Robust Real-Time Visual SLAM Using Scale Prediction and Exemplar Based Feature Description

Denis Chekhlov, Mark Pupilli, Walterio Mayol and Andrew Calway
Department of Computer Science
University of Bristol, UK
{chekhlov,pupilli,wmayol,andrew}@cs.bris.ac.uk

Abstract

Two major limitations of real-time visual SLAM algorithms are the restricted range of views over which they can operate and their lack of robustness when faced with erratic camera motion or severe visual occlusion. In this paper we describe a visual SLAM algorithm which addresses both of these problems. The key component is a novel feature description method which is both fast and capable of repeatable correspondence matching over a wide range of viewing angles and scales. This is achieved in real-time by using a SIFT-like spatial gradient descriptor in conjunction with efficient scale prediction and exemplar based feature representation. Results are presented illustrating robust real-time SLAM operation within an office environment.

1. Introduction

Significant advances have been made in real-time estimation of the 3-D pose of a moving camera using vision in uncalibrated environments. It requires simultaneous estimation of a scene map, and is related to simultaneous localisation and mapping (SLAM) in robotics [3]. Applications are in areas such as Wearable Computing, in which the camera is either hand-held or attached to a user. This makes it a challenging task using vision alone. It is best tackled by stochastic filtering, and pioneering work was done by Davison [2], using Kalman filtering and efficient feature matching. Eade and Drummond [4] recently demonstrated a comparable system using the FastSLAM algorithm [12]. Both approaches give impressive real-time performance, albeit over restricted areas with smooth motions and minimal visual occlusion.

Unfortunately, these restrictions are a major stumbling block to use in real applications. A central difficulty is the reliance on simple template matching for feature correspondence in order to achieve real-time operation. Under smooth motions, filter predictions of feature locations are reliable and hence matching ambiguity can be minimised. However, when faced with erratic motion, such as camera shake, or significant occlusion, the lack of discrimination leads to mismatch and filter instability. Similarly, as features are viewed from wider angles, surrounding regions deviate from the templates and matching becomes unreliable, again resulting in failure. Matching ambiguity is addressed to some extent by Pupilli and Calway [13] using a particle filter, although the approach lacks full covariance. Wide angled viewing can be accounted for by warping templates based on estimated or assumed surface normals, as in [11, 4], but this has limitations and also fails to address the ambiguity problem. Of course, recovery from tracking failure can always be achieved by relocalising the camera using an auxiliary process, as demonstrated in [15] for example, but this needs to be put off for as long as possible to avoid repeated re-initialisation and hence reduced performance.

A more desirable approach is to seek greater discrimination in feature matching so as to improve robustness when filter uncertainty increases. Techniques such as the Scale-Invariant Feature Transform (SIFT) [9] and maximally stable regions [10], for example, have been shown to give reliable feature matching over a wide range of viewing angles and scales. However these methods were designed primarily for off-line matching and object recognition, and aim for full invariance to compensate for the lack of view information. This makes them inefficient for SLAM systems, in which estimates of camera position and direction are available. This is exploited by Chekhlov *et al*. [1], who demonstrate that scale predictions from the SLAM filter can be used to increase the efficiency of feature matching using descriptors similar to that used in the SIFT. The resulting algorithm demonstrates significant performance gains over that previously achieved, including the ability to recover SLAM operation following camera shake and occlusion. What this work failed to address, however, was the issue of wide angle viewing of features, and this proves to be a limitation when seeking to extend operation over wider physical areas.

1-4244-1180-7/07/$25.00 ©2007 IEEE

In this paper we tackle this problem by utilising an exemplar based representation of feature regions (corresponding to affine transformations of the initialisation region), which when combined with the descriptor gives increased wide angle matching. However, it turns out that for real-time operation, this can only be achieved by adopting a different scale prediction strategy to that used in [1]. This results in an alternative formulation and initial results suggest that robustness is significantly increased. The paper is organised as follows. In the next section we briefly outline the SLAM system, followed by a detailed description of the feature matching process, which is the main contribution of the paper. Results are then presented showing successful real-time SLAM operation in an office environment, including withstanding camera shake, occlusion and wide angled viewing.

2. Visual SLAM Using Stochastic Filtering

We use a stochastic filtering framework in a similar manner to that in [2, 4, 1], based around an unscented Kalman filter (UKF) [7], primarily for ease of implementation. The system has the usual predictor-corrector structure as illustrated in Fig 1. As each video frame is processed, we aim to estimate the current 3-D camera pose and update estimates of the 3-D position of scene points, all with respect to a known world coordinate frame. The filter state therefore has the form $\mathbf{x} = (\mathbf{v}, \mathbf{z})$, where $\mathbf{v} = (\mathbf{q}, \mathbf{t})$ encodes the camera pose via the quaternion $\mathbf{q}$, representing the orientation, and the position vector $\mathbf{t}$, and $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)$ denotes the 3-D position vectors for M scene points. During SLAM operation, the number of points will change, as points are added to and removed from the map, and thus the filter state dimension needs to be variable.

The filter requires a process model and an observation model [7], defining the state dynamics and the relationship between the state and measurements taken from the video frames. We assume a constant position motion model for the camera, *i.e.* a random walk, which gives greater robustness to erratic motion [13], and we also assume that the scene is rigid. This leads to the following process model

$$f(\mathbf{x}, \mathbf{n}) = (\mathbf{t} + \mathbf{n}_\tau, \Delta\mathbf{q}(\mathbf{n}_\omega) \otimes \mathbf{q}, \mathbf{z}) \qquad (1)$$

where $\mathbf{n} = (\mathbf{n}_\tau, \mathbf{n}_\omega)$ is a 6-D noise vector, assumed to be from $\mathbf{N}(0, Q_n)$, $\Delta\mathbf{q}(\mathbf{n}_\omega)$ is the incremental quaternion corresponding to the Euler angles defined by $\mathbf{n}_\omega$, and $\otimes$ denotes quaternion multiplication. Note the non-additive noise component within the quaternion part, which is necessary to give an unbiased distribution in rotation space [1].

The filter observations are assumed to be corrupted positions of projected scene points within the current video frame. For a given scene point, its projection corresponds to a transformation into the camera co-ordinate system

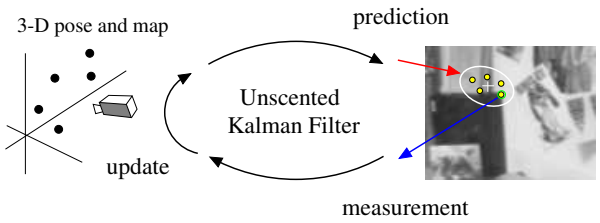


Figure 1. Stochastic filtering for visual SLAM, illustrating predictor-corrector operation in which measurements found within predicted search regions are used to update camera pose and scene map estimates.

followed by perspective projection onto the image plane. Thus, for the mth point, this gives a 2-D image point $\mathbf{u}(\mathbf{z}_m, \mathbf{v}) = \Pi(R(\mathbf{q})(\mathbf{z}_m - \mathbf{t}))$, where $R(\mathbf{q})$ denotes the rotation matrix corresponding to the normalised quaternion $\mathbf{q}$ and $\Pi()$ denotes standard pin-hole projection for a calibrated camera. The observation model is then the concatenation of the projected points with additive noise:

$$h(\mathbf{x}, \mathbf{w}) = (\mathbf{u}(\mathbf{z}_1, \mathbf{v}) + \mathbf{w}_1, \ldots, \mathbf{u}(\mathbf{z}_M, \mathbf{v}) + \mathbf{w}_M) \qquad (2)$$

where the multivariate noise vector $\mathbf{w}$ is from $\mathbf{N}(0, R_w)$.

The filter provides successive estimates of the mean state and its covariance. As illustrated in Fig. 1, an iteration involves generating a mean and covariance prediction for the next state via the process model in (1), collecting measurements of feature positions from the current video frame, and then updating the mean and covariances using the Kalman equations based on the observation model in (2). Both the process and observation models are non-linear and thus we require an approximation to the KF, hence our use of the unscented KF. Crucially, the mean and covariance predictions from the filter are used to constrain the search for features as illustrated in Fig. 1. This utilises the global structure built up within the scene map and can be implemented using the unscented transform [7]. To minimise computation, candidate feature points within search regions, indicated in yellow in Fig. 1, are detected using a fast salient point operator (we use that proposed by Rosten and Drummond [14]), and the most likely feature positions are then found by comparison with feature measurements taken around each scene point at initialisation. This comparison is critical for successful operation of the filter and it is the main concern of this paper. We return to it in the next section.

There are, however, several additional components that need to be in place for real-time SLAM operation. We briefly describe these here; readers are referred to [2, 4, 13] for more details. To bootstrap the filter we position the camera parallel to a test pattern in the scene, with known feature points, and this sets the scale factor as well as providing an initial map with which to begin tracking the camera pose. As the camera moves away from the test pattern, new scene points are initialised into the map concurrent with

tracking. Potential points are selected from candidate 2-D salient points in unexplored regions and their 3-D position initialised using factored sampling along the corresponding projection ray in conjunction with feature matching in subsequent frames. As depth estimates converge, the new points are incorporated into the map, although care needs to be taken to ensure correct initialisation of covariances [2, 1]. As tracking proceeds the 3-D position estimates converge, hence enabling wider area operation. Points which fail to be matched over successive frames are pruned from the map in order to minimise computation. Both the initialisation of new points and the subsequent tracking of camera pose therefore depends critically on successful matching of features across frames and it is on this that we now concentrate.

3. Robust Feature Matching for Visual SLAM

Matching features across successive video frames is a challenging task to achieve in real-time, especially when operating in cluttered environments and over wide viewing angles, when perspective effects can become significant. Template matching, as used in [2, 4], is an attractive option for SLAM since it minimises image processing effort, which is essential for real-time operation. When search regions are small, which reduces the chances of mismatch, the approach can be effective, although as viewing angles increase matching will become problematic. The latter can be addressed by warping templates, as in [11, 4] for example, but this has limitations. A greater difficulty occurs when the camera pose becomes uncertain, due to sudden erratic motion or occlusion for example, and search regions correspondingly increase, resulting in widespread mis-match due to the lack of discrimination inherent in template matching and leading to tracking failure (an example of this is shown in Fig. 6). Greater discrimination in feature matching is therefore required if increased robustness is to be obtained.

3.1. Scale Prediction in Feature Matching

Considerable work has been done in developing robust feature matching for off-line systems, in applications such as object recognition, and techniques such as the SIFT developed by Lowe [9], have demonstrated highly discriminate matching. However, these methods are very general, in that they assume that no a priori information is available about camera views, and hence attempt to build in sufficient invariance to compensate for this lack of prior knowledge. The situation in a SLAM system is different, in that estimates of the camera pose are available, and thus the use of a 'full invariance' descriptor is wasteful. Instead, the camera pose estimates can be used to reduce the need for full invariance in the descriptor and hence minimise computation and potentially increase robustness. This is the approach adopted in [1], where the estimates of camera position are used to predict the changes in scale between features in different frames. This in turn is used to compute spatial gradient descriptors, similar to those used in the SIFT, at scales which compensate for the change in camera position.

The matching algorithm operates as follows. When map points are initialised, descriptors are built at multiple scales. This is a one off overhead which is done concurrently with SLAM operation. In subsequent frames, descriptors at potential corresponding points are generated at frame resolution and the estimated change in camera position obtained from the filter is used to predict which of the original descriptors they should be compared with. The uncertainty in camera position as indicated by the estimated covariance is also used to define a range of scale descriptors on which to base the comparison. Thus, as camera position becomes uncertain, the range of descriptors tested widens, hence increasing the likelihood of determining a correct match. This is particularly significant in relocating the camera should tracking be interrupted. Equally important is that when tracking is consistent, then computational efficiency is increased by reducing the range of potential candidates that need to be tested. The resulting algorithm was shown to give robust performance, capable of recovering SLAM operation even after severe camera shake or total visual occlusion.

3.2. Exemplar Based Feature Matching

A limitation of the above approach however is that the spatial gradient descriptors have restricted view angle invariance. As the camera moves away from where points are initialised and features are viewed from increasing angles, matching becomes less reliable and tracking stability is reduced. It is this issue that we address in this paper. To do so, we employ an exemplar approach in which a given feature point is represented by multiple descriptors corresponding to a set of affine transformations of the region surrounding the point. The motivation here is that we approximate the change in appearance of a feature by an affine transformation and that the resulting set of descriptors will populate the area of 'descriptor space' corresponding to different viewing angles. A similar approach to modelling changes in feature appearance was adopted by Lepetit and Fua [8].

A difficulty with this approach, however, is that the number of descriptors that need to be generated at initialisation becomes large if multiple scales are used as in [1]. To overcome this we employ an alternative strategy to compensate for scale changes. At initialisation, descriptors are generated for affine transformations of the region surrounding a feature. In subsequent frames, descriptors are generated at scales determined from the estimates of camera position, with descriptors at multiple scales being generated when the

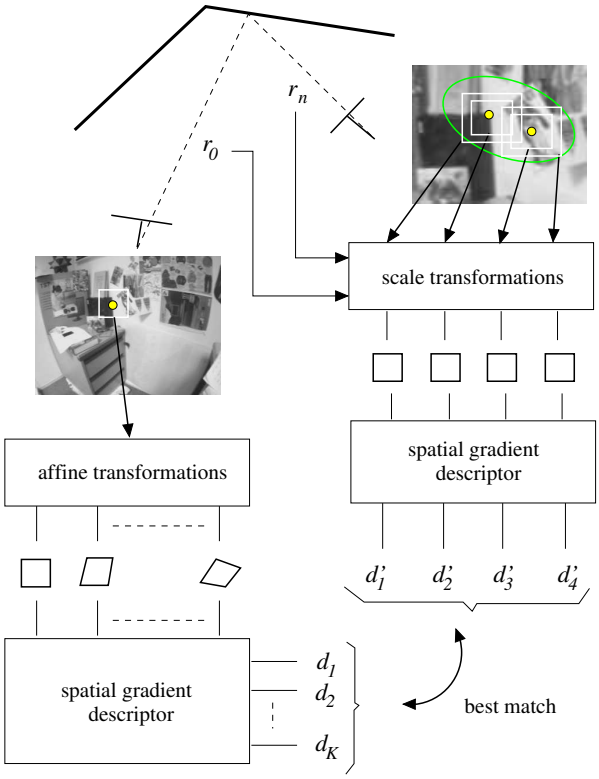
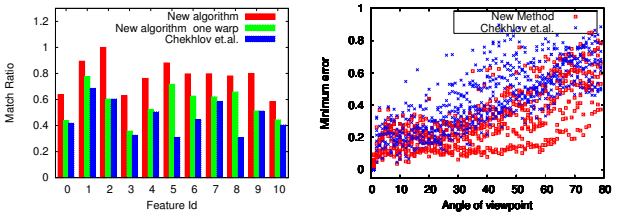


Figure 2. Feature matching using spatial gradient descriptors, affine exemplars and scale prediction.

camera position becomes uncertain. This is the reverse of the strategy adopted in [1], in that scale is now accounted for in the current frame rather than in the initialisation frame. But in doing so, it frees up time at initialisation to generate descriptors for the set of affine warps, and allows real-time operation.

Figure 2 illustrates the feature matching algorithm. Having identified a 2-D point for initialisation, we generate a set of affine transformations of the surrounding region. We exclude pure 2-D rotations since these are accommodated for by compensating for dominant orientation when generating the descriptors [9]. This leaves a 3-D parameterisation for the transformations of the form $A = R_\theta^{-1} S R_\theta$, where R_θ denotes a 2-D rotation by angle θ and $S = \text{diag}[s_1, s_2]$ defines the scaling along each dimension [8]. In the experiments we sampled this parameterisation in order to give around 60 affine warps per feature. Spatial gradient descriptors, compromising of sets of orientation histograms [9], are then generated for each affine 'patch', giving the set of descriptors $d_1 \ldots d_K$. In the experiments we used patch sizes of 22×22 and 4×4 histograms, giving descriptors with 128 elements.

In order to match the same feature in later frames, descriptors at one or more scales are generated about candidate points. The scales are determined by the change in



(a) (b)

Figure 3. Synthetic planar sequence comparison between the new method and that in [1] : (a) percentage of correct matches for each map point; (b) minimum descriptor error over all features against viewing angle.

camera position as indicated by the relative change in depth from the 3-D map point concerned (r_0 and r_n in Fig. 2). It is important to build in the uncertainty in the estimated parameters from the filter to ensure robust matching, especially when tracking is interrupted and loss of position occurs. This can be done efficiently by using the unscented transform to compute covariance values for the relative depth of features and use this to determine a range of scales over which to compute descriptors for a given point in the current frame. Thus, in Fig. 2 descriptors at two scales are being generated for each candidate point in frame n. The matched point is then selected as that associated with the descriptor having the minimum euclidean distance, below a given threshold, to one of the exemplar descriptors generated at initialisation.

4. Results

We tested the new algorithm by performing SLAM in an office environment using a calibrated hand-held webcam with a resolution of 320×240 pixels. Tracking was initialised with four known map points corresponding to the corners of a planar black rectangle on a white background placed in the scene. Performance assessment commenced once the test pattern became out of view. We also compared performance with feature matching based on normalised correlation and the previous method in [1]. For all experiments SLAM operation was in real-time, typically around 20 fps with around 25 map features. It should be noted, however, that our use of the UKF is not optimal (it has complexity of order N^3) and that speed up would be achieved when using either an EKF or a fastSLAM implementation. Our interest here has been in gaining more robust feature matching whilst maintaining similar processing time, rather than on optimising overall SLAM speed. Video results can be found at `www.cs.bris.ac.uk/Research/Vision/Realtime/`.

4.1. Performance against view point changes

In the first experiment we compare the matching capabilities of the new algorithm with that in [1]. For this we generated a synthetic sequence consisting of the camera moving in front of a texture mapped planar surface. The camera was rotating about a fixed origin on the plane, enabling matching performance to be recorded against viewing angle. To ensure fair comparison, the same feature points were initialised into the scene map for both methods. Figure 3a shows the percentage of correct matches for each feature all frames for both methods and for the case when only a single descriptor was generated at initialisation in the new method, *i.e*. without exemplars. The latter is similar to the method in [1] except that the scale compensation is reversed. Note that matching performance is significantly better when exemplars are included. This is confirmed in Fig. 3b, which shows the best matching descriptor error for each feature against viewing angle. Whereas errors begin to increase significantly at angles around 40 degrees for the method in [1], errors remain low up to around 60 degrees for the new method.

4.2. Feature Matching Performance

We observed similar gains in performance with live SLAM operation. As an illustration, Fig. 4 shows views through the camera with projected map points superimposed for the three different methods. The ellipses indicate search regions derived from the filter, green (light grey) indicates a match and red (dark grey) indicates a mis-match. When the camera is positioned roughly fronto-parallel to initialised features, as in the left column, all three methods give good matches. However, as the camera pans around, it is only the new method with exemplars that can maintain good matching performance. This is confirmed in Fig. 5a which shows the average percentage of matches over all frames for the new method (red/dark line) and the method in [1], averaged over 20 runs. The significant difference is between frames 200 and 800, when there was significant changes in viewing angle. Between frames 900 and 1500 there were bouts of camera shake and occlusion (indicated by sudden drops in matches) but viewing angle was similar to that at initialisation and so both methods give comparable performance.

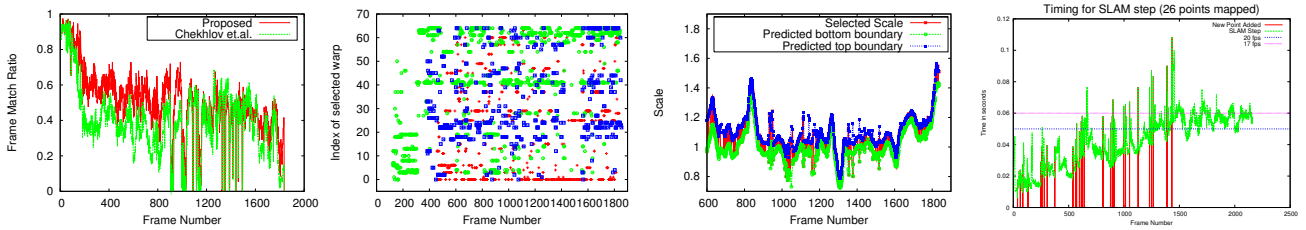
To illustrate the key components of the new method, Figs 5b and 5c show the exemplars selected as the best match over all frames for three features and the changes in scale compensation over all frames for one feature, respectively. Note the wide range of exemplars used, indicating the compensation for viewing angle, and the changes in scale range, particularly during bouts of shake and occlusion. In the latter, the blue and green lines indicate the upper and lower levels of the scale range. Figure 5d shows processing times

Figure 4. Feature matching performance during SLAM, comparison between the new method and that in [1] : (top) new method; (middle) new method without exemplars; (bottom) method in [1].

per frame for one run in which 26 features were mapped, with steady state operation at around 17-20 fps. As noted ealier, we would anticipate that increased frame rate would be achieved using an EKF implementation.

4.3. Erratic Motion and Occlusion

In the final experiments we illustrated the ability of the new method to recover following bouts of camera shake and visual occlusion. As noted earlier, robustness to such unpredictable camera motion is essential if visual SLAM algorithms are to be used in real applications and this aspect of performance is one of the main motivations for the work described here. It is also worth emphasising that it is clearly unrealistic to aim to maintain tracking during all forms of erratic motion; sufficiently severe changes in camera position, especially when combined with visual occlusion, can always break SLAM operation. However, if repeated re-initialisation and hence reduced performance is to be avoided, then sufficient robustness needs to be built in to withstand the types of erratic movement that may occur during 'normal use' in real applications. This includes a degree of camera shake, temporary occlusion and perhaps combinations of the two. The algorithm presented here does manage to recover from such episodes, as illustrated in Figures 6 and 7. In the former we have compared performance with a standard visual SLAM algorithm using template matching in the form of normalised correlation, similar to that used



(a) (b) (c) (d)

Figure 5. Feature matching performance during SLAM. (a)-(c) Comparison between the new method and that in [1]: (a) average percentage of matches for both methods over all frames; (b) best matching warp for three features over all frames for new method; (c) variation in scale range for one feature for new method. (d) Processing time per frame during SLAM operation for mapping 26 features (vertical lines indicate initialisation of new features).

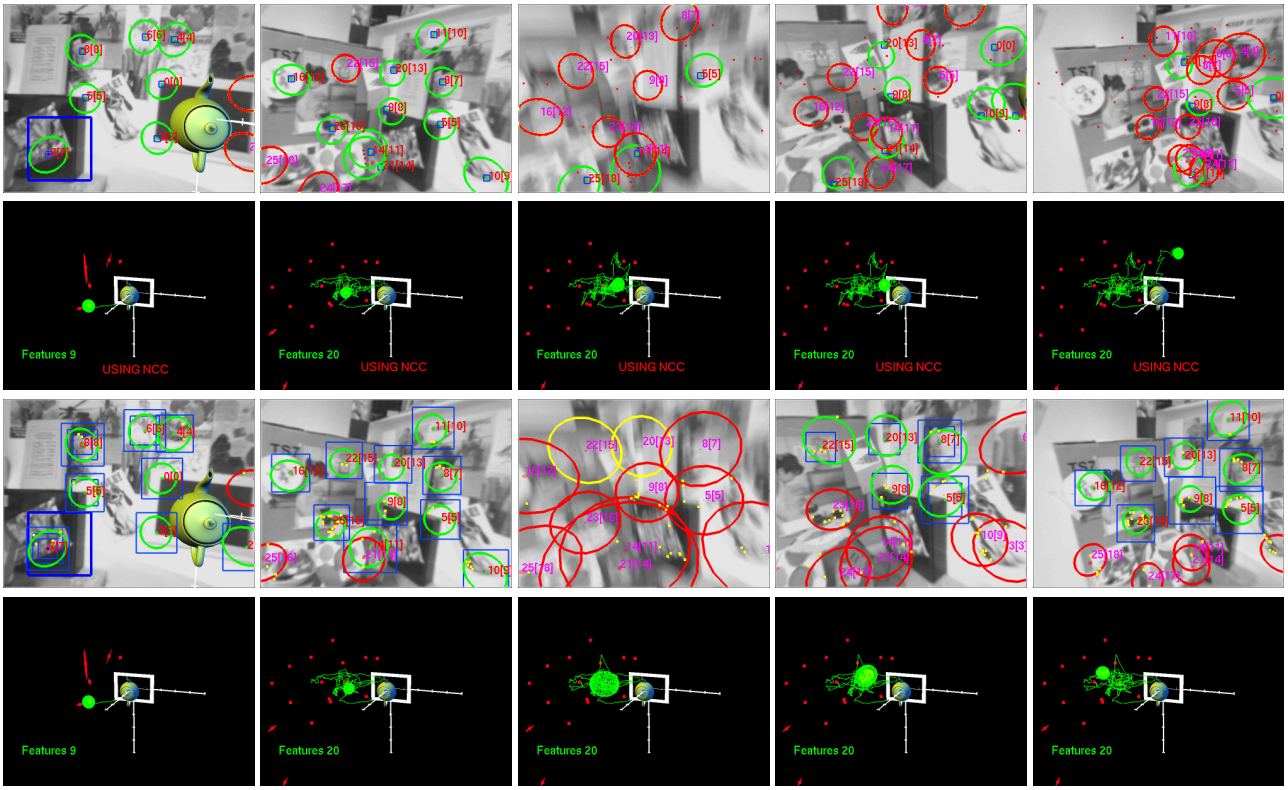


Figure 6. Performance comparison of the new method (bottom two rows) with that based on normalised cross correlation matching (top two rows). Note the successful recovery of the new method following camera shake and the failure of the correlation system.

in [2]. The figures show both the view through the camera with projected map points and associated search regions and an external 3-D view showing the estimated camera position, its trajectory and the associated position covariance, indicated by an ellipsoid.

In Fig. 6 the frames show SLAM operation before during and after fairly severe camera shake. The main thing to note is that the template matching approach (top two rows) fails completely after the shake, whilst the new method successfully recovers. Note also that during shake the covariance estimate obtained in the new method increases significantly, resulting in large search regions, and that the descriptors enable successful matching over such regions when shaking ceases. In contrast, the covariance estimate in the template matching version does not grow so much due to false matches obtained, due to the lack of discrimination, which further prevents relocation once shaking has ceased. A particularly impressive example is shown in Fig. 7, where severe visual occlusion is combined with movement of the camera to a different viewing point. Despite

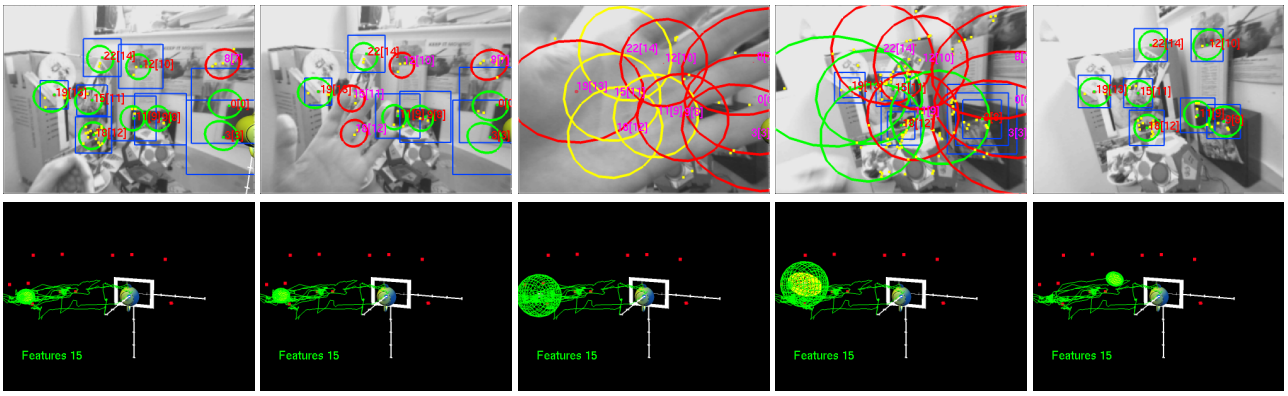


Figure 7. Frames illustrating the performance of the new method during SLAM operation showing recovery from severe visual occlusion during which the camera undergoes a large change in position.

this the new method successfully recovers. The template matching method failed completely in this example.

5. Conclusions

We have presented a new method for visual SLAM which demonstrates improved performance over existing methods. The use of affine exemplars combined with scale prediction and feature description gives robust matching, even allowing recovery of operation following significant camera shake and visual occlusion. The demonstrated improvement in matching capability over that in [1] would suggest that this new approach is to be preferred. Current work is focused on improving the wide area operation of the system, particularly in feature management, and we aim to extend the method into areas such as the kidnapped camera problem.

Acknowledgements

This work was funded by ORSAS UK and the EPSRC Equator IRC.

References

[1] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway. Real-time and robust monocular slam using predictive multi-resolution descriptors. In *2nd Int Symp on Visual Computing*, 2006.

[2] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. Int Conf on Computer Vision*, 2003.

[3] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Trans on Robotics and Automation*, 17(3):229–241, 2001.

[4] E. Eade and T. Drummond. Scalable monocular slam. In *Proc. Int Conf on Computer Vision and Pattern Recognition*, 2006.

[5] I. Gordon and D. Lowe. Scene modelling, recognition and tracking with invariant image features. In *Int Symp on Mixed and Augmented Reality*, 2004.

[6] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. IEEE Int Conf on Robotics and Automation*, 2006.

[7] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proc. Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.

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

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

[10] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc British Machine Vision Conf*, 2002.

[11] N. Molton, I. Ried, and A. Davison. Locally planar patch features for real-time structure from motion. In *Proc. British Machine Vision Conf*, 2004.

[12] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc Int Joint Conf on Artifi cial Intelligence*, 2003.

[13] M. Pupilli and A. Calway. Real-time visual slam with resilience to erratic motion. In *Proc. Int Conf on Computer Vision and Pattern Recognition*, 2006.

[14] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. European Conf on Computer Vision*, 2006.

[15] B. Williams, P. Smith, and I. Reid. Automatic relocalisation for a single-camera simultaneous localisation and mapping system. In *Proc. Int Conf on Robotics and Automation*, 2007.