Unified Loop Closing and Recovery for Real Time Monocular SLAM

Ethan Eade and Tom Drummond
Machine Intelligence Laboratory, Cambridge University
{ee231, twd20}@cam.ac.uk

Abstract

We present a unified method for recovering from tracking failure and closing loops in real time monocular simultaneous localisation and mapping. Within a graph-based map representation, we show that recovery and loop closing both reduce to the creation of a graph edge. We describe and implement a bag-of-words appearance model for ranking potential loop closures, and a robust method for using both structure and image appearance to confirm likely matches. The resulting system closes loops and recovers from failures while mapping thousands of landmarks, all in real time.

1 Introduction

Existing real time monocular SLAM (RTMS) systems rely mainly on tracking to perform localisation at each time step, using a motion model and active search to constrain the camera trajectory and update the map. However, the tracking assumptions are easily violated by unmodeled motion or failure to find landmarks in the video due to blur, occlusion, or an insufficient appearance model. Tracking failure causes incorrect data association and motion estimation, leading to catastrophic corruption of the structure estimate.

Even when the camera motion and environment are favorable, the statistical filtering often delivers inconsistent results on a large scale. This problem is not confined to RTMS, but plagues metric SLAM in general. Resulting maps are locally correct, but globally incoherent, and nontrivial loops are rarely closed by the standard active search techniques.

Such fragility and inaccuracy makes RTMS unusable for most real-world sequences, and motivates the development of active recovery and loop closing algorithms. Recovery typically refers to relocalisation of the camera pose following a tracking failure, while loop closing refers to data association between two distinct parts of the map even when tracking is proceeding smoothly.

We present a unified method for both recovery from failure and active closing of loops in a graph-based RTMS system, using both appearance and structure to guide a localisation search. Crucially, the system continues to map the environment after tracking failure occurs. Upon recovery, the new and old maps are efficiently joined, so no mapping work is wasted or lost. The operations are simple in the context of the graph representation. For recovery, two connected components of the graph are joined into one, whereas for loop closure, two nodes in the same connected component become directly connected, improving the global coherence of the map. The resulting system runs in real time while mapping thousands of landmarks, recovering from multiple tracking failures and closing loops.

2 Related Work

2.1 Real Time Monocular SLAM

One of the first convincing implementations of real time SLAM with a single camera is that of Davison et al[3], which uses active search for image patches and an Extended Kalman Filter (EKF) to map up to 100 landmarks in real time.

More recent work by Eade and Drummond[5] partitions the landmark observations into nodes of a graph to minimise statistical inconsistency in the filter estimates. Each graph node contains landmark estimates conditioned on the local observations, and the edges represent pose transformations (with scale) between the nodes. Because this graph-based SLAM approach can map many landmarks and allows iterative optimisation over graph cycles, we use it as a basis for our system.

Klein and Murray [7] take a novel approach to RTMS, in which tracking and mapping are based on carefully selected key-frames, and a global bundle adjustment over key-frame poses runs in the background while pose tracking runs at frame-rate. This yields excellent results for environments within the limits of the global optimisation. Our method for detecting loop closures and recovering could be applied directly to this approach, as each key-frame is synonymous to a node in our system.

2.2 Recovery

The algorithm of Pupilli and Calway[11] uses a particle filter to model pose, which makes the tracking robust to erratic motion, but fails to account for dependence between the camera and landmark estimates, and cannot make coherent maps for many landmarks.

Williams et al.[15] present a robust relocalisation method built on top of Davison's system. Classification with randomised trees[8] yields image-to-landmark matches, from which pose is recovered when tracking has failed. However, classification using randomised trees breaks down in the domain of thousands of classes, and the online class training and storage cost (30ms, 1.25MB per landmark) is prohibitive when dealing with many landmarks each time step.

2.3 Loop Closing

Davison's system has been extended to allow loop closing when two sub-maps have independently chosen the same landmarks from similar viewpoints[1]. However, the loop closing is not real time (taking more than 1 minute), and the loop detection conditions are rarely satisfied in practice[14].

Loop closing using visual appearance is not a novel idea; the richness of camera data makes it particularly suited to the task of recognizing similarity. Dudek and Jugessur[4] use descriptors derived from principal component analysis over Fourier transformed image patches to describe and match frames, and then use a vote over descriptors to choose a database image. Newman et al.[10] build a similarity matrix to evaluate the statistical significance of matching images when laser range data also matches.

Sivic and Zisserman[12] apply the bag-of-words model used in text retrieval to perform content-based retrieval in video sequences. Affine-invariant descriptors extracted from the videos are clustered at training time, and then quantised to the cluster centers

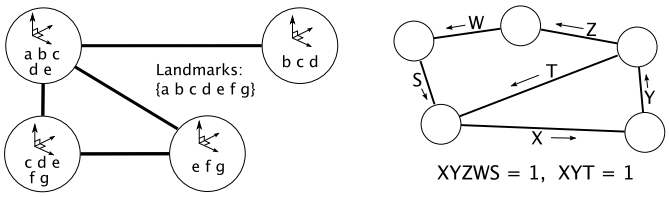

Figure 1: Each node has its own coordinate system and Gaussian landmark estimate independent from other nodes. Edges represent the transformations between nodes induced by shared landmark estimates, with cycles providing additional constraints

at run time to yield visual word histograms in the images. Potential matches are ranked using the term-frequency-inverse-document-frequency metric.

The appearance-based SLAM work of Cummins and Newman[2] applies the bag-of-words method within a probabilistic framework to detect loop closures. A generative model of word expression yields a likelihood of observed words over stored places, permitting maximum-likelihood data association and update of the place's appearance model parameters. While the system delivers high accuracy visual matching, the generative model must be computed offline and the model update cost at each time step is high.

Very recent work by Williams et al. [14] uses the randomised-trees relocalisation method described above to close loops in submap-based SLAM. This has the drawbacks listed above for randomised-trees classification. Further, relocalisation is tried against each submap in turn, in a brute-force manner. Our approach focuses the search using a visual appearance model.

3 Local SLAM Algorithm

Our SLAM system with unified recovery and loop closing extends the graph-based RTMS system of by Eade and Drummond[5]. This section very briefly describes the graph representation of the map estimate and the operation of the system, henceforth called GraphSLAM.

3.1 GraphSLAM Overview

GraphSLAM stores landmark estimates in graph nodes, and maintains estimates of the similarity transformations between nodes. The nodes are statistically independent of each other, as observations of landmarks in each video image are used to update at most one node (where the observation model is nearly linear). However, landmarks are not strictly partitioned between nodes – indeed, the estimates of landmarks shared between two nodes determine the transformation estimate of an edge between the nodes.

The graph is a piecewise-Gaussian representation of landmark estimates. Camera pose is always represented relative to the active node, which can change at each time step. There is no global coordinate frame (see Fig. 1). Instead, estimates are transformed between local coordinate frames via edges.

3.2 Nodes

Within each node, observations are combined using an information filter, yielding a Gaussian posterior with dimension $3N$ for N landmarks. Landmark estimates are stored in inverse-depth coordinates to make the observation model more linear. A bound is placed on the maximum number of landmark estimates per node, so that the update computation time is also bounded.

3.3 Edges and Traversal

An edge between two nodes represents an estimate of the scaled Euclidean transformation between the nodes' coordinate frames. The transformation is constrained by the estimates of landmarks shared between the two nodes (when mapped from one node to the other through the edge, they should align with maximum likelihood).

Each edge cycle in the graph also provides a constraint on every edge in the cycle (each cycle should compose to the identity transformation), permitting iterative optimisation of the edge parameters without modifying the nodes.

3.4 Observations

At each time step, landmark estimates from nearby nodes are projected into the image, determining a gated search region. The patch associated with each landmark is affinely warped to reflect the current pose estimate, and the predicted region in the appropriate image octave is searched for the patch using normalised cross correlation.

When fewer than a specified number of landmarks are visible, new landmarks are chosen in image locations given by an interest point detector. Patches are acquired from the image, and the initial landmark estimates are added to the active node.

3.5 Basic Modifications

We replace the interest point detector used by [5] with a scale space extrema detector, so that each detected interest point has an image scale. The appearance patch of a new landmark is sampled at this scale, and localised in the appropriate octaves of subsequent video frames. This results in more robust operation when small-scale image features are rare. Also, landmark detection must be stable across viewpoint change to allow loop closure and recovery from novel viewpoints.

We also allow multiple connected components in the graph. When tracking is proceeding without trouble, the active node remains in the current connected component. But when few observations can be made, tracking has failed, and a new connected component is created. SLAM operation then starts fresh, with no landmarks in the current active node. Disjoint connected components may later be reconnected as described below.

4 Loop Closing and Recovery Candidate Selection

To close loops or recover, we first select candidate nodes likely to correspond to the active node, using an appearance model based on visual words. Section 5 describes how the coordinate transformation from a candidate node to the current pose is sought.

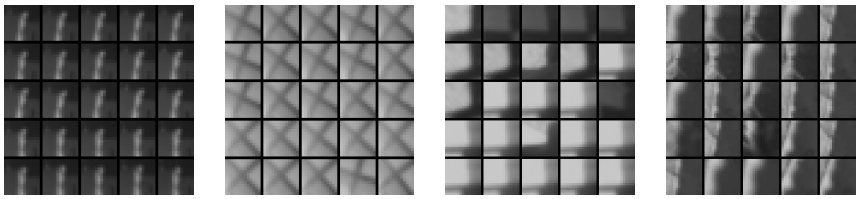

Figure 2: Example image patches that quantise to each of four words in the vocabulary

Both for our coarse bag-of-words appearance model, and for the local landmark descriptor database described in Section 5.2, we use viewpoint-invariant descriptors of scale- and rotation-normalised patches. We compute a SIFT[9] descriptor in the appropriate scale and orientation with a two-by-two spatial grid and four angle bins per spatial bin. These 16-D descriptors are less distinctive than the standard 128-D SIFT decriptors, but are more efficient to compute, store, and compare, and perform well for our application.

4.1 Bag-of-words Appearance Model

We use a bag-of-words appearance model to find nodes that are likely to have similar appearance to the current video image. Visual bag-of-words approaches[12][2] generally extract feature descriptors from an image, quantise the descriptors to a fixed "vocabulary" of visual words, and use the histogram of observed words as an image descriptor. An inverted index or generative model is used to identify images or places that are likely to match the query image.

The vocabulary is typically trained offline from representative training data. To avoid requiring any offline training requirements, we build the vocabulary incrementally during operation. The words of the vocabulary are characterised by the descriptors described above, computed from interest points in each video image.

In order to avoid adding descriptors of unstable or fluke features to the vocabulary, we maintain both a main database V holding the current vocabulary, and a young word database Y containing candidates for addition to V. For each interest point in an image, we compute its descriptor $\mathbf{d}$ and its nearest neighbors $\mathbf{w} \in V$ and $\mathbf{y} \in Y$. Let r_G be the quantisation radius of both V and Y.

- If both $\mathbf{w}$ and $\mathbf{y}$ are farther than r_G away from $\mathbf{d}$, $\mathbf{d}$ is added to Y and assigned a default time-to-live $ttl(\mathbf{y})$ and a counter value $count(\mathbf{y}) = 0$.
- If $\|\mathbf{y} - \mathbf{d}\| < \|\mathbf{w} - \mathbf{d}\|$, $count(\mathbf{y})$ is incremented and $ttl(\mathbf{y})$ reset to the default.
- Otherwise, $\mathbf{d}$ is already sufficiently represented in V, and it quantises to $\mathbf{w}$.

At each time step, $ttl(\mathbf{y})$ is decremented for all $\mathbf{y} \in Y$. If $count(\mathbf{y})$ reaches a threshold before $ttl(\mathbf{y}) = 0$, then $\mathbf{y}$ is moved from Y to the V. Otherwise, it is discarded.

Offline clustering results suggest reasonable values for r_G. When millions of descriptors harvested from many sequences are clustered using k-means, the cluster radius varies inversely with the number of clusters. Grouping into 2000 words yields an r.m.s. cluster radius of 0.34, while grouping into 4000 words gives a radius of 0.29. Using either of the static offline vocabularies at run time yields matching performance qualitatively similar

to our online vocabulary building. We choose a cluster radius $r_G = 0.3$, and the online vocabulary typically converges to 3000 words. See Fig 2 for example quantisations.

4.2 Appearance Search

For each graph node, the system stores a list of words observed in video images while that node has been active, and the occurrence count of each word. If the occurrence count of a word in this list is above a threshold (we use 3), then the word is 'expressed' by that node. Given the existing vocabulary words $\mathbf{W}$ observed by the current video image, the occurrence counts of all $\mathbf{w} \in W$ in the active node are incremented. Then a term--frequency-inverse-document-frequency scheme (similar to that of [12]) is used to rank the nodes that express any words in W. The highest-ranked k nodes not already connected to the active node are candidate node matches to the current view. We use $k = 3$.

5 Loop Closing and Recovery

Here we detail how loop closing or recovery proceeds between the active node and a candidate node. Section 4 describes how candidate nodes are chosen.

5.1 Loop Closing $\equiv$ Recovery

Loop closing and recovery in our system are the same event under slightly different circumstances. Loop closure occurs when a new edge is created between the active node and another node in the same connected component, creating a cycle in the graph. Recovery occurs when a new edge is created between the active node and a node in a different connected component, thus merging them into one connected component (see Fig. 3).

This unification of loop closure and recovery has important benefits to SLAM. Firstly, the system is always mapping; it just creates a new graph component when failure occurs, to represent subsequent map estimates. There need not be a separate behavior when 'lost' – as long as the failure event is reliably detected, a new component is created and the map remains uncorrupted.

Secondly, and more crucially for extended operation, recovery-as-component-reconnection means that no mapping opportunities are wasted. If tracking fails near the beginning of a long loop, a recovery mechanism like the one described by [15] can not relocalise until the original landmarks are once again visible. In contrast, our system immediately starts mapping in a new graph component, and when the early landmarks reappear, the map of the greater part of the loop is connected with that of the beginning.

5.2 Local Landmark Appearance Model

The appearance-based candidate selection method in Section 4 chooses graph nodes whose observed landmarks are likely to be visible in the current video image. To localise with respect to such a node, correspondences between features in the video image and landmark estimates in the candidate node must be established. To this end, each graph node maintains a local appearance model of its landmark estimates. This is distinct from the global bag-of-words visual appearance model, and is used only for matching candidate nodes' landmarks to keypoints in new video images for loop closing.

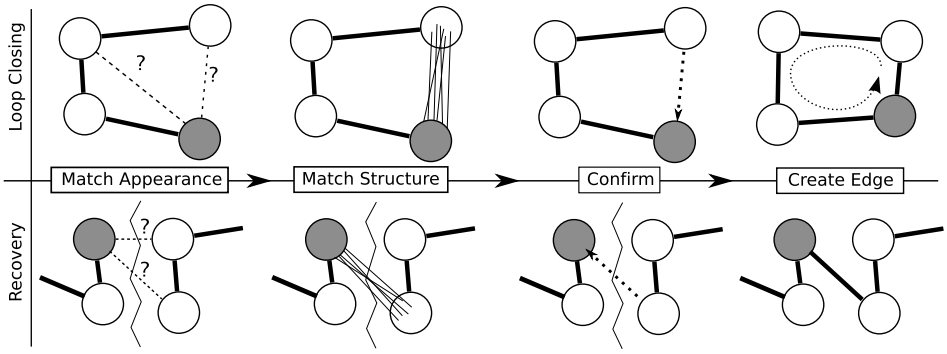

Figure 3: Loop closing and recovery: Candidate matching nodes are chosen by visual appearance. Then structure is matched using landmark appearance models, and a candidate edge is created. Observations are made via the candidate edge until it is promoted, at which point a cycle is created (top) or two components are connected (bottom). The active node is shaded

A set of descriptors represents the various appearances of all observations of all landmarks in the node. The set S_i (for node N_i) is built incrementally: Whenever a landmark L_j is observed in N_i, a descriptor $\mathbf{d}_j$ is computed at the position and scale of the observation. Let $\mathbf{e}_k \in S_i$ be the nearest neighbor in S_i to $\mathbf{d}_j$, and a descriptor of landmark L_k. If $\|\mathbf{e}_k - \mathbf{d}_j\| > r_L$ or $k \neq j$, then $S_i \leftarrow S_i \cup \{\mathbf{d}_j\}$. That is, the descriptor is added to S_i if its nearest neighbor is sufficiently distant or describes a different landmark.

Thus variations in the appearance of a landmark are represented in S_i to within distance r_L in descriptor space. We use $r_L = 0.15$, which gives a much finer representation of descriptor variation than in the global bag-of-words vocabulary. This choice of r_L is guided by the observation that even descriptors of the same image patch after rotations, scalings, and small deformations vary within a radius 0.1 in descriptor space.

5.3 Descriptor Matching and Robust Model Fitting

For all of the interest points detected in each video image, the descriptors described above are computed and matched to their nearest neighbours in a candidate node's local landmark appearance model. For every landmark in the candidate's database, there might be many local interest point descriptors matching to it. We use MLESAC[13] to find the correct correspondences. Any three matches from descriptors to distinct landmarks in the candidate node determine a pose[6]. For many such poses, the maximum-likelihood set of inlier correspondences are computed, with a fixed log-likelihood of outliers of -5.0. Up to 200 random three-point-pose hypotheses are tried. This is similar to the approach of [15].

5.4 Candidate Edge Trial Period

If the maximum likelihood set of inliers from pose-fitting is large enough (we require 8 inliers), matching is considered successful. A candidate edge is created from the candidate node to the current active node, using the pose result from MLESAC as the edge

transformation. The candidate edge does not act as a standard edge in the graph – the camera cannot traverse it, and landmark observations that would result in node updates are not made through it.

Instead, after the pose estimate has been constrained by standard observations, additional landmark predictions from candidate nodes are made via any candidate edges pointing into the active node. The success or failure of active search for such landmarks serves only to score the viability of the candidate edges. We use a simple heuristic: if the ratio of total failed to successful predictions exceeds a threshold, the candidate edge is discarded. When the inverse ratio exceeds the threshold, the candidate edge is validated, and a loop closure or recovery event occurs.

5.5 Connecting the Nodes

When a candidate edge is promoted to a normal graph edge, either a cycle is created in the graph, or two components are connected. In the first case, the existing graph cycle optimizer will incrementally adjust the graph to satisfy the constraint created by the new edge cycle.

This second case represents a recovery event. If the tracking failure that created the newer component was very recent, almost no mapping has occurred in the new component before reconnection. To simplify the graph in this common recovery situation, the ages of the two components being merged are checked. If the newer component is very young, it is discarded as ephemeral, and the camera pose is transformed back into the older component's matched node, through the newly created edge. The edge is then discarded, and SLAM continues in the original component.

6 Results

We have implemented our method for a dual-core computer. Image searches and filter updates happen in parallel with interest point detection, descriptor computation, and bag-of-words maintenance. On a 2.2 GHz Pentium Core 2 Duo, per-frame processing never exceeds 33 ms, with loop detection/recovery requiring no more than 6 ms. The system successfully closes loops and recovers from tracking failure in both indoor and outdoor sequences, while operating in real time and mapping thousands of landmarks.

We use a completely planar real scene as a basic test of reconstruction accuracy. The camera hovers at typical viewing distance h above one part of the scene, before being kidnapped to the other half. The system continues mapping in a new component. When the camera again views the original portion of the scene, the two components are matched and reconnected. The final map contains 251 landmarks. All 226 landmarks with depth uncertainty $\sigma < h/50$ are no farther than $h/100$ from the maximum likelihood plane.

In an outdoor sequence, the camera moves in an elliptical loop, with the camera facing outwards. Rough camera motion causes tracking failure, but the system immediately recovers. Extended failure occurs when the camera is suddenly rotated toward the ground. Mapping of novel views then continues in a new component. As the camera returns to near the starting point, a node in the first connected component is recognised and matched, and the components are merged. As the trajectory continues around the loop a second time, the loop itself is closed. The resulting map contains 1043 landmarks.

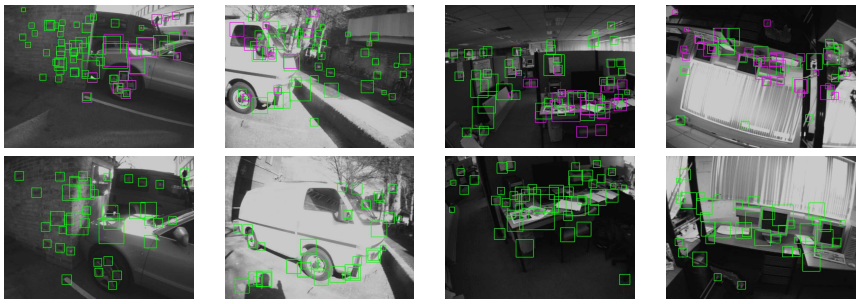

Figure 4: Top: video frames of loop closure or recovery events. Bottom: the most similar previous view of the scene. Normal observations are green, while observations via candidate edges are magenta

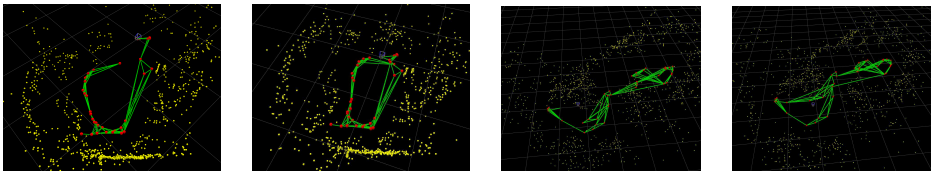

Figure 5: Before and after loop closure in two sequences. Landmarks are yellow, graph edges are green, nodes are red, and the camera is a small frustum. Each pair is from consecutive time steps (33 ms apart), before further incremental refinement by the optimiser

In an indoor scene, a complex external loop is traversed and closed. Then the camera is repeatedly kidnapped from one part of the environment to another, with new viewpoints significantly different from the originals. In all cases, recovery occurs within 15 frames. The final map contains 1402 landmarks.

7 Future Work

The efficient method we have presented greatly improves the robustness of real time monocular SLAM, but is not flawless. The worst failure mode of the system is spurious loop closure given extensive repeated structure. In testing, this occurs only in synthetic sequences with large repeating textures. The problem is particularly difficult to solve in general, as repeated structure at arbitrary scales might be encountered. A probabilistic model for appearance-based loop closure, as in [2], could mitigate the issue.

Another problem is that the cycle optimisation treats the edge transformation estimates as independent, though they are in fact correlated through the node estimates. This results in over-confident and incorrect global maps when many loops are optimised. We plan to address this using conservative local graph optimisations.

While the bag-of-words appearance model is sufficiently distinctive for our test environments, we intend to evaluate its performance and discrimination in much larger environments (where the graph is significantly larger).

8 Acknowledgements

We thank Gerhard Reitmayr for many useful discussions about this work. We gratefully acknowledge the financial support of the NSF (GRF grant DGE-0639132).

References

[1] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos. Mapping Large Loops with a Single Hand-Held Camera. In *Robotics: Science and Systems*, June 2007.

[2] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proc. IEEE Int'l Conf. Robotics and Automation*, 2007.

[3] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, 2007.

[4] G. Dudek and D. Jegessur. Robust place recognition using local appearance based methods. In *Proc. 2002 Int'l Conf. Robotics and Automation*, 2002.

[5] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *Proc. 11th IEEE Int'l Conf. Computer Vision*, 2007.

[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] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM Int'l Symp. Mixed and Augmented Reality*, 2007.

[8] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1465–1479, 2006.

[9] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[10] P. Newman, D. Cole, and K.L. Ho. Outdoor slam using visual appearance and laser ranging. In *Proc. 2006 IEEE Int'l Conf. Robotics and Automation*, 2006.

[11] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proc. 2005 British Machine Vision Conference*, 2005.

[12] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. 9th Int'l Conf. Computer Vision*, 2003.

[13] P. H. S. Torr and A. Zisserman. Mlesac: a new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, 2000.

[14] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardos. An image-to-map loop closing method for monocular slam. In *Proc. Int. Conf. Intelligent Robots and and Systems*, 2008.

[15] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *Proc. 11th IEEE Int'l Conf. Computer Vision*, 2007.