RGBD Relocalisation Using Pairwise Geometry and Concise Key Point Sets

Shuda Li and Andrew Calway*

Abstract— **We describe a novel RGBD relocalisation algorithm based on key point matching. It combines two components. First, a graph matching algorithm which takes into account the pairwise 3-D geometry amongst the key points, giving robust relocalisation. Second, a point selection process which provides an even distribution of the 'most matchable' points across the scene based on non-maximum suppression within voxels of a volumetric grid. This ensures a bounded set of matchable key points which enables tractable and scalable graph matching at frame rate. We present evaluations using a public dataset and our own more difficult dataset containing large pose changes, fast motion and non-stationary objects. It is shown that the method significantly out performs state-of-the-art methods.**

I. Introduction

Relocalisation of RGB and depth (RGBD) sensors involves estimating the pose of a 'lost' sensor with respect to a 3-D map given a query RGBD frame which has complete or partial overlap with the map. Previous approaches can be divided into two groups: those based on key frames [1], [2], [3], [4] and those based on key points [5], [6], [7], [8]. The former match the query frame with previously captured and synthesised key frames and use the known poses of the latter to derive the query pose. This avoids maintaining and matching large sets of key points, leading to fast implementation if low resolution frames are utilised at the expense of accuracy. The effective range of relocalisation is also limited to being close to that of the trajectories used to capture key frames - performance reduces significantly when the new pose has a wide baseline or different viewing direction from that of the closest pose on a trajectory.

The alternative is to make use of key point matching, in a similar manner to that employed in RGB simultaneous localisation and mapping (SLAM), as in [9] for example. The approach was used in early RGBD relocalisation methods [5], [6], [7], [6]. Such approaches have the advantage that they are able to recover new poses which are significantly different from the views used to build a map. More recent methods have sought to improve relocalisation by making use of absolute and relative depth information [8]. However, these methods, in common with the earlier RGB methods, suffers from two disadvantages: first, the number of key points grows rapidly with the map size, hence limiting scalability; and second, some visual characteristics of the scene, such as surface reflectance, low texture and other artifacts, such as motion blur, can render the key point matching, and hence the relocalisation, unreliable.

In this paper we describe a novel key point based method designed to address the above limitations. It combines two components. The central component is an efficient graph matching algorithm which takes into account the pairwise 3-D geometry amongst the key points in addition to local appearance and surface normal information. The use of pairwise geometry enables the method to overcome the ambiguities caused by visual characteristics and artifacts, giving robust relocalisation. An overview of the process is illustrated in Fig. 1.

As with all key point matching schemes we need to limit the number of points to achieve scalability - rising numbers due to a growing map will eventually render matching intractable at frame rate. We address this by using a concise representation in which the 'most matchable' key points are evenly distributed over the scene within a volumetric grid (VG), with points selected by a non-maximum suppression algotithm within voxels. This reduces the number of points with only a small corresponding reduction in relocalisation accuracy compared to that obtained with a much larger set.

When coupled with standard RANSAC pose estimation and iterative closest point (ICP) pose refinement, the resulting method is very robust to large perspective distortion, sparse features caused by motion blur, repetitive scene texture and even non-stationary scenarios in which there is substantial movement of objects. We demonstrate that the method gives performance well beyond that of existing key frame and key point based methods, for both existing datasets and our own more challenging dataset.

II. Background

A good example of RGBD relocalisation using key frames is that described in [1]. This is based on generating sets of tiny synthetic views around the viewing sphere and linearly regressing a new pose from the synthetic poses by identifying a subset of best matches with a low resolution version of the query frame. However, a limitation is that computation grows exponentially with frame resolution, hence limiting accuracy. Glocker et al. [3] improve upon the method by encoding each RGBD frame using a binary code and matching frames using Hamming distances. This allows the use of higher resolution frames and more reference data, hence increasing accuracy. However, as pointed out in [3], a limitation lies in the fact that ”the camera view should not be substantially different from the views represented by the key frame”.

A variant of the frame based methods is to adopt a machine learning framework by training a random regression forest which associates each RGBD pixel with a distribution of

*All authors are with the Department of Computer Science, University of Bristol, UK {csxsl, csadc}@bristol.ac.uk.

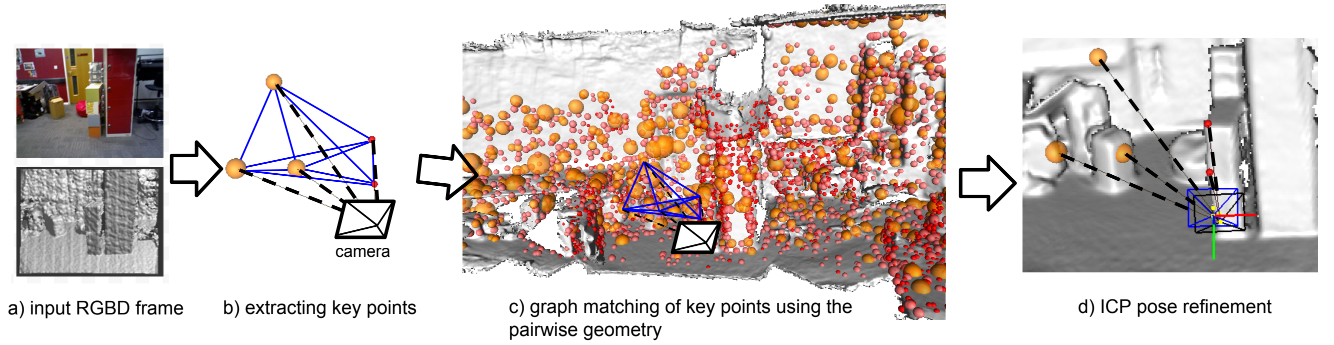


Fig. 1: Overview of the relocalisation method. From an input RGBD frame (a), multi-scale features with 3-D information are extracted (b) and matched to those in the global reference map by taking pairwise 3-D geometry into account (c); The matches are used to estimate an initial pose using standard pose estimation method. It is shown as the black camera in (d). Finally, it is refined by ICP and shown as the blue camera in (d).

camera poses. In [2], a pose hypothesis is calculated from 3 randomly sampled pixels. Each pixel is associated with a 3-D coordinate thanks to the random forest. The best hypothesis is selected using ICP refinement. Guzman-rivera et al. [4] improve Shotton's method by constraining the hypotheses to be marginally independent. Both methods avoid extracting features at the cost of being also restricted to poses calculated by interpolating reference poses and a need for a large training set of representative frames.

As noted earlier, the alternative is to match key points in the query frame with those in the map as described in [5], [6], [7], [8]. Notably in [8], which is closest to our own method, standard appearance descriptor based matching is supplemented by utilising the depth information to impose 3-D distance consistency between pairs of key points along a chain starting from a given 'anchor', where the latter is given by the best matching pair. The method is fast and can relocalise frames at some distance away from those used to construct the map. However, it has the disadvantage that identifying reliable anchors is problematic, leading to instability, and in common with other key point methods based on appearance descriptors, it suffers both from a lack of scalability as the map size increases and unreliable matching when dealing with visual ambiguity, low texture or motion blur, for example.

In this work we address these issues by: (i) imposing pairwise 3-D distance consistency over a fully connected graph, thus avoiding the use of chains and anchors; (ii) enhancing the method by also imposing normal-normal and normal-position consistency between the key points, and (iii) making use of a concise key point representation which significantly increases scalabilty. The resulting method gives significantly improved performance, not only over that in [8], but also over state-of-the-art key frame methods.

III. System Overview

We start by giving an overview of the complete tracking, mapping, and relocalisation system. Fig. 2 illustrates the main components and their interaction. The structure is similar in form to the KinectFusion system described in [10]. Incoming RGBD frames are aligned in 3-D by matching key points to give pose estimates and hence enable fusing of the depth maps to give a global map and track the 3-D trajectory of the sensor. For each frame, ICP is used to refine both the pose and the depth estimates. The global map is stored in the form of dense surface data encoded within a volumetric truncated signed distance function (TSDF). Alongside this we introduce a VG structure, which is populated with key points as map building proceeds. Each voxel in the VG is constrained to contain a single 'most matchable' key point to ensure an even distribution of points across the scene whilst reducing the number of points in the representation, hence aiding scalability. Further details are given in Section IV.

Specifically, given an incoming RGBD frame, it is transferred into the tracking module where key points are matched with those in the previous frame to estimate an initial sensor pose. This initial pose is then refined using ICP and on convergence, the incoming frame is integrated into the global 3-D map, i.e. the depths are fused into the TSDF and the key points are considered for merging into the VG based on non-maximum suppression. If ICP fails to converge, indicating tracking failure, the incoming frame is passed to the relocalisation module. Relocalisation is similar to tracking, except that, in relocalisation, the key points are matched with the VG point set as illustrated in Fig. 1. In the following sections we give details of the key point representation and the graph matching algorithm.

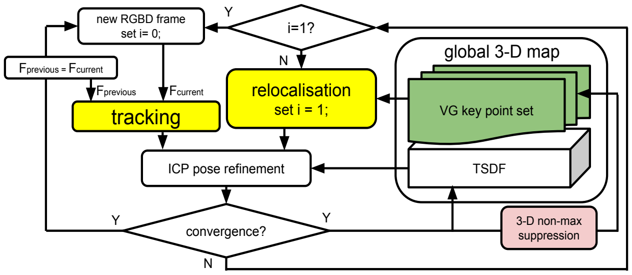


Fig. 2: Main components of the tracking, mapping and relocalisation system.

IV. Key point representation

We first describe the representation of key points and how they are incorporated into the VG structure. Each key point is represented by a binary appearance descriptor (we used BRISK [11] in the experiments), its 3-D position in world coordinates, an associated surface normal, a matchability score and the angle α between the viewing direction when it was first observed and the surface normal. The 3-D position determines which voxel in the VG the key point may occupy (using quantisation) and the matchability score and angle α are used for selecting the best key point for each voxel using non-maximum suppression.

The matchability score is based on the binary appearance descriptor and defined as the one subtracts the ratio of the Hamming distance with its first nearest neighbour (NN) to that of its second NN within a set of previously seen descriptors. This is similar to that adopted in [12], [13], for example. We experimented with two different sets of descriptors: those in the current and previous frames; and those in the complete map. We found no significant difference in terms of matching performance, although the latter gave marginally better results, but at the expense of greater computational cost.

The concise VG representation for the key points is built using a form of non-maximum suppression. Specifically, given a set of key points to be merged into the VG (corresponding to the inliers resulting from the matching used in tracking), their 3-D positions are first quantized according to the resolution of the grid (see below), thus identifying the voxel that they may occupy. If the voxel is already occupied and/or there is more than one contending key point, then the matchability scores and angles α are compared to determine the ‘most matchable’ point.

We experimented with a number of comparison metrics but found only marginal differences. In the experiments we adopted the simple approach of selecting the point with the highest matchability score and smallest angle (points observed in parallel with their surface normal being regarded as more reliable for matching), with random selection being used in the case of different points meeting each criterion.

The above process results in a much smaller key point set, usually several magnitudes smaller, but we found it to have similar reliability performance for matching compared with keeping all the points. To illustrate, Fig. 3 shows a reliability comparison between using all points and using the non-maximum suppressed point set. We built a map using a single video sequence and then subsequently matched each frame of the same sequence to the map using either all the key points or the non-maximum supressed set.

The graph in Fig. 3 shows both the number of key points used in each case and the reliability of each set measured by the percentage of confident matches out of all points per frame, where a match is confident if the ratio of the Hamming distance of its 1st-NN to that of its 2nd-NN is lower than 0.6 [12]. In this experiment we used 700 frames from ’living room’ sequence number 2 in the ICL dataset [14].

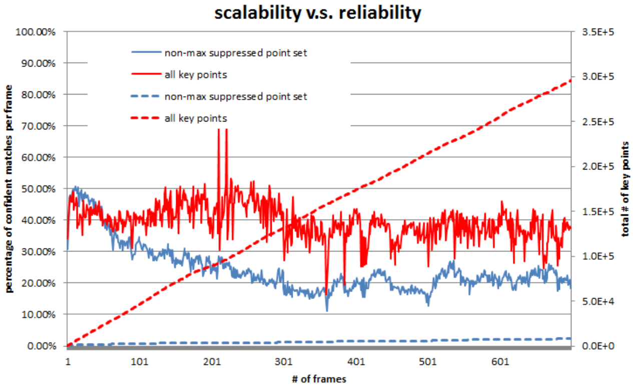


Fig. 3: Comparison of the matching reliability and scalability between using all key points and using the reduced non-maximum suppressed set. The red/blue solid lines show the reliability of matching per frame and the red/blue dotted lines show the number of points being added to the map per frame.

Note that although the size of the sets becomes drastically different, the reliability of the reduced point set is only slightly lower (around 10-15%) than using all points on average. Moreover, we observed in experiments that this sacrificing of reliability for scalability makes no noticeable difference to the accuracy of relocalisation.

We also investigated the option of using multilevel VGs, with a different voxel resolution on each level, thus enabling the inclusion of different non-maximum suppression volume sizes. Key points were allocated to levels according to their *absolute scale* defined as $s = rD/f$, where r is the radius associated with the 2-D region of interest for the point, f is the focal length, and D is the depth estimate associated with the key point centre. This is preferred to the 2-D scale r as it takes into account the relative distance of the camera when observing the corresponding 3-D point. Quantization of s was used to allocate points to levels and matching was constrained to be between points on the same level. We also experimented with putting points that lay on quantization boundaries in both adjacent levels to account for uncertainties in the depth and 2-D scale.

However in the experiments we found that this use of multiple levels did not have a significant impact, with only a marginal improvement in performance using two levels with sharing of boundary points over that obtained using a single, three or more levels. Our view is that this outcome was due in part to the type of sequences used in the experiments which did not have a significant amount of scale changes, although this needs further investigation.

V. Graph matching with pairwise constraints

In this section we describe the graph matching algorithm which incorporates pairwise 3-D geometry constraints amongst the key points in each set being matched. The algorithm is applicable to any set of key points, although as noted above tractability becomes problematic for very large set sizes. In the following we assume that we are attempting to match key points in a current frame to those representing a 3-D map, whether that consists of the full set of key points or the reduced non-maximum suppressed set.

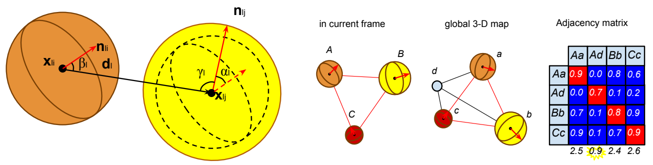


Fig. 4: Geometric relationships used in the pairwise geometric constraints and examples of matching pairs and corresponding adjaceny matrix.

A. Pairwise constraints

We use three types of pairwise constraints based on inter-point geometric relationships, namely distance consistency, normal-normal consistency, and normal-position consistency. Assume we have two pairs of candidate matching points (p_{ci}, p_{mi}) and (p_{cj}, p_{mj}), where c and m indicate whether a point is from the current frame or the map, respectively. We denote the 3-D location and unit normal of a point by $\mathbf{x}_{li}$ and $\mathbf{n}_{li}$, respectively, where $l \in \{c, m\}$, as illustrated in Fig. 4. The Euclidean distance between points in the same set is then $d_l = \left\|\mathbf{x}_{li} - \mathbf{x}_{lj}\right\|_2$ and the normalized direction from p_{lj} to p_{li} is $\mathbf{d}_l = \left(\mathbf{x}_{li} - \mathbf{x}_{lj}\right)/d_l$. The angle between normals is $\alpha_l = \arccos(\mathbf{n}_{li} \cdot \mathbf{n}_{lj})$ and we also define the angles between the normals and $\mathbf{d}_l$: $\beta_l = \arccos(\mathbf{n}_{li} \cdot \mathbf{d}_l)$ and $\gamma_l = \arccos(\mathbf{n}_{lj} \cdot \mathbf{d}_l)$, where '$\cdot$' denotes the dot product. These relationships are illustrated in Fig. 4. Note that where possible we have dropped the indices i and j to simplify notation.

The consistency measures are then derived by comparing the above relationships for the point pairs in each set, i.e. if the above two matching pairs are good, then we would expect the geometric relationship between p_{ci} and p_{cj} to be very similar to that between p_{mi} and p_{mj} since each pair would refer to the same two 3-D points in the scene (which is assumed to be rigid). Hence we define the *distance consistency* measure as $d_{cm} = |d_c - d_m|$, the *normal-normal consistency* measure as $\alpha_{cm} = |\alpha_c - \alpha_m|$ and the *normal-position consistency* measures as $\beta_{cm} = |\beta_c - \beta_m|$ and $\gamma_{cm} = |\gamma_c - \gamma_m|$. The measures are then combined into a single consistency score F_{ij} for pairs i and j as defined below, where F_{ij} is high if the pairs are consistent.

$$F_{i,j} = \begin{cases} e^{-d_H} & \text{if } i = j \\ 0 & \text{else if } d_c = 0 \text{ or } d_m = 0 \\ e^{-(d_{cm} + \alpha_{cm} + \beta_{cm} + \gamma_{cm})} & \text{else} \end{cases} \tag{1}$$

where d_H is the Hamming distance between the binary descriptors of the pair (p_{ci}, p_{mi}) and all the pairs $(p_{ci}, p_{mj}) \in S$. S is the set of candidate matching pairs. For example, as illustrated in Fig. 4 (middle and right), let $S = \{(A,a), (A,d), (B,b), (C,c)\}$. When pairs i and j share a point, e.g. $(p_{ci}, p_{mi}) = (A,a)$ and $(p_{cj}, p_{mj}) = (A,d)$, i.e. both of them matched with the point A, then we set $F_{ij} = F_{ji} = 0$. This is convenient for applying a 1-to-1 constraint as described below. Note that in experiments we found that relative weighting of the distance and directional consistency measures has little impact on performance.

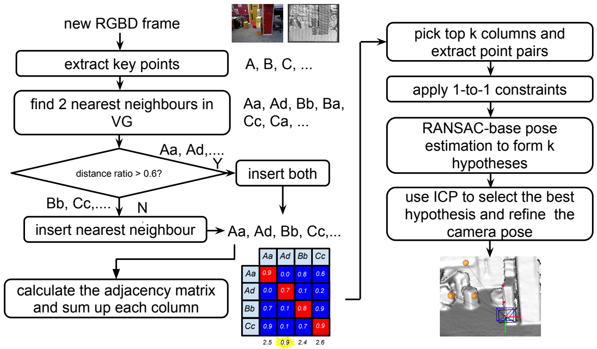

new RGBD frame
extract key points
A, B, C, ...
find 2 nearest neighbours in VG
Aa, Ad, Bb, Ba, Cc, Ca, ...
distance ratio > 0.6?
Aa, Ad,....
Y
insert both
Bb, Cc,....
N
insert nearest neighbour
Aa, Ad, Bb, Cc,...
calculate the adjacency matrix and sum up each column
Aa Ad Bb Cc
Aa 0.9 0.0 0.8 0.6
Ad 0.0 0.7 0.1 0.2
Bb 0.7 0.1 0.8 0.9
Cc 0.9 0.1 0.7 0.9
2.5 0.9 2.4 2.6
pick top k columns and extract point pairs
apply 1-to-1 constraints
RANSAC-base pose estimation to form k hypotheses
use ICP to select the best hypothesis and refine the camera pose

Fig. 5: Relocalisation module in detail.

B. Adjacency matrix

Equation (1) is then used to construct an adjacency matrix (AM) corresponding to a fully connected graph. As shown in Fig. (4) (right), the diagonals in AM then represent the descriptor similarities between candidate matched points and the off-diagonals correspond to the pairwise geometric consistency between each pair of candidate matches.

Fig. 5 illustrates the relocalisation process and the use of the AM. The points A, B and C are extracted from the current frame and points a, b, c and d are from the 3-D map (see also Fig. 4 middle). First, both sets of points are matched using binary descriptors. If the matching reliability is higher than 0.6, the matches with the 1-NN will be added into a candidate list (Bb and Cc); otherwise, both matches with 1-NN and 2-NN (Aa and Ad) will be added into the list. An AM representing a fully connected graph of the matches in the candidate list is then contructed using equation (1).

Note that the above AM can be solved using a standard graph matching algorithm such as [15]. However this is computational expensive and would prevent frame rate operation. Instead, we adopt an alternative sub-optimal approach, but one which we found to give comparable performance. This operates as follows. We simply sum up each column of the AM and then pick the top k columns with the highest summation. This is because the summation of a column describes the overall consistency of a candidate match with all other matches. From each selected column, we select the elements whose consistency scores are higher than a threshold τ. For example, in Fig. 5, assuming $\tau = 0.5$, then the inliers extracted from column Cc are Aa, Bb and Cc. The element Ad can be removed as an outlier.

Overall, we can form k candidate lists of matches. Then, we run RANSAC pose estimation on each candidate lists to estimate camera pose hypotheses. The best hypothesis will be selected using ICP by aligning the depth with the dense surface data of the map. As shown in Fig. 4, the column Cc and Aa are the top 2 columns, from which two feature sets can be extracted. The resulting matches of feature pairs are retrieved by going through each element in the column.

C. 1-to-1 constraint

The resulting matching pairs may contain 1-to-many or many-to-1 matches. We apply the 1-to-1 constraint by firstly identifying them and then selecting the most consistent matches. We find all elements in the adjacency matrix M where $M(i,j) = 0$, it indicates that the pair i and j has been matched with the same key point. We then only keep the largest element out of $M(i,p)$ and $M(j,p)$ where p is the index of the column out of the k top columns.

VI. Evaluation and experiments

Implementation: We implemented the method using the OpenCV library 3.0 beta, with parameters set as follows. We adopt the fast multiscale Hessian key point detector [16] with threshold 100, the binary descriptor BRISK [11] with 4 octave layers and the GPU-accelerated brute force point matching with distance ratio threshold 0.6. The pose is estimated using an improved RANSAC based absolute orientation algorithm [17], [18] using both 3-D points and normals of corresponding points. The RANSAC iteration is set as 100. We adopted the ICP from the Point Cloud Library 1.7 [19]. We used a volumetric TSDF with resolution of 512^3 and a two-level VG with resolution 128^3 and 64^3 with boundary sharing for the key point representation. The input depth maps were bilaterally filtered to reduce noise and the surface normals extracted using the cross product between neighbouring vertices [10]. The only parameter that needs to be tuned in the method is the consistency threshold τ. We tuned this using the 'living room' sequence from the ICL synthetic dataset [14] and found that a value of 0.05 gave the best performance, although it is not paricular sensitive to the precise value.

Comparison: We compared the performance of the proposed method with four other RGBD relocalisation methods. These were a standard feature-based method using Inverse Document Frequency analysis (IDF), the Anchor and Chain method (A&C) [8], the Randomized Ferns method (FERNS) [3] and the tiny image method [1](TINY). For FERNS, the total number of randomly selected testing points was 512. We used 5 possible pose hypotheses in each method and poses were refined using the same ICP algorithm.

Datasets: We used the '7 scenes' dataset [3], [2], [4] for evaluation of our method. The dataset provides a large amount of reference data to build maps which is favorable for methods relying on extensive training. Note that the proposed method is free of any specific training stage.

To increase difficulty, we also introduce a dataset of our own. We deliberately moved the sensor away from the reference trajectories used to build maps when capturing frames for testing to include more perspective distortion (see Fig. 6). The number of frames for building the reference map are considerably less than in the '7 scenes' dataset as well. The data are captured in 3 different scenes, namely 'desktop', 'laboratory' and 'lecture hall', which vary in physical size and scene texture. In each scene, we perform two types of movements: normal and fast movement for introducing motion blur. In addition, objects were moved around to create non-stationary scenes. The ground truth is captured using the tracking mode of Glocker's approach, the FERNS [3]. For non-stationary sequences, we acquire the ground truth by manually identifying the rigid part for tracking.

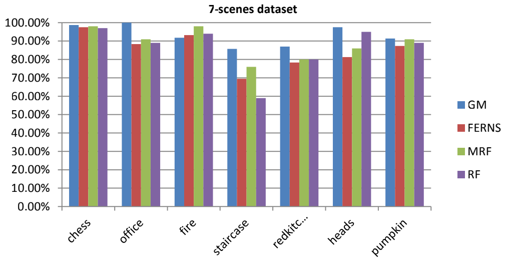


Fig. 7: Comparison of relocalisation success rates for maps built using all reference data.

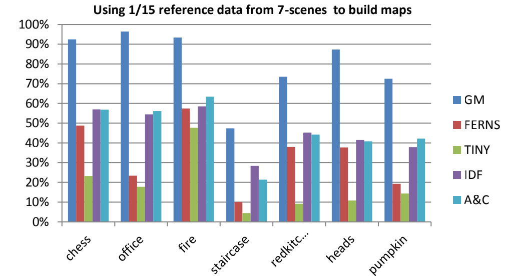


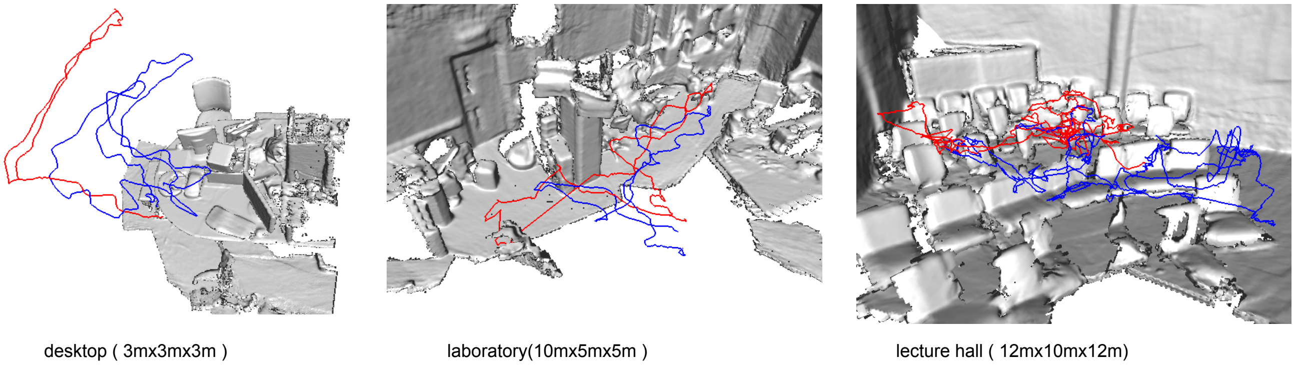
Fig. 8: Comparison of relocalisation success rates for maps built using 1 in every 15 frames.

Error metric: We measured the performance of relocalisation by success rate. For the '7 scenes' dataset, we adopt the same error metric as in [4], i.e., the estimated camera poses lie within 5cm translational error and 5 degree angular error of the ground truth; for our own dataset, we require it to be within 15cm translational error from the ground truth.

Results and discussion: We first compared our method, which we call graph matching (GM), with the key frame based methods FERNS, MRF and RF using all reference data in the '7 scenes' dataset. The numerical precision of MRF and RF were taken from the experimental reports in [4]. In Fig. 7, it can be seen that GM outperforms the other methods except for the 'fire' sequence. Possibly, it is because of unreliable features around the plant as pointed out in [13]. Note that GM performs significantly better than the key frame based approaches for the 'office' and the 'staircase' where objects are self similar and scene textures are repetitive. We hypothesize that this is due primarily to the use of the pairwise constraints in the matching.

In the second experiment, we used 1 in every 15 frames from the reference data to build maps. With the sparser reference data, the difference between the poses on the reference trajectories and those in testing increases and hence it becomes more difficult to relocalise. We compared our method with the image based approaches, FERNS and TINY, and the feature based approaches, IDF and A&C. The results are shown in Fig. 8. We observe that FERNS and TINY are especially sensitive to sparse reference data, arguably because the are key-frame based. GM performs significantly better than all of the other approaches.

In the last experiment, we compared GM with FERNS, GEE, IDF and A&C using our own '3 scenes' dataset. As Fig. 9 shows, GM outperforms all of the other approaches. Note that since the ground truth is captured using the tracking mode of FERNS, we expect that the result should favour


desktop (3mx3mx3m)

laboratory(10mx5mx5m)

lecture hall (12mx10mx12m)

Fig. 6: 3-D reference maps in our own ’3 scenes’ dataset. The testing trajectory (red) were deliberately moved away from those used to build the map (blue). The sequences of the ’desktop’, ’laborartory’ and ’lecture hall’ contain 500, 350 and 900 frames respectively.

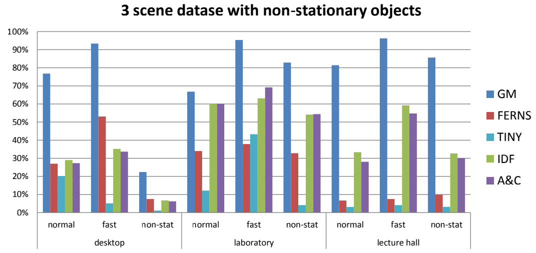


Fig. 9: Comparison using ’3 scenes’ dataset.

FERNS accordingly. However, in reality it is not the case. This is probably because we used the same ground truth poses to build the reference map for all other methods.

Timings: Implementation was done on a PC equipped with an I7-3770S CPU and a Nvidia Titan Black GPU. The key point extraction takes about 6ms. The pairwise matching takes 1-2 ms. The GPU-accelerated adjacency matrix computation and point matching takes about 4ms. The 3-D non-max suppression and key point integration takes about 10ms. The RANSAC-based pose estimation takes about 20ms and ICP refinement 90ms. Overall, relocalisation runs at about 12 frames per second.

VII. Conclusion and Future Work

We have described a novel algorithm for relocalisation of a RGBD sensor. Compared with the existing approaches, the method shows significant improvement in successful relocalisation rate, thanks to the pairwise geometric constraints used in matching and the use of a concise key point representation. There are a number of possible directions for future work. We intend to investigate the use of more features such as 3-D features and edges. We also plan to test the method for large scale out-door scene relocalisation.

References

[1] A. Gee and W. Mayol-Cuevas, “6D Relocalisation for RGBD Cameras Using Synthetic View Regression,” in *British Machine Vision Conference (BMVC)*, 2012.

[2] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, “Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images,” in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[3] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, “Real-Time RGB-D Camera Relocalization,” in *IEEE/ACM Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, Oct. 2013.

[4] A. Guzman-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi, “Multi-Output Learning for Camera Relocalization,” in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D Mapping : Using Depth Cameras for Dense 3D Modeling of Indoor Environments,” in *RGBD Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS*, 2010.

[6] A. S. Huang, A. Bachrach, P. Henry, M. Krainn, D. Maturana, D. Fox, and N. Roy, “Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera,” in *Intl. Symposium on Robotics Research (ISRR)*, 2011.

[7] S. Lieberknecht, A. Huber, S. Ilic, and S. Benhimane, “RGB-D Camera-based Parallel Tracking and Meshing,” in *IEEE/ACM Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, Oct. 2011.

[8] J. Martinez-Carranza, A. Calway, and W. Mayol-cuevas, “Enhancing 6D Visual Relocalisation with Depth Cameras,” in *Intl. Conf. on Intelligent Robot Systems (IROS)*, 2013.

[9] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway, “Robust Real-Time Visual SLAM Using Scale Prediction and Exemplar Based Feature Description,” in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[10] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion : Real-Time Dense Surface Mapping and Tracking,” in *IEEE/ACM Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[11] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust Invariant Scalable Keypoints,” in *Intl. Conf. on Computer Vision (ICCV)*, A. Abe and R. Oehlmann, Eds., 2011.

[12] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Intl. Journal of Computer Vision (IJCV)*, vol. 60, no. 2, Nov. 2004.

[13] W. Hartmann, M. Havlena, and K. Schindler, “Predicting Matchability,” in *IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[14] A. Handa, T. Whelan, J. Mcdonald, and A. J. Davison, “A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[15] M. Leordeanu and M. Hebert, “A Spectral Technique for Correspondence Problems Using Pairwise Constraints,” in *Intl. Conf. on Computer Vision (ICCV)*, Oct. 2005.

[16] H. Bay, T. Tuytelaars, and L. V. Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, June 2008.

[17] R. J. Micheals and T. E. Boult, “On the Robustness of Absolute Orientation,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2000.

[18] B. K. Horn, “Closed-form Solution of Absolute Orientation Using Unit Quaternions,” *Journal of the Optical Society of America A*, vol. 6, no. 4, Apr. 1987.

[19] R. B. Rusu and S. Cousins, “3D is here : Point Cloud Library (PCL),” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.