Image and Vision Computing 29 (2011) 524–532

Contents lists available at ScienceDirect

Image and Vision Computing

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

Keyframe-based recognition and localization during video-rate parallel tracking and mapping[☆]

R.O. Castle, D.W. Murray [*]

Active Vision Laboratory, Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

ARTICLE INFO

Article history:
Received 20 July 2010
Received in revised form 5 January 2011
Accepted 17 May 2011

Keywords:
Wearable vision
Augmented reality
Object recognition
Structure from motion

ABSTRACT

Generating situational awareness by augmenting live imagery with collocated scene information has applications from game-playing to military command and control. We propose a method of object recognition, reconstruction, and localization using triangulation of SIFT features from keyframe camera poses in a 3D map. The map and keyframe poses themselves are recovered at video-rate by bundle adjustment of FAST image features in the parallel tracking and mapping algorithm. Detected objects are automatically labeled on the user's display using predefined annotations. Experimental results are given for laboratory scenes, and in more realistic applications.

1. Introduction

Providing timely assistance to a user by augmenting live imagery with graphical annotations requires knowledge at each instant of the camera's pose relative to some scene-based coordinate frame, and knowledge of whether and where objects of interest occur in the scene. Geometric model-based approaches were the first to be sufficiently parsimonious to meet these requirements on modest hardware (*e.g.* [1–4]). Their drawback however was that the scene needed to be modeled visually and geometrically, but also simply and with little recourse to actual appearance information. This was a forlorn task, evidenced for example by the difficulty in visually tracking even basic manufactured objects using wireframes.

Recent years have seen substantial progress made towards more complete responses to both requirements. For the first, following the considerable advance made a decade ago in the off-line solution to the dense structure from motion problem [5–8], steady improvement has been made in on-line frame-rate solutions using both recursive techniques and batch methods. In these categories one might highlight the demonstration of single camera simultaneous localization and mapping (SLAM) [9,10], and the development of high quality visual odometry involving keyframes [11,12]. For the second requirement, the transformational advances in appearance-based recognition have shown that groupings of apparently non-descript low-level features allow recognition of individual objects and classes of objects [13–15]. Moreover, the same features used for recognition have a strong foundation in geometry, allowing them to double up as features for localization.

This paper is concerned with a combination of these approaches applied to augmented reality (AR). Using the parallel tracking and mapping method [16], the camera is tracked from frame-to-frame over the short term, and its pose along with the positions of the 3D map points optimally recovered at keyframes by bundle adjustment [17,8]. At the same time, known objects are detected and recognized from Lowe's SIFT descriptors [13], but computed only in the keyframes. The objects are located by determining the SIFT features' 3D structure by triangulation — that is, by treating the keyframe camera poses determined by bundle adjustment as known and fixed quantities. The method, of which a brief précis appeared in [18], is applied to assistive applications in wearable vision.

1.1. Context

Renewed interest and progress in the use of a single camera for frame-rate recovery of structure and camera motion in unconstrained and markerless environments owe much to Davison's work on monoSLAM [9,10]. The utility of the sparse structural map it delivered was demonstrated for AR in [19] and wearable computing in [20], although in both the relationship between the AR and the underlying scene was inferred by the user.

Lowe's SIFT [13] provides robust feature detection and description, facilitating matching over a wide range of scales and under changes in orientation and illumination. The AR system developed by Gordon and Lowe [21] used SIFT to build 3D models, replacing hand-crafted 3D CAD models and removing the need for the modeler to guess

[☆] This paper has been recommended for acceptance by Henrik Iskov Christensen.

[*] Corresponding author. Tel.: +44 1865 273106.
E-mail addresses: bob@robots.ox.ac.uk (R.O. Castle), dwm@robots.ox.ac.uk (D.W. Murray).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.05.002

which features were best to track. The 3D models were built off-line using bundle adjustment, and the system used on-line to recover the camera pose relative to the detected object. While their work was aimed primarily at model tracking, our interest here is in building maps of an environment and automatically recognizing and localizing objects *within* these mapped environments, allowing AR elements to be placed upon them. The detection of objects provides an additional layer of information to the generated maps, rather than having a system that only tracks when an object is detected.

The immediate precursor to this work employed monoSLAM and SIFT to recognize and localize objects within a 3D map [22]. However, a disadvantage of extended Kalman filter-based SLAM is that the size of map, and hence the extent of camera exploration, is constrained by the EKF's $O(M^2)$ complexity in the number of map points: on current portable processors 30 Hz frame-rate operation is maintained for map sizes up to $M \approx 10^2$ points. Indeed the problem was exacerbated in [22] by the object localization process. Whenever an object was recognized and localized, the positions of three points on the object were incorporated as additional measurements into the SLAM filter. While this successfully married object and map structure, the addition of yet more objects had the consequence of further restricting the extent of the map.

Although the broad aim here is similar, the method proposed differs in several significant respects. First, monoSLAM is replaced by the keyframe-based method of parallel tracking and mapping (PTAM) [16]. In a single map of M map points and K keyframes PTAM has approximately $O(MK^2)$ complexity, allowing maps of several thousands of points to be managed at frame rate. The complexity of bundle adjustment eventually becomes cubic in K, but here is dominated by populating Jacobian matrices. Another difference is the use of multiple maps. Although a single map in PTAM has greater extent than those in monoSLAM, in typical use it still covers only some 30 m of linear visual interest. Several ways of extending the mapped area have been explored, such as sub-mapping [23,24] and constant time relative frame bundle adjustment [25,26]. Here, however, a pragmatic approach – and one entirely suited to cameras held by an intelligent user – is to build multiple maps and allow a relocalizer to detect when mapped areas are entered or left. Unlike in sub-mapping for robot navigation [23], in wearable vision there is no imperative to provide detailed geometrical transformations between separate maps.

As significant are the changes made in the way objects are handled. In [22], objects were located from a single view using the known size of the object, and the location fed back to the EKF. Here, once an object is detected and recognized, its structure and location are computed using multi-view triangulation, keeping the camera poses in the keyframes fixed. The objects become movable augmentations to the map rather than being embedded in it. Another difference is that the motion model in monoSLAM's EKF required the depth/speed scaling ambiguity to be resolved beforehand by calibration, which in turn required the object's size to be predetermined to avoid a conflict of scales. In the current method, because the recovery of object structure uses camera positions fixed by PTAM, and there is no feedback, there is no longer need to define the size of objects. A further difference occurs in the frequency and timing of processing. In [22], detection and localization of objects occurred continually, so that even when the camera was stationary the same view would be re-processed and objects re-located. This was wasteful of the limited sampling opportunities available. Instead we exploit the dense and well separated keyframes that are used to build the map to search the entire mapped environment for known objects.

The remainder of the paper is organized as follows. Section 3 describes how objects are detected, recognized and located from keyframe images. The results of experiments are given in Sections 4 and 5, and include the method's application in three AR scenarios. Closing remarks are made in Section 6. First, however, we provide an overview of the camera tracking and mapping method.

2. Camera tracking and scene mapping

The method of object detection and localization proposed here is actually built upon PTAMM [27], a multiple map and multiple camera extension to the PTAM algorithm. While the ability to build multiple maps will be used in the experiments, this detail is unimportant at this stage.

Maintaining track of the camera's pose, summarized in the first column of Fig. 1, is the most pressing task and is run at every frame, treating the 3D map as fixed. Once acquired, each image is sampled into a 4-level pyramid, and corner features found at positions $\boldsymbol{x}_i$ at each scale by the FAST-10 detector [28,29].

A constant velocity filter (but one for which lack of measurement forces the velocity to decay) provides a prior camera pose $\boldsymbol{\pi}$, and potentially visible map points $\boldsymbol{X}_i$ are projected into the image at $\boldsymbol{x}(\boldsymbol{\pi}, \boldsymbol{X}_i, \boldsymbol{C})$, where $\boldsymbol{C}$ holds the known camera intrinsics and lens distortion parameters. A few tens of matches are sought at coarse scale to estimate the pose using robust minimization, followed by up to 1000 matches at fine scale to re-optimize pose. Both optima are found using some ten iterations of Gauss–Newton [30] with a Huber M-estimator cost function (*c.f.*[31]) based on the summed reprojection error. A determination is made on the basis of spatial coverage of the scene as to whether a new keyframe should be offered to the mapping thread. The criterion used is that the distance of the camera from any existing keyframe exceeds the average scene depth. (This translational criterion has recently been justified using entropy reduction metrics by Holmes [32], who also suggests a further angular one.) Lastly, the new camera position is used to re-render graphics onto the current image.

The mapping process, the second thread of Fig. 1, runs continually, optimizing all the map points $\{\boldsymbol{X}_i\}$, $i = 1 \ldots I$ and all but one keyframe camera poses $\{\boldsymbol{\pi}_k\}$, $k = 2 \ldots K$ in a bundle adjustment, using Levenberg–Marquardt [33] to minimize the reprojection error wrapped in a Tukey M-estimator cost function [34].

As noted earlier, when spatial coverage demands, the tracking thread will offer a new keyframe to the map-maker. All map points are projected into the keyframe using its pose estimated from the tracking process, and matched where possible. New map points are instantiated by seeking unmatched features in regions away from matched features, and performing epipolar search for matches in a neighboring keyframe. Any matches found are triangulated to yield new 3D map points. After the new keyframe has been added, a local bundle adjustment is performed involving the latest keyframe and its four nearest neighbors, along with all map points observed by them. Then the process continues with a full bundle adjustment of all keyframes and map points. The local adjustment helps reduce the overall processing cost.

To initialize the map at the outset, the user chooses a keyframe pose $\boldsymbol{\pi}_1 \equiv [\mathrm{R}_1|\boldsymbol{T}_1] = [\mathrm{I}|\mathbf{0}]$. The camera is moved to a new position with care to allow features to be tracked in the image alone, and this is chosen as the second keyframe. Nistér's relative pose algorithm [35] determines $\boldsymbol{\pi}_2 = [\mathrm{R}_2|\boldsymbol{T}_2]$. Pairs of matching FAST corners from the two images are then used to triangulate the initial set of 3D scene points. Hardcoded is the assumption that $|\boldsymbol{T}_2 - \boldsymbol{T}_1| \approx 0.1$ m so that a reasonable though arbitrary scale can be applied to depth and speed.

Examples of the maps are given later, in Figs. 2b, 4c, and so on.

3. Objects and keyframes

The third thread of Fig. 1 summarizes the handling of objects: their detection, recognition and localization. The thread's computation is all but independent of tracking and mapping, as it uses SIFT features throughout rather than FAST, and its outputs are augmentations to the 3D map and do not influence the map's evolution.

Furthermore, the thread does not use the image stream directly, but processes only keyframe images, and not necessarily in time order.

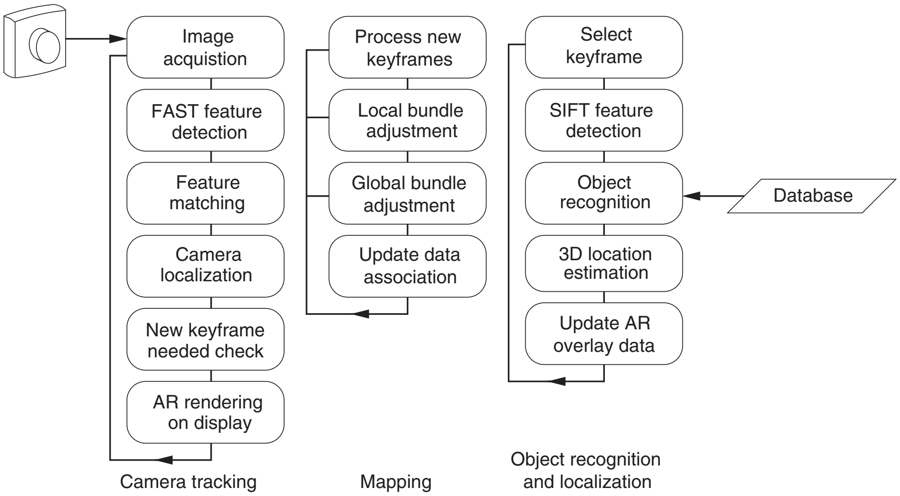

Fig. 1. The tracking and mapping threads of PTAMM, and the object recognition and localization thread developed here. The object thread uses only keyframe images and camera poses.

3.1. Keyframe selection for object detection

The near independence of recognition and mapping allows freedom in the selection of which keyframe to process next. One obvious order would be exactly that in which they are added to the map. However, when the camera explores a new area of the scene, keyframes are added more rapidly than the object recognition thread can manage, leading to a backlog. In this case another approach would be always to process the most recently arrived keyframe first, and hope that there is time later to clear the backlog. However, there are considerations that make these poor stratagems. First is that the thread must not only detect objects, but also localize them; next is that two keyframes containing the same detected object are required for localization; and, last, providing information on the area where the camera is looking currently is a priority.

Keyframes are therefore considered in pairs — the first processed is that keyframe whose position and orientation are closest to the camera's current pose; and the second is that which is most visually similar to the first. To assist the search for this pair, whenever a keyframe is added to the map the map-maker records which keyframe already in the map is most similar to the new one. This becomes its parent, and the parent also records that this new keyframe is its child, forming a bidirectional tree: for examples see Figs. 2b and 4b. The similarity measure used is the number of map

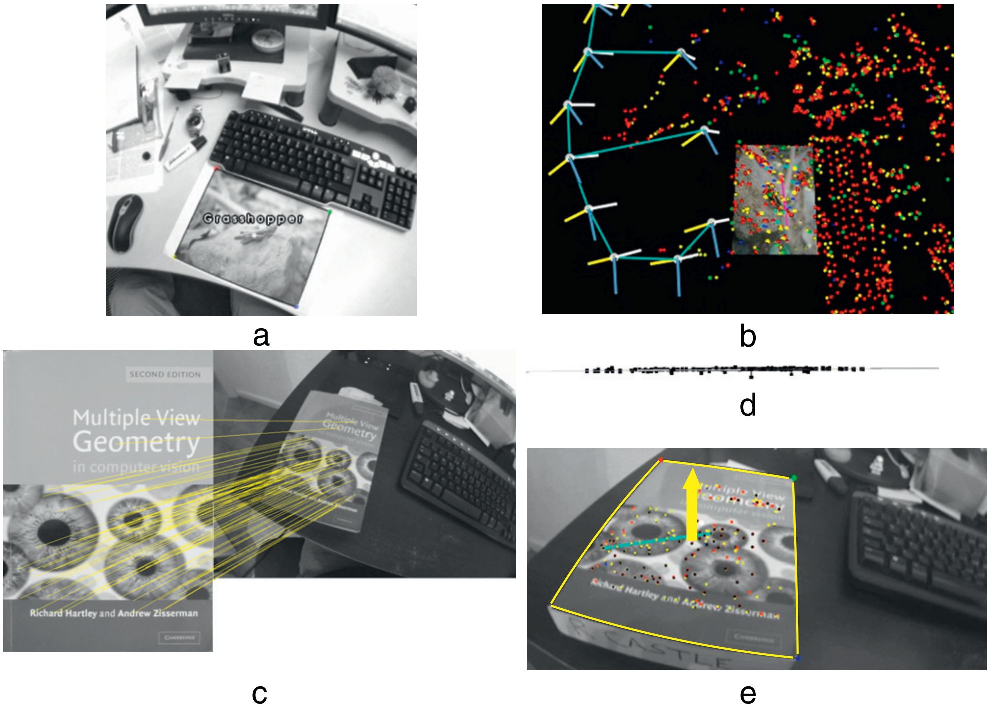
a
b
c
d
e
Grasshopper
SECOND EDITION
Multiple View Geometry in computer vision
Richard Hartley and Andrew Zisserman

Fig. 2. (a) A still cut from a desktop sequence, and (b) part of the map showing reconstructed 3D points and keyframes. Note that the bidirectional tree linking keyframes do not take on a linear structure. (c) Recovered SIFT keypoints matched to those in the database. (d) An edge-on view of their 3D triangulation, and (e) the located object. For clarity the overlaid graphics in (e) have had their colors changed. (See note at end for location of video material.)

points that the two keyframes have in common. This is an efficient search as each map point holds a list of keyframes in which it has been observed. If the first keyframe of the pair contains no objects, or any found objects are already localized, or all the relatives are already processed, then the thread defaults to processing the most recent unprocessed keyframe.

In a keyframe k selected for processing, SIFT descriptors and their locations $(\boldsymbol{\sigma}_k^l, \boldsymbol{x}_k^l, l=1\ldots L_k)$ are extracted from its image and are stored in the keyframe structure. These keypoint descriptors are compared with those in a database using Beis and Lowe's approximate best-bin-first modification of kd-trees [36]. If the number of keypoints matched between the keyframe image and any given object's database entry exceeds a threshold, that object is flagged as potentially visible.

3.2. Object reconstruction and localization

Once an object is deemed visible in two or more keyframes, its location is determined by triangulation from the locations of SIFT features matched across keyframes. The keyframes are treated as fixed, using their poses optimized in PTAMM's bundle adjustment. The reconstruction is at first quite general, but, when appropriate, reconstructed points can be fitted to the underlying shape of the model.

With just two views, an algebraic residual is minimized using a linear method [37]. Modulo scale, the two (distortion-corrected) observations of the homogeneous scene point $\boldsymbol{X}$ are $\boldsymbol{x}_{a,\ b}=\mathrm{P}_{a,\ b}\boldsymbol{X}$, where the projection matrix $\mathrm{P}_{a,\,b}=\mathrm{K}[\mathrm{R}_{a,\,b}|\boldsymbol{T}_{a,\,b}]$ for each view involves known intrinsic calibration and keyframe poses. Combining these,

$$\mathrm{A}\boldsymbol{X} = \begin{bmatrix} x_a\boldsymbol{p}_{a3}-\boldsymbol{p}_{a1} \\ y_a\boldsymbol{p}_{a3}-\boldsymbol{p}_{a2} \\ x_b\boldsymbol{p}_{b3}-\boldsymbol{p}_{b1} \\ y_b\boldsymbol{p}_{b3}-\boldsymbol{p}_{b2} \end{bmatrix}\boldsymbol{X} = \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{p}_{ar}$ is the r-th row of P_a, *etc.*, and the residual is minimized when $\boldsymbol{X}$ is, up to scale, the column of V corresponding to the smallest singular value in the singular value decomposition $\mathrm{UDV}^\top \leftarrow \mathrm{A}$. As more observations are added, Levenberg–Marquardt is used to minimize error in the image: using the result from Eq. (1) as a starting point, the inhomogeneous $\boldsymbol{X}$ is found as

$$\boldsymbol{X} = arg\,\underset{\boldsymbol{X}^*}{min} \sum_k ||\boldsymbol{x}_k-\boldsymbol{x}(\boldsymbol{X}^*, \mathrm{P}_k)||_2. \tag{2}$$

Although the keyframe camera poses are treated as fixed during a particular triangulation, if any camera pose changes during later bundle adjustments by PTAMM, all objects observed within that keyframe are relocalized using Eq. (2).

The i-th object's database entry comprises views I from known poses $\boldsymbol{\pi}$ around the object, 3D keypoints $\boldsymbol{U}$ on the object, their corresponding projections $\boldsymbol{u}$ and SIFT descriptors $\boldsymbol{\sigma}$, and two sets of 3D points A and B which indicate the positions of AR graphical annotations to be displayed to the user and the object's boundaries:

$$\begin{aligned} O_i = \Big\{ & \left\{I_{v,}\boldsymbol{\pi}_v\right\}_{v=1\ldots V_{i,}} \left\{\boldsymbol{U}_j\right\}_{j=1\ldots J_{i,}} \ \left\{\boldsymbol{u}_{k,}\boldsymbol{\sigma}_{k,}v_{k,}j_k\right\}_{k=1\ldots K_{i,}} \\ & \left\{A_{a,}\text{“AR − markup”}\right\}_{a=1\ldots A_{i,}} \ \left\{B_b\right\}_{b=1\ldots B_i}\Big\}. \end{aligned} \tag{3}$$

Quantities v_k and j_k point to the view and 3D feature giving rise to measured feature k.

In the absence of noise, the positions of keypoints $\boldsymbol{U}=(\boldsymbol{U}, V, W)^\top$ on the database object are within a similarity transformation of the corresponding $\boldsymbol{X}=(X, Y, Z)^\top$ recovered on the actual object. Treating both as inhomogeneous vectors the scaling, rotation and translation between them is found by minimizing

$$\sum_{i \in \text{matches}} |\boldsymbol{X}_i - s\mathrm{R}\boldsymbol{U}_i - \boldsymbol{T}|^2 \tag{4}$$

using the closed-form solutions to the absolute orientation problem of Horn et al. [38] and Faugeras and Hébert [39]. Denoting the centroids of the matched $\boldsymbol{U}_i$ and $\boldsymbol{X}_i$ as $\overline{\boldsymbol{U}}$ and $\overline{\boldsymbol{X}}$, and writing $\hat{\boldsymbol{U}}_i = (\boldsymbol{U}_i - \overline{\boldsymbol{U}})$ and $\hat{\boldsymbol{X}}_i = (\boldsymbol{X}_i - \overline{\boldsymbol{X}})$, the optimal rotation (represented as a unit quaternion q) is the eigenvector corresponding to the largest eigenvalue of a 4×4 matrix $\sum_i \mathrm{M}_i$ with

$$\mathrm{M}_i = \begin{pmatrix} 2\hat{\boldsymbol{U}}\cdot\hat{\boldsymbol{X}} & \left(\hat{V}\hat{Z}-\hat{W}\hat{Y}\right) & \left(\hat{W}\hat{X}-\hat{U}\hat{Z}\right) & \left(\hat{U}\hat{Y}-\hat{V}\hat{X}\right) \\ \left(\hat{V}\hat{Z}-\hat{W}\hat{Y}\right) & 2\hat{U}\hat{X} & \left(\hat{U}\hat{Y}+\hat{V}\hat{X}\right) & \left(\hat{U}\hat{Z}+\hat{W}\hat{X}\right) \\ \left(\hat{W}\hat{X}-\hat{U}\hat{Z}\right) & \left(\hat{U}\hat{Y}+\hat{V}\hat{X}\right) & 2\hat{V}\hat{Y} & \left(\hat{V}\hat{Z}+\hat{W}\hat{Y}\right) \\ \left(\hat{U}\hat{Y}-\hat{V}\hat{X}\right) & \left(\hat{U}\hat{Z}+\hat{W}\hat{X}\right) & \left(\hat{V}\hat{Z}+\hat{W}\hat{Y}\right) & 2\hat{W}\hat{Z} \end{pmatrix} \tag{5}$$

where the subscript i after each letter inside the matrix is omitted to save space. The optimal scale and translation are

$$s = \sqrt{\left(\sum_i |\hat{\boldsymbol{X}}_i|^2\right) / \left(\sum_i |\hat{\boldsymbol{U}}_i|^2\right)} \tag{6}$$

and

$$\boldsymbol{T} = \overline{\boldsymbol{X}} - s\,\mathrm{R}(\mathrm{q})\overline{\boldsymbol{U}}, \tag{7}$$

where R(q) is the rotation matrix derived from the unit quaternion.

3.3. Specialization to planar objects

For the planar objects experimented with here, a number of specializations are possible. First, the database entry no longer requires multiple views and fully 3D positions. Instead we use a single frontal image and locations are all of the form $\boldsymbol{U}=(U, V, 0)^\top$. Secondly, when matching between observed features and database features, *single* view constraints can be invoked to reduce mismatches. For planar model parts we remove outliers by using RANSAC [40] to estimate the homography between the in-plane database feature positions and the keyframe feature positions, and determine potential visibility from the cardinality of the consensus set of inliers. This robust fitting is merely used as a method of segmentation, not localization, and the homography itself is discarded. A third possibility when the underlying object has known (and simple) surface shape is to clean up the 3D structure by robust fitting to that shape.

3.4. Objects and multiple maps

Object localization is carried out within the map being used by the camera at the time. Each map is independent and its information is private, so that a particular object may appear in more than one map with an independent location in each. Any difference in location could be irksome for the user if the camera were to relocalize repeatedly from one map to another at their boundary. Two practical steps reduce the likelihood of this occurring. First, as noted earlier, our maps tend to be of discrete places of interest, not continuous maps for navigation. Second, if two maps A and B do abut, their boundary needs not be sharp. The overlapping region introduces hysteresis, and the transition (when the camera loses track) traveling from map $A \rightarrow B$ will not coincide with that from $B \rightarrow A$.

Table 1
Average times for tasks in the object recognition and localization thread for the book and posters experiments.

Processing	Average time (ms)	
Tasks	Book (Fig. 2)	Posters (Fig. 3)
Keyframe selection	1	1
SIFT feature extraction	1530	1747
Database matching	198	287
Outlier rejection	34	53
Reconstruction	5	3
Object localization	3	2
Total	1770	2093

4. Implementation and results

The system is implemented in C++, and the results reported here were obtained when running under Linux on a 2.20 GHz Intel Dual Core processor.

As an illustration of the output from the tracking and mapping threads, Fig. 2a shows a still from an experiment where the user moved the camera around a desktop scene, and Fig. 2b gives a view of part of the adjusted 3D point map and keyframe positions from a viewpoint to the right hand side of the keyboard. The links between keyframes represent the bidirectional tree use to quicken keyframe pair selection.

Fig. 2c shows the frontal view of the database object. Of its 1245 keypoints, some 67 keypoints have been matched to the particular keyframe image. Outlying matches were filtered using RANSAC. The object was found visible in 29 keyframes in this particular sequence. Fig. 2d shows the edge-on view of the 3D triangulation with its high degree of planarity, and the object located in the scene from the user's viewpoint. In all, some 224 database keypoints were observed in the 29 keyframes, of which 191 were localized and all were classified as inliers. The average times taken for each of the recognition and localization stages are shown in the middle column of Table 1. It can be seen that the SIFT processing dominates, followed by database matching.

The rendering process used to generate views for the user, as in Fig. 2d *etc.*, takes account of lens distortion by first undistorting the current camera image and rendering it as a background, overlaying graphical elements, then distorting the entirety back so that the camera image has its original form.

4.1. Multiple objects

A second experiment involves a larger number of objects. The database used contains 16 objects with a total of 31,910 keypoints, and all 13 observed objects are successfully recognized and localized. Fig. 3a shows the final frame from the sequence, with the detected objects outlined. The half above the diagonal shows the user's view with the AR labeling on each detected object, and half below shows the keypoints that have been localized. Fig. 3b shows three views demonstrating how one of these objects becomes better localized as further measurements are included in Eq. (2).

Fig. 3c, d shows perspective graphics of the recovered map with the added objects from above and from a general viewpoint. The individually located objects show collective coplanarity. It should be noted that placing the objects together on the wall plane does not affect the individual localizations, but merely gives an opportunity to examine collective quality. Fitting a plane to all objects and scaling the results to the known size of the scene show that the standard deviation across the entire scene about the zero mean is some 0.02 m.

The rightmost column of Table 1 shows the timings for this experiment. The database used is around 25 times larger than the previous experiment but, due to the best-bin-first lookup, the average search time increases only by some 45% to 287 ms.

5. AR applications

5.1. Art gallery

In this experiment the system was used to identify paintings in a gallery. The gallery database has 37 paintings with some 75,000 features. PTAMM's multiple-map capability was used, and a separate map built for each of the gallery walls.

On the hardware used, we find the limit to a single map's size as some 20, 000 points viewed in 150 keyframes. However, the size must

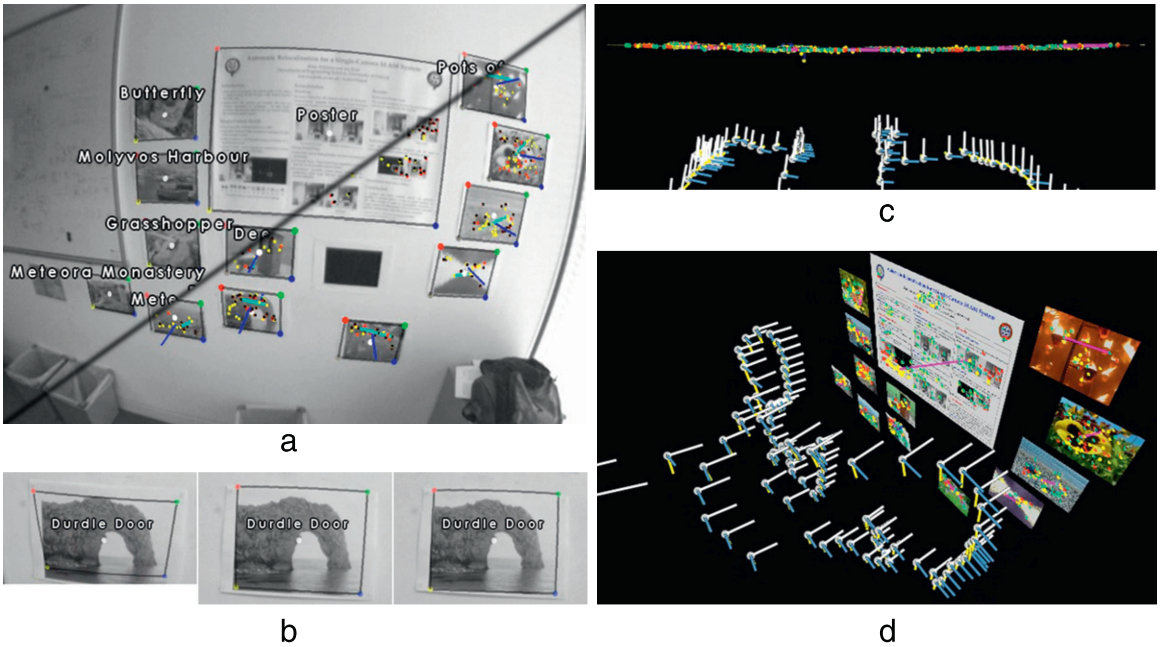

Fig. 3. (a) A hybrid view showing (top) the AR overlays seen by the user, and (bottom) details of keypoints and the object normals. All 13 items are recognized and located, and lie within the plane of the wall within experimental error. (b) The improvement of localization as further keypoint matches are added to the triangulation of the object structure. (c, d) Overhead and general graphic views of the keyframe positions and recovered objects. (See note at end for location of video material.)

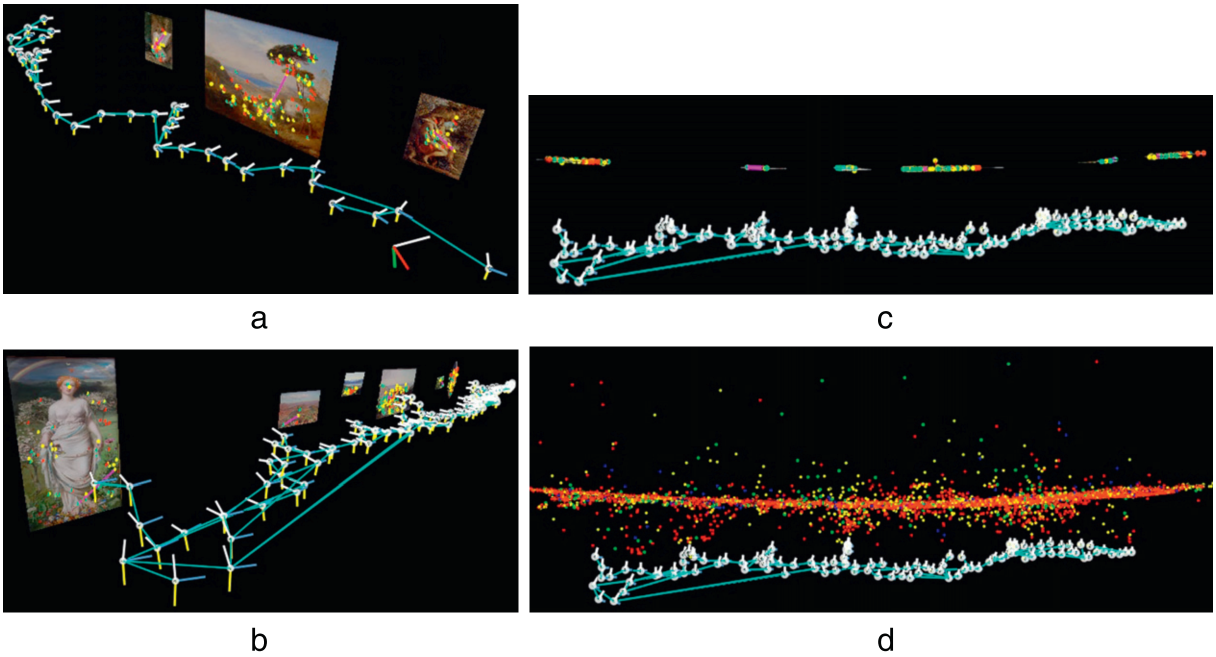

Fig. 4. (a, b) The keyframe poses and paintings recognized and located in two of the several maps made within a gallery. Using the example map in (b), overhead views show in (c) the 3D points recovered from FAST matches and bundle adjustment in PTAMM and in (d) the 3D points triangulated from SIFT matches associated with the objects. (See note at end for location of video material.)

be reduced if a high rate of exploration of unmapped areas is required — the greater the rate of keyframe addition, the greater the rate of bundle re-adjustment required. Holmes [32] has implemented automated sub-mapping for PTAM, but here maps are initiated by the user.

3D views showing the detected paintings, the keyframes, and the tree structure linking the keyframes, are given in Fig. 4a, b. Fig. 4c, d shows overhead views of the map of Fig. 4b with and without PTAMM's map points. The paintings are well located within the center of the point cloud. Fig. 5 shows a recognized painting with its AR label detailing the painting's title and artist [41].

Once maps have been constructed and objects detected, the system can be placed in a read-only mode which prevents further addition of keyframes, and allows a user to explore without accidentally corrupting the maps. When the user leaves one mapped area, the system becomes lost and attempts to relocalize within one of the maps by comparing heavily sub-sampled camera images with similarly treated versions of the keyframe images [27]. When a possible match is found, an attempt is made to track within that map from the putative current pose. If successful for a few frames, tracking continues, otherwise relocalization is repeated. In this experiment, relocalization into all maps was successful, but the search was protracted. The repeating pattern of the gallery wallpaper evident in Fig. 5 caused considerable visual aliasing.

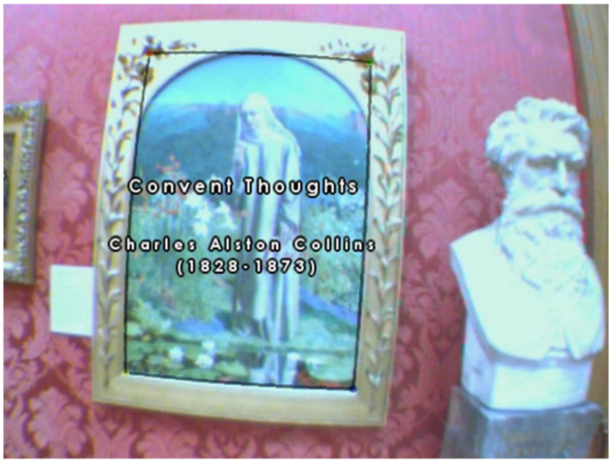

Fig. 5. "Convent Thoughts" [41] recognized and located, with AR overlay.

For the paintings not fully detected and localized in this experiment the principal modes of failure were that: (i) certain paintings were too small in the keyframes for SIFT to match; (ii) some had too few distinctive features to be recognized; and (iii) some were observed in too few keyframes to be localized.

5.2. Street scene

Fig. 6a shows example views of shop fronts from a street for which a database of some 11×10^3 keypoints was constructed. Fig. 6b shows a typical still view cut from the sequence captured by a user during a live run as he walked along the street wearing a shoulder-mounted camera, and Fig. 6c shows the 3D map and keyframe poses recovered by PTAMM. Only static features are captured: moving vehicles and pedestrians passing along the street are easily excluded by the tracking thread, as features on them do not obey the camera's motion model and thus are not matched from frame to frame. Fig. 6d shows a number of shop fronts detected and localized within the 3D point cloud.[1]

The numbers of detected and then localized keypoints on each object are given Table 2. They appear perilously small — if not for recognition, then for localization. However, recall that the camera poses are already known and that localization is by triangulation of strongly distinctive features.

While structure recovery using PTAMM outdoors proves quite routine and robust, using monoSLAM on larger scale outdoor scenes such as street scenes was rarely successful, as reported in [22]. For

[1] The location can be viewed using Google's Street View by searching on *10 Little Clarendon Street, Oxford, UK.*

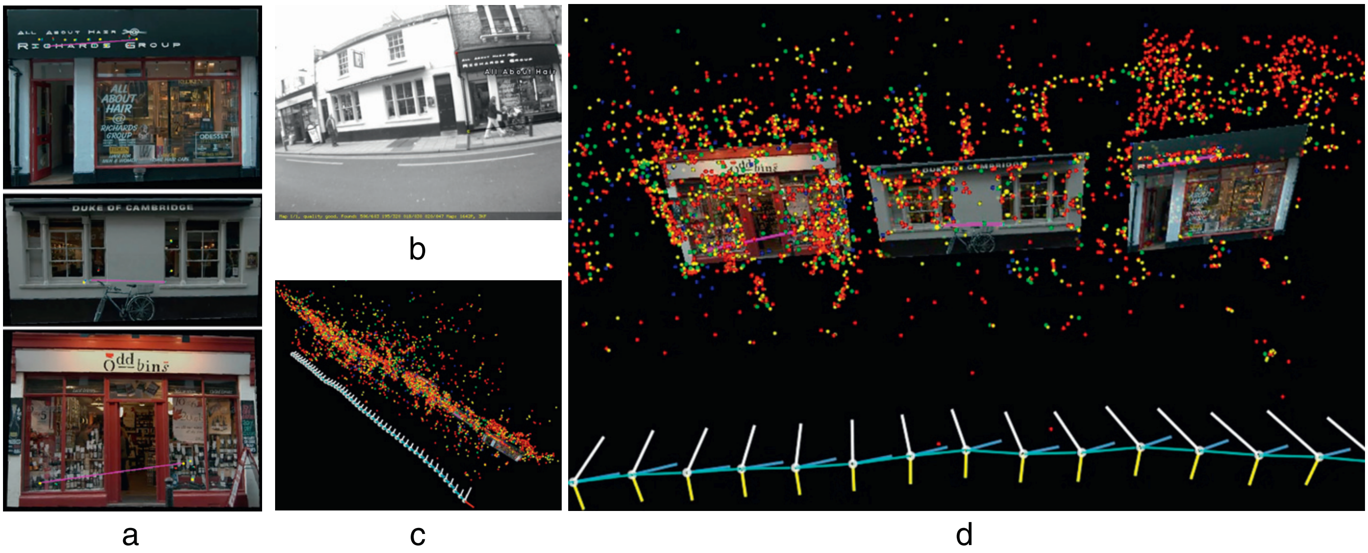

Fig. 6. (a) The database entries for *All about Hair*, the *Duke of Cambridge* and *Oddbins*. (b) A still from the live street sequence. (c) The map of the entire street. (d) The map and three recognized and located premises seen in the view in (b). (See note at end for location of video material.)

comparison with the point clouds in Fig. 6c, d, on similar sequences monoSLAM recovered the sparse map of Fig. 7a, and sometimes failed prematurely as in Fig. 7b.

5.3. Oscilloscope tutorial

This final application involves the use of AR to guide a novice through the use of a piece of equipment — here an oscilloscope. The experiment mirrors that in [22] which used monoSLAM.

A fronto-parallel image of the oscilloscope was captured, and locations of interest on it (the locations *A* in Eq. (3)) were identified and associated with AR-markup that provides a particular instruction to the user. When the user explores the environment, the oscilloscope is automatically detected, and they are prompted to begin the tutorial. Not all the markup is shown at once: rather, as each task is completed the user presses a button on the computer to advance to the next item of instruction.

Fig. 8 shows a few frames from the sequence, with AR labels directing the user and a circle placed over the button or dial of interest. In this example, the tutorial guides the user through powering up the oscilloscope, setting the dials to the correct positions, and connecting a probe to the correct socket.

For this object we find detection and recognition performs satisfactorily to some ±20° from frontal. Unlike the method presented in [22], this merely restricts the region in which *improvements* to localization can occur. Once located, as Fig. 8c shows, the addition of markup survives to far more oblique angles. Indeed, if the oscilloscope goes out of view, the system is unaffected, because the location of the oscilloscope is known in the world coordinate frame.

An informal comparison with the method based on mono-SLAM in [22] suggests that, while both methods can be made to perform robustly with practice, the present method allows the camera to be moved freely as the *user* demands, whereas the earlier method required the camera to be moved conservatively as monoSLAM demands.

6. Concluding remarks

This paper combines video-rate camera tracking and keyframe-based reconstruction of the 3D scene and camera poses from FAST features, with object recognition and localization by matching and triangulating SIFT features between keyframes. The object detection process runs in parallel with, but largely independently of, the 3D mapping and camera tracking processes. A method of keyframe selection is presented which prioritizes object recovery in the area of

Table 2
Numbers of keyframes and keypoints involved in the street scene.

Premises in street	Keyframes found in	Database keypoints	
		Found	Localized
All about Hair	4	26	17
Duke of Cambridge PH	2	12	6
Oddbins	3	21	9

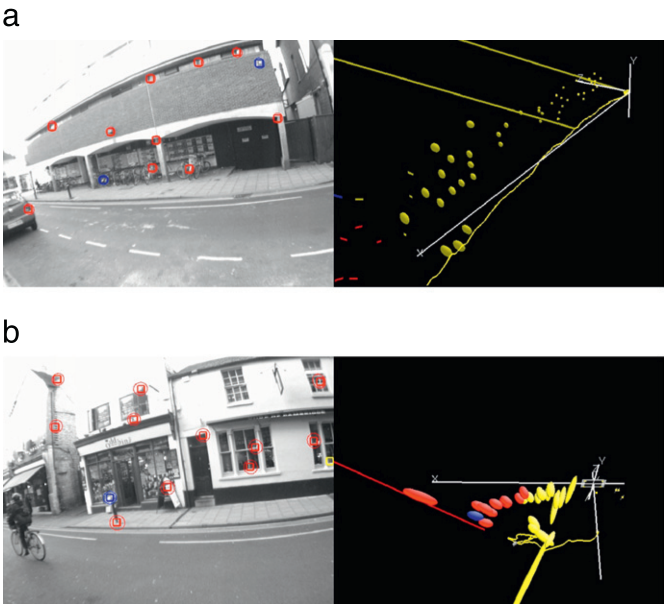

Fig. 7. MonoSLAM maps at the same street location. The map recovered in (a) is complete but very much sparser than that of Fig. 6c, and that in (b) is a premature failure to be compared with Fig. 6d. (The red positions and covariances denote map points which are predicted visible and are matched to observations, blue denotes predicted visible but not matched, and yellow are not predicted visible.)

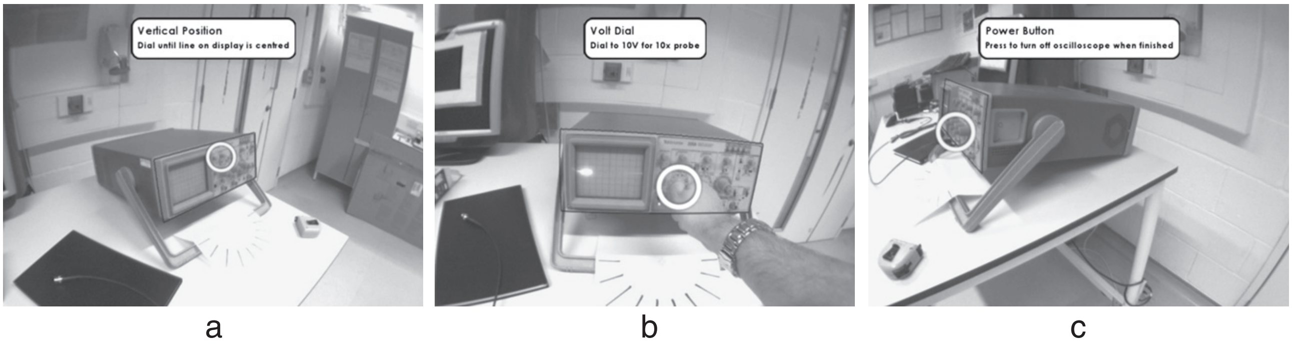
Vertical Position
Dial until line on display is centred
Volt Dial
Dial to 10V for 10x probe
Power Button
Press to turn off oscilloscope when finished
a
b
c

Fig. 8. AR elements are placed with respect to locations within the object's extent. As the user advances through the tutorial different overlays are displayed. (See note at end for location of video material.)

the scene being looked at currently but which will automatically search past areas for known objects. Objects become additions to the underlying map rather than being inextricably embedded in it. The method has been explored using laboratory scenes, and tested in more realistic applications — one a guide to an art gallery, the next a similar application recognizing commercial premises in the street, and the other providing a tutorial on use of equipment.

An advantage of the near independence of the two methods is that no coupled failure modes have been introduced. Furthermore, the 'natural' processing rate of the object recognition and localization thread becomes the more relaxed rate of keyframe acquisition rather than frame capture, and the proposed keyframe selection method allows for variations in this rate. Nonetheless, the computation time of SIFT remains incommensurate with the other tasks, a bottleneck which is further constricted by SIFT's competing with the map maker and tracker threads on a dual core machine. Relief would come of course from switching to a machine that has additional processing cores, or to one which supports a programmable graphics processor implementation of SIFT (*e.g.* [42]). However, both of these hardware features are at present scarce in portable laptops. An alternative would be to use a cheaper feature descriptor such as the SIFT-fern hybrid of Wagner et al. [43].

Whatever the efficiency of feature computation, scaling to larger databases will require adoption of hierarchical methods: the need for such organization based on contextual or other priors is well-rehearsed (*e.g.* [44]). However, we re-iterate that as object recognition, localization, and subsequent annotation are all independent of map building, they do not impact the scaling of the underlying map. As mentioned earlier, PTAMM maps are limited in size by how quickly the user wishes to explore and thence add new keyframes which trigger readjustment. Larger maps could be built offline for later use circumscribing the real time constraints.

We have noted the improvements that the current method affords over that in [22] in (i) the freedom of camera movement and (ii) the size of region explorable. We also highlight that objects are reconstructed by triangulation at the scale of the map in the object's locale. No conflict is introduced between object and map scales of the sort discussed in [22], and the overall quality of map recovery is greater.

These advantages are underpinned by the more fundamental benefit of keyframe-based SLAM over EKF-SLAM recently observed by Strasdat et al. [45]. They simulated a camera moving identically through environments with different numbers of landmarks and taking different numbers of views, and determined the reductions in Shannon entropy in the final camera state over that obtained for a basic scene using just two views and twelve landmarks. They found that increasing the density of landmarks always increases the reduction (*i.e.*, always increases information) though with diminishing returns, whereas increasing the density of views has a marginal effect.

Video material

Supplementary video material may be found at www.robots.ox.ac.uk/ActiveVision/Publications/castle_murray_ivc2011.

Acknowledgments

This work was supported by the Platform Grant Virtual Oxford from UK Engineering and Physical Science Research Council (Grant EP/D037077). The authors thank the Director of the Ashmolean Museum of Art and Archaeology for permission to experiment within its galleries.

References

[1] C. Harris, C. Stennett, RAPiD — a video rate object tracker, Proc 1st British Machine Vision Conference, 1990, pp. 73–78.
[2] D. Lowe, Fitting parameterized three-dimensional models to images, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (5) (1991) 441–450.
[3] D. Gennery, Visual tracking of known three-dimensional objects, International Journal of Computer Vision 7 (3) (1992) 243–270.
[4] T. Drummond, R. Cipolla, Real-time visual tracking of complex structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 932–946.
[5] A.W. Fitzgibbon, A. Zisserman, Automatic camera recovery for closed or open image sequences, Proc 5th European Conf on Computer Vision, Vol. 1, 1998, pp. 311–326.
[6] D. Nistér, Automatic dense reconstruction from uncalibrated video sequences, Ph. D. thesis, Royal Institute of Technology KTH, Stockholm, Sweden (March 2001).
[7] M. Pollefeys, L.V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, Visual modeling with a hand-held camera, International Journal of Computer Vision 59 (3) (2004) 207–232.
[8] W. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment — a modern synthesis, in: B. Triggs, A. Zisserman, R. Szeliski (Eds.), Vision Algorithms: Theory and Practice, Vol. 1883 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 298–372.
[9] A.J. Davison, Real-time simultaneous localisation and mapping with a single camera, Proc 9th IEEE Int Conf on Computer Vision, II, 2003, pp. 1403–1410.
[10] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6) (2007) 1052–1067.
[11] D. Nistér, O. Naroditsky, J. Bergen, Visual odometry for ground vehicle applications, Journal of Field Robotics 23 (1) (2006) 3–20.
[12] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Real time localisation and 3d reconstruction, Proc 24th IEEE Conf on Computer Vision and Pattern Recognition, I, 2006, pp. 363–370.
[13] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[14] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, Proc 21st IEEE Conf on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. 264–271.
[15] A. Opelt, A. Pinz, A. Zisserman, Learning an alphabet of shape and appearance for multi-class object detection, International Journal of Computer Vision 80 (1) (2008) 16–44.
[16] G. Klein, D.W. Murray, Parallel tracking and mapping for small AR workspaces, Proc 6th IEEE/ACM Int Symp on Mixed and Augmented Reality, 2007.
[17] C. McGlone, E. Mikhail, J. Bethel, Manual of Photogrammetry, 5th Edition, American Society of Photogrammetry and Remote Sensing, Bethesda, MD, 2004.

[18] R.O. Castle, D.W. Murray, Object recognition and localization while tracking and mapping, Proc 8th IEEE/ACM Int Symp on Mixed and Augmented Reality, 2009, pp. 179–180.

[19] A.J. Davison, W.W. Mayol, D.W. Murray, Real-time localisation and mapping with wearable active vision, Proc 2nd IEEE/ACM Int Symp on Mixed and Augmented Reality, 2003, pp. 18–27.

[20] W.W. Mayol, A.J. Davison, B.J. Tordoff, D.W. Murray, Applying active vision and SLAM to wearables, in: P. Dario, R. Chatila (Eds.), International Symposium on Robotics Research, Siena, Italy, October 19–21, 2003, Vol. 15, Springer, 2003, pp. 325–334.

[21] I. Gordon, D.G. Lowe, What and where: 3D object recognition with accurate pose, in: J. Ponce, M. Hebert, C. Schmid, A. Zisserman (Eds.), Toward Category-Level Object Recognition, Springer-Verlag, 2006, pp. 67–82.

[22] R.O. Castle, G. Klein, D.W. Murray, Combining monoSLAM with object recognition for scene augmentation using a wearable camera, Image and Vision Computing 28 (12) (2010) 1548–1556.

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

[24] E. Eade, T. Drummond, Unified loop closing and recovery for real time monocular SLAM, Proc 18th British Machine Vision Conference, 2008.

[25] S.A. Holmes, G. Sibley, G. Klein, D.W. Murray, Using a relative representation in parallel tracking and mapping, Proc 2009 IEEE Int Conf on Robotics and Automation, 2009.

[26] C. Mei, G. Sibley, M. Cummins, P. Newman, I. Reid, RSLAM: a system for large-scale mapping in constant time using stereo, International Journal of Computer Vision Online first. doi:10.1007/s11263-010-0361-7.

[27] R.O. Castle, G. Klein, D.W. Murray, Video-rate localization in multiple maps for wearable augmented reality, Proc 12th IEEE Int Symp on Wearable Computers, 2008.

[28] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, Proc 9th European Conf on Computer Vision, 2006.

[29] E. Rosten, R. Porter, T. Drummond, Faster and better: a machine learning approach to corner detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 105–119.

[30] J.E. Dennis, Nonlinear Least-Squares, Academic Press, London, 1977, pp. 269–312.

[31] Z.Y. Zhang, Parameter estimation techniques — a tutorial with application to conic fitting, Image and Vision Computing 15 (1) (1997) 59–76.

[32] S.A. Holmes, Challenges in real-time slam: curbing complexity, cultivating consistency, D.Phil. thesis, Department of Engineering Science, University of Oxford (2010).

[33] D.W. Marquardt, An algorithm for the least-squares estimation of non-linear parameters, Journal of the Society Industrial Application Math 11 (2) (1963) 431–441.

[34] J.W. Tukey, Exploratory Data Analysis, Addison-Wesley, Reading MA, 1977.

[35] D. Nistér, An efficient solution to the five-point relative pose problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6) (2004) 756–777.

[36] J.S. Beis, D.G. Lowe, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, Proc 1997 IEEE Computer Society Conf on Computer Vision and Pattern Recognition, 1997, pp. 1000–1006.

[37] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, 2004.

[38] B.K.P. Horn, H.M. Hilden, S. Negahdaripour, Closed form solution of absolute orientation using orthonormal matrices, Journal of the Optical Society of America. A 5 (7) (1988) 1127–1135.

[39] O. Faugeras, M. Hébert, A 3d recognition and positioning algorithm using geometric matching between primitive surfaces, Proc. 8th. Int. Joint Conf. on Artificial Intelligence, IJCAI-83, 1983, pp. 996–1002.

[40] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communication ACM 24 (6) (1981) 381–395.

[41] C.A. Collins, Convent Thoughts, Western Art Collection, Ashmolean Museum, Oxford (Painted 1850–51).

[42] S.N. Sinha, J. Frahm, M. Pollefeys, Y. Genc, Feature tracking and matching in video using programmable graphics hardware, Machine Vision and Applications 22 (1) (2011) 207–217, doi:10.1007/s00138-007-0105-z.

[43] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg, Pose tracking from natural features on mobile phones, Proc 7th IEEE/ACM Int Symp on Mixed and Augmented Reality, 2008.

[44] M.J. Choi, J.J. Lim, A. Torralba, A.S. Willsky, Exploiting hierarchical context on a large database of object categories, Proc 28th IEEE Conf on Computer Vision and Pattern Recognition, 2010.

[45] H. Strasdat, J.M.M. Montiel, A.J. Davison, Real-time monocular SLAM: why filter? Proc 2010 IEEE Int Conf on Robotics and Automation, 2010.