with highlighted 3D laser, stereo vision, DGPS and laptop.

As laser and DGPS are utilized for ground truth construction, the vision sensor are selected for validation of the framework. Three types of performances are evaluated to test the effectiveness and efficiency of the proposed TLF and memorizing mechanism: the accuracy and robustness, the complexity when running in

[1] https://www.stereolabs.com

[2] http://www.velodynelidar.com

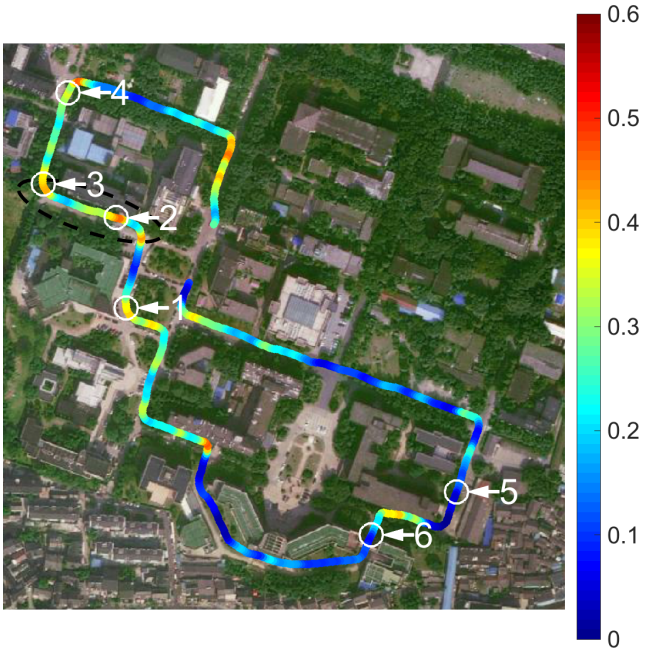


Fig. 7: Satellite imagery of the place where the dataset is collected. The highlighted path is passed by the robot in each session with a length of more than 1km, resulting in 23km over 6.5 hours across 3 days. The color of the trajectory indicates the localization failure rate with respect to the position. The number highlights the change during sessions at a specific position. The circled region corresponds to the circled part in Fig. 2c.

long term, as well as the improvement when deploying the memorizing mechanism. The navigation algorithm is validated by an autonomous running in the mapped area by the robot. The comparative technique selected for baseline is the State-of-art global consistent stereo vision based ORB-SLAM2 (Mur-Artal et al (2015)).

7.1 Accuracy and robustness

To evaluate the localization accuracy, we select three pairs of sessions in different time of day to compare the localization accuracy listed in Tab. 2. Given the relative pose estimation $p_{i,t}$, the ground truth relative pose $\bar{p}_{i,t}$ is obtained by picking the corresponding anchor node and current node and then computing the relative pose between the two absolute poses. The error for each pair of ground truth and estimated result is calculated by

$$e = p_{i,t}\bar{p}_{i,t}^{-1} \tag{7}$$

where e is the error pose. Following the measures in Mcmanus et al (2013), the lateral error is the x-component and the heading error is the yaw-component in the error pose. The statistics of the error are shown in Tab. 3.

Table 1: Overview of dataset

ID	Start Time	Duration (mm:ss)	Total Time (hh:mm:ss)
s1	2017/03/03 07:52:31	17:44	00:17:44
s2	2017/03/03 09:20:13	18:45	01:46:27
s3	2017/03/03 10:23:11	18:14	02:48:543
s4	2017/03/03 11:48:03	18:17	04:13:49
s5	2017/03/03 12:59:16	19:12	05:25:57
s6	2017/03/03 14:34:43	19:24	07:01:36
s7	2017/03/03 16:05:54	18:39	08:32:02
s8	2017/03/03 17:38:14	18:01	10:03:44
s9	2017/03/07 07:43:30	17:54	96:08:53
s10	2017/03/07 09:06:04	18:46	97:32:19
s11	2017/03/07 10:19:45	19:04	98:46:18
s12	2017/03/07 12:40:29	18:42	101:06:40
s13	2017/03/07 14:35:16	19:01	103:01:46
s14	2017/03/07 16:28:26	17:59	104:53:54
s15	2017/03/07 17:25:06	18:34	105:51:09
s16	2017/03/07 18:07:21	19:49	106:34:39
s17	2017/03/09 09:06:05	17:50	145:31:24
s18	2017/03/09 10:03:57	17:52	146:29:18
s19	2017/03/09 11:25:40	18:17	147:51:26
s20	2017/03/09 15:06:14	19:13	151:32:56
s21	2017/03/09 16:31:34	19:36	152:58:39

Table 2: Sessions for localization accuracy evaluation

First session	Second session	Description
s1	s9	Dawn
s5	s11	Noon
s8	s15	Evening

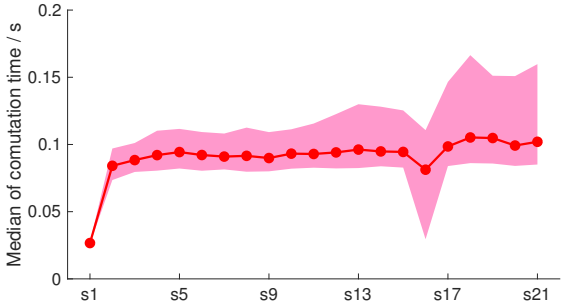


Fig. 8: The evolution of computational time for 21 sessions using TLF.

For the large difference between mean and median, the main cause is the insufficient matching, leading to erroneous observation or localization failure. As one can see, TLF gives much better robustness against these erroneous observations in both lateral and heading error, since the standard deviation and mean are much smaller than that of ORB-SLAM, and for median lateral and heading error, it can be found that the two methods are almost the same, at least illustrating that the two methods can achieve similar performances in localization accuracy.

Table 3: Localization accuracy for ORB-SLAM and TLF

	Dawn					
	Lateral RMSE (m)	Lateral Std. (m)	Lateral Median (m)	Heading RMSE (°)	Heading Std. (°)	Heading Median (°)
ORB-SLAM	3.762	3.252	**0.132**	4.658	4.026	**0.981**
TLF	**0.461**	**0.371**	0.150	**3.898**	**2.928**	1.648
	Noon					
	Lateral RMSE (m)	Lateral Std. (m)	Lateral Median (m)	Heading RMSE (°)	Heading Std. (°)	Heading Median (°)
ORB-SLAM	3.177	2.903	**0.156**	**4.649**	4.019	**0.495**
TLF	**0.596**	**0.490**	**0.156**	4.862	**3.844**	1.807
	Evening					
	Lateral RMSE (m)	Lateral Std. (m)	Lateral Median (m)	Heading RMSE (°)	Heading Std. (°)	Heading Median (°)
ORB-SLAM	4.823	4.207	**0.145**	5.013	4.344	**0.609**
TLF	**0.730**	**0.620**	0.192	**3.986**	**2.890**	1.808

We further look into the robustness against the erroneous observations. If ORB-SLAM is regarded as relative localizer, its global consistent optimizer propagates the error from erroneous observation to other poses, which finally also affects the relative localization. From Tab. 2 one can find that TLF achieves better RMSE and standard error than ORB-SLAM while the medians of both methods are similar, meaning that the relative localization of TLF is more stable. This can also be seen from the circled regions in Fig. 2c and Fig. 7, which are corresponding to the same location. ORB-SLAM mistakenly observed a sharp movement at the first run, resulting in that the second run of ORB-SLAM had a large offset (Fig. 2c), which is catastrophic for navigation under assumption of global consistency. Although our framework suffered from wrong measurement at the same location, which causing high failure rate (Fig. 7), the error will not propagate to the later localization due to local property, which ensures correct navigation.

7.2 Complexity

Besides the vulnerability to error, the cost of global consistency also includes the growing complexity of storage and computational time. We feed all the 21 sessions to both ORB-SLAM and TLF. The mean computational time for each localization is recorded. For ORB-SLAM, the system crashes in the 3rd session due to the overflow of the 8GB memory, since all the keyframe poses are included in the global bundle adjustment. For TLF, the result is shown in Fig. 8. One can see that the evolution of time with respect to the number of sessions keeps constant, reflecting a bounded complexity. The main variation of the system is the number of calling global localization, as it takes much longer time than position tracking and is related to the number of nodes in the TLF. Therefore, the constant evolution of the time also suggests that the number of nodes in the TLF is getting stable, controlling the global localization indirectly. This comparison clearly shows that the complexity of our system is much lower than the global consistency system.

7.3 Lifelong learning

The memorizing mechanism is the crucial component enable long term autonomy of robot through lifelong learning. We evaluate the effectiveness of this module by comparing the rate of successful localization with and without the memorizing mechanism. The experiment follows the configuration above by feeding 21 sessions to the TLF. The success rate is $\frac{N_{success}}{N_{call}}$. A success of the localization is defined in Section 6. The result is shown in Fig. 9. We can see three results: First, when the memorizing mechanism is on, the success rate gradually increases, and stably stays around 80%. Session 16 is special, as it is the first time that the robot run at night, so the rate falls down. Second, when the memorizing mechanism is off, the success rate decreases in each day with respect to the change of time within one day. Because the map only includes Session 1, so the success rate in later sessions drops down because the difference between the current session and the map is getting larger. In addition, in Session 9 and 17, the rate is high as the map and the current session are in the same time of a day. Third, the performance of TLF with memorizing mechanism is dominantly better than that without memorizing mechanism. Actually, the performance of the latter is the lower bound of that of the former, since more information are memorized by the robot. With this comparison, the value of memorizing mechanism is clearly validated, showing that the robot's lifelong learning of environment is possible.

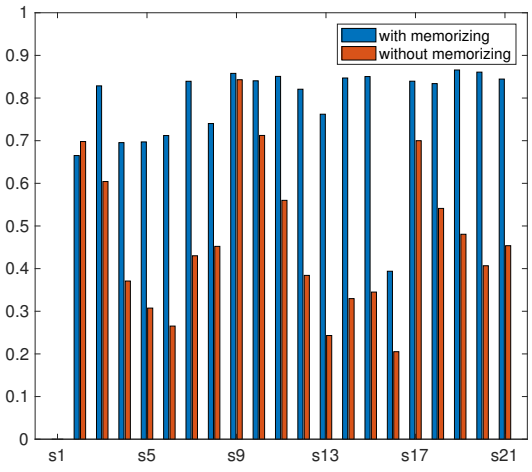


Fig. 9: Localization success rate with and without the memorizing mechanism.

To evaluate the quality of localization with memorizing mechanism, we demonstrate the distribution of localization error across multiple sessions in Fig. 10. One can see that the error of the system keeps the similar levels of localization. This result indicates that even under the changing environment (different time of the day), the quality of the localization does not degenerate, which is contributed by the memorizing mechanism and the relative localization. It quantitatively verifies the effectiveness of the proposed framework, illustrating its promising performance in long term operation of mobile robots. Note that the localization error of Session 16 is significantly larger, which is due to the first time of running at night.

Still refer to Fig. 7, we further look into the places where the localization success rate is low. The selected pictures shown in Fig. 11 are the failure localization at some places. One can see that the main reason to localization failure is the serious change of illumination and the low dynamic disturbance, which can also be seen from Fig. 12. The former significantly changes the shadow, resulting in very different texture on the ground, which confuses the localizer. The latter obviously changes a large portion in the images, even confusing the human. By remembering these changes by the memorizing mechanism, though the success rate is still not very high, the localization are improved as shown in Fig. 9. These results verifies the importance of lifelong learning, and the possibility of deploying TLF in long term operation. For the places with low rate of localization failure, the illumination change is much slighter, and the texture on the building lead to a more stable feature matching in visual localization.

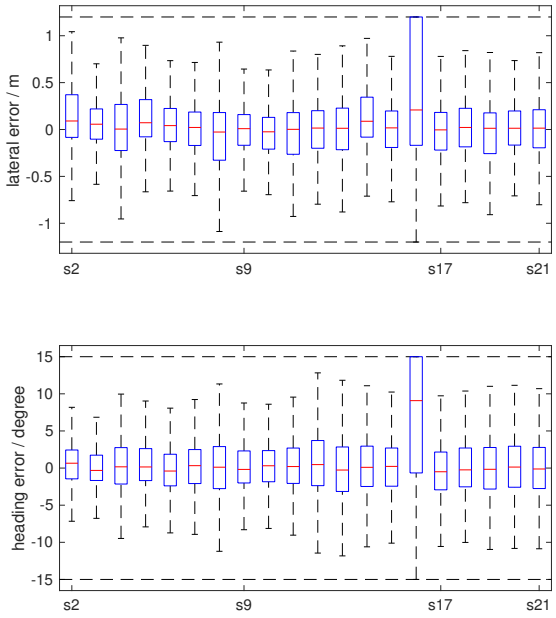


Fig. 10: Localization accuracy reflected by lateral error (top) and heading error (bottom) with respect to sessions.

To show that the memorizing mechanism is feasible respecting to storage limitation, size of the map after each session is presented in Fig. 13. Pruning strategy proposed in Section 5 started from the second day. Firstly, a drop in the number of nodes occurs after the first run in the second day, because the forgetting mechanism is applied, redundancy accumulated in the first day is pruned. Secondly, an obvious increase appears after the last run in the second day, since the night data appears, which is not included in the first day. Thirdly, the trend of nodes growing is slow down with respect to the first day, and fluctuates around zero at last. These findings validate the feasibility of the memorizing mechanism to maintain the storage of the map in long term.

7.4 Navigation

The final task of TLF is to provide the navigation to the mobile robot. The start and goal of the path is set the same as the 21-session data. We validate the navigation on TLF by letting the robot run in this 1.1km path autonomously. The real trajectory of the robot is calculated by aligning the laser scan to the laser built map as above. The intermediate goals are selected every

Fig. 11: Some examples of places highlighted in Fig. 7 for good and bad localization. Images of the first 4 rows from top to bottom are captured in place 1 to 4 with low successful localization rate indicated in Fig. 7. The last two rows are captured in place 5 and 6 with high successful localization rate.

Fig. 12: Cases of feature points matching. Row 1 to 3 present 3 examples of failure case, while row 4 is a successful one.

g steps. The robot aims at this series of goals consecutively. The indicator is designed similar to (7) as follows

$$e = p_{t,goal}\bar{p}_{t,goal}^{-1} \tag{8}$$

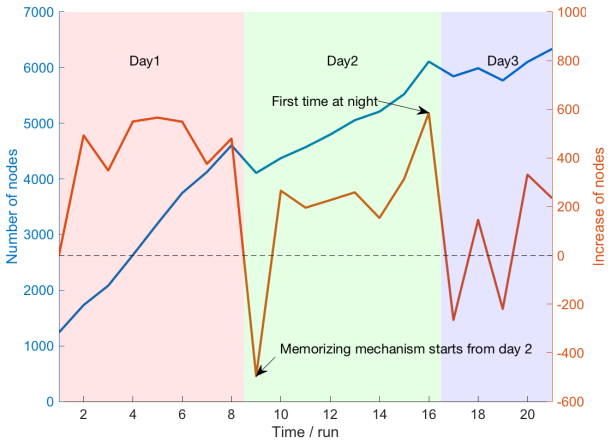


Fig. 13: Numbers of nodes after competition of each session. Pruning of nodes start from day 2 (session 9).

where $p_{t,goal}$ is the current goal in the robot pose at time t, and $\bar{p}_{t,goal}$ is the ground truth built by laser data. In this part, the rooted squared lateral error and heading error are employed to demonstrate the comparison of their distribution more clear, which is shown in Fig. 14. As the metric only exists locally in TLF, the error when $g = 10$ is much more centering on 0, thus much lower than $g = 50$, which agrees with our theoretic derivation that the error of the navigation is

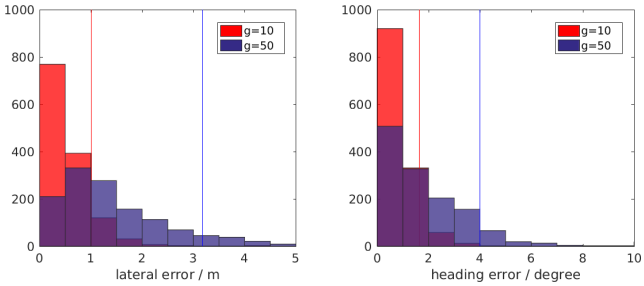


Fig. 14: The distribution of the error of goal in robot coordinates at each timestep. The vertical lines are the mean plus the standard deviation.

controlled by g. When $g = 10$, the heading of the robot provided by the TLF is about 2 degrees compared to the laser results, hence sufficient for the robot to reach the goal autonomously, thus deployed in our real robot experiment. Human intervention occurs only three times during more than 20 minutes' autonomous navigation to give the way to pedestrians and a car. It is because this experiment is intended to validate sufficiency of our localization method for navigation in sense of path following, no obstacle avoidance is used. The whole navigation is attached in the video which is captured by the left camera of the forward looking stereo camera, validating the effectiveness of the framework.

7.5 Limitations

From the results above, we thoroughly verify the advantages of TLF compared to global consistent framework. Then we talk about the limitations of TLF. First, if the navigation is the goal, then TLF is a better choice with its performance and lightweight. However, the cost of the TLF is the loss of reconstruction. If we want to reconstruct the environment for visualization or simulation, then globally consistent optimization is the right choice, but TLF provides an architecture decoupling the localization and navigation from the global optimization. Second, in TLF, the same place with different illumination is hard to be detected by loop closure, which is also the case in global consistent framework. TLF avoids the risk of incorrect loop closure by constraining the error in local, but it cannot add more loop closures edges, which may affect the selection of $Q(t)$ in neighborhood selection. For practical application, inclusion of low-cost GPS is a potential way to identify more edges. This topic is studied in the front-end related works, which is beyond the scope of this paper.

8 Conclusion

In this paper, we present the TLF for robot's localization and navigation in long term. Specifically, the framework makes use of the relative localization and navigation to avoid the growing complexity and vulnerability against erroneous observations in the global consistent framework. Besides, the memorizing mechanism is proposed to add the lifelong learning capability to the mobile robot, enabling a better long term autonomy. With TLF, the robot is expected to be more reliable in real dynamic environment without losing the accuracy compared with conventional global consistent framework. This hypothesis is further verified by the experiments on a 23km 3-day dataset. Finally, the autonomous navigation based on the TLF completes the validation of all functions.

In the future, as mentioned by limitations, we will investigate the front end to further improve the edge identification in the TLF. The inclusion of the topological loop closure in the TLF is another possibility since it is much more robust against the environmental change.

Acknowledgements This work was supported by the National Nature Science Foundation of China (Grant No. U1609210, No. 61473258 and No. 61621002).

References

Angeli A, Doncieux S, Meyer JA, Filliat D (2009) Visual topological slam and global localization. In: IEEE International Conference on Robotics and Automation, pp 4300–4305

Besl PJ, Mckay ND (1992) Method for registration of 3-d shapes. In: Robotics - DL tentative, pp 239–256

Blaer P, Allen P (2002) Topological mobile robot localization using fast vision techniques. In: IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA, pp 1031–1036 vol.1

Cadena C, Galvez-L, Pez D, Tardos JD, Neira J (2012) Robust place recognition with stereo sequences. Robotics, IEEE Transactions on 28(4):871–885

Carlevaris-Bianco N, Kaess M, Eustice RM (2014) Generic node removal for factor-graph slam. Robotics IEEE Transactions on 30(6):1371–1385

Choi J, Maurer M (2016) Local volumetric hybrid-map-based simultaneous localization and mapping with moving object tracking. IEEE Transactions on Intelligent Transportation Systems 17(9):2440–2455

Churchill W, Newman P (2013) Experience-based navigation for long-term localisation. International Journal of Robotics Research 32(14):1645–1661

Corcoran P, Winstanley A, Mooney P, Middleton R (2011) Background foreground segmentation for slam. IEEE Transactions on Intelligent Transportation Systems 12(4):1177–1183

Cummins M, Newman P (2008) Fab-map: Probabilistic localization and mapping in the space of appearance. International Journal of Robotics Research 27(6):647–665

Dellaert F, Kaess M (2006) Square root sam: Simultaneous localization and mapping via square root information smoothing. International Journal of Robotics Research 25(12):1181–1203

Dissanayake G, Durrant-Whyte H, Bailey T (2000) A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In: IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA, pp 1009–1014 vol.2

Eustice RM, Singh H, Leonard JJ (2006) Exactly sparse delayed-state filters for view-based slam. IEEE Transactions on Robotics 22(6):1100–1114

Fox D, Burgard W, Dellaert F, Thrun S (1999) Monte carlo localization: efficient position estimation for mobile robots. In: Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, Usa, pp 343–349

Furgale P, Barfoot TD (2010) Visual teach and repeat for long range rover autonomy. Journal of Field Robotics 27(5):534–560

Gálvez-López D, Tardos JD (2012) Bags of binary words for fast place recognition in image sequences. IEEE Transactions on Robotics 28(5):1188–1197

Geiger A, Ziegler J, Stiller C (2011) Stereoscan: Dense 3d reconstruction in real-time. In: Intelligent Vehicles Symposium (IV), 2011 IEEE, Ieee, pp 963–968

Huang GP, Mourikis AI, Roumeliotis SI (2009) A first-estimates jacobian ekf for improving slam consistency. Springer Tracts in Advanced Robotics 54:373–382

Huang GP, Mourikis AI, Roumeliotis SI (2010) Observability-based rules for designing consistent ekf slam estimators. International Journal of Robotics Research 29(5):502–528

Huang S, Dissanayake G (2007) Convergence and consistency analysis for extended kalman filter based slam. IEEE Transactions on Robotics 23(5):1036–1049

Kaess M, Ranganathan A, Dellaert F (2008) isam: Incremental smoothing and mapping. IEEE Transactions on Robotics 24(6):1365–1378

Konolige K, Bowman J, Chen JD, Mihelich P, Calonder M, Lepetit V, Fua P (2010) View-based maps. International Journal of Robotics Research 29(8):941–957

Konolige K, Marder-Eppstein E, Marthi B (2011) Navigation in hybrid metric-topological maps. In: IEEE International Conference on Robotics and Automation, pp 3041–3047

Krüsi P, Bücheler B, Pomerleau F, Schwesinger U, Siegwart R, Furgale P (2015) Lighting invariant adaptive route following using iterative closest point matching. Journal of Field Robotics 32(4):534–564

Kummerle R, Grisetti G, Strasdat H, Konolige K (2011) G 2 o: A general framework for graph optimization. In: IEEE International Conference on Robotics and Automation, pp 3607–3613

Kurt Konolige MA (2008) Frameslam: From bundle adjustment to real-time visual mapping. In: IEEE Trans. on Robotics, pp 1066–1077

Latif Y, Cadena C, Neira J (2013) Robust loop closing over time for pose graph slam. International Journal of Robotics Research 32(32):1611–1626

Lauer M, Stein D (2015) A train localization algorithm for train protection systems of the future. IEEE Transactions on Intelligent Transportation Systems 16(2):970–979

Lee GH, Fraundorfer F, Pollefeys M (2013) Robust pose-graph loop-closures with expectation-maximization. In: Ieee/rsj International Conference on Intelligent Robots and Systems, pp 556–563

Liao Y, Wang Y, Liu Y (2016) Graph regularized auto-encoders for image representation. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society PP(99)

Liu M, Pradalier C, Pomerleau F, Siegwart R (2012) The role of homing in visual topological navigation. In: Ieee/rsj International Conference on Intelligent Robots and Systems, pp 567–572

Lowry S, Snderhauf N, Newman P, Leonard JJ (2016) Visual place recognition: A survey. IEEE Transactions on Robotics 32(1):1–19

Mcdonald J, Kaess M, Cadena C, Neira J, Leonard JJ (2013) Real-time 6-dof multi-session visual slam over large-scale environments. Robotics & Autonomous Systems 61(10):1144–1158

Mcmanus C, Furgale P, Stenning B, Barfoot TD (2013) Lighting-invariant visual teach and repeat using appearance-based lidar. Journal of Field Robotics 30(2):254C287

Mcmanus C, Churchill W, Maddern W, Stewart AD (2014) Shady dealings: Robust, long-term visual localisation using illumination invariance. In: IEEE International Conference on Robotics and Automation,

pp 901–906

McManus C, Upcroft B, Newman P (2015) Learning place-dependant features for long-term vision-based localisation. Autonomous Robots 39(3):363–387

Milford MJ, Wyeth GF (2012) Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In: IEEE International Conference on Robotics and Automation, pp 1643–1649

Montemerlo M, Thrun S, Koller D, Wegbreit B (2002) Fastslam:a factored solution to the simultaneous localization and mapping problem. In: Eighteenth national conference on Artificial intelligence, pp 593–598

Mur-Artal R, Montiel JMM, Tards JD (2015) Orb-slam: A versatile and accurate monocular slam system. IEEE Transactions on Robotics 31(5):1147–1163

Newman P, Sibley G, Smith M, Cummins M, Harrison A, Mei C, Posner I, Shade R, Schroeter D, Murphy L (2009) Navigating, recognizing and describing urban spaces with vision and lasers. International Journal of Robotics Research 28(1112):1406–1433

Pascoe G, Maddern W, Stewart AD, Newman P (2015) Farlap : Fast robust localisation using appearance priors 2015

Paton M, Mactavish K, Warren M, Barfoot TD (2016) Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In: Ieee/rsj International Conference on Intelligent Robots and Systems, pp 1918–1925

Paton M, Pomerleau F, Mactavish K, Ostafew CJ, Barfoot TD (2017) Expanding the limits of visionbased localization for longterm routefollowing autonomy. Journal of Field Robotics 34

Rybski PE, Roumeliotis S, Gini M, Papanikopoulos N (2008) Appearance-based mapping using minimalistic sensor models. Autonomous Robots 24(3):229–246

Simhon S, Dudek G (1998) A global topological map formed by local metric maps. In: Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on, IEEE, vol 3, pp 1708–1714

Thrun S, Montemerlo M (2006) The graph slam algorithm with applications to large-scale mapping of urban structures. International Journal of Robotics Research 25(5):403–429

Tully S, Kantor G, Choset H (2012) A unified bayesian framework for global localization and slam in hybrid metric/topological maps. International Journal of Robotics Research 31(3):271–288

Walcott-Bryant A, Kaess M, Johannsson H, Leonard JJ (2012) Dynamic pose graph slam: Long-term mapping in low dynamic environments. In: Ieee/rsj International Conference on Intelligent Robots and Systems, pp 1871–1878

Wang Y, Xiong R, Li Q, Huang S (2013) Kullback-leibler divergence based graph pruning in robotic feature mapping. In: European Conference on Mobile Robots, pp 32–37

Wang Y, Xiong R, Huang S (2015) A pose pruning driven solution to pose feature graphslam. Advanced Robotics 29(10):1–16

Wang Y, Huang S, Xiong R, Wu J (2016) A framework for multi-session rgbd slam in low dynamic workspace environment. Caai Transactions on Intelligence Technology 1(1):90–103

Wolf DF, Sukhatme GS (2005) Mobile robot simultaneous localization and mapping in dynamic environments. Autonomous Robots 19(1):53–65