



# Multi-Session Visual SLAM for Illumination-Invariant Re-Localization in Indoor Environments

Mathieu Labbé\* and François Michaud

Department of Electrical Engineering and Computer Engineering, Interdisciplinary Institute of Technological Innovation (3IT), Université de Sherbrooke, Sherbrooke, QC, Canada

For robots navigating using only a camera, illumination changes in indoor environments can cause re-localization failures during autonomous navigation. In this paper, we present a multi-session visual SLAM approach to create a map made of multiple variations of the same locations in different illumination conditions. The multi-session map can then be used at any hour of the day for improved re-localization capability. The approach presented is independent of the visual features used, and this is demonstrated by comparing re-localization performance between multi-session maps created using the RTAB-Map library with SURF, SIFT, BRIEF, BRISK, KAZE, DAISY, and SuperPoint visual features. The approach is tested on six mapping and six localization sessions recorded at 30 min intervals during sunset using a Google Tango phone in a real apartment.

**Keywords:** localization, visual SLAM (simultaneous localization and mapping), robot vision, feature matching, mobile robotics

## OPEN ACCESS

### Edited by:

Javier Civera,  
University of Zaragoza, Spain

### Reviewed by:

Tomas Krajník,  
Czech Technical University in Prague,  
Czechia

Francisco Jesús Bonin Font,  
University of the Balearic Islands,  
Spain

### \*Correspondence:

Mathieu Labbé  
Mathieu.M.Labbe@  
USherbrooke.ca

### Specialty section:

This article was submitted to  
Robot and Machine Vision,  
a section of the journal  
Frontiers in Robotics and AI

**Received:** 26 October 2021

**Accepted:** 11 April 2022

**Published:** 16 June 2022

### Citation:

Labbé M and Michaud F (2022) Multi-Session Visual SLAM for Illumination-Invariant Re-Localization in Indoor Environments. *Front. Robot. AI* 9:801886. doi: 10.3389/frobt.2022.801886

## 1 INTRODUCTION

Visual SLAM (Simultaneous Localization and Mapping) frameworks using hand-crafted visual features work relatively well in static environments, as long as there are enough discriminating visual features with moderated lighting variations. To be illumination invariant, a trivial solution could be to switch from vision to LiDAR (Light Detection and Ranging) sensors, but compared with cameras they are often too expensive or bulky for some applications. Illumination-invariant re-localization using a conventional camera is not trivial, as visual features taken during the day under natural light conditions may look quite different than those extracted during the night under artificial light conditions. In our previous work on visual loop closure detection (Labbé and Michaud, 2013), we observed that when traversing multiple times the same area where atmospheric conditions are changing periodically (like day–night cycles), loop closures are more likely to be detected with locations of past mapping sessions that have similar illumination levels or atmospheric conditions. Based on that observation, in this paper, we present a multi-session approach to derive illumination-invariant maps using a full visual SLAM approach.

The idea of improving re-localization in illumination-changing environments by mapping multiple times the same area (Dayoub and Duckett, 2008; Churchill and Newman, 2013; Bürki et al., 2016; Mühlfellner et al., 2016; Paton et al., 2018) is often addressed by lifelong localization systems (Konolige and Bowman, 2009). A lifelong localization system would be able to adapt the map to changes in the environment to avoid degrading localization performance overtime. Doing so, there is still a risk that the robot incorrectly updates, significantly decreasing localization performance. In practice, most current navigation systems work in two phases: the SLAM phase to construct the map of the environment, then a localization-only phase when the robot navigates autonomously to

accomplish its tasks without modifying the map. In this context, a human can correct gross errors in the constructed prior to launching autonomous navigation, and at some point an SLAM phase can be reinitiated to update the map overtime. Launching manually those updates could increase the maintenance of the robots operating in highly dynamic environments (e.g., store, warehouse); thus, a lifelong localization system would be preferred. While the approach presented in this paper shares some concepts with lifelong systems, it principally targets applications using the two-phase navigation approach for environments generally static (e.g., house, office, residence for elderly people) but having large illumination variations caused by windows or artificial lights. Therefore, the main research questions this paper focuses are:

- Which visual feature is the most robust to illumination variations in indoor environments?
- How many mapping sessions are required during the SLAM phase so that robust re-localization during the localization phase is possible through day and night without having to update the map?
- To avoid having a human teleoperate a robot at different times of the day to create the multi-session map, would it be possible to acquire the consecutive maps simply by re-localizing from the map acquired in a previous session?

By addressing these questions, the main contributions of this work are: 1) an in-depth comparison of popular visual feature approaches for illumination-invariant indoor re-localization, 2) an adaptation of an open-source SLAM framework to create multi-session maps that are robust to illumination variations, and 3) guidelines to create such multi-session maps by an autonomous robot itself.

This paper is organized as follows. **Section 2** presents similar works to our multi-session map approach, which is described in **Section 3**. **Section 4** presents the comparative results between the visual feature used and the number of sessions required to expect the best re-localization performance. **Section 5** discusses limitations and possible improvements of the approach, while **Section 6** concludes this paper.

## 2 RELATED WORK

The approach presented in this paper shares some similarity with the general concept of the *Experience Map* (Churchill and Newman, 2013). An *experience* is referred to as an observation of a location at a particular time. A location can have multiple experiences describing it. New experiences of the same location are added to the experience map if re-localization fails during each traversal of the same environment. Re-localization of the current frame in the experience map is done concurrently against all experiences of a location, thus requiring multi-core CPUs to do it in real time as more and more experiences are added. To avoid examining all experiences, predicting next experiences to localize on (Linegar et al., 2015; Krajník et al., 2017b) or selecting visually similar experiences around the current location (Paton

et al., 2018) can be used to test the most likely ones based on the current state of the environment.

To avoid using multiple experiences of the same locations, SeqSLAM (Milford and Wyeth, 2012; Sünderhauf et al., 2013) matches sequences of visual frames instead of trying to re-localize robustly each individual frame against the map. The approach assumes that the robot takes relatively the same route (with the same viewpoints) at the same velocity, thus seeing the same sequence of images across time. This is a fair assumption for cars (or trains), as they are constrained to follow a lane at regular velocity. However, for indoor robots having to deal with obstacles, their path can change over time, thus not always replicating the same sequences of visual frames.

Adding more and more experiences to a map can increase its size over time, and so are computation time and memory usage. Some approaches try to limit the size of data in the map while keeping the same level of re-localization performance. In Cooc-Map (Johns and Yang, 2013), local features taken at different times of the day are quantized in both the feature and image spaces, and discriminating statistics can be generated on the co-occurrences of features. This produces a more compact map instead of having multiple images representing the same location, while still having local features to recover full motion transformation. A similar approach is done in Ranganathan et al. (2013) where a fine vocabulary method is used to cluster descriptors by tracking the corresponding 3D landmark in 2D images across multiple sequences under different illumination conditions. For feature matching with this learned vocabulary, instead of using a standard descriptor distance approach (e.g., Euclidean distance), a probability distance is evaluated to improve feature matching. In Bürki et al. (2016), a selective landmark strategy is used to reduce the data bandwidth shared across a fleet of vehicles by transferring only the minimal number of landmarks from a remote multi-session map for efficient re-localization at the time the vehicle is operating. Like in our paper but for the outdoor case, they also made a specific dataset to create incrementally a multi-session map from successive trajectories taken during sunset to capture the most illumination variations. Similarly in Mühlfellner et al. (2016), a multi-session map called the Summary Map is created from merging multiple traversals of the same areas. Halodová et al. (2019) made an extensive comparison of map management techniques that maximize re-localization performance over time while pruning past features to limit the size of the map. These last three papers are quite complementary to ours, where the same basic multi-session concept is used, but they are more focusing on strategies to reduce the multi-session map size than the choice of the best visual feature to use (which could also impact the map size). Other approaches rely on pre-processing the input images to make them illumination-invariant before feature extraction, by removing shadows (Corke et al., 2013; McManus et al., 2014) or by trying to predict them (Lowry et al., 2014). This improves feature-matching robustness in strong and changing shadows. In Li et al. (2016), the auto-exposure effect is removed using a high dynamic range (HDR) map. To increase robustness against large appearance difference between seasons, in Neubert et al. (2013), they predict how the images taken during winter would look like

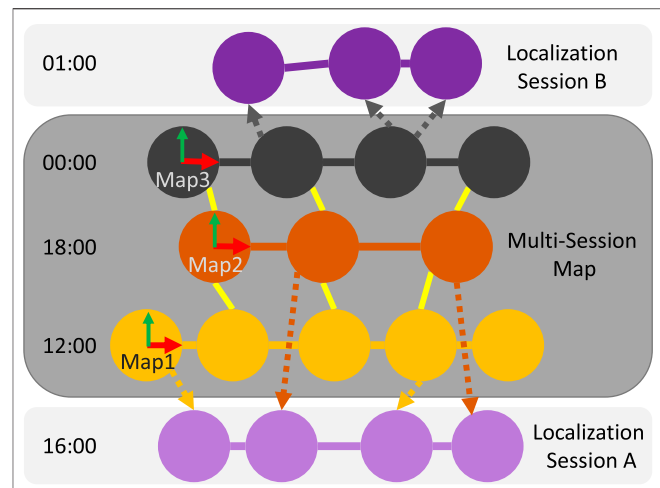
in its map taken during summer, which improves re-localization in the same area during winter (or vice versa). Most of those approaches present results on datasets recorded outdoors with a car or a train, while in this paper we present the results in an indoor setting, enhancing indoor-related works like Dayoub and Duckett (2008), Konolige and Bowman (2009), Krajník et al. (2017b) by explicitly addressing the robustness of re-localization in indoor illumination varying environments.

At the local visual feature level, common hand-crafted features such as SIFT (Lowe, 2004) and SURF (Bay et al., 2008) in outdoor experiences have been compared across multiple seasons and illumination conditions (Valgren and Lilienthal, 2007; Ross et al., 2013) to reveal some of their limitations. To overcome limitations caused by illumination variance of hand-crafted features, machine learning approaches have also been used to extract descriptors that are more illumination-invariant. In Neubert and Protzel (2015), Krajník et al. (2017a), hand-crafted features have also been compared against trained descriptors, demonstrating better place recognition performance in outdoor settings. In Carlevaris-Bianco and Eustice (2014), a neural network has been trained to track interest points in time-lapse videos so that it outputs similar descriptors for the same tracked points independently of illumination. However, only descriptors are learned, and the approach still relies on hand-crafted feature detectors. More recently, SuperPoint (DeTone et al., 2018) introduced an end-to-end local feature detection and descriptor extraction approach based on a neural network. The illumination-invariance comes from carefully making a training dataset with images showing the same visual features under large illumination variations. Other place recognition approaches using learned global descriptors exist (Sünderhauf et al., 2015; Arandjelovic et al., 2016; Sarlin et al., 2019), but this paper focuses more on the comparison of local (hand-crafted or learned) features that are generally used in classic visual SLAM pipelines.

### 3 MULTI-SESSION SLAM FOR ILLUMINATION-INVARIANT RE-LOCALIZATION

The current approach is divided into two main phases: 1) the SLAM phase to construct a multi-session map containing most illumination variations of the same locations, and 2) a localization-only phase in which the robot would navigate to do its tasks using the pre-built map. In the two phases, the same re-localization approach is used, and in the context of SLAM the first phase is also referred to as loop closure detection. This section mainly describes the multi-session SLAM phase, and the differences with the localization-only phase are described at the end of the section.

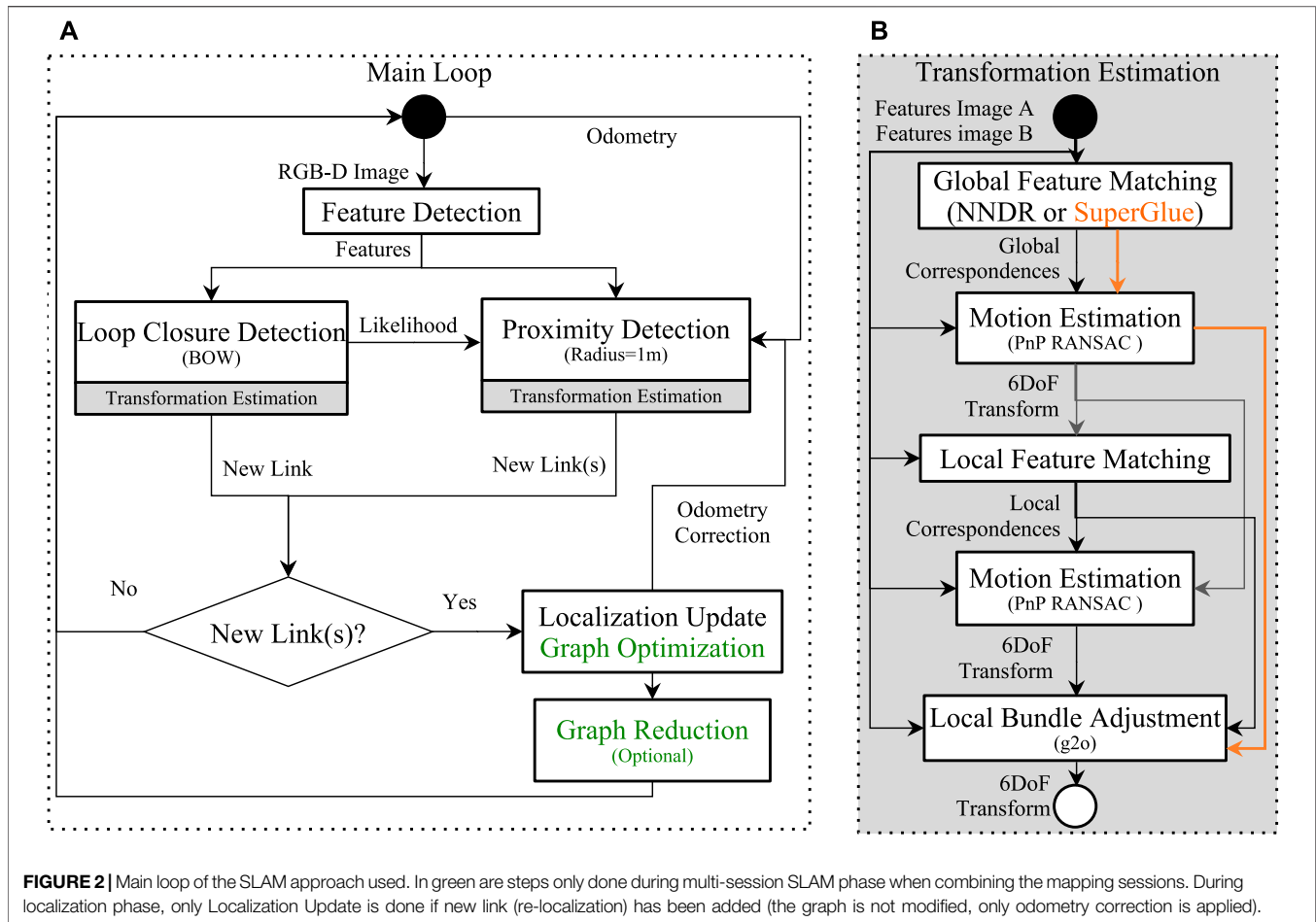
Similarly to Bürki et al. (2016) and Paton et al. (2018), one major difference of our multi-session SLAM phase and the *Experience Map* is that the interconnections of locations between the sessions are not purely topological but they also include six DoF constraints, making it possible to transform all locations in the same global coordinate frame. As the presented approach for the SLAM phase is targeting autonomous systems



**FIGURE 1** | Structure of the multi-session map. Three sessions taken at different times have been merged together during the SLAM phase by finding loop closures between them (yellow links). Each map has its own coordinate frame. During the localization-only phase, Localization Session A (16:00) is re-localized in relation to both day sessions in the map (12:00 and 18:00), and Localization Session B (01:00) is only re-localized on the night session (00:00).

that will capture by themselves the different illumination conditions of the same environment instead of having a person teleoperating or driving the robot many times, it is preferable for the navigation system that the robot can always be localized in the same coordinate frame. When creating the multi-session map, the mapping sessions should have enough similar illumination conditions from a previous session to follow correctly the original trajectory (in the coordinate frame of the first session), but experience sufficient illumination variations to add new locations. As robots do not have infinite memory, the number of duplicated locations in the map should be also minimized while ideally achieving the same re-localization performance.

The visual feature chosen can have an impact on the final size of the multi-session map depending on how much they are illumination invariant or not. Some visual features are fast to compute and are light in memory, but a lot of them would be required to represent the different illumination states of the environment. Other visual features are more robust to illumination variation while being heavier in computation and memory, but less of them would be required to capture all variations of the environment. Therefore, the choice of visual features may impact how many sessions are required to achieve similar re-localization performance. To evaluate this, our approach is designed to be independent of the visual features used, which can be hand-crafted or neural network based, while being integrated in full SLAM conditions using for instance a library like RTAB-Map. RTAB-Map (Labbé and Michaud, 2019) is a Graph-SLAM (Grisetti et al., 2010) approach that can be used with camera(s) and/or with a LiDAR. This paper focuses on the first case where only a camera is available for re-localization. The structure of the map is a pose-graph with nodes representing each



**FIGURE 2** | Main loop of the SLAM approach used. In green are steps only done during multi-session SLAM phase when combining the mapping sessions. During localization phase, only Localization Update is done if new link (re-localization) has been added (the graph is not modified, only odometry correction is applied).

image acquired at a fixed rate, and links representing the six DoF transformations between them. **Figure 1** presents an example of the resulting multi-session map created during the SLAM phase from three sessions taken at three different hours (12:00, 18:00, and 00:00). Two additional localization sessions are also shown, one during the day (16:00) and one at night (01:00), which represent two examples that would be conducted during the localization-only phase. The dotted links represent to which nodes in the graph the corresponding frame has been re-localized on. The goal is to have new frames re-localizing on nodes taken at a similar time, and if localization time falls between two mapping times, re-localization could jump between two sessions or more inside the multi-session map. Inside the multi-session map, each individual map is transformed in the same global coordinate frame (Map1 in this example) so that when the robot re-localizes on a node of a different session, it does not jump between different coordinate frames.

**Figure 2** presents the main loop of the SLAM algorithm used during the SLAM phase, which can be done online or offline. After a new frame and its pose are received, visual features are extracted from the RGB image with their 3D positions estimated using the depth image and camera calibration. Visual features can be any of the ones implemented in OpenCV (Bradski and Kaehler, 2008), which are SURF (Bay et al., 2008), SIFT

(Lowe, 2004), BRIEF (Calonder et al., 2010), BRISK (Leutenegger et al., 2011), KAZE (Alcantarilla et al., 2012), and DAISY (Tola et al., 2009). The SuperPoint (DeTone et al., 2018) neural network-based feature has also been integrated for comparison.

Two methods are used to find loop closures: a global one called Loop Closure Detection (LCD), and a local one called Proximity Detection (PD). LCD is not limited to only nodes of the current mapping session, but it also includes all nodes from all past sessions when updating its loop closure hypotheses. This makes the approach able to seamlessly find constraints between sessions that are used to merge multiple sessions together during the Graph Optimization step. The bag-of-words (BOW) approach (Sivic and Zisserman, 2003) is used to evaluate loop closure hypotheses over all previous images from all sessions, independently of the odometry estimate. The BOW vocabulary used in this paper is incremental based on FLANN's KD-Trees (Muja and Lowe, 2009), and quantization of features to visual words is done using the Nearest Neighbor Distance Ratio (NNDR) approach (Lowe, 2004). After quantization of the features of the current frame into the vocabulary, BOW uses an inverted index voting-scheme to retrieve past images with the same visual words, to significantly reduce the likelihood estimation time with all previous images. The likelihood is

then fed to a Bayes filter to estimate loop closure hypotheses (Labbé and Michaud, 2013). The Bayes filter helps filter spurious wrong likelihood (because of noise), so that a node in the map should score high in likelihood on many consecutive frames for its hypothesis to grow. When a loop closure hypothesis reaches a predefined threshold, a loop closure is detected. In contrast to LCD, PD looks for nodes around the current position of the robot for possible loop closures based on the current odometry estimate. Nodes of the map's graph inside a fixed radius of the current position are then selected as candidates for proximity detection. In our previous work (Labbé and Michaud, 2017), proximity detection was introduced primarily in RTAB-Map for LiDAR rotational invariant re-localization. A slight modification is made for this work to use it with a camera. Previously, the closest nodes in a fixed radius around the current position of the robot were sorted by distance, then PD was used against the top three closest nodes, adding the same number of constraints to the graph if all of them are accepted. However, in a visual multi-session map, the closest nodes may not have images with the most similar illumination conditions than the current one. Similar to BOW selection in Paton et al. (2018), by using the likelihood computed during LCD, nodes inside the proximity radius are sorted from the most to less visually similar (in terms of BOW's inverted index score). Visual PD is then done using the three most similar images around the current position in a fixed radius. If PD fails because the robot's odometry drifted too much since the last re-localization, LCD is still done in parallel to re-localize the robot when it is lost.

For both loop closures and proximity detections, six DoF transformations are computed following the steps of Transformation Estimation (TE) of **Figure 2**. A global feature matching is done using a nearest-neighbor (NN) approach with feature descriptors between the corresponding frames. With the feature correspondences, a first transformation between the frames is computed using the Perspective-n-Point (PnP RANSAC) approach (Bradski and Kaehler, 2008). Using that previous transformation as a motion estimate, the 3D features from the first frame are then projected into the second frame for local feature matching using a fixed-size window. This second step generates better matches to compute a more accurate transformation using PnP. Depicted by the orange arrows in **Figure 2**, if the visual feature type used is SuperPoint, the SuperGlue approach (Sarlin et al., 2020) can be optionally used for global feature matching. SuperGlue uses a neural network trained to find correspondences between SuperPoint features, generating more correspondences than the classic NNDR approach. In that case, the second local feature matching step along with the second motion estimation step is skipped. For both approaches, the resulting transform is further refined using a local bundle adjustment approach (Kummerle et al., 2011).

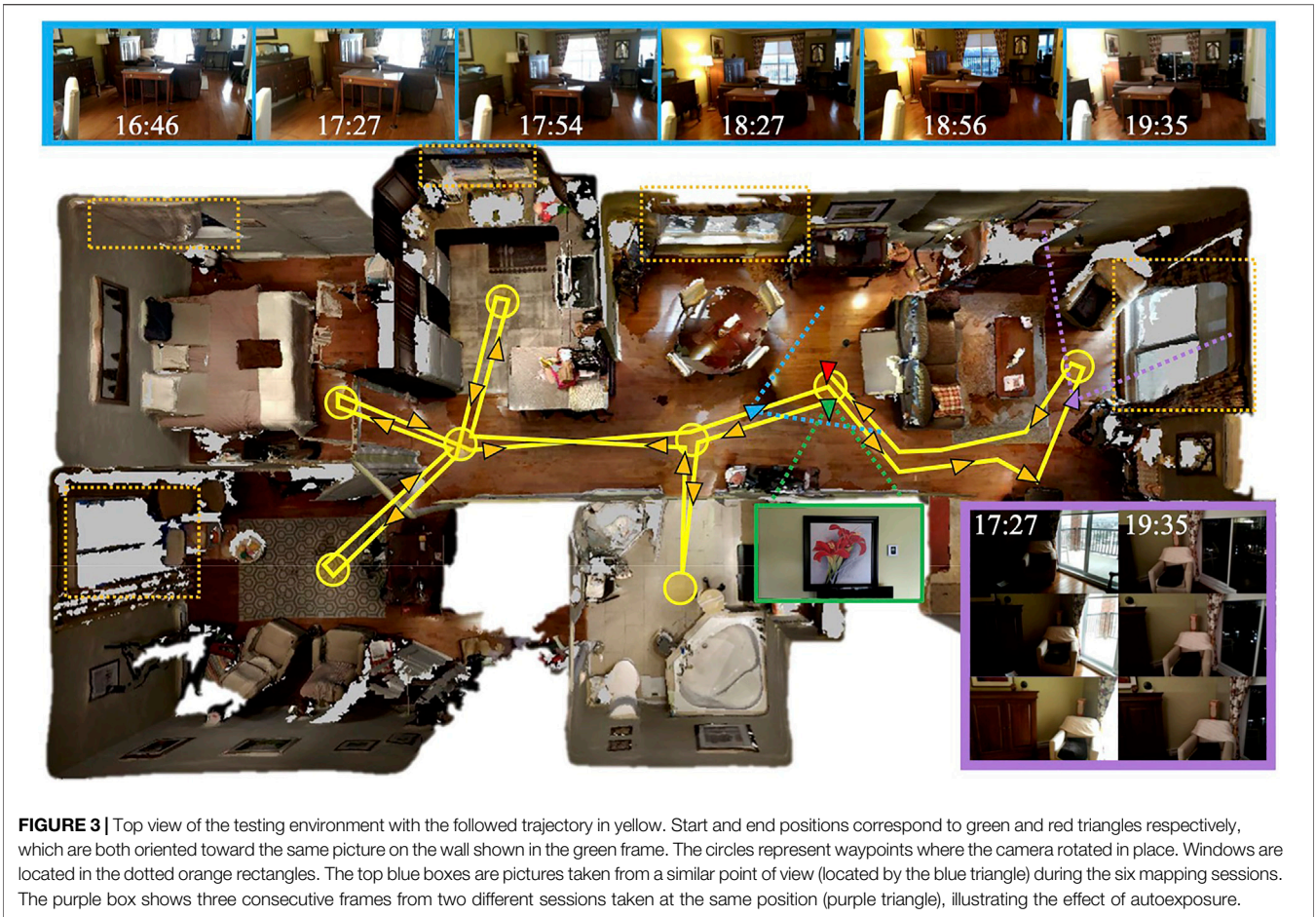
When loop closures are detected, the pose-graph is optimized using GTSAM (Dellaert, 2012) with the new constraints, implicitly transforming all sessions into the same coordinate frame as long as there is at least one loop closure between the sessions. This means that when a loop closure happens for the first time with an older session, the whole current map is

automatically transformed in the coordinate frame of the oldest map. This may cause large re-localization jumps when these events happen. However, once the maps are merged, the next re-localization jumps should be proportional to odometry drift and how long the robot has not been re-localized. Finally, a Graph Reduction (GR) approach can be used to reduce the size of the map when loop closures have been previously added to the graph, thus reducing memory usage of the algorithm. This process is explained in detail in Labbé and Michaud (2017) and is similar to the approach in Churchill and Newman (2013) where if re-localization is successful, no new experiences are added. In summary, a node having a loop closure with an older node can be removed by merging the loop closure links to its neighbor nodes, thus keeping the graph at the same size when new data are acquired as long as there are loop closures. The graph will increase in size only when a loop closure has not been detected (e.g., the location has changed too much or a new area is visited).

After the multi-session map is created, the localization-only phase is done following the same main loop than **Figure 2**, but without the green steps (the pose-graph is not modified). Another difference is that to limit processing time, when estimating transformations of the top three identified nodes from LCD and PD, as soon as a first transformation is accepted the others are not tested. In the next section, both LCD and PD are referred to as re-localization during the localization-only phase.

## 4 RESULTS

To address the three research questions presented in **Section 1**, a dataset has been recorded before and after sunset to capture the full spectrum of illumination variations between the day and night. **Figure 3** illustrates how the dataset has been acquired in a home in Sherbrooke, Quebec in March 2019. An ASUS Zenfone AR phone (with Google Tango technology) running the RTAB-Map Tango App has been used to record data for each session following the same yellow trajectory, similarly to what a robot would do patrolling the environment. The poses are estimated using Google Tango's visual inertial odometry approach, with RGB and registered depth images recorded at 1 Hz. To be able to combine offline the maps into multi-session maps, as described in **Section 4.2**, the trajectory started and finished in front of a highly visual descriptive location (i.e., first and last positions shown as green and red arrows, respectively) to make sure that each consecutive mapping session is able to re-localize on start from the previous session. Note that this assumption could be also valid for a robot by placing a highly discriminating sign visible at its docking station. This ensures that all maps are transformed in the same coordinate frame of the first map after graph optimization. Between 16:45 (daylight) and 19:45 (nighttime), two mapping sessions were recorded back-to-back to get a mapping session and a localization session taken roughly at the same time. The time delay between each mapping session is around 30 min. Overall, the resulting dataset has six mapping sessions (Numerical Index-Time: 1-16:46, 2-17:27, 3-17:54, 4-18:



**FIGURE 3 |** Top view of the testing environment with the followed trajectory in yellow. Start and end positions correspond to green and red triangles respectively, which are both oriented toward the same picture on the wall shown in the green frame. The circles represent waypoints where the camera rotated in place. Windows are located in the dotted orange rectangles. The top blue boxes are pictures taken from a similar point of view (located by the blue triangle) during the six mapping sessions. The purple box shows three consecutive frames from two different sessions taken at the same position (purple triangle), illustrating the effect of autoexposure.

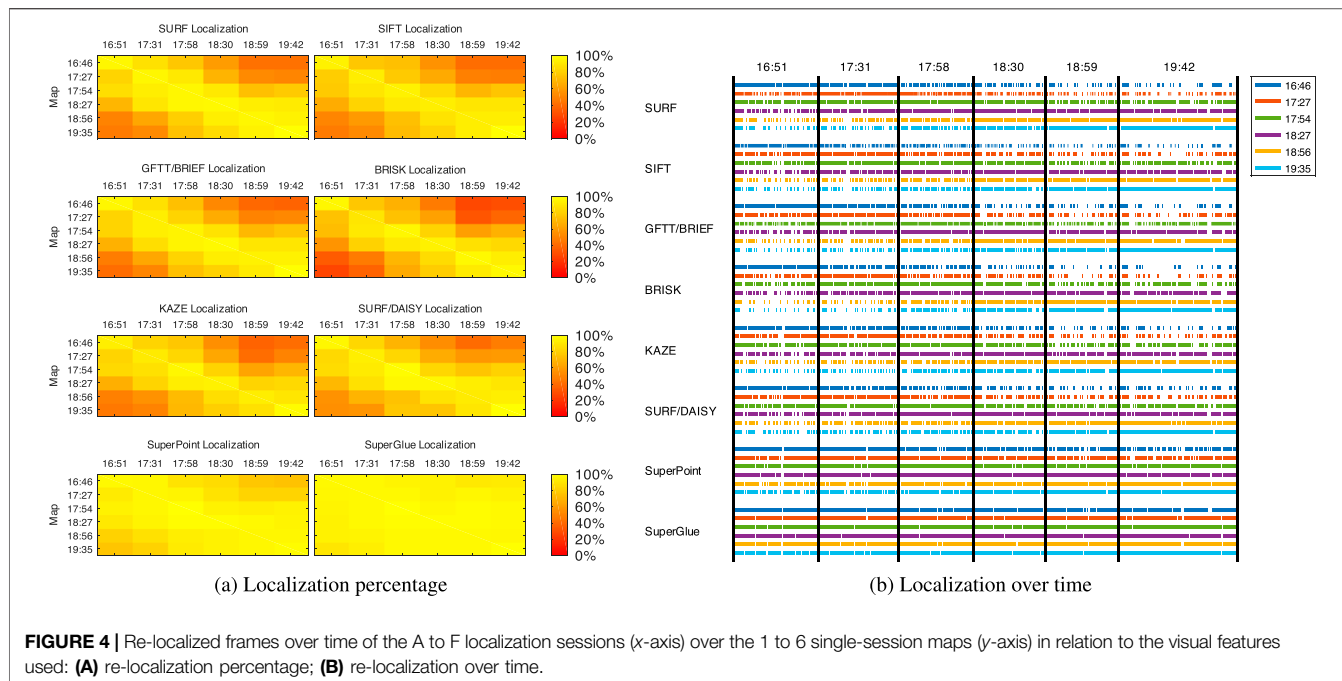
**TABLE 1 |** RTAB-Map's parameters.

Name	Description	Value
Kp/DetectorStrategy	Feature detector	Variable
Kp/MaxFeatures	Maximum visual words per frame	400
Vis/CorGuessWinSize	Local feature matching window size	40 pix
Vis/CorNNDR	NNDR for binary features	0.8
Vis/CorNNDR	NNDR for float features	0.6
Vis/CorNNTtype	Feature matching approach	1 (NN) or 6 (SG)
Vis/FeatureType	Feature detector	Variable
Vis/MaxFeatures	Maximum visual features per frame	1,000
Vis/MinInliers	Minimum PnP inliers	20
Reg/RepeatOnce	Second local feature matching	true (false with SG)
RGBD/LocalRadius	Proximity detection local radius	1 m
RGBD/ProximityMaxPaths	Maximum nodes tested by proximity detection	3
Rtabmap/LoopThr	Loop closure detection threshold	0.11

27, 5-18:56, 6-19:35) and six localization sessions (Alphabetical Index-Time: A-16:51, B-17:31, C-17:58, D-18:30, E-18:59, F-19:42).

The top blue boxes of **Figure 3** show the images of the same location taken during each mapping session. To evaluate the influence of natural light coming from the windows during the day, all lights in the apartment were on during all sessions except for the one in the living room that we turned on when the room

was getting darker (see the top images at 17:54 and 18:27). Besides natural illumination changing over the sessions, the RGB camera had auto-exposure and auto-white balance enabled (which could not be disabled by Google Tango API on that phone), causing additional illumination changes depending on where the camera was pointing, as shown in the purple box of **Figure 3**. The left sequence (17:27) illustrates what happened when greater lighting comes from outside, with auto-exposure making the inside very



dark when the camera passed by the window. In comparison, doing so at night (shown in the right sequence 19:35) did not result in any changes. Therefore, for this dataset, most illumination changes are coming either from natural lighting or auto-exposure variations.

For the implementation, OpenCV 4.2.0 and RTAB-Map 0.20.15 have been used. **Table 1** presents RTAB-Map's parameters used. Note that "Kp/MaxFeatures" parameter means that only 400 features of the 1,000 extracted from each frame ("Vis/MaxFeatures") with highest response are quantized to BOW vocabulary, to limit vocabulary size over time. Experimentally, we found that "Vis/CorNNDR = 0.6" works better when features are more discriminative (i.e., have float descriptors) and set to 0.8 for binary features. The feature detector value can be SURF (SU), SIFT (SI), BRIEF (BF), BRISK (BK), KAZE (KA), DAISY (DY), and SuperPoint (SP). The SuperPoint variant with SuperGlue feature matching is named SG. Note that results using other binary features available in OpenCV such as ORB Rublee et al. (2011) and FREAK Alahi et al. (2012) are very similar to BRIEF in terms of processing time, memory, and re-localization performance, and therefore only BRIEF results are presented in this paper.

To evaluate re-localization performance, the metric used is the percentage of frames that are re-localized during the localization phase, i.e., the number of frames correctly re-localized on the total number of frames in a localization session. For example, if the localization session has 300 frames and only 200 frames are re-localized, localization performance is 66%. A correct re-localization means that the localized frame represents the same real location than the corresponding frame in the map. In all our experiments below, no wrong re-localized frames were accepted by the algorithm, because similarity was insufficient to trigger a re-localization (LCD hypotheses < Rtabmap/LoopThr

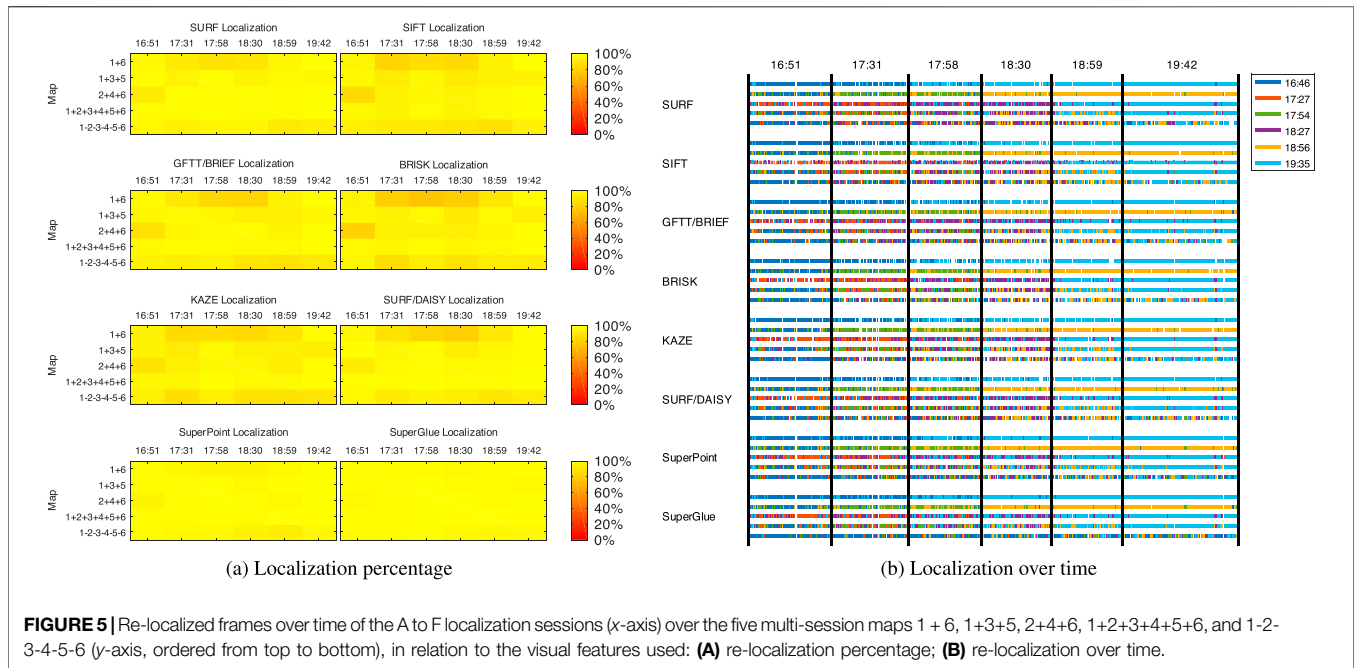
or that TE rejected them because of lack of visual inliers ( $< \text{Vis}/\text{MinInliers}$ ).

#### 4.1 Single-Session Re-Localization

The first experiment done with this dataset examines re-localization performance of different visual features for a single mapping session, thus establishing our baseline performance. **Figure 4A** shows the percentage of frames re-localized of the localization sessions (A to F) over the mapping sessions (1–6) independently, for each visual feature listed in **Section 3**. **Figure 4B** shows more precisely when every frame has been re-localized on each mapping session, thus visualizing the distribution of the re-localization. As expected by looking at the diagonals, re-localization performance is best (and with less gaps) when re-localization is done using a map taken at the same time of the day (i.e., with very similar illumination condition). In contrast, re-localization performance is worst when re-localizing at night using a map taken on the day, and vice versa. SuperPoint (with or without SuperGlue) is the most robust descriptor to large illumination variations, while binary descriptors such as BRIEF and BRISK are the most sensitive.

#### 4.2 Multi-Session Re-Localization

The second experiment evaluates re-localization performance using multi-session maps created using maps generated at different times from the six mapping sessions. To create different combinations of multi-session maps from the six individual map sessions recorded, the selected individual maps are replayed back to back offline as input streams to a new SLAM process. This new SLAM process can detect the transition between input maps to internally create a new session. Because all mapping sessions started in front of the same



**TABLE 2** | Cumulative re-localization performance (%) and average re-localization jumps (mm) of the six localization sessions on each map for each visual feature used.

Map	Re-localization (%)								Visual inliers (%)								Jumps (mm)							
	SU	SI	BF	BK	KA	DY	SP	SG	SU	SI	BF	BK	KA	DY	SP	SG	SU	SI	BF	BK	KA	DY	SP	SG
1	68	65	65	58	65	66	87	94	11	10	15	11	12	11	19	23	58	56	53	60	54	45	39	41
2	74	69	71	62	69	74	89	96	11	10	14	10	12	10	19	23	41	38	41	46	42	37	36	36
3	86	82	84	78	81	86	95	98	13	11	16	11	14	13	23	27	43	43	42	48	50	41	33	32
4	89	85	87	82	84	88	96	98	13	12	16	11	15	13	23	29	38	48	42	44	41	36	31	31
5	80	76	79	71	7	79	91	96	13	12	16	12	15	13	22	27	39	45	46	49	46	39	34	36
6	76	75	74	67	72	76	90	96	13	12	17	12	15	13	22	26	41	46	44	45	43	37	36	36
1+6	94	91	92	87	90	90	97	<b>99</b>	14	13	17	12	17	14	25	30	38	37	38	46	36	37	29	31
1+3+5	97	95	96	95	94	97	<b>99</b>	<b>99</b>	16	14	18	13	18	16	27	32	31	57	37	44	42	32	29	28
2+4+6	98	95	96	93	95	97	98	<b>99</b>	16	14	19	13	18	16	27	32	32	37	38	38	38	31	27	27
1+2+3+4+5+6	<b>99</b>	<b>97</b>	<b>98</b>	<b>98</b>	<b>98</b>	<b>99</b>	<b>99</b>	<b>99</b>	<b>18</b>	<b>16</b>	<b>20</b>	<b>14</b>	<b>20</b>	<b>18</b>	<b>29</b>	<b>34</b>	<b>26</b>	<b>29</b>	<b>34</b>	<b>37</b>	<b>32</b>	<b>28</b>	<b>25</b>	<b>24</b>
1-2-3-4-5-6	94	91	92	91	90	93	96	98	12	12	16	11	14	12	21	23	43	45	51	52	48	43	40	44
1 2 3 4 5 6	97	95	96	95	95	96	98	98	17	14	19	<b>14</b>	18	16	27	31	27	32	32	<b>35</b>	<b>31</b>	<b>28</b>	26	27

In bold are the numbers corresponding to the map with best performance for each visual feature.

highly visual descriptive location, LCD can detect a loop closure with the previous session to merge, through graph optimization, the internal sessions in the same global graph. As more input data are streamed to the new SLAM process, more loop closures are detected between and inside sessions. Different combinations of multi-session maps are tested: 1+6 combines the two mapping sessions with the highest illumination variations (time 16:46 with time 19:35); 1+3+5 and 2+4+6 are multi-session maps generated by assuming that mapping would occur every hour, and 1+2+3+4+5+6 is the combination of all maps taken at 30 min interval. Those multi-session maps have been merged without graph reduction, thus keeping all nodes of all sessions. For comparison, the multi-session map 1-2-3-4-5-6 represents assembled maps with graph reduction enabled. **Figure 5** illustrates, for each multi-session map, the resulting re-localization performance and re-localized frames over time. Except for SuperPoint (with and without SuperGlue) which

shows similar high performance for all multi-session maps, merging more sessions with different illumination conditions increases re-localization performance for all visual features, with best performance using the 1+2+3+4+5+6 multi-session map and with 1-2-3-4-5-6 not far behind. Note that for those multi-session maps, **Figure 5B** also shows by color to which map in the tested multi-session map a frame has been re-localized. For example, for 2+4+6 and 1+2+3+4+5+6 maps (third and fourth lines), most re-localizations after 19:42 have been on frames added from Map 6 (19:35). For the reduced map 1-2-3-4-5-6 (last line), re-localizations are more distributed against all mapping sessions for every localization session.

**Table 2** presents cumulative re-localization performance over single session and multi-session maps for each visual feature. As expected, multi-session maps improve re-localization performance. The map 1|2|3|4|5|6 corresponds to the case when only the session map taken at the closest time of each



**TABLE 3** | Graph size and RAM usage (MB) computed using Valgrind's Massif tool of each map for each visual feature used.

Map	Nodes	SU	SI	BF	BK	KA	DY	SP	SG
1	202	104	167	64	80	91	233	205	205
2	201	101	162	64	79	83	227	189	189
3	204	103	165	64	80	87	231	195	195
4	191	94	151	60	74	80	211	183	183
5	175	85	134	55	67	71	189	161	161
6	234	111	176	71	86	96	251	200	200
1+6	436	215	343	134	166	186	483	404	404
1+3+5	581	288	457	179	223	244	644	542	542
2+4+6	626	303	481	191	236	253	681	559	559
1+2+3+4+5+6	1,207	583	924	367	454	489	1,303	1,077	1,077
1-2-3-4-5-6	<b>Table 4</b>	176	308	128	169	162	404	252	184
Constant overhead		90	90	90	225	90	155	1,445	1,445

localization session is used (e.g., the cumulative performance of the diagonal results in **Figure 4A**). While this seems to result in similar performance compared with the multi-session cases, this could be difficult to implement robustly over multiple seasons (where general illumination variations would not always happen at the same time) and during weather changes (cloudy, sunny, or rainy days would result in changes in illumination conditions for the same time of day). Another challenge is how to make sure that maps remain correctly aligned together in the same coordinate frame of the original map and also during the whole trajectory. A global localization drift could then happen over time, which is referred to as the “photocopy” of a “photocopy” effect (Halodová et al., 2019). In contrast, with the multi-session approach, the selection of which mapping session to use is done implicitly by selecting the best candidates of loop closure detection across all sessions. There is therefore no need to have a priori knowledge of the illumination conditions before doing re-localization. All sessions are also correctly aligned with regard to the origin of the original map.

Even if we did not have access to ground truth data recorded with an external global localization system, the odometry from Google Tango for this kind of trajectory and environment does not drift very much. Thus, evaluating re-localization jumps caused by odometry correction can provide an estimate of re-localization accuracy. The last columns of **Table 2** present the average distance of the jumps occurring during localization. The maps 1+2+3+4+5+6 and 1|2|3|4|5|6 produce the smallest jumps, which can be explained by three factors: 1) the high number of visual inliers (middle columns) when computing the transformation between two frames; 2) smaller gaps between re-localized frames (that would increase odometry drift); and 3) the presence of more frames with the same illumination level in the map. With these two maps, re-localization frames can be matched with map frames taken roughly at the same time, thus giving more and better inliers.

Regarding computation resources, the multi-session approach requires more memory usage, as the map is at most six times larger in our experiment than a single session of the same environment if graph reduction is not applied. **Table 3** presents the memory usage (RAM) required for re-localization using the different maps configurations, along with the constant

**TABLE 4** | Graph size for the 1-2-3-4-5-6 multi-session map for each visual feature used and the percentage of nodes removed in comparison with the 1+2+3+4+5+6 map.

	SU	SI	BF	BK	KA	DY	SP	SG
Nodes	349	395	401	426	395	362	272	198
Reduction	71%	67%	67%	65%	67%	70%	77%	84%

RAM overhead (e.g., loading libraries and feature detector initialization) shown separately at the bottom. Graph reduction with SuperPoint is higher than other features (see **Table 4**), which can be explained by being the most illumination-invariant feature, causing more nodes to be reduced. With SuperGlue, as more feature correspondences can be found between frames, thus accepting more re-localizations, the reduction is higher and the final map size is even smaller than most individual map sessions. From the 198 nodes remaining in the final reduced map, 144 nodes are coming from Map 1, 9 from Map 2, 13 from Map 3, 8 from Map 4, 12 from Map 5, and 11 from Map 6. **Table 5** presents the average re-localization time (on an Intel Core i7-9750H CPU and a GeForce GTX 1650 GPU for SuperPoint and SuperGlue). Feature detection time depends only on the feature type, and with all maps, this is what takes the most processing time per frame. TE time is also independent of the map size, but dependent on the number of features extracted per frame. Using BOW's inverted index search, loop closure detection computation does not require significantly more time to process for multi-session maps (at most +4 ms for graph and vocabulary six times larger) than for single-session maps. However, the multi-session maps require more memory, which could be a problem on small robots with limited RAM. With graph reduction, memory usage can be reduced to a level between single-session and two-session maps. Comparing the visual features used, BRIEF requires the least processing time and memory. Even if it generates the most features per frame and a larger vocabulary, its descriptor is so small that less RAM is used. TE time is also the lowest with SuperPoint, as less features are extracted per frame. However, it requires significantly more memory (even more than multi-session maps of other features

**TABLE 5 |** Average re-localization time and features per frame for each visual feature used, along with descriptor dimension and number of bytes per element in the descriptor.

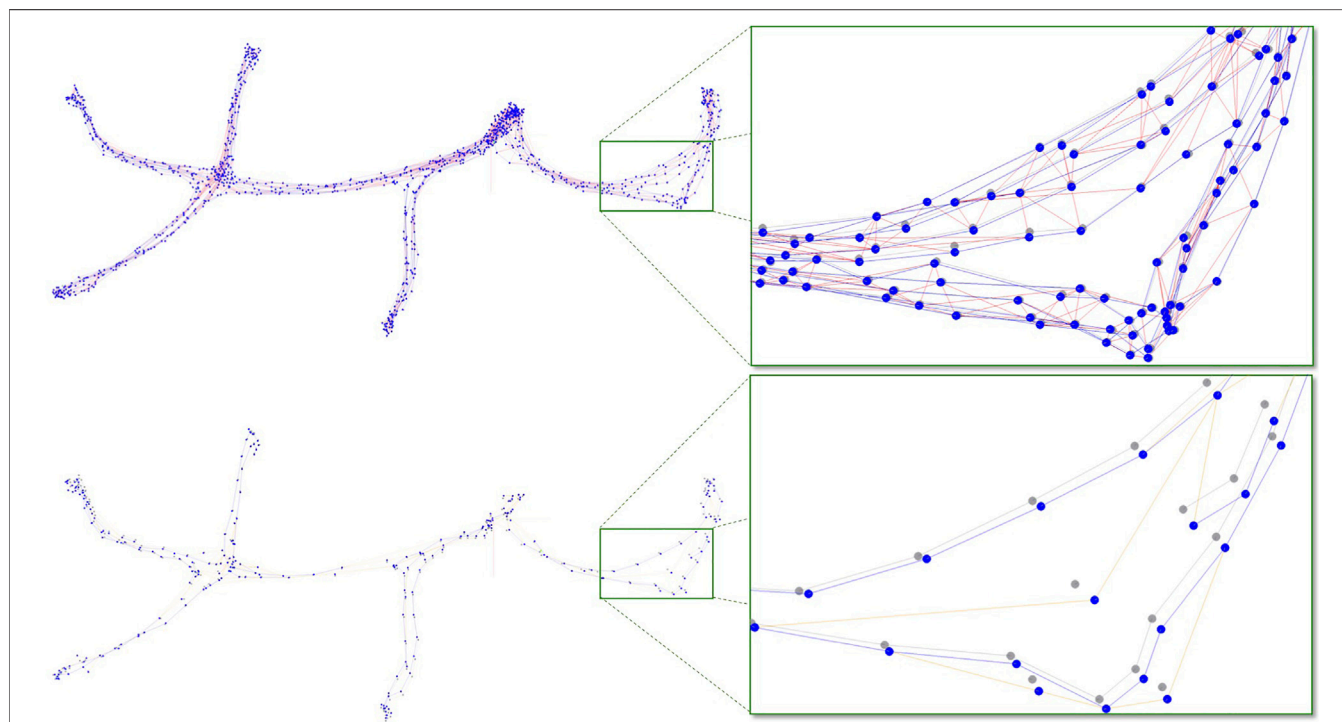
		SU	SI	BF	BK	KA	DY	SP	SG
Feature detection (ms)	-	39	152	15	332	397	89	85	85
Loop closure detection (ms)	with single-session maps	7	8	7	8	7	8	8	8
	with 1+2+3+4+5+6 map	11	11	11	12	10	11	11	11
	with 1-2-3-4-5-6 map	8	8	8	9	7	9	8	8
Transformation est. (ms)	-	35	30	37	36	26	37	24	64
Features/frame	-	847	770	889	939	640	847	472	472
Vocabulary size ( $\times 10^3$ )	with single-session maps	42	46	59	58	40	39	35	35
	with 1+2+3+4+5+6 map	231	248	331	327	214	209	181	181
	with 1-2-3-4-5-6 map	78	92	119	127	78	75	49	35
Descriptor dimension	-	64	128	32	64	64	200	256	256
Descriptor bytes/elem	-	4	4	1	1	4	4	4	4

**TABLE 6 |** ATE (mm) comparison with and without graph reduction.

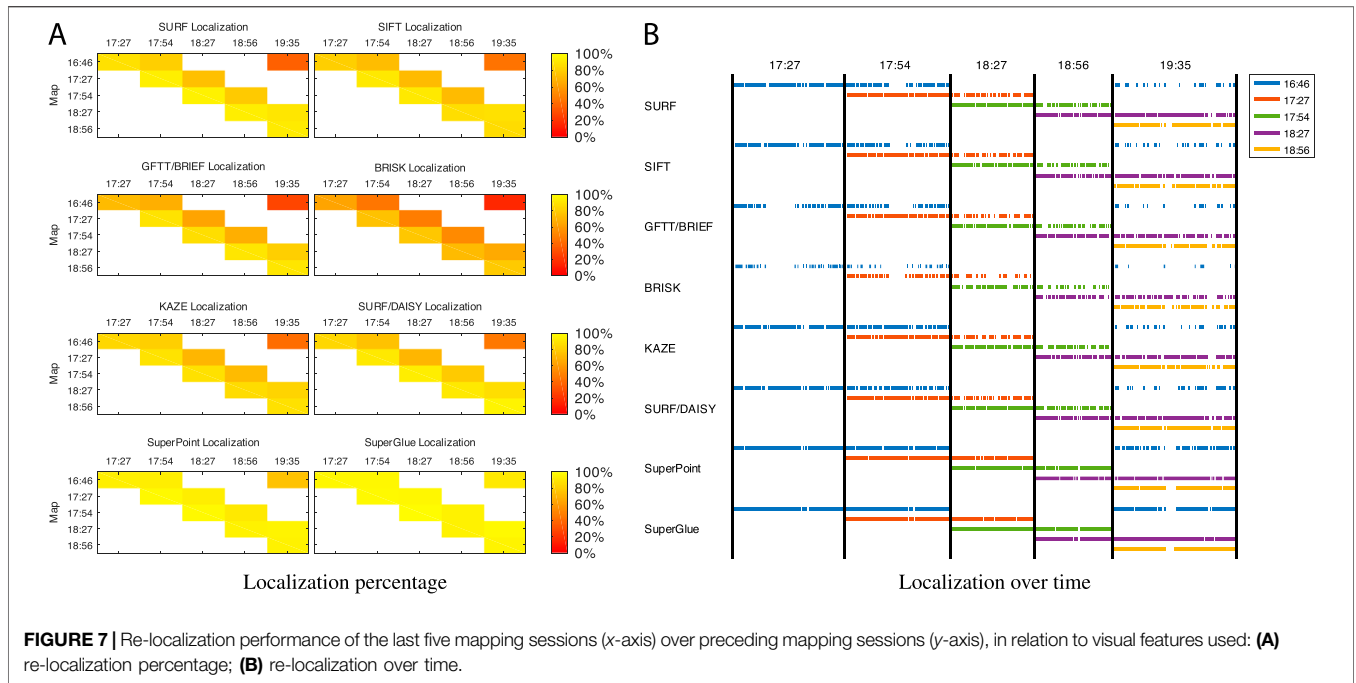
Map	SU	SI	BF	BK	KA	DY	SP	SG
1+2+3+4+5+6	17	21	17	20	25	19	13	13
1-2-3-4-5-6	41	63	68	64	48	57	49	56

without graph reduction) because of its high dimensional descriptor and a large NVidia’s CUDA library overhead in RAM. SuperGlue adds a 40 ms overhead on TE when used.

As shown in Tables 2, 3, re-localization performance for hand-crafted features with graph reduction (1-2-3-4-5-6 map) is better with less nodes than on single and 1+6 maps, but is lower than on 1+3+5, 2+4+6, and 1+2+3+4+5+6 maps. However, there is significantly less memory used when graph reduction is enabled. Another observation is that the average re-localization jumps are higher on 1-2-3-4-5-6 than with other multi-session maps. A first reason is that with graph reduction, the number of visual inliers is lower (at similar level than with single maps) because there are less frames that would have exactly



**FIGURE 6 |** Comparison of the multi-session maps 1+2+3+4+5+6 (top) and 1-2-3-4-5-6 (down) with SuperPoint feature. On the right are the zoomed parts of the corresponding rectangles on the left. Gray nodes correspond to what is considered to be the ground truth. Loop closure and odometry links are shown in blue and red, respectively. Orange links are created when reducing the graph (loop closure links propagated to neighbor nodes when a node is removed).



the same illumination level than the frame to re-localize. Another reason is that maps with graph reduction would be less correctly optimized (i.e., not representing as well the environment than other multi-session maps), as there are less constraints in the graph. Without graph reduction, more odometry links are kept in the graph (VIO generates more accurate transforms between frames than re-localization using only RGB-D data); thus, the map would be better optimized. To test this hypothesis, as ground truth is not available for this dataset, the map 1+2+3+4+5+6 has been reprocessed offline to add more links between all sessions. For each node in the graph, the closest node not already linked to it is tested with the TE approach. If TE is accepted, a new loop closure is added to the graph. This whole process is repeated five times with all nodes in the map. The resulting map is then expected to be even closer to a real ground truth because of the added constraints. To make sure of this, the generated dense point cloud is inspected qualitatively to validate that there are no double surfaces or objects. **Table 6** shows the absolute trajectory error (ATE) (Sturm et al., 2012) results with and without graph reduction. The ATE is smaller on the maps without graph reduction because all constraints of all sessions are kept. **Figure 6** illustrates the error by superposing the optimized poses (blue nodes) on the ground truth poses (gray nodes). With graph reduction, the blue and the corresponding gray nodes are less overlapping, meaning that the final optimized graph represents less well the environment, thus higher re-localization jumps would be expected, as observed in **Table 2**.

### 4.3 Consecutive Session Re-Localization

The results presented in **Section 4.2** suggest that the best re-localization results are when using the six mapping sessions merged together. Having six maps to record before doing navigation can be a tedious task if an operator has to

**TABLE 7 |** Maximum distance (m) traveled while not being re-localized and the percentage of frames re-localized under 55 cm of the last re-localization.

Interval Between Maps (min)	SU	SI	BF	BK	KA	DY	SP	SG
30	1.45 93%	1.45 89%	2.94 90%	1.92 92%	1.72 90%	1.58 92%	0.98 95%	0.84 97%
60	4.50 78%	3.59 74%	4.68 73%	4.68 68%	4.65 74%	4.65 74%	1.38 90%	1.11 95%
120	4.50 29%	4.75 34%	5.82 28%	5.82 20%	4.79 38%	4.65 34%	2.74 72%	1.11 87%

teleoperate the robot many times and at the right time. It would be better to “teach” once the trajectory to follow and have the robot repeat the process autonomously for the mapping sessions. The problem is if the robot cannot re-localize robustly on its previous trajectory, it may not be able to reproduce it completely, thus failing at capturing the required data. **Figure 7** shows the re-localization performance using a previous mapping session. The diagonal values represent the case when localization occurs every 30 min using the previous map. Results just over the main diagonal are if localization is done each hour using a map taken 1 hour ago (e.g., for the 1+3+5 and 2+4+6 multi-session cases). The top-right lines are for the 1+6 multi-session case during which the robot would be activated only at night while trying to re-localize using the map learned during the day. Having low re-localization performance is not that bad, but re-localizations should be evenly distributed; otherwise, the robot may get lost before being able to re-localize after having to navigate using dead-reckoning over a small distance. The

maximum distance that the robot can robustly recover from depends on the odometry drift: if high, frequent re-localizations would be required to correctly follow the planned path. Looking at **Figure 7B**, SURF, SIFT, KAZE, DAISY, and SuperPoint (with or without SuperGlue) are features that do not give large gaps if maps are taken 30 min after the other. For maps taken 1 h after the other, only KAZE, SuperPoint, and DAISY do not show large gaps. Finally, SuperPoint may be the only one that could be used to only map the environment twice (e.g., one at day and one at night) and re-localize robustly using the first map. **Table 7** shows the largest distance (gap) in meters that the robot would have traveled on dead-reckoning in **Figure 7B**, depending on if the maps were taken 30, 60, or 120 min apart. The percentage shows how many frames were re-localized under 55 cm of the previous re-localization. The number 55 has been chosen as the maximum distance between two consecutive frames taken at 1 Hz while walking at 55 cm per second.

## 5 DISCUSSION

Multi-session seems a valid approach to improve visual re-localization robustness to illumination changes in indoor environments. The dataset used in this paper is however limited to 1 day. Depending on whether it is sunny, cloudy, or rainy, or because of variations of artificial lighting conditions in the environment or if curtains are open or closed, more mapping sessions would have to be taken to keep high re-localization performance over time. During weeks or months, changes in the environment (e.g., furniture changes, items that are being moved, removed, or added) could also influence performance. Continuously updating the multi-session map to adapt over time to environment changes could be a solution (Labbé and Michaud, 2017) which however, as the results suggest, would require more RAM even if graph reduction is enabled. For very long-term continuous multi-session mapping, a solution using RTAB-Map could be to enable its memory management approach (Labbé and Michaud, 2013), which would limit the size of the map in RAM. Another complementary approach to graph reduction could be to remove offline nodes on which the robot did not re-localize for a while (like weeks or months). Each node in the map would have to keep information about when was the last time a new frame has been re-localized on it. For example, if a room in the house has been renovated or redecorated, the robot could eventually definitely “forget” the old room images while keeping only the new ones. Similarly, the more formal probabilistic approach to model feature persistence from Rosen et al. (2016) could also be integrated to remove features from the map that have “vanished” over time from the environment.

To construct the multi-session map from consecutive sessions, we almost followed exactly the same trajectory every time, so the camera orientation and position were very similar between the trajectories. On a robot, it may be not always the case. If the robot has to avoid a dynamic obstacle and moves out of its trajectory, even if the odometry is accurate, it may get lost because the point of view of the robot would be too different from the ones in the map (assuming that the robot has only a single camera with limited field of view). After detecting that the robot cannot plan

for a while in the map (because it has drifted too much), a way to recover from this could be to plan a path in the center of the current room, independently of the global map, by using only the local map around the robot. This would work only if the center of the rooms has been captured in the mapping sessions. To do so, during the initialization of the first session of the multi-session map, the robot should follow general navigation rules like staying as far as possible of obstacles, so naturally it would map center of corridors and rooms, which will be easier afterward to re-localize on when appending new sessions with different illumination levels. Instead having the robot re-mapping multiple times the environment, a digital twin of the target environment could be created to simulate illumination variations. In Caselitz et al. (2020), all possible lighting variations (based on combinations of lamps that can be on or off) including shadows are simulated in real time using the latest ray tracing technology. The camera can then be robustly tracked in the environment even if lights are turned off or on (creating drastic changes of illumination) during re-localization. Re-localization could then go beyond the recorded trajectory with same points of view. However, if the environment structurally changes, the digital twin would need to be updated at some point, which may not be as simple as recording a new mapping session with the robot itself.

In terms of limitations, this visual re-localization approach would obviously not work in perfect darkness. RTAB-Map can use LiDAR or ToF (time of flight) camera geometric data to refine re-localization’s transformation estimation (Labbé and Michaud, 2019), but it cannot do global re-localization without the discriminative visual features of a standard camera. This visual-based approach could be compatible with a camera system or robot equipped with lights, but it has yet to be tested. On some applications, the re-localization jump errors (around 2–6 cm) presented in the results may be also too high. As observed, the higher the number of inliers in TE is, the lower the re-localization jumps would be (greater accuracy). The Vis/Inliers parameter could be increased to accept only re-localization with higher number of inliers, at the cost of less frames re-localized. Note that re-localizing less often (creating large gaps of dead-reckoning) will produce higher re-localization jumps and also increase the chance to become completely lost. There is then a trade-off to think about.

Bai et al. (2019) suggest that neural networks in visual SLAM are becoming as competitive and even better than classical approaches. While end-to-end localization approach like PoseNet (Kendall et al., 2015) is not currently as competitive as classical SLAM approaches in indoor setting, replacing parts of the classic pipeline by their neural network counterparts can indeed increase robustness. Results in our paper suggest that the usage of SuperPoint as a feature detector increases the overall performance of re-localization against illumination changes. As mentioned in **Section 2**, the usage of a learned global descriptor (like NetVLAD (Arandjelovic et al., 2016)) could also improve likelihood accuracy in the replacement of BOW. The integration in this paper of SuperGlue for feature matching helps to get more feature correspondences than the classic nearest-neighbor approach when illumination is different between mapping and localization sessions. For transformation estimation, an approach such as DSAC (Brachmann and Rother, 2021) could be used to improve re-localization accuracy in the

replacement of the classic PnP RANSAC approach used in this paper. While the robustness to illumination is significantly increased using neural networks, computational requirements exposed in this paper show that they may not be used efficiently on all systems. If the system capabilities are limited (e.g., no GPU), RTAB-Map could still rely on classic methods using the proposed multi-session approach to get similar robustness to illumination, at the cost of having more sessions to capture. However, if the system can run those neural networks, it is recommended to use them with RTAB-Map to decrease the number of recorded sessions required for optimal re-localization performance.

## 6 CONCLUSION

The results in this paper suggest that regardless of the visual features used, similar re-localization performance is possible using a multi-session approach. The choice of the visual features could then be based on computation and memory cost, specific hardware requirements (like a GPU), or licensing conditions. The more illumination invariant the visual features are, the less sessions are required to reach the same level of performance. Graph reduction can further decrease significantly memory usage of multi-session maps while keeping high re-localization performance, at the cost of slightly worst re-localization accuracy. As an improvement, a better selection of which nodes to keep in the multi-session map using a strategy described in Mühlfellner et al. (2016), Halodová et al. (2019) may help improve re-localization performance and accuracy when graph reduction is applied.

In future works, we plan to test this approach on a real robot to study if multiple consecutive sessions could indeed be robustly recorded autonomously with standard navigation algorithms.

## REFERENCES

- Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). “Freak: Fast Retina Keypoint,” in IEEE Conf. Computer Vision and Pattern Recognition, 510–517. doi:10.1109/cvpr.2012.6247715
- Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). “Kaze Features,” in European Conference on Computer Vision (Springer), 214–227. doi:10.1007/978-3-642-33783-3\_16
- Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). “Netvlad: Cnn Architecture for Weakly Supervised Place Recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 5297–5307. doi:10.1109/cvpr.2016.572
- Bai, X., Huang, M., Prasad, N. R., and Mihovska, A. D. (2019). “A Survey of Image-Based Indoor Localization Using Deep Learning,” in 2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC) (IEEE), 1–6. doi:10.1109/wpmc48795.2019.9096144
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vis. Image Understanding* 110 (3), 346–359. doi:10.1016/j.cviu.2007.09.014
- Brachmann, E., and Rother, C. (2021). “Visual Camera Re-localization from Rgb and Rgb-D Images Using Dsac,” in IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/tpami.2021.3070754
- Bradski, G., and Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, Inc.
- Bürki, M., Giltschenski, I., Stumm, E., Siegwart, R., and Nieto, J. (2016). “Appearance-based Landmark Selection for Efficient Long-Term Visual Localization,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4137–4143. doi:10.1109/iros.2016.7759609
- Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). “Brief: Binary Robust Independent Elementary Features,” in European Conf. Computer Vision (Springer), 778–792. doi:10.1007/978-3-642-15561-1\_56
- Carlevaris-Bianco, N., and Eustice, R. M. (2014). “Learning Visual Feature Descriptors for Dynamic Lighting Conditions,” in IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2769–2776. doi:10.1109/iros.2014.6942941
- Carlevaris-Bianco, N., Ushani, A. K., and Eustice, R. M. (2016). University of michigan north Campus Long-Term Vision and Lidar Dataset. *Int. J. Robotics Res.* 35 (9), 1023–1035. doi:10.1177/0278364915614638
- Caselitz, T., Krawez, M., Sundram, J., Van Loock, M., and Burgard, W. (2020). “Camera Tracking in Lighting Adaptable Maps of Indoor Environments,” in 2020 IEEE International Conference on Robotics and Automation (ICRA) (IEEE), 3334–3340. doi:10.1109/icra40945.2020.9197471
- Churchill, W., and Newman, P. (2013). Experience-based Navigation for Long-Term Localisation. *Int. J. Robotics Res.* 32 (14), 1645–1661. doi:10.1177/0278364913499193
- Corke, P., Paul, R., Churchill, W., and Newman, P. (2013). “Dealing with Shadows: Capturing Intrinsic Scene Appearance for Image-Based Outdoor Localisation,” in IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2085–2092. doi:10.1109/iros.2013.6696648
- Dayoub, F., and Duckett, T. (2008). “An Adaptive Appearance-Based Map for Long-Term Topological Localization of mobile Robots,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 3364–3369. doi:10.1109/iros.2008.4650701
- Dellaert, F. (2012). *Factor Graphs and GTSAM: A Hands-On Introduction*. Georgia Institute of Technology. Technical report.

Testing over multiple days and weeks could give also a better idea of the approach’s robustness on a real autonomous robot. The outdoor RobotCar (Maddern et al., 2017) or NCTL (Carlevaris-Bianco et al., 2016) datasets could be used to evaluate if the same conclusions can be applied to outdoor scenarios including seasonal changes.

## DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

## AUTHOR CONTRIBUTIONS

ML: conception of the work; acquisition, analysis and interpretation of data for the work; drafting the paper. FM: revising the work/paper critically for important intellectual content; provide approval for publication of the content. ML and FM: agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

## FUNDING

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC RGPIN-2016-05096) and Fonds de recherche du Québec – Nature et technologies (FRQNT), INTER Strategic Network (2020-RS4-265381, 2018-RS-203302).

- DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). "Superpoint: Self-Supervised Interest point Detection and Description," in IEEE Conf. Computer Vision and Pattern Recognition Workshops, 224–236. doi:10.1109/cvprw.2018.00060
- Grisetti, G., Kümmerle, R., Stachniss, C., and Burgard, W. (2010). A Tutorial on Graph-Based SLAM. *IEEE Intell. Transport. Syst. Mag.* 2 (4), 31–43. doi:10.1109/its.2010.939925
- Halodová, L., Dvořáková, E., Majer, F., Vintr, T., Mozos, O. M., Dayoub, F., et al. (2019). "Predictive and Adaptive Maps for Long-Term Visual Navigation in Changing Environments," in IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS) (IEEE), 7033–7039. doi:10.1109/iros40897.2019.8967994
- Johns, E., and Yang, G.-Z. (2013). "Feature Co-occurrence Maps: Appearance-Based Localisation throughout the Day," in IEEE Int. Conf. Robotics and Automation, 3212–3218. doi:10.1109/icra.2013.6631024
- Kendall, A., Grimes, M., and Cipolla, R. (2015). "Posenet: A Convolutional Network for Real-Time 6-dof Camera Relocalization," in Proceedings of the IEEE international conference on computer vision, 2938–2946. doi:10.1109/iccv.2015.336
- Konolige, K., and Bowman, J. (2009). "Towards Lifelong Visual Maps," in Proc. IEEE/RISJ Int. Conf. on Intelligent Robots and Systems, 1156–1163. doi:10.1109/iros.2009.5354121
- Krajník, T., Cristóforis, P., Kusumam, K., Neubert, P., and Duckett, T. (2017a). Image Features for Visual Teach-And-Repeat Navigation in Changing Environments. *Robotics Autonomous Syst.* 88, 127–141. doi:10.1016/j.robot.2016.11.011
- Krajník, T., Fentanes, J. P., Santos, J. M., and Duckett, T. (2017b). Frement: Frequency Map Enhancement for Long-Term mobile Robot Autonomy in Changing Environments. *IEEE Trans. Robot.* 33 (4), 964–977. doi:10.1109/tro.2017.2665664
- Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). "g2o: A General Framework for Graph Optimization," in IEEE Int. Conf. Robotics and Automation, 3607–3613.
- Labbé, M., and Michaud, F. (2013). Appearance-based Loop Closure Detection for Online Large-Scale and Long-Term Operation. *IEEE Trans. Robot.* 29 (3), 734–745. doi:10.1109/tro.2013.2242375
- Labbé, M., and Michaud, F. (2017). Long-term Online Multi-Session Graph-Based Splam with Memory Management. *Auton. Robot* 42, 1133–1150. doi:10.1007/s10514-017-9682-5
- Labbé, M., and Michaud, F. (2019). RTAB-map as an Open-Source Lidar and Visual Simultaneous Localization and Mapping Library for Large-Scale and Long-Term Online Operation. *J. Field Robotics* 36 (2), 416–446. doi:10.1002/rob.21831
- Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). "BRISK: Binary Robust Invariant Scalable Keypoints," in IEEE Int. Conf. Computer Vision, 2548–2555. doi:10.1109/iccv.2011.6126542
- Li, S., Handa, A., Zhang, Y., and Calway, A. (2016). "Hdrfusion: Hdr Slam Using a Low-Cost Auto-Exposure Rgb-D Sensor," in 2016 Fourth International Conference on 3D Vision (3DV) (IEEE), 314–322. doi:10.1109/3dv.2016.40
- Linegar, C., Churchill, W., and Newman, P. (2015). "Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation," in IEEE International Conference on Robotics and Automation (ICRA) (IEEE), 90–97. doi:10.1109/icra.2015.7138985
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comp. Vis.* 60 (2), 91–110. doi:10.1023/b:visi.0000029664.99615.94
- Lowry, S. M., Milford, M. J., and Wyeth, G. F. (2014). "Transforming Morning to Afternoon Using Linear Regression Techniques," in IEEE Int. Conf. Robotics and Automation, 3950–3955. doi:10.1109/icra.2014.6907432
- Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). 1 Year, 1000 Km: The Oxford RobotCar Dataset. *Int. J. Robotics Res.* 36 (1), 3–15. doi:10.1177/0278364916679498
- McManus, C., Churchill, W., Maddern, W., Stewart, A. D., and Newman, P. (2014). "Shady Dealings: Robust, Long-Term Visual Localisation Using Illumination Invariance," in IEEE Int. Conf. Robotics and Automation, 901–906. doi:10.1109/icra.2014.6906961
- Milford, M. J., and Wyeth, G. F. (2012). "SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy winter Nights," in IEEE Int. Conf. Robotics and Automation, 1643–1649. doi:10.1109/icra.2012.6224623
- Mühlheller, P., Bürki, M., Bosse, M., Derendarz, W., Philippsen, R., and Furgale, P. (2016). Summary Maps for Lifelong Visual Localization. *J. Field Robotics* 33 (5), 561–590. doi:10.1002/rob.21595
- Muja, M., and Lowe, D. G. (2009). "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in Proc. Int. Conf. Computer Vision Theory and Application, 331–340.
- Neubert, P., and Protzel, P. (2015). "Local Region Detector+ Cnn Based Landmarks for Practical Place Recognition in Changing Environments," in 2015 European Conference on Mobile Robots (ECMR) (IEEE), 1–6.
- Neubert, P., Sünderhauf, N., and Protzel, P. (2013). "Appearance Change Prediction for Long-Term Navigation across Seasons," in European Conference on Mobile Robots (IEEE), 198–203. doi:10.1109/ecmr.2013.6698842
- Paton, M., MacTavish, K., Berczi, L.-P., van Es, S. K., and Barfoot, T. D. (2018). "I Can See for miles and miles: An Extended Field Test of Visual Teach and Repeat 2.0," in Field and Service Robotics (Springer), 415–431. doi:10.1007/978-3-319-67361-5\_27
- Ranganathan, A., Matsumoto, S., and Ilstrup, D. (2013). "Towards Illumination Invariance for Visual Localization," in IEEE Int. Conf. Robotics and Automation, 3791–3798. doi:10.1109/icra.2013.6631110
- Rosen, D. M., Mason, J., and Leonard, J. J. (2016). "Towards Lifelong Feature-Based Mapping in Semi-static Environments," in IEEE International Conference on Robotics and Automation (ICRA) (IEEE), 1063–1070. doi:10.1109/icra.2016.7487237
- Ross, P., English, A., Ball, D., Upcroft, B., Wyeth, G., and Corke, P. (2013). "A Novel Method for Analysing Lighting Variance," in Australian Conf. Robotics and Automation.
- Ruble, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "ORB: An Efficient Alternative to SIFT or SURF," in Proceedings IEEE International Conference on Computer Vision, 2564–2571. doi:10.1109/iccv.2011.6126544
- Sarlin, P.-E., Cadena, C., Siegwart, R., and Dymczyk, M. (2019). "From Coarse to fine: Robust Hierarchical Localization at Large Scale," in CVPR. doi:10.1109/cvpr.2019.01300
- Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). "SuperGlue: Learning Feature Matching with Graph Neural Networks," in CVPR. doi:10.1109/cvpr42600.2020.00499
- Sivic, J., and Zisserman, A. (2003). "Video Google: A Text Retrieval Approach to Object Matching in Videos," in Proceedings International Conference on Computer Vision, Nice, France, 1470–1478. doi:10.1109/iccv.2003.1238663
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). "A Benchmark for the Evaluation of RGB-D SLAM Systems," in Proceedings IEEE/RISJ International Conference on Intelligent Robots and Systems, 573–580. doi:10.1109/iros.2012.6385773
- Sünderhauf, N., Neubert, P., and Protzel, P. (2013). "Are We There yet? Challenging SeqSLAM on a 3000 Km Journey across All Four Seasons," in Proc. of Workshop on Long-Term Autonomy, IEEE Int. Conf. Robotics and Automation, 1–3.
- Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M. (2015). "On the Performance of Convnet Features for Place Recognition," in IEEE/RISJ international conference on intelligent robots and systems (IROS) (IEEE), 4297–4304. doi:10.1109/iros.2015.7353986
- Tola, E., Lepetit, V., and Fua, P. (2009). Daisy: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (5), 815–830. doi:10.1109/TPAMI.2009.77
- Valgren, C., and Lilienthal, A. J. (2007). "SIFT, SURF and Seasons: Long-Term Outdoor Localization Using Local Features," in 3rd European Conf. Mobile Robots, 253–258.

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Labbé and Michaud. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.