Real-time 6-DOF Multi-session Visual SLAM over Large Scale Environments

J. McDonald[a,*], M. Kaess[c], C. Cadena[b], J. Neira[b], J. J. Leonard[c]

[a] *Department of Computer Science, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland*
[b] *Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain*
[c] *Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA*

Abstract

This paper describes a system for performing real-time multi-session visual mapping in large-scale environments. Multi-session mapping considers the problem of combining the results of multiple simultaneous localisation and mapping (SLAM) missions performed repeatedly over time in the same environment. The goal is to robustly combine multiple maps in a common metrical coordinate system, with consistent estimates of uncertainty. Our work employs incremental smoothing and mapping (iSAM) as the underlying SLAM state estimator and uses an improved appearance-based method for detecting loop closures within single mapping sessions and across multiple sessions. To stitch together pose graph maps from multiple visual mapping sessions, we employ spatial separator variables, called anchor nodes, to link together multiple relative pose graphs.

The system architecture consists of a separate front-end for computing visual odometry and windowed bundle adjustment on individual sessions, in conjunction with a back-end for performing the place recognition and multi-session mapping. We provide experimental results for real-time multi-session visual mapping on wheeled and handheld datasets in the MIT Stata Center. These results demonstrate key capabilities that will serve as a foundation for future work in large-scale persistent visual mapping.

Keywords: bundle adjustment, place recognition, stereo, anchor nodes, iSAM.

1. Introduction

Despite substantial recent progress in visual simultaneous localisation and mapping (SLAM) [1], many issues remain to be solved before a robust, general visual mapping and navigation solution can be widely deployed. A key issue in our view is that of *persistence* – the capability for a robot to operate robustly for long periods of time. As a robot makes repeated transits through previously visited areas, it cannot simply treat each mission as a completely new experiment, not making use of previously built maps. However, nor can the robot treat its complete lifetime experience as "one big mission", with all data considered as a single pose graph and processed in a single batch optimisation. We seek to develop a framework that achieves a balance between these two extremes, enabling the robot to leverage off the results of previous missions, while still adding in new areas as they are uncovered and improving its map over time.

The overall problem of persistent visual SLAM involves several difficult challenges not encountered in the basic SLAM problem. One issue is dealing with dynamic environments, requiring the robot to correct for long-term changes, such as furniture and other objects being moved, in its internal representation; this issue is not addressed in this paper. Another critical issue, which is addressed in this paper, is how to pose the state estimation problem for combining the results of multiple mapping missions efficiently and robustly.

Cummins defines the multi-session mapping problem as "the task of aligning two partial maps of the environment collected by the robot during different periods of operation [2]." We consider multi-session mapping in the broader context of life-long, persistent autonomous navigation, in which we would anticipate tens or hundreds of repeated missions in the same environment over time. As noted by Cummins, the "kidnapped robot problem" is closely related to multi-session mapping. In the kidnapped robot problem, the goal is to estimate the robot's position with respect to a prior map given no *a priori* information about the robot's position.

Also closely related to the multi-session mapping problem is the multi-robot mapping problem. In fact, multi-session mapping can be considered as a more restricted case of multi-robot mapping in which there are no direct encounters between robots (only indirect encounters, via observations made of the same environmental structure). Kim *et al.* presented an extension to iSAM to facilitate online multi-robot mapping based on multiple pose graphs [3]. This work utilised "anchor nodes", equivalent to the "base nodes" introduced by Ni and Dellaert for decomposition of large pose graph SLAM problems into submaps of efficient batch optimisation [4], in an approach called Tectonic Smoothing and Mapping (T-SAM). Our work builds on the approach of Kim *et al.* [3] to perform multi-session visual mapping by incorporating a stereo odometry frontend in conjunction with a place-recognition system for identifying inter-

*Corresponding author. Tel.: +353 1 708 4589; fax: +353 1 708 3848.
Email addresses: johnmcd@cs.nuim.ie (J. McDonald), kaess@mit.edu (M. Kaess), cesarcadena.lerma@gmail.com (C. Cadena), jneira@unizar.es (J. Neira), jleonard@mit.edu (J. J. Leonard)

and intra-session loop closures.

This paper makes a number of extensions to the work presented in [5]. In particular, in [5] we provided preliminary results of a multi-session visual SLAM system based on the architecture shown in Fig. 1. Here we expand the discussion and give details of changes that we have made to increase the system's overall robustness and to permit real-time processing over large scale environments. Results are provided demonstrating robust multi-session visual SLAM processing on datasets including (i) up to four separate sessions (totalling > 45 minutes of video at 20fps), (ii) wheeled and handheld sensors, (iii) indoor and outdoor sequences, and, (iv) sequences involving full 6-DOF-motion (i.e. ascending and descending stairs). We also present a comprehensive quantitative evaluation of the system using the above datasets.

The remainder of the paper is organised as follows. In the next section we review related work in the area focussing on multi-session and multi-robot approaches to localisation and mapping. Section 3 provides an overview of the system architecture, with details of front-end processing including the stereo odometry and single-session visual SLAM given in sections 3.1 and 3.2, respectively. In Section 3.3 we describe the approach used to integrating the quaternion-based representation for rotations and homongeneous point representation into the iSAM optimisation process. The back-end modules including visual place recognition and multi-session visual SLAM are explained in sections 3.4 and 3.5, respectively. Experimental results and a comprehensive quantitative analysis of the system's performance is provided in Section 4. Finally, Section 5 provides concluding remarks and potential future directions for the research.

2. Related work

Several vision researchers have demonstrated the operation of visual mapping systems that achieve persistent operation in a limited environment. Examples of recent real-time visual SLAM systems that can operate persistently in a small-scale environment include Klein and Murray [6], Eade and Drummond [7], and Davison *et al.* [8, 9]. Klein and Murray's system is highly representative of this work, and is targeted at the task of facilitating augmented reality applications in small-scale workspaces (such as a desktop). In this approach, the processes of tracking and mapping are performed in two parallel threads. Mapping is performed using bundle adjustment. Robust performance was achieved in an environment as large as a single office. While impressive, these systems are not designed for multi-session missions or for mapping of large-scale spaces (e.g., the interior of a building).

One exception to this has been the extension to PTAM developed by Castle *et al.* [10] to permit several cameras to work in multiple maps, both separately or simultaneously. Here the approach to dealing with large-scale environments is to permit the user to decide what regions to map. Each map is bounded in size and operates independently of other maps in the system. To switch between maps the current frame is matched against a set of subsampled keyframes from all existing maps. The authors provide impressive results of the technique's operation in a building scale environment. A key difference between this approach and our work is that the system does not estimate the transformation between the submaps and therefore does not provide a global estimate of the environment.

There have also been a number of approaches reported for large-scale visual mapping. Although a comprehensive survey is beyond the scope of this paper we do draw attention to the more relevant stereo-based approaches. Perhaps the earliest of these was the work of Nistér *et al.* [11] on stereo odometry. In the robotics literature, large-scale multi-session mapping has been the focus of recent work of Konolige *et al.* in developing view-based mapping systems [12, 13]. Our research is closely related to this work, but has several differences. A crucial new aspect of our work in relation to [13] is the method we use for joining the pose graphs from different mapping sessions. In the view-based mapping approach, Konolige et al. employ the Toro incremental optimisation algorithm to allow for real-time performance. Due to the fact that Toro requires that all poses are connected in a single graph, at the beginning of each new session the new pose-graph is immediately connected to the last pose from the previous session through what they refer to as a "weak link". The weak links are added with a very high covariance and subsequently deleted after place recognition is used to join the pose graphs [13]. In our approach, which extends [3] to full 6-DOF, we use anchor nodes as an alternative to weak links. Here each session is represented initially as a disjoint pose-graph with each pose stored relative to that pose-graph's anchor node. When the place recognitions system identifies an encounter between two separate sessions, the encounter induces a constraint between the two associated poses and the anchor nodes of the associated pose-graphs. Since the pose-graphs are each represented relative to the anchors nodes, their use provides a more efficient and consistent way to stitch together the multiple pose graphs resulting from multiple mapping sessions. Further details on this aspect of our system are provided in Section 3.5. In addition, our system has been applied to hybrid indoor/outdoor scenes, with hand-carried (full 6-DOF) camera motion.

3. System overview

In this section we describe the architecture and components of a complete multi-session stereo visual SLAM system. This includes a stereo visual SLAM frontend, a place recognition system for detecting single and multi-session loop closures, and a multi-session state-estimation system. A schematic of the system architecture is shown in Fig. 1. The system uses a submapping approach in conjunction with a global multi-session pose graph representation. Optimisation is performed by applying incremental and batch SAM to the pose graph and the constituent submaps, respectively. Each submap is constructed over consecutive sets of frames, where both the motion of the sensor and a feature-based map of the scene is estimated. Once the current submap reaches a user-defined maximum number of poses, 15 in our system, the global pose graph is augmented with the resultant poses.

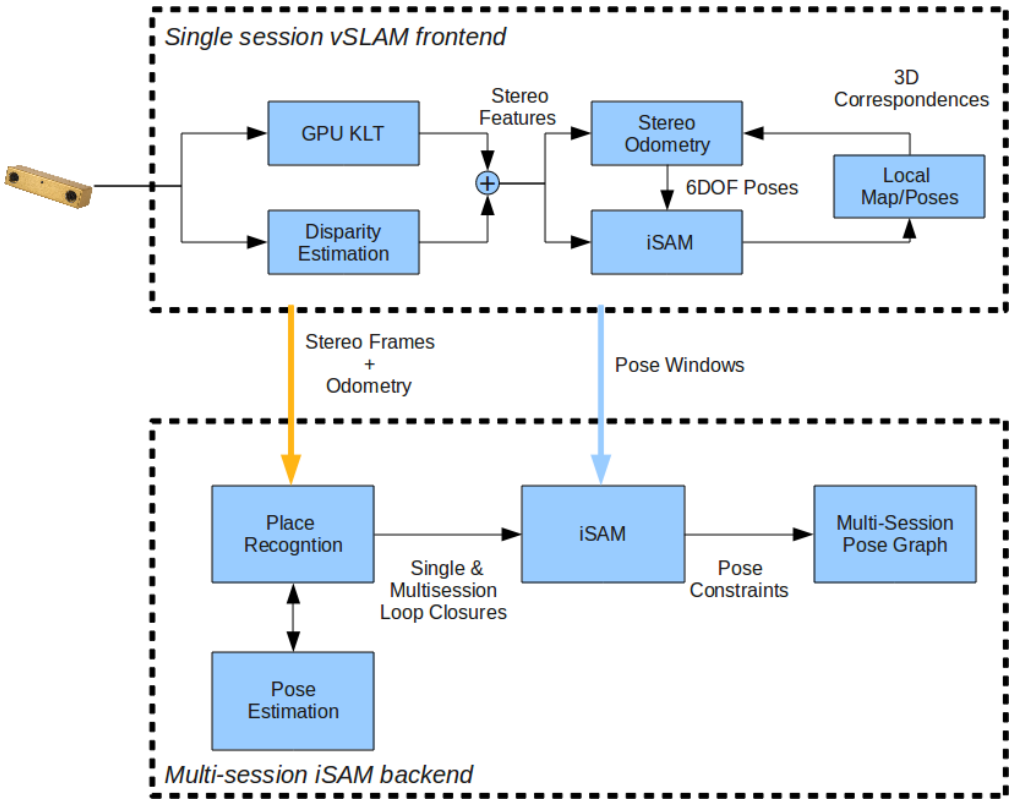


Figure 1: Internal architecture of windowed and multi-session visual SLAM (vSLAM) processes.

In parallel to the above, as each frame is processed, the visual SLAM frontend communicates with a global place recognition system for intra- and inter-session loop closure detection. When a loop closure is detected, pose estimation is performed on the matched frames, with the resultant pose and frame-id's passed to the multi-session pose graph optimisation module.

3.1. Stereo odometry

Within each submap the inter-frame motion and associated scene structure is estimated via a stereo odometry frontend. The most immediate benefit of the use of stereo vision is that it avoids issues associated with monocular systems, including the inability to estimate scale and indirect depth estimation. The stereo odometry approach we use is similar to that presented by Nistér *et al.* [11].

Our stereo odometry pipeline tracks features using a standard robust approach, followed by a pose refinement step. For each pair of stereo frames we first track a set of Shi-Tomasi corners in the left frame using the KLT tracking algorithm. The resulting tracked feature positions are then used to compute the corresponding feature locations in the right frame. Approximate 6-DOF pose estimation is performed through the use of a RANSAC-based 3-point algorith [14]. The input to the motion estimation algorithm consists of the set of tracked feature positions and disparities within the current frame and the current estimates of the 3D locations of the corresponding landmarks. In our work we have found that ensuring that approximately 50 features are tracked between frames results in a reliable pose estimate through the 3-point RANSAC procedure. Finally, accurate pose estimation is achieved by identifying the inliers from the estimated pose and using them in a Levenberg-Marquardt optimisation that minimises the reprojection error in both the left and right frames.

In our stereo odometry implementation we use a GPU-based KLT tracker [15]. This minimises the load on the CPU (by delegating the feature detection and tracker to the GPU) and exploits the GPU's inherent parallel architecture to permit processing at high frame rates. In parallel to this we compute a disparity map for the frame, which is then combined with the results of the feature tracker, resulting in a set of stereo features.

In order to maintain an adequate number of features we detect new features in every frame whilst at the same time setting a minimum inter-feature distance in the KLT tracker. A consequence of this approach is that the system tries to ensure that there is a good distribution of features over the entire frame.

3.2. Single session visual SLAM

Deriving a pose graph representation from the stereo odometry system involves two levels of processing. The first of these optimises over the poses, features and 3D structure within a local bundle adjustment window. As each new frame is added, a full batch optimisation is performed. The second step transfers optimised poses to the pose graph after a fixed maximum number of frames is reached. The resulting pose graph structure contains no point features and can be optimised efficiently even for a large number of poses.

We apply smoothing in combination with a homogeneous point representation to the local window to improve the pose estimates obtained from visual odometry. In contrast to visual odometry, smoothing takes longer range constraints into account, which arise from a single point being visible in multiple frames. The homogeneous representation allows dealing with points at infinity, see Triggs et al. [16] or Hartley and Zisserman [17]. Points close to or at infinity cannot be represented correctly by the conventional Euclidean parameterisation.

After removing the over-parameterisation of both rotation and homogeneous point representations (see Section 3.3), the optimisation problem is solved with a standard least-squares solver. We use the iSAM library [18] to perform batch smoothing with Powell's Dog-Leg algorithm [19]. iSAM represents the optimisation as a factor graph, a bipartite graph containing variable nodes, factor nodes and links between those. Factor nodes, or short factors, represent individual probability densities

$$f_i(\Theta_i) = f_i(x_{j_i}, p_{k_i}) \propto \exp\left(-\frac{1}{2}\left\|\Pi(x_{j_i}, p_{k_i}) - z_i\right\|^2_{\Sigma_i}\right) \quad (1)$$

where $\Pi(x,p)$ is the stereo projection of a 3D point p into a camera of given 3D pose x, yielding the predicted stereo projections (u_L, v) and (u_R, v), $z_i = (\hat{u}_L, \hat{u}_R, \hat{v})$ is the actual stereo measurement, and Σ_i represents the Gaussian image measurement noise. iSAM then finds the least-squares estimate Θ^* of all variables Θ (camera poses and scene structure combined) as

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \prod_i f_i(\Theta_i) \quad (2)$$

In order to reduce the computational requirements of the approach we employ a pose decimation scheme, whereby a threshold is applied on the translational and rotational motion of the sensor between poses. In our current implementation the first pose of each session corresponds to the first frame of the input sequence and is initialised to the 6-DOF origin for that session. Subsequent to this each new pose corresponds to the first frame where the inter-frame motion is at least 0.2m or 0.2rad from the last pose. The effect of this is to reduce the total number of poses within the pose graph, whilst maintaining the accuracy of the final trajectory and map estimates. A secondary but important advantage of this approach is that it also decreases drift, in particular when the sensor is stationary.

When the smoothing window reaches a maximum size or when a loop closure is detected all poses and associated odometry are transferred to the current session's pose graph, and a new local window is initialised. By including all poses from a window, as opposed to just the first or first and last pose (as is the case in other approaches) we ensure that we can represent loop closures between arbitrary frames within the pose graph. Full details of the loop closure handling are provided in Section 3.5. To initialise a new window we use the last pose of the previous window in conjunction with all landmarks that correspond to features that are tracked into the current frame.

The pose graph is again optimised using the iSAM library [18], but this time using the actual incremental iSAM algorithm [20] to efficiently deal with large pose graphs. In contrast to the stereo projection factors f_i in the smoothing formulation above, we now use factors g_i

$$g_i(\Theta_i) = g_i(x_{j_i}, x_{j'_i}) \propto \exp\left(-\frac{1}{2}\left\|(x_{j'_i} \ominus x_{j_i}) - c_i\right\|^2_{\Xi_i}\right) \quad (3)$$

that represent constraints c_i with covariances Ξ_i between pairs of poses as obtained by local smoothing or by loop closure detection. We use the notation $x_d = x_b \ominus x_a$ from Lu and Milios [21] for representing pose x_b in the local frame of pose x_a ($x_b = x_a \oplus x_d$).

3.3. Quaternions and homogeneous points

The projective representation of point features as well as the quaternion representation of rotations that we use here result in over-parameterisations that can be resolved in much the same way.

The solution for quaternions is well known; an accessible explanation can be found in Grassia [22]. The unit sphere $S^3 = \left\{\mathbf{q} \in \mathbb{R}^4 : \|\mathbf{q}\| = 1\right\}$ can be identified with the set of unit quaternions which form a 3-dimensional Lie group under quaternion multiplication. There is a two-to-one covering map from S^3 onto $SO(3)$ (antipodal points are identified because $\mathbf{q}$ represents the same rotation as $-\mathbf{q}$). The matrix Lie algebra of $SO(3)$ is $\mathfrak{so}(3)$, the set of skew-symmetric matrices, see Hall [23]. Because they have three parameters they provide a minimal local parameterisation of rotations through an exponential map (typically evaluated using Rodrigues' formula). The elements of the Lie algebra of S^3 can be identified with the tangent space $\mathbb{R}^3$ at the identity. Many mappings exist from $\mathbb{R}^3$ to S^3, here we use the following one from Grassia [22]:

$$\exp\left(\frac{\mathbf{d}}{2}\right) = \begin{pmatrix} \frac{1}{2}\operatorname{sinc}\left(\frac{1}{2}\|\mathbf{d}\|\right)\mathbf{d} \\ \cos\left(\frac{1}{2}\|\mathbf{d}\|\right) \end{pmatrix} \quad (4)$$

where $\mathbf{d} \in \mathbb{R}^3$ coincides with the axis/angle representation of a rotation. As for every minimal representation of rotations there are singularities, here at multiples of 2π, though they can be avoided by forcing $\mathbf{d}$ to fall into the range $(-\pi, \pi]$, while still allowing for all possible rotations. An existing quaternion $\mathbf{q}$ is updated by an increment $\mathbf{d}$ using quaternion multiplication $\mathbf{q}\exp\left(\frac{\mathbf{d}}{2}\right)$.

We show that the projective parameterisation in 3D is isomorphic to unit quaternions, allowing the use of the same exponential map. The projective parameterisation uses homogeneous four-vectors $\mathbf{p} = (x,y,z,w)^\top \in \mathbb{R}^4 \backslash \{0\}$ with the zero vector excluded. A Euclidean point (x,y,z) is written in homogeneous coordinates as $\lambda(x,y,z,1)$ for $\lambda \in \mathbb{R}\backslash\{0\}$, while points at infinity satisfy $w = 0$. In this real projective space $\mathbb{RP}^3$, points along lines through the origin are equivalent by the relation $\mathbf{p} \sim \lambda\mathbf{p}, \lambda \in \mathbb{R}\backslash\{0\}$. The set of homogeneous points $\mathbf{p} \in \mathbb{R}^4, \|\mathbf{p}\| = 1$ with unit norm spans the 3-sphere S^3 and provides a double cover of this real projective space (antipodal points are identified). Therefore, the same exponential map as for quaternions is applicable to normalised homogeneous points.

An alternative map is presented in Hartley and Zisserman [17, Appendix 6.9.3] that uses Householder transformations instead of quaternion multiplication. Both are valid, and we have not seen a major difference in their convergence. Note that both methods only work well for bundle adjustment as long as the cameras are near the origin, which is easy to satisfy for our windowed bundle adjustment. Intuitively, the parameterisation becomes more nonlinear for cameras with center far from the origin.

3.4. Place recognition

Place recognition is an important component in the context of large-scale, multi-robot and multi-session SLAM, where algorithms based on visual appearance are becoming more popular when detecting locations already visited, also known as loop closures. In this work we have implemented a place recognition module based on the recent work of [24, 25], which demonstrated robust and reliable performance.

The place recognition module has the following two stages:

- The first stage is based on the bag-of-words (BoW) method [26], which is implemented in a hierarchical way [27]. This implementation enables quick comparisons of an image at time t with a database of images in order to find those that are similar according to a normalized similarity score η_c. Then, there are three possibilities: if $\eta_c \geq \alpha^+$, the match is considered highly reliable and accepted, if $\alpha^- < \eta_c < \alpha^+$, the match is checked by conditional random field (CRF)-Matching in the next step, otherwise the match is ignored. In our implementation, η_c is the ratio between the BoW score computed between the current image and the candidate and the image one second ago in the database, as follows:

$$\eta_c(t,t') = \frac{s(t,t')}{s(t,t-1)} \tag{5}$$

  The minimum confidence expected for a loop closure candidate is $\alpha^- = 0.15$ and for a loop closure to be accepted is $\alpha^+ = 0.8$. For each session a new image is added to the database whenever the sensor's motion exceeds a threshold of 0.2m or 0.2rad based on the output from frontend's motion estimation.

- The second stage consists of checking the previous candidates with CRF-Matching in 3D and image spaces (near and far information). The CRF-Matching approach is an algorithm based on Conditional Random Fields (CRF) [28] proposed for matching 2D laser scans [29] and for matching image features [30]. CRF-Matching is a probabilistic model that is able to jointly reason about the association of features. In [24] CRF-Matching was extended to reason in 3D space about the association of data provided by a stereo camera system in order to verify loop closures hypothesis. This verification stage was improved in [25] taking into account the far information, the remaining information in one image without 3D information. We compute the negative log-likelihood $\Lambda^{\mathscr{G}}_{t,t'}$ from the maximum a posteriori (MAP) association between the current scene in time t against the candidate scene in time t'. We accept the match only if the normalized similarity scores assert $\eta_{3D} \leq \beta_{3D} \wedge \eta_{Im} \leq \beta_{Im}$, with:

$$\eta_{\mathscr{G}} = \frac{\Lambda^{\mathscr{G}}_{t,t'}}{\Lambda^{\mathscr{G}}_{t,t-1}} \tag{6}$$

  where $\mathscr{G}$ indicates the graph, $3D$ or Im. $\beta_{\mathscr{G}}$ is a control parameter that defines the level of similarity we demand for $(t,t-1)$ in terms of close range β_{3D}, and far range β_{Im}. Where smaller values for β means a higher demand. In our current implementation we use $\beta_{3D} = \beta_{Im} = 2$. Fig. 2 summarises the steps involved in the place recognition scheme.

This place recognition module exploits the efficiency of the BoW to detect revisited places in real-time. CRF-Matching is a more computationally demanding data association algorithm because it uses much more information than BoW. For this reason, only the positive results of BoW are verified by CRF-Matching.

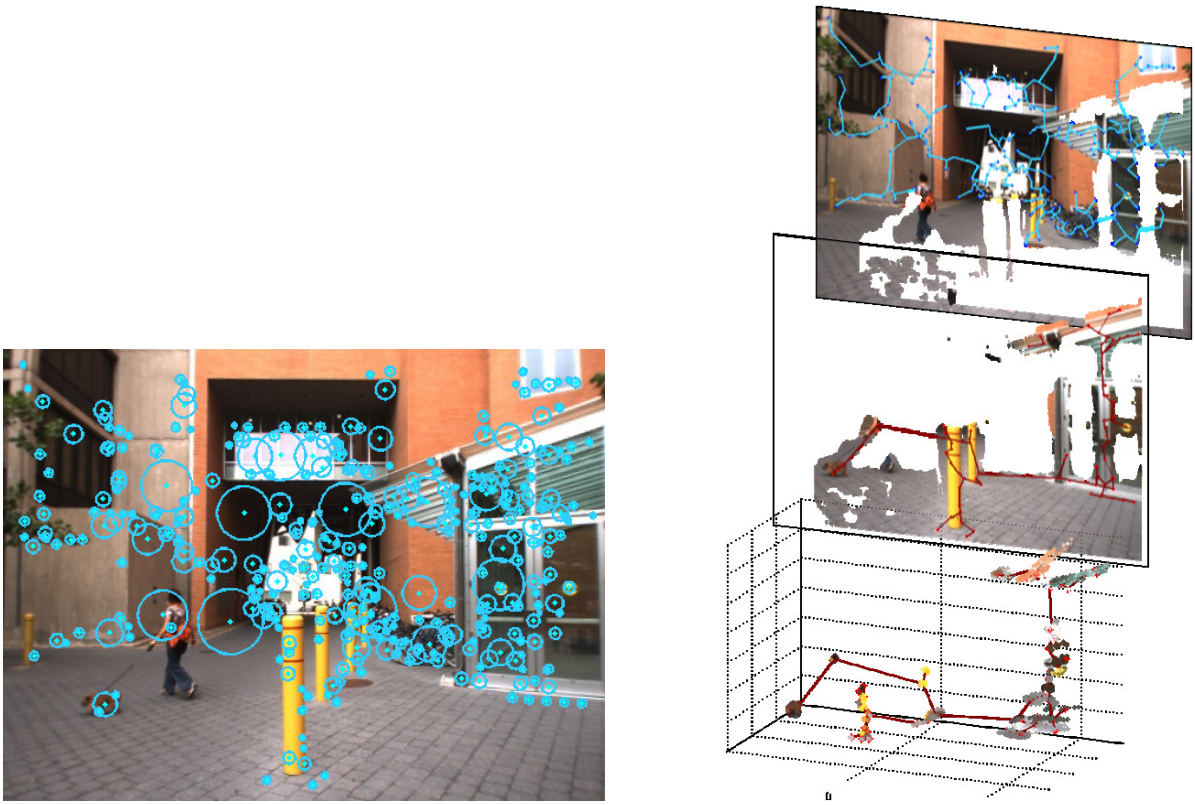
3.5. Multi-session visual SLAM

For multi-session mapping we use one pose graph for each robot/camera trajectory, with multiple pose graphs connected to one another with the help of "anchor nodes", as introduced in Kim *et al.* [3] and Ni and Dellaert [4].

In this work we distinguish between intra-session and inter-session loop closures. Processing of loop closures is performed firstly with each frame corresponding to a new pose being input to the above place recognition system. These candidate frames are matched against previously input frames from all sessions. On successful recognition of a loop closure the place recognition system returns the matched frame's session and frame identifier in conjunction with a set of stereo feature correspondences between the two frames. These feature sets consist of lists of SURF feature locations and stereo disparities. Note that since these features are already computed and stored during the place recognition processing, their use here does not place any additional computational load on the system.

These feature sets serve as input to the same camera orientation estimation system described in Section 3.1. Here the disparities for one of the feature sets are used to perform 3D reconstruction of the points in the scene. These 3D points are passed with their corresponding 2D features from the second image into a 3-point algorithm-based RANSAC procedure. Finally the estimated orientation is iteratively refined through a non-linear optimisation procedure that minimises the re-projection error in conjunction with the disparity.

Inter-session loop closures introduce encounters between pose graphs corresponding to different visual SLAM sessions. An encounter between two sessions s and s' is a measurement that connects two robot poses x^s_j and $x^{s'}_{j'}$. This is in contrast to measurements between poses of a single trajectory, which are of one of two types: The most frequent type of measurement connects successive poses, and is derived from visual odometry



(a) BoW step **(b)** CRF step

Figure 2: Illustration of steps involved in the combined BoW-CRF place recognition scheme. (a) For each input stereo pair we compute a set of SURF-features for one image of the stereo pair to provide the input of the BoW stage. (b) For the CRF stage we compute the two minimum spanning trees (MST), one for features with 3D information (near features), and the second for the remaining ones, with image information (far features). In (b), we show the two resulting graphs: in blue the graph for far features ($\mathcal{G}_{Im}$), in dark red the graph for near features ($\mathcal{G}_{3D}$). We apply CRF-Matching over both graphs. The minimum spanning tree of $\mathcal{G}_{3D}$ is computed according to the metric coordinates, projected over the middle image only for visualisation. In the bottom, we show $\mathcal{G}_{3D}$ in metric coordinates with the 3D point cloud (textured) of each vertex in the tree. The MST provides a representation of the dependencies between features in a scene, and allows for robust consistency checks of feature associations between scenes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and the subsequent local smoothing. A second type of measurement is provided by intra-session loop closures.

The use of anchor nodes [3] allows at any time to combine multiple pose graphs that have previously been optimised independently. The anchor node Δ^s for the pose graph of session s specifies the offset of the complete trajectory with respect to a global coordinate frame. That is, we keep the individual pose graphs in their own local frame. Poses are transformed to the global frame by pose composition $\Delta^s \oplus x_i^s$ with the corresponding anchor node.

In this relative formulation, the pose graph optimisation remains the same, only the formulation of encounter measurements involves the anchor nodes. The factor describing an encounter between two pose graphs will also involve the anchor nodes associated with each pose graph. The anchor nodes are involved because the encounter is a global measure between the two trajectories, but the pose variables of each trajectory are specified in the session's own local coordinate frame. The anchor nodes are used to transform the respective poses of each pose graph into the global frame, where a comparison with the measurement becomes possible. The factor h describing an encounter c_i is given by

$$h(x_j^s, x_{j'}^{s'}, \Delta^s, \Delta^{s'}) \propto \exp\left(-\frac{1}{2}\left\|((\Delta^s \oplus x_j^s) \ominus (\Delta^{s'} \oplus x_{j'}^{s'})) - c\right\|_\Gamma^2\right) \tag{7}$$

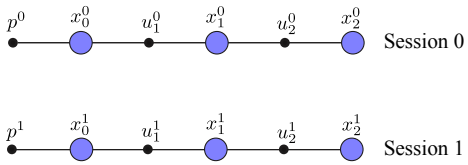
where the index i was dropped for simplicity. These inter-session constraints are incorporated into the cost function defined in Eq. 2 in the same way as the other measurements. Fig. 3 provides a graphical example of this process [3]. The concept of relative pose graphs generalises well to a larger number of sessions. The number of anchor nodes depends only on the number of sessions.

Finally we note that although the PR system currently returns only the best match, it is possible to employ other strategies. For example the current architecture could be extended to allow the PR system to return multiple candidates. Here, each candidate would then be processed independently, i.e. by performing the consistency check, etc., prior to being integrated into the posegraph.

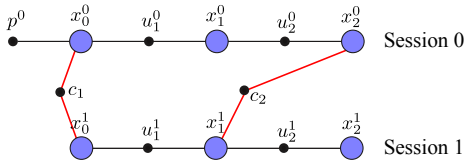
4. Experiments and results

In this section we assess the performance of our system in a number of different scenarios, using a dataset that was collected at the Ray and Maria Stata Center at MIT over a period of months. This building is known for its irregular architecture and provides a good testing ground for visual SLAM techniques in general.

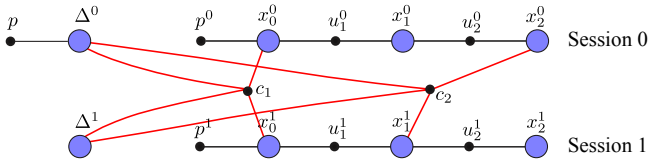
The dataset includes indoor, outdoor, and mixed sequences captured using robotic and manually wheeled platforms and a handheld camera with full 6-DOF movement (e.g. ascending and descending stairs, etc.). All image sequences were captured using a Point Grey Bumblebee colour stereo camera with a baseline of 11.9cm and where both lenses had a focal length of 3.8mm. The sensor resolution of each camera was 1024×768 pixels which we subsampled to 512×384 pixels



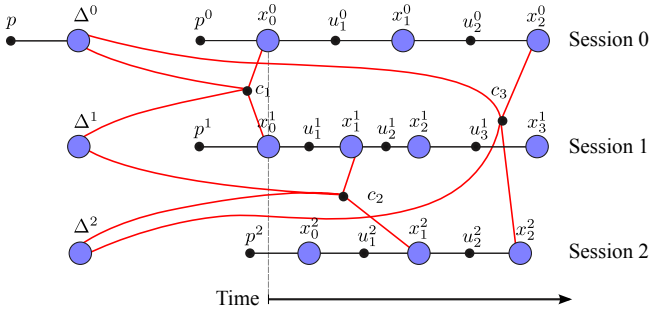
(a) Two pose graphs for two sessions without encounters. Each trajectory is anchored by a prior p^s on its first pose that can be chosen arbitrarily (typically chosen to be the origin). For simplicity, we omit the concept of loop closing constraints here.



(b) Two encounters expressed as additional constraints connecting the pose graphs of the two sessions. Note that although in the original work of Kim *et al.* [3] encounters can be based on one robot observing the other (yielding a synchronized constraint such as c_1), in this work we only consider encounters due to common observations of the environment detected by the place recognition module (yielding constraints that connect poses created at arbitrary times).



(c) The same encounters as in (b), but using the relative formulation from Kim *et al.* [3]. Here anchors Δ^s are introduced for each trajectory that specify the offset of the trajectory with respect to a common global frame. Each encounter measurement now additionally connects to the anchors of both trajectories.



(d) The relative pose graph formulation generalizes to more than two sessions, and does not require the pose graphs to be synchronized or even to start at the same time or location. This example shows three encounters between three sessions.

Figure 3: Illustration of the use of anchor nodes for multiple relative pose graphs, from Kim *et al.* [3]. In contrast to typical SLAM, pose constraints in our system can influence more than two variables. For visualization, we therefore use the factor graph representation, which directly corresponds to the entries in the measurement Jacobian as described in [31]. Small black discs represent measurements (factors), and larger blue shaded circles represent variables. The lines indicate dependencies, where black lines are used for normal pose constraints and red lines for encounters.

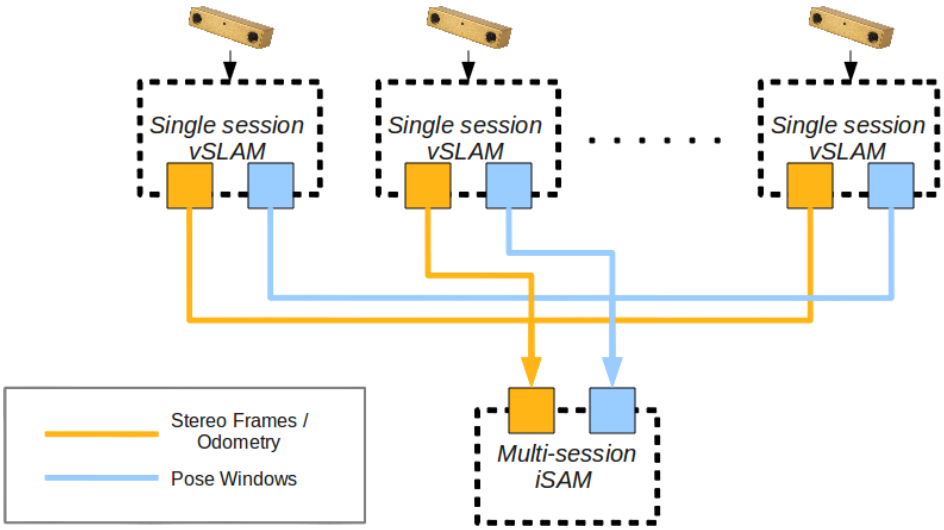


Figure 4: Multi-session visual SLAM architecture, see Fig. 1 for more details on the single-session frontend and the multi-session backend.

prior to processing. The wheeled platforms also included horizontally mounted 2D LiDAR scanners. Although we do not use the LiDAR sensors in our system, the accompanying laser data allows us to compare the performance of our technique to that of a laser-based scan matcher in restricted wheeled platform scenarios (see Section 4.1 for details).

The complete multi-session visual SLAM system follows the architecture shown in Fig. 4. Here each input image sequence constitutes a separate session and is processed independently by a dedicated vSLAM frontend. The output of each frontend is processed by the multi-session backend, which is responsible for multi-session place recognition and pose graph estimation.

The internal components of the frontend and backend (see Fig. 1) are implemented as a set of loosely coupled processes that communicate using the *Lightweight Communications and Marshalling* (LCM) robot middleware system [32]. This permits straightforward parallelism between the components of the system, hence minimising the impact on all modules due to fluctuations in the load of a particular module (e.g. due to place recognition deferring to CRF processing). Furthermore the overall architecture can be transparently reconfigured for different setups (e.g. from single CPU to multi-core or distributed processing).

In the remainder of this section we provide both a qualitative and quantitative assessment of the system's performance. The quantitative assessment is based on three separate experiments; one which assesses the system on a single-session scenario and two which assess the system on multi-session scenarios. Details of the datasets used in each of the experiments are provided in Table 1. For each of the experiments, processing was carried out on an Intel® Core™ i7 940 2.93GHz based machine with 8GB of RAM and an nVidia® GeForce® 9800 GT graphics card.

	Experiment 1	Experiment 2	Experiment 3
Indoor/Outdoor	Indoor	Indoor	Outdoor
Num. sessions	1	4	3
Seq. length	20.3 min	45 min	4.7 min
Num. poses	883	1562	1406
Intra-session	112	4	0
Inter-session	0	260	13

Table 1: Description of experimental datasets.

4.1. Single-session visual SLAM results

In this section we provide results from a number of single session SLAM experiments. To do this we have applied the system in single session mode (i.e. only running a single frontend) across a variety of sequences from the Stata Center dataset. The results show that the system is capable of operating over extended sequences in both indoor, outdoor and mixed environments with full 6-DOF motion.

For example, two feature-based maps for outdoor sequences are shown in Fig. 5. Here, for (a), the underlying grid is at a scale of 10m, where the trajectory is approximately 100m in length. An example image from the sequence is shown in the inset with the GPU KLT feature tracks overlaid on the left frame. Fig. 5 (b) shows a similar scale sequence that includes full 6-DOF motion, where the user has carried a handheld camera up a stairs.

To evaluate the accuracy of the frontend we compare its trajectory estimate to that of a scanmatching algorithm applied to the corresponding LiDAR data. In the absence of loop closures we have found the system to have drift of approximately 1%-3% in position during level motion (i.e. without changes in pitch angle). To demonstrate this, Fig. 6 shows two maps with two trajectories, both taken from the same sequence. The black contour shows the 2D LiDAR scanmatcher-based map. The scanmatcher's estimated pose is shown by the dark blue trajectory, which can be seen more clearly in the lower right-hand inset. The distance between grid lines in the figure is 2m. From the

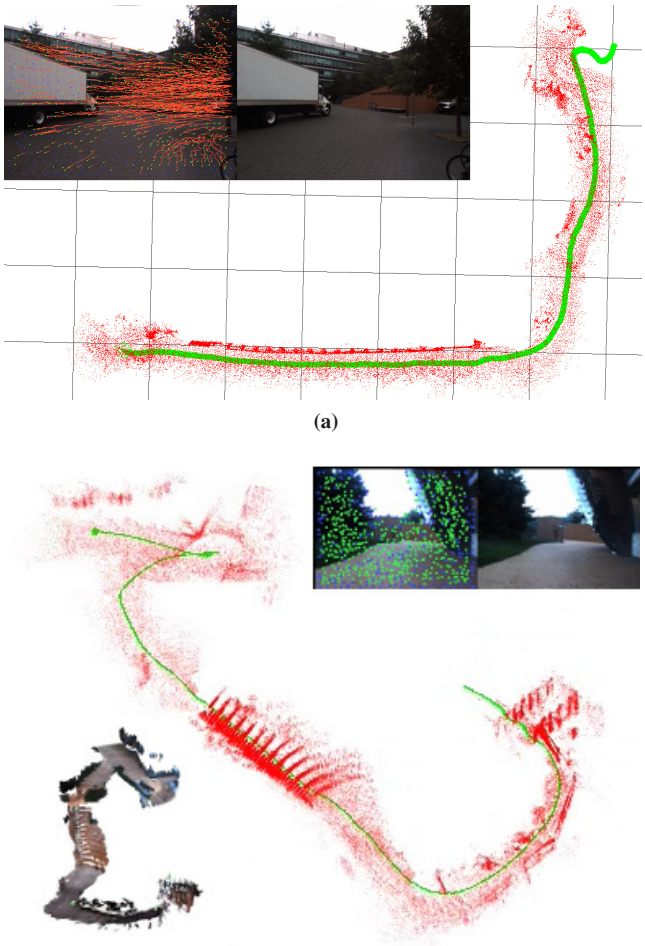


Figure 5: Single session visual SLAM processing including full 6-DOF motion.

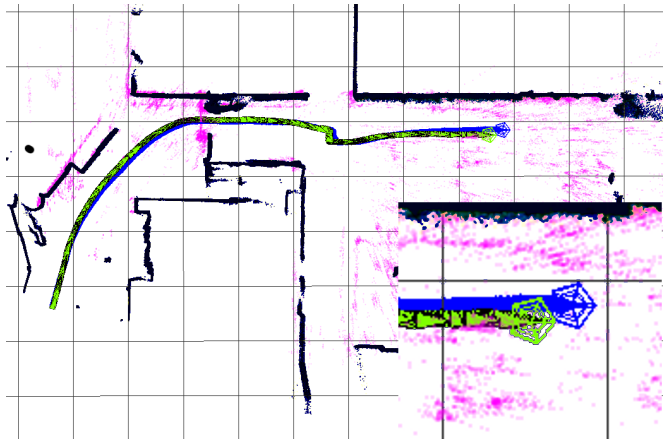


Figure 6: Comparison of drift in single session visual SLAM against 2D LiDAR scan matcher over a 20m trajectory. Grid scale is 2m. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

figure the horizontal displacement of the final poses is approximately 60cm with a total trajectory of approximately 20m.

An example of the accumulated error in position due to drift is shown in Fig. 7. Here the dataset consists of an image sequence taken over an indoor area within in the Stata Center. The grid is at a scale of 5m with the sequence taken by travelling on a large loop over a space of approximately 35m$\times$15m. The image at the top shows the result of the motion estimate in the absence of a loop closure. The majority of the drift in this example is due to the tight turn approximately two-thirds of the way through the sequence, where the divergence between each traversal of the hallway can be seen.

The center figure shows the result of the correction applied to the pose graph due to a sequence of loop closures occurring at the area highlighted by the red box. Here it can seen that the pose graph sections showing the traversals of the hallway are much more coincident and that the misalignment in corresponding portions of the map is reduced considerably. The figure also shows the accuracy of the map relative to the ground truth CAD floorplan.

As mentioned in the previous section, in order to measure the computational requirements of the system we evaluated the processing times of each component, based on three sets of sequences, each drawn from the Stata dataset. The output map and trajectory for Experiment 1: the single session indoor sequence (see Table 1) is shown in Fig. 8. In total the input sequence was 20.3 minutes in length (i.e. 24360 frames). Also shown in the table is the final number of poses in the pose graph, and the number of intra-session loop closures. As can be seen from the results, the pose decimation scheme described in Section 3.2 significantly reduces the number of poses, in this case to 883. It is important to point out that the ratio of poses to frames in this example is low due to the fact that the camera is mounted on a B21 robot which is moving slowly through the environment. As will be seen in the next section, when the camera moves quickly such as with handheld sequences this ratio is approximately an order of magnitude higher.

Table 2 provides summaries of the run-times for the principal modules of the system, where the first two columns provide the mean and variance for Experiment 1. Note that the times for feature tracking and stereo odometry modules correspond to the computational cost for each frame of the input sequence. However given that each of the iterations of the windowed bundle adjustment and place recognition modules only occur on each new keyframe (i.e. pose), the times provided are per keyframe. Finally since the pose graph updates only occur when the frontend window reaches 15 poses or whenever a loop closure occurs, the times shown in this row are per window. Given that each of the modules runs as a separate process, with communication handled via LCM, each task is handled by a separate core of the CPU.

Fig. 9a-9c provide plots of the computation time for each iteration of the windowed bundle adjustment, place recognition, and pose graph iSAM, respectively. As can be seen from Fig. 9c the total number of updates to the pose graph is approximately 160 with a total of 883 poses in the final pose graph. Also, from this graph it is possible to see (i) the increase in computation time as the size of the pose graph grows, and (ii) the impact due

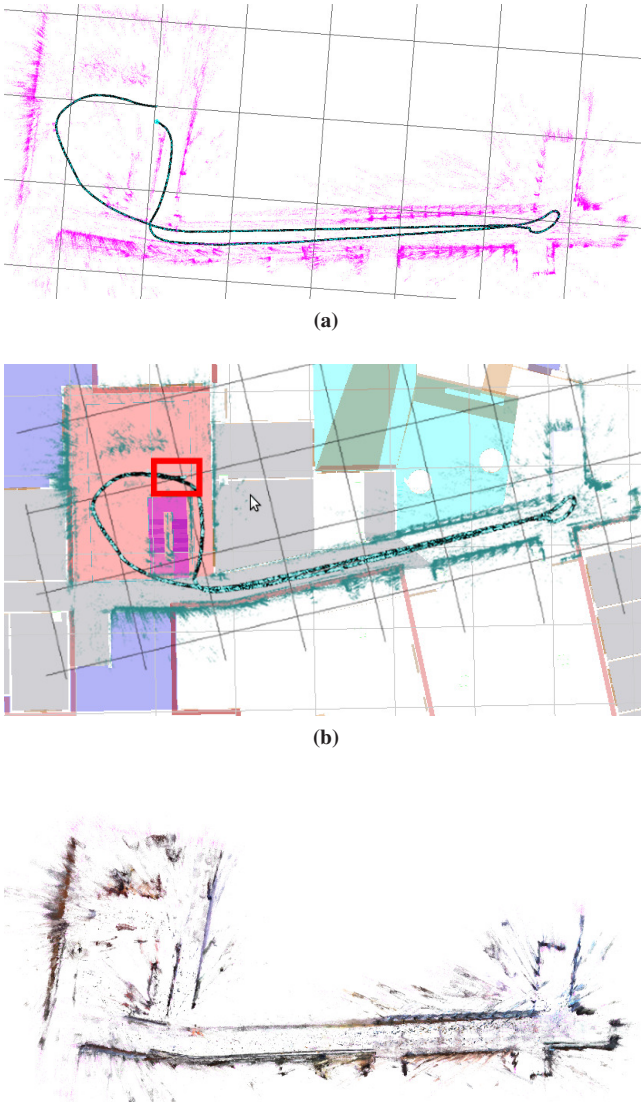


Figure 7: Single-session dataset containing a large loop. Here the grid scale is at 5m. (a) Map and pose graph prior to loop closure showing drift in position and structure. (b) Map and pose graph showing correction in the position and structure due to a series of loop closures in the area shown by the red square. Background image shows ground truth CAD floorplans of the environment. (c) Textured version of figure (b). (In order to fully interpret this figure the reader is referred to the online colour version.)

to loop closures.

Fig. 9d-9e provide plots of the cumulative computation time as a function of processed input sequence time for each of the modules in the frontend and backend respectively. Given that each of these modules runs in parallel, each on a separate core of the CPU, it is clear from this graph that the overall system is capable of running in real-time.

Although the odometry system has shown to be robust over maps of the order of hundreds of meters, two failure modes for the system are due to tracker failure during (i) high-speed motion, and (ii) low-texture or low-contrast environments, which can also cause errors whereby the disparity estimation fails over a large set of features. In the current system we address this through reinitialising the tracker and inserting a new pose where the motion relative to the previous pose is set to zero with large covariance. Hence we keep the two sections of the pose graph (i.e. at either side of the failure) topologically connected whilst capturing the high degree of uncertainty between them. This is done with the intention that future loop closures between the sections will provide adequate constraints to correct for this uncertainty.

A frame where such an odometry failure occurred in Experiment 1 is shown in the right-most frame at the top of Fig. 8 (i.e. image D). Here lack of texture in the environment results in a low feature count and hence an inability to estimate the camera's motion. Fig. 8 also highlights the location of the frame in the map. At the point in the sequence where this failure occurs the map diverges, however a subsequent loop closure close to point C in the map corrects for this drift. As can be seen from comparison, the final estimated structure is in close agreement with the ground truth floor plan.

We note that the above approach can be avoided for short-term tracking failures by incorporating inertial sensors. For longer tracking failures, an alternative approach that we are currently investigating is the possibility of using multi-session SLAM, whereby odometry failure results in the creation of a new session with a weak prior on the initial position. This disjoint session is treated the same as any other session. When a new encounter does occur, the session can be reconnected to the global pose graph.

4.2. Multi-session visual SLAM results

To evaluate the multi-session performance of the system we tested it on the Experiments 2 and 3 datasets detailed in Table 1. The rationale for choosing the datasets was that the Experiment 2 dataset contains, as a subset, the single session dataset from Experiment 1, and therefore allows a direct comparison to be made between the system's single- and multi-session operation. The sequences used in Experiment 3 differ from the other datasets in that they were captured using a handheld Bumblebee camera in an outdoor environment and contain subsequences where the user ascends and descends stairs, and hence provide a much larger range of motion in all 6-DOF.

Fig. 11 provides a number of different views of the output of the system for the Experiment 2 dataset. Fig. 11a – 11d show the results for the individual sessions including trajectories and

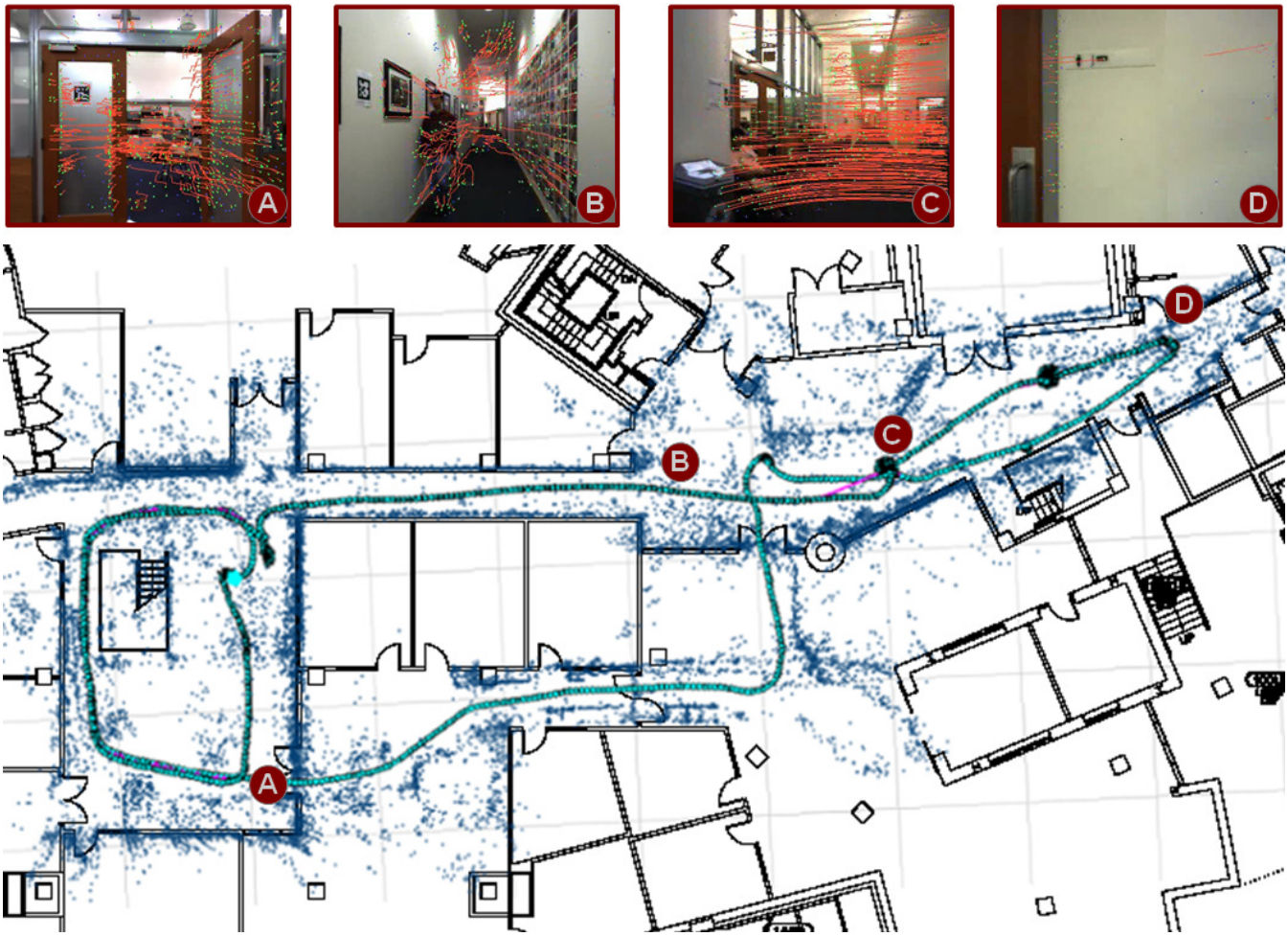
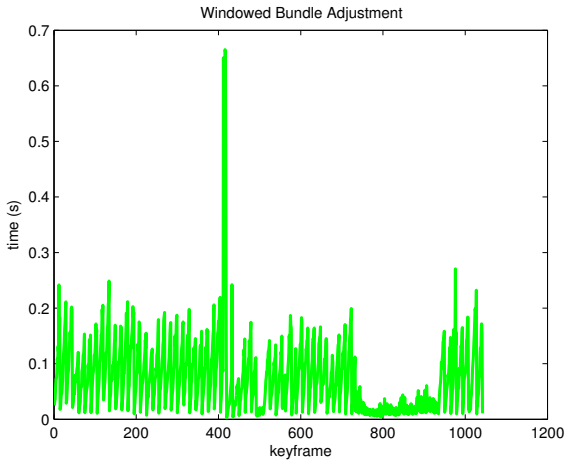


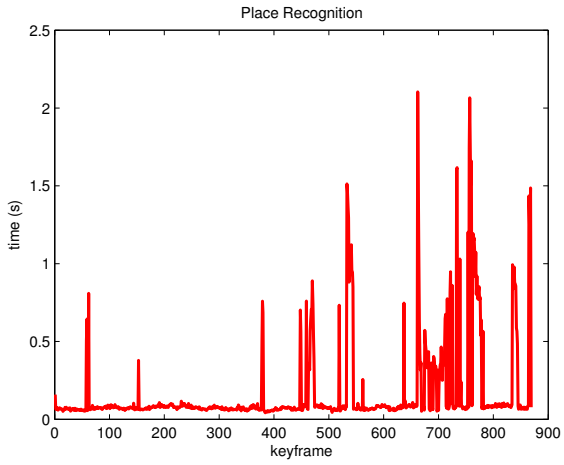
Figure 8: Top-down orthographic projection of the results for the Experiment 1 single session dataset described in Table 1. Here the map is overlaid on an architectural floorplan for comparison with ground-truth. The reference grid is shown at a scale of 5m. Images A–D displayed above the map show sample frames from the sequence with their approximate location highlighted on the map. Further details can be found in Section 4.1. (In order to fully interpret this figure the reader is referred to the online colour version.)

	Experiment 1 Single Session Indoor		Experiment 2 Multi-session Indoor		Experiment 3 Multi-session Outdoor	
	Mean(s)	Variance(s)	Mean(s)	Variance(s)	Mean(s)	Variance(s)
Feature tracking	0.0309	3.29×10^{-5}	0.0315	4.90×10^{-5}	0.0374	4.8×10^{-5}
Stereo odometry	0.0084	1.61×10^{-4}	0.0083	1.62×10^{-4}	0.0021	5×10^{-5}
WBA	0.0755	0.0048	0.0665	0.0029	0.0611	0.0026
Place recognition	0.1703	0.0832	0.1794	0.0792	0.1049	0.0148
iSAM	0.0996	0.0040	0.1960	0.0242	0.1242	0.0170

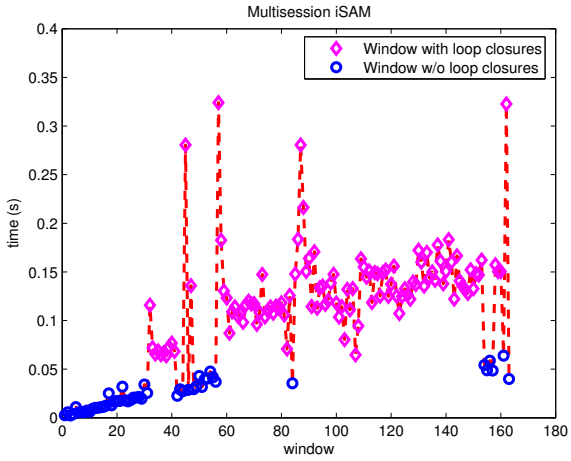
Table 2: Mean runtimes and variances for each iteration of each module of the system for each of the experiments reported in the paper. Feature tracking and stereo odometry are executed for each frame of the input sequence. Windowed Bundle Adjustment (WBA) and Place Recognition are executed for each pose. Incremental updates to the pose-graph are computed via iSAM once for each window.



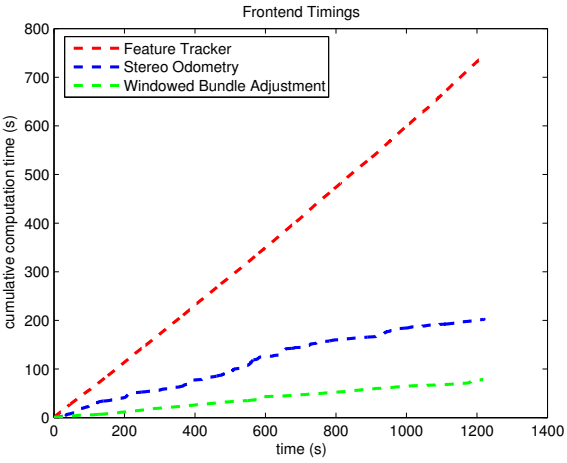
(a)



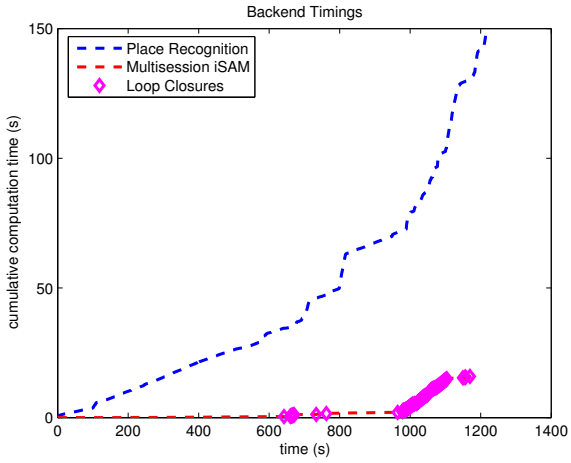
(b)



(c)



(d)



(e)

Figure 9: Timings for Experiment 1: single session indoor sequence. (a) and (b) show the processing times per keyframe for windowed bundle adjustment and place recognition, respectively. (c) shows the processing time for multisession iSAM for each window, distinguishing between windows with and without loop closures. (d) and (e) show the cumulative processing time for the individual modules of the frontend and backend, respectively.

maps. Fig. 11e shows the trajectories where the elevation increases with time and the purple links between the trajectories correspond to the intra- and inter-session loop closures. Finally, Fig. 11f shows the complete multisession pose-graph and map including all four sessions. This is overlaid on a ground-truth floorplan of the corresponding region of the Stata Center.

As reported in Table 1, in this experiment the input dataset consisted of 4 separate sessions, which, when combined, corresponded to a total of 45 minutes of video at 20 fps. The final multi-session pose graph contains 1562 poses and spans an area of approximately $75m \times 25m$. Comparing the output to that of Experiment 1, the ratio of the number of poses in the final trajectory to the length of the input video sequence is of the same order. Timings for each of the modules of the system for this dataset are given in columns 3 and 4 of Table 2.

Fig. 10 provides a set of plots for the module execution times for Experiment 2 equivalent to those shown for Experiment 1 in Fig. 9. As can be seen from the multi-session results, each of the modules except for the multi-session iSAM module have similar mean execution times to the single-session operation. The reason for the difference in iSAM is due to the fact that the complexity of the optimisation is a function of the number of poses in the pose graph.

Fig. 13 shows the estimated map and trajectory for the Experiment 3 dataset, where again the grid is at a scale of 5m. Fig. 13a shows a plan view of the map, where it can be seen that the total area covered by the 3 sessions is approximately $110m \times 80$m. A side view of the map where the 6DOF motion of the camera is apparent is shown in Fig. 13b. Given that the sequences in this dataset are taken from a handheld camera the motion of the sensor is at much higher velocities and as a consequence the ratio of the number of poses to the number of frames processed is an order of magnitude higher than in the two previous experiments. In particular, although the total length of the image sequences is 4.7 minutes compared to the 45 minutes of the indoor multi-session experiment, the number of poses are within 10% of each other.

Columns 5 and 6 of Table 2 provide details of the timings for Experiment 3. Two important differences with the outdoor dataset were that the appearance of the scene was far more textured and as such the disparity estimation and 3D feature tracking was more reliable. The effect of this can be seen in the speed up of the stereo odometry computation which was principally due to significantly less iterations of the RANSAC procedure. Detailed plots of the timings for Experiment 3 are shown in Fig. 12.

One issue encountered during the outdoor sequences was intra-frame aliasing of SURF features (e.g. due to the repeated red bricks). This resulted in a number of true positive loop closures from the place recognition system being rejected due to a failure of the geometric consistency test, which was in turn due to a failure of the SURF correspondence estimation. For example in Experiment 3 the total number of loop closures was 13. This is the reason why, although the total number of poses is similar to Experiment 2, the iSAM processing time is lower.

5. Conclusions

In this paper we have presented a real-time 6-DOF multi-session visual SLAM system. The principal contribution of the paper is to integrate all of the components required for a multi-session visual SLAM system using iSAM with the anchor node formulation [3]. In particular this is the first example of an anchor node-based SLAM system that (i) uses vision as the primary sensor, (ii) operates in general 6-DOF motion, (iii) includes a place recognition module for identifying encounters in general environments, and (iv) derives 6-DOF pose constraints from those loop closures within these general environments (i.e. removing the need for fiducial targets, as were used in [3]).

We have demonstrated this system in indoor and outdoor environments using both wheeled and handheld sensors as input. We have presented examples of single- and multi-session pose graph optimisation and map construction, and provided a comprehensive quantitative assessment of the system's performance in a number of different scenarios.

Multi-session visual mapping can provide a solution to the problem of large-scale persistent localisation and mapping. In the future we plan to extend the results published here to incorporate the entire Stata dataset described in the Section 4. Furthermore we intend to evaluate the approach in online collaborative mapping scenarios over extended timescales.

Acknowledgments

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the Irish National Development Plan, by the U.S. Office of Naval Research (ONR) grants N00014-10-1-0936 and N00014-12-1-0093, and by the Dirección General de Investigación of Spain under projects DPI2009-13710, DPI2009-07130. The authors gratefully acknowledge this support.

The authors would like to thank Hordur Johannsson and Maurice Fallon for their assistance in the collection of the Stata datasets, and David Rosen for his comments on Section 3.3. The authors would also like to thank the reviewers for their comments.

References

[1] J. Neira, A. Davison, J. Leonard, Guest editorial special issue on visual SLAM, IEEE Trans. Robotics 24 (5) (2008) 929–931, ISSN 1552-3098, doi:10.1109/TRO.2008.2004620.

[2] M. Cummins, Probabilistic localization and mapping in appearance space, Ph.D. thesis, University of Oxford, 2009.

[3] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, S. Teller, Multiple relative pose graphs for robust cooperative mapping, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), Anchorage, Alaska, 3185–3192, 2010.

[4] K. Ni, D. Steedly, F. Dellaert, Tectonic SAM: Exact, out-of-core, submap-based SLAM, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), 1678–1685, 2007.

[5] J. McDonald, M. Kaess, C. Cadena, J. Neira, J. Leonard, 6-DOF multi-session visual SLAM using anchor nodes, in: European Conference on Mobile Robotics, Örbero, Sweden, 2011.

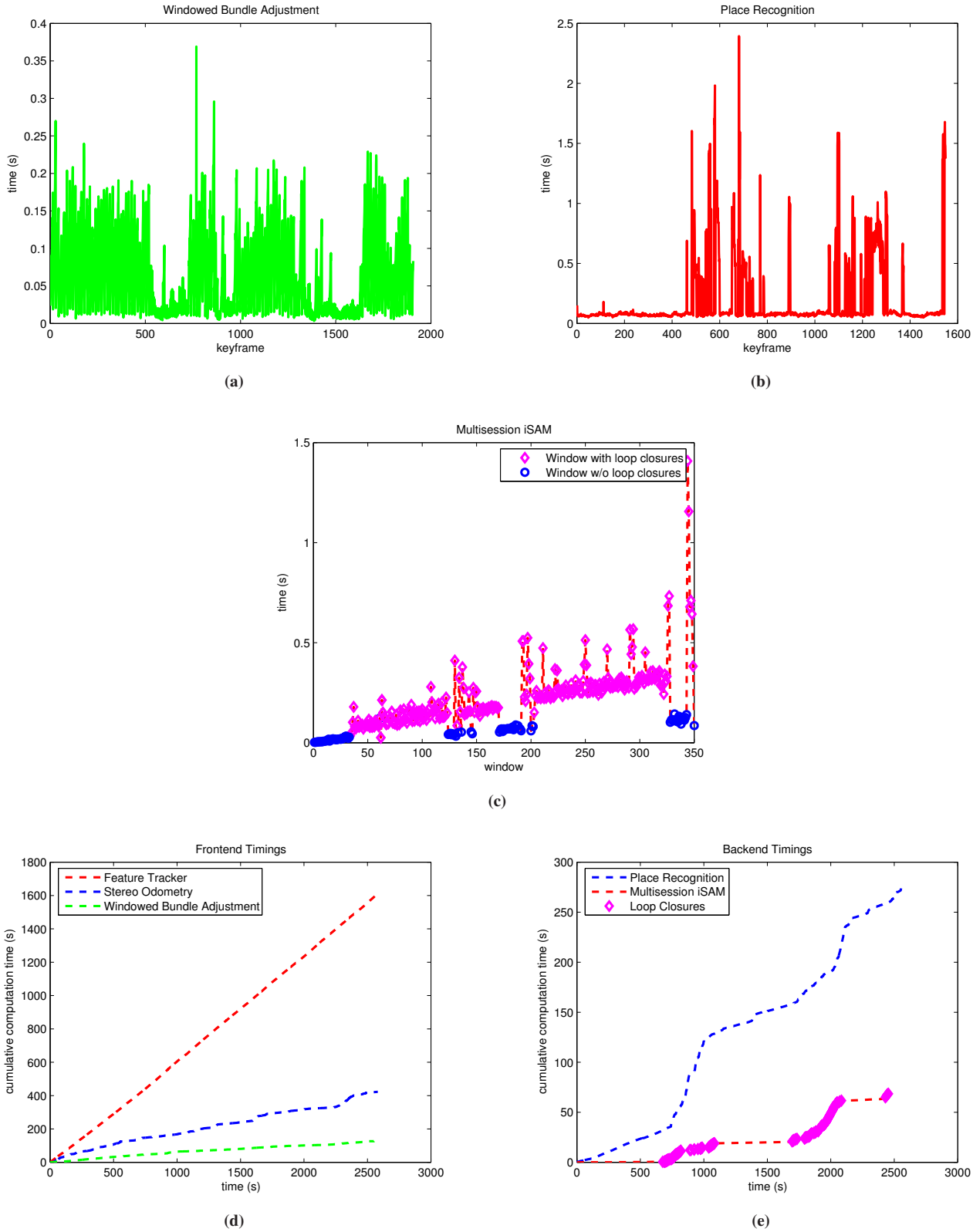


Figure 10: Timings for Experiment 2: multi-session indoor sequences. (a) and (b) show the processing times per keyframe for windowed bundle adjustment and place recognition, respectively. (c) shows the processing time for multisession iSAM for each window, distinguishing between windows with and without loop closures. (d) and (e) show the cumulative processing time for the individual modules of the frontend and backend, respectively.

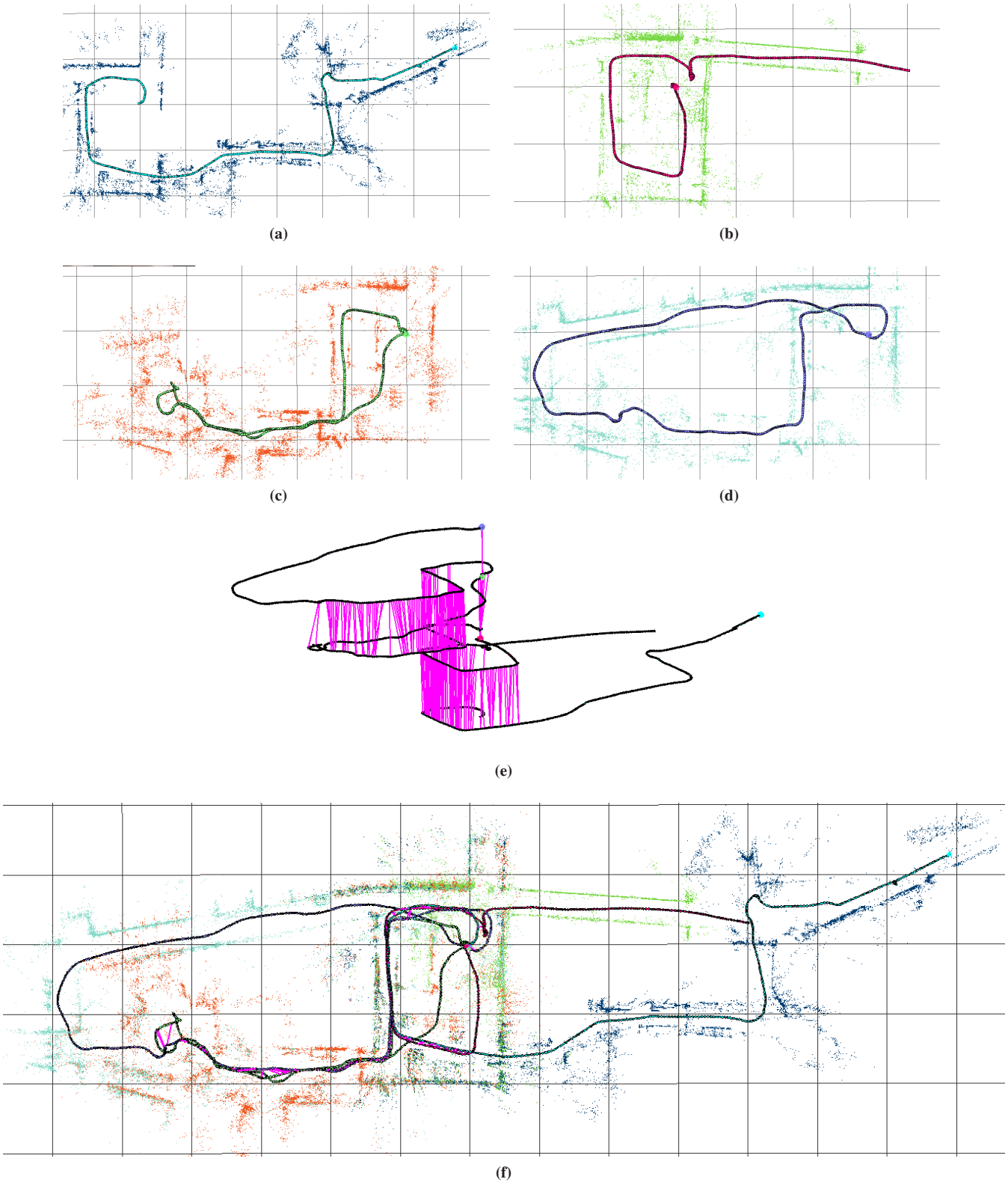
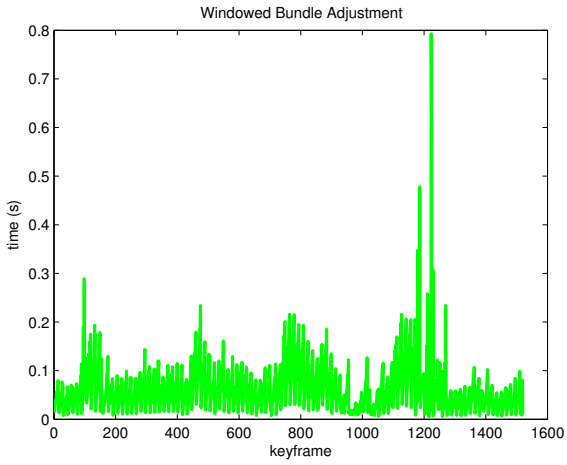
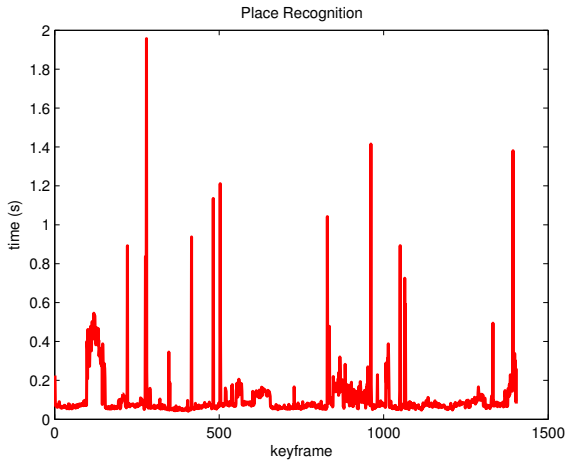


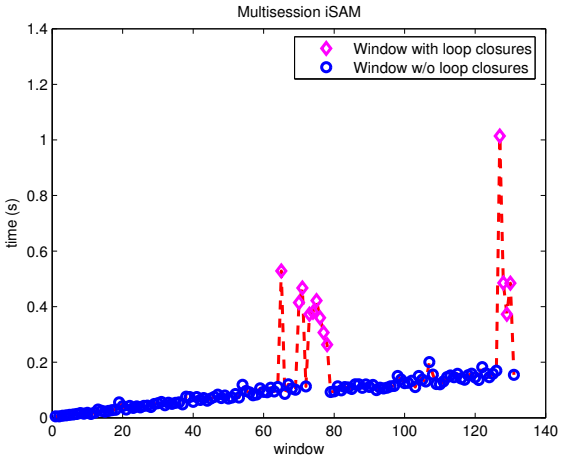
Figure 11: Stata Center second floor dataset with four separate sessions captured over a $75m \times 25m$ area. The underlying grid is set at a 5m scale in all figures. (a) – (d) show maps and pose graphs for each individual session. (e) shows the detected loops within and between pose graphs where the z-axis increase with time. (f) shows the combined multi-session map and pose graph. See Section 4.2 for further details. (In order to fully interpret this figure the reader is referred to the online colour version.)



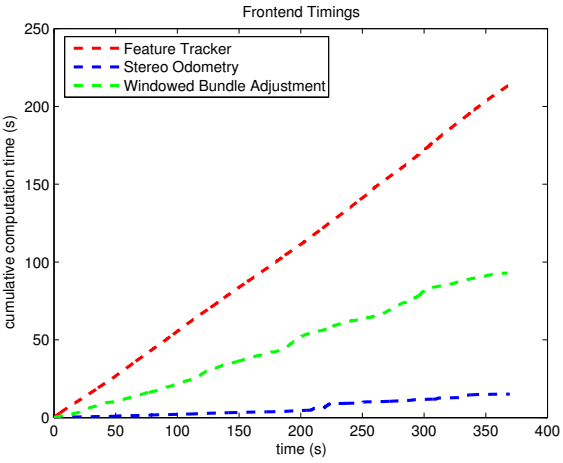
(a)



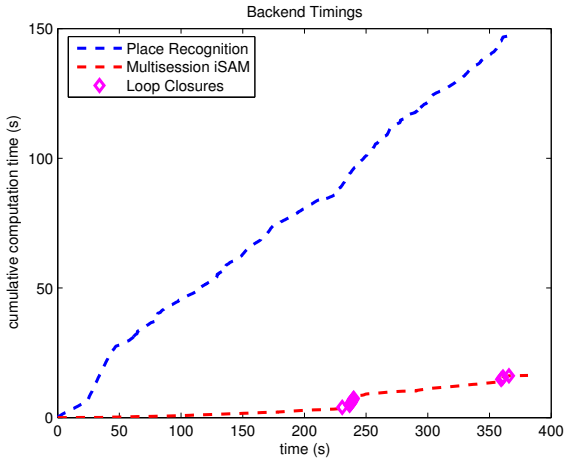
(b)



(c)

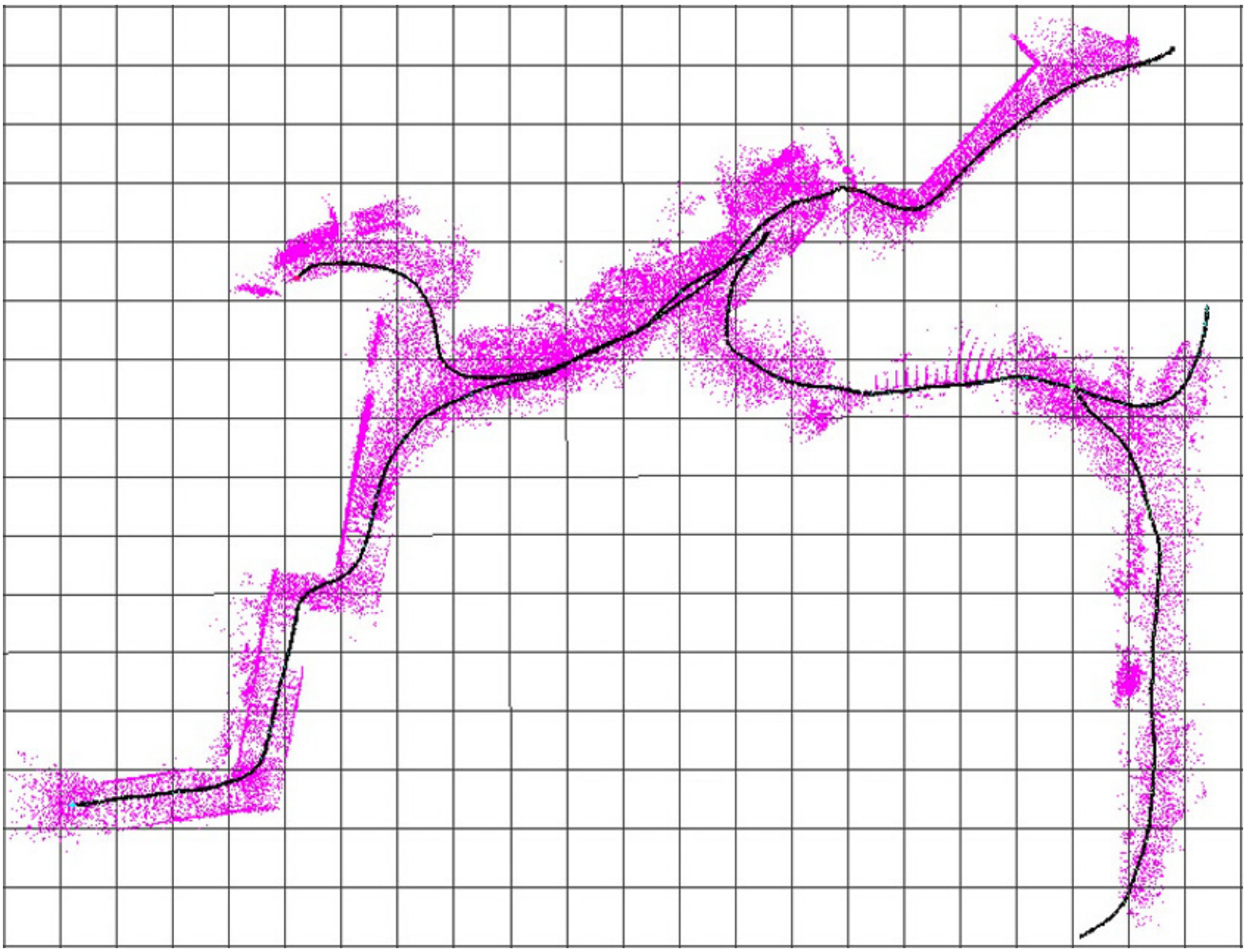


(d)

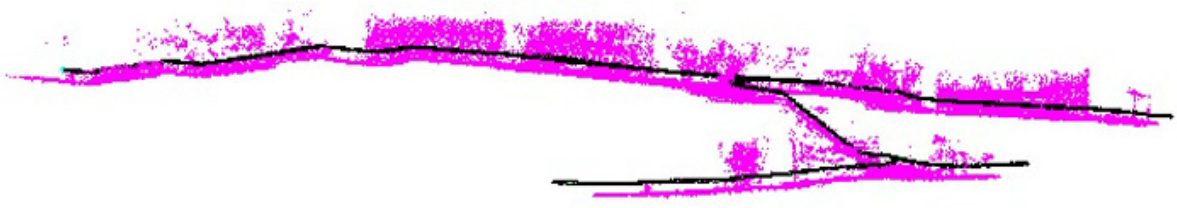


(e)

Figure 12: Timings for Experiment 3: multi-session outdoor handheld sequences. (a) and (b) show the processing times per keyframe for windowed bundle adjustment and place recognition, respectively. (c) shows the processing time for multisession iSAM for each window, distinguishing between windows with and without loop closures. (d) and (e) show the cumulative processing time for the individual modules of the frontend and backend, respectively



(a)



(b)

Figure 13: Multi-session results for outdoor datasets from Experiment 3 including 3 sessions. (a) shows a plan view where the grid is at a scale of 5m. (b) shows an orthographic side view where the full 3D structure is apparent, in particular the stairs. See Section 4.2 for further details.

[6] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR), Nara, Japan, 225–234, 2007.

[7] E. Eade, T. Drummond, Unified loop closing and recovery for real time monocular SLAM, in: British Machine Vision Conference, 2008.

[8] A. Davison, Real-time simultaneous localisation and mapping with a single camera, in: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, 1403–1410, 2003.

[9] H. Strasdat, J. Montiel, A. Davison, Real-time monocular SLAM: why filter?, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), 2010.

[10] R. Castle, G. Klein, D. Murray, Wide-area augmented reality using camera tracking and mapping in multiple regions, Computer Vision and Image Understanding 115 (6) (2011) 854 – 867, ISSN 1077-3142, doi:10.1016/j.cviu.2011.02.007, URL http://www.sciencedirect.com/science/article/pii/S1077314211000701.

[11] D. Nister, O. Naroditsky, J. Bergen, Visual odometry for ground vehicle applications, J. of Field Robotics 23 (1) (2006) 3–20, ISSN 1556-4967, doi:10.1002/rob.20103, URL http://dx.doi.org/10.1002/rob.20103.

[12] K. Konolige, J. Bowman, Towards lifelong visual maps, in: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 1156–1163, 2009.

[13] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, P. Fua, View-based maps, Intl. J. of Robotics Research 29 (8) (2010) 941–957.

[14] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395, ISSN 0001-0782, doi:http://doi.acm.org/10.1145/358669.358692.

[15] C. Zach, D. Gallup, J.-M. Frahm, Fast gain-adaptive KLT tracking on the GPU, in: Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on, 1 –7, doi:10.1109/CVPRW.2008.4563089, 2008.

[16] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment – a modern synthesis, in: W. Triggs, A. Zisserman, R. Szeliski (Eds.), Vision Algorithms: Theory and Practice, vol. 1883 of *LNCS*, Springer Verlag, 298–372, 2000.

[17] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, 2003.

[18] M. Kaess, H. Johannsson, J. Leonard, Incremental smoothing and mapping (iSAM) library, http://people.csail.mit.edu/kaess/isam, 2010–2011.

[19] D. Rosen, M. Kaess, J. Leonard, An incremental trust-region method for robust online sparse least-squares estimation, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), St. Paul, MN, 1262–1269, 2012.

[20] M. Kaess, A. Ranganathan, F. Dellaert, iSAM: incremental smoothing and mapping, IEEE Trans. Robotics 24 (6) (2008) 1365–1378.

[21] F. Lu, E. Milios, Globally consistent range scan alignment for environmental mapping, Autonomous Robots 4 (1997) 333–349.

[22] F. Grassia, Practical parameterization of rotations using the exponential map, J. Graph. Tools 3 (1998) 29–48, ISSN 1086-7651.

[23] B. Hall, Lie Groups, Lie Algebras, and Representations: An Elementary Introduction, Springer, 2000.

[24] C. Cadena, D. Gálvez, F. Ramos, J. Tardós, J. Neira, Robust place recognition with stereo cameras, in: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010.

[25] C. Cadena, J. McDonald, J. Leonard, J. Neira, Place recognition using near and far visual information, in: Proceedings of the 18th IFAC World Congress, 2011.

[26] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: Intl. Conf. on Computer Vision (ICCV), vol. 2, IEEE Computer Society, Los Alamitos, CA, USA, ISBN 0-7695-1950-4, 1470, doi:http://doi.ieeecomputersociety.org/10.1109/ICCV.2003.1238663, 2003.

[27] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, vol. 2, ISSN 1063-6919, 2161–2168, doi:10.1109/CVPR.2006.264, 2006.

[28] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 282–289, URL citeseer.ist.psu.edu/lafferty01conditional.html, 2001.

[29] F. Ramos, D. Fox, H. Durrant-Whyte, CRF-matching: conditional random fields for feature-based scan matching, in: Robotics: Science and Systems (RSS), 2007.

[30] F. Ramos, M. Kadous, D. Fox, Learning to associate image features with CRF-matching, in: Intl. Sym. on Experimental Robotics (ISER), 505–514, 2008.

[31] F. Dellaert, M. Kaess, Square Root SAM: Simultaneous localization and mapping via square root information smoothing, Intl. J. of Robotics Research 25 (12) (2006) 1181–1203.

[32] A. Huang, E. Olson, D. Moore, LCM: lightweight communications and marshalling, in: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010.