Autonomous Robots (2022) 46:437–449
https://doi.org/10.1007/s10514-021-10031-8

Image based Localization under large perspective difference between Sfm and SLAM using split sim(3) optimization

Deepak Rajamohan[1] · Jonghyuk Kim[2] · Matt Garratt[1] · Mark Pickering[1]

Received: 4 March 2021 / Accepted: 16 December 2021 / Published online: 11 February 2022

Abstract

Image based Localization (IbL) uses both Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) data for accurate pose estimation. However, under conditions where there is a large perspective difference between the SfM images and SLAM keyframes, the SfM-SLAM co-visibility graph becomes sparse. As a result, the scale drift can increase especially when using monocular SLAM as part of the IbL framework. The drift rarely gets corrected at loop closure due to its large magnitude. We propose a split affine transformation approach that uses SfM-SLAM information along with Sim(3) optimization to minimize the scale drift. Experiments are performed using an image dataset collected in a campus environment with different trajectories, showing the improvement in scale drift correction with the proposed method. The SLAM data was collected close to plainly textured structures like buildings while SfM images were captured from a larger distance from the building facade which leads to a challenging navigation scenario in the context of IbL. Localizing mobile platforms moving close to buildings is an example of such a case. The paper positively impacts the widespread use of small autonomous robotic platforms, which is to perform an accurate outdoor localization under urban conditions using only a monocular camera.

Keywords Image based localization · Monocular SLAM · Structure from motion · Trajectory optimization

1 Introduction

Urban canyons render GPS signals inaccurate for use in drone or robotic platform navigation due to multi-path reflections and signal blocking. Camera sensor based Simultaneous Localization and Mapping (Visual-SLAM) algorithms are widely used to mitigate the disadvantages of GPS but suffer from accumulating drift. It leads to the infusion of smooth but increasing noise in 3D camera pose estimation. Since in many practical cases the navigable zones can be pre-mapped, one can combine information from a previously known structure from motion (SfM) map of the region and a live SLAM map to improve the accuracy of 3D pose estimation. This approach is called Image based Localization (IbL) in the literature, a brief overview of which could be found in Wu *et al.*'s paper (Yihong et al. 2018). Unlike pure SLAM algorithms, IbL methods can provide global localization if the SfM map used is spatially consistent with respect to a global map. Given that the SfM map is pre-generated, one could afford to use time consuming optimization processes to achieve very high accuracy and then use them for IbL. Hence autonomous applications like robo-taxies, drone and ground based last-mile delivery systems could use IbL approaches to improve operational safety under dense urban conditions.

The 3D points in the SfM model correspond to a corner feature identified on an SfM image called observation. There could be multiple such observations as the same corner feature could be visible across multiple SfM images. By averaging the feature descriptor of all the observations, a single value is assigned to the corresponding 3D point as used in Mur-Artal et al. (2015). The 3D points can now be independently matched to the corner features identified on the platform's SLAM keyframe image, leading to SfM-SLAM 3D-2D matches. For a calibrated camera, these matches provide a geometric relationship that can be exploited directly

✉ Deepak Rajamohan
deepak.rajamohan@adfa.edu.au

Jonghyuk Kim
jonghyuk.kim@uts.edu.au

Matt Garratt
m.garratt@adfa.edu.au

Mark Pickering
M.Pickering@adfa.edu.au

[1] University of New South Wales, Canberra, Australia

[2] University of Technology Sydney, Sydney, Australia

to obtain pose information and are called direct methods in the literature. Alternatively, indirect methods use the features extracted from the live camera image, to search a bag-of-features dictionary and retrieve similar looking images from the SfM map. Hence unlike direct methods, only a coarse localization is achieved within the SfM map. The focus of this paper leans towards the direct methods as it can be used to continuously refine the SLAM trajectory and associated camera poses using the SfM-SLAM 3D-2D matches.

A recent survey (Piasco et al. 2018) discusses the use of various local and global feature types in the IbL algorithm. However, just choosing a robust feature that is resilient to time induced appearance changes and with good affine invariance properties does not suffice in building a practical IbL solution, if there exists large perspective differences SfM and SLAM data. Example for such a scenario would be when localizing pedestrians or delivery robots on foot path close to buildings, whilst the pre-existing map was generated by street-view cars or drones that cannot fly close to buildings. This is accompanied by scale difference and the sparseness of the SfM data, adding to the challenges in localization and trajectory optimization. Sparse SfM data is an anticipated problem when implementing large-scale solutions or online solutions where the SfM map has to reside on the mobile platform. In such cases, common observations between SfM and SLAM is not always available in abundance. To the best of our knowledge ,there has not been many efforts in tackling the mentioned challenges collectively in the context of IbL. We also handle the varying scale drift problem by splitting the trajectory. The key contributions of our work are:

- A novel *split* Sim(3) optimization to handle the varying and sparse co-visibility between SfM and SLAM.
- Demonstration of the IbL method in a campus environment where SfM and SLAM have a large perspective difference.

The trajectory optimization in the proposed framework is achieved in two steps. First, a coarse rigid 3D affine transformation between SfM and SLAM coordinate systems, that is known either by rough GPS coordinates or by one-time calibration, is refined using only the global position information derived from SfM data using EPnP algorithm. Then as the SLAM front end and the background optimization continues, a trajectory correction procedure which includes optimization using additional SfM-SLAM constraints and scale drift correction, is triggered periodically as a second step. Here, the available trajectory is split into smaller sections depending on the availability of 3D-2D match information and changes in affine transformation along the trajectory. The affine transformation for these individual sections are then estimated and refined. Both E-PnP based position constraints and SfM-SLAM reprojection constraints are used to refine the individual sections. The keyframes in each of these sections are then collectively used to correct for scale drift using Sim(3) optimization. The remainder of the paper is organized as follows. Section 2 discusses the contribution of this work in the context of existing literature. Section 3 explains the technicalities of the proposed IbL framework while Sect. 4 discusses the experimental setup and results.

2 Related work

SLAM and IbL methods have different roots but recent works have synergised these two classes of algorithms for better localization and trajectory optimization. SLAM algorithms have evolved from filtering based approaches (Huang and Dissanayake 2006; Castellanos et al. 2007; Paz et al. 2008) that perform an incremental pose estimation to modern trajectory optimization methods, introduced by Lu and Milios in the paper (Feng and Milios 1997). The famous PTAM by Klein and Murray (2007) is one such work that optimizes the geometric relationships among the available keyframes and the map. The paper by Cadena et al. (2016) talks about this evolution in detail. IbL methods on the other hand started as a purely single view localization technique (Irschara et al. 2009) within an SfM map. Hence one of the main research directions was the visual place recognition problem. In the realm of direct IbL methods, the common algorithms used to estimate single camera view pose were the versions of PnP algorithm. A minimum of 3 SfM 3D points whose 2D projection is known are required for estimation using the P3P algorithm (Gao et al. 2003), but most of the literature makes use of the Efficient-PnP (EPnP) algorithm (Lepetit et al. 2009) which uses four virtual points calculated from the n 3D points ($n \geq 4$). The general idea is to solve a system of linear equations by substituting the 3D point (x, y, z) and the corresponding 2D projection (u, v) in Equ. 1 with known intrinsic camera parameters (f_x, f_y, u_0, v_0). This is similar to a camera calibration problem and the 6-DoF camera pose parameters $(r_{11} - r_{33}, t_1, t_2, t_3)$ can be estimated given sufficient number of SfM-SLAM 3D-2D matches. Here λ is the scale factor. Depending on the quality of the features used for estimating the matches, the data could be plagued with outliers requiring a RANSAC version of the algorithm (Albl et al. 2016; Kneip et al. 2011; Larsson et al. 2017; Kukelova et al. 2013). These single view IbL methods were eventually used for continuous localization in later methods (Middelberg et al. 2014; Ventura et al. 2014).

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

Similar to SLAM literature, some authors have approached continuous localization within a global SfM map using Kalman filtering frameworks (Lim et al. 2012; DuToitet al. 2017). Middelberg et al. (2014) work is an example for the use of SfM data within a PTAM inspired background optimization approach to estimate a better trajectory. This and similar IbL methods like (Iwami et al. 2018; Ramalingam et al. 2010; Arth et al. 2015; Germain et al. 2019; Geppert et al. 2019; Mur-Artal and Tardós 2017; Stenborg et al. 2018), are an extension of SLAM that uses additional information to solve the localization problem and are discussed in sect. 2.1. They formulate the problem as the localization of sequence of images within an SfM map rather than localizing one camera view at a time. Section 2.1.3 talks about a subclass of methods in sequence localization that focus on drift correction using both SfM and SLAM data. Though, the proposed method uses the SLAM keyframe sequence along with PnP poses to align the trajectory similar to the methods cited above, it does the alignment periodically, rather than adopting an online approach. Hence, the proposed method also falls under the research of multi agent SLAM where multiple agents explore a scene with the maps generated by an agent available to the other at certain intervals. A single agent mapping at different points of time is also a related scenario. Section 2.2 briefly covers such related literature.

2.1 Localization of image sequence

2.1.1 Feature choice

The quality and quantity of the SfM-SLAM feature matches are key to the success of an IbL framework. One of the important problems is that the SfM and SLAM images could show considerable differences in appearance, angle of view and scale. The survey paper by Piasco et al. (2018) provides a detailed account of different feature types that are used to mitigate scale, orientation, and illumination changes. Several point feature types like SIFT (Lowe 2004), SURF(Bay et al. 2006), ORB(Rublee et al. 2011) and BRISK(Leutenegger et al. 2011) have been tried independently (Tareen et al. 2018) or in combination with geometric features like lines that connect two point features (Ferber et al. yyy). Efficient, long term matching across seasonal changes has be attempted by semantic segmentation based visual localization (Stenborg et al. 2018). Following the popular trend, Convolutional Neural Network (CNN) based approaches like (Arandjelovic et al. 2016; Gordo et al. 2017; Radenović et al. 2016; Zhou et al. 2020) have been used for visual localization. They use efficient global descriptors that consider the entire image for matching between the query image and the database of images. However, the use of image sequences should help in avoiding the use of state-of-the-art but computationally expensive features, as claimed by the authors of Stenborg et al. (2020). They propose that weaker algorithms extracted SIFT features are sufficient for establishing quality SfM-SLAM relationship. In fact, feature selection depends on the nature of the application which determines the type of image data handled. For example, the work described in Iwami et al. (2018) deals with SLAM and SfM images captured approximately from the same distance to the buildings and hence a faster ORB feature is used. However, to handle large perspective differences, SIFT features could be a better option as chosen in this work.

2.1.2 Additional constraints

Improving the accuracy of sequence localization by using new optimization frameworks that make use of a variety of constraints is another common research area. Efforts have been made to use features derived from the use of new sensors that lead to additional constraints. Ramalingam et al. (2010) proposed an IbL method using images of an upward-looking omni-directional camera for navigating dense urban canyons where a city-scale 3D model is available. The skyline of most places are unique and they extract an edge based feature that can be matched to the skyline feature generated from the 3D model. Geppert et al. (2019) use a multi-camera system that has cameras of varying field of view and direction of view to establish pose priors and then use a visual inertial odometry pipeline for pose refinement. Borges et al. (2010) uses 3D edge maps from a laser point cloud to match with edges of the live camera image and provide a solution for industrial transport vehicles in both indoor and outdoor conditions. Arth et al. (2015) propose an IbL approach using 2.5D maps that are widely available from different mapping platforms like Google. The straight lines observed on the camera are projected on to the un-textured building blocks from the 2.5D map. Assuming that the observed straight lines are either horizontal or vertical, a 2D-3D matching process constrains the camera pose alignment. There has also been a number of work (Lynen et al. 2015; Lynen et al. 2020; Mur-Artal and Tardós 2017) that uses inertial sensors to help constrain the background optimization process in addition to the visual odometry constraints of the sequence.

2.1.3 Focus on trajectory drift correction

In general, the sequence localization based methods determine an initial alignment (Iwami et al. 2018; Middelberg et al. 2014; Ventura et al. 2014) between the SfM and SLAM coordinate system. Consecutively, the 3D-2D matches provide additional geometric constraints that are then used for continuous re-alignment of every new image of the sequence. The use of such constraints in the background optimization process of the SLAM or visual odometry algorithm inherently (Middelberg et al. 2014; Stenborg et al. 2020;

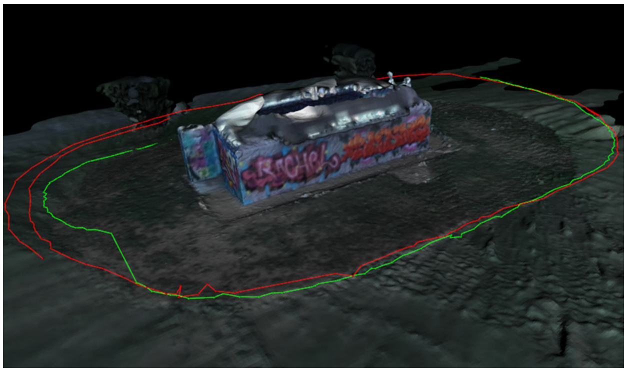


Fig. 1 Graffiti dataset trajectory comparison - The red line represents a SfM trajectory generated out of SfM images, while the green shows a SLAM trajectory. The SLAM trajectory is transformed to SfM coordinates for the comparison (Color figure online)

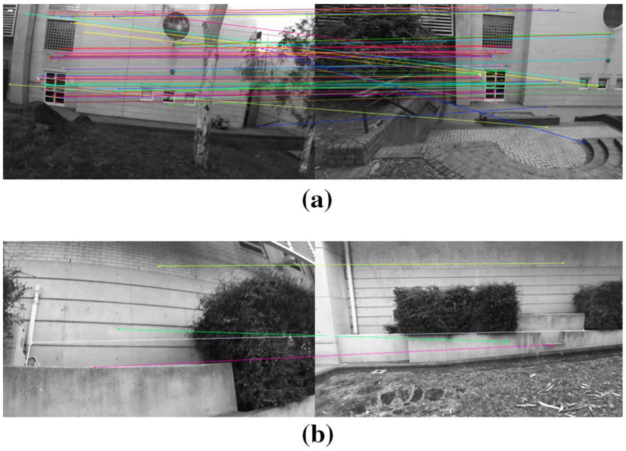


Fig. 2 SfM-SLAM 3D-2D matches (not 2D-2D feature matches): **a** SfM image matched to a SLAM keyframe at a similar pose, and **b** SfM image from a distance matched to SLAM keyframe close to building showing fewer matches

Geppert et al. 2019) avoids drift, provided, there is enough SfM-SLAM matches. Drift depends on the navigating environment and some methods do explicitly talk about drift handling Iwami et al. (2018) while demonstrating with showcased datasets. The use of sparse SfM data leads to small number of matches. With an aim of avoiding drift, (Geppert et al. 2019) try to maximize these matches by using a multi-camera system while using a sparse SfM map. Alternatively, Stenborg et al. (2020) uses a generalized camera baseline (Pless et al. 2003) to constrain the matching process. However, monocular SLAM based IbL approaches that use sparse SfM maps, inevitably suffers from drift especially when navigating close to buildings. Figure 2b shows sparse matches between a SLAM keyframe image (left) and an SfM image (right). Such keyframes belong to sections of the SLAM trajectory that have few 3D-2D matches due to high perspective differences. SLAM keyframes from the section of the trajectory which is closer to the SfM images show abundant matches as seen in Fig. 2a.

The co-visibility graph of a SLAM trajectory is defined by a matrix that explains how many landmark 3D points are shared between each keyframe view. A trajectory with a dense co-visibility graph is less likely to drift as the common information between the keyframes directs the background optimization process to a unique solution. A favourable scenario where this can occur is when the camera moves around a feature-rich structure. The graffiti dataset shown in Fig. 1 is one such example where the inherent drift in the SLAM trajectory is minimal. Adding to that, the perspective difference between the SfM and SLAM data of the graffiti dataset is also low. On the other hand the building dataset, used in this paper faces many of the challenges discussed in sect. 1 leading to drift in SLAM trajectory. The SfM-SLAM co-visibility graph visualized in Fig. 3 shows a sparse and dense overlap of SfM and SLAM information for the graffiti and building dataset respectively. The distribution of SfM-SLAM matches and the sparse SfM-SLAM co-visibility graphs shown in Fig. 3a and b respectively, explains the cause of 7Dof drift in the building dataset trajectory including scale drift. Scale drift correction in the context of IbL has not been explored much in the literature. Recently (Iwami et al. 2018) proposed a framework similar to the one proposed here, that uses ORB-SLAM for odometry and scale drift correction by using geotagged images from a google-street view dataset. The high pose accuracy of the geo-tagged images are attributed to SfM algorithms (Klingner et al. 2013) that can be used to refine their inertial sensor based pose estimation. A pose graph optimization in Sim(3) manifold corrects the 7DoF drift using the SfM-SLAM geometric constraints. The geo-tagged images are the counterparts of the globally localized PnP poses available in the proposed work. However, unlike the geo-tagged images that have accurate position information in the context of the application discussed in Iwami et al. (2018) , the PnP poses are noisy and cannot be used independently as pivots in the optimization process.

Overall, in the past, very few attempts have been made to address scale drift in the context of IbL. The proposed framework differs from the related literature by using the information whenever abundantly available by using a split affine strategy to correct for scale drift.

2.2 Multi-agent SLAM

As the proposed method uses accumulated trajectory for optimization instead of online optimization of individual poses like the ones discussed in Sect. 2.1, it is similar to a multi-agent SLAM problem. Multi-agent SLAM or collaborative SLAM deals with the problem of fusing maps generated by independent mapping agents. A pre-existing SfM map used can also be considered as a map generated by the same agent

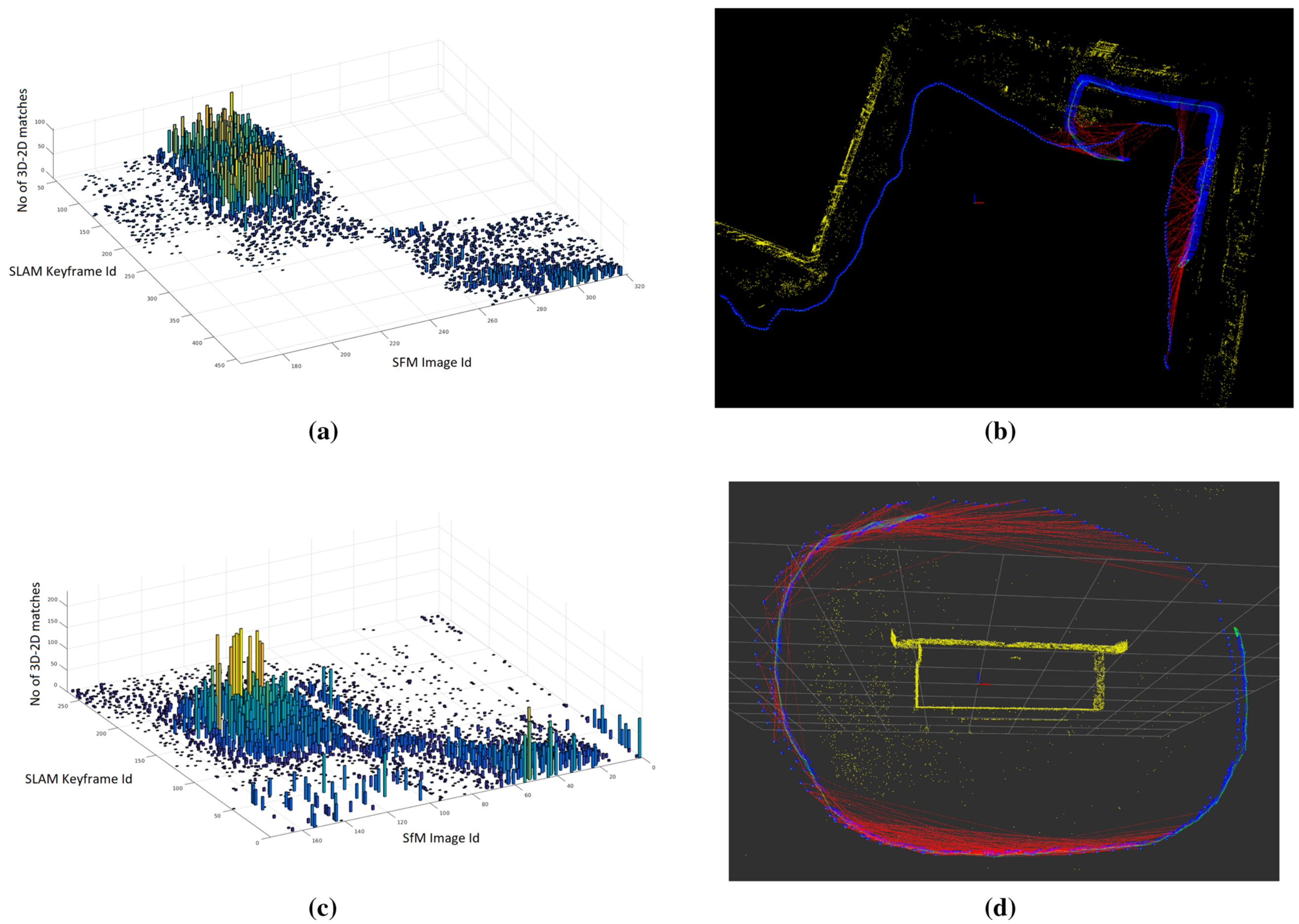


Fig. 3 **a**,**c** SLAM-SLAM match distribution along SLAM trajectory for building and graffiti dataset respectively, **b**,**d** Corresponding SfM-SLAM co-visibility graph (edges shown in red)(Color figure online)

at an earlier time (McDonald et al. 2013). Most methods follow a centralized approach(Schmuck and Chli 2019) where the maps from each agent is periodically sent through to a server. However, methods also use a decentralized architecture where the maps generated by agents are available to each other (Carrillo-Arce et al. 2013; Cieslewski et al. 2018) periodically and would draw parallels to the proposed work which interacts with the locally available SfM maps from time to time. In such methods the inter agent loop closure events gather common information between the maps of two different agents and optimally fuse them. The proposed method triggers the SfM-SLAM loop closure event periodically as the SLAM data always revisits the known SfM map. While centralized (Howe and Novosad 2005) and decentralized (Carrillo-Arce et al. 2013) filtering approaches for multi-SLAM, have been proposed that implicitly estimate the relative transformation between the local maps, other works (Bosse et al. 2004) model the relationship between the local maps as a graph and optimizes using the edge constraints. Similarly, the periodic trajectory correction step in the proposed method reduces the re-projection error between the SfM and SLAM data using Lavenberg-Marquardt (LM) background optimization. Thus, the proposed method uses localization concepts from both sequence localization and multi-agent SLAM literature discussed above and can be considered as a hybrid method.

3 Methodology

The proposed IbL framework in Fig. 4 shows the processing pipeline that is tailored to solve the navigation problem with the challenges discussed in the previous sections. It uses the SfM and SLAM data to first estimate optimal affine transformations for different trajectory sections and later uses them to address the scale drift. This section explains the individual modules of the proposed framework.

3.1 Affine transformation cost functions

As mentioned in the previous section, in order to register the SLAM trajectory on an SfM map, one needs to first apply a

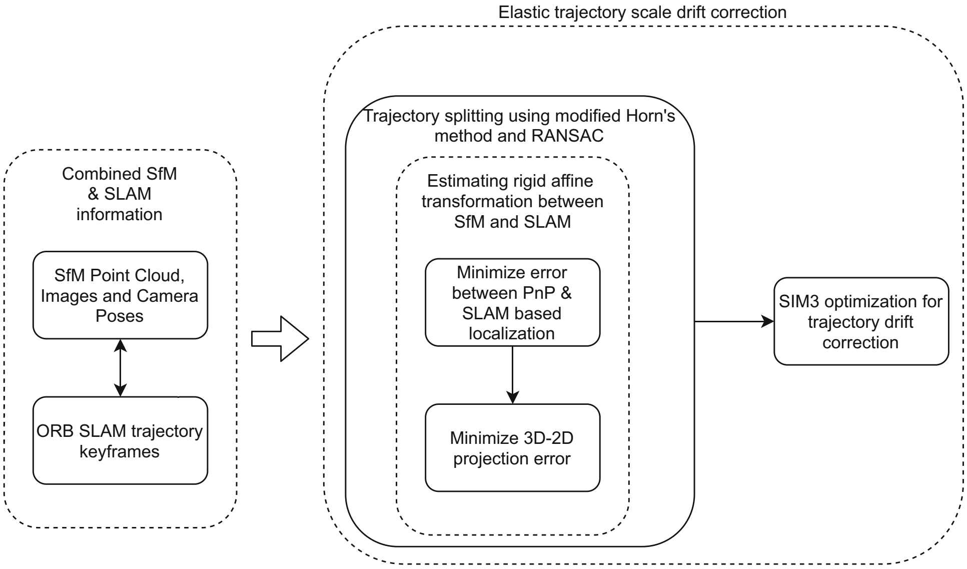

Elastic trajectory scale drift correction
Combined SfM & SLAM information
SfM Point Cloud, Images and Camera Poses
ORB SLAM trajectory keyframes
Trajectory splitting using modified Horn's method and RANSAC
Estimating rigid affine transformation between SfM and SLAM
Minimize error between PnP & SLAM based localization
Minimize 3D-2D projection error
SIM3 optimization for trajectory drift correction

Fig. 4 Proposed Image based Localization framework showing the steps in periodic drift correction

rigid affine transformation to the SLAM coordinate system. The approach proposed here utilizes a set of cost functions similar to Middelberg et al. (2014) to estimate the rigid affine transformation between the SLAM coordinate system and the SfM coordinate system. However, the process is adapted to use ORB features for SLAM trajectory and GPU-SIFT algorithm (Wu yyy) extracted SIFT features for SfM-SLAM matches. In addition, a set of procedural steps are introduced to account for the drift in the trajectory by introducing elasticity in the tranformation and these will be discussed in Sect. 3.2.

In this paper, the SLAM and SfM coordinate systems are referred to as local and global coordinate systems respectively. The cost functions in Eqs. 2 and 3 utilize two different relationship between the local and global coordinates to arrive at the optimal affine transformation parameters. The first cost function uses the 3D to 3D match between the local and global coordinate systems. The ORB SLAM provides an incremental estimate of the keyframe pose and it can be justified to assume that the drift in the SLAM trajectory is negligible within a small section of the trajectory. The keyframe positions in local coordinate system are estimated using these poses and are referred to as ${t_i}^L$. Their corresponding global positions ${t_i}^G$ are estimated using the E-PnP algorithm. A coarse affine transformation is found and then minimized using a cost function that involves ${t_i}^L$ and ${t_i}^G$. It reduces the square of distance between ${t_i}^G$ and the transformed version of ${t_i}^L$ as per Eq. 2. In case the known initial affine transformation parameters are not close to the local minima, the same cost function is used to find the least-squares solution of the affine parameters.

$$\underset{T_1}{\operatorname{argmin}} \sum_i ||{t_i}^G - T_1({t_i}^L)||_2^2 \tag{2}$$

Another cost function for refining T_1 is to use the SfM-SLAM 3D-2D matches. Assuming that the intrinsic and extrinsic parameters of the i^{th} SLAM keyframe is accurately known and together represented by a matrix K_i, the j^{th} SfM 3D point ${P_j}^G$ can be projected on to it. The sum of errors d between the calculated projection and the known 2D feature match p_{ij} has to be minimized according to the Eq. 3. Here, T_2 is the affine transformation that converts the local to global coordinate system and is initialized by T_1 from the previous step. The rigid affine transformation estimation and refinement step discussed in this section aligns the SLAM data with SfM.

$$\underset{T_2}{\operatorname{argmin}} \sum_i \sum_j d(p_{ij}, K_i T_2^{-1}({P_j}^G)) \tag{3}$$

Further, the authors of Middelberg et al. (2014) modify the background bundle adjustment (BA) step of a SLAM algorithm to include the additional SfM data. They assume that the initial affine transformation step initializes the parameters of the cost function of BA close to the global minimum and do not handle the scale drift explicitly. Hence merely applying this strategy to datasets with wide scale drifts leads to the BA step to fail as they get stuck in local minima. To prevent this, we handle the scale drift explicitly as explained in Sect. 3.2 instead of relying only on the background optimization process.

3.2 Handling scale drift

3.2.1 RANSAC splitting of trajectory

In an ideal world, a single affine transformation relates the SLAM and SfM coordinate system. However this is not the case as the SLAM trajectory drifts due to a number of reasons like insufficient landmarks being tracked and lack of lateral motion. The latter condition is common in a scenario of navigating close to buildings. There would be sections of trajectory with abundant information where the drift is low and sections with increasing drift due to lack of information. So, different sections of the trajectory have different affine relationship with their corresponding bunch of PnP poses. The idea is to split the trajectory into groups of contiguous keyframes that have different affine relationships. Adding to the challenge, the drift in the trajectory occurs in all seven degrees of freedom (DoF) including scale. While the next Sect. 3.2.2 covers the procedure for scale drift correction using pose graph optimization, this section describes the preparation required for the trajectory optimization.

In order to initialize the optimal affine transformation estimation for different segments of the trajectory as explained in the previous Sect. 3.1, a coarse estimate of affine transformation is found using the closed-form method of Horn et al. (1988). This is a problem of finding the transformation between two sets of 3D points by considering the camera positions of SLAM keyframe and PnP poses. A minimum of three such non-collinear pairs is required to estimate T. But splitting the trajectory means there could be collinear sections which can lead to multiple solutions for rotation estimation. Hence Horn's method is modified to use all of the camera pose information instead of mere position. The translation and scale estimation routines are not altered but the rotation matrix R which is estimated using the quaternion equivalent of Eq. 4 is fed with additional camera pose information.

$$\underset{R}{\operatorname{argmax}} \sum_{i=1}^{n} r'_{l,i} \cdot R(r'_{r,i}), \tag{4}$$

$$r'_l = r_l - \overline{r}_l \tag{5}$$

$$r'_r = r_r - \overline{r}_r, \tag{6}$$

where r_l, r_r are the coordinates of left and right coordinate systems and r'_l, r'_r are the vectors with respect to the centroids $\overline{r}_l, \overline{r}_r$ as per Eqs. 5 and 6. The left and right coordinate systems are the SLAM and SfM coordinate systems respectively.

In addition to the camera centers, the camera axes have useful information that has not been exploited for rotation estimation. The trick is to translate each of those vectors to the centroid and include them in the rotation estimation Eq. 4, thereby reducing the ambiguity for 3D rotation estimation.

Before estimating the transformation, as a first step, the entire trajectory is randomly sampled and the consensus set of inliers are found using a large threshold. This is to filter out PnP poses that have large errors. In the second step, a low threshold value is used in the RANSAC process to come up with a model T. However, the samples and inlier picking process is modified to include only contiguous keyframes. Samples are selected only if they are not separated by more than 10 keyframes (KFs). The inliers are selected incrementally moving on either side of the trajectory and stopping when inliers are not found for a consecutive 10 KFs. This helps the grouping of inliers based on T calculated for a certain section of the trajectory. The procedure is documented in the pseudocode (Algorithm 1).

Previously, the PnP poses are estimated for a SLAM keyframe only if there are over 20 well distributed 3D-2D matches. So only the SLAM keyframes that have a corresponding PnP pose are considered for calculating scale drift correction. During the process of modified RANSAC, overlapping inlier groups might be obtained. In that case, the largest of these overlapping groups are chosen neglecting others. Each of the inlier groups corresponds to an affine model T_n which is used to initialize the optimization process. They also correspond to different scale drifts enabling us to apply scale drift correction with multiple pivot poses that are fixed during optimization.

Algorithm 1: RANSAC splitting of trajectory

```
Result: KF clusters with different affine $T_n$ relationship to SfM
        map
Initialization1 - Exclude SLAM KFs that does not have a
corresponding PnP pose;
Initialization2 - Perform RANSAC with large threshold t to
remove outlier PnP poses;
while $itr < maxItr$ do
    get KF samples $s_1, s_2, s_3$ s.t $max\_id(s_1, s_2, s_3)$ -
    $min\_id(s_1, s_2, s_3) < 10$;
    compute rigid affine $T$;
    find inlierCount in the neighbourhood;
    if inlierCount > initialInlierCountThreshold then
        note down inlier groups;
    end
    if inlierCount > inlierCountThreshold then
        update $maxItr$;
        inlierCountThreshold = inlierCount;
    end
    $itr = itr + 1$
end
```

3.2.2 Trajectory optimization process

The SLAM trajectory optimization is traditionally done using bundle adjustment (BA) . Even though IbL approaches have SfM-SLAM data for a better constrained BA, drift correction still doesn't work if there is intermittent outage in lateral motion information. Such drifts are generally handled at loop closures in state of the art SLAM methods using pose graph optimization which was originally introduced by Feng and Milios (1997). A loop closure scenario provides additional constraints between the latest section of the trajectory and an old section of the trajectory in addition to the relative transformation constraints between successive keyframes. Using

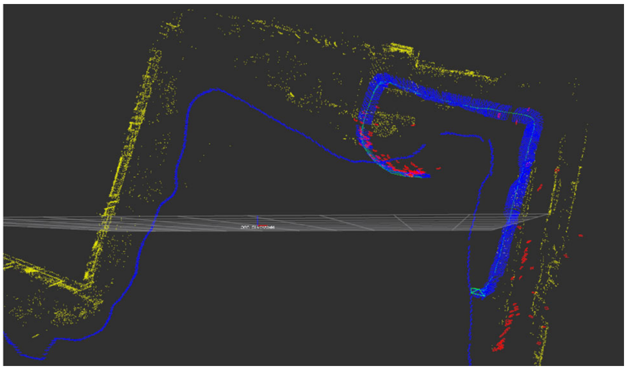


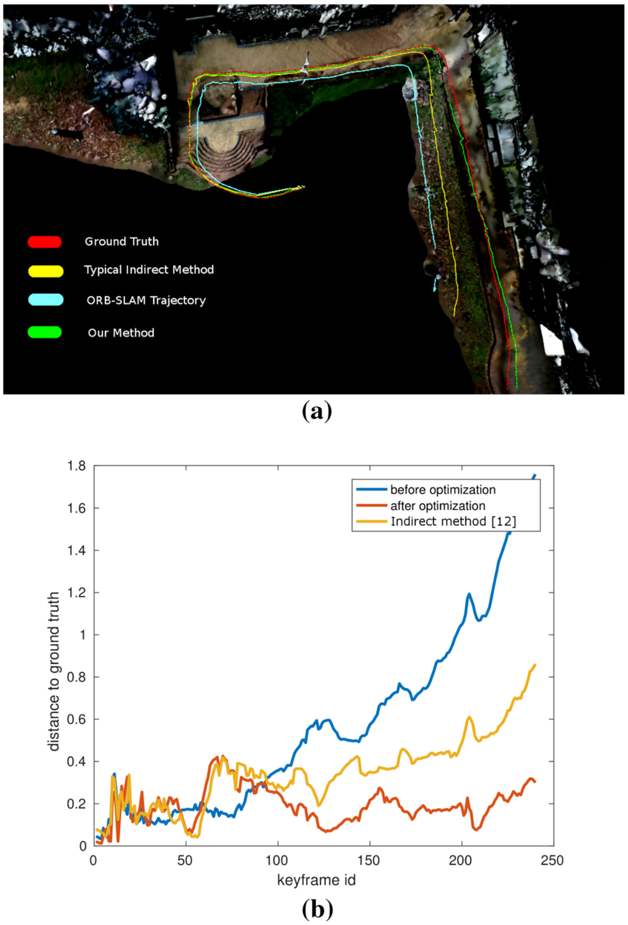
Fig. 5 Red dot clusters (PnP Poses) show two regions of rich SfM and SLAM shared information

such constraints, a pose graph optimization algorithm called graphSLAM (Grisetti et al. 2010) performs a Lavenberg-Marquardt optimization over the SE(3) Lie group manifold. The advantage of manifold optimization is that it avoids over parametrization of camera pose like using rotation matrices and quaternions which does not allow moving freely in the parameter space. For example, the SE(3) manifold is smooth and locally Euclidean allowing optimization step increments. Since scale drift is also involved, Strasdat et al. (2010) introduced pose graph optimization on the Sim(3) manifold, which is also a type of Lie group. In our approach we follow a Sim(3) optimization similar to ORB SLAM paper by Mur-Artal et al. (2015) where they use it during loop closure. However, we adapt the procedure to initialize different sections of the trajectory with individual affine transformations estimated at the end of trajectory splitting processes explained in Sect. 3.2.1 and then perform Sim(3) optimization.

The graph nodes in the optimization problem are built by converting the SE3 camera poses of the keyframe into a Sim(3) matrix S. This is done by using the same rotation matrix and translation that describes the pose but adding a scale parameter and setting it to unity. The relative pose between the keyframes is also represented by a Sim(3) matrix which helps in deriving the edge constraints in the graph. The error function that constitutes these edges is defined in the Sim(3) manifold space using a logarithmic map as per Eq. 7 suitable for LM optimization. Here ξ is the parameter vector in the manifold space. Using an exponential map in Eq. 8, the increments in the smooth manifold space can be converted to the parameter space to reconstruct the Sim(3) transformation matrix.

$$\xi = log_{sim_3}(S) \tag{7}$$

$$S = exp_{Sim_{(3)}}(\xi) \tag{8}$$



(a)

(b)

Fig. 6 **a** Implementation of indirect method (Middelberg et al. 2014) (yellow), Before (cyan) and after (green) Scale drift correction vs ground truth (red) for building dataset **b** Drift plot along the trajectory length (Color figure online)

The original SLAM keyframes are classified into corrected and non-corrected vertices of the pose graph depending on the groupings made in Sect. 3.2.1. Corrected vertices are those SLAM keyframes poses for which both the SfM-SLAM constraints and the inter-SLAM constraints are known. Non-corrected vertices are those SLAM keyframe poses for which only the inter-SLAM constraints are known. Depending on the number of common landmarks visible across keyframes the connected vertices in the co-visibility graph are identified and their edge constraints are initialised. Figure 5 shows an example SLAM trajectory with two sections that have different affine transformations T_1 and T_2. These sections use these transformations to add an additional edge constraint that has non-unity scale parameters. One of the keyframes in each section is pivoted during the optimization procedure. The graph is then optimized leading to the distribution of scale drift to the rest of the trajectory.

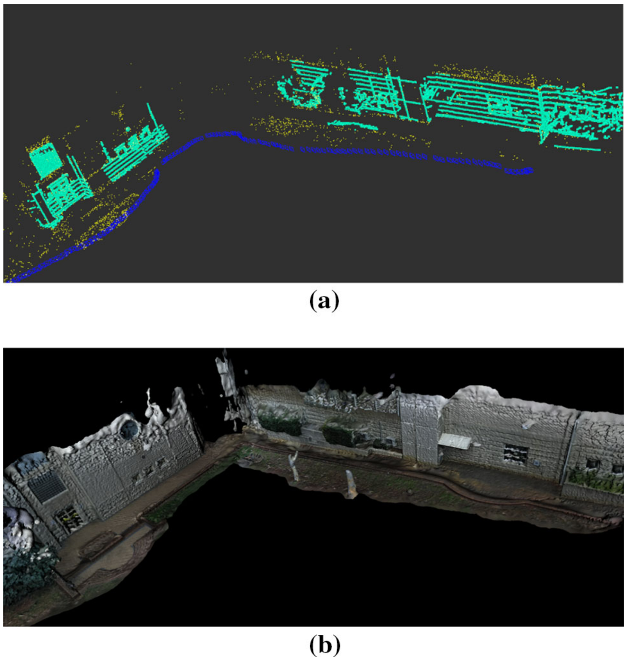


Fig. 7 **a** Edges on SfM images projected to SfM model **b** SfM model on which the edges were projected

4 Experiments

4.1 Experimental setup

This paper addresses the problem of navigation close to buildings using a calibrated monocular camera. The SLAM trajectory is corrected for 7DoF drift using information from a sparse SfM map. The SfM map, with its point cloud and the camera poses, are built using COLMAP (Schonberger and Jan-Michael 2016) as it provides good documentation on the map building process. The code of ORB-SLAM (Mur-Artal et al. 2015), which is a robust real-time monocular SLAM was adapted to include the SfM data. New background optimization procedures as explained in Sect. 3 were added. However, real-time constraints are not considered in this paper.

The dataset used in the experiments is that of our university building. The SLAM trajectory was collected using a calibrated monocular camera and walking close to the walls where GPS signals experience multi-path reflections, and thus accurate geo-referencing was not possible as in Iwami et al. (2018). The offline experiment was performed in a ROS environment by streaming the collected data. ROS visualization tools were used for generating the figures showing the trajectory. Usually, to generate the ground truth trajectory, the SLAM trajectory keyframe images are included inside the SfM map building process and the estimated camera poses are considered as ground truth. Being significantly different from the SfM images, the SLAM keyframes often end up with erroneous pose estimates even with strenuous bundle adjustment. Hence the SLAM trajectory was optimized using the proposed framework after removing the erroneous 3D-2D matches and then resulting camera poses were inserted to the SfM map building process to generate the ground truth.

4.2 Scale drift correction results and discussion

Figure 6 shows the results comparing the trajectory with and without applying the proposed framework. While the Figure 6a compares the actual trajectory obtained by various means against the ground truth, Figure 6b shows the plot of the Euclidean distance of the n^{th} keyframe camera origin from their corresponding ground truth location for each trajectory. The cases considered for comparison are: (a) trajectory generated by applying a typical indirect method like (Middelberg et al. 2014), (b) plain ORB-SLAM trajectory without applying any drift correction, (c) after applying proposed scale drift correction and (d) ground truth trajectory. The results clearly show that the corrected trajectory is considerably closer to the ground truth when compared to the drifted trajectory which was the output from ORB-SLAM and the trajectory corrected using a typical indirect method. The drift comparison plot also proves that the proposed framework performs better against the implementation of a typical indirect method. In the building dataset, the two sections of the trajectory which had enough PnP poses (as seen in Figure 5) helped correct the scale drift and distribute it to sections that did not have enough SfM information.

To further demonstrate the effectiveness of the proposed algorithm in 3D pose correction, an augmented reality style visualization was implemented. The edges on SfM images were projected on to the SfM model using camera parameters as shown in Fig. 7. The 3D edge points on the model surface are then re-projected to the SLAM keyframes and visualized as shown in Fig. 8. Edges appear registered when a keyframe of the SLAM trajectory is closer to the ground truth pose as the re-projection error is minimal. The Fig. 8a, c, e and g on the left column shows the re-projected edges on the sampled keyframes when the implementation of Middelberg et al. (2014) was used for trajectory correction. The Fig. 8b, d, f and h on the right column shows the re-projected edges on sampled keyframes when the proposed algorithm was used for trajectory correction which clearly shows lower re-projection error.

5 Conclusion

The paper thus proposes a novel IbL framework that focuses on the SfM-SLAM sparse co-visibility problem, which typically arises if there exists a large perspective difference between the SfM and SLAM data. Our IbL method utilizes

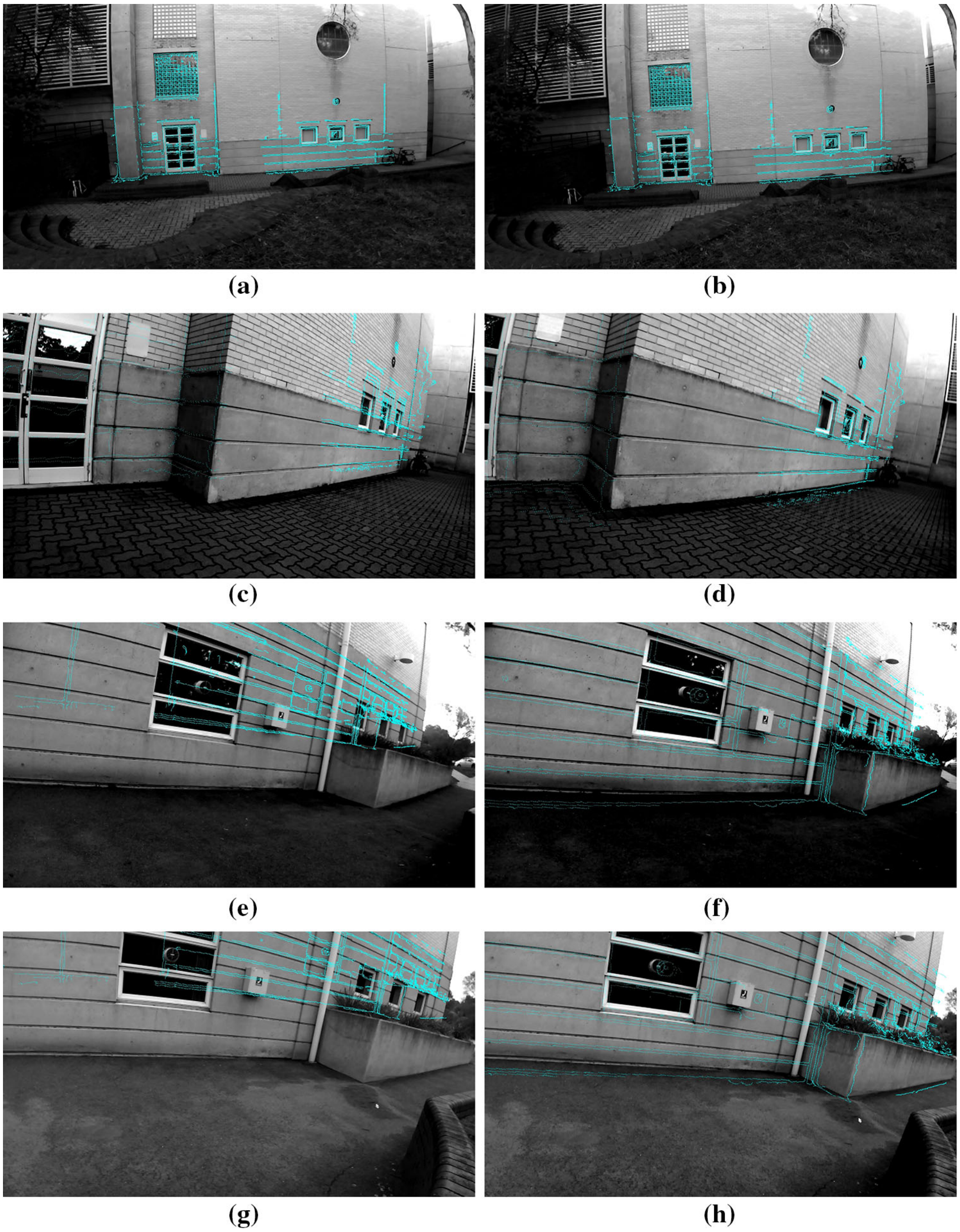

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 8 Left Column **a**,**c**,**e**,**g** Re-projected edges on SLAM keyframes when using implementation of [12] for trajectory correction, Right Column **b**,**d**,**f**,**h** Re-projected edges on SLAM keyframes when using proposed algorithm

a split Sim(3) optimization, which can effectively handle the varying co-visibility between the SfM and SLAM data. The experimental results and visualization demonstrate that the errors which include the scale drift can be accurately corrected compared to the existing methods. Higher localization accuracy promises reliable operations for autonomous applications using drones and ground vehicles under GPS denied environments like urban canyons.

In this paper, the SfM and SLAM images were matched using sparse SIFT features to derive the geometric constraints. However, many more untapped information like edges and dense pixels can improve the co-visibility between SfM and SLAM. As a future extension, such extra information will be used in conjunction with sparse feature matches to further optimize the trajectory.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10514-021-10031-8.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

References

Albl, C., Kukelova, Z., & Pajdla, T. (2016). Rolling shutter absolute pose problem with known vertical direction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3355–3363.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307.

Arth, C., Pirchheim, C., Ventura, J., Schmalstieg, D., & Lepetit, V. (2015). Instant outdoor localization and SLAM initialization from 2.5D maps. *IEEE Transactions on Visualization and Computer Graphics, 21*(11), 1309–1318.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. In *European Conference on Computer Vision*, (pp. 404–417). Springer.

Borges, P., Zlot, R., Bosse, M., Nuske, S., & Tews, A. (2010). Vision-based localization using an edge map extracted from 3D laser range data. In *2010 IEEE International Conference on Robotics and Automation*, (pp. 4902–4909). IEEE.

Bosse, Michael, Newman, Paul, Leonard, John, & Teller, Seth. (2004). Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research, 23*(12), 1113–1139.

Cadena, Cesar, Carlone, Luca, Carrillo, Henry, Latif, Yasir, Scaramuzza, Davide, Neira, José, et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics, 32*(6), 1309–1332.

Carrillo-Arce, L. C., Nerurkar, E. D., Gordillo, J. L., & Roumeliotis, S. I. (2013). Decentralized multi-robot cooperative localization using covariance intersection. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 1412–1417). IEEE.

Castellanos, J. A., Martinez-Cantin, R., Tardós, J. D., & Neira, J. (2007). Robocentric map joining: Improving the consistency of ekf-slam. *Robotics and Autonomous Systems, 55*(1), 21–29.

Cieslewski, T., Choudhary, S., & Scaramuzza, D. (2018). Data-efficient decentralized visual slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 2466–2473). IEEE.

DuToit, R. C., Hesch, J. A., Nerurkar, E. D., & Roumeliotis, S. I. (2017). Consistent map-based 3d localization on mobile devices. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 6253–6260). IEEE.

Feng, Lu., & Milios, Evangelos. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots, 4*(4), 333–349.

Ferber, M., Sastuba, M., Grehl, S., & Jung, B. Combining SURF and SIFT for challenging indoor localization using a feature cloud.

Gao, Xiao-Shan., Hou, Xiao-Rong., Tang, Jianliang, & Cheng, Hang-Fei. (2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(8), 930–943.

Geppert, M., Liu, P., Cui, Z., Pollefeys, M., & Sattler, T. (2019). Efficient 2d-3d matching for multi-camera visual localization. In *2019 International Conference on Robotics and Automation (ICRA)*, (pp. 5972–5978). IEEE.

Germain, H., Bourmaud, G., & Lepetit, V. (2019). Sparse-to-dense hypercolumn matching for long-term visual localization. In *2019 International Conference on 3D Vision (3DV)*, (pp. 513–523). IEEE.

Gordo, Albert, Almazan, Jon, Revaud, Jerome, & Larlus, Diane. (2017). End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision, 124*(2), 237–254.

Grisetti, Giorgio, Kummerle, Rainer, Stachniss, Cyrill, & Burgard, Wolfram. (2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine, 2*(4), 31–43.

Horn, Berthold KP., Hilden, Hugh M., & Negahdaripour, Shahriar. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A, 5*(7), 1127–1135.

Howe, E., & Novosad, J. (2005). Extending slam to multiple robots.

Huang, S., & Dissanayake, G. (2006). Convergence analysis for extended kalman filter based slam. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, (pp. 412–417). IEEE.

Irschara, A., Zach, C., Frahm, J. M., & Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, (pp. 2599–2606). IEEE.

Iwami, K., Ikehata, S., & Aizawa, K. (2018). Scale drift correction of camera geo-localization using geo-tagged images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, p. 0.

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '07*, (pp. 1–10), Washington, DC, USA, IEEE Computer Society.

Klingner, B., Martin, D., & Roseborough, J. (2013). Street view motion-from-structure-from-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 953–960.

Kneip, L., Scaramuzza, D., & Siegwart, R. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR 2011*, (pp. 2969–2976). IEEE.

Kukelova, Z., Bujnak, M., & Pajdla, T. (2013). Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 2816–2823).

Larsson, V., Kukelova, Z., & Zheng, Y. (2017). Making minimal solvers for absolute pose estimation compact and robust. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 2316–2324).

Lepetit, Vincent, Moreno-Noguer, Francesc, & Fua, P. (2009). Epnp: Efficient perspective-n-point camera pose estimation. *International Journal of Computer Vision, 81*(2), 155–166.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, (pp. 2548–2555). IEEE.

Lim, H., Sinha, S. N., Cohen, M. F., & Uyttendaele, M. (2012). Real-time image-based 6-DOF localization in large-scale environments. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1043–1050). IEEE.

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

Lynen, S., Sattler, T., Bosse, M., Hesch, J. A., Pollefeys, M., & Siegwart, R. (2015). Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, vol. 1, p. 1.

Lynen, Simon, Zeisl, Bernhard, Aiger, Dror, Bosse, Michael, Hesch, Joel, Pollefeys, Marc, et al. (2020). Large-scale, real-time visual-inertial localization revisited. *The International Journal of Robotics Research, 39*(9), 1061–1084.

McDonald, John, Kaess, Michael, Cadena, Cesar, Neira, Jose, & Leonard, John J. (2013). Real-time 6-dof multi-session visual slam over large-scale environments. *Robotics and Autonomous Systems, 61*(10), 1144–1158.

Middelberg, S., Sattler, T., Untzelmann, O., & Kobbelt, L. (2014). Scalable 6-DOF localization on mobile devices. In *European Conference on Computer Vision*, (pp. 268–283). Springer.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics, 31*(5), 1147–1163.

Mur-Artal, Raúl., & Tardós, Juan D. (2017). Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters, 2*(2), 796–803.

Paz, L. M., Tardós, J. D., & Neira, J. (2008). Divide and conquer: Ekf slam in $o(n)$. *IEEE Transactions on Robotics, 24*(5), 1107–1120.

Piasco, N., Sidibé, D., Demonceaux, C., & Gouet-Brunet, V. (2018). A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition, 74,* 90–109.

Pless, R. (2003). Using many cameras as one. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, vol. 2, (pp. II–587). IEEE.

Radenović, F., Tolias, G., & Chum, O. (2016). Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision*, (pp. 3–20). Springer.

Ramalingam, S., Bouaziz, S., Sturm, P., & Brand, M. (2010). SKYLINE2GPS: Localization in urban canyons using omni-skylines. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 3816–3823). IEEE.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, (pp. 2564–2571). IEEE.

Schmuck, Patrik, & Chli, Margarita. (2019). Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics, 36*(4), 763–781.

Schonberger, J. L ., & Jan-Michael F. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113.

Stenborg, E., Sattler, T., & Hammarstrand, L. (2020). Using image sequences for long-term visual localization. In *2020 International Conference on 3D Vision (3DV)*, (pp. 938–948). IEEE.

Stenborg, E., Toft, C., & Hammarstrand, L. (2018). Long-term visual localization using semantically segmented images. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 6484–6490). IEEE.

Strasdat, H., Montiel, J., & Davison, A. J. (2010). Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2(3), 7.

Tareen, S. A. K., & Saleem, Z. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, (pp. 1–10). IEEE.

Ventura, Jonathan, Arth, Clemens, Reitmayr, Gerhard, & Schmalstieg, Dieter. (2014). Global localization from monocular slam on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics, 20*(4), 531–539.

Wu, C. SiftGPU manual.

Yihong, W., Tang, F., & Li, H. (2018). Image-based camera localization: an overview. *Visual Computing for Industry, Biomedicine, and Art, 1*(1), 1–13.

Zhou, L., Luo, Z., Shen, T., Zhang, J., Zhen, M., Yao, Y., Fang, T., & Quan, L. (2020). Kfnet: Learning temporal camera relocalization using kalman filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 4919–4928.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Deepak Rajamohan is currently at the final stage of his Ph.D. in aerospace engineering at the University of New South Wales in Canberra, Australia. He received his Master of Science in Computer Science at the International Institute of Information Technology, Hyderabad, India in 2015 and bachelor of engineering in Electronics and Communication Engineering at Anna University, India in 2008. His research interest includes Simultaneous Localization and Mapping (SLAM), image registration, 3D-2D registration, Computer Vision and Geographical Information System.

Dr. Jonghyuk Kim has been affiliated at the Robotics Institute (RI) and the School of Mechanical and Mechatronics Engineering at UTS since 2019. Dr Kim has obtained his PhD degree in Field Robotics at the University of Sydney in 2004, pioneering the area of airborne simultaneous localisation and localisation (SLAM) problem on UAV platforms. He was a lecturer/senior lecturer at the Research School of Engineering at the ANU during 2005 to 2019 and was on 3 years of secondment to DATA61 Automap project in Canberra. He studied Electronics Engineering at Chungnam National University, receiving BS/MS degrees in 1997/99 respectively, and was a postdoctoral fellow at the Australian Centre for Field Robotics (ACFR) and Centre for Autonomous Systems (CAS) in Sydney until 2005. He is the recipient of the prestigious Charles Sharpe Beecher Prize and Award from IMechE, UK, 2005 for his contributions to data fusion for multiple UAVs, and he was recognised as an Emerging Researcher from ATSE in 2015. He has served a secretarial role for ARAA (Australian Robotics and Automation Association), co-chaired ACRA (Australasian Conference in Robotics and Automation) in 2008/2015, and served Associated Editor roles for IEEE-IROS (2008/2019) and IEEE ICRA (2010).

Dr. Matt Garratt is affiliated with the School of Engineering and Information Technology. His research focuses on sensing, guidance and control for autonomous systems. He is an associate editor for the IEEE Transactions on Artificial Intelligence and the IEEE Robotics and Automation Letters. Matt was chair of the IEEE Computational Intelligence Society Task Force on the ethics and social implications of computational intelligence from 2017-2020 and remains as an active member. Some of his research successes include demonstration of terrain following using vision for an unmanned helicopter, landing an unmanned helicopter onto a moving deck simulator and control of helicopters using neural networks. Some of his current research projects include achieving autonomous flight in cluttered environments using monocular cameras and range sensors, landing UAVs on moving platforms, adaptive flight control for flapping wing and rotary wing vehicles, and self-organising swarms of Unmanned Systems.

Dr. Mark Pickering is affiliated with the School of Engineering and Information Technology. His research experience includes Digital Signal Processing, Image Processing, Medical Image Registration, Video and Image Compression, and Remote Sensing