Transforming Multiple Visual Surveys of a Natural Environment Into Time-Lapses

Shane Griffith, Frank Dellaert, Cedric Pradalier

Transforming Multiple Visual Surveys of a Natural Environment Into Time–Lapses

Journal Title
XX(X):1–23

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Shane Griffith[1, 2], Frank Dellaert[1], and Cédric Pradalier[1,2]

Abstract

This paper presents a new framework to help transform visual surveys of a natural environment into time–lapses. As data association across year-long variation in appearance continues to represent a formidable challenge, we present success with a map–centric approach, which builds on 3D vision for visual data association. We use a foundation of map point priors and geometric constraints within a dense correspondence image alignment optimization to align images and acquire loop closures between surveys. This framework produces many loop closures between sessions. Outlier loop closures are filtered in the frontend and in the backend to improve robustness. From the result map, the Reprojection Flow algorithm is applied to create time–lapses.

The evaluation of our framework on the Symphony Lake Dataset, which has considerable variation in appearance, led to year–long time–lapses of many different scenes. In comparison to another approach based on using ICP plus a homography, our framework produced more and better quality alignments. With many scenes of the 1.3 km environment consistently aligning well in random image pairs, we next produced 100 time–lapses across 37 surveys captured in a year. Approximately one third had at least 20 (out of usually 33) well-aligned images, which spanned all four seasons. With promising results, we evaluated the pose error of misaligned image pairs and found that improving map consistency could lead to even better results.

Keywords

visual survey, data association, dense correspondence, visual SLAM, time–lapse, field robotics

1 Introduction

Visual surveys of a natural environment can lead to a large collection of image sequences, which this paper may help to transform into time–lapses. Figure 1 shows an example. One survey collects images over the length of a natural environment, which may consist of hundreds of unique scenes. As multiple surveys are acquired, image sequences start to form through the time elapsed at each scene. A transformation from multiple visual surveys into time–lapses connects the surveys and manifests the time elapsed through a set of well–aligned images (in this paper, time–lapses are presented after manually sorting them for the well–aligned images) at each scene.

A considerable research operation to collect years of image sequences of a natural environment may accumulate visual conditions that stand in the way of producing time–lapses. As changes in Nature add richness to a growing collection of image sequences, they also obscure determining which photos capture which scenes, and how images of the same scene align with one another (due to the variation in appearance between surveys, perceptual aliasing, and the unstructured environment). We can extract the motion within an image sequence, and we have pose priors from a GPS receiver and a compass. Yet, we need some way to short the variation in appearance of a natural environment and address the high likelihood that many may be incorrect.

We take a map–centric approach based on the use of visual SLAM (terms are defined and indexed in Fig. 2) which can, due to position–based correspondence, provide map point priors for robust data association. During a repeated survey, if the cameras are well–localized, the map from a different session can be projected onto the new images to provide position–based—independent of appearance—correspondence priors. Anything that keeps the same position across observations may have them, given that there are no large occlusions, a constraint that may be likely near a similar viewpoint. In a natural environment, trees, rocks, logs, and other objects that lack agency are prime examples. The Reprojection Flow algorithm (Griffith and Pradalier 2016, also see Sec. 6) exploits, for example, reprojected map points as priors to anchor dense correspondence. For multiple surveys whose maps and trajectories are consistent, the Reprojection Flow algorithm can be repeatedly applied to image pairs of the same scene to create a time–lapse (as we do in Sec. 8.5).

Given a way to get time–lapses from a consistent set of maps and trajectories, the next question is: how can we make multiple visual surveys of a large–scale natural environment consistent? The connectivity among them may be important; we may only be able to acquire a sparse subset of loop–closures among a subset of the surveys. Yet, the variation in appearance can lead to noisy loop closures. Visual data

[1]College of Computing, Georgia Institute of Technology, Atlanta, USA
[2]GeorgiaTech Lorraine, Metz, France

Corresponding author:
Shane Griffith, GeorgiaTech Lorraine, Metz, France
Email: sgriffith7@gatech.edu

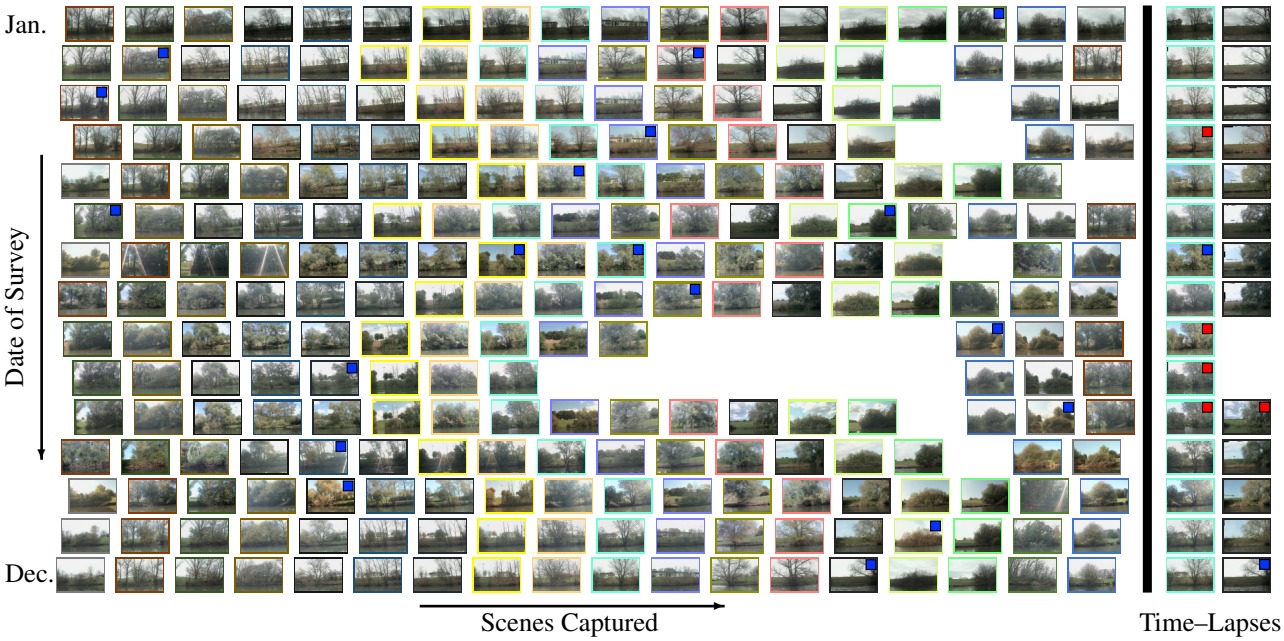

Figure 1. Depicting the transformation of unaligned, visual surveys of a natural environment into time–lapses using our approach. This example shows 15 of 37 sessions and 18 of 100 scenes from our evaluation in Sec. 8. Each survey consists of a video and the camera trajectory from a robot as it moved through the environment. Surveys from different dates are initially unaligned. Our framework provided the ability to acquire loop closures across considerable variation in appearance, which was a part of a complete pipeline to transform the surveys into time–lapses. **left)** A hand–selected, reference image from a particular scene and survey is automatically found in the other surveys. The result set of images is shown bordered with the same color. There are 18 different sets. **right)** The images from two of the scenes aligned into time–lapses. Red squares denote misaligned images. Blue squares denote reference images. Each manually sorted set of images compose a time–lapse.

association may only be effective between surveys captured around the same time (e.g., a month). It may also be effective across longer timelines (e.g., a year), to other surveys captured in the same season. With multiple surveys connected by loop–closures along a chain, loop closures between surveys at the beginning and the end of the chain may be needed to keep the ends from drifting apart. This may give multi–session optimization enough information to bring surveys into alignment. Thus, a map–centric approach may need a maximized amount of accurate loop closures between sessions.

This paper introduces a framework to assist a human in transforming multiple visual surveys of a natural environment into time–lapses. That is, our framework computes the time–lapse of all the images at the same scene, but they are presented after they are manually sorted. We create a map–centric approach for obtaining loop closures across challenging variation in appearance between sessions. Our pipeline has three stages: 1) single–session SLAM; 2) inter–session loop closure (ISLC) search; and 3) multi–session optimization. Rather than try to match feature descriptors of individual landmarks across sessions, a session's landmarks are assumed to be time–dependent. Data association occurs through dense correspondence. During single–session SLAM (Sec. 4) a trajectory and a map are acquired for each survey, which are independent (no shared local image features and no loop closures) of those for other surveys. During ISLC search (Sec. 5), surveys are connected at the snapshots where dense image correspondence is verified. A dense correspondence maps local image features across surveys, from which localization can be performed to acquire ISLCs, and in turn during multi–session optimization (Sec. 7), multiple sessions be made consistent.

We evaluated our framework on a large dataset of surveys of a 1.3 km natural environment (Sec. 3). Applying our framework to each year of the dataset showed that a large number of loop closures were produced. The map consistency was evaluated by aligning random image pairs from one year of 37 surveys and then manually labeling their precision. In 1000 image alignments, our framework outperformed an approach based on the use of ICP plus a homography. With many scenes in the environment that consistently aligned well, we next produced 100 time–lapses at random scenes and found many of them to capture the seasonal change. Poor image alignments were found to occur where pose error reduced the accuracy of position–based correspondence. Our results show that map point correspondence priors and geometric constraints within a dense correspondence image alignment optimization could be used to achieve data association across the year–long variation in appearance of a natural environment.

2 Related Work

Transforming visual surveys into time–lapses using a consistent map touches on many challenging areas of data association. We first describe related time–lapse work (Sec. 2.1). We then touch on scalability and focus on robustness in related work for visual data association (Sec. 2.2 and 2.3) and backend optimization (Sec. 2.4 and 2.5). Prior work follows (Sec. 2.6).

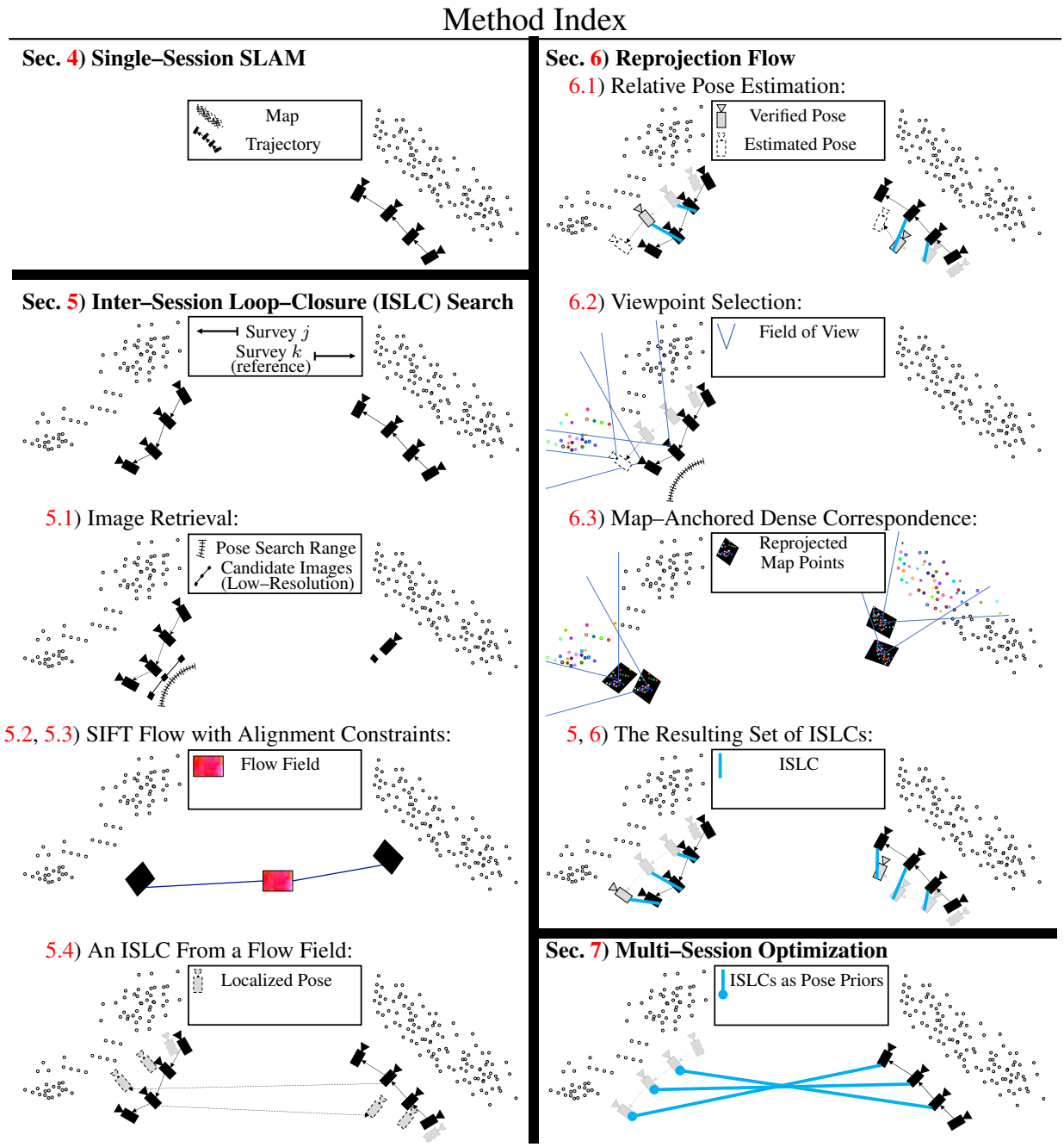

Figure 2. Primary methods indexed by section.

(Sec. 4) *Simultaneous localization and mapping (SLAM)* is the process used to acquire a *map* of 3D landmarks viewed over a *trajectory* of 6D camera poses.
Single session refers to one particular deployment for one trajectory and map.
Multi–Session refers to multiple deployments, each with its own trajectory and map.

(Sec. 5) A *loop closure* specifies a 6D pose transform between nonconsecutive poses.
An *inter–session* loop closure (ISLC) is one between poses from two different sessions.
Loop closures are acquired using *data association*, the process of matching image data captured at different times.

(Sec. 5.1) *Image retrieval* finds an image for data association to a reference image using a preliminary data association step.

(Sec. 5.2) *SIFT Flow* is one particular *dense correspondence* approach to data association, which produces a *flow field* that matches each pixel across an image pair.

(Sec. 5.3) *Alignment constraints* keep the flow field geometrically consistent.

(Sec. 5.4) *Localization* computes the 6D pose transform from the correspondences between two images.

(Sec. 6) *Reprojection Flow* boosts data association success rates and accuracy between surveys using the geometric
(Sec. 6.1) information in a map and poses, but it only applies near verified loop closures where the pose transforms can be estimated between sessions.

(Sec. 6.2) *Viewpoint selection* finds an image for data association without a preliminary data association step.

(Sec. 6.3) A *map–anchored* dense correspondence is one with priors defined by sparse, reprojected map–points.

(Sec. 7) *Multi–session optimization* is the process that aligns multiple maps and trajectories using the ISLCs between them.

2.1 Building Time–Lapses

The related work on building time–lapses most similar to ours creates time–lapses from multiple surveys. Dong et al. (2017) acquire a dense point cloud from each session and then align them into a 4D point cloud for precision agriculture of a peanut farm. Loop–closures are acquired by applying a homography to find SIFT feature correspondences (time interval < 1 week between sessions). Milford et al. (2014) apply SeqSLAM to align images from multiple image sequences of a natural environment. Image pairs are aligned by applying an affine transformation to the correspondences obtained using an adapted SeqSLAM approach. Like Milford et al. (2014), our approach aligns environment–long sequences of surveys, yet it is map–centric like that of Dong et al. (2017), which provides much of the robustness to variation in appearance.

Publicly available photos of popular landmarks also capture a representative set for a transform into time–lapses. Techniques for large–scale scene reconstruction from mined internet photos adapt well into time–lapses as the reconstruction is temporally ordered. Martin-Brualla et al. (2015b,a) reconstruct scenes into time–lapses by building a depth map for each viewpoint at each instance in time. A color profile is computed for each 3D track, from which the scene is reconstructed into a time–lapse. Zhou et al. (2015) maximize correspondence consistency among the mesh of correspondences, or 'flowweb', of an image collection to align them. Our approach is also designed to make time–lapses whose images are more closely aligned, and it uses 3D structure build them.

2.2 Scalable Visual Data Association

Although offline processing of surveys does not need real-time scalability in visual data association for storage and retrieval, scalability can become a bottleneck if they are intractable (Sivic and Zisserman 2003). We use pose priors and the co-visibility of map points (Sec. 6.2) to mitigate the bottleneck. Related work has also applied co-visiblity heuristics, namely for appearance–based localization. Because visual features typically co-occur with other visual features on the same objects, appearance–based matching can exploit the distribution of features that may be observed there (Cummins and Newman 2008). An image is likely to match a query image if its visual features are highly co-occurring with those of the query image, as measured using mutual information in a Chow Liu tree. In the formulation of covisibility graphs (Jones and Soatto 2011; Stumm et al. 2013), a query image is localized to the node and its neighbors with the highest visual word frequency–inverse document frequency.

A number of other approaches are formulated for map maintenance to keep localization time small (Mühlfellner et al. 2016; Dymczyk et al. 2015). Sattler et al. (2011) store a descriptor for each 3D map point and localize as soon as enough correspondences are found. Dymczyk et al. (2015) acquire a summary map, of which landmarks are those likely to be matched in future runs and trajectories are those that capture novel structure. Linegar et al. (2015) prioritize the sessions to which localization is attempted. In our approach, the map of each survey is left as-is. But viewpoint selection avoids any descriptor comparison when the relative poses between sessions is known. Before a localization is acquired, the image retrieval in Sec. 5.1 gets a boost in scalability due to its low-res image alignment, similar to the idea to use compact image templates to keep image comparison fast (Milford and Wyeth 2012; Arroyo et al. 2015).

2.3 Robust Visual Data Association

Methods towards robust visual data association in outdoor environments have overcome variation in appearance in many ways (Lowry et al. 2016). Here, related work is organized into six areas: 1) image feature matching; 2) image sequence matching; 3) image modification; 4) dense correspondence; 5) video alignment; and 6) exploiting databases.

2.3.1 image feature matching New methods on local features (Krajnik et al. 2015; Gálvez-López and Tardos 2012), image patches (McManus et al. 2014), and whole images (Naseer et al. 2018) continue to find new ways to achieve robustness for matching. State-of-the-art descriptors for condition–invariant matching come primarily from neural networks (Sunderhauf et al. 2015; Chen et al. 2017; Khaliq et al. 2018; Garg et al. 2018). Although descriptors from an off-the-shelf network have less power when evaluated on a natural environment (Griffith and Pradalier 2017; Gomez-Ojeda et al. 2015), a neural network that is specifically designed and trained on images from a natural environment can acquire invariance to the conditions of its scenes (Gomez-Ojeda et al. 2015; Lopez-Antequera et al. 2017; Olid et al. 2018).

Although this paper does not use a neural network for visual data association, it is complementary and unbiased to the particular appearance–based approach. Higher data association accuracy could allow for longer periods of time between surveys. And across surveys where appearance–based data association sparsely spans the range of variation in appearance, where the training data is mismatched or is limited, where perceptual aliasing is high and the relative poses between surveys are accurate, or where verification with a map is desired, reprojected map points could provide anchors for visual data association.

2.3.2 image sequence matching Several approaches are tailored for matching a sequence of images, which add robustness to variation in appearance where single images may be hard to match. Sequences of image templates can be matched directly (Milford 2013; Arroyo et al. 2015), paired up in a network flow (Naseer et al. 2018), or as nodes of the data association graph (Vysotska and Stachniss 2016). Much shorter sequences of descriptors from a CNN may produce comparable accuracy (Facil et al. 2019). Naseer et al. (2018) showed that image sequences can be matched as a solution to the network flow problem through a cost matrix of matched descriptors. Access to a GPS and compass can provide, however, comparable coarse matching accuracy across surveys for which the appearance is similar, and this level of accuracy can be maintained year–round (Griffith and Pradalier 2017).

2.3.3 image modification Images may also be modified to account for the difference in condition between two

different surveys. Removing the illumination (Corke et al. 2013) or other general factors (Lowry and Milford 2016) could improve feature matching in those changing conditions (Corke et al. 2013). Two images can also be made more similar by adding a particular condition to one image Neubert et al. (2013). This paper avoids modifying the visual appearance in favor of relying on the scene geometry to gain robustness to variation in appearance.

2.3.4 dense correspondence Methods for dense correspondence match every pixel across two images, which inherently defines a transform between them, and which subsequently can make them suited to visual data association in a natural environment. In contrast to local image features or image patches, whole images capture the manifold structure of a scene (Oliva and Torralba 2006), a pattern that may be more persistent across appearance change. In contrast to whole image matching, a dense correspondence also defines how one image transforms into another, which can make it less sensitive to changes in viewpoint. Our paper builds on dense correspondence to gain its advantages to variation in appearance.

A dense correspondence may exist between two images whether they capture the same scene or different scenes. SIFT Flow demonstrated the dense correspondence of two images by aligning whole images of SIFT features (Liu et al. 2011). Improvements to the methodology of SIFT Flow has resulted in better computation time (e.g. Kim et al. 2013) and matching capability (e.g. Kim et al. 2017a). As the latest approaches have specifically focused on nonrigid dense correspondence (e.g. Kim et al. 2017b, 2018), prior work showed the integration of basic feature matching constraints for rigid dense correspondence (Griffith and Pradalier 2016). Sec. 5.3 shows how we add epipolar and forward–reverse matching constraints to SIFT Flow to improve its matching power for rigid dense correspondence.

2.3.5 video alignment Video alignment can simplify the registration task as it allows for a few simplifying assumptions that reduce problem complexity. Video sequences captured while driving, for example, can be aligned by assuming the camera is only rotated between frames and then estimating a homography between images (Diego et al. 2011). The alignments may not be exact, however, because the camera also often has a translation component. For video registration meant for more general applications, Sand and Teller (2004) demonstrated an approach towards video matching that estimates a dense correspondence field using pixel matches and optical flow. Like the work of Sand and Teller (2004), our approach does not model occlusion boundaries, which can limit image alignment quality as the viewpoint is changed.

2.3.6 exploiting databases In addition to curating the data saved for localization (e.g., Le and Milford 2018), there is also significant effort towards exploiting the large amount of data to improve data association success rates. A large source of data for localization is often available due to prior experience. Churchill and Newman (2013) showed that increased localization rates are possible if a new 'experience' of a scene is saved each time localization fails. Multiple experiences are acquired where scene change is more significant. Zhou et al. (2016) train a neural network to infer the 3D model of an object given its query image, which is subsequently used to infer the correspondence between two images of the same object type. They showed that correspondence across significant change in appearance can be achieved if the 3D structure is known. Reprojection Flow (Sec. 6) is based on the same principle: reprojected 3D points can indicate how to anchor image alignment.

2.4 Scalable Backend Optimization

Scalability in backend optimization is achieved in a number of ways (Cadena et al. 2016). Our multi-session optimization in Sec. 7 breaks the optimization into subgraphs, motivated by the scalability of Ni et al. (2007) and McDonald et al. (2013). The divide-and-conquer approach of Ni et al. (2007) was extended to multi-session SLAM by McDonald et al. (2013). One anchor variable is defined between each pair of sessions to represent the pose transform between them. The formulation is best if the poses are locally well-constrained. Our approach does not use anchor variables, but instead optimizes over all the loop closures between multiple sessions. Although several papers have shown scalability for real–time operation by keeping the pose graph small (Carlevaris-Bianco and Eustice 2013; Johannsson et al. 2013), or the optimization over it small (Sibley et al. 2010; Kaess et al. 2012), neither our implementation of single–session SLAM (Sec. 4) nor our implementation of multi–session optimization (Sec. 7) are used for real-time operation.

2.5 Robust Backend Optimization

The backend optimization may also have to be robust to outliers in data association due to the high possibility of bad loop closures (Thrun et al. 2004; Ferguson et al. 2004). Robustness can be explicitly added in an optimization over loop closure constraints. Our multi–session optimization in Sec. 7 employs expectation maximization, which has been used to eliminate outlier loop closures of distributed mapping algorithms Dong et al. (2015); Indelman et al. (2014). Other techniques have optimized for the likelihood of a tree of loop closure candidates (Ferguson et al. 2004), optimize for clusters of inliers that share consensus with the odometry (Latif et al. 2013), and optimize for graph consistency (Graham et al. 2015).

A number of approaches also filter outliers by encoding the uncertainty within the factor graph. Switchable constraints are binary variables that can be added for each loop–closure to filter poor ones (Sünderhauf and Protzel 2012), which have been extended so that outliers are dynamically rejected (Agarwal et al. 2013), and to account for multiple hypotheses of variables (Olson and Agarwal 2013). Carlone et al. (2014) implicitly modeled switchable constraints in smart factors, which are an abstraction for the support variables of an optimization problem. They replace the support variables to provide a reduced set of constraints on the set of target variables for optimization. In Sec. 4, we include smart projection factors in the single–session SLAM problem. Carlone and Calafiore (2018) gain robustness to spurious measurements by modeling constraints using noise distributions that account for large errors.

2.6 Prior Work and this paper

This paper builds on and follows up to a line of prior work. Rather than work towards the most efficient, scalable SLAM, this work defines a way to address data association across the variation in appearance of a natural environment, particularly using the spatial and the temporal information which may be provided by SLAM and multi–session optimization. Griffith et al. (2015) found that SIFT Flow could be used for data association across consecutive surveys. Griffith and Pradalier (2017) showed SIFT Flow worked best among several appearance–based methods and its limit in appearance–based data association is around three months between surveys. Griffith and Pradalier (2016) created the Reprojection Flow algorithm, which uses a consistent map and poses to find images of the same scene and then to anchor image alignment to the final dense correspondence. We collected the Symphony Lake Dataset (Sec. 3; Griffith et al. 2017), which is the dataset used for all of our evaluations.

This paper addresses the transformation of multiple visual surveys of a natural environment into time–lapses. It defines a map–centric approach for obtaining loop closures across challenging variation in appearance between sessions. Given that appearance–based data association using SIFT Flow is effective up to three months, a search for loop closures is applied between pairs of surveys up to that time limit. Between pairs of surveys, we apply our new pipeline for visual data association (extending initial work in Griffith and Pradalier 2016), which now integrates more filters and geometric information to help maximize robustness to poor loop closure candidates. Inconsistent loop closures are now filtered during loop closure acquisition and a batch, multi-session optimization. Time–lapses are now produced across the full environment.

3 Symphony Lake Dataset

We describe and evaluate our work in the context of the Symphony Lake Dataset (Griffith et al. 2017), which is a collection of multiple surveys of a lakeshore. Our dataset is presented here (rather than with the experiments in Sec. 8) to provide an example reference, but also because some of our methods would slightly change for a different dataset. For example, we use a constant velocity assumption rather than odometry constraints to perform SLAM (Sec. 4). As we describe in Sec. 8.2 some changes can be made to help generalize our approach to more datasets (e.g., from the related work of Dong et al. 2017; Pradalier et al. 2019, add a homography to pre-align images before extracting descriptors. In the Symphony Lake Dataset, images of the same scenes were likely captured near the same scale and the same orientation). A discussion of the parameter values we used is saved for Sec. 9.2.

3.1 Surveys

The Symphony Lake Dataset was collected as part of our work towards long–term inspection and monitoring. The dataset consists of 130 visual surveys of the shore of Symphony Lake in Metz, France. Over 3.5 years of variation in appearance were captured in the surveys, which were collected roughly bi-weekly between 2014 Jan. 6 and 2017 Oct. 30. Each survey follows the perimeter of the approx. 1.3 km lakeshore and captures images of it from the perspective of an unmanned surface vessel (USV). The GeorgiaTech–Loraine Clearpath Kingfisher (see Fig. 3), a catamaran–style USV, was deployed to collect each survey.

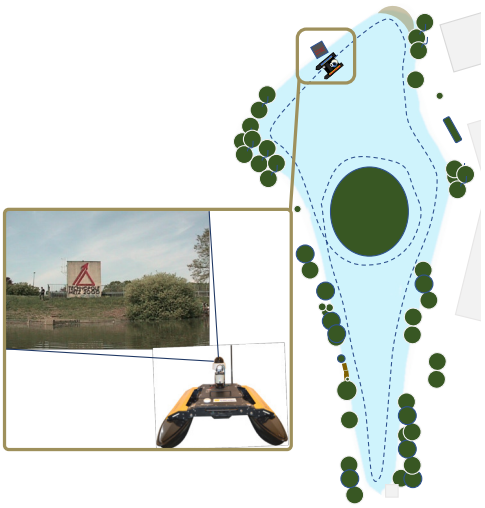

Figure 3. GTL's Clearpath Kingfisher collected the surveys of the Symphony Lake Dataset. It navigated the path shown as the dotted line with its camera facing towards the shoreline.

3.2 Robot

The GTL Kingfisher has a pan–tilt–zoom camera, with which 704x480@10fps images are captured, a GPS (2.5 m accuracy), a compass (10 degrees yaw accuracy), and an IMU, with which its trajectory is captured, and a 2D laser range–finder, with which its route along the shore is planned. A state–lattice motion planner uses the output from the laser to identify the best path. The path it chose was the one that kept the robot 10 m from the shore. That distance was usually far enough from the shore that it avoided collisions with small debris, yet was usually close enough for it to capture the scene well.

3.3 Data Collection

The data collected per survey, j, consisted of a sequence of images, $\mathcal{I}^j = \{\mathcal{I}^j_t\}_{t=1}^{n_j}$, each associated with a measured 6D camera pose, $P^j = \{p^j_t\}_{t=1}^{n_j}$. Images were initially captured at 10Hz, but we use the 1 Hz downsampled set of n_j frames due to their significant overlap. The height stayed constant within a survey, but it may have changed by a meter between surveys. Because we had no way to measure that change, however, it is approximated to zero for all the surveys. Similarly, because the boat does not have odometry values between successive poses, the velocity of the boat at each frame, $Z^j = \{z^j_t\}_{t=1}^{n_j}$, is used for a kinematic constraint. The acceleration as read from the IMU, $Y^j = \{y^j_t\}_{t=1}^{n_j}$, is integrated over time to get the USV's angular velocity, z^j_t.

3.4 Feature Extraction

Feature extraction per survey involved identifying keypoints, $\mathcal{M}^j_t = \{m^{j,t}_\psi\}_{\psi=1}^{n_{j,t}}$, in an image, $\mathcal{I}^j_t$, and tracking them for the duration they were visible (with an average accuracy of approx. 3 pixels). An image was first subdivided into a 12x20

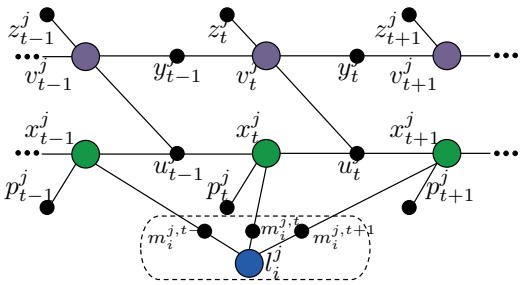

Figure 4. Factor graph of the single–session SLAM optimization problem. A colored node corresponds to a variable to be optimized. A black node corresponds to a factor, which is a constraint on the values of its connected variables. The dotted line depicts a smart factor, which encapsulates a landmark variable and its factors.

grid to identify where to extract new keypoints and where existing keypoint tracks were likely to be found. Up to five Harris corners in empty grid cells identified new landmarks. Each was tracked using the Kanade–Lucas–Tomasi (KLT, Lucas and Kanade 1981) feature tracking algorithm.

4 Single–Session Visual SLAM

The first step of survey processing consists in applying single–session visual SLAM to acquire the trajectory and the map for each survey. Although our framework processes data from all the surveys, connections are not yet acquired between them. Instead, each survey is optimized independently of the others to map its keypoint tracks into landmarks and to localize each pose at the frames along its trajectory.

Single–session visual SLAM is formulated as a batch, pose graph SLAM using the landmark feature tracks, measurements of the camera poses, and prior knowledge of the camera motion as shown in Fig. 4. Variable vertices (colored) are the values to be optimized and factor vertices (black) constrain the values of the variables they connect to. The variables include the camera poses, x_t^j, the camera velocities, v_t^j, and the landmark positions, l_i^j. The factors are derived from measurements of the camera poses, p_t^j, of the change in p_t^j and of the IMU, z_t^j, of the USV's relatively constant speed, y_t^j, and of the landmark feature tracks, $\mathcal{M}_t^j$. Our assumption that the boat moves with constant velocity is used to form a kinematic constraint, u_t^j, which defines the boat's change in pose. Fusing the information in this form enables a fast optimization for a low–error variable assignment.

The optimized estimate of each pose, $\hat{x}_t^j$, velocity, $\hat{v}_t^j$, and landmark, $\hat{l}_i^j$, in the factor graph is found using bundle adjustment. Bundle adjustment simultaneously refines the values of the 6D camera poses, the 6D camera velocities, and the 3D landmark positions to reduce the total error. The nonlinear minimization of error proceeds using the Levenberg–Marquardt algorithm. The GTSAM framework was utilized to perform this step (Dellaert 2012). Within the same framework, smart factors were also utilized, which employ the Schur complement to partition landmarks from poses, and thus yield a more robust result in less time (Carlone et al. 2014).

The result of this procedure for the j^{th} survey is the set, $\Pi_j = \{X^j, V^j, L^j\}$, for $X^j = \{\hat{x}_t^j\}_{t=1}^{n_j}$, $V^j = \{\hat{v}_t^j\}_{t=1}^{n_j}$, and $L^j = \{\hat{l}_i^j\}_{i=1}^{N_j}$.

5 Inter–Session Loop–Closure (ISLC) Search

After multiple surveys are collected and optimized, connections are acquired between them during the inter–session loop closure (ISLC) search (see Fig. 5). An ISLC connects two surveys with a pose transform, which is extracted from a pair of aligned images (a formal definition is given in Sec. 5.4.4). In this framework, the dense correspondence of two images defines their alignment (Sec. 5.2). We use a dense correspondence approach (i.e., SIFT Flow) to visual data association because 1) it provides a potentially more accurate alignment function (compared to e.g., a homography computed from local image correspondences); and 2) it may short a large degree of variation in appearance. Between surveys of a natural environment, however, it can still fail to provide *any* accurate correspondences. Therefore, we add to its power with a pipeline of constraints to help filter and circumvent errors—from the addition of alignment constraints (Sec. 5.3), to outlier removal (Sec. 5.3.1 and 5.4.1), to the localization setup (Sec. 5.4.2 and 5.4.3), and finally to localization verification (Sec. 5.4.4). Section 6 presents Reprojection Flow, which can provide map point correspondence priors between two surveys once a loop–closure is found.

5.1 Image Retrieval

Data association between two surveys, j and k, begins with image retrieval, which seeks the best candidate image, $\mathcal{I}_a^j$, from survey j at time a for data association to a reference image, $\mathcal{I}_b^k$, from survey k at time b. It is implemented in this work to reduce computations of full–resolution dense correspondence (which can be computationally expensive) that are likely inaccurate. The search first identifies the poses $\hat{x}_{\text{I}}^j..\hat{x}_{\text{II}}^j$, from survey j near the pose $\hat{x}_b^k$. For our dataset, nearby poses are those within 5 m and 20 degrees of $\hat{x}_b^k$. The search then tests the corresponding image candidates, $I_{\text{I}}^j..I_{\text{II}}^j$, for alignment. A low–resolution dense correspondence (a good indicator of the full–resolution correspondence quality) is computed for each pair $\{\mathcal{I}_b^k, I_\gamma^j\}_{\gamma=\text{I}}^{\text{II}}$. This search is parallelized. An image $\mathcal{I}_a^j$ is found if at least one of $\{\mathcal{I}_b^k, I_\gamma^j\}_{\gamma=\text{I}}^{\text{II}}$ has a verified alignment (as defined in Sec. 5.3.1). The one whose alignment is most verified with the reference image is the one that is returned.

5.2 SIFT Flow

This paper uses SIFT Flow (Liu et al. 2011) to compute dense correspondence, which consists of aligning whole images worth of SIFT (Lowe 2004) features. The idea is that, although some areas of an image are uninformative, a matched scene manifold—as defined by SIFT features—could anchor an alignment. Two images with very different appearance would be aligned along the manifold, with the uninformative regions taking values in the neighborhood it defined. Thus, a SIFT feature is extracted for every pixel of

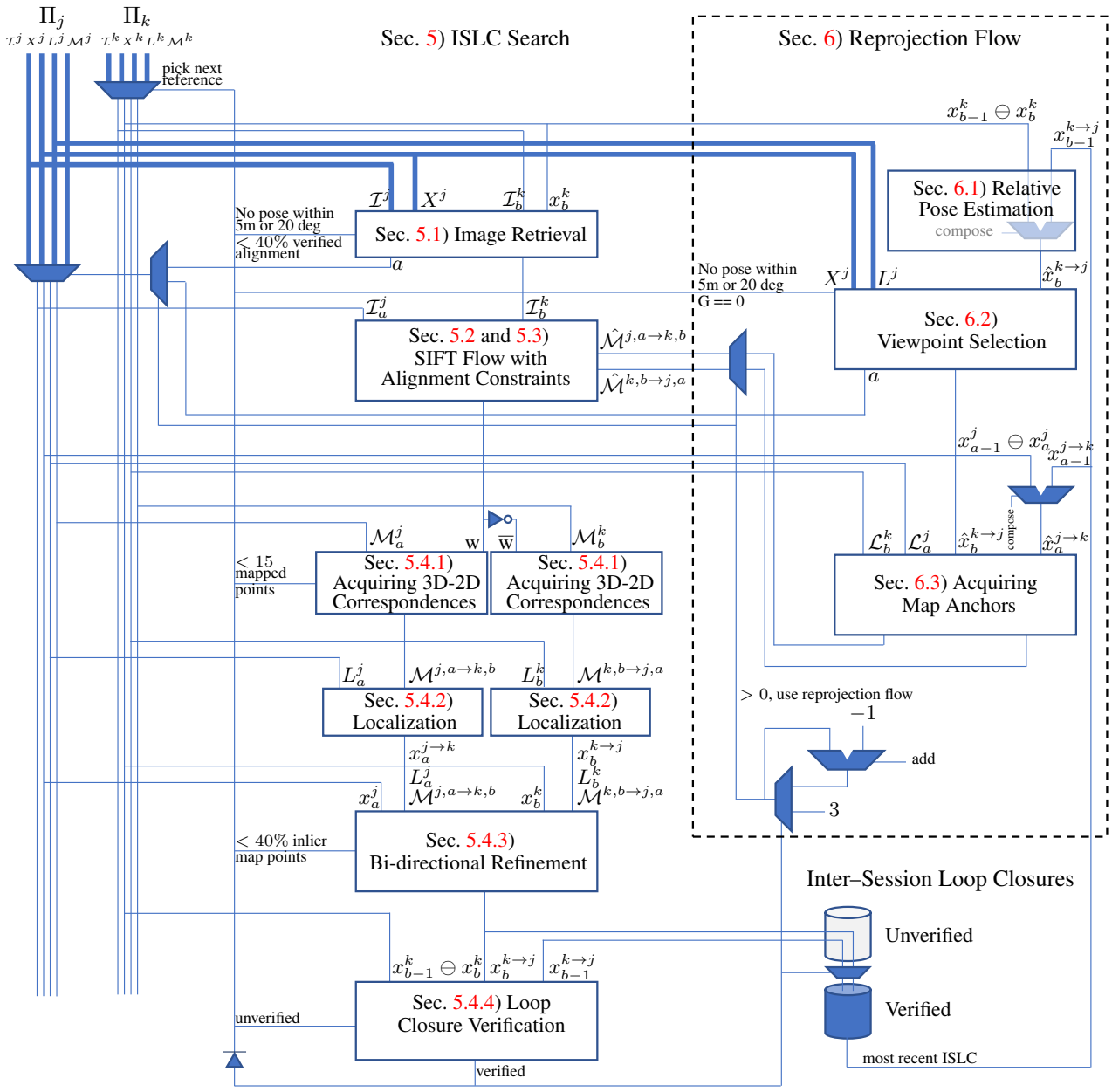

Figure 5. Visual data association between **(left)** two surveys using **(middle)** inter–session loop closure (ISLC) search (Sec. 5) and **(right)** Reprojection Flow when an ISLC is acquired (Sec. 6). Reprojection Flow is used up to three times without success before it is disabled. The logic here specifies the search with Reprojection Flow in the forward direction, but it is also used in the reverse direction. Also, the most recent ISLC is not necessarily between the times $a-1$ and $b-1$. See the text for details.

$\mathcal{I}_b^k$, which produces the SIFT image, S_b^k. Two SIFT images, S_b^k, S_a^j, are what are to be aligned.

Image alignment is defined as an optimization using a Markov Random Field (MRF). Each variable in the MRF corresponds to a pixel of S_b^k. Edges connect the variables for adjacent pixels. A pixel, $q \in S_b^k$, is assigned a flow $\mathrm{w}(q) = \{u_q, v_q\}$, where $u_q, v_q \in [-\mathrm{h}..\mathrm{h}]$, and $q + \mathrm{w}(q) \in S_a^j$. The quality of a flow is measured in terms of the descriptor match quality (data), how similar it is to the flow of adjacent pixels (smoothness), and how large it is (regularization), as defined by the alignment energy:

$$E(\mathrm{w}) = \sum_q min(|S_b^k(q) - S_a^j(q + \mathrm{w}(q))|_1, \mathrm{t}) + \sum_{r \text{ adj. } to\ q} min(\alpha|u_q - u_r|, \mathrm{d}) + min(\alpha|v_q - v_r|, \mathrm{d}) + \sum_q \nu|u_q + v_q| \quad (1)$$

The minimized alignment energy, $E(\overset{*}{\mathrm{w}})$ is computed using a coarse–to–fine alignment down an image pyramid with four layers. The initial flow field, w, is of images that are downsampled by a factor of 2^4. Whereas the flow field doubles in size with successive layers, the hypothesis space for each variable shrinks, which telescopes the correspondence. The truncation term, t, has value equal to the

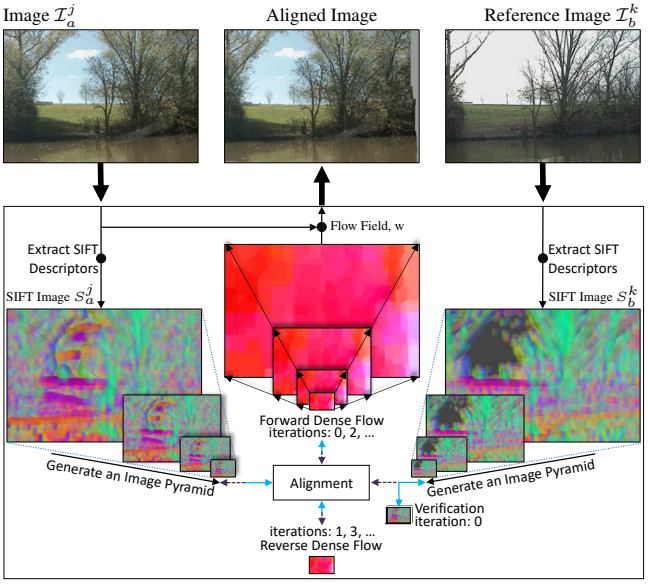

Figure 6. Depiction of image alignment using SIFT Flow plus alignment constraints (improvements we made to SIFT Flow). A SIFT Image is computed for each of the two input images. Each one is downsampled into an image pyramid with four layers. Image alignment proceeds from the top layer of the image pyramid down, with multiple iterations of alignment constraints applied at the top layer. An alignment is verified in iteration 0. To apply alignment consistency constraints, the forward flow field is computed in even iterations; the reverse flow field the odd iterations. Epipolar constraints are applied after iteration 0 and, unlike the alignment consistency, are also applied in the larger layers of the image pyramid.

median of the descriptor distances between S_b^k and S_a^j. The other parameter values (α, ν, d, and h) are listed in Sec. 9 and match what we used in all prior work.

5.3 Alignment Constraints

We added alignment constraints to the SIFT Flow framework to improve it in two ways: 1) to help identify whether an alignment may be informative; and 2) to help keep an alignment consistent with scene structure (see Fig. 6). An informative alignment may be robust to noise, which is a property that can be *verified* (Sec. 5.3.1). Without verification, the alignment process is terminated. A verified alignment can likely be, in turn, optimized. *Alignment consistency constraints* (Sec. 5.3.2) and *epipolar constraints* (Sec. 5.3.3) are generic feature matching constraints that we adapted to SIFT Flow to optimize verified alignments.

5.3.1 Alignment Verification The robustness of an alignment is tested immediately after obtaining the low resolution correspondence from the top of SIFT Flow's alignment pyramid. Noise is added to one image and the image pair is aligned a second time to test how much of the dense correspondence is retained. This is similar to the idea of 'adversarial perturbation,' wherein noise is added to an input image to test the robustness of a neural network (see, e.g., Dvijotham et al. 2018). The second alignment verifies the first one if a large percentage of the two dense correspondences match, which indicates that information may have been acquired (Sutton 2001; Stoytchev 2009). For our robot and the environment it captured, a dense correspondence of two images is verified if, after shifting one of the images up and to the right three pixels and then re-aligning them, at least 40% of the second dense correspondence matches the first. Note that, as implemented, alignment verification happens as part of image retrieval.

5.3.2 Alignment Consistency Constraints Verified alignments are optimized with the help of an alignment consistency constraint. The consistency of an image alignment is measured using the alignment in the reverse direction. Because the dense correspondence is directional, that is, from one image to the other, a somewhat different one may be computed for the reverse direction. This may be likely for highly self–similar scenes. Matching the correspondence in the forward and the reverse directions may help reduce perceptual aliasing.

The alignment consistency is implemented as an iterative two–cycle correction in the low resolution stage of SIFT Flow. There, several iterations are inexpensive. Pixels of the sift image S_b^k are first matched to pixels of the sift image S_a^j, then of S_a^j to S_b^k, and so on over at most 19 iterations. An odd number is used to end up at the forward flow, with which the next layer of the image alignment pyramid is initialized. Fewer than 19 iterations are performed if the consistency breaches 95% within one pixel. At that point the flow fields in both directions are consistent with one another.

Each iteration includes a modification to Eq. 1 to correct ambiguous correspondences. The data term of Eq. 1 is appended with the value

$$\text{cyc} = 16 \times ||\text{w}(q) - \text{w}^{prev}(q + \text{w}(q))||_2. \quad (2)$$

This term is the L_2 distance between the correspondence of the forward flows, w, and the flows of the previous iteration, w^{prev}, which are in the reverse direction. It is larger for pixel correspondences that diverge from consistency with the flow in the opposite direction. Its addition to the data term (rather than its multiplication to that) gradually pulls the forward and the reverse alignments into agreement.

5.3.3 Epipolar Constraints Verified alignments are also optimized using an application of epipolar constraints. Corresponding points between two images of the same, static scene should fall on epipolar lines. The original implementation of SIFT Flow lacked epipolar constraints. Here, SIFT Flow's lack of that constraint represents an opportunity for us to exploit more information for static dense correspondence.

Epipolar constraints guide image alignment after an initial set of correspondences are acquired. The very first set of correspondences is available after iteration 0 of the two–cycle consistency correction. Epipolar constraints are computed for each iteration thereafter using the previous flow field. They are also computed for successively larger layers of the image pyramid using the correspondences from the last.

Fundamental matrix estimation using RANSAC defines the epipolar constraint for each pixel. The data term of Eq. 1 is multiplied with the value

$$\text{epi} \propto 1 - \mathcal{N}(\mu, \delta), \quad (3)$$

where μ is the L_2 distance to the epipolar line from $q + \mathrm{w}(q)$ and $\delta = 2.5$. The data term is multipled by the epipolar constraint in order to strongly influence the flow to obey epipolar geometry.

The use of two–cycle consistency and epipolar constraints changes Eq. 1 to:

$$E(\mathrm{w}) = \sum_q min(|S_b^k(q) - S_a^j(q + \mathrm{w}(q))|_1, \mathrm{t}) \times \mathrm{epi} + \mathrm{cyc} \quad (4)$$
$$+ \sum_{r \text{ adj. } to \ q} min(\alpha|u_q - u_r|, \mathrm{d}) + min(\alpha|v_q - v_r|, \mathrm{d})$$
$$+ \sum_q \nu|u_q + v_q|$$

This formulation was found to produce the best alignments, which obeyed both geometric constraints among most correspondences.

5.4 An ISLC From a Flow Field

An accurate flow field, w, between images $\mathcal{I}_a^j$ and $\mathcal{I}_b^k$ specifies the dense correspondence of one image to another and is used to solve the PnP problem. The PnP problem is that of finding the pose of a camera from a set of 3D-2D correspondences. The result is a localized pose, which can become an inter–session loop closure constraint. There are four steps in the process: Sec. 5.4.1) acquiring 3D-2D correspondences from the flow field and the landmarks of each survey; Sec. 5.4.2) inter–session localization using the 3D-2D correspondences; Sec. 5.4.3) dual refinement of the two localized poses; and Sec. 5.4.4) a one-step loop–closure verification.

5.4.1 Acquiring 3D-2D Correspondences Each landmark from one image is mapped to a 2D coordinate of the other using the flow field, which results in two sets of 3D-2D correspondences (one for each direction) (see Fig. 7). More formally, the 2D coordinates of landmarks $\mathcal{M}_a^j = \{m_\psi^{j,a}\}_{\psi=1}^{n_{j,a}}$ that were observed in $\mathcal{I}_a^j$ are mapped to pixels of $\mathcal{I}_b^k$ as $\mathrm{w}(m_\psi^{j,a}) \rightarrow m_\psi^{j,a \rightarrow k,b}$. The flipped flow, $\overline{\mathrm{w}}$, which is obtained with reverse lookup, provides $\overline{\mathrm{w}}(m_\varphi^{k,b}) \rightarrow m_\varphi^{k,b \rightarrow j,a}$, for $\varphi \in 1..n_{k,b}$.

The mapping of landmarks through a flow field provides an approximation, which is further refined using epipolar constraints. Landmarks are, in this framework, assumed to lack feature descriptors for matching across surveys. Our framework's substitute is a mapping through the flow field, a result that may have slightly diverged from the true landmark locations. Thus, after mapping all the landmarks, the subset that satisfies epipolar geometry are retained. The correspondences are discarded if there are fewer than 15, which typically occurs when the flow misaligns scene structures.

5.4.2 Localization A set of 3D-2D correspondences is used to localize the camera pose of one survey to the other survey, as shown in Fig. 7, which is the perspective-n-point (PnP) problem. The 6D pose that corresponds to the mapped 2D image coordinates, x_a^j or x_b^k, is localized to the survey for which the 3D points are given, k or j, respectively. Because there are two directions of 3D-2D correspondences, a dual localization problem is formulated in which both camera poses are localized together.

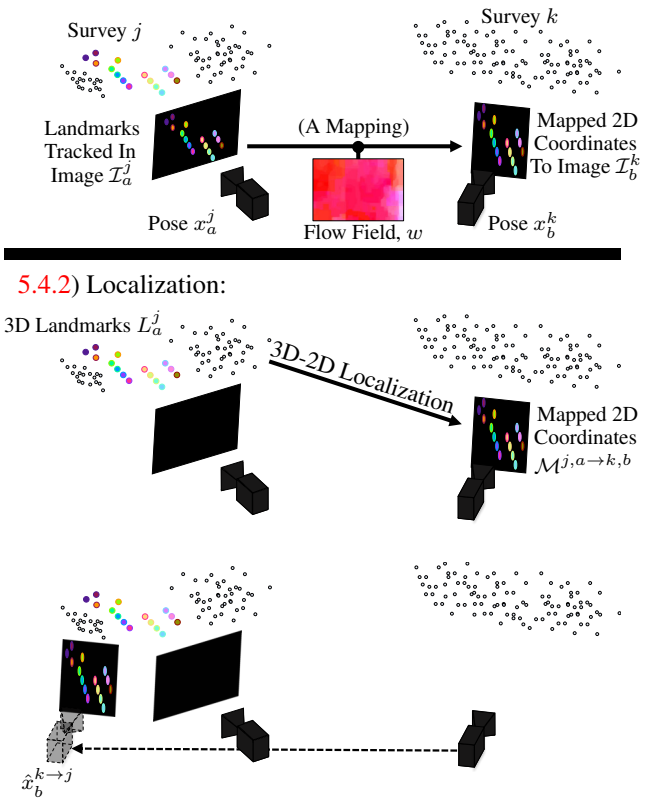

Figure 7. Localization to a prior survey after using a flow field to acquire 3D-2D correspondences. 5.4.1) A flow field defines a mapping from pixels of one image to another, with which the landmarks L_a^j seen in $\mathcal{I}_a^j$ are mapped to pixels $\mathcal{M}^{j,a \rightarrow k,b}$ of $\mathcal{I}_b^k$. 5.4.2) Localization proceeds as bundle adjustment using 100 iterations of RANSAC, each with 15 random 3D-2D point correspondences of the tuple $(L_a^j, \mathcal{M}^{j,a \rightarrow k,b})$. The result is the localized pose x_b^k in survey j, i.e., $x_b^{k \rightarrow j}$.

Localization is performed by applying bundle adjustment to a factor graph that represents a random sample of 3D-2D correspondences. Points of the tuple $(L_a^j, \mathcal{M}^{j,a \rightarrow k,b})$ are the 3D-2D correspondences used to compute $x_b^{k \rightarrow j}$, pose x_b^k in survey j (see Fig. 7). A set of 15 correspondences is randomly sampled from the tuple. A factor graph is created with one node for the pose and one localization factor for each of the 15 correspondences (a localization factor is a projection factor with a constant landmark and its implementation here is due to Beall and Dellaert 2014). The application of bundle adjustment minimizes the reprojection error (the error between the tracked pixel location of a landmark and its reprojected location) of the factors.

The estimate of $x_b^{k \rightarrow j}$ is refined in multiple iterations of RANSAC. The new estimate in each iteration is graded according to the number of inlier 3D-2D correspondences. Inliers have a reprojection error of less than 6.0 pixels. A better value for $x_b^{k \rightarrow j}$ is acquired if the estimate has more inliers. RANSAC is stopped after 100 iterations.

The same procedure is applied to $(L_b^k, \mathcal{M}^{k,b \rightarrow j,a})$ to get $x_a^{j \rightarrow k}$.

5.4.3 Bi-Directional Refinement The RANSAC procedure provides a close initial estimate of $x_a^{j\to k}$ and of $x_b^{k\to j}$, which are further refined using an expectation–maximization bi-directional bundle adjustment. Because the pair of tuples correspond to the same flow, the estimate of $x_a^{j\to k}$ is tied to the estimate of $x_b^{k\to j}$. If the two estimates are left as-is, optimized separately, the difference between them could skew later image and survey alignments. Bi-directional bundle adjustment may pull them into closer agreement. Additionally, in contrast to the RANSAC step, all the inlier 3D-2D correspondences are used in each iteration of expectation maximization.

A two–variable factor graph that corresponds to $x_a^{j\to k}$ and $x_b^{k\to j}$ is used to represent the bi-directional bundle adjustment. A factor is added to represent the constraint that

$$x_b^{k\to j} = x_a^j \oplus (x_a^{j\to k} \ominus x_b^k), \tag{5}$$

where $\oplus$ is the compose operation in the SE(3) lie group, and $\ominus$ the between operation. A localization factor is also added for each inlier 3D-2D correspondence. The poses $x_b^{k\to j}$ and $x_a^{j\to k}$ and the reprojection error for each 3D-2D correspondence are updated after each iteration, which can change the set of inliers. The optimization is terminated when the number of inliers stops changing or after 15 iterations. The result is discarded if fewer than 40% of the 3D-2D correspondences are inliers.

5.4.4 Loop Closure Verification An inter–session loop closure is acquired if the localized pose $x_a^{j\to k}$ passes a one-step verification using the nearest localized pose and the known change in pose (see Fig. 8 and e.g. Latif et al. 2013). This verification step is similar in principle to alignment verification (Sec. 5.3.1): A localized pose that is an informative one may be robust to noise, which is a property that can be verified. Once verified, the localized pose index, (j, a), the reference pose index, (k, b), and the transform between their poses, $x_a^j \ominus x_b^{k\to j}$, are composed into an ISLC

$$(j, a, k, b, x_a^j \ominus x_b^{k\to j}), \tag{6}$$

which is added to the set of ISLCs, H^j, for survey j that are used for multi-session optimization (Sec. 7). An ISLC that corresponds to $x_b^{k\to j}$ is also added to the set H^k for survey k, which simplifies applying the same constraint to both surveys.

The pose $x_a^{j\to k}$ is verified using a set of 3D points, the known change in pose, and the localized pose that is nearest in the sequence, e.g. suppose the one captured at time $a-1$, i.e. $x_{a-1}^{j\to k}$, which may not yet have been verified itself. An estimate $\hat{x}_a^{j\to k}$ is computed using the known change in pose between x_{a-1}^j and x_a^j as

$$\hat{x}_a^{j\to k} = x_{a-1}^{j\to k} \oplus (x_{a-1}^j \ominus x_a^j). \tag{7}$$

The 3D landmarks observed at x_b^k are projected onto both $x_a^{j\to k}$ and $\hat{x}_a^{j\to k}$. Both localized poses $x_a^{j\to k}$ and $x_{a-1}^{j\to k}$ are verified if at least 25% of the points project onto both images and their average reprojection error is less than 6.0 pixels.

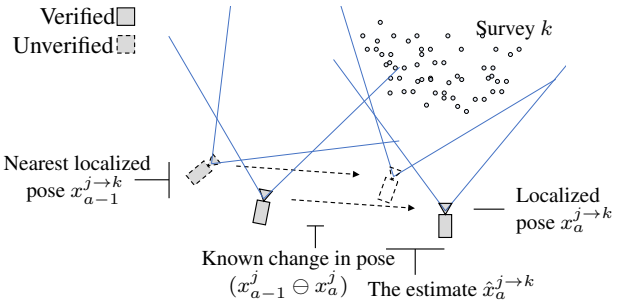

Figure 8. One–step verification of the localized pose $x_a^{j\to k}$ using the nearest localized pose, e.g., suppose $x_{a-1}^{j\to k}$, and the known change in pose, $(x_{a-1}^j \ominus x_a^j)$. The map points observed at x_b^k are projected onto the localized pose, $x_a^{j\to k}$, and the estimate, $\hat{x}_a^{j\to k}$. **solid)** The loop–closure is verified (at which point it is added to the set of ISLCs) if the map points project onto nearby pixels of both images ($x_{a-1}^{j\to k}$ is consistent with $x_a^{j\to k}$). **dotted)** The loop–closure remains unverified if the map points project onto distant pixels ($x_{a-1}^{j\to k}$ is inconsistent with $x_a^{j\to k}$).

6 Reprojection Flow

Reprojection Flow (Griffith and Pradalier 2016) can provide map point correspondence priors between two images when the pose transforms between them are known. Map points are reprojected from one survey onto another to acquire the priors. They may help to anchor the image alignment process to the correct dense correspondence when the appearance of a scene has changed. They may also help guide an alignment when perceptual aliasing is high. After estimating the localization of a pose (Sec. 6.1), the *reprojection of map points* determines which viewpoint is selected (Sec. 6.2) and where dense correspondence is anchored (Sec. 6.3).

6.1 Relative Pose Estimation

Relative pose estimation is the step of estimating the next localized pose in the sequence, $\hat{x}_{a+1}^{j\to k}$, which with a consistent map and poses, enables viewpoint selection and data association before using any information from appearance. The pose estimate $\hat{x}_{a+1}^{j\to k}$ can prespecify which of the landmarks L^k project onto $\mathcal{I}_{a+1}^j$ (similar to the depth map projection of LSD-SLAM Engel et al. 2014), which allows us to perform viewpoint selection (Sec. 6.2) without the use of image feature descriptors. Thus, viewpoint selection is appearance–invariant given a consistent map and poses. (Occlusions can affect, however, the accuracy of viewpoint selection without an additional heuristic to further limit the set of points that is considered 'visible'. A simple heuristic to add is a constraint on the camera pose.) The pose estimate also prespecifies where the landmarks L_b^k project onto $\mathcal{I}_{a+1}^j$ for the 2D coordinates $\hat{\mathcal{M}}^{k,b\to j,a+1}$, which can be used to anchor dense correspondence (Sec. 6.3) before using any information about the visual appearance of the scene. Note, the dense correspondence obtained using map point anchors may not be appearance–invariant. The estimate of $\hat{x}_{a+1}^{j\to k}$ is computed using the pose transform, $(x_a^j \ominus x_a^{j\to k})$, and the known change in pose, $(x_a^j \ominus x_{a+1}^j)$, as

$$\hat{x}_{a+1}^{j\to k} = x_a^{j\to k} \oplus (x_a^j \ominus x_{a+1}^j). \tag{8}$$

This equation is equivalent to that of Eq. 7.

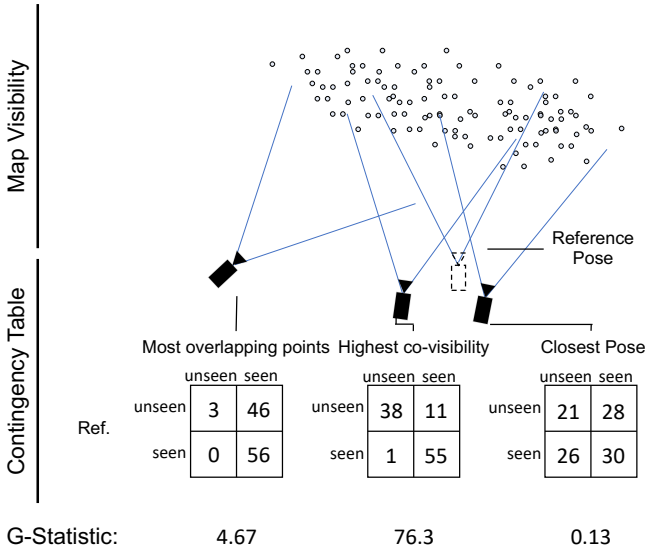

Figure 9. Viewpoint selection using the co-visibility of reprojected map points. The viewpoint with the most similar set of seen and unseen map points to a reference pose, as captured using a contingency table, has the highest co-visibility, and is the one for which the G-statistic is maximized.

Reprojection Flow is used during the ISLC search where the relative poses between two surveys are estimated. Image retrieval is replaced with the viewpoint selection of Sec. 6.2 and full image alignment with the map–anchored dense correspondence of Sec. 6.3. Because Reprojection Flow boosts image alignment, the search for ISLCs proceeds backwards and forwards from a new ISLC, sometimes reattempting image alignment where it previously failed without map anchors. The use of Reprojection Flow in a particular search direction is stopped after unsuccessfully aligning three image pairs in a row or after encountering an ISLC.

6.2 Viewpoint Selection

To identify two images of the same scene, we chose the approach that two viewpoints capture the same scene if the same set of map points projects onto them (see Fig. 9). In contrast to an approach based on feature matching, this information can be independent of the time scale across which viewpoint selection is performed. Over a year, the appearance of a scene could change negligibly or completely. If we were to rely on a visual feature descriptor (e.g., SIFT) to find the same scene, the difficulty of viewpoint selection could escalate with the variation in appearance of the environment. Given a consistent map and localized poses, however, the set of reprojected map points can provide information that is independent of appearance.

There are a number of ways to identify the same viewpoint in multiple surveys using a consistent map and localized poses. Two images capture the same scene if, for example, given one camera pose, a nearby pose is pointing in a similar direction. If that heuristic was used, however, the scene contents would be unaccounted for—a distant scene may not be visible in both images. Alternatively, the number of map points that are visible in both images could be maximized. Yet, one image may capture a much larger area than the other. In this paper, the image with the most similar viewpoint to a reference image views roughly the same set of map points.

Viewpoint selection utilizes *co-visibility*, a heuristic for maximizing the mutual information of reprojected map points (computed similarly to FAB-MAP from Cummins and Newman 2008, but here based on point projection rather than to identify whether a place has been seen before using appearance features). Co-visibility is based on the property that a map point either projects onto an image or not. A viewpoint has high co-visibility to a reference image if the map points that project onto it also project onto the reference image, and the rest project outside of both images. Two viewpoints have low co-visibility if many map points project onto one image and not the other.

To calculate the co-visibility of two viewpoints, co-visibility statistics for all the map points are accumulated in a two–variable contingency table. The two rows of the table correspond to 'seen' and 'unseen' map points for one viewpoint, the two columns of the table for the other viewpoint, in the form:

	'unseen'	'seen'
'unseen'	N_{00}	N_{01}
'seen'	N_{10}	N_{11}

,

The co-visibility of two viewpoints is calculated using the G-statistic, a method from statistical analysis, which has been applied in robotics to, e.g., measure co-movement in Griffith et al. (2011), as:

$$G = 2\sum_{i=0}^{1}\sum_{j=0}^{1} N_{ij} \ln\left(\frac{N_{ij}(N_{00}+N_{01}+N_{10}+N_{11})}{(N_{0j}+N_{1j})(N_{i0}+N_{i1})}\right), \tag{9}$$

The co-visibility to a reference image is calculated for each candidate image of a survey using Eq. 9. The equation is maximized for the viewpoint with the highest co-visibility.

6.3 Map–Anchored Dense Correspondence

The set of reprojected map points that are co-visible in an image pair is used to anchor their alignment. Each map point specifies a precise correspondence between the images of two well–localized cameras when projected onto them (see Fig. 10). This *reprojection flow* directly constrains the pixels where the map points are reprojected. Indirectly, reprojected map points anchor the alignment consistency constraints, define the epipolar lines to which the other pixels are constrained, initialize their hypothesis spaces to average flow of the map points, and limit the range of their hypothesis spaces. Collectively, a dense correspondence may be nearly fully specified using Reprojection Flow before using any information about appearance.

Map point priors are added to image alignment using SIFT Flow to obtain the final dense correspondence. Although the appearance aids less in the alignment of images from opposite seasons, it can improve the alignment of images captured during similar time periods. Furthermore, the smoothed dense correspondence created by the MRF optimization may help reduce artifacts created by strong map point anchors. Those anchors are only correct up to the

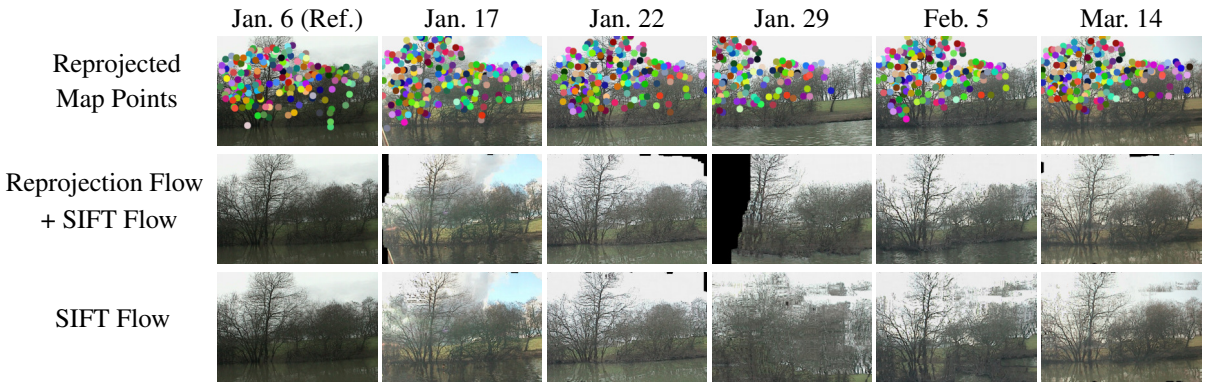

Figure 10. Map–anchored dense correspondence using Reprojection Flow for one scene from the Symphony Lake Dataset. Keypoint tracks from the reference survey (top left image) are shown reprojected onto images of the same scene from other surveys (top row). The locations of reprojected map points are the priors that anchor SIFT Flow to the final dense correspondence (middle row). Image alignment using the off-the-shelf version of SIFT Flow is provided for comparison (last row). Note that errors in the alignments produced using Reprojection Flow in this example occur in the areas of the images without reprojected map points (see e.g., the shoreline of the Jan. 29 image).

magnitude of their reprojection error. Thus, the alignment energy is as specified in Eq. 4 except for pixels that are map–anchored as part of Reprojection Flow (abbreviated to *rf* here), whose alignment energy is:

$$\begin{aligned} E(\mathrm{w}) = \mathrm{rf} + \mathrm{cyc} \\ + \sum_{r \text{ adj. } to\ q} min(\alpha|u_q - u_r|, \mathrm{d}) + min(\alpha|v_q - v_r|, \mathrm{d}) \\ + \sum_{q} \nu|u_q + v_q| \end{aligned} \tag{10}$$

Because the data term of the energy function is a function of scene appearance, it is replaced at the pixels where reprojected map points are specified. A suitable value for rf is calculated using the median of the data terms, t (from Eq. 1). That is,

$$\mathrm{rf} \propto (1 - \mathcal{N}(\kappa, \mathrm{s})) \times \mathrm{t}, \tag{11}$$

where κ is the pixel location of the reprojected point and s is the reprojection error divided by the image scaling factor. The cycle consistency, cyc, from Eq. 2 is still used. No alignment verification is performed when Reprojection Flow is used to guide the image alignment. Rather than project map points from all the surveys onto each image, only those from the two surveys that correspond to the two images are used, which limits the reprojection error to one direction.

Reprojected map points also define an initial hypothesis space at each pixel, which may help reduce perceptual aliasing for two reasons. First, the dense correspondence is initialized near the correct alignment (given an accurate pose transform between surveys), which is calculated as the average reprojection flow for all the landmarks of the reference image. Without Reprojection Flow, it is initialized to a zero-vector flow, which with the regularization term may create a bias in favor of the wrong alignment. Second, the hypothesis space is the L_∞ distance from the average flow of the reprojected map points. With a nearly correct initialization and a small hypothesis space, less information may be needed to pull the image into the correct alignment.

Note that using Reprojection Flow during the ISLC search is susceptible to being locked to inaccurate dense correspondences. Without discriminative appearance (e.g., a switch to gray images after reaching a verified localization), our current formulation of Reprojection Flow could keep dense correspondence at the map point priors. This is different from the case of images from different seasons, whose SIFT images may mismatch, but which may be highly discriminative. Discriminative appearance features counteract inaccurate map point anchors. In cases where a series of inaccurate map point anchors have locked a small series of images into a misalignment, resulting in inaccurate ISLCs, their inconsistency with the larger set of ISLCs could lead to their removal during multi-session optimization.

7 Multi–Session Optimization

The third step of survey processing consists in applying multi–session optimization to acquire consistent maps and trajectories for a set of surveys. The ISLCs between them are the constraints that indicate how to align them (see Fig. 11). Because visual feature descriptors are not shared among surveys, some of the ISLCs are *temporal loop closures*, which connect surveys at the beginning and the ends of the chain of surveys and may keep a long chain of surveys from drifting apart.

The constraints of the multi–session optimization can be represented using one large factor graph of multiple surveys and ISLCs, but we optimize over subgraphs due to the need for scalability and robustness. Bundle adjustment applied to the full graph may otherwise become intractable in peak memory and optimization runtime as the number of surveys is increased (Ni et al. 2007; McDonald et al. 2013). The full graph can be, fortunately, easily partitioned into subgraphs by replacing each ISLC with a pose prior (see Fig. 11). Subgraphs are thus optimized in parallel over several iterations. At the end of each iteration, the ISLC pose priors are updated using the result from the previous iteration. Compared to an optimization over the full graph, subgraph optimization can be lightweight, fast, and accurate (Ni et al. 2007).

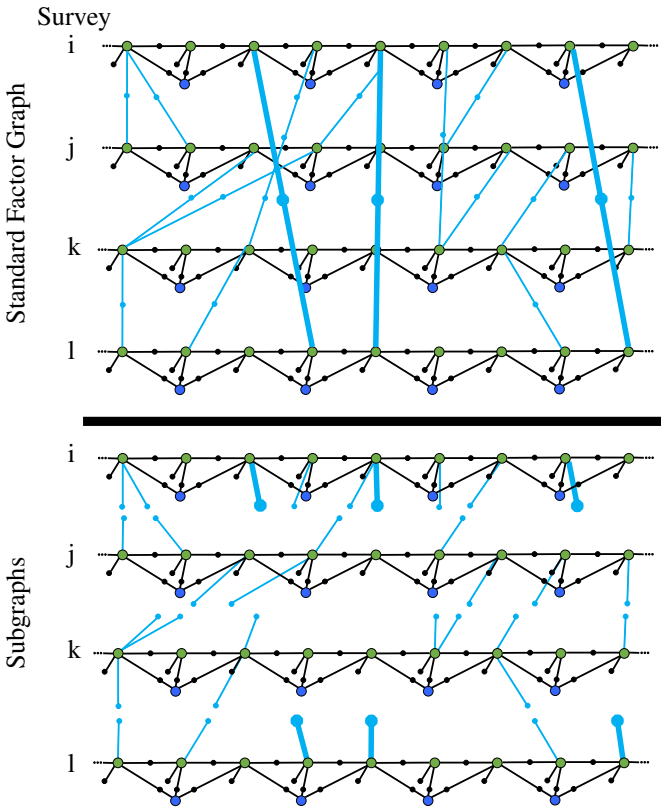

Figure 11. An example factor graph of the multi–session optimization and its conversion into subgraphs. The graph for each survey is nearly identical to that from single–session SLAM, in Fig. 4. However, instead of using velocity variables and a constant velocity assumption to constrain changes in camera poses, the changes in poses computed in Sec. 4 are used for that constraint. Blue lines represent loop closures between surveys. Thick blue lines delineate temporal loop closures, which are demarcated to bring attention to the fact that they may keep a long chain of surveys from drifting apart. Smart factors are used, but they are omitted in this visualization.

The high likelihood of noisy ISLCs and the often weak constraints between poses within each survey lead us to our expectation maximization implementation of subgraph optimization. Expectation maximization is used to filter poor ISLCs over multiple iterations. An optimization that includes inaccurate ISLCs would otherwise pull surveys into an inaccurate alignment. Between the multiple iterations of the parallel bundle adjustment of subgraphs, before the ISLC pose priors are recomputed, the error of each ISLC is calculated to find outliers. Outlier ISLCs are deactivated for the next iteration.

The multi–session optimization is defined in Algorithm 1. For a number of surveys, n_Π, each one, j, with optimized trajectories, X^j, and landmarks, L^j, measurements of landmarks, M^j, and inter–session loop–closures, H^j, an iterative bundle adjustment is applied to recover the multi–session–optimized trajectories, $\mathcal{X}^j$, and maps, $\mathcal{L}^j$. Two stages of optimization are performed (referenced by *state*), which include a series of optimizations with every ISLC, followed by an expectation maximization series in which the inconsistent ISLCs are removed. Each series is implemented in multiple iterations (lines 4–11) with a weighted update (line 10) to gradually pull each survey into agreement with one another (a nonweighted update is susceptible to a nonconverging oscillation in some cases). Each survey is optimized in parallel (lines 6–10), with graph construction (line 7), bundle adjustment using the Levenberg–Marquardt algorithm (line 8), a measure of convergence (line 9), and a weighted update (line 10) applied separately to each survey. The optimization is considered converged when the median change in position, c^j, averaged over all the surveys, $\mathcal{C}$, has changed by less than 0.01 m (line 4).

Algorithm 1 Subgraph multi–session optimization. A dot above the pose symbols in line 9 denotes the distance is only over the position, not both the position and the orientation.

Input X^j, L^j, M^j, H^j for $j \in 1..n_\Pi$
Output $\mathcal{X}^j$, $\mathcal{L}^j$ for $j \in 1..n_\Pi$

1: $\{\mathcal{X}^j, \mathcal{L}^j\} \leftarrow \{X^j, L^j\}$ `for` $j \in 1..n_\Pi$
2: enum $state\{ALL = 0, FILTERED, DONE\}$
3: **for** $s = state::ALL$; s != $state::DONE$; $++s$ **do**
4: **while** $\Delta\mathcal{C} > 0.01$ **do**
5: $H^j \leftarrow$`UpdateISLCs`$(H^j, \{\mathcal{X}^j, \mathcal{L}^j\}_{j=1}^{n_\Pi}, s)$
6: **for** $j \in 1..n_\Pi$ **do** ▷ in parallel
7: $G^j \leftarrow$`ConstructGraph`(X^j, M^j, H^j)
8: $\{\hat{\mathcal{X}}^j, \mathcal{L}^j\} \leftarrow$`BundleAdjustment`$(G^j)$
9: $c^j \leftarrow$ `Median`$(\{\| \dot{\hat{x}}_t^j - \dot{x}_t^j \|_2\}_{t=1}^{n_j})$
10: $x_t^j \leftarrow 0.9 \times \hat{x}_t^j + 0.1 \times x_t^j$ `for` $t \in 1..n_j$
11: $\mathcal{C} \leftarrow \frac{1}{n_\Pi} \sum c^j$
12: **return** $\{\mathcal{X}^j, \mathcal{L}^j\}_{j=1}^{n_\Pi}$

Inconsistent inter–session loop closures (line 5) are filtered in the second stage to help boost map consistency. Incorrect ISLCs are identified using reprojection error. For an ISLC between pose x_a^j and x_b^k, four different tests for outsize reprojection error are applied, which involve the 3D-to-2D point sets: 1) $(\mathcal{L}_a^j, \mathcal{M}^{j,a})$; 2) $(\mathcal{L}_b^k, \mathcal{M}^{k,b})$, 3) $(L_a^j, \mathcal{M}^{j,a \to k,b})$; and 4) $(L_b^k, \mathcal{M}^{k,b \to j,a})$. A threshold is computed using the reprojection error, r_a^j, which is measured using $(L_a^j, \mathcal{M}^{j,a})$. If any of the four tests of reprojection error exceed $3 \times max(r_a^j, \frac{1}{n_\Pi} \sum_j \frac{1}{n_j} \sum_a r_a^j)$, the ISLC is marked as an outlier and goes unused until some later update changes it back.

8 Experiments

The experiments evaluate our approach using the Symphony Lake Dataset (Sec. 3). We first show that our framework can be applied to a dataset of that size and complexity (Sec. 8.1). It finds abundant data association across its images, and can optimize all the maps and trajectories in tractable time. We next used the map with Reprojection Flow to align random image pairs across different time intervals. We compared the results to related approaches to show by how much the map helped image alignment (Sec. 8.2). The comparison goes one step further to show that the well–aligned images from this paper are more often superior (Sec. 8.3). We then divided image alignment quality by scene and found that many well–aligned images could be expected in many time–lapses (Sec. 8.4). That led us to produce 100 time–lapses of random scenes, of which several had a large number of well–aligned images (Sec. 8.5). With promising results, we evaluated the pose error of misaligned image pairs and found

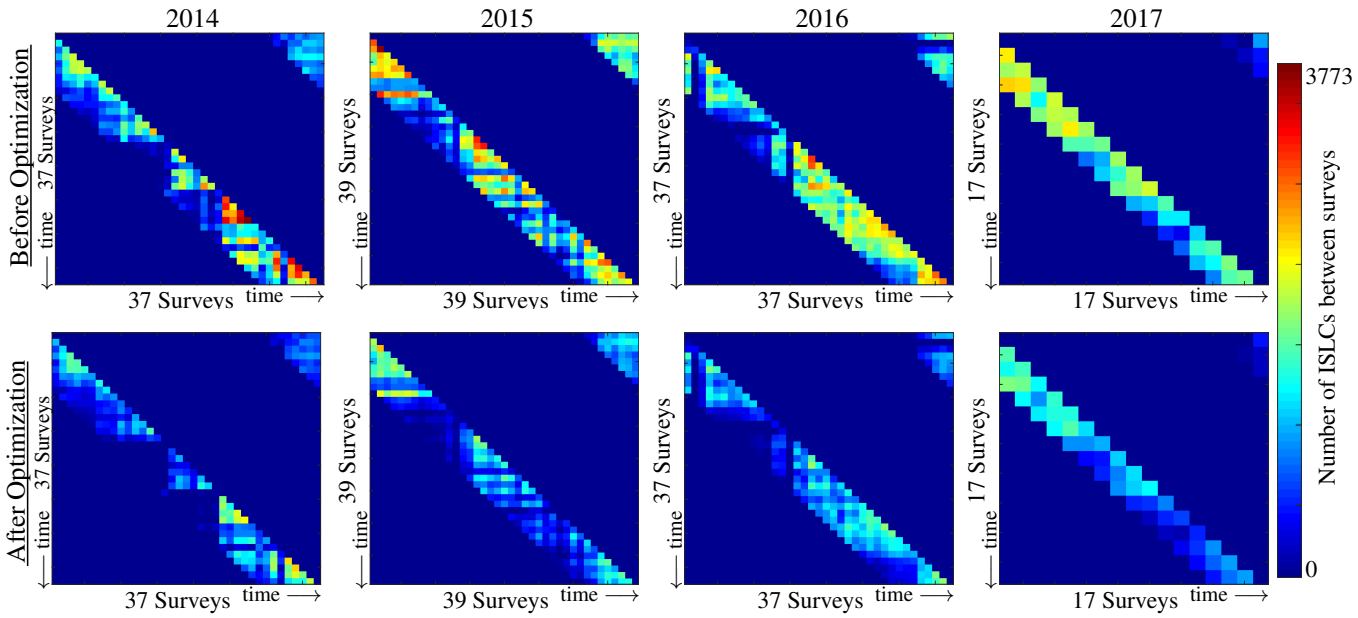

Figure 12. Inter–session loop closure connectivity for each year of the Symphony Lake Dataset **top)** before and **bottom)** after optimization. Each grid cell represents the number of ISLCs between two surveys. A figure has more grid cells if that year had more surveys. The grid cells for 2017 are larger because there were fewer surveys in that year. The ISLC search was also limited to three, rather than eight, surveys because we captured them less frequently. The nonzero cells in the top-right of each grid account for the ISLC connectivity across the time between the beginning and the end of each year.

that improving map consistency in future work could lead to even better results (Sec. 8.6).

8.1 Aligning One Year of Surveys

We first applied our framework to each year of surveys of the Symphony Lake Dataset to characterize its runtime and data association performance. The framework was applied four times for the four years of surveys between 2014 and 2017. For one survey from the dataset, a run of single–session SLAM had peak memory usage of nearly 16GB and completed in about two minutes (on a 2.4GHz machine). The average reprojection error of an optimized map and camera trajectory was approx. 3.5 pixels (Griffith and Pradalier 2017), which indicated that each map–trajectory tuple was individually consistent.

The data association pipeline of Sec. 5 was effective in providing a large number of inter–session loop closures between surveys of a natural environment, as shown in the top row of Fig. 12. For the Symphony Lake Dataset, we only ran the image alignment pipeline between surveys within three months of each other, which are the ones that typically had appearance–based alignments. The few ISLCs that could have been obtained beyond that was not worth the extra runtime. Because surveys were captured roughly bi-weekly, ISLC search was run from each survey to each of its eight previous surveys. The search was similarly applied to pairs of surveys between the beginning and the end of the year, which created a temporal loop–closure. For the set of surveys from 2014, for example, Sec. 5 was applied to a total of $37 \times 8 = 296$ pairs, which resulted in $332,441$ ISLCs. The runtime on an average pair of surveys took approx. 5-7 hours (mostly consumed by SIFT Flow). We used a cluster of 20 nodes to collect the constraints for each year of surveys in approx. one week. The use of Reprojection Flow within ISLC search added approx. $1.3\times$ more ISLCs.

A large number of inter–session loop closures were also retained after multi–session optimization, as shown in the bottom row of Fig. 12, which may represent a consistent set. Approximately 58% of the ISLCs were retained. After 6-10 iterations with all the ISLCs, each of the sets took five more iterations to converge again with filtering. Each survey was optimized in 30-45 seconds, with each iteration of multi–session optimization taking double that for 37 surveys on a machine with 32 threads (the runtime of optimization was in proportion to the number of surveys and the number of machine threads). The update step of the filtering stage (line 5 of Alg. 1) added approx. 1 minute to each iteration. The total optimization runtime was approx. 25 minutes for a set of 37 surveys. For comparison, a single iteration of bundle adjustment over the standard, full graph (shown in the top half of Fig. 11) took longer than 24 hours so was terminated.

The patterns of connectivity varied for different pairs of surveys, but were similar before and after optimization. Connectivity decreased between pairs away from the diagonal, consistent with the increased amount of time between surveys. It also dropped out between surveys with large differences in lake levels, notably to a group of surveys in 2014 and to two surveys in 2016. Image pairs between those surveys had the most variation in appearance. The matching pattern of connectivity after optimization indicates that the multi-session optimization result had a similar goodness-of-fit to the constraints among all the surveys. The evaluation does not indicate, however, how consistent the map is.

8.2 Image Alignment Quality

We measure the map consistency using the image alignment quality for image pairs of random scenes between random surveys. In this evaluation, images are aligned and hand–labeled to measure how consistent the multi–session

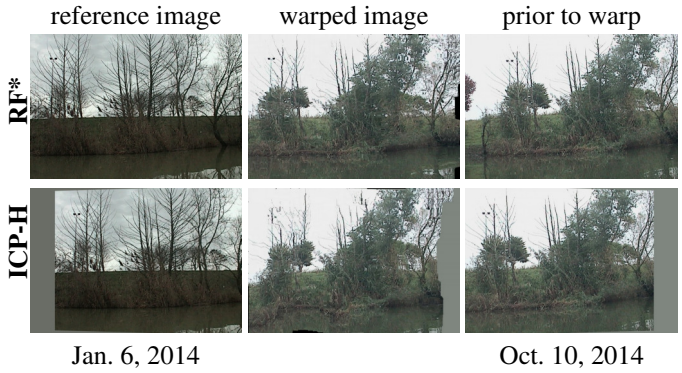

Figure 13. Example comparison of image alignment using RF* vs. using ICP-H (The method of Pradalier et al. 2019, to which we add SIFT Flow, which can make the alignment more precise). Whereas RF* uses the reprojection of map points to set the hypothesis space, in the latter approach, a homography is applied to parallelize the image planes. The ICP-H image pair may be nearly aligned. To this ICP-H pair we add, however, alignment using SIFT Flow. In this figure, only the image pair aligned using RF* is well–aligned. The ICP-H approach set the hypothesis space to the wrong regions of the two images.

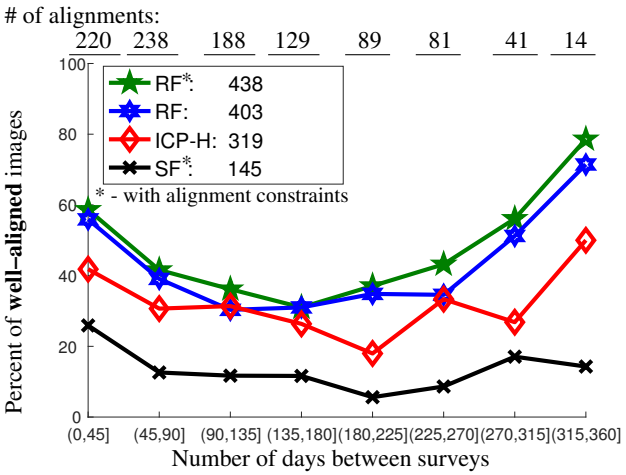

Figure 14. Comparison of alignment quality over time shown as the percent of well–aligned images per time interval. A single alignment was of two images of the same scene taken from two different surveys. Each method aligned the same set of 1000 random image pairs, generated from the 2014 surveys from the Symphony Lake Dataset. The top row shows the number of alignments in each time interval. The y-axis plots the percent of those well–aligned images.

optimization result is. The criteria for hand–labeling image pairs may be more clear with the video. The majority of scene content should appear to line up when the images are flickered back-and-forth. It is, however, subjective in some cases due to ambiguities and perceptual aliasing across seasons. We have measured the alignment quality using different heuristics in prior evaluations, but a better metric is to use the hand–labeled alignment quality. The trend of the hand–label metric best matches the qualitative image alignment quality. No ground truth was available.

For the comparison, 1000 random image pairs of the same scenes were selected, aligned, flickered back and forth in a display, and then manually labeled well–aligned or not for four different methods:

SF* SIFT Flow with image alignment constraints
RF* Reprojection Flow with image alignment constraints
RF Reprojection Flow without constraints
ICP-H an image alignment approach based on the use of ICP (on 2D LiDAR data), a homography, and multi–session optimization from Pradalier et al. (2019), to which we added SIFT Flow, which can make the alignment more precise.

Pradalier et al. (2019) produced a consistent map and trajectories of the Symphony Lake Dataset by applying an ICP algorithm to the 2D laser scan data of each survey, which produced a result they used to facilitate image alignment. They first applied ICP to the laser scan data to get pose transforms between images from different surveys. The sequence of 2D transforms were added to a factor graph of keyframes, one every 20 m and 1 minute degree, which was optimized for all the surveys in a year. After multi–session optimization, they showed that an image pair of the same scene could be nearly aligned by applying a homography to parallelize the image planes. Indeed, a homography also removes changes in scale that can affect SIFT feature matching. Thus, for the comparisons in this paper, we added SIFT Flow with a small hypothesis space (see the bottom row of Fig. 13), which can limit perceptual aliasing, can keep the alignment quality higher on average, and can make for a fairer comparison.

The four methods are distinguishable in the number of high–quality alignments they produced, as shown in Fig. 14. SIFT Flow produced significantly fewer well–aligned images, which shows that using a map to guide image alignment was, for these cases, significantly better than not. The best method at every time interval was Reprojection Flow, which relied most on the map to guide image alignment. The dip in the alignment quality towards six months indicated that the variation in appearance had an effect on all four methods.

8.3 *Comparing Reprojection Flow to the ICP-Homography Approach*

The number well–aligned images of Pradalier et al. (2019) showed that its performance was in many cases close to Reprojection Flow, which motivated a direct comparison of the aligned images to better gauge any difference in alignment quality. The aligned image pair of both methods were placed side-by-side in a flickering display. The result that better aligned the scene contents was manually identified. Otherwise, if neither was better than the other, the pair was labeled comparable. The process was repeated for all 1000 image pairs of Sec. 8.2.

The result in Fig. 15 shows that the two methods produced comparable image alignments in about half the cases. For the rest of the image pairs, Reprojection Flow produced better alignments twice as often as ICP-H, a trend unaffected by the change in time scales in a year. Most of the differences in image alignment quality appeared to derive from differences in map consistency. Where the maps were incorrect, the images were setup for an incorrect alignment, as shown in e.g. the bottom row of Fig. 13. Many of the comparisons were often between image pairs where neither aligned well,

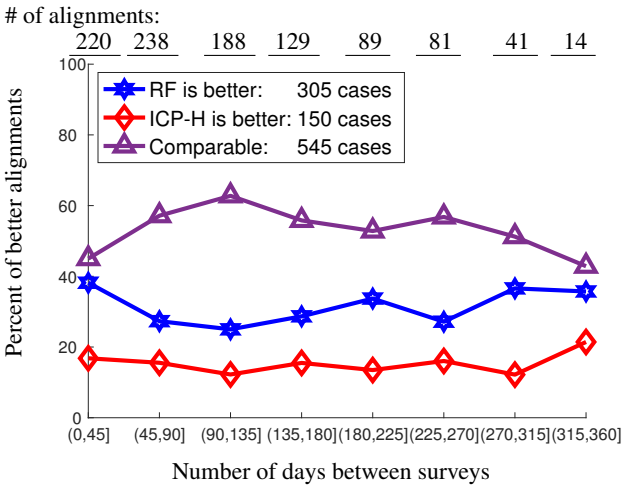

Figure 15. Grading Reprojection Flow to ICP-H by comparing 1000 random image pairs that were aligned with both methods. Reprojection Flow was applied without alignment constraints for this comparison, which kept its function closer to that of ICP-H. The comparison divides the 1000 image pairs into eight intervals of time between surveys, in increments of 45 days, to show that the trend was unaffected by the variation in appearance.

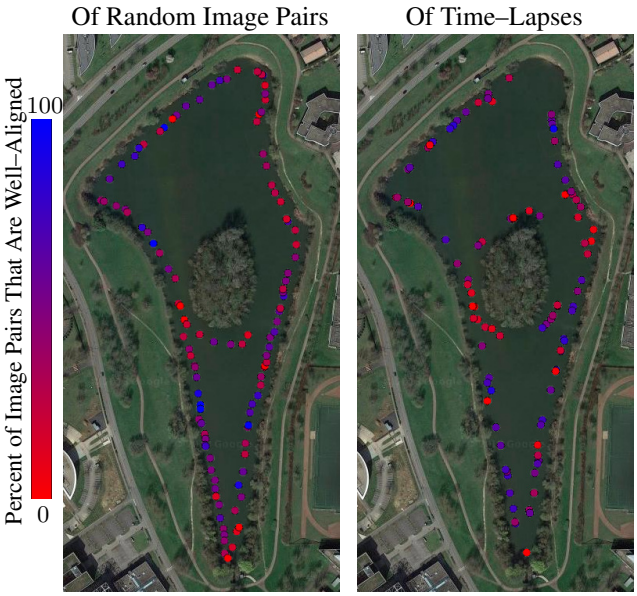

Figure 16. Alignment quality around Lake Symphony for the **left)** image pairs of Sec. 8.2 and the **right)** time–lapses of Sec. 8.5. The similarity of the results of the two sets indicates that our method may be robust to difficult variation in appearance at some locations. The satellite view is from Google Maps.

but some parts of one image pair aligned well. That made the trend in this figure different from that of Fig. 14.

8.4 Image Alignment Quality By Scene

We next plotted the image alignment accuracy by scene to determine by how much different scenes affected alignment performance, and where complete time–lapses (with an image from every survey) could be possible. The average image alignment quality of 43.8% suggested that complete time–lapses could not be produced unless the images aligned better at different scenes, which was likely. A cover set of the environment was identified for one survey (the cover set was acquired by applying the method of Griffith and Pradalier 2017, to the June 25, 2014 survey) and then each of the 1000 image pairs was added to the nearest scene (the position where the reference image had the min L_2 distance). If the scene had at least four image pairs, then the percent of well–aligned image pairs was plotted.

Image alignment quality varied substantially by scene, as shown on the left side of Fig. 16. Some scenes along the shoreline had many well–aligned images whereas other scenes had none. The scenes with the most well–aligned images were along the straights. Fewer well–aligned images were produced along curves in the path. Thus at certain locations our approach to dense correspondence showed robustness to difficult variation in appearance. Yet, the number of well–aligned image pairs did not demonstrate that our method was robust to a full year of variation in appearance, or if those were locations where a large number of image pairs were from sessions captured around the same time. We next tested whether this result held true across complete time–lapses with a year of variation in appearance.

8.5 Producing Time–Lapses

With a high likelihood of more complete time–lapses at some scenes in the environment, the next step was to create them. To create a time–lapse, a reference image was randomly chosen from a random survey, and then an image of the same scene from every other survey was selected and aligned using Reprojection Flow with constraints. The quality of the time–lapse was manually labeled. First the reference image and each image of the time–lapse were flickered to keep only well–aligned image pairs. Then the time–lapse was repeatedly scrolled through to keep only the images that added to it (also well–aligned). Two examples are shown in Figs. 17 and 18. The process was repeated 100 times.

The results show that the quality of the time–lapses was consistent with the image alignment quality of Sec. 8.2; although the time–lapse quality varied, some locations aligned particularly well. The quantitative time–lapse quality is shown in Fig. 19 and shown by place in the right of Fig. 16. Approximately a third of the time–lapses had about two thirds or more of well–aligned images. These time–lapses typically spanned all four seasons. The misaligned image pairs did not consistently have significantly more variation in appearance, and were not always from consecutive surveys. Instead, the reprojected map points appeared to mismatch the correct alignment (see Fig. 20 top).

Some effects of aligning a set of images into a time–lapse include the lack of variation in viewpoint and the addition of noise in the result. Before applying image alignment, a set of images of a scene was a time–lapse whose variation in viewpoint sometimes detracted from the collection. After, the noise added due to the alignment process sometimes detracted from it (see Fig. 20 bottom). Having accurate maps and very similar viewpoints helped minimize that noise to create visually smooth transitions. Noise often was, however, a side effect of image alignment.

Figure 17. Timelapse of one scene of Symphony Lake from 32 surveys captured between 2014 Jan. 6 and 2014 Dec. 22. The images were selected and aligned to the reference image using Reprojection Flow.

Figure 18. Timelapse of one scene of Symphony Lake from 37 surveys captured between 2014 Jan. 6 and 2014 Dec. 22. The images were selected and aligned to the reference image using Reprojection Flow.

8.6 Pose Error of Misaligned Image Pairs

Because any pose error could cause errors in the locations of reprojected map points and could lead to misaligned images, we next measured the magnitude of the pose error for misaligned image pairs taken from the time–lapse set. A misaligned image was selected for the evaluation if it appeared to share several strong features with the reference image, which simplified the labeling task. A map point in the reference image was selected and the corresponding point in the selected image was hand–labeled. After hand–labeling at least 15 correspondences, a one-way localization was performed, similar to that described in Sec. 5.4.2. The localized pose was used as the ground truth if the map points from the reference image projected onto their locations in the selected image. If the projected map points were incorrect, the set of hand–labeled correspondences was refined until they did or were discarded. The process was repeated for 100 misaligned image pairs.

Figure 21 shows that the pose error was nonnegligible for almost all of the misaligned images. The poses of misaligned images had a median translation error of 1.06 m and a median orientation error of 3.15 degrees. Pose error this high caused reprojected map points to be far off from their correct locations. Even for the pose with the least error of 12cm and 1/2 a degree, error in the reprojected map points was visible as slight misalignment of more distant scene contents. For that image, however, the variation in viewpoint was also a primary cause of misalignment. Although a foreground object was aligned well, the background behind it was pulled out of alignment. Images from more similar viewpoints with accurately projected map points aligned best.

9 Discussion

Our effort at creating and relying on geometric information and consistent maps was key for our framework to achieve data association and produce time–lapses across the year–long variation in appearance of a natural environment. Scene structure and geometric constraints helped mitigate the

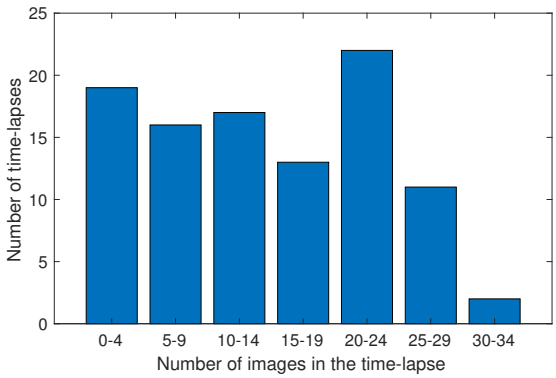

Figure 19. Success of 100 random time–lapses measured as how many images each one consisted of (after manually sorted). About a third of them had 66% or more well–aligned images. Most of the time–lapses were created from approximately 33 image alignments. Although Symphony Lake Dataset had 37 surveys from 2014, typically only about 33 captured the same scene.

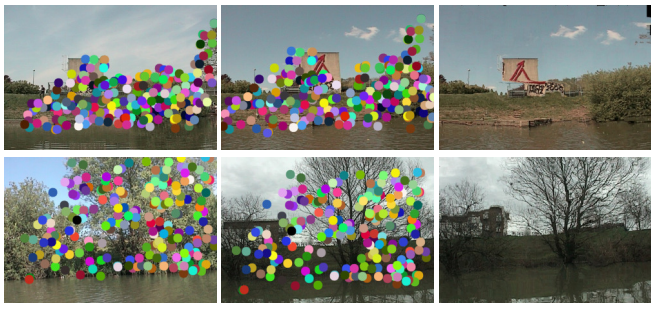

Figure 20. Noise in aligned images. **(top)** Map points from two surveys are projected onto the reference image (left) and the image to be aligned (middle). Their inconsistency caused the error of the aligned image (right), which otherwise had a strong appearance–based correspondence. **(bottom)** The alignment process added noise to the tree structure in the well–aligned image (right), even though the map point priors were consistent.

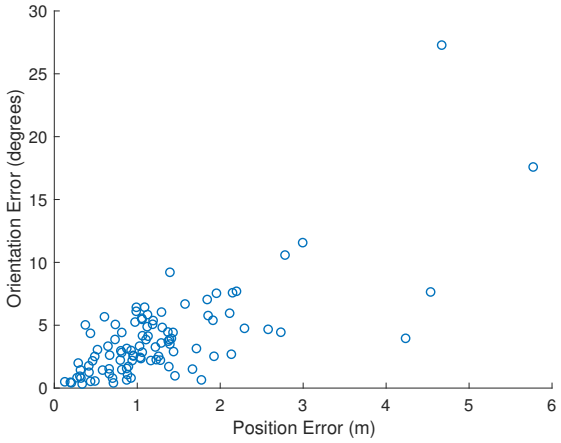

Figure 21. Pose error for 100 misaligned image pairs. The median error of the points here is 1.06 m and 3.15 degrees.

variation in appearance to provide correspondences between surveys. Reprojection Flow was then key for bringing more difficult image pairs into alignment. Because our approach was based on the use of geometric constraints to create map point correspondence priors, its accuracy was sometimes limited, however, by the consistency to which multiple sessions were aligned. Issues and limitations are described in Sec. 9.1, followed by a discussion of the parameters used with our dataset (Sec. 9.2).

9.1 Considerations for Ongoing Work

The final alignment when using Reprojection Flow is strongly influenced by the map point priors. Flexibility in image alignment is limited to the dispersion of the map points relative to the average flow. For an inconsistent map, reprojected map points would not correspond to scene content and would hold the map to the wrong places. For a consistent map, however, a small hypothesis space may help reduce perceptual aliasing. Making the rf constraint a function of reprojection error provided a way to balance the tightness of the anchor with other appearance information. If we used the same, strong prior for every map point, images may more likely be misaligned where the map points have very high reprojection error. Weaker priors may more likely give the alignment too much flexibility, increasing the likelihood of perceptual aliasing. We do not claim this mixture is the optimal one.

Although we showed robust data association due to the use of map point correspondence priors, our approach is also sensitive to the time elapsed between surveys because it relies on appearance–based data association to acquire a consistent map. If we had collected surveys less frequently, there may have been too few loop closures. As shown in Fig. 12, the July surveys had relatively few constraints to the others. A lack of temporal loop closures would have left drift between sessions. In that case, only the sessions within the range of appearance–based data association (three months) could have been aligned into time–lapses.

The shortage of constraints between some of the surveys in Fig. 12 indicates more robustness to the variation in appearance may be needed. Robustness could be best added through changes that lead to more loop closures with appearance–based data association, rather than by adding more filtering to remove outliers. If we tried to tighten parameters to keep only the best matching images, that could reduce the connectivity between surveys beyond what may be needed to get a consistent set of maps and trajectories.

A primary way to get more ISLCs may be to update SIFT Flow. SIFT features do not have condition invariance, unlike descriptors from some neural networks (e.g., Olid et al. 2018). When SIFT features are extracted at a single scale, as in SIFT Flow, they are also scale dependent. Scale and other changes in viewpoint may possibly be corrected by first applying a homography before SIFT Flow (Dong et al. 2017; Pradalier et al. 2019). For the specific case of the July surveys, which had a higher lake level that led to the varied viewpoint, adding a prealignment using a homography may help with acquiring more loop closures.

A few additional technical corrections could be made to expand this framework for more general use. Viewpoint selection may be improved using more constraints to help find the poses that best correspond to the same scene (e.g., add a pose constraint, or reconstruct a mesh to delineate foreground, background, and occluded features as in Lin et al. 2019). Also, our formulation of image alignment should be formulated to handle occlusions differently.

Currently, image alignment handles occlusions by filling in the areas of the image that become unoccluded. That is, dense correspondence is set to a reverse rather than to an onto mapping. A mapping 'onto' could leave holes in the image. Yet, leaving holes may be more accurate than replacing that content with nearby content of the same image. Currently, epipolar constraints may filter any map-point correspondences that would otherwise be within the filled-in occluded regions.

Two limitations are reiterated:

- Occlusions. A large change in viewpoint can lead to background map points and those on occluded objects holding the image in different ways than the foreground points.
- Non-discriminative appearance during the ISLC search. The map point correspondence priors could keep dense correspondence locked into a series of bad alignments if the appearance–based features of the scene are too uninformative to pull the dense correspondence out of misalignment.

9.2 Parameter values

A number of parameters were defined and tuned in the making of this framework. They are summarized in Table 1. Our framework has many parameters because it has many different steps. SIFT Flow and RANSAC were the two off-the-shelf methods. Factor graph optimization was implemented using GTSAM, whose noise models are left out. Each step was typically tested and tuned individually and then their parameter values were left as-is after integration.

Parameter values were found that applied well within the Symphony Lake Dataset. Some parameters were made adaptive if a particular setting was insufficient. For example, the inlier/outlier threshold at the end of Sec. 7 on multi–session optimization was initially a single value of six pixels, but that was inconsistent with the varied reprojection error after single–session SLAM. Other parameter values may change when this framework is applied to different datasets. In that case, the pixel value limits could be made proportional to the image resolution. However, most may have broader applicability.

Although the alignment constraints have more parameters than the other steps of our framework, our formulation in this paper improved upon that of prior work to make it more general. This paper introduced image alignment verification, which replaced our previous use of an alignment energy threshold (of 1120000) and an alignment consistency threshold (of 95%) to distinguish well–aligned images. In our prior work, those thresholds applied well to a 100m section of shore that we initially evaluated against. For larger stretches they were ineffective because those values do not robustly correspond to well–aligned images. The use of 19 iterations of two-cycle consistency is, however, retained because testing showed that the consistency typically converged (but not necessarily to 95% or more pixels) before that or not at all. The 95% threshold was set because the first layer of image alignment is an approximation, where a 95% consistency is, in our case, correct enough to proceed with the alignment of larger resolution layers.

Table 1. Summary of parameters for the different parts of our framework. Factor graph weights are omitted.

Sec. 5.1: Image Retrieval	
5 m, 20 deg.	max pose distance to consider an image for alignment
3	max consecutive attempts of using RF during the search without success
Sec. 5.2: SIFT Flow	
$\alpha = 255$, d $= 10200$	alignment energy function parameters
$\nu = 0.255$	
h $= 11{,}5{,}3{,}1$	hypothesis space size down the image pyramid
100	iterations of message passing
Sec. 5.3: Alignment constraints	
(3,3) pixels	image translation applied for alignment verification
40%	min proportion of matching correspondences at which an image alignment is verified
at most 19	iterations of two–cycle consistency
95% $\leq$ 1 pixel	stopping criterion at which the forward and reverse flows are consistent
16	multiplier weight for the cycle consistency term
2.5	value for the epipolar constraint term
Sec. 5.3.3, 5.4.1, and 6.3: Fundamental matrix estimation	
3 pixels	RANSAC reprojection threshold
0.999	probability the fundamental matrix is correct
Sec. 5.4.2: Localization	
15	min number of correspondences required for localization, the number used in each iteration of RANSAC for an initial estimate
100	iterations of RANSAC
6 pixels	max acceptable reprojection error of an inlier 3D-to-2D correspondence
Sec. 5.4.3: Bi-Directional Refinement	
at most 15	iterations of expectation maximization
40%	min proportion of inlier correspondences at which localization is successful
Sec. 5.4.4: Loop–closure verification	
25%	min image overlap required of two localized poses to verify them
6 pixels	max average reprojection error at which two loop–closures agree
Sec. 7: Multi–session optimization	
3	multiplier for the ISLC inlier/outlier threshold
0.01 m	convergence criterion as the change in the average median change in position
0.9	weight for the weighted update

More tuning was done to find the 40% threshold than to tune the $(3, 3)$ pixel shift we used to verify an alignment. We used the shift to address perceptual aliasing. Aliasing occurred in data association due to the reflective lake on the bottom of images and due to the similar shore contents to the sides of the images. Shifting the images was a way to identify whether perceptual aliasing occurred. The best threshold was found by inspecting the results.

10 Conclusion

This work provided a transform from multiple visual surveys of a natural environment into time–lapses, and subsequently produced several time–lapses for a year of surveys. Our framework's use of geometric information and consistent maps, integrated into a dense correspondence optimization, led to visual data association across significant variation in appearance. The ISLC search pipeline found a large number

of accurate loop closures, which created the connectivity needed to bring multiple surveys into alignment. Using a multi–session optimization algorithm that filtered outlier loop closures helped lead to a consistent map for a year of surveys. Although a long time–lapse at every location along the shore was not produced, our results show promise on how to obtain one from a consistent map of a natural environment and the dense correspondences of its images.

11 Future Work

Although our framework produced time–lapses for several scenes, higher map consistency would help get complete time–lapses at every scene. Improving appearance–based data association could help lead to a more consistent map. For example, upgrading SIFT Flow to use learned features (Kim et al. 2017a; Benbihi et al. 2019), or other features specifically designed and trained for data from a natural environment (Olid et al. 2018), upgrading the dense correspondence framework itself (with, e.g., 3DCC Zhou et al. 2016), or designing our own may be promising ways to boost image alignment performance. Another boost may be possible if the KLT feature tracks that provided the map points are replaced (with, e.g., Ilg et al. 2017; Wu and Pradalier 2018). Furthermore, Chahine and Pradalier (2018) demonstrated semi-dense visual SLAM on the Symphony Lake Dataset, which could later be used with ICP (similar to, e.g., Park et al. 2019) to provide another source of loop closures between surveys.

Our map–point priors could also act as a supervision signal in training a neural network to align images (as in e.g., Dong et al. 2018). If the alignment optimization is replaced by a neural network, the manual labeling function we applied in Sec. 8 (to determine the alignment quality and the pose error) could potentially be learned and then inferred automatically (analogous to flow and matchability of Zhou et al. 2016) (see also Kendall et al. 2015, for learning to infer pose).

Future work could also benefit from a ground truth dataset of localized poses. The known pose–transforms of Sec. 8.6 facilitated one evaluation in this paper, but they could be more useful. A ground truth set would facilitate a broader set of comparisons and evaluations of new algorithms applied to the Symphony Lake Dataset. This set could also help replace evaluation by hand–labeling the image alignment quality.

References

Agarwal P, Tipaldi GD, Spinello L, Stachniss C and Burgard W (2013) Robust map optimization using dynamic covariance scaling. In: *IEEE International Conference on Robotics and Automation*. pp. 62–69.

Arroyo R, Alcantarilla PF, Bergasa LM and Romera E (2015) Towards life-long visual localization using an efficient matching of binary sequences from images. In: *IEEE International Conference on Robotics and Automation*. pp. 6328–6335.

Beall C and Dellaert F (2014) Appearance-based localization across seasons in a metric map. In: *IEEE/RSJ IROS Workshop on Planning, Perception and Navigation*.

Benbihi A, Geist M and Pradalier C (2019) Elf: Embedded localisation of features in pre-trained cnn. In: *IEEE International Conference on Computer Vision*.

Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I and Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics* 32(6): 1309–1332.

Carlevaris-Bianco N and Eustice RM (2013) Generic factor-based node marginalization and edge sparsification for pose-graph slam. In: *IEEE International Conference on Robotics and Automation*. pp. 5748–5755.

Carlone L and Calafiore GC (2018) Convex relaxations for pose graph optimization with outliers. *IEEE Robotics and Automation Letters* 3(2): 1160–1167.

Carlone L, Kira Z, Beall C, Indelman V and Dellaert F (2014) Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In: *IEEE International Conference on Robotics and Automation*. pp. 4290–4297.

Chahine G and Pradalier C (2018) Survey of monocular slam algorithms in natural environments. In: *IEEE Conference on Computer and Robot Vision*. pp. 345–352.

Chen Z, Maffra F, Sa I and Chli M (2017) Only look once, mining distinctive landmarks from ConvNet for visual place recognition. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 9–16.

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

Corke P, Paul R, Churchill W and Newman P (2013) Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localization. In: *IEEE International Conference on Intelligent Robots and Systems*. pp. 2085–2092.

Cummins M and 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 (2012) Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM. URL `https://research.cc.gatech.edu/borg/sites/edu.borg/files/downloads/gtsam.pdf`.

Diego F, Ponsa D, Serrat J and López AM (2011) Video alignment for change detection. *IEEE Transactions on Image Processing* 20(7): 1858–1869.

Dong J, Boots B, Dellaert F, Chandra R and Sinha S (2018) Learning to align images using weak geometric supervision. In: *IEEE International Conference on 3D Vision*. IEEE, pp. 700–709.

Dong J, Burnham JG, Boots B, Rains G and Dellaert F (2017) 4D crop monitoring: Spatio-temporal reconstruction for agriculture. In: *IEEE International Conference on Robotics and Automation*. pp. 3878–3885.

Dong J, Nelson E, Indelman V, Michael N and Dellaert F (2015) Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5807–5814.

Dvijotham K, Gowal S, Stanforth R, Arandjelovic R, O'Donoghue B, Uesato J and Kohli P (2018) Training verified learners with

learned verifiers. *arXiv preprint arXiv:1805.10265* .

Dymczyk M, Lynen S, Cieslewski T, Bosse M, Siegwart R and Furgale P (2015) The gist of maps-summarizing experience for lifelong localization. In: *IEEE International Conference on Robotics and Automation*. pp. 2767–2773.

Engel J, Schöps T and Cremers D (2014) Lsd-slam: Large-scale direct monocular slam. In: *European Conference on Computer Vision*. Springer, pp. 834–849.

Facil JM, Olid D, Montesano L and Civera J (2019) Condition-invariant multi-view place recognition. *arXiv* ArXiv:1902.09516.

Ferguson D, Morris A, Haehnel D, Baker C, Omohundro Z, Reverte C, Thayer S, Whittaker C, Whittaker W, Burgard W et al. (2004) An autonomous robotic system for mapping abandoned mines. In: *Advances in Neural Information Processing Systems*. pp. 587–594.

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

Garg S, Suenderhauf N and Milford M (2018) LoST? Appearance-invariant place recognition for opposite viewpoints using visual semantics. *arXiv preprint arXiv:1804.05526* .

Gomez-Ojeda R, Lopez-Antequera M, Petkov N and Gonzalez-Jimenez J (2015) Training a convolutional neural network for appearance-invariant place recognition. *arXiv* ArXiv:1505.07428.

Graham MC, How JP and Gustafson DE (2015) Robust incremental slam with consistency-checking. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 117–124.

Griffith S, Chahine G and Pradalier C (2017) Symphony lake dataset. *International Journal of Robotics Research* 36: 1151–1158.

Griffith S, Dellaert F and Pradalier C (2015) Robot-Enabled Lakeshore Monitoring Using Visual SLAM and SIFT Flow. In: *RSS Workshop on Multi-View Geometry in Robotics*.

Griffith S and Pradalier C (2016) Reprojection flow for image registration across seasons. In: *British Machine Vision Conference*.

Griffith S and Pradalier C (2017) Survey registration for long–term natural environment monitoring. *Journal of Field Robotics* 34(1): 188–208.

Griffith S, Sukhoy V and Stoytchev A (2011) Using sequences of movement dependency graphs to form object categories. In: *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 715–720.

Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A and Brox T (2017) Flownet 2.0: Evolution of optical flow estimation with deep networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2.

Indelman V, Nelson E, Michael N and Dellaert F (2014) Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 593–600.

Johannsson H, Kaess M, Fallon M and Leonard JJ (2013) Temporally scalable visual slam using a reduced pose graph. In: *IEEE International Conference on Robotics and Automation*. pp. 54–61.

Jones ES and Soatto S (2011) Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *International Journal of Robotics Research* 30(4): 407–430.

Kaess M, Johannsson H, Roberts R, Ila V, Leonard JJ and Dellaert F (2012) iSAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research* 31(2): 216–235.

Kendall A, Grimes M and Cipolla R (2015) Posenet: A convolutional network for real-time 6-DOF camera relocalization. In: *IEEE International Conference on Computer Vision*. pp. 2938–2946.

Khaliq A, Ehsan S, Milford M and McDonald-Maier K (2018) A holistic visual place recognition approach using lightweight CNNs for severe viewpoint and appearance changes. *arXiv preprint arXiv:1811.03032* .

Kim J, Liu C, Sha F and Grauman K (2013) Deformable spatial pyramid matching for fast dense correspondences. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. pp. 2307–2314.

Kim S, Lin S, Jeon SR, Min D and Sohn K (2018) Recurrent transformer networks for semantic correspondence. In: *Advances in Neural Information Processing Systems*. pp. 6127–6137.

Kim S, Min D, Ham B, Jeon S, Lin S and Sohn K (2017a) FCSS: Fully convolutional self-similarity for dense semantic correspondence. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 6560–6569.

Kim S, Min D, Lin S and Sohn K (2017b) DCTM: Discrete-continuous transformation matching for semantic flow. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4529–4538.

Krajnik T, Cristóforis P, Nitsche M, Kusumam K and Duckett T (2015) Image features and seasons revisited. In: *IEEE European Conference on Mobile Robots*. pp. 1–7.

Latif Y, Cadena C and Neira J (2013) Robust loop closing over time for pose graph slam. *International Journal of Robotics Research* 32(14): 1611–1626.

Le H and Milford M (2018) Large scale visual place recognition with sub-linear storage growth. *arXiv preprint arXiv:1810.09660* .

Lin CH, Wang O, Russell BC, Shechtman E, Kim VG, Fisher M and Lucey S (2019) Photometric mesh optimization for video-aligned 3d object reconstruction. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

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*. IEEE, pp. 90–97.

Liu C, Yuen J and Torralba A (2011) SIFT flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(5): 978–994.

Lopez-Antequera M, Gomez-Ojeda R, Petkov N and Gonzalez-Jimenez J (2017) Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters* 92: 89–95.

Lowe D (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.

Lowry S and Milford MJ (2016) Supervised and unsupervised linear learning techniques for visual place recognition in changing environments. *IEEE Transactions on Robotics* 32(3): 600–613.

Lowry S, Sunderhauf N, Newman P, Leonard JJ, Cox D, Corke P and Milford MJ (2016) Visual place recognition: A survey. *IEEE Transactions on Robotics* 30(1): 1–19.

Lucas BD and Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: *International Joint Conference on Artificial Intelligence*. pp. 674–679.

Martin-Brualla R, Gallup D and Seitz SM (2015a) 3d time-lapse reconstruction from internet photos. In: *IEEE International Conference on Computer Vision*. pp. 1332–1340.

Martin-Brualla R, Gallup D and Seitz SM (2015b) Time-lapse mining from internet photos. *ACM Transactions on Graphics* 34(4): 62.

McDonald J, Kaess M, Cadena C, Neira J and Leonard JJ (2013) Real-time 6-DOF multi-session visual SLAM over large-scale environments. *Robotics and Autonomous Systems* 61(10): 1144–1158.

McManus C, Upcroft B and Newman P (2014) Scene signatures: Localized and point-less features for localization. In: *Robotics: Science and Systems*.

Milford M (2013) Vision-based place recognition: How low can you go? *International Journal of Robotics Research* 32(7): 766–789.

Milford M, Firn J, Beattie J, Jacobson A, Pepperell E, Mason E, Kimlin M and Dunbabin M (2014) Automated sensory data alignment for environmental and epidermal change monitoring. In: *Australasian Conference on Robotics and Automation*. Australian Robotic and Automation Association, pp. 1–10.

Milford M and Wyeth G (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.

Mühlfellner P, Bürki M, Bosse M, Derendarz W, Philippsen R and Furgale P (2016) Summary maps for lifelong visual localization. *Journal of Field Robotics* 33(5): 561–590.

Naseer T, Burgard W and Stachniss C (2018) Robust visual localization across seasons. *IEEE Transactions on Robotics* 34(2): 289–302.

Neubert P, Sunderhauf N and Protzel P (2013) Appearance change prediction for long-term navigation across seasons. In: *IEEE European Conference on Mobile Robots*. pp. 198–203.

Ni K, Steedly D and Dellaert F (2007) Tectonic SAM: Exact, out-of-core, submap-based SLAM. In: *IEEE International Conference on Robotics and Automation*. pp. 1678–1685.

Olid D, Fácil JM and Civera J (2018) Single-view place recognition under seasonal changes. In: *IROS 2018 workshop on Planning, Perception, and Navigation for Intelligent Vehicles*.

Oliva A and Torralba A (2006) Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research* 155: 23–36.

Olson E and Agarwal P (2013) Inference on networks of mixtures for robust robot mapping. *International Journal of Robotics Research* 32(7): 826–840.

Park C, Kim S, Moghadam P, Guo J, Sridharan S and Fookes C (2019) Robust photogeometric localization over time for map-centric loop closure. *IEEE Robotics and Automation Letters* 4(2): 1768–1775.

Pradalier C, Aravecchia S and Pomerleau F (2019) Multi-session lake-shore monitoring in visually challenging conditions. In: *12th Conference on Field and Service Robotics*.

Sand P and Teller S (2004) Video matching. *ACM Transactions on Graphics* 23(3): 592–599.

Sattler T, Leibe B and Kobbelt L (2011) Fast image-based localization using direct 2D-to-3D matching. In: *IEEE International Conference on Computer Vision*. pp. 667–674.

Sibley G, Mei C, Reid I and Newman P (2010) Vast-scale outdoor navigation using adaptive relative bundle adjustment. *International Journal of Robotics Research* 29(8): 958–980.

Sivic J and Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: *IEEE International Conference on Computer Vision*. pp. 1470–1477.

Stoytchev A (2009) Some basic principles of developmental robotics. *IEEE Transactions on Autonomous Mental Development* 1(2): 122–130.

Stumm E, Mei C and Lacroix S (2013) Probabilistic place recognition with covisibility maps. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4158–4163.

Sünderhauf N and Protzel P (2012) Switchable constraints for robust pose graph slam. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1879–1884.

Sunderhauf N, Shirazi S, Jacobson A, Dayoub F, Pepperell E, Upcroft B and Milford M (2015) Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. In: *Robotics: Science and Systems*.

Sutton R (2001) Verification: The Key to AI. http://webdocs.cs.ualberta.ca/ sutton/IncIdeas/KeytoAI.html.

Thrun S, Thayer S, Whittaker W, Baker C, Burgard W, Ferguson D, Hähnel D, Montemerlo M, Morris A, Omohundro Z et al. (2004) Autonomous exploration and mapping of abandoned mines. *IEEE Robotics & Automation Magazine* 11(4): 79–91.

Vysotska O and Stachniss C (2016) Lazy data association for image sequences matching under substantial appearance changes. In: *IEEE International Conference on Robotics and Automation*. pp. 213–220.

Wu X and Pradalier C (2018) Illumination robust monocular direct visual odometry for outdoor environment mapping. In: *IEEE International Conference on Robotics and Automation*.

Zhou T, Krahenbuhl P, Aubry M, Huang Q and Efros AA (2016) Learning dense correspondence via 3D-guided cycle consistency. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 117–126.

Zhou T, Lee YJ, Yu SX and Efros AA (2015) FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. pp. 1191–1200.