6D Relocalisation for RGBD Cameras Using Synthetic View Regression

Andrew P. Gee
http://www.cs.bris.ac.uk/~gee
Walterio Mayol-Cuevas
http://www.cs.bris.ac.uk/~wmayol

Visual Information Laboratory
University of Bristol
Bristol, UK

Abstract

With the advent of real-time dense scene reconstruction from handheld cameras, one key aspect to enable robust operation is the ability to relocalise in a previously mapped environment or after loss of measurement. Tasks such as operating on a workspace, where moving objects and occlusions are likely, require a recovery competence in order to be useful. For RGBD cameras, this must also include the ability to relocalise in areas with reduced visual texture. This paper describes a method for relocalisation of a freely moving RGBD camera in small workspaces. The approach combines both 2D image and 3D depth information to estimate the full 6D camera pose. The method uses a general regression over a set of synthetic views distributed throughout an informed estimate of possible camera viewpoints. The resulting relocalisation is accurate and works faster than framerate and the system's performance is demonstrated through a comparison against visual and geometric feature matching relocalisation techniques on sequences with moving objects and minimal texture.

1 Introduction

The ability to build 3D maps of a small environment is useful for a number of tasks beyond visualisation, e.g. mobile computing, robotic planning and cognitive assistance. These types of located maps can represent a *workspace* where some action or observation needs to occur. For a robot explorer, a small area of operation can be as simple as a neighbourhood around a door that needs to be opened; for an augmented reality system, the map provides a reference for anchoring annotations; and, for a cognitive assistant operating in, for example, a kitchen environment, it can help to prime regions to locate objects to guide the completion of a task.

Maps of this kind can nowadays be built in real-time with commodity hardware using single cameras [11, 15] or RGBD sensors [9, 14]. This real-time aspect brings further possibilities and constraints to the way in which these maps can be used.

While most of the work in this area has been concerned with the estimation of structure, one key aspect to enable useful operation is the ability to relocalise from the moment the system *shows up* in the environment or after loss of measurement. Furthermore, realistic environments such as a workspace where moving objects and occlusions are likely, require a recovery competence able to tolerate these changes.

http://dx.doi.org/10.5244/C.26.113

In the case of a system using 2D images as input, local image descriptors have been used for real-time relocalisation, *e.g.* [3, 19]. However, these methods rely on the scene having enough well-textured regions, which limits the areas of operation. RGBD cameras allow operation in non-textured areas and under other challenging conditions as more information per image location is available, as well as the output being less prone to be affected by shadows, uniformly coloured areas or motion blur due to the faster sampling rate on the depth channel.

This paper considers the evaluation of methods for relocalisation specifically for RGBD cameras and tested in small workspace scenes that have moving objects and/or minimal texture. Two existing local feature-based methods are considered and a new method proposed that uses synthetically generated views within a regression framework and that is capable of estimating 6D camera pose at framerate. We also show some results of novelty detection of objects that were not in the original map and some results towards the goal of constant relocalisation as an alternative to conventional camera tracking.

The remainder of the paper takes the following layout. Section 2 introduces the relocalisation problem and the related work in this area. Section 3 introduces the datasets that we use for evaluating the relocalisation methods. Section 4 describes the new relocalisation method and its practical implementation details. Section 5 contains the comparison of the method against existing approaches and additional motivating examples. We conclude in Section 6 with a summary of the work.

2 Relocalisation Problem

In the context of RGBD cameras, relocalisation is the problem of determining the 6D pose of the camera with respect to an existing model of the environment, without any prior information about the pose in recent frames. This type of situation occurs when, for example, the camera loses measurements, perhaps due to a prolonged period of occlusion or erratic motion, or because it has exited and subsequently reappeared in the mapped region of the environment.

There are several key characteristics of relocalisation methods to consider: (i) *speed* to enable relocalisation during normal motion; (ii) *accuracy* to enable tracking to converge and resume successfully, or alternatively, to allow the use of constant relocalisation in place of conventional tracking; (iii) *scalability* to be able to handle a suitable size of map and range of viewpoints; and (iv) *robustness* to changes in the map, such as occlusion and dynamic objects, and to challenging environments, such as areas with limited texture or repeated geometric structure.

State of the art mapping methods are now able to construct dense textured models of environments. Some representations for these models include overlapping keyframes [15], surfel clouds [9], volumetric signed distance functions [14], and textured occupancy grids [12]. These representations are all capable of generating 3D RGB point clouds, or synthesising novel RGBD views over a wide range of viewpoints by projecting the underlying model into a virtual camera. This opens the possibility of using existing relocalisation and pose estimation algorithms from the point cloud and monocular camera literature.

Local feature methods rely on distinctive geometric [16] or visual [2, 6, 19] features extracted from the map, which can be repeatably matched to features extracted from the current camera view of the world. Typically, RANSAC [8] pose estimation is then used to robustly handle outliers and noise and compute an optimised pose. However, detection and

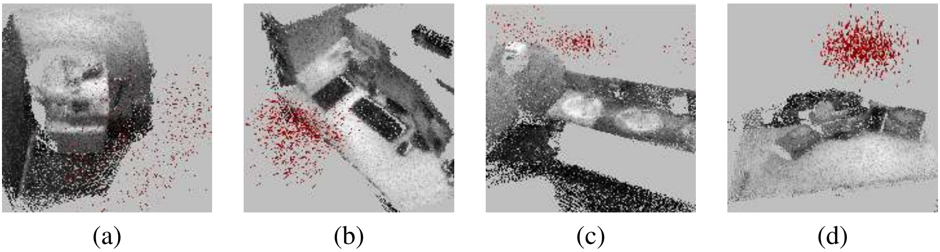

Figure 1: The four different maps in the dataset: (a) copier; (b) desktop; (c) basins; (d) bottles. Red arrows indicate the sampled poses of synthetic views used for relocalisation. The number of samples is $m = 1000$.

extraction of features can often be a relatively slow process. There can also be additional problems in practice, due to the difficulties of matching geometric features extracted from a 3D map to those from a 2.5D point cloud offered by the sensor, and also due to the sensitivity to viewpoint changes of visual features extracted on non-planar surfaces, though this can be mitigated to some extent by using synthetic views [10].

In contrast, view-based methods construct descriptors for different views of the map and match these to the current camera view. One simple but effective approach uses NCC matching of small, blurry 2D keyframe images and chooses the nearest neighbour to the current view as the estimated pose [11]. Another related approach uses color and orientation data to interpolate pose between densely sampled views [7]. Similarly, view-dependent geometric descriptors, such as VFH [17], can be generated from real and synthetic depth images and used in the same way. Yet another approach is particle filter estimation of pose using online synthesis of views from the map [12]. Typically, the comparison of views is relatively fast in these methods, but online view synthesis can be slow, particularly if graphics hardware is unavailable or if the underlying map representation is complex. This can pose difficulties when deciding which views to store or synthesise. Additionally, estimated pose accuracy is strongly related to the density and coverage of views, and occlusions and moving objects can make matching much more challenging.

We begin by introducing the textured occupancy grid datasets that we use to evaluate the relocalisation methods. Then, in the following sections, we present a new view-based method that estimates camera pose from a set of synthetic views. In contrast to related methods, such as [7, 11, 12], this method uses synthetic RGBD images to estimate full 6D pose at framerate using a regression framework.

3 Datasets

An RGBD SLAM system [5] was used to generate datasets for testing. The SLAM method is an extension of the quadrifocal visual odometry system proposed by Comport *et al.* [4], which minimises intensity and depth information to perform dense spatial matching between pairs of RGBD images. A textured occupancy grid representation of the scene at a resolution of 1cm is then generated by fusing depth and intensity images from multiple frames. However, the choice of SLAM system is not critical and other methods capable of rendering synthetic views, such as KinectFusion [14], could be used instead.

We created separate maps and test sequences in four different environments, which we describe as *copier*, *desktop*, *basins* and *bottles*. An Asus Xtion Pro Live RGBD camera was used as our sensor and the constructed maps are shown in Fig. 1. Each environment was chosen for its challenging conditions: the *copier* sequence includes dynamic elements, such as the opening of a paper tray, coarse geometric structure and low visual texture; the *desktop* sequence includes novel viewpoints that were not visited during the mapping phase, such as full rotation around the optic axis; the *basins* sequence contains occlusions during hand washing, repetitive geometric structure and low visual texture; and the *bottles* sequence involves the introduction of new objects and occlusions during a labelling and packing task.

Ground truth for the datasets was generated by running the RGBD SLAM system on the test sequences in tracking mode, i.e. without updating the existing map. This provided an estimated camera trajectory which can be compared against the poses estimated by the relocalisation methods. Frames where tracking failed were removed from the ground truth set. Relocalised position error is defined as the Euclidean distance between the ground truth and relocalised camera position, and relocalised orientation error as the minimum rotation angle between the ground truth and relocalised camera orientation quaternions.

The datasets also contain keyframes along the trajectory of the camera, captured during the mapping process, with a minimum distance criterion of 10° and 5.0cm to control their spacing. The poses of these keyframes are used to assist the sampling of the synthetic views used by our view-based relocalisation method. Additionally, the image content of these keyframes was used to extract the visual features used by the feature-based relocalisation method described in Section 5.

4 Regression over Synthetic Views

The relocalisation problem can be formulated as a minimisation problem, where the goal is to find the set of camera pose parameters $\mathbf{x} = [\mathbf{t}, \ln(\mathbf{q})] \in \mathbb{SE}_3$, that minimises the distance measure

$$\mathbf{x} = \arg\min_{\hat{\mathbf{x}}} \|\mathbf{I}_0 - \mathbf{I}(\hat{\mathbf{x}}, \mathcal{M})\|, \tag{1}$$

where $\mathbf{t}$ is a 3D position vector, $\mathbf{q}$ is a quaternion representing rotation, $\mathbf{I}(\hat{\mathbf{x}}, \mathcal{M})$ is the synthetic view generated from the map $\mathcal{M}$ at pose $\hat{\mathbf{x}}$, and $\mathbf{I}_0$ is the true RGBD image from the camera. The j-th RGBD image $\mathbf{I}_j = [\mathbf{u}_j, \mathbf{v}_j, \rho_j, \mathbf{c}_j]$ is composed of n pixels, where $[u_{ji}, v_{ji}]$ are the image coordinates, ρ_{ji} is the depth value, and c_{ji} is the colour of the i-th pixel. For our purposes it is sufficient to store grey intensity values in $\mathbf{c}$, since this avoids some problems caused by colour changes under different lighting and reduces the dimensionality of the problem without losing too much information about the appearance of the scene.

We treat the estimation of $\mathbf{x}$ in Eq. 1 as a general regression problem over a set of m synthetic views $\mathbf{I}_j$ and their poses $\mathbf{x}_j$, for $j = 1 \ldots m$. Using the Nadaraya-Watson estimator [13, 18], we can approximate the camera pose $\tilde{\mathbf{x}}$ from the set of synthetic views as

$$\tilde{\mathbf{x}} = \frac{\sum_{j=1}^{m} \mathbf{x}_j K(\|\mathbf{I}_0 - \mathbf{I}_j\|/h)}{\sum_{j=1}^{m} K(\|\mathbf{I}_0 - \mathbf{I}_j\|/h)}, \tag{2}$$

where K is a kernel function centred on each sample with bandwidth h. In this case, we opt for a Gaussian kernel, such that

$$\tilde{\mathbf{x}} = \frac{\sum_{j=1}^{m} \mathbf{x}_j d_j}{\sum_{j=1}^{m} d_j}, \tag{3}$$

$$d_j = exp\left(-\frac{1}{n\alpha}\sum_{i=1}^{n}\left(\frac{(c_{0i}-c_{ji})^2}{\sigma_{ci}^2}+\frac{(\rho_{0i}-\rho_{ji})^2}{\sigma_{\rho i}^2}\right)\right), \tag{4}$$

where σ_c and σ_ρ are vectors of standard deviations in the intensity and depth computed per pixel over all of the sample views $\mathbf{I}_j$, for $j = 1\ldots m$, and α is a scaling factor that controls the smoothness of the regression. The estimate $\tilde{\mathbf{x}}$ is therefore a normalised weighted sum, where the contribution of each sample view is determined by the normalised Euclidean distance between the sample view and the current camera view.

4.1 Synthetic View Sampling

One of the key differences between our work and previous relocalisation systems is that, enabled by the recovered 3D map, we can generate novel synthetic views that have not been visited by the system during mapping and that are considered likely poses where relocalisation will be needed. This enhances the power of the sampling used by the regression framework but introduces the issue of knowing which views should be generated and how.

Synthetic views are generated by raycasting from a virtual camera into the textured occupancy grid map. The raycasting method finds the depth for each pixel by searching for the maximum occupancy probability value along the ray. Linear interpolation along the ray direction provides sub-voxel accuracy. Intensity values are read directly from the voxel and are not interpolated. However, every synthetic view image is normalised for intensity according to

$$c_{ji} \leftarrow \frac{c_{ji}-\bar{c}_j}{\sigma_{c_j}}, \tag{5}$$

where $\bar{c}_j$ is the mean intensity and σ_{c_j} is the standard deviation of intensities in the j-th image. Generation of synthetic views can be relatively slow, taking $\sim$ 30ms per view on a 2.66GHz Intel Core2 CPU. But, importantly for our case, the synthetic views can be generated in advance and offline or even, though not done here, generated on a GPU. Except where specifically noted, we have used a resolution of 80×60 pixels in this paper.

The selection of sample poses for the synthetic views is an important part of the relocalisation algorithm and could be optimised according to the expected camera motion and required accuracy. The computational and memory costs increase linearly with the number of sampled views, so the correct balance needs to be found between densely sampling every possible pose around the map and selectively choosing poses in the most likely areas.

Here we have adopted the approach of randomly sampling poses around the keyframe poses captured along the mapping trajectory (c.f. Section 3), but it would be equally simple to apply the same approach to any other appropriate trajectory. For each of the m sampled synthetic views, a keyframe pose is randomly selected and a random Gaussian perturbation with 10° and 5.0cm standard deviation is applied. Synthetic views are resampled if fewer than 50% of the pixels intersect with the map. During the sampling process, the statistics for σ_c and σ_ρ, required by the regression algorithm, are also calculated and stored.

The discussion of sampling strategy underlines the importance of covering enough of the operational environment, but also implicitly questions the use of a *more invariant than needed* approach, such as using generic local feature descriptors. Note that it is possible for relocalisation to initialize a tracker to explore new areas and, if needed, it is trivial to extend the set of synthetic views used for regression at run time.

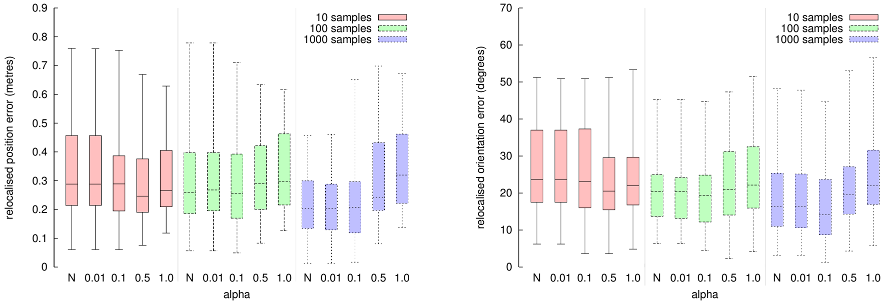

Figure 2: Effect of smoothing factor α and number of samples on relocalised pose error. An α value of 'N' indicates nearest neighbour matching of the current view to the synthetic view samples. Whiskers represent 5^{th} and 95^{th} percentile. Results are from the *copier* dataset.

Figure 1 shows example sampled poses with a sample count of $m = 1000$. Our approach of sampling around keyframes provides efficient coverage of the most likely viewpoints. For example, the *bottles* dataset was captured with a wearable camera and both the mapping and test sequence trajectories have minimal translational motion characteristic of a real assembly task on a workbench. This results in a tight distribution of sample poses on this map.

4.2 Parameter Selection

Smoothing Factor and Sample Size An advantage of the regression method is that only the smoothing factor α needs tuning. Figure 2 shows relocalisation error results for different values of α and numbers of samples m on a subset of the *copier* dataset. Increasing the density of samples improves accuracy ($\sim 20-30\%$ improvement in median accuracy for 1000 samples versus 10 samples), and increasing smoothing is more effective when the number of samples is low and regression can bridge the gaps in the sampling space ($\sim 10\%$ improvement in median accuracy versus nearest neighbour for $\alpha = 0.5$ at $m = 10$ samples). We have chosen to use $\alpha = 0.1$ and $m = 1000$ samples for the rest of this paper.

Resolution and Computational Cost Figure 3 (top row) shows the effect of resolution on relocalisation error. Although we use a resolution of 80×60 for our other results, there appears to be considerable scope to reduce resolution without significantly affecting accuracy. At a resolution of 80×60, each distance measurement between the current view and a sampled synthetic view takes $\sim 60\mu$s, enabling regression over 1000 samples in 60ms on a 2.66GHz Intel Core2 CPU. Reducing the resolution to 20×15 reduces these timings an order of magnitude to $\sim 4\mu$s and 4ms respectively, and also reduces the memory footprint for their storage substantially by about 94%. This potentially enables more samples to be stored and tested or the algorithm to be run on mobile platforms with lower computing power.

Effect of Fusion We have also tested the efficacy of the fusion of depth and intensity information in the distance measure defined in Eq. 4. Figure 3 (bottom row) shows relocalisation error results for the system using depth-only, intensity-only and fused distance measures. The fused distance measure provides a small improvement in median pose accuracy but significantly reduces the spread in the position error results. Note that the intensity-only results are also good enough to suggest that the synthetic views with regression could be used for relocalisation of monocular cameras observing the same map.

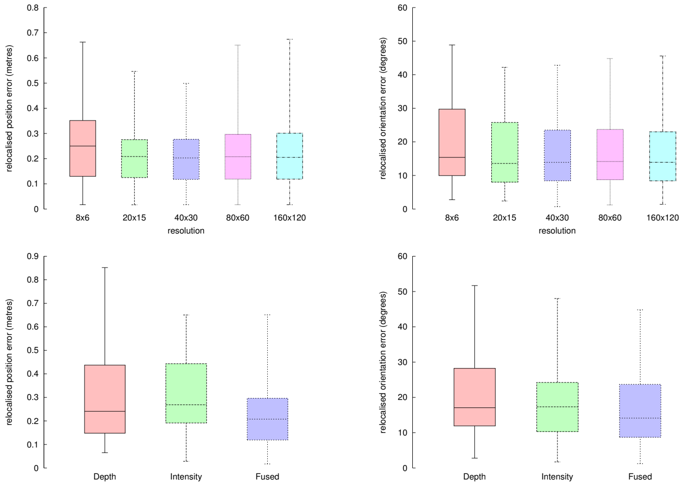

Figure 3: Effect of (top row) resolution and (bottom row) depth and intensity fusion on relocalised pose error. Whiskers represent 5th and 95th percentile. Results are from the *copier* dataset.

5 Experiments and Results

5.1 Comparison with Feature-based Methods

The performance of the proposed relocalisation method has been evaluated by comparison against visual and geometric feature-based relocalisation methods on the datasets described in Section 3. We selected SURF [1] and FPFH [16] as descriptors because their publicly available implementations and performance make them good choices for a relocalisation system. We also compare against a naive use of the synthetic views, using the nearest neighbour to the input image.

The SURF relocalisation method uses the OpenCV 2.3.1 library implementation to find matches between the features extracted from the stored keyframes (c.f. Section 3) and the current view. Since the RGBD images provide depth information for both the stored keyframes and the current view, the RANSAC PnP algorithm can be used to estimate the relocalised 3D pose. We use a Fast Hessian threshold of 600 and run 100 RANSAC iterations with a maximum correspondence distance of 8 pixels. We extract features from images at a resolution of 640 × 480 pixels, obtaining up to 500 features per image, and use a k-d tree for approximate nearest neighbour feature matching. Visual descriptors like SURF are likely to struggle in minimal texture environments, but 2D descriptors are often used for relocalisation in 6D visual SLAM and thus worth including in our comparison.

A perhaps more suitable feature for RGBD cameras are 3D geometric feature descriptors. The FPFH relocalisation method uses the PCL 1.5 library implementation and SAC-IA [16] pose estimation (without ICP refinement). The FPFH features are extracted from point cloud models of the maps (generated from the textured occupancy grids by aligning synthetically generated views of the map and downsampling on a 0.05m voxel grid). These can then be matched to the FPFH features extracted from the current RGBD image point cloud (which is downsampled to the same voxel grid resolution). We use a normal radius of 0.15m and an FPFH radius of 0.3m and run 100 SAC-IA iterations with a minimum sample distance of

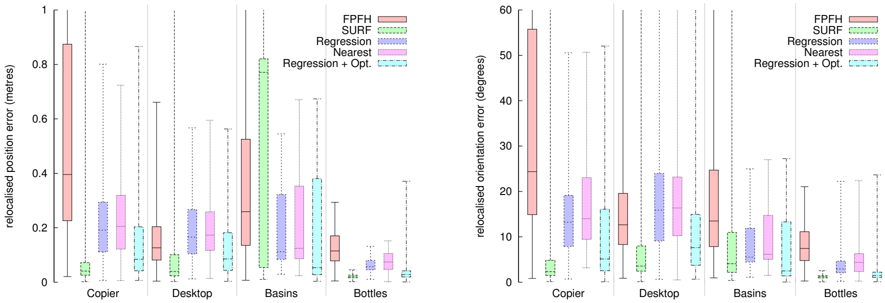

Figure 4: Ground truth error comparisons. 'Nearest' denotes nearest neighbour matching of current view to synthetic view samples. 'Regression + Opt.' denotes results after optimisation from regression pose estimate. Whiskers represent 5^{th} and 95^{th} percentile.

FPFH	SURF	Regression (80 × 60)	Regression (20 × 15)
∼ 620ms	∼ 255ms	60ms	4ms

Table 1: Approximate timings. SURF includes 180ms feature extraction, 15ms matching and 60ms RANSAC PnP pose estimation. FPFH includes 120ms downsampling and feature extraction, and 500ms SAC-IA pose estimation. Regression timings are for 1000 synthetic views without optimisation (10 optimisation iterations take 20ms).

0.35m and maximum correspondence distance of 0.1m. We extract FPFH features for every point in the downsampled cloud, obtaining up to 500 features per view.

The relocalisation error results of the comparison are presented in Fig. 4. The regression method performs consistently across the datasets. Initial accuracy is generally less than the best feature-based method, but running 10 iterations of the RGBD tracking optimisation (c.f. Section 3, [5]) causes convergence of the pose towards ground-truth and still at a fraction of the time demanded by SURF. This demonstrates that the synthetic view pose estimates are sufficiently accurate to resume tracking quickly. Typical timings obtained for the different methods are shown in Table 1.

Figure 5 shows common success and failure modes. The most common failure mode of the regression system occurs when the camera moves to a viewpoint outside of the set of synthetic view samples (e). Occlusion also sometimes causes problems (f). The SURF method performs worst in the low-texture *basins* dataset (g), and the FPFH method performs worst in the low-geometry *copier* dataset (h). All methods struggle with the structural ambiguity in the *basins* dataset (i). The FPFH and SURF results also exhibit varying performance due to the inherent randomness of the RANSAC estimation process. Increasing the number of RANSAC iterations reduces this to some extent, but further increases the computational requirements.

5.2 Constant Relocalisation for Segmentation

The speed and quality of the pose estimation using the synthetic views relocalisation method enables the possibility of using constant relocalisation as an alternative to conventional tracking of the camera. Wearable cameras often experience occlusions and erratic motion that cause tracking systems to fail, and a constant relocalisation approach avoids continuous switching between relocalisation and tracking. This could be particularly beneficial in appli-

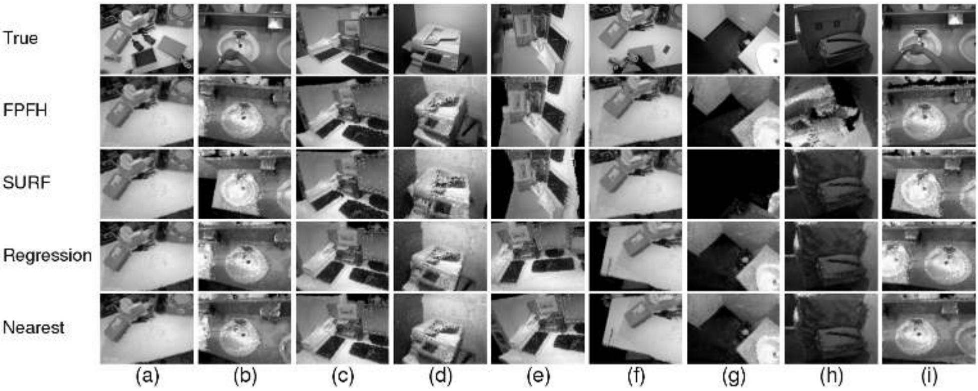

Figure 5: Selected examples: (a)-(d) successful relocalisation; (e)-(i) failure cases. See main text for details. Images show ground-truth camera view and synthetic views generated from relocalised poses. Videos of full sequences available from this paper's entry at www.cs.bris.ac.uk/Publications.

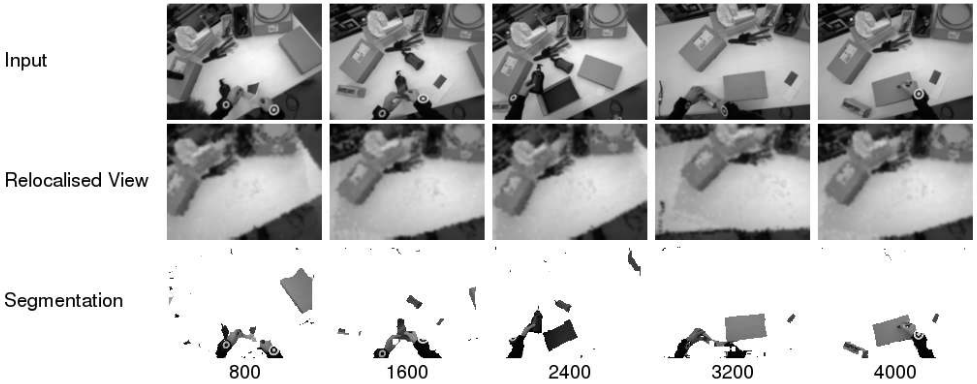

Figure 6: Segmentation using constant relocalisation pose estimation. Video of segmentation results available from this paper's entry at www.cs.bris.ac.uk/Publications.

cations such as workspace assistance, where dense tracking can be used to quickly segment objects from the background map, enabling efficient object detection and task monitoring.

Figure 6 shows an example of segmentation performed using our relocalisation method from a wearable camera on the *bottles* sequence using 20×15 pixels resolution views and without any further pose optimisation. The segmentation detects foreground pixels using a weighted distance measure on each pixel.

$$d_i = \tau((1-\lambda)(c_i - \hat{c}_i) + \lambda(\rho_i - \hat{\rho}_i), \tag{6}$$

where $\tau(r)$ indicates Tukey's robust M-estimator function, λ is a weighting parameter, (c_i, ρ_i) are the intensity and depth of the i^{th} pixel in the current camera image, and $(\hat{c}_i, \hat{\rho}_i)$ are the equivalent pixel values in the synthetic view at the relocalised camera pose. Segmentation results are fairly good, though some noise occurs around depth discontinuities. Challenging cases include small objects, due to image resolution, and planar objects, which rely purely on intensity difference for segmentation, e.g. the box lid in frame 2400.

6 Conclusions

The paper presents a system for 6D RGBD relocalisation at framerate using regression over a set of synthetic views generated from an informed estimate of possible camera viewpoints. The method is fast and accurate and copes with small changes to the environment and low texture workspaces. The performance of the system was demonstrated by comparison to visual and geometric feature matching relocalisation techniques on four test sequences including occlusions and moving objects. The speed and accuracy of the method enabled us to demonstrate it for the task of replacing camera tracking with constant relocalisation and use the pose for segmenting objects online.

Acknowledgements This work was supported by EU-FP7 (Project COGNITO, ICT-248290).

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: speeded up robust features. *Computer Vision and Understanding (CVIU)*, 110(3):346–359, 2008.

[2] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway. Robust real-time visual SLAM using scale prediction and exemplar based feature description. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[3] D. Chekhlov, W. Mayol-Cuevas, and A. Calway. Appearance based indexing for relocalisation in real-time visual SLAM. In *Proc. British Machine Vision Conf. (BMVC)*, pages 363–372, September 2008.

[4] A.I. Comport, E. Malis, and P. Rives. Real-time quadrifocal visual odometry. *International Journal of Robotics Research (IJRR)*, 2010.

[5] D. Damen, A.P. Gee, W. Mayol-Cuevas, and A. Calway. Egocentric real-time workspace monitoring using an RGB-D camera. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[6] E. Eade and T. Drummond. Unified loop closing and recovery for real-time monocular SLAM. In *Proc. British Machine Vision Conf. (BMVC)*, September 2008.

[7] M. Felsberg and J. Hedborg. Real-time view-based pose recognition and interpolation for tracking initialization. *Journal of Real-Time Image Processing*, 2(2-3):103–115, 2007.

[8] 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*, 24(6):381–395, 1981.

[9] 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 *Proc. Int. Symp. on Experimental Robotics (ISER)*, 2010.

[10] A. Irschara, C. Zach, J-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[12] J. Mason, S. Ricco, and R. Parr. Textured occupancy grids for monocular localization without features. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[13] E.A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

[14] 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 *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2011.

[15] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: dense tracking and mapping in real-time. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2011.

[16] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3d registration. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2009.

[17] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, October 2010.

[18] G.S. Watson. Smooth regression analysis. *Indian Journal of Statistics Series A*, 26(4): 359–372, 1964.

[19] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2007.