6-DOF Multi-session Visual SLAM using Anchor Nodes

John McDonald* Michael Kaess‡ Cesar Cadena† José Neira† John J. Leonard‡
*Department of Computer Science, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland
†Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain
‡Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA

***Abstract*— This paper describes a system for performing 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. We provide experimental results for multi-session visual mapping in the MIT Stata Center, demonstrating key capabilities that will serve as a foundation for future work in large-scale persistent visual mapping.**

***Index Terms*— multi-session visual SLAM, lifelong learning, persistent autonomy**

I. Introduction

Despite substantial recent progress in visual SLAM [17], 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 [3]."

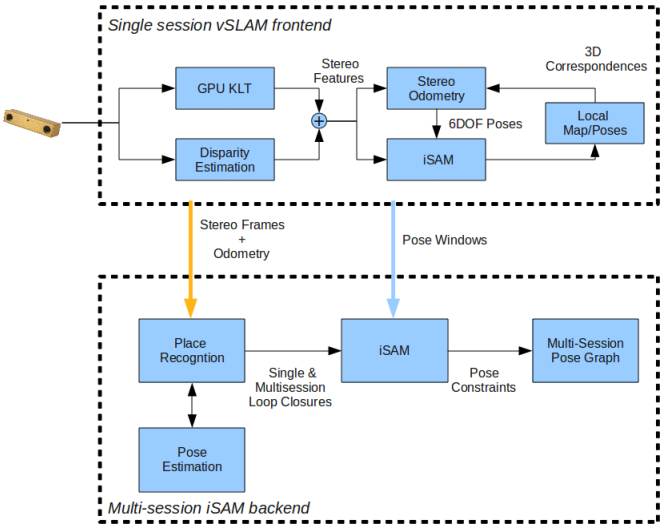


Fig. 1: Internal architecture of windowed and multi-session visual SLAM (vSLAM) processes.

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 [11]. 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 [18], in an approach called Tectonic Smoothing and Mapping (T-SAM). Our work extends the approach of Kim *et al.* [11] to perform multi-session visual mapping by incorporating a stereo odometry frontend in conjunction with a place-recognition system for identifying inter- and intra-session loop closures.

II. 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 [12], Eade and Drummond [5], and Davison *et al.* [4, 8]. 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).

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.* [19] 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 [14, 13]. Our research is closely related to this work, but has several differences. A crucial new aspect of our work in relation to [14] is the method we use for joining the pose graphs from different mapping sessions. Konolige and Bowman join pose graphs using "weak links", which are used to connect disjoint sequences. The weak links are added with a very high covariance and subsequently deleted after place recognition is used to join the pose graphs [14]. In our approach, which extends [11] to full 6-DOF, we use anchor nodes as an alternative to weak links; the use of anchor nodes provides a more efficient and consistent way to stitch together the multiple pose graphs resulting from multiple mapping sessions. In addition, our system has been applied to hybrid indoor/outdoor scenes, with hand-carried (full 6-DOF) camera motion.

III. 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 Figure 1. The system uses a sub-mapping 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 built up 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.

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.

IV. 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 inability to estimate scale and indirect depth estimation. The stereo odometry approach we use is similar to that presented by [19].

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 Harris 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 algorithm [6]. The input to the motion estimation algorithm consists of the set of tracked features 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 implementation of the above stereo odometry pipeline we use a GPU based KLT tracker [25]. 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 fifth frame, or when the number of feature tracks in the current frame drops below a certain threshold. A consequence of keeping the number of features in a given frame high, whilst at the same time setting a minimum inter-feature distance in the KLT tracker, is that it helps to ensure a good distribution of the resulting feature set over the image.

V. 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 structure within a local 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 parameterisation 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 point parameterisation $p = (x, y, z, w)$ allows dealing with points at infinity [24]. Points close to or at infinity cannot be represented correctly by the conventional Euclidean formulation. Even for points that are not at infinity, convergence of the smoothing optimisation is typically improved.

We use exponential maps based on Lie group theory to deal with overparameterised representations. In particular we use Quaternions to represent orientations in 3D space. Quaternions consist of four parameters to represent three degrees of freedom, therefore causing problems for conventional least-squares algorithms. Using an exponential map, as described for example in [7], reduces the local updates during optimisation to three parameters. The homogeneous point parameterisation suffers from the same problem, and indeed the same solution can be applied as for Quaternions after realising that both are equivalent to the 3-sphere S^3 in $\mathbb{R}^4$ if normalised.

With overparameterisations removed, the optimisation problem can now be solved with standard least-squares solvers. We use the iSAM library [9] to perform batch smoothing with Powell's Dog Leg algorithm. 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

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

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

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

When the smoothing window reaches a maximum size, 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 loop closures between arbitrary frames can be dealt with within the pose graph. Full details of the loop closure handling is provided in Section VII. 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 being optimised using the iSAM library [9], but this time using the actual incremental iSAM algorithm [10] 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

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

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 $x_d = x_a \ominus x_b$ from Lu and Milios [16] for representing pose x_a in the local frame of pose x_b ($x_a = x_b \oplus x_d$).

VI. 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 [1, 2], which demonstrated robust and reliable performance.

The place recognition module has the following two components:

- The first component is based on the bag-of-words method (BoW) [23] which is implemented in a hierarchical way [20]. 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 the score s. Then, there are three possibilities, if $s \geq \alpha^+ \lambda_t$ the match is considered highly reliable and accepted, if $\alpha^- \lambda_t < s < \alpha^+ \lambda_t$ the match is checked by conditional random field (CRF)-Matching in the next step, otherwise the match is ignored. In our implementation, λ_t is the BoW score computed between the current image and the previous one in the database. The minimum confidence expected for a loop closure candidate is $\alpha^- = 0.15$ and for a loop closure to be accepted is $\alpha^+ = 0.8$. The images from one session are added to the database at one frame per second and with the sensor in motion, i.e. during the last second, the sensor's motion according to the visual odometery module might be greater than 0.2m or 0.2rad.
- The second component consists of checking the previous candidates with CRF-Matching in 3D space. CRF-Matching is an algorithm based on Conditional Random Fields (CRF). Lafferty *et al.* [15] proposed CRF for matching 2D laser scans [21] and for matching image features [22]. CRF-Matching is a probabilistic model that is able to jointly reason about the association of features. In [1] CRF-Matching was extended to reason in 3D space about the association of data provided by a stereo camera system. We compute the negative log-likelihood $\Lambda_{t,t'}$ 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 $\Lambda_{t,t'} \leq \Lambda_{t,t-1}$.

This module exploits the efficiency of 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 considered for CRF-Matching.

VII. 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.* [11] and Ni and Dellaert [18].

In this work we distinguish between intra-session and inter-session loop closures. Processing of loop closures is performed firstly with each candidate frame 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 IV. Here the disparities for one of the feature sets are used to perform 3D reconstruction of their preimage points. 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 reprojection 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_j^s and $x_{j'}^{s'}$. 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 and the subsequent local smoothing. A second type of measurement is provided by intra-session loop closures.

The use of anchor nodes [11] 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, 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 also involves 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

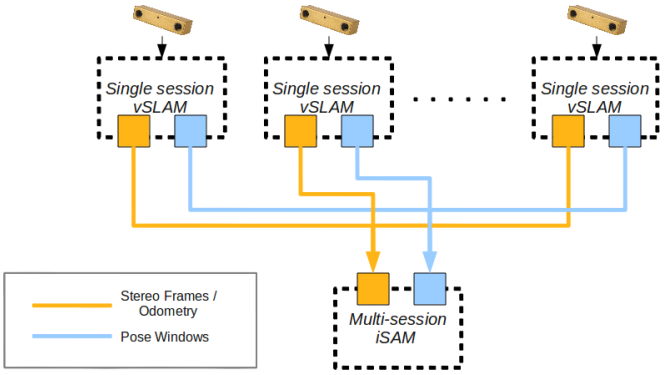


Fig. 2: Multi-session visual SLAM processing

an encounter c_i is given by

$$h(x_j^s, x_{j'}^{s'}, \Delta^s, \Delta^{s'}) \propto \exp\left(-\frac{1}{2}\left\|((\Delta^s \oplus x_j^s) \ominus (\Delta^{s'} \oplus x_{j'}^{s'})) - c\right\|_\Gamma^2\right) \tag{4}$$

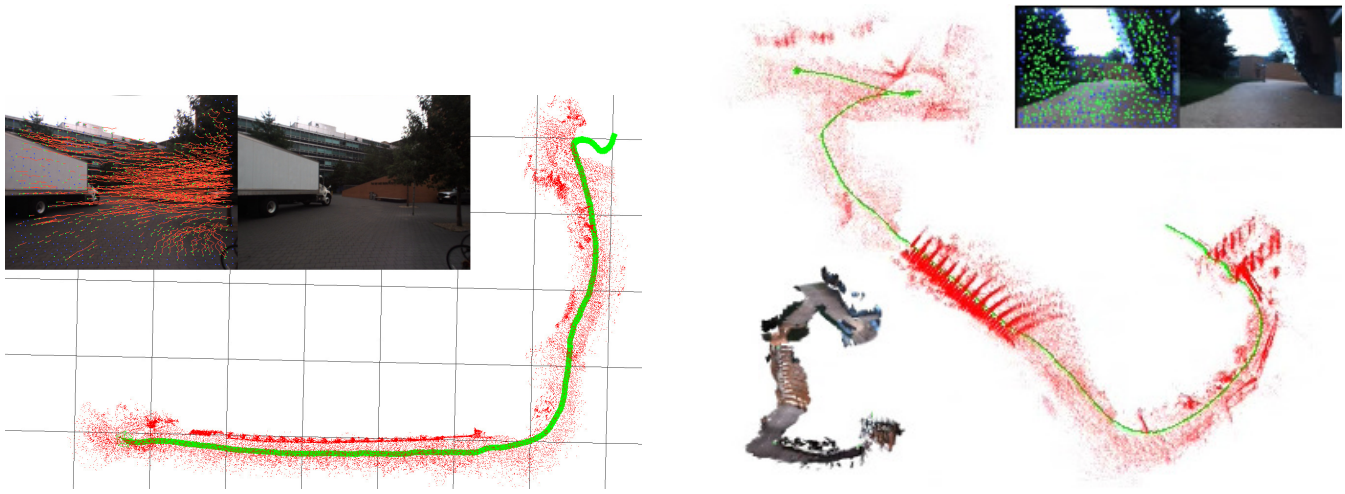
where the index i was dropped for simplicity. The concept of relative pose graphs generalises well to a larger number of robot trajectories. The number of anchor nodes depends only on the number of robot trajectories.

VIII. Experiments and Results

In this section we present results of the performance of our system for both single- and multi-session processing. The dataset that we use was collected at the Ray and Maria Stata Center at MIT over a period of months. This building in known for its irregular architecture and provides a good testing ground for visual SLAM techniques in general.

The dataset includes indoor and outdoor (and mixed) sequences captured from both a wheeled platform and using a handheld camera with full 6-DOF movement (e.g. ascending and descending stairs, etc.). All images 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 wheeled platform also included a horizontally mounted 2D SICK laser scanner and a spinning LiDAR. 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 3D scenarios (i.e. 2D + rotational movement).

The complete multi-session visual SLAM system follows the architecture shown in Fig. 1, and is implemented as a set of loosely coupled processes that communicate via the *Lightweight Communications and Marshalling* (LCM) robot middleware system. 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). Futhermore the overall architecture can be transparently reconfigured for different setups (e.g. from single CPU to multi-core or distributed processing).



(a) (b)

Fig. 3: Single session visual SLAM processing including full 6-DOF motion.

A. Single-Session Visual SLAM Results

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

Two example feature-based maps from outdoor sequences are shown in Fig. 3. 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. 3 (b) shows a similar scale sequence that includes full 6-DOF motion, where the user has carried a handheld camera up a stairs.

In the absence of loop closing 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. 4 shows two maps with two trajectories, both taken from the same sequence. The yellow contour shows a 2D LiDAR based map computed from applying a scanmatching algorithm to the output of horizontal LiDAR scanner attached to the cart. 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 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. 5. Here the dataset consists of an image sequence taken over an indoor area within in the Stata Center. Here 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 here is due to the tight turn at the right-hand end of the sequence, where the divergence between each traversal of the hallway can be clearly seen.

The center figure shows the result of the correction applied to the pose graph due to a sequence of loop closures occuring 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 accuracy of the map relative to the ground truth CAD floorplan.

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 in low-texture or low contrast environments, or where the disparity estimation fails over a large set of features, e.g. due to aliasing. We do not address this situation in the current system, however the standard approach of incorporating inertial sensors is a natural solution to this problem. An alternative approach that we are currently investigating is the possibility of using multi-session SLAM as a solution to this problem, 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. A future paper will present results of this approach.

B. Multi-Session Visual SLAM Results

To test the full multi-session visual SLAM system, we took two sequences from the same area as shown in Fig. 5 and processed each through a separate instance of the visual SLAM frontend. Results of each of the separate sessions are shown in Fig. 6 (a) and 6 (b), with the combined multi-session results shown in Fig. 6 (c). Again, loop closure occurred in the same area as shown in Fig. 5 (b). Finally Fig. 6 (d) shows a textured version of the same map. The scale of the grid is 2m for Figures (a) & (b), and 5m for Figures (c) & (d).

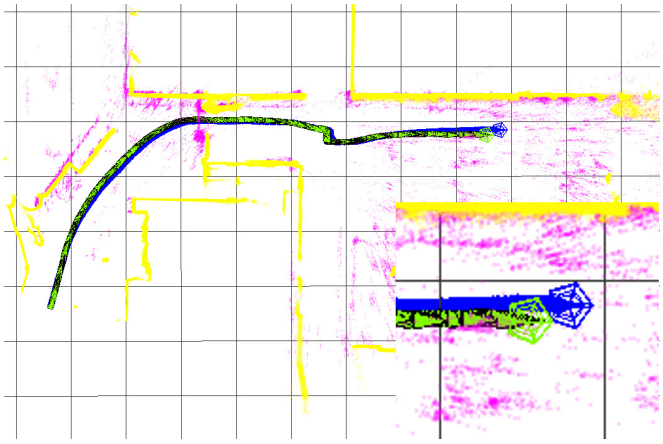


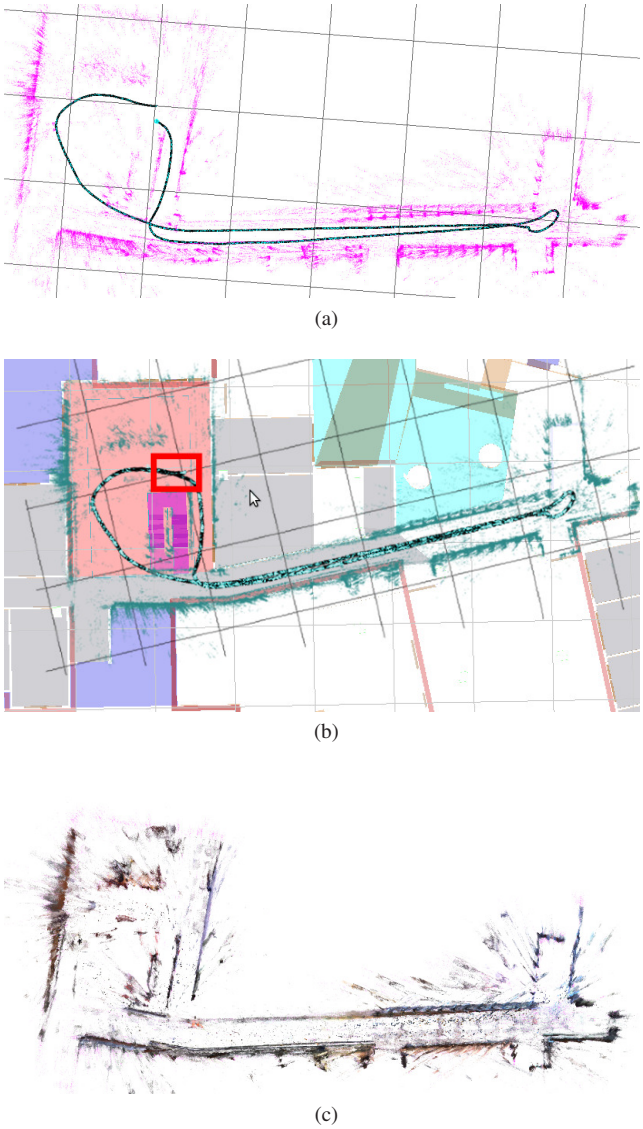
Fig. 4: Comparison of drift in single session visual SLAM against 2D LiDAR scan matcher over a 20m trajectory. Grid scale is 2m.

IX. Conclusions

In this paper we have presented a 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 [11]. 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 was used in [11]).

We have demonstrated this system in both indoor and outdoor environments, and have provided examples of single- and multi-session pose graph optimisation and map construction. We have also shown the effects of loop closures within single-session mapping in reducing drift and correcting map structure.

Multi-session visual mapping provides 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 VIII. Furthermore we intend to evaluate the approach in online collaborative mapping scenarios over extended timescales.



(a)

(b)

(c)

Fig. 5: 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).

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, 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.

References

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

[2] C. Cadena, J. McDonald, J. Leonard, and J. Neira. Place recognition using near and far visual information. In *Proceedings of the 18th IFAC World Congress*, August 2011. To appear.

[3] M. Cummins. *Probabilistic Localization and Mapping in Appearance Space*. PhD thesis, University of Oxford, 2009.

[4] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410, 2003.

[5] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular SLAM. In *British Machine Vision Conference*, 2008.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[7] F.S. Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3:29–48, Mar 1998.

[8] J.M.M. Montiel H. Strasdat and A.J. Davison. Real-time monocular SLAM: Why filter? In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2010.

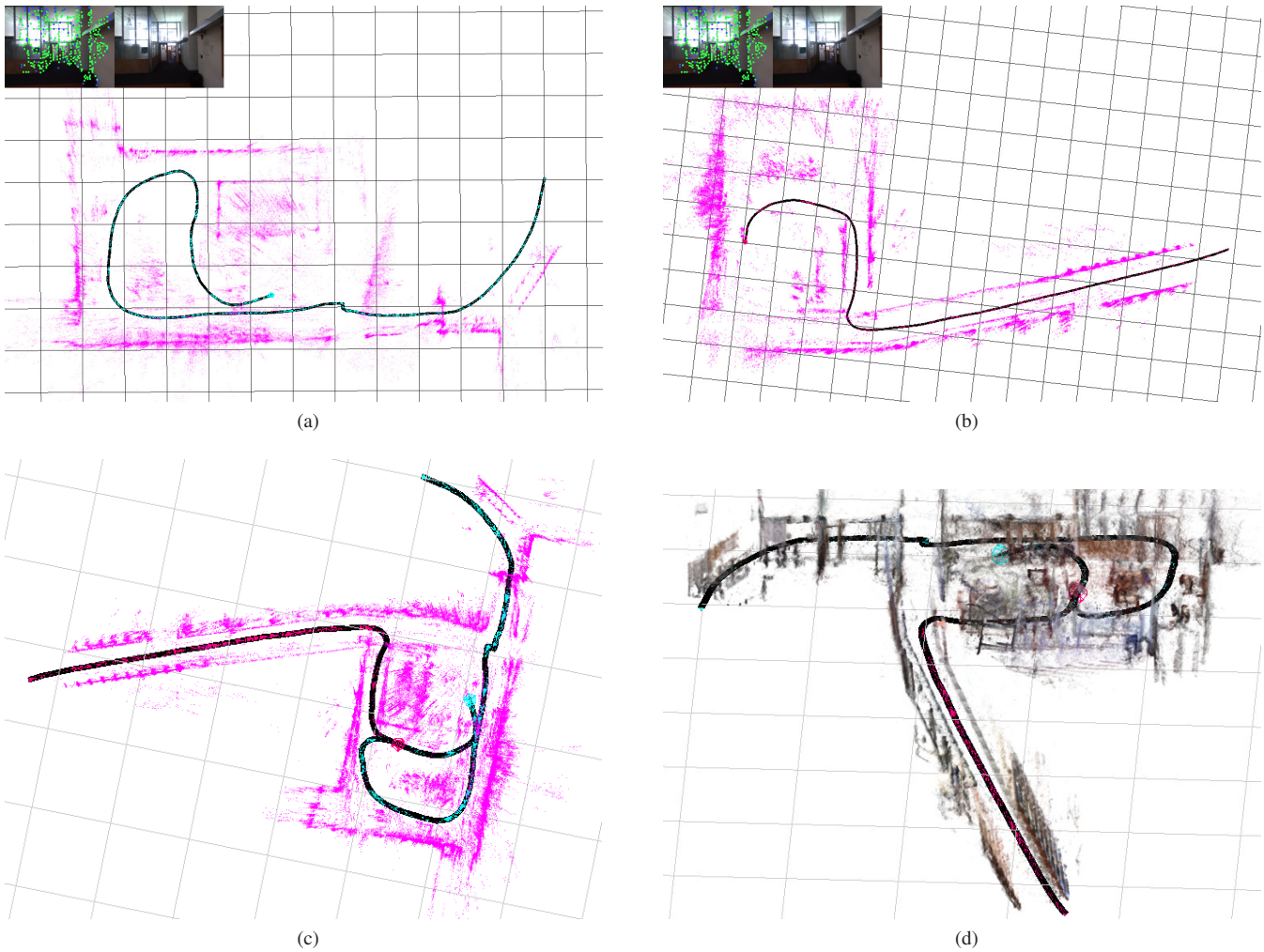


Fig. 6: Stata Center second floor dataset with two separate sessions captured over an $850m^2$ area. (a) Map and poses for session 1. (b) Map and poses for session 2. (c) Multi-session pose graphs after inter-session loop closure showing transformed maps. (d) Textured version of figure (c).

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

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

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

[12] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE Computer Society, 2007.

[13] K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *Intl. J. of Robotics Research*, 29(10), 2010.

[14] K. Konolige and J. Bowmand. Towards lifelong visual maps. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1156–1163, 2009.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[16] F. Lu and E. Milios. Globally consistent range scan alignment for environmental mapping. *Autonomous Robots*, 4:333–349, April 1997.

[17] J. Neira, A.J. Davison, and J.J. Leonard. Guest editorial special issue on visual SLAM. *IEEE Trans. Robotics*, 24(5):929–931, Oct 2008.

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

[19] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *J. of Field Robotics*, 23(1):3–20, 2006.

[20] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 2161 – 2168, 2006.

[21] F. Ramos, D. Fox, and H. Durrant-Whyte. CRF-Matching: Conditional Random Fields for feature-based scan matching. In *Robotics: Science and Systems (RSS)*, 2007.

[22] F. Ramos, M.W. Kadous, and D. Fox. Learning to associate image features with CRF-Matching. In *Intl. Sym. on Experimental Robotics (ISER)*, pages 505–514, 2008.

[23] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Intl. Conf. on Computer Vision (ICCV)*, volume 2, page 1470, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[24] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, Sep 1999.

[25] C. Zach, D. Gallup, and 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*, pages 1 –7, June 2008.