A Constant-Time Efficient Stereo SLAM System

Christopher Mei[1]
Gabe Sibley[2]
Mark Cummins[2]
Paul Newman
Ian Reid
{cmei,gsibley,mjc,pnewman,ian}@robots.ox.ac.uk

Department of Engineering Science
University of Oxford
Oxford
OX1 3PJ
UK

Abstract

Continuous, real-time mapping of an environment using a camera requires a constant-time estimation engine. This rules out optimal global solving such as bundle adjustment. In this article, we investigate the precision that can be achieved with only local estimation of motion and structure provided by a stereo pair. We introduce a simple but novel representation of the environment in terms of a sequence of relative locations. We demonstrate precise local mapping and easy navigation using the relative map, and importantly show that this can be done without requiring a global minimisation after loop closure. We discuss some of the issues that arise from using a relative representation, and evaluate our system on long sequences processed at a constant 30-45 Hz, obtaining precisions down to a few metres over distances of a few kilometres.

1 Introduction

The goal of building an autonomous platform using vision sensors has encouraged many developments in low-level image processing and in estimation techniques. Recent improvements have led to real-time solutions on standard hardware. However these often rely on global solutions that do not scale with the size of environment. Furthermore, few systems integrate a loop closure or a relocalisation mechanism that is essential for working in non-controlled environments where tracking assumptions are often violated. In this work, we demonstrate the integration of three key components: (i) a representation of the global environment in terms of a *continuous* sequence of relative locations; (ii) a visual processing front-end that tracks features with sub-pixel accuracy, computes temporal and spatial correspondences, and estimates precise local structure from these features; (iii) a method for loop-closure which is independent of the map geometry, and solves the loop closure and kidnapped robot problem; to produce a *system* capable of mapping long sequences over large distances with high precision, but in constant time.

The contribution of the article comes from the integration of recent advances with new visual processing components to achieve the robustness, precision and speed we aim for

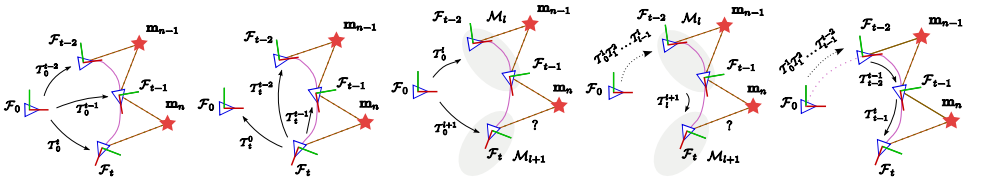
1. Work supported by EPSRC grants GR/T24685/01 and EP/D037077.
2. Work supported by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence, Guidance Ltd, and UK EPSRC (CNA and Platform Grant EP/D037077/1).

through careful engineering of the system. The environment is represented by a set of continuous relative frames and estimation is done on a frame-to-frame basis. Details of relative bundle adjustment (RBA) that leads to improved precision in this framework can be found in [21]. We also introduce the idea of using the "true scale" of features (available because we have depth from binocular stereo) to provide scale invariance, and a method for automatically achieving a good distribution of features across the image in order to avoid degenerate structure. We demonstrate the efficacy of our careful engineering with reference to a number of challenging sequences, showing performance at and beyond the current state-of-the-art. In particular we show our system mapping long loops in outdoor environments, in spite of difficult imaging conditions, achieving precision of a few metres over a number of kilometres travelled.

SLAM, and visual SLAM in particular, has been a very active field in recent years. A number of approaches have been proposed for estimating the motion of a single camera and the structure of the scene, for example [6], which uses an Extended Kalman filter (EKF). The EKF – at least in the standard form – gives rise to a fundamental limitation in the number of features that can be mapped, because of the quadratic complexity of the EKF algorithm. Recent real-time monocular systems have used local [17] or global [7, 8, 12] bundle adjustment as the underlying estimator. This choice is motivated by a complexity linear in the number of landmarks (but cubic in the number of poses) enabling the processing of more landmarks than an EKF leading to an overall better precision. A careful choice of key frames is however required to keep the solving tractable. [17] does not provide a loop closing mechanism and cannot as such reduce drift when returning to a previously explored region. [12] uses a separate thread for bundle adjustment enabling the building of accurate maps for small environments but is not adapted to large scale exploration. [8] shares similarities with the present work, combining a relative representation with a non-probabilistic loop closure mechanism. In our work however, the image processing is adapted to stereo vision and the focus is on precise exploration *without* requiring global graph minimisation.

Recent research has also provided some real-time solutions using stereo pairs [13, 19, 25]. [19] relies on local bundle adjustment but does not address the problem of loop closing. [25] relies on two stereo pairs and an IMU. The authors provide a loop closing mechanism but do not address the problem of obtaining locally or globally optimal maps. The closest related work is that of Konolige and Agrawal [13]. The difference with the presented work is the focus on recovering globally metrically coherent maps thus rendering the complexity of the system linear in the number of poses. However for many applications (path planning, object manipulation, dynamic object detection, ...) a relative local map is sufficient and can be recovered in constant time (the complexity being related to the size of the working space). It is also important to note that global metric maps and relative maps are not equivalent (see Section 2.2) and share different properties. As will be discussed, in certain cases, imposing global constraints can be detrimental.

The remainder of the paper is structured as follows: Section 2 presents the continuous relative representation and map management, Section 3 details the steps to efficiently and robustly track the features and build the map, Section 4 describes the relocalisation and loop closing mechanism that relies on the "true scale" descriptors and finally experimental validation of the system is described in Section 5.



(a) Global (b) Robo-centric (c) Global Sub-mapping (d) Relative Sub-mapping (e) Continuous relative representation (CRR)

Figure 1: Different pose and landmark representations with **m** for landmarks, **F** for frames, **T** for transforms and **M** for sub-maps where applicable. Dashed lines represent measurements. Landmarks are connected to their base frames by filled lines. '?' indicates the difficulty in sharing information between sub-maps.

2 Map management

2.1 World representations

The position of the robot in the world can be represented in different ways (Fig. 1):

Global coordinates. (Fig. 1(a)) This is the most common representation. An arbitrary initial frame (usually set to be the identity transform) is chosen and all subsequent position and landmark estimates are represented with respect to this frame.

Robo-centric coordinates. (Fig. 1(b)) This is similar to using global coordinates but the initial frame is chosen to be the current robot position. The map has to be updated at each new position estimate. This representation has been shown to improve consistency for EKF SLAM estimation [3].

Relative representation. [2, 8, 10, 13]. (Fig. 1(d),1(e)) In this framework, each camera position is connected by an edge transform to another position forming a graph structure. There is no privileged position and recovering landmark estimates requires a graph traversal (eg breadth first search or shortest path computation).

Sub-maps. (Fig. 1(c), 1(d)) Sub-maps consist in representing a map by local frames and can be used with any of the previously discussed map representations. There are mainly two reasons for using sub-maps: reducing computation and improved consistency (mainly in filtering frameworks to reduce the effect of propagating inconsistent statistics).

In this work, the robot position and map are represented in a continuous relative framework (CRR) (Fig. 1(e)). This approach is beneficial for two reasons. First, it allows constant time state-updates even when loop-closures are detected and relative bundle adjustment (RBA) is applied [21]. Second, optimisation using CRR effectively handles problems inherent in sub-mapping, such as map merging and splitting, data duplication and inconsistency.

2.2 Continuous Relative Representation (CRR)

A continuous relative representation (CRR) was chosen to represent the world as described in Fig. 2. The continuous line between poses indicates estimated transforms obtained during the exploration. We define an active region to be the set of poses within a given distance in the graph to the current pose. In the example, an active region of size two was chosen.

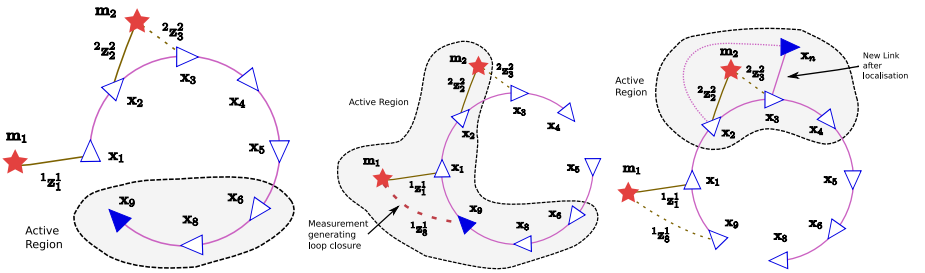


Figure 2: Relative representation. Triangles represent robot poses ($\mathbf{x}_i$) with the current pose filled in. Stars represent landmarks ($\mathbf{m}_i$). ${}^k\mathbf{z}_i^j$ indicates a measurement of landmark j from frame i with base frame k. (a) Graph representing a robot trajectory. The active region of size two contains the latest two poses. (b) Trajectory after loop closure. The robot in $\mathbf{x}_9$ makes an observation of landmark $\mathbf{m}_1$. This provides an estimate between poses $\mathbf{x}_1$ and $\mathbf{x}_9$ represented by the new link in the graph. The active region of size two, discovered by breadth first search in the graph, now comprises the older poses $\mathbf{x}_1$ and $\mathbf{x}_2$ because of the added link. The link between $\mathbf{x}_4$ and $\mathbf{x}_5$ still exists but is no longer represented as we do not enforce the composition of the transforms along cycles to be the identity. (c) Trajectory after localisation. The robot was previously connected to $\mathbf{x}_2$ but the latest estimate of its position to the poses in the active region has shown a closer proximity to $\mathbf{x}_3$. In this case, the old link is discarded and a new link is created, here between $\mathbf{x}_n$ and $\mathbf{x}_3$.

Generally, the active region will be comprised of the latest poses. However, in the case of a loop closure, as illustrated by Fig. 2(b), older poses will also form the active region.

The active region provides a way to find the landmarks visible from the current frame. A landmark *base frame* is defined as the frame in which the landmark's 3-D coordinates are kept (in this work we used the frame where the the first landmark observation was made). Landmarks with base frames belonging to poses from the active region are projected into the current frame by composing the transforms along the edges. These estimates are then used to establish matches and compute the position of the robot. In the current setting we do not use the uncertainty provided by the estimates for the data association but use a fixed-size window. After a loop closure, the newly created edge can be used to transform 3-D points into the current frame and therefore to find their projections. This enables data association *without* requiring a global minimisation, in contrast with a global filtering framework in which a state update would be required after loop closure to provide precise estimates. In this work, the discovery of the active region is made by a breadth first search (BFS) in the graph built during the exploration.

It should be noted that the global solution minimising the reprojection error in a relative framework is not equivalent to the global solution using a global reference with the same measurements as discussed in greater detail in [21]. In the case of unobservable ego-motion (e.g. if the robot uses a means of transport such as a lift or car), the map cannot be represented in a Euclidean setting. Methods that use a global frame would fail but a relative representation still holds. This would also be the case with sub-mapping if the boundaries overlap such regions. This observation leads to a change in perspective where the importance of detecting changes in the environment becomes apparent, global constraints (such as those imposed in a pose relaxation frameworks) should not be applied blindly.

3 Stereo visual processing

Attention to the detail of the visual processing is important in achieving precision, robustness and speed. This section describes the steps performed for each incoming frame pair.

A. Pre-processing and feature extraction

Each incoming image is rectified using the known camera calibration parameters to ensure efficient scanline searches of corresponding left-right matches [11]. The image intensities for left and right images are then corrected to obtain the same mean and variance. This step improves the left-right matching scores, enabling better detection of outliers.

The feature locations used in this work are provided by the FAST corner extractor [20] that provides good repeatability at a small computational cost. To obtain resilience to blur and enable matching over larger regions in the image, FAST corners are extracted at different levels of a scale-space pyramid computed with one image per octave for computational efficiency. In practise we used three pyramid levels in processing indoor sequences, but found that two levels were sufficient for our outdoor sequences; the type of camera, and the amount of motion blur expected are factors which influence this parameter.

The corner extraction threshold is initially set at a value providing a compromise between number of points and robustness to noise. This threshold is then decreased or increased at each time-step to ensure a minimal number of points. This proved sufficient to adapt to the strong changes in illumination typically encountered in outdoor sequences.

B. Pose initialisation with image-based gradient descent

The apparent image motion of features in visual SLAM is typically dominated by the ego-rotation, and large inter-frame rotation is a common failure mode for systems based on an inter-frame feature search. To improve robustness, an estimate of the 3-D rotation is obtained using the algorithm described in [16]. This algorithm minimises the sum-of-squared-distance of image intensity using a second-order gradient descent minimisation (ESM). This estimate is then used to guide the search for temporal feature correspondences.

C. Temporal feature matching and pose estimation

The current 6-DOF pose is initialised to the rotation computed in the previous stage, and a translation of zero relative to the previous pose. The 3-D coordinates of the landmarks (i.e. the map) computed through the graph representing the relative poses (as detailed in Section 2.2) are then projected into the left and right images of the current stereo pair and matched in a fixed-sized window to the extracted FAST corners using mean SAD (sum of absolute difference with the mean removed for better resilience to lighting changes) on image patches of size 9×9. This step is then followed by image sub-pixel refinement using ESM. Matches whose score fall below a threshold after ESM are rejected.

Having established 3-D to 2-D matches, standard robust methods are used [9, 11, 19] to localise the current camera pose, minimising the total reprojection error over both views.

D. Initialising new features

To achieve good accuracy we aim to measure between 100-150 features, with a fairly even spatial distribution across the image, at every time-step. Thus after temporal matches have been established we seek to initialise new features in the current frame. In the left view, FAST corners are ranked by a distinctiveness score (we use the Harris corner score). However rather than simply take the highest ranking features, we also try to ensure a good spread of features across the images to improve conditioning.

To this end we maintain a quadtree representation of the left-image domain in which each cell in the quadtree contains the number of 3-D features that currently project to that

cell (these are done during temporal matching). Potential new features, in order of their distinctiveness scores, are tested against the quadtree to ensure that the number in each cell does not exceed a pre-set maximum percentage. On selecting a potential feature from the left image, its correspondence in the right image is sought by scanline search in our rectified images. The left-right match is refined to sub-pixel accuracy via ESM minimisation [1] between 9x9 image patches. These left-right sub-pixel coordinates are then used to initialise the 3-D landmark estimate by triangulation and the current frame becomes the base frame for this landmark. In addition, for each new 3-D landmark, we compute a SIFT descriptor used for relocalisation and loop closure, with efficient computation of the scale (Section 4).

4 Relocalisation and loop closure

The method described above is generally successful in the normal course of operation. Nevertheless failures are almost inevitable, and there is a need for more resilient methods that enable the system to recover (i.e. relocalise) after such failures, as well as to be able to detect a return to previously mapped areas (i.e. loop closure). For both processes we make use of the SIFT descriptors computed on feature initialisation.

4.1 True scale

The well-known SIFT [15] descriptor is built at a scale determined by finding an extremum in scale in a Difference of Gaussians (DoG) pyramid. The most expensive part of the algorithm is generally the computation of the image scale space. Previous work [4] has also investigated solutions to avoid this cost in the case of monocular SLAM using the estimated camera pose. An alternative is proposed here for stereo pairs that avoids the knowledge of camera position. Landmark descriptors are built corresponding to regions in the world of same physical size - we call this the "true scale" of the feature. This provides the property of matching only regions of same 3-D size which is not necessarily the case with DoG features. It can be achieved at no extra computational cost as the left-right matching that provides the 3-D location is required in any case for the motion estimation. True scale requires choosing a set of 3-D sizes for different depth ranges called "rings" to ensure the projection size of the 3-D template lies within a given pixel size range (Fig. 3).

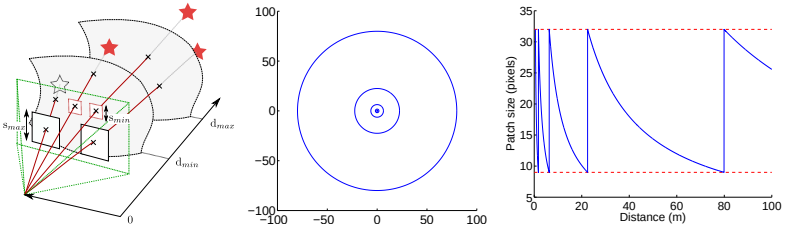
We adopt a pinhole camera model with a focal length f. The size in the image of the projection of a 3-D region of space is inversely proportional to its distance to the centre of projection. Let s_{max} and s_{min} be the maximal and minimal size of square templates that could be matched reliably. s_{max} corresponds to a certain landmark distance d_{min}. The maximal acceptable distance corresponding to s_{min} (Fig. 3(a)) is then:

$$d_{max} = \frac{s_{max}}{f},\ d_{min} = \frac{s_{min}}{f} \ \Rightarrow\ d_{max} = \frac{s_{max}}{s_{min}} d_{min}$$

For typical values: d_{min} =0.5 m, $s_{max} = 32$ pixels and $s_{min} = 9$ pixels, $d_{max} \approx$1.8 m which is insufficient to cover the full depth range. A solution is to use different 3-D sizes in "rings" to cover the depth range. Figure 3(b) shows the different distance discontinuities in 2-D corresponding to s_{min} and s_{max} with the camera at the centre. Figure 3(c) illustrates the image template size according to depth. When relocalising or computing loop closures, only features belonging to the same ring are matched.

Let d_n be the smallest distance for the ring n, using geometric series:

$$d_1 = d_{min},\ d_{n+1} = \left(\frac{s_{max}}{s_{min}}\right) d_n \Leftrightarrow d_{n+1} = \left(\frac{s_{max}}{s_{min}}\right)^n d_{min},\ n \geq 0$$



(a) (b) (c)

Figure 3: True scale distance rings. (a) A fixed 3-D region size between distances d_{min} and d_{max} projects to template sizes ranging from s_{max} to s_{min} with size in $\frac{1}{d}$. For $d > d_{max}$, a bigger fixed 3-D size can be computed to provide image templates within the same pixel size range. (b) This figure illustrates the different bands for true scale computation that become bigger with distance. (c) Patch sizes according to distance.

The template size as a function of the landmark distance, $s(d)$ can thus be computed from:

$$n = \lfloor \frac{\log(d/d_{min})}{\log(s_{max}/s_{min})} \rfloor + 1, \quad s(d) = s_{max}\left(\frac{d_n}{d}\right) = s_{max}\left(\frac{s_{max}}{s_{min}}\right)^{n-1}\frac{d_{min}}{d}$$

A SIFT descriptor is built for each a landmark once the scale has been estimated.[1]

4.2 Relocalisation

The system uses a standard relocalisation mechanism when data association fails between consecutive frames. The "true scale" SIFT descriptors (Section 4.1) between the current and the previous frames are matched by direct comparison. 3-point-pose RANSAC [9] is then applied to robustly find the pose of the platform. If this step fails, loop closure (Section 4.3) is attempted on subsequent frames. This approach takes advantage of the stereo setting and avoids the training and memory requirement of methods such as randomised trees [14, 24].

4.3 Loop closure

For loop closure we rely on fast appearance based mapping [5][2]. This approach represents each place using the bag-of-words model developed for image retrieval systems in the computer vision community [18, 23]. At time k the appearance map consists of a set of n_k discrete locations, each location being described by a distribution over which appearance words are likely to be observed there. Incoming sensory data is converted into a bag-of-words representation; for each location, a query is made that returns how likely it is that the observation came from that location's distribution or from a new place. This allows us to determine if we are revisiting previously visited locations.

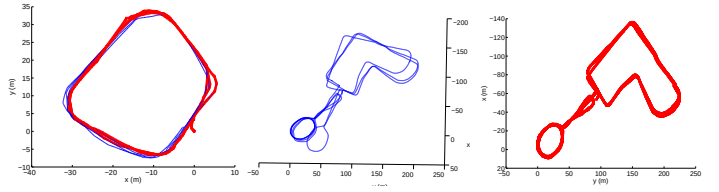
In a filtering framework, incorrect loop closures are often catastrophic as the statistical estimates are corrupted. The CRR enables recovery from erroneous loop closures as removing the incorrect graph link and bad measurements returns the system to its previous state.

4.4 Key frames and localisation

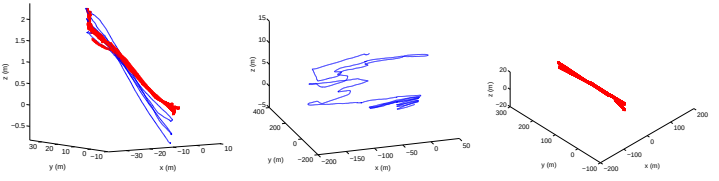
When exploring environments over long periods of time, memory usage becomes an issue. The presented system uses a simple heuristic similar to that presented in [17] to decide what frames to keep. Furthermore, when exploring a previously mapped area detected by loop closure, the system localises with respect to the active region thus reducing the amount of new

[1] The SIFT implementation is based on the code provided by A. Vedaldi. www.vlfeat.org/~vedaldi/

[2] A version of FABMAP is available at www.robots.ox.ac.uk/~mobile



(a) Begbroke top view (b) New College without loop closure (c) New College top view with loop closure



(d) Begbrokde side view (e) New College side view without loop closure (f) New College side view with loop closure

Figure 4: Estimated trajectories for the data sets detailed in Tab. 1. The trajectories are shown processed with loop closure (red trajectory with crosses) and without (blue continuous line).

	Begbroke	New College
Distance Travelled	1.08 km	2.26 km
Frames Processed	23K	51K
Reprojection Error Min/Avg/Max	0.003 / 0.17 / 0.55 pixels	0.03 / 0.13 / 1.01 pixels
Accuracy without loop closure	∼1m in (x-y) plane, ∼1m in z	∼15-25m in (x-y) plane, ∼15m in z
Accuracy with loop closure	∼1cm in (x-y) plane, ∼1cm in z	∼10cm in (x-y) plane, ∼10cm in z

Table 1: Results for the Begbroke and New College data sets.

features created and simultaneously reducing drift (Fig. 2(c) illustrates this mechanism). The metric for deciding when to connect poses is based on the distance (typically 1 m) and angle (10 deg) between frames and a minimum number of tracked landmarks (typically 50%).

5 Experimental results

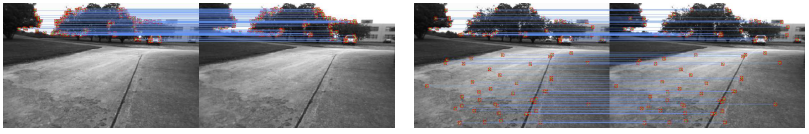
The full system was tested on numerous sequences of up to 2km. A total of 5km was traversed and over 300K images processed. The system exhibits robustness to motion blur, fog, lighting changes, lens flare and dynamic objects. In this work, the FABMAP vocabulary had 10000 words generated from a separate sequence of 11200 frames.

The performance on two specific sequences are detailed in Tab. 1 and the estimated trajectory can be found in Fig. 4[3]. No ground truth is available for these two sequences but the robot was driven so that its trajectory would overlap. The accuracy reported in the table was measured in the x-y plane and along the z-axis with and without loop closure. Even without a global relaxation, the loop closure greatly improves the accuracy.

The average computation time on an Intel 2.40GHz Quad CPU with only one core running totals 27.5ms (∼36Hz). A typical breakdown is: Pre-processing 10ms; Tracking 5ms; RANSAC 1.5ms; Localisation 4ms; Left-right matching 2ms; SIFT descriptors 5ms.

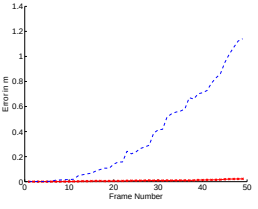
We found that the following components contribute to the overall performance:

[3]The New College dataset set is available online at www.robots.ox.ac.uk/NewCollegeData/ [].

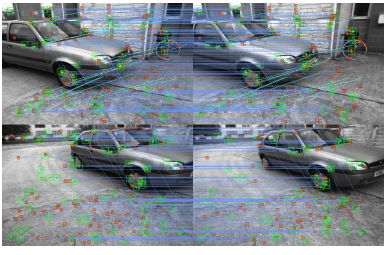


(a) (b)

Figure 5: Quadtrees provide a way to spread features in the image. (a) Taking the strongest Harris scores might lead to poorly constrained estimates. (b) Using a quadtree, it is possible to obtain points that do not necessary have high Harris scores but provide strong constraints.



(a) Sub-pixel refinement. The blue dashed line and the red solid line with crosses show respectively the translation error with and without sub-pixel refinement over a 10 m simulated sequence.



(b) This figure shows a successful relocalisation. FAST features are matched using SIFT descriptors with scale invariance provided by the left-right matching.

Figure 6: Sub-pixel refinement and relocalisation.

1. **Sub-pixel refinement** Sub-pixel refinement was found to be essential to obtain precise trajectory estimates. A simulated sequence shown in Fig. 6(a) illustrates this importance.
2. **Quadtree** Figure 5 shows how quadtrees affect the spreading of features for outdoor sequences. Vegetation typically gives strong Harris corner responses but often poorly constrains the estimates.
3. **Multi-level Quadtree** It was found that using quadtrees over 2-4 image pyramid levels helps with motion blur.
4. **Loop closures** Figures 4(a)-4(f) show the results with and without loop closure. Loop closure substantially reduces drift without requiring a global minimisation. While the loop closure mechanism is not strictly constant time, 103000 keyframes were processed with a maximum processing time for any observation of 44.1 ms.
5. **True scale descriptors** The use of true scale greatly improved the robustness in difficult conditions such as the strong inter-frame motion shown in Fig. 6(b).

6 Conclusion

This paper described a stereo system that demonstrated how a continuous relative representation (CRR) combined with careful engineering (true scale, subpixel minimisation and quadtrees) can provide constant-time precise estimates, efficiency and good robustness. The CRR framework is more than a simple re-parametrisation and leads to a different cost function. It is possible to represent trajectories that cannot be embedded in a Euclidean space - an assumption common to most previous work. The CRR opens up the prospect of versatile robots that can continuously estimate their trajectory even with unobservable ego-motion such as when using the same means of transport as humans.

References

[1] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE International Conference on Intelligent Robots and Systems*, 2004.

[2] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 2004.

[3] J.A. Castellanos, J. Neira, and J.D. Tardós. Limits to the consistency of ekf-based slam. In *IFAC*, 2004.

[4] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway. Robust real-time visual slam using scale prediction and exemplar based feature description. In *ICCV*, 2007.

[5] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[6] A. J. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. in Pattern Analysis and Machine Intelligence*, 2007.

[7] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *ICCV*, 2007.

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

[9] Martin A. Fischler and Robert 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.

[10] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17:242–257, 2001.

[11] R. Hartley and A. Zisserman. *Multiple View geometry in Computer vision*. Cambridge university press, 2000.

[12] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007.

[13] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, Oct. 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2004832.

[14] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. in Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

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

[16] C. Mei, S. Benhimane, E. Malis, and P. Rives. Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE Transactions on Robotics*, 2008.

[17] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real-Time Localization and 3D Reconstruction. In *IEEE Conference of Vision and Pattern Recognition*, 2006.

[18] D. Nistér and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference of Vision and Pattern Recognition*, 2006.

[19] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 2006.

[20] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *ICCV*, 2005.

[21] G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems Conference*, 2009.

[22] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.

[23] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal for Robotics Research*, 28(5):595–599, 2009.

[24] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. In *ICCV*, 2007.

[25] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. S. Sawhney. Ten-fold improvement in visual odometry using landmark matching. In *ICCV*, 2007.