Robust Single Camera Relocalisation in Large-Scale Environments

Stephen J. Thomas and Bruce A. MacDonald and Karl A. Stol

***Abstract*—We present a fast method for single camera relocalisation that scales to very large environments. The relocaliser is integrated into a parallel tracking and mapping framework in which a map consisting of 3D points with visual descriptors is constructed online. When tracking fails the relocaliser is invoked and employs state-of-the-art visual feature description and matching techniques to efficiently estimate the camera's pose with respect to the map so that tracking can resume reliably. We demonstrate that the proposed algorithms are robust, capable of real-time operation, and scale to maps which are more than an order of magnitude larger than the current state-of-the-art.**

I. Introduction

Real-time camera pose tracking can be used to estimate the 6-DOF pose of a camera relative to a map or model of the surrounding environment. This map or model may be known in advance, or may be generated on the fly, a problem commonly known as Simultaneous Localisation and Mapping (SLAM), Structure from Motion (SfM), or Parallel Tracking and Mapping (PTAM). Real-time vision-based tracking systems generally require a prior on the camera pose to limit the search space for feature correspondences and achieve high framerates. Furthermore, vision-based tracking requires a sufficient number of distinctive features to be visible in order to reliably establish the feature correspondences from which the camera pose is derived. Rapid camera motion, occlusion, motion blur, featureless surfaces and repetitive patterns frequently produce limited or erroneous correspondences and consequently tracking becomes unstable or fails completely. In the absence of a pose prior from other sensors the camera is now completely lost and the search-space for feature correspondences can no longer be constrained. Consequently, *camera relocalisation* is considerably more computationally expensive than frame-to-frame pose tracking. In order to retain the high-framerates achieved by camera pose tracking, we propose a separate camera relocalisation method which is only invoked when tracking becomes unstable or fails.

II. Related Work

A. Image-to-Map Feature Correspondences and Random Sampling Consensus (RANSAC)

Chekhlov et. al. [1] proposed a camera relocalisation method in which the search-space for feature correspondences was reduced by introducing fast indexing based on appearance. Low order Haar wavelet coefficients are used to provide a coarse estimate of spatial gradients around a feature point, and these coefficients are quickly compared to eliminate feature pairings which are highly unlikely to yield a true correspondence. The set of potential correspondences is then refined using the well known Scale-Invariant Feature Transform (SIFT) [2] descriptor. A camera pose estimate is derived from the final set of correspondences using the RANSAC formulation by Fischler and Bolles [3], which searches for a consensus set and the associated camera pose by testing multiple hypotheses using the 3-point absolute pose algorithm. Chekhlov et. al. [1] demonstrated the robustness of the relocalisation on a number of short sequences with success rates ranging between 60-97% and incorrect relocalisation rates below 2.2%. No indication of the frame-rate or execution time is given, however the number of full descriptor comparisons was reduced to 5% compared to exhaustive matching. However, the computational cost of computing and matching SIFT descriptors is considerable, and the matching time still grows linearly with map size.

Williams et. al. [4] proposed a relocaliser based on online learning of patch appearance. The system is based on the feature matching approach of Lepetit and Fua [5] which treats real-time feature recognition as a classification problem. Each time new features are added to the map, the relocaliser of Williams et. al. [4] trains 10-20 new randomised tree classifiers to identify these features. When tracking fails the classifiers are used on each new frame to generate a list of potential correspondences. A camera pose estimate is derived from these correspondences using the RANSAC formulation of Fischler and Bolles [3]. The system was tested on a dual-core 2.7GHz CPU using a map with 54 features, from which the classifier returned 7 inliers and 35 outliers, and the relocaliser produced a pose estimate in 19ms. The approach was only demonstrated on maps with up to 80 features, and does not scale well due to the high training times and memory requirement [4].

Guan et. al. [6] proposed a camera relocalisation method which combined fast feature description using Compact Signatures [7] with camera pose estimation and outlier rejection using the Progressive Sample Consensus (PROSAC) algorithm [8]. Compact Signature descriptors can be computed very efficiently, thus the major bottleneck of the system was feature matching. Matching was accelerated by comparing descriptors using the sum of absolute differences (SAD) on a graphics processing unit (GPU). The resulting set of correspondences generally had a very high proportion of outliers and consequently the traditional three-point RANSAC algorithm required a large number of iterations to obtain a camera pose estimate. Guan et. al. [6] proposed to address this problem by using the PROSAC algorithm which biases the random correspondence selection to firstly pick correspondences which are more likely to be inliers based on their feature similarity scores. In this way the algorithm may arrive at a solution which satisfies the termination criterion earlier. The system was tested using a quad-core Xeon 2.66 GHz CPU, NVIDIA GeForce GTX260 GPU and 640 × 480 pixel webcam. The maps used for the experiments contained up to 4,215 map points, and the relocaliser required 49ms per frame on average to determine a camera pose estimate.

B. Keyframe/Node Appearance Matching

In the PTAM framework of Klein and Murray [9] the map consists of a small number of *keyframes* in addition to triangulated 3D map points. Rather than using the 3D map points for camera relocalisation the keyframes are assigned a single descriptor which enables similar keyframes to be identified directly. When tracking fails the system attempts to resume from the position of the keyframe with the most similar appearance. The keyframe appearance descriptor is a *Small Blurry Image (SBI)* which is obtained by downscaling the keyframe image and convolving the result with a zero mean Gaussian. The similarity between these descriptors is evaluated using normalised cross-correlation (NCC). Experimental results indicate that the method can deal with several hundred keyframes in real-time. However, the approach has a high rate of incorrect relocalisation and the descriptor is only robust to relatively small changes in position and viewpoint.

Eade and Drummond [10] presented a method for relocalisation within a graph-based SLAM framework which uses both appearance and structure to guide a relocalisation search. The map in this framework consists of a small number of *nodes* (synonymous with keyframes) in addition to 3D map points. The first stage in the relocalisation process is to identify a set of candidate nodes that are most similar to the current frame based on a bag-of-words appearance model. The visual bag-of-words approach involves extracting feature descriptors from the image, quantising the descriptors according to a fixed *vocabulary* of visual words, and constructing a histogram of word frequencies to be used as the node descriptor. The vocabulary is built incrementally online using compact 16-D SIFT [2] descriptors which are less distinctive than the standard 128-D SIFT descriptors but much more efficient to compute, store, and compare. Using the bag-of-words model the nodes are ranked based on how well they express the current image, and the top three candidate nodes are selected. A nearest-neighbour search is then used to match the SIFT descriptors from the current image to each of the candidate node's associated 3D map point descriptors. Finally, Maximum Likelihood RANSAC (MLESAC) [11] is employed to find the maximum-likelihood set of correspondences and the associated camera pose. The system was tested on an indoor scene in which a complex external loop was traversed and closed. The camera was then repeatedly kidnapped from one part of the environment to another, with new viewpoints significantly different from the originals. The relocalisation algorithm required no more than 6ms per frame, and recovery always occurred within 15 frames for a 1402 feature map.

Each of these systems was evaluated using a different performance metric, making a comparison extremely difficult. Furthermore, relocalisation times and accuracies are highly dependent on the image features extracted, which in turn are dependent on scene structure. However, a common limitation of these systems is that they only scale to small environments (80 - 4,215 features) due to the high computational cost and combinatorial explosion of feature matching. We propose a novel relocaliser which scales to maps with tens to hundreds of thousands of features, while retaining the real-time performance necessary for practical applications.

III. A Novel Vision-based Relocaliser for Very Large Maps

The relocaliser proposed in this paper employs state-of-the-art approaches to feature description, feature matching and robust camera pose estimation to produce a novel relocaliser that significantly outperforms existing approaches. The vision-based system is capable of handling maps containing tens of thousands of features in real-time, and absolute position and attitude sensors can be used to constrain the relocalisation search-space providing scalability to even larger maps. The relocaliser has been integrated into the PTAM framework of Klein and Murray [9] which constructs a map consisting of keyframes and 3D map points in real-time. The system was developed for a micro aerial vehicle (MAV) with a downward pointing camera, thus the system must be capable of relocalisation in both indoor and outdoor environments from vastly different viewpoints and altitudes.

IV. Image and Map Point Feature Description

The PTAM algorithm finds correspondences between images by detecting keypoints within a four level image pyramid using the FAST-10 keypoint detector [12] and a threshold based on the Shi-Tomasi score [13]. Keypoints are then matched by comparing image patches which are pre-warped based on prior estimates of the camera poses. When tracking fails no prior pose estimate is available, thus the patch pre-warping which enables rotation and scale invariant matching is impossible. As demonstrated in [1], [10] rotation-invariant feature descriptors such as SIFT provide an excellent basis for establishing correspondences in the absence of a prior pose estimate. However, scale-invariant matching using SIFT-like descriptors requires the keypoint detector to provide an estimate of the keypoint scale. Lindeberg [14] introduced the concept of identifying points which are well-localised in scale by identifying maxima (blobs) in a normalised Laplacian scale-space. This is the approach adopted by the computationally expensive SIFT and SURF [15] detectors. The STAR keypoint detector from the OpenCV library [16] attempts to reduce the computational cost of this approach by approximating each level of the Laplacian scale-space with very few computations using the difference between rotated squares. However, many authors such as Chekhlov et. al. [1] opt to sacrifice scale-invariance in order to use significantly faster detectors such as FAST. Table I summarises the keypoint detection times for these detectors, and also a seven level image pyramid using the FAST-10 detector for the first 800 × 640 pixel image of the Graffiti dataset[1].

From Table I it is evident that the FAST based keypoint detectors are considerably faster, however we must

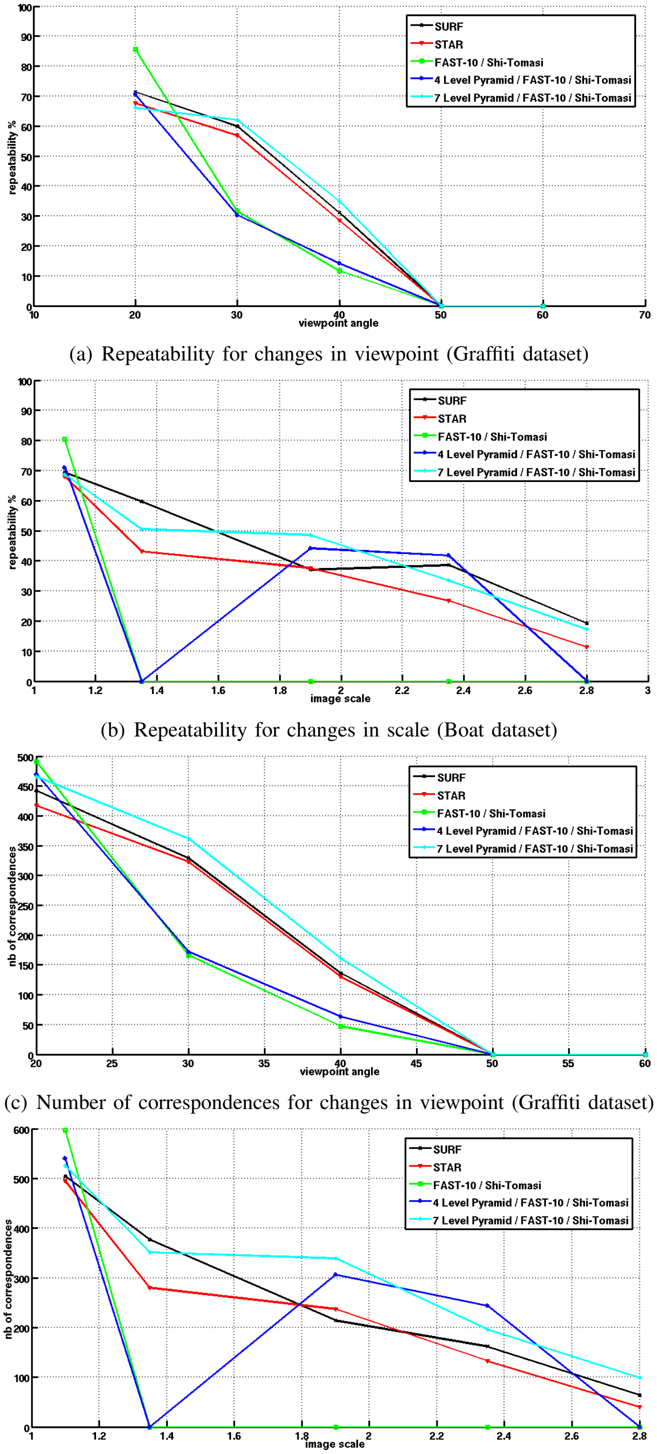
[1] http://www.robots.ox.ac.uk/∼vgg/research/affine/.

TABLE I
COMPARISON OF COMPUTATION TIMES

Keypoint Detector	Computation Time (ms)
SIFT	867.6
SURF	121.4
STAR	31.2
FAST-10 / Shi-Tomasi	5.2
4 Level Pyramid / FAST-10 / Shi-Tomasi	8.8
7 Level Pyramid / FAST-10 / Shi-Tomasi	15.4

also consider the quality of the keypoints identified by each detector. This is achieved using the standard framework and graffiti and boat image sequences of Mikolajczyk et. al. [17]. This framework evaluates each detector's *repeatability*, which expresses its ability to find the same physical points under different viewing conditions. Figures 1(a) and 1(b) illustrate each detector's repeatability under changes in viewing angle and scale respectively. Figures 1(c) and 1(d) show the number of corresponding regions found under changes in viewing angle and scale respectively. All plots are based on the 800 strongest keypoints and the standard overlap threshold of 40%. The results indicate that the single-scale FAST-10 detector has relatively poor robustness to changes in viewpoint, and extremely poor robustness to scale changes. It is also evident that the coarse scale-space of the 4-Level pyramid used by PTAM results in poor robustness to scales which fall between pyramid levels. However, the 7-level pyramid covering the same number of octaves, but with finer scale sampling outperforms both the SURF and STAR detectors in almost all cases.

In addition to evaluating the detectors alone, it is crucial to consider the impact their accuracy has on matching performance. Again we use the standard framework and graffiti and boat image sequences of Mikolajczyk et. al. [17]. In this case the matching performance is evaluated using the *recall-precision* criterion. Recall is the ratio between the number of correctly matched regions and the total number of corresponding regions between two images. The standard plots show recall vs. 1-precision, where 1-precision is the ratio between the number of incorrectly matched regions and the number of matched regions. Therefore, a good descriptor should have a high recall rate for low 1-precision. Figures 2(a) and 2(b) illustrate image matching performance using the SCARF descriptor [18] under changes in viewing angle and scale respectively. It is clear that the SURF detector is superior in almost all cases. However, surprisingly the number of pyramid levels (1,4 or 7) for the FAST-10 detector has marginal influence on matching performance for changes in scale. Considering the trade-off between matching performance and computation time, we have chosen to utilise the 4-Level pyramid FAST corners for relocalisation. As real-time performance is crucial we employ the SCARF descriptor for keypoint matching as it has matching performance comparable to SURF, but can be computed approximately 25 times faster, and each descriptor occupies just 60 bytes and can be compared almost 64 times faster [18].



(a) Repeatability for changes in viewpoint (Graffiti dataset)

(b) Repeatability for changes in scale (Boat dataset)

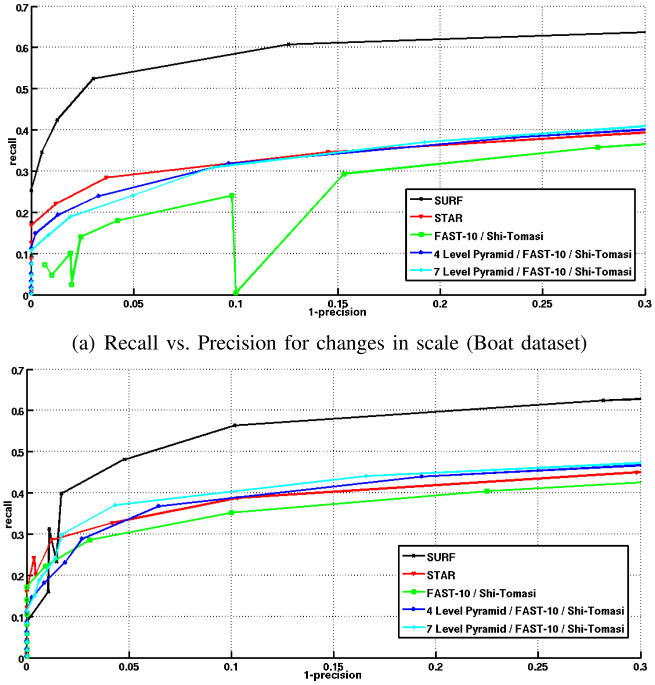
(c) Number of correspondences for changes in viewpoint (Graffiti dataset)

(d) Number of correspondences for changes in scale (Boat dataset)

Fig. 1. Keypoint detector performance under various transformations

V. IMAGE-TO-MAP FEATURE CORRESPONDENCES

If the uncertainty in the camera's pose is known then the spatial search for matches between the current image and map can be constrained, which accelerates the matching process



(a) Recall vs. Precision for changes in scale (Boat dataset)

(b) Recall vs. Precision for changes in viewpoint (Graffiti dataset)

Fig. 2. Keypoint matching performance under various transformations

significantly. However, when tracking fails pose uncertainty grows without bound to the point where the entire map must be considered and the matching process becomes a bottleneck. This prompts us to consider a two stage approach to relocalisation. In the first stage only a local map (containing up to 3000 features) around the point where tracking failed is considered and the camera motion is modelled as a random walk with a constant linear velocity. If the camera fails to relocalise before the position uncertainty grows beyond the bounds of the local map, then the second relocalisation stage which considers the entire map (up to 150,000 features) is invoked. A local map centered around any given position is obtained by storing keyframes in a Priority R-tree [19] which enables very efficient indexing of the K nearest-neighbour (k-NN) keyframes. Each keyframe maintains a list of observed map features, thus a local map can be rapidly constructed from the retrieved keyframes. This approach also lends itself towards the direct integration of absolute position estimates from external sensors such as GPS. Each time a GPS measurement is received we can simply update the camera position estimate, position uncertainty and local map keyframes.

Regardless of whether we are matching to the local or global map, identifying correspondences is the bottleneck for real-time relocalisation [1]. The best match for each feature in the current frame is found by identifying its nearest neighbour(s) in the set of map features based on the SCARF descriptors and a similarity metric. No algorithms were identified that could determine the exact nearest neighbors of the 60-dimensional SCARF descriptors any faster than an exhaustive search. However, several publicly available software libraries provide approximate nearest-neighbour search algorithms which have been compared and evaluated for both relocalisation stages.

OpenCV[16] provides an approximate nearest-neighbours search algorithm based on the KD-tree, known as the Best-Bin-First (BBF) algorithm [20]. The Approximate Nearest Neighbor library (LIBANN) [21] offers both a KD-tree and box-decomposition tree (bd-tree) based approximate nearest-neighbour search [22]. The LSHKIT library [23] offers an efficient approximate nearest-neighbour search algorithm based on local sensitivity hashing (LSH) [24]. The Fast library for approximate nearest-neighbours (FLANN) [25] provides approximate nearest neighbour searches based on either a set of randomised KD-trees or a hierarchical k-means tree [26]. A brute-force search based on the sum of absolute differences (SAD) is also considered for local map matching, but is infeasible for global map matching.

For local map matching the primary concern is responsiveness. It is desired that the system drops as few frames as possible before relocalisation occurs and tracking resumes. Therefore, we impose a constraint which requires the framerate to remain above 15 frames per second (fps), and compare the matching performance of each algorithm given the same time budget. Note that all searches with the exception of brute-force require time to construct a search index. This construction time must be factored into the time-budget as the local map is different each time the relocaliser is invoked. A framerate of 15fps provides a time-budget of 66.7ms per frame, however keypoint detection, description, local map generation and robust pose estimation requires up to 25ms, thus approximately 40ms remains for matching.

Table II summarises the construction and search times for each algorithm using a dataset of 100 random images covering four distinct image categories: graffiti street art, boats, bark and walls. Note that the first two columns always sum to 40ms. Therefore, even if the local map is updated every frame the framerate will remain above 15fps. The final column indicates the peak framerate that can be achieved when the local map is not updated. Each of the algorithms with the exception of the brute-force search is approximate, therefore we also consider their *recall-precision* performance based on nearest-neighbour matching of the 800 strongest keypoints. Figure 3 illustrates the algorithm's performance for NN matching to a local map with 3000 features. It is evident that the brute-force SAD search has the best matching performance, however the hierarchical k-means tree also performs well and provides a trade-off between a $\sim 10\%$ reduction in recall and a higher peak framerate (22.3fps). In practice relocalisation generally occurs within several frames, thus higher recall is preferred to a higher peak framerate and consequently the proposed system utilises the brute-force SAD search for local map matching.

For global map matching a brute-force search is infeasible and an approximate nearest-neighbour algorithm must be adopted. For global map matching responsiveness is still of primary importance, however, given the size of the global maps (up to 150,000 features) the framerate constraint is relaxed to 1 frame per second. Table III summarises the search index

TABLE II

COMPARISON OF NEAREST-NEIGHBOUR SEARCH ALGORITHMS FOR LOCAL MAP MATCHING

Algorithm	Construction	Search	Peak FPS
LSHKIT - LSH	8ms	32ms	17.0
LIBANN - BD-tree	8ms	32ms	17.0
LIBANN - KD-tree	3ms	37ms	15.7
OpenCV - BBF	7ms	33ms	16.8
FLANN - KD-tree	9ms	31ms	17.3
FLANN - Kmeans Tree	22ms	18ms	22.3
Brute-force SAD	N/A	40ms	15.0

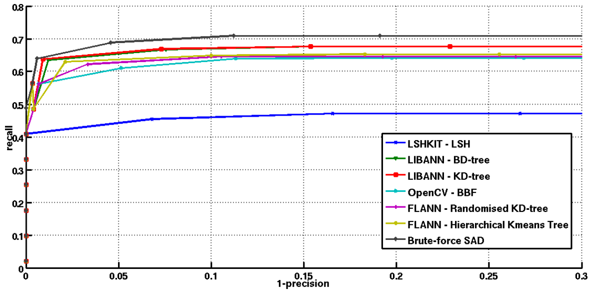


Fig. 3. Recall vs. Precision for approximate nearest-neighbour matching on a local map with 3000 features

construction times for a 150,000 feature map. Note that while a low construction time is still important, for global map matching the search index is only constructed once and is therefore not considered for the framerate constraint. Figure 4 illustrates each algorithms recall-precision performance based on nearest-neighbour matching of the 800 strongest keypoints to the 150,000 feature map. It is clear that the two algorithms from FLANN have superior recall performance, however the two algorithms have the worst construction costs. On a multi-core processor it is possible to construct the global search index in parallel to initial local map searches. Therefore in most cases the Randomised KD-tree construction will be completed before global matching is invoked.

TABLE III

NEAREST-NEIGHBOUR SEARCH INDEX CONSTRUCTION TIMES FOR GLOBAL MAP MATCHING

Algorithm	Construction time
LSHKIT - LSH	61ms
LIBANN - BD-tree	656ms
LIBANN - KD-tree	297ms
OpenCV - BBF	2140ms
FLANN - KD-tree	2365ms
FLANN - Kmeans Tree	9761ms

VI. OUTLIER REJECTION AND INDEPENDENT HYPOTHESIS FORMULATION

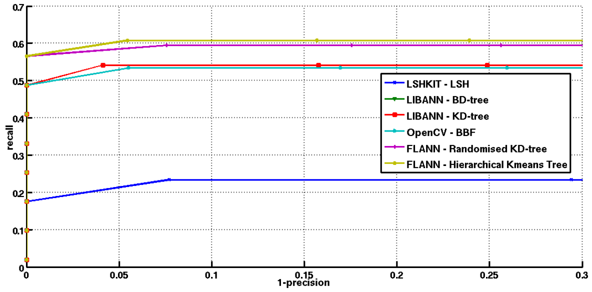


Fig. 4. Recall vs. Precision for approximate nearest-neighbour matching on a global map with 150000 features

The proportion of correct correspondences (inliers) yielded by the matching algorithms presented in the previous section tends to be fairly high even for extremely large maps, however a robust approach to camera pose estimation is still vital. A large proportion of the outliers can be rapidly eliminated by applying an extremely fast implementation of density-based spatial clustering (DB-SCAN) [27] to the set of corresponding map features. This groups together features which may be observed by a single keyframe, and any clusters with very few features (≤ 5) are disregarded. A camera pose estimate for each of the remaining clusters is then obtained using the well-known three-point pose algorithm [28] embedded within a Preemptive Random Sample Consensus (Preemptive-RANSAC) [29] framework. Unlike standard RANSAC, pre-emptive RANSAC generates a fixed number of hypotheses upfront, concurrently scores all hypotheses according to a robust likelihood function, and periodically eliminates the most unlikely to avoid excessive scoring of useless hypotheses. Like Guan et. al. [6] when generating hypotheses we reject sets of correspondences which refer to the same map point, and map points which are very close together or approximately collinear. In addition, if an external absolute pose estimate or external attitude estimate is available we use these and their associated uncertainties to eliminate unlikely hypotheses.

VII. EXPERIMENTAL EVALUATION

The system is able to operate in real-time and successfully relocalises following tracking failure in both indoor and outdoor environments. Figure 5 shows the keyframes and map points generated by a camera traversing a large loop of a laboratory containing repetative patterned carpet, featureless walls and many almost identical computers, desks and chairs. Rapid camera motion at several points of the trajectory causes tracking to fail, but in each case the relocaliser recovered immediately. Once the trajectory was completed the resulting map consisted of 821 keyframes and approximately 45,000 map points. To test the relocaliser the camera was blinded, relocated and/or reoriented and then unblinded (kidnapped). If the blinded interval was short then local map relocalisation was attempted, and on average required 61.2ms per frame (best-case 15ms, worst-case 190ms), and relocalisation always occurred within one second. If the blinded interval was long then global map relocalisation was invoked and required an average of 1034ms per frame (best-case 818ms, worst-case 1134ms) and relocalisation always occurred within four seconds. Across the 130 trials the relocaliser had a success rate above 99%, with a single incorrect pose estimate resulting in an incorrect relocalisation rate of less than 1%.

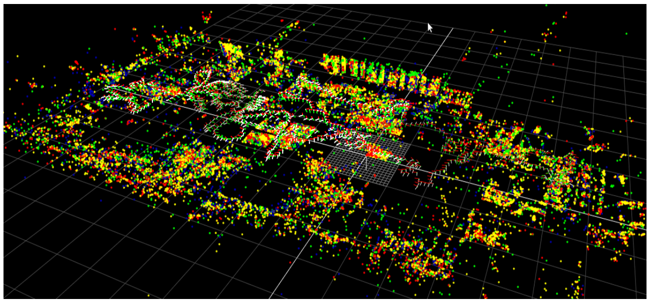


Fig. 5. Global map of 821 keyframes and approximately 45,000 map points

VIII. Conclusions

The relocalisation system presented is able to quickly generate robust camera pose estimates for very-large scale maps, even when prior knowledge of the camera pose is unavailable. Robust feature correspondences between the current image and map are obtained using a pyramidal FAST keypoint detector, real-time SCARF feature descriptors, and an approximate nearest neighbour algorithm based on randomized KD-trees. Density based clustering of the 3D point-to-image correspondences yields a small number of correspondence sets from which potential pose hypotheses are obtained efficiently via a three-point absolute pose algorithm embedded within a Preemptive RANSAC framework. The system employs a two stage approach to relocalisation in which initially only a local map (containing up to 3000 features) around the point where tracking failed is considered. If the camera fails to relocalise before the position uncertainty grows beyond the bounds of this local map, then the second relocalisation stage considers the entire map (up to 150,000 features). We demonstrate through both simulated and practical experiments that the local map relocalisation is both robust and responsive, achieving a framerate of $\sim$ 15fps. Furthermore, the global map relocalisation can handle maps more than an order of magnitude larger than the current state-of-the-art while maintaining a framerate of approximately $\sim$ 1fps and a success rate above 99%.

References

[1] D. Chekhlov, W. Mayol-Cuevas, and A. Calway, "Appearance based indexing for relocalisation in real-time visual SLAM," in *In 19th Bristish Machine Vision Conference*, 2008, pp. 363–372.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.

[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[4] B. Williams, G. Klein, and I. Reid, "Real-time SLAM relocalisation," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, oct. 2007, pp. 1 –8.

[5] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 1465–1479, September 2006.

[6] T. Guan, L. Duan, Y. Chen, and J. Yu, "Fast scene recognition and camera relocalisation for wide area augmented reality systems," *Sensors*, vol. 10, pp. 6017–6043, 2010.

[7] M. Calonder, V. Lepetit, K. Konolige, J. Bowman, P. Mihelich, and P. Fua, "Compact signatures for high-speed interest point description and matching," in *ICCV*, 2009.

[8] O. Chum and J. Matas, "Matching with PROSAC progressive sample consensus," in *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '05. IEEE Computer Society, 2005, pp. 220–226.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

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

[11] P. H. S. Torr and A. Zisserman, "Mlesac: a new robust estimator with application to estimating image geometry," *Comput. Vis. Image Underst.*, vol. 78, pp. 138–156, April 2000.

[12] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *PAMI*, 2009.

[13] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 1994, pp. 593–600.

[14] T. Lindeberg, "Feature detection with automatic scale selection," *IJCV*, vol. 30, no. 2, pp. 79–116, 1998.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vision*, vol. 65, pp. 43–72, November 2005.

[18] S. J. Thomas, B. A. MacDonald, and K. A. Stol, "Real-time robust image feature description and matching," in *Proceedings of the 10th Asian conference on Computer vision - Volume Part II*, ser. ACCV'10, 2011, pp. 334–345.

[19] L. Arge, M. D. Berg, H. Haverkort, and K. Yi, "The priority r-tree: A practically efficient and worst-case optimal r-tree," *ACM Trans. Algorithms*, vol. 4, pp. 9:1–9:30, March 2008.

[20] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *In Proc. IEEE Conf. Comp. Vision Patt. Recog*, 1997, pp. 1000–1006.

[21] D. Mount and S. Arya, "ANN: A Library for Approximate Nearest Neighbor Searching," [online]. http://www.cs.umd.edu/ mount/ANN/ [accessed 28 July 2011], 2010.

[22] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, vol. 45, pp. 891–923, November 1998. [Online]. Available: http://doi.acm.org/10.1145/293347.293348

[23] W. Dong, "LSHKIT: A C++ Locality Sensitive Hashing Library," [online]. http://lshkit.sourceforge.net/ [accessed 28 July 2011], 2009.

[24] W. Dong, Z. Wang, M. Charikar, and K. Li, "Efficiently matching sets of features with random histograms," in *Proceeding of the 16th ACM international conference on Multimedia*, ser. MM '08, 2008, pp. 179–188.

[25] M. Muja, "FLANN - Fast Library for Approximate Nearest Neighbors," [online]. http://people.cs.ubc.ca/ mariusm/index.php/FLANN/FLANN [accessed 28 July 2011], 2011.

[26] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. INSTICC Press, 2009, pp. 331–340.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, 1996, pp. 226–231.

[28] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *IJCV*, vol. 13, no. 3, pp. 331–356, 1994.

[29] D. Nistér, "Preemptive ransac for live structure and motion estimation," in *ICCV*, Washington, DC, USA, 2003, pp. 199–206.