Robotics and Autonomous Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A comparison of loop closing techniques in monocular SLAM

Brian Williams [a,*], Mark Cummins [a], José Neira [b], Paul Newman [a], Ian Reid [a], Juan Tardós [b]

[a] *Department of Engineering Science, University of Oxford, United Kingdom*
[b] *Department of Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Loop closing
Monocular SLAM
Submapping

ABSTRACT

Loop closure detection systems for monocular SLAM come in three broad categories: (i) map-to-map, (ii) image-to-image and (iii) image-to-map. In this paper, we have chosen an implementation of each and performed experiments allowing the three approaches to be compared. The sequences used include both indoor and outdoor environments and single and multiple loop trajectories.

1. Introduction

Loop closure detection is an important problem for any SLAM system and, since cameras have become a common sensor in robotics applications, more people are turning towards vision based methods to achieve it. In this paper, we compare three quite different approaches to loop closure detection for a monocular SLAM system. The approaches essentially differ in where the data association for detecting the loop closure is done — in the metric map space or in the image space. The three approaches are as follows:

- **Map-to-map** — Correspondences are sought between features in two submaps taking into account both their appearance and their relative positions. In this paper we look at the method of Clemente et al. [1], who applied the variable scale geometric compatibility branch and bound (GCBB) algorithm to loop closing in monocular SLAM. The method looks for the largest compatible set of features common to both maps, taking into account both the appearance of the features and their relative geometric location.
- **Image-to-image** — Correspondences are sought between the latest image from the camera and the previously seen images. Here, we discuss the method of Cummins et al. [2,4]. Their method uses the occurrences of image features from a standard vocabulary to detect that two images are of the same part of the world. Careful consideration is given to the distinctiveness of the features — identical but indistinctive observations receive a low probability of having come from the same place. This is done to minimise false loop closures.
- **Image-to-map** — Correspondences are sought between the latest frame from the camera and the features in the map. We examine the method of Williams et al. [5] who find potential correspondences to map features in the current image and then use RANSAC with a three-point-pose algorithm to determine the camera pose relative to the map.

First, we describe the underlying monocular SLAM system used during the experiments. Then, we outline in more detail the chosen implementation of each of the different approaches to loop closure. Results are then given on the performance of each algorithm at closing loops in three different environments. Then one of these sequences is used for more extensive experiments to allow quantitative comparisons to be made between the three methods.

2. The monocular SLAM system

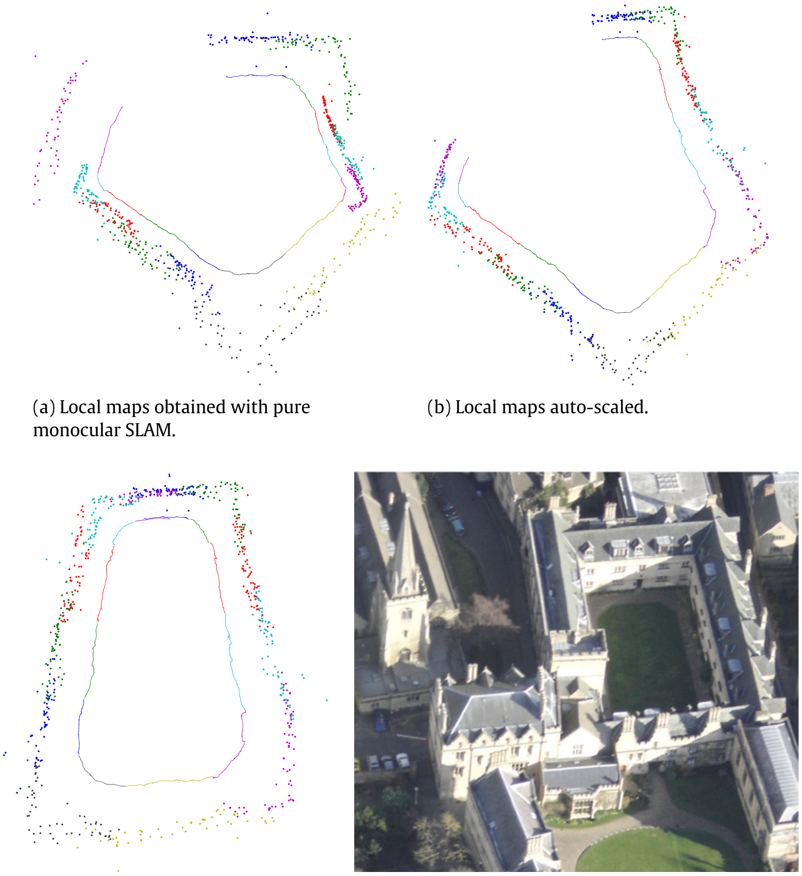
The monocular SLAM system we use is derived from Davison's original system [6,7], but with a few improvements to bring it up to date. The underlying system is essentially the same as the system described in [1] but with our own relocalisation module [3] to recover from situations where the system becomes lost. We have also added a system to prevent premature loop closure and added the ability to perform independent map joining. Here we give a brief description of the system, so details of the loop closing system can be better understood.

2.1. Map building

The monocular SLAM system tracks the pose of a handheld camera while simultaneously building a map of point features in 3D using the EKF. The points are initialised using the inverse depth parameterisation [8], and they are recognised in subsequent frames via normalised cross correlation. An image patch is stored

[*] Corresponding author. Tel.: +44 1865283049.
E-mail address: bpw@robots.ox.ac.uk (B. Williams).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.06.010



(a) Local maps obtained with pure monocular SLAM.

(b) Local maps auto-scaled.

(c) After loop closure.

(d) Aerial view of the courtyard.

Fig. 1. Pembroke college sequence: Twelve submaps with a total of 848 features were made during the 70 m trajectory around the courtyard.

when the feature is initialised, but is warped to correspond with the current camera pose estimate. To speed up the observation of features, the image is only searched in an ellipse given by the uncertainty in the camera and feature estimate in a process called active search. By gating the search in this way the chances of incorrect data association are reduced. This is further helped by the use of the joint compatibility branch and bound algorithm (JCBB) [9] which detects observations which are incompatible with the others and rejects them.

Despite the improvement given by active search and JCBB, there is still a chance of incorrect data association, particularly near loop closures when the system can believe that distant features are again visible and attempt to measure them. If the system is allowed to observe these features as usual, it will likely make incorrect data association due to the large uncertainty in the camera pose relative to these features. Our approach is to prevent the system from making these observations and delay the loop closure until a separate loop close module has detected it (Section 3). To determine which observations to attempt, we make use of the covisibility data from all the features in the map.

With every set of observations, a tally is updated indicating which features have been successfully observed together. Using this information, a simple graph is constructed where a vertex corresponds to each feature, and the edges indicate those that have been observed together. This graph provides an easy way of determining which features are in the local neighbourhood and which are not. Those which are distant in the graph are not eligible for observation since their relative position to the local features is very uncertain and attempting their observation would likely lead to incorrect data association. Readers should note that another way of determining feature covisibility in a stochastic map is to compute the inverse covariance, the information matrix. Features that have been covisible at some point will have a high value of co-information.

2.2. Larger maps

Due to the accumulation of linearisation errors in the EKF algorithm as well as the increase in update time, we limit our system to quite small local maps (around 70 features). To map larger regions, the Hierarchical SLAM [10] technique is used. This allows the system to map an environment by building a series of submaps, each of which is small enough to allow the system to be run in real-time as well as reducing linearisation errors. This method was already applied to monocular SLAM in [1] but here we show it working in more complex environments with multiple loops.

As each new submap is created, the origin of its base reference frame is stored in the state vector of the submap from which branched off. This transformation is then used to determine the relative position of the two submaps. However, for monocular SLAM, this transformation must also include the scale difference which is determined as follows. Each new submap is created with new features initialised at the location of some of the features in the previous map. The geometry of these common features in each submap is used to determine the relative scale. Since the features were newly initialised, information is not shared between the submaps and they remain independent. This scale correction can be seen in Fig. 1(a) and (b).

Please cite this article in press as: B. Williams, et al., A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.06.010

The set of submaps can be represented as a graph where edges contain the similarity transform giving the relative orientation between submaps. As in the Atlas system [11], the entire map can be represented relative to the current submap. To do this, the transformations are composed along a tree built using Dijkstra's algorithm and the uncertainties of the relative transformations. If the camera begins to revisit the neighbouring submap in the graph, this is detected by projecting the features from the neighbouring submap into the current image. If a greater number of features should be visible from the neighbouring submap than the current submap then the transition is made. To retain the independence between the submaps, the relocalisation algorithm is used to determine the pose and uncertainty of the camera relative to the new map before resuming tracking.

When loop closure is detected, the similarity transform between the two submaps is determined and a new edge is added to the graph allowing the system to fully traverse the loop. However, though the transformations between submaps will be good locally, cycles in the graph allow the map estimate to be improved for the purposes of global reasoning.

The Hierarchical SLAM system allows the global map estimate to be refined by imposing the constraint that the composition of all the transformations around each loop in the map should equal the identity. The best global estimate can then be found using non-linear optimisation techniques. This process can be delayed until a global map is required and it must be reestimated if submaps within the loop are revisited causing updates in the relative transformations.

The result of this optimisation process can be seen for a single loop in Fig. 1 or in a more complex multi-loop environments in Fig. 4.

3. Detecting loop closure

In order to close loops in a map, the system must recognise when it has returned to a previously mapped region of the world. Essentially, at this point two regions in the map are found to be the same region in the world even though their position is incompatible given the uncertainty estimate in the map — the classic loop closure problem. The system must then be able to calculate the transformation needed to align these two regions to 'close the loop'. Since an incorrect loop closure can be disastrous for most SLAM systems, a good loop closure detection system should give very few (ideally zero) false positives while still detecting many of the true positives.

In the following sections, we describe three methods for detecting loop closure based on quite different approaches. We will later test the performance of all three algorithms.

3.1. Map-to-map matching: Clemente et al.

Clemente et al. [1] presented a method to close loops in monocular SLAM maps based on finding correspondences between common features in different submaps. The algorithm used is a variable scale version of the original geometric compatibility branch and bound algorithm (GCBB) [12]. The system uses both similarity in visual appearance (unary constraints) and relative distances between features (binary constraints) to find the largest compatible set of common features between two submaps. Once a consistent set has been found, the relative scale, rotation, and translation needed to align the two submaps can easily be determined.

The system was shown to work in [1] where it found a set of five common features between the first and last submaps in a large loop.

3.2. Image-to-image matching: Cummins et al.

Cummins et al. [2] have developed a method to detect loop closures based on recognising the visual appearance of previously seen places. The matching is performed by detecting in each image the presence or absence of features from a visual vocabulary [13] based on SURF features [14], which is learned off-line from training data. Note that the training data consists of generic images not collected in the environment where loop closure detection is performed. The system takes into account the probabilities of features appearing together, and is able to work out the probability that two images show the same region of the world. This method does not depend on a metric map being created since it only compares images directly. However, it can be used with a metric map if the camera pose relative to such a map can be found for each image as well as the relative pose between two images for the loop closure. Much work has been done on this problem in the field of computer vision [15].

3.3. Image-to-map matching: Williams et al.

In [5] a loop closure detection method is proposed which is based on a relocalisation technique used to recover from tracking failures [3]. This relocalisation module determines the pose of the camera relative to a map of point features by finding correspondences between the image and the features in the map. The pose is then determined from the correspondences using RANSAC and the three-point-pose algorithm [16].

The relocalisation module is able to run faster than the frame rate through the use of a fast matching algorithm [3] based on the randomised fern classifier [17]. While the features are being tracked, each successful observation is used to train the classifier. This classifier is fast but it has a high false positive rate. Incorrect classifications are handled using RANSAC.

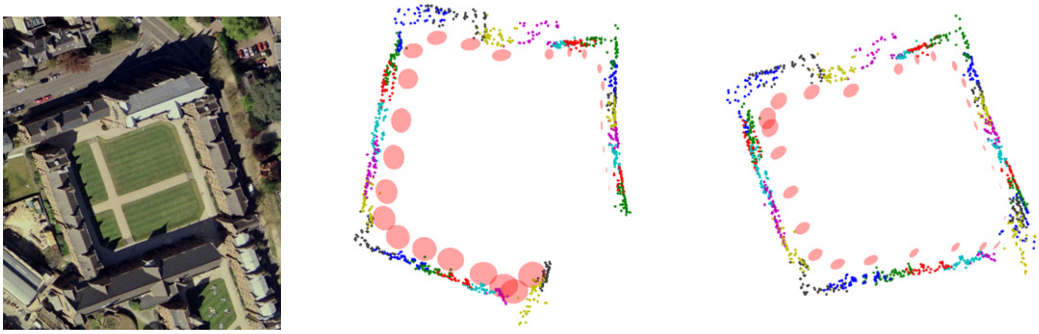
To detect loop closures, the system uses the module to attempt relocalisation in distant regions of the map according to either the feature covisibilities or, if submapping is used, then in other submaps. When a relocalisation is successful, it gives a correspondence between the current pose being tracked, and the pose given by the relocalisation elsewhere in the map. This gives the translation and rotation needed to align the two regions, but a single pose is not enough to determine the scale difference. To achieve this, the camera is tracked for some time in both regions (while freezing one of the maps so information is not counted twice), and this common trajectory can be used to find the transformation between the two regions including the relative scale difference (Fig. 5).

4. Results

The loop closure detection techniques were tested on three different image sequences. One of these sequences was then chosen for more extensive quantitative testing of each algorithm using a second lap of the same loop. First we will discuss the general performance of the algorithms in the three sequences before presenting the quantitative results with more discussion on the process of detecting loop closure with each algorithm.

4.1. General performance

We have used the monocular SLAM system to build a map of three different environments. Due to the size of the environments, the system builds a series of submaps as the camera is moved facing the wall. Each new submap is begun by initialising new features in the same image locations as those just observed as the



(a) Aerial view of the courtyard. (b) Before loop closure. (c) After loop closure.

Fig. 2. Keble college sequence: Twenty eight submaps with a total of 1983 features were made during the 230 m trajectory around the courtyard. Ellipses indicate 1σ uncertainty bounds for the origin of the base reference frame for each submap.

last submap finished. These common features can then be used to fix the relative scale between submaps as shown in Fig. 1.

Even after the scale between submaps has been corrected, the maps still exhibit a common problem, that although the camera has returned to the same region in the world, this is not reflected in the map. A loop closure detection system is needed to recognise that the system has traversed a loop so the map can be corrected accordingly. We have used all three algorithms to try to detect the loop closures in each sequence.

4.1.1. *Keble College sequence*

The first sequence we tested is the one used by Clemente et al. in [1] where the map-to-map method was originally proposed for a monocular SLAM. In this sequence, the camera moves around a courtyard in Keble College, Oxford. The map built using this sequence can be seen in Fig. 2. As already shown in [1], the map-to-map method can find the correspondence between the first and last submaps in this sequence. We then tested the other two methods on this sequence and they both successfully detected the loop closure event.

Robust loop closure detection in repetitive environments is a well known problem. In several locations in this sequence, two distinct regions of the world are very similar in both appearance and geometry. An example of this is shown in Fig. 6 where the majority of the scene is almost identical but the structure to either side of the archway is different. The image-to-image method uses temporal information from recent keyframes to build up confidence in a loop closure. Without this temporal prior, taking into account this single image, a loop closure is hypothesised between these two images. The image-to-map method can also be made to hypothesise a loop closure here if the thresholds on the number of inliers for RANSAC are reduced so that it is able to ignore the conflicting information to the sides of the archway. Commonly suggested techniques to avoid these false loop closures are to use more observations as is done here, or to gate using the Mahalanobis distance in the global metric map.

4.1.2. *Pembroke College sequence*

The next sequence also records the trajectory of the camera as it moves around a university courtyard. In the Pembroke College sequence though, two laps were recorded. Later we will use this second lap to test the three algorithms more thoroughly.

The map created for this sequence can be seen in Fig. 1. Both the image-to-image and the image-to-map methods were very successful in this sequence. The map-to-map method did not reliably detect the loop closure event since during some runs through the sequence the SLAM system did not initialise enough common features in the same locations when reobserving the start of the loop. The results for this sequence will be discussed more extensively in Section 4.2.

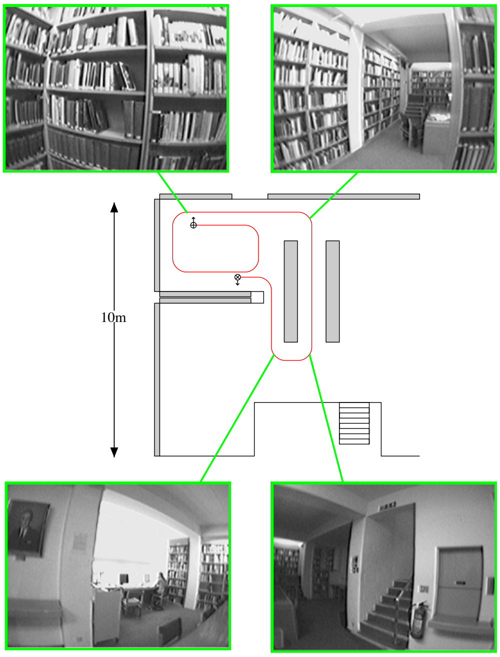


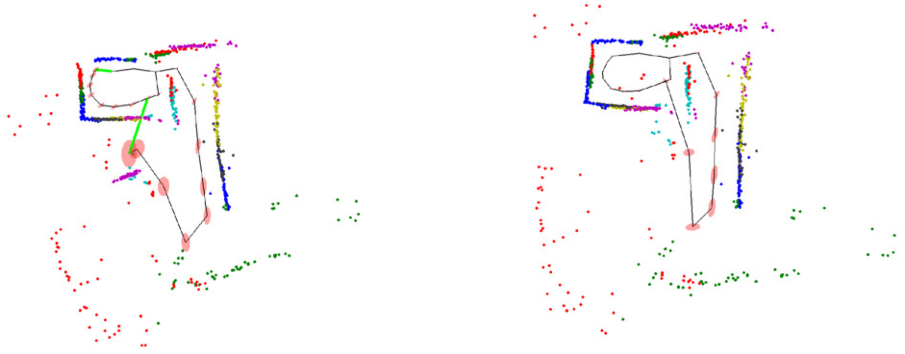
Fig. 3. Library sequence: This diagram shows the room used for the library sequence. The camera starts at ⊕ and follows the trajectory around a multi-loop path through the library to ⊗. The arrows indicate the direction the camera faces which is roughly perpendicular to the direction of motion. Four example frames are given for the sequence indicating the range of appearance of the scene. This sequence demonstrates the systems ability to retraverse previous submaps and close multiple loops. The results for this sequence are shown in Fig. 4.

4.1.3. *Library sequence*

The final sequence was taken inside a library with a trajectory that included multiple loops as shown in Fig. 3. The map created by the SLAM system for this sequence is shown in Fig. 4. This sequence required the more complex submapping framework outlined earlier. After closing the first loop, the system retraverses several old submaps before branching off a new series of submaps as it moves around the second loop. Though both loop closures are performed online and the new edge is added to the graph immediately, the global map is only optimised afterwards. This is performed in Matlab and took 1.5 s for ten iterations. In practice, the global map is only needed for global problems like path planning along a novel route.

Again, the image-to-image and the image-to-map algorithms were able to easily detect the loop closure events in this sequence while the map-to-map method had difficulty. Due to the rich texture of the bookshelves in the library, there were many potential places for the tracking system to place map features.

Please cite this article in press as: B. Williams, et al., A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.06.010



(a) Before loop closure optimisation.

(b) After loop closure optimisation.

Fig. 4. **Library sequence:** Twenty submaps were created as the system built a map of the library shown in Fig. 3. Ellipses indicate 1σ uncertainty bounds for the origin of the base reference frame for each submap. The graph connecting submaps is shown in black with thick edges indicating links made using loop closure detection. The scale of the two submaps in the lower part of the figure was not well estimated due to the camera undergoing mostly rotation with little translation at this part of the sequence.

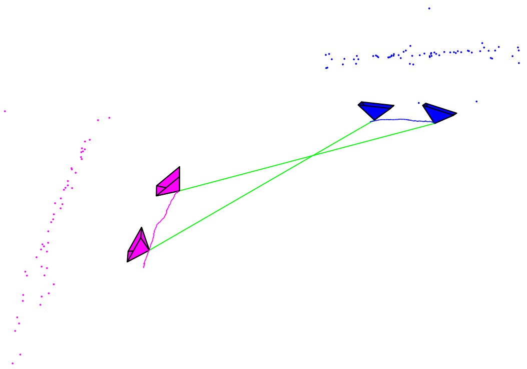


Fig. 5. Image-to-map loop closure for the Pembroke College sequence. While tracking in the twelfth submap (left), the system relocalises in the first submap (right). The transformations between the two supmaps is found by first aligning the common trajectories, and then enforcing the constraint that the two sets of corresponding camera poses (linked by straight lines) are equal.

This made it quite unlikely that they would be placed in the same location when the camera revisits a place making it impossible for the map-to-map method to succeed.

4.2. Extensive testing

We have evaluated the performance of the algorithms further by checking their susceptibility to false positives and their run time. These tests are performed using the Pembroke College sequence.

4.2.1. Map-to-map matching: Clemente et al.

On some runs of the sequence, when the system comes to close a loop using the map-to-map method, it is able to successfully find common features between the two maps as shown in Fig. 7(a). Unfortunately, during the loop closure, there is no guarantee that the system will have initialised features in exactly the same place in the two overlapping submaps. In fact, in our experiments to date, we have found submaps with sufficient common features to detect the loop closure in this way to be rare. Fig. 11(a) shows an example of the same frame being tracked in two different submaps. Despite the large number of features visible, only two features are common to both maps. This is not enough to determine the transformation between the submaps.

Even getting a corresponding set of features does not guarantee a true correspondence between the two submaps. Fig. 7(b) shows that the GCBB algorithm also found sets of five "common" features between eight other pairs of submaps. We were unable to find a threshold able to reliably distinguish between true positives and false positives for the maps created by our SLAM system.

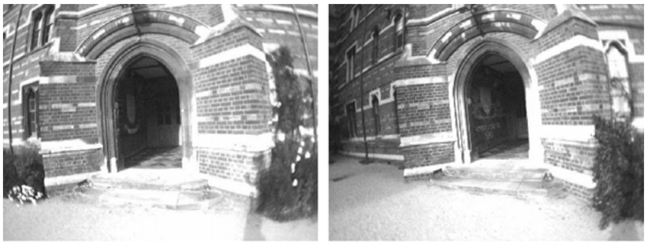


Fig. 6. Despite these two regions of the world having both similar appearance and structure, they belong to different parts of Keble College courtyard. A loop closure detection system should indicate a match and a higher level system should then be used to mark it as false using either the uncertainties in the global metric map or more observations before and after the archway.

During our tests, the variable scale GCBB algorithm took around 100 ms[1] to compare two submaps. When the SLAM system finishes one submap, there is easily time to compare this submap to all previous submaps before the next one is completed.

4.2.2. Image-to-image matching: Cummins et al.

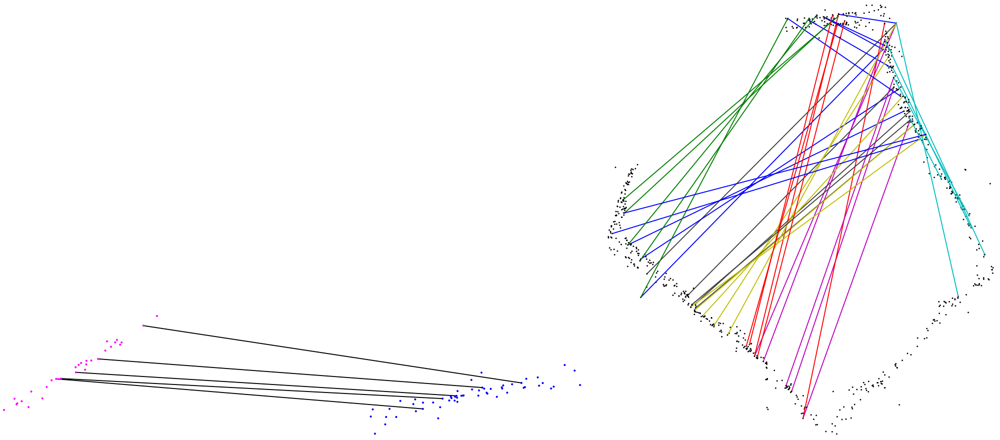
The image-to-image matching method of Cummins et al. is designed to work with non-overlapping key frames. When run on a robot, the odometry is used to trigger key frame capture. Without odometry, we simply used every 40th frame of the video to test the system. Ideally though, an automatic key frame detector should be used.

The loop closure detection system determines for each of these input images if it is a new place or a loop closure. On the Pembroke College sequence, the algorithm correctly gave a high probability that each image was a new place until the camera had traversed the loop and returned to the start of the loop. At this point, the system gave a high probability (99.9%) that the most recent image corresponded to an image at the start of the sequence (Fig. 8(a)).

To test the reliability of the loop closure detection, we computed loop closures for every frame from a second lap of the Pembroke courtyard, against the set of images from the first lap. This simulates the 'kidnapped robot situation', a sudden transition from the end of the first loop to a random part of the courtyard. This tests if the algorithm would be able to detect a loop closure at each position. The results are shown in Fig. 8(b) where frames that matched an image in the previous loop are marked. A threshold

[1] Tests were done on a Dual Core 3 GHz machine.

Please cite this article in press as: B. Williams, et al., A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.06.010



(a) **Map-to-map matching:** Loop closure detected using the method of Clemente et al. [1]. The system finds a set of five features consistent in both geometry and appearance between the first and last submaps. It is only successful if the SLAM system has initialised common features in the two submaps. In another run of the same sequence, the loop closure was missed (Fig. 11(a)).

(b) **Map-to-map reliability:** Matching was attempted between every pair of non-consecutive submaps. Shown here are the eight false positives sets with five correspondences.

Fig. 7. The performance of the map-to-map method in the Pembroke College sequence.

was chosen that removes all false positives to allow comparison with the image-to-map method. The system found matches that met this probability threshold in 8% of attempts indicating that the system would be able to close the loop at these positions. The precision–recall curve in Fig. 10 shows the effect of the probability threshold on the reliability of the system.

On each image, the algorithm takes on average 283 ms to run. Much of this time (73 ms) is taken up by SURF feature detection. This method relies on this descriptor which is richer yet slower than the randomised fern classifier. The overall speed is slower than the frame rate, however, the loop closing algorithm does not need to be run on every frame.

4.2.3. Image-to-map matching: Williams et al.

For each frame, there is usually enough remaining time after tracking to attempt relocalisation in one other submap. The system cycles through submaps until a relocalisation is successful, indicating a loop closure. For the Pembroke College sequence, the system successfully detected the loop closure as the features in the original map came back into view (Fig. 9(a)). Note that for this method, no common features are needed between submaps as they are for the map-to-map method.

The reliability of this loop closure method was tested using the same 'kidnapped robot' situation we used to test the image-to-image method. The system was allowed to continue searching for loop closures as the camera continued around the courtyard for a second lap. For the test, the system attempts relocalisation in every submap for every frame. The results of this test can be seen in Fig. 9(b).

The method takes 10–15 ms to find potential matches to map features in each image. The remaining time is used to run Ransac on the matches to determine the pose. This is usually found within a few milliseconds if a valid pose exists for those matches. This is fast enough to allow the algorithm to run on a single submap after the system has finished tracking in each frame.

5. Discussion

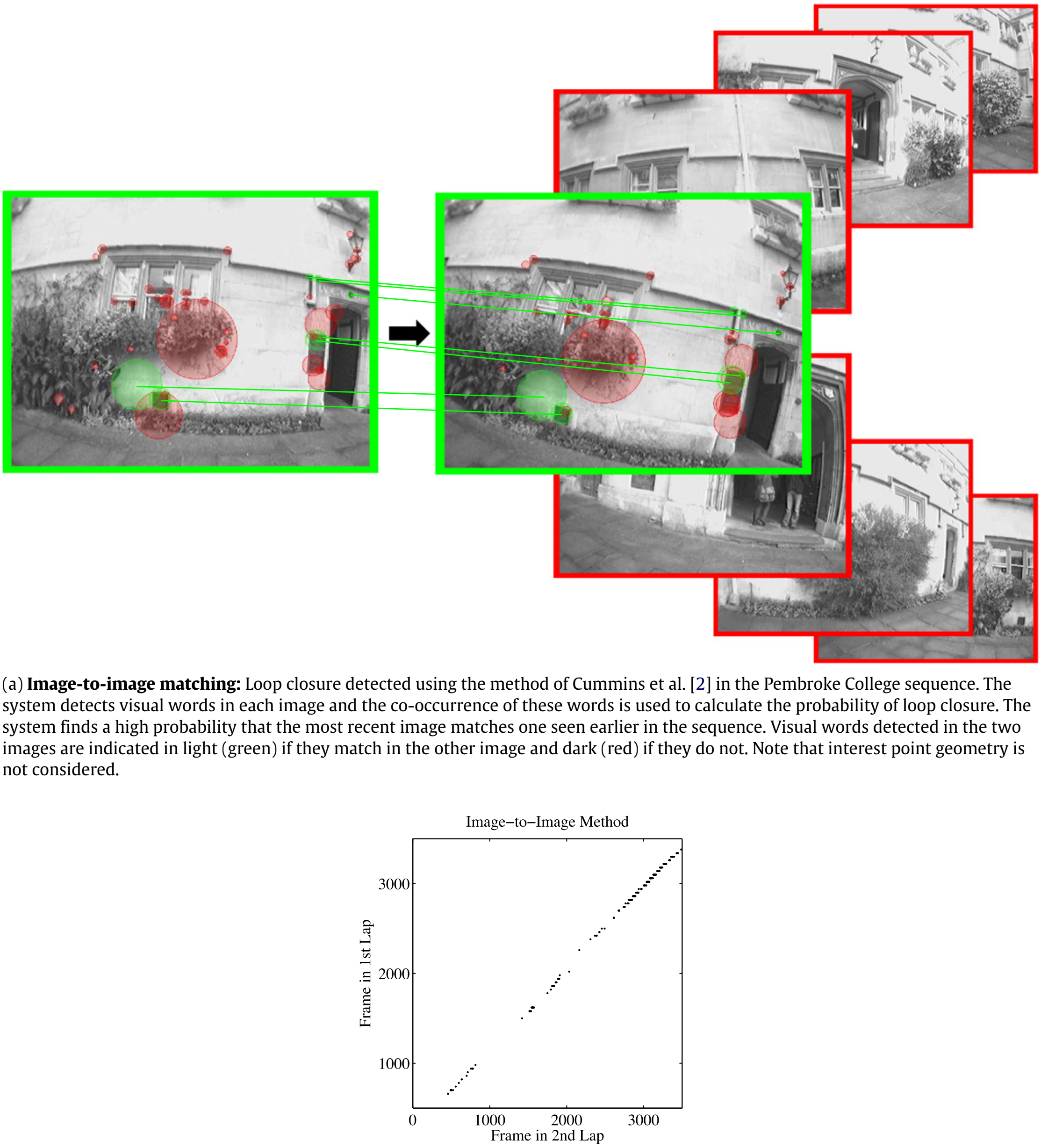
The results of our quantitative testing of the three methods using the second lap of the Pembroke College sequence are shown in Fig. 10. The salient points that should be taken from this are:

- To create the curve for the image-to-map method, we varied the threshold on the fraction of successful landmark observations after a potential relocalisation. All other parameters were left at their default values. In practice, we require 50% of observations to succeed giving the 20% recall at 100% precision quoted in this paper.
- The curve for the image-to-image method is more complete since a single threshold (the match probability) can be varied to achieve a larger range of recall. The performance is similar to the image-to-map method with only a few extra false positives detected with high probability.
- The curve for the map-to-map method has fewer points. The threshold varied here is the number of landmarks in the compatible set. The twelve submaps built in the second lap were matched to the twelve from the first lap. In this run, a single true positive was found with a seven common features. For all lower thresholds many false positives were also found leading to a steep drop in precision.

We would not want the reader to infer too much from these numerical results. The actual numbers found would depend on the precise sequence used and the characteristics of the environment, but the general trends in the graph are representative of our overall findings. Of most importance are the relative performance, benefits, and failure modes of each algorithm. These aspects are now discussed more qualitatively.

We found the map-to-map matching technique of Clemente et al. is successful when a sufficient number of common features exists between two submaps. For instance, it reliably detect the overlap between consecutive submaps where common features are intentionally added to determine the relative scale. However, in general it is unsuitable for the sparse maps created by the monocular SLAM system. These maps are designed to be good enough to track the camera but otherwise as sparse as possible to allow faster updates. Perhaps a map-to-map based method would be more suitable if the maps contained higher level information or there were more consistency on which potential features are added to the map.

The image-to-image matching technique of Cummins et al. works well since it can be tuned to remove all false positive while still detecting 8% of true positives for the Pembroke College sequence. As the probability threshold is lowered, the first few false positives could easily be removed with a simple essential



(a) **Image-to-image matching:** Loop closure detected using the method of Cummins et al. [2] in the Pembroke College sequence. The system detects visual words in each image and the co-occurrence of these words is used to calculate the probability of loop closure. The system finds a high probability that the most recent image matches one seen earlier in the sequence. Visual words detected in the two images are indicated in light (green) if they match in the other image and dark (red) if they do not. Note that interest point geometry is not considered.

(b) **Image-to-image reliability:** A second lap of the Pembroke College sequence was used to test the reliability of the image-to-image method. Correspondences were found between every frame in a second lap and every 40th frame in the first lap. With a threshold chosen to remove all false positives the system was successful in 8% of attempts. Gaps are in regions of the world with lots of foliage (where the image-to-map method also struggles).

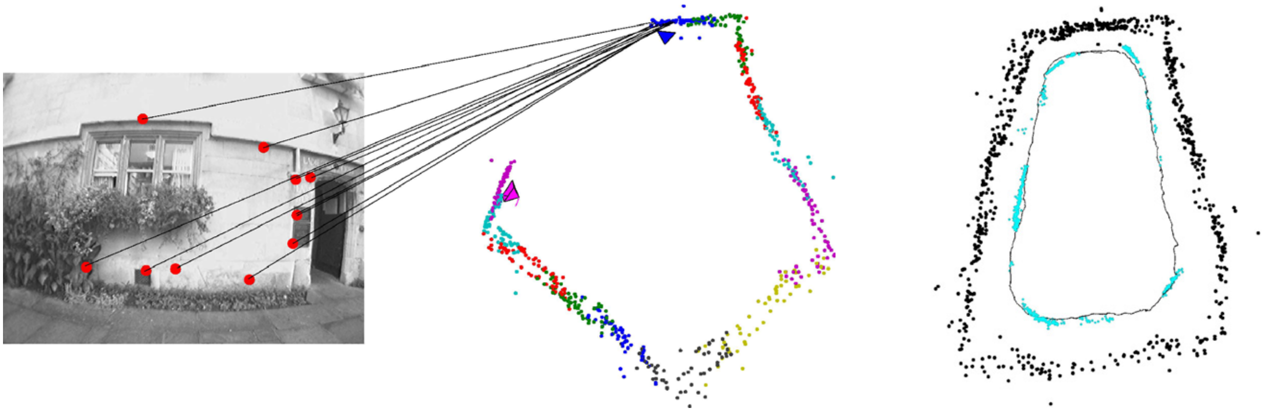
Fig. 8. The performance of the image-to-image method in the Pembroke College sequence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

matrix test for geometric compatibility. This can be seen for the false match with highest probability in Fig. 11(b). With this test, the performance of the image-to-image method would be equivalent to the image-to-map method. The fact that the image-to-image method does not rely on a good metric map existing can also be an advantage. This makes the system more flexible and allows it to work in a greater variety of applications. Finally, the method scales well to very large environments and has been shown to work on datasets of several kilometers [18].

The best performance here was found using the image-to-map matching technique of Williams et al. with a true positive rate of 20% for the Pembroke College sequence. The image-to-map method is able to prune more false positives than the image-to-image method by making use of the geometry information of the features detected in the image since these features have already been mapped by the SLAM system. It is fast enough to detect loop closure while tracking but it requires a large amount of memory for the randomised fern classifier (1.25 MB per map feature).

Table 1
A summary of the pros and cons of each of the three loop closure detection methods.

Method	Pros	Cons
Map-to-Map	Finds alignment when common features exist.	Sufficient common features are unlikely. False positives have similar number of ‘common’ features.
Image-to-Image	Detects true loop closures throughout the environment when tuned for 100% precision. Does not require metric map. Scales well to large environments.	Offline learning of good vocabulary required. Does not make use of geometric information. Does not give metric transformation directly.
Image-to-Map	Detects true loop closures throughout the environment when tuned for 100% precision. Online training for map feature appearance. Relative transformation between submaps with scale is computed from trajectory.	Requires good metric map. Very memory intensive.



(a) **Image-to-map matching:** Loop closure detected using the method of Williams et al. [3]. While tracking in the last submap, the system finds a camera pose consistent with the features in the first submap. The common trajectory is used to determine the relative rotation translation and scale needed to align the submaps.

(b) **Image-to-map reliability:** Relocalisation was attempted on every frame of a second lap. The light dots show the camera poses recovered relative to the map and trajectory created on the first lap (black). This indicates that loop close would be successful for these frames. Successful in 20% of frames. No false positives.

Fig. 9. The performance of the image-to-map method in the Pembroke College sequence.

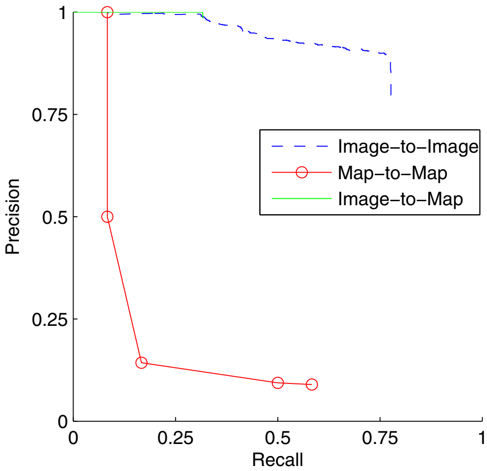


Fig. 10. The performance of the three loop closure detection methods was assessed by testing the ability to recognise loop closure at each position of the second lap of the Pembroke College sequence. The map-to-map method suffers due to the lack of common features in submaps built of the same region of the courtyard. The image-to-map and image-to-image method both perform very well but the image-to-image method has slightly more false positives. Many of these could be ruled out with geometric constraints in the visual word correspondences.

This limits its use to environments not much larger than those considered here. It is this classifier though that the system relies on for fast training and recognition of the map features in each image.

Perhaps even better performance could be achieved through a hybrid system combining the benefits of the image-to-image and the image-to-map methods. The loop closure detection system developed by Eade and Drummond [19] does just this. They first use a bag of visual words approach to establish which submap is in view. This stage is similar to the image-to-image method tested in this paper. Then, local landmarks are identified in the image and the camera pose relative to the landmarks is determined in a similar way to the image-to-map method. This global to local approach harnesses the strengths of each method.

6. Conclusion

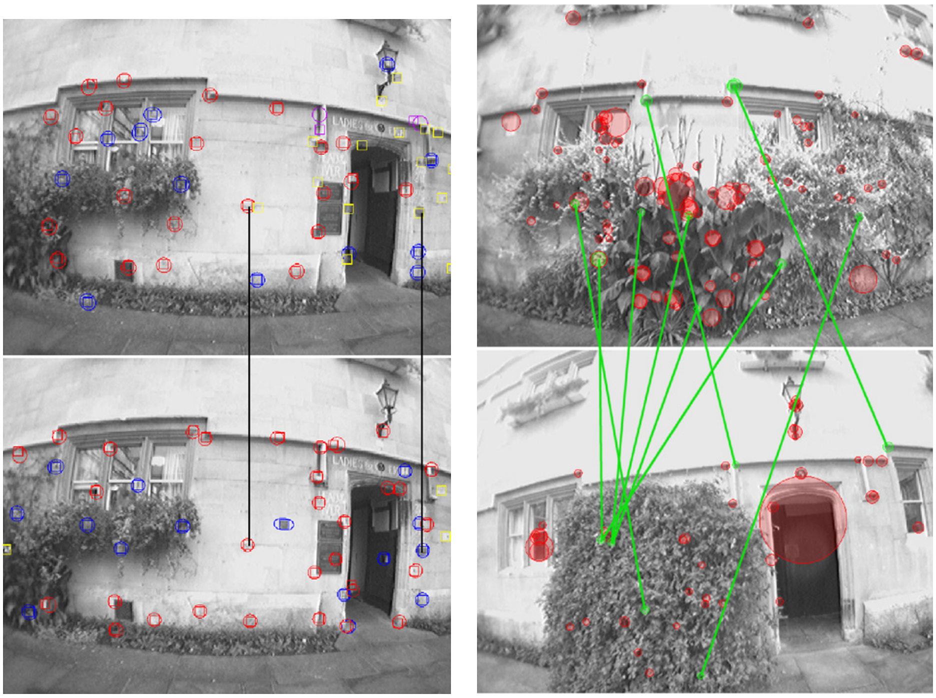
We have tested three quite different approaches to detecting loop closure for monocular SLAM systems. Experiments were performed in both indoor and outdoor environments using the Hierarchcal SLAM technique to build a sequence of submaps (Table 1).

We found the map-to-map matching technique to be unsuitable for monocular SLAM because the sparse maps contain too little information to reliably detect true correspondences while ruling out false ones.

The image-to-image method was shown to work well. However, to use this method for correcting a metric map, the relative pose and scale between corresponding images would need to be determined. The method would benefit from making some use of the relative positions of the detected visual words to remove some obvious false positives.

The image-to-map method works well and returned the highest number of true positives with no false positives. This is achieved by taking as much information as possible into account when detecting the loop closures. Unfortunately, this method does not scale well to larger environments like the image-to-image method.

Please cite this article in press as: B. Williams, et al., A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.06.010



(a) **Map-to-map method:** False negative in the Pembroke College sequence. Shown are features in the first and last submap for the same frame. Only two of the features are actually common to both submaps making it impossible for the map-to-map method to detect the loop close. Two maps made of the same part of the world do not necessarily have any common features.

(b) **Image-to-image method:** False positive with matching probability of 99.9935% in the Pembroke College sequence. The detected visual words are indicated in each image with light (green) circles if they match the other image and dark (red) circles if they do not. This false positive could easily be discarded if the geometric information were used for the detected visual words.

Fig. 11. Failure modes for the loop closure detection systems. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Acknowledgements

We gratefully acknowledge the financial support of the EPSRC (grant GR/T24685, EP/D037077, and a studentship to BW), the Royal Society (International Joint Project), the European Union (Project RAWSEEDS FP6-IST-045144), and the Direccion General de Investigacion of Spain (Project DPI2006-13578).

References

[1] L. Clemente, A. Davison, I. Reid, J. Neira, J.D. Tardós, Mapping large loops with a single hand-held camera, in: Proc. Robotics Science and Systems, 2007.

[2] M. Cummins, P. Newman, FAB-MAP: Probabilistic localization and mapping in the space of appearance, The International Journal of Robotics Research 27 (6) (2008) 647–665.

[3] B. Williams, G. Klein, I. Reid, Real-time SLAM relocalisation, in: Proc. International Conference on Computer Vision, 2007.

[4] M. Cummins, P. Newman, Accelerated appearance-only SLAM, in: Proc. IEEE International Conference on Robotics and Automation, 2008.

[5] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, J.D. Tardós, An image-to-map loop closing method for monocular SLAM, in: Proc. IEEE International Conference on Intelligent Robots and Systems, 2008.

[6] A.J. Davison, Real-time simultaneous localisation and mapping with a single camera, in: Proc. IEEE International Conference on Computer Vision, 2003.

[7] A. Davison, I. Reid, N. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Transaction on Pattern Analysis and Machine Intelligence (2007).

[8] J.M.M. Montiel, J. Civera, A.J. Davison, Unified inverse depth parametrization for monocular SLAM, in: Proc. Robotics Science and Systems, 2006.

[9] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Transactions on Robotics and Automation (2001) 890–897.

[10] C. Estrada, J. Neira, J.D. Tardós, Hierarchical SLAM: Real-time accurate mapping of large environments, Transactions on Robotics 1 (4) (2005).

[11] M. Bosse, P. Newman, J. Leonard, S. Teller, SLAM in large-scale cyclic environments using the atlas framework, The International Journal of Robotics Research 23 (12) (2004) 1113–1139.

[12] J. Neira, J.D. Tardós, J.A. Castellanos, Linear time vehicle relocation in SLAM, in: Proc. International Conference on Robotics and Automation, 2003.

[13] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: Proc. IEEE International Conference on Computer Vision, 2003.

[14] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded up robust features, in: Proc. European Conference on Computer Vision, 2006.

[15] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Second ed., Cambridge University Press, 2004.

[16] M.A. Fischler, R.C. Bolles, RANdom SAmple Consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[17] V. Lepetit, P. Fua, Keypoint recognition using randomized trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1465–1479.

[18] M. Cummins, P. Newman, Highly scalable appearance-only SLAM – FAB-MAP 2.0, in: Proc. Robotics Science and Systems, 2009.

[19] E. Eade, T. Drummond, Unified loop closing and recovery for real time monocular SLAM, in: Proc. European Conference on Computer Vision, 2008.

Brian Williams is a final year D.Phil student with the Active Vision Group in the Department of Engineering Science at the University of Oxford. His research focuses on real-time monocular SLAM using a handheld camera. He is particularly interested in relocalisation and loop closure detection for these systems.

Mark Cummins is a final year D.Phil student with the Mobile Robotics Group in the Department of Engineering Science at the University of Oxford. His research focuses on appearance-based navigation methods that infer position from visual appearance alone, without keeping track of metric position.

José Neira was born in Bogotá, Colombia, in 1963. He received the M.S. degree from the Universidad de los Andes, Bogotá, and the Ph.D. degree from the University of Zaragoza, Zaragoza, Spain, in 1986 and 1993, respectively, both in computer science. He is currently an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza, where he teaches compiler theory, computer vision, and mobile robotics. His current research interests include autonomous robots, data association, and environment modelling.

Paul Newman is a Reader in Engineering Science at the University of Oxford where he heads up the Mobile Robotics Group (MRG). He is also a tutorial fellow in Engineering at New College. Before moving to Oxford in 2003 he was a research scientist at MIT. He was the organiser and editor of the 'Robotics and Cognition' Foresight Cognitive Systems Project Research Review. He is an editor of the International Journal of Robotics Research and the Journal of Field Robotics. He is currently a IEEE Robotics and Automation Society Distinguished Lecturer for Europe.

Ian Reid is a Reader in Engineering Science and Fellow of Exeter College, at the University of Oxford where he jointly heads the Active Vision Group. His research has touched an many aspects of computer vision, concentrating on algorithms for visual tracking, control of active head/eye robotic platforms (for surveillance and navigation), SLAM, visual geometry, novel view synthesis and human motion capture. He serves on the editorial boards of Image and Vision Computing Journal and IPSJ Transactions on Computer Vision Applications.

Juan Tardós was born in Huesca, Spain, in 1961. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 1985 and 1991, respectively. He is currently a Full Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza, where he is in charge of courses in robotics, computer vision, and artificial intelligence. His current research interests include simultaneous localisation and mapping (SLAM) and perception and mobile robotics.

Please cite this article in press as: B. Williams, et al., A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.06.010