2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras

R O Castle, D J Gawley, G Klein, and D W Murray

Abstract— **This paper presents a system which combines single-camera SLAM (Simultaneous Localization and Mapping) with established methods for feature recognition. Besides using standard salient image features to build an on-line map of the camera's environment, this system is capable of identifying and localizing known planar objects in the scene, and incorporating their geometry into the world map. Continued measurement of these mapped objects improves both the accuracy of estimated maps and the robustness of the tracking system. In the context of hand-held or wearable vision, the system's ability to enhance generated maps with known objects increases the map's value to human operators, and also enables meaningful automatic annotation of the user's surroundings. The presented solution lies between the high order enriching of maps such as scene classification, and the efforts to introduce higher geometric primitives such as lines into probabilistic maps.**

I. INTRODUCTION

Much of the groundwork in wearable vision has focused on where and how to place cameras on the wearer's body, and how to supply graphically augmented video to the wearer and a remote assistant if present. Cameras have been attached to the torso to recover ambient information [1]; to the head [2], [3] and wrists [4] to provide more specialized task oriented views; and have been mounted onto inertially and visually controlled platforms to afford some degree of independence from the wearer's motion [5], [6]. Providing the wearer and remote assistant with enhanced views is a valuable application, as indeed is the use of wearable cameras for visual memory augmentation [7], [8]. However, visual sensing can and should operate at a more profound level, providing the wearer with autonomous advice as to what is where, where to go next, and so on. There are many shared problems here with robot navigation; but there are also sharp contrasts: principally, these are that a camera's human "carrier" is highly intelligent, but geometrically sloppy, and not amenable to precise or timely control.

Despite these contrasts, the two basic preconditions for autonomous wearable vision are similar to those in robot navigation [9]. The first is that the camera must be able to establish its location in an initially unknown environment, and later be able to re-address areas of the scene. In the wearable domain, some work has used special fiducial markers [10] and pre-mapped targets [11]. A more general approach was taken by Mayol *et al.* [9], [12], who adapted Davison's monocular method of simultaneous localization and mapping (SLAM) [13], and demonstrated that their active wearable was able to fixate upon successive locations or landmarks while continuing to accumulate scene structure and motion.

The authors are with the Active Vision Laboratory, Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK. www.robots.ox.ac.uk/ActiveVision [bob,djg,gk,dwm]@robots.ox.ac.uk

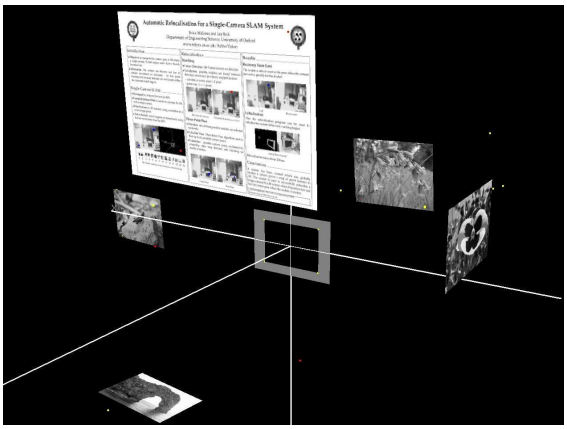


Fig. 1. A 3D view of the SLAM map with the identified objects.

The second requirement, and the focus of this paper, is that at least some of those landmarks be tagged with higher-level meaning, permitting a dialogue between the vision system and its wearer. The method introduced here, of which exemplar results are shown in Fig. 1, builds on the single camera SLAM approach by inserting into the map features from objects of known size which have been *recognized* in the scene. Here the objects are planar — a book cover, a picture, or similar — but this is not a fundamental restriction. We suggest how objects can be incorporated into the state with minimal disruption, and used to advantage not only in visualisation, but also in improving the accuracy and overall scale of the map.

The problem being addressed is of wider relavance. In general, when SLAM is used either with traditional range sensors or with cameras, the maps themselves are represented as a sparse set of feature point locations. These maps are sufficient for a robot to perform tasks such as (semi-) autonomous navigation, or to infer some metric information about its surrounds. However, for the method to progress into large scale field environments and become more ubiquitously applicable, a richer map is required: one that incorporates more than just point features, and can more easily be interpreted by a human observer.

There are several current active research directions which aim to augment or interpret robot maps in some way. One such approach uses methods from machine learning and attempts to divide a map into a small number of given classes such as office, corridor, or doorway (e.g. [14]). More ambitious work attempts to determine these labels automati-

cally. For example, in [15] recorded images are analysed and categorized into distinctive categories, which are then used to characterise areas of the map. At a slightly lower level is geometric map augmentation: one recent approach used geometric primitive fitting combined with model selection to produce a reduced representation of a map [16].

However, these approaches perform post-processing, running apart from the SLAM itself. In this work, by using the same point representation for both localization and recognition, we have a method where both can progress in tandem.

Section II describes the method of object identification using Lowe's Scale Invariant Feature Transform (SIFT), and the nature of the objects used in this work. Section III explains the method of object location using single view geometry and Section IV briefly reviews monocular SLAM before explaining how the additional 3D object locations are inserted into the SLAM map. An initial experimental evaluation is described in Section V and the paper closes with some directions for future work.

II. Detection and Identification from Point Features

Our first aims are (i) to detect and identify known planar objects in the scene and (ii) to determine their location in the world frame from just a single image. The location will serve as a measurement for the SLAM process.

To unify recognition, localization and SLAM, we adopt a point-based representation throughout, and use features which exhibit scale and rotation invariance, allowing an object to be detected and re-detected over wide fields of view. Mikolajczyk and Schmid's evaluation of descriptors [17] identified Lowe's SIFT descriptor [18] as being the most resistant to common image deformations. The reader familiar with SIFT might skip to subsection II-B below.

A. Detection using SIFT

Object detection is a four stage process. In the first step a scale space pyramid is created by successive convolutions with a Gaussian. A quasi-Laplacian operator is applied at each scale by computing difference of Gaussians (DoG) in adjacent images in the scale pyramid. The second stage computes extrema in the output by searching the octaves of the DoG pyramid, with each pixel compared against its spatial neighbours at the same scale and its scale neighbours above and below. If the current pixel is the largest or the smallest it is recorded as an extremum. Location to subpixel accuracy follows by fitting a 3D quadratic function around the extrema. The extrema are filtered for contrast and edge response, the latter using the ratio of principal curvatures.

The third stage creates the descriptors. Each keypoint is assigned an orientation by forming a histogram of the orientations of the pixels at the extremum's scale around the extremum, and selecting the dominant orientation. The descriptor is formed from the orientations and magnitudes of the pixels around the keypoint by sampling over a 16×16 array aligned to the keypoint's orientation. This array is broken down into 4×4 subregions, and within each subregion

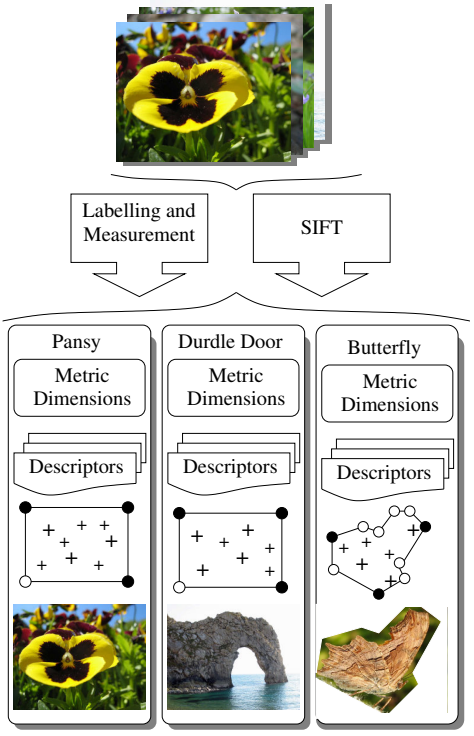


Fig. 2. The object database contains planar objects (here pictures), the list of SIFT descriptors and their locations $\boldsymbol{X}_o$, (crosses), and the locations of boundary points, three of which are marked for use in the SLAM map.

the orientations are used to fill an orientation histogram of eight bins weighted by the magnitudes of the pixels and a Gaussian to give pixels nearer the extrema more weight than those further away. The combination of these orientation histograms from these subregions gives the 128-long descriptor. The descriptor is then normalised to provide invariance to illumination change.

The final stage of SIFT is to match the keypoints. Lowe [18] uses both the first and second nearest neighbours to identify matches. Each keypoint in the database image is compared to each keypoint in the scene image by calculating the Euclidean distance from one to the other. A record is kept of which two scene keypoints are the nearest to the database image keypoint. Their distance ratio is calculated, and if less than a threshold the nearest neighbour is considered a match to the database keypoint. A correlation of keypoints on the database image to the keypoints on the scene image is now known and any unmatched keypoints are discarded.

B. The object database

In our work we build a database of planar objects. To construct an entry, an image of the object is captured and, after correcting for radial distortion, SIFT descriptors are computed and stored along with their image positions $\boldsymbol{x}_o$. The image need not be fronto-parallel, and so it is also necessary to compute and store the homography $\mathtt{H}_o$ between the scene and image by choosing $n \geq 4$ image points whose corresponding scene points can be located in a Euclidean coordinate frame. In 2D homogeneous coordinates, points in

the object plane are $\boldsymbol{X}_o = [X_o, Y_o, 1]^\top$, and up to scale

$$\boldsymbol{X}_o = \mathtt{H}_o^{-1}\boldsymbol{x}_o \ . \tag{1}$$

As illustrated in Fig. 2, the database entry contains the list of SIFT descriptors and their scene locations $\boldsymbol{X}_o$. In addition we store the locations of boundary points to define the object extent, and, as explained later, flag three of them for use in the SLAM map. The image of the object is rectified by the homography so that it appears as a fronto-parallel view.

C. Object detection

SIFT is run at regular intervals on the current video frame after removing radial distortion, and the detected features are matched to the stored keypoints of the known objects. As noted earlier Lowe's second nearest neighbour method is used to generate a set of candidate matching descriptors [18]. If the number of matched points from any given object's database entry to the current image is greater than a threshold (in our case eight) we regard that object as a candidate. Because of repeated structure or other scene confusion, some of the features may be incorrectly matched. However, as the database objects are known to be planar, the database scene points $\boldsymbol{X}_o$ and currently observed image points $\boldsymbol{x}_i$ are related by a plane-to-plane homography

$$\boldsymbol{x}_i = \mathtt{H}_i\boldsymbol{X}_o \ . \tag{2}$$

RANSAC is used to estimate the homography $\mathtt{H}_i$ and, if a sufficiently large consensus set is found, we infer that the database object is visible in the current frame.

III. Single view localization

Having determined an object is visible we recover its location by decomposing the homography between scene and current image.

In the Euclidean object-centred coordinate frame, the object lies in the plane $Z_o = 0$, and 3D homogeneous points on the object are $\boldsymbol{X}_o^{(4\times1)} = [X_o, Y_o, 0, 1]^\top$. In any view i, the projection can therefore be written in terms of extrinsic and intrinsic parameters as $\boldsymbol{x}_i = \mathtt{K}_i[\mathtt{R}_i|\boldsymbol{t}_i]\boldsymbol{X}_o^{(4\times1)}$. Hence

$$\boldsymbol{x}_i = \mathtt{K}_i\mathtt{A}_i\boldsymbol{X}_o \ , \tag{3}$$

where $\mathtt{A}_i = [\boldsymbol{r}_{i1}\ \boldsymbol{r}_{i2}\ \boldsymbol{t}_i]$ contains the translation $\boldsymbol{t}_i$ and the first two columns of the rotation matrix $\mathtt{R}_i$, all modulo a scaling factor. Using the homography already computed as the output of RANSAC and assuming known camera calibration $\mathtt{K}_i$,

$$\begin{bmatrix}\boldsymbol{r}_{i1} & \boldsymbol{r}_{i2} & \boldsymbol{t}_i\end{bmatrix} = \mathtt{K}_i^{-1}\mathtt{H}_i \ , \tag{4}$$

again up to scale. Because the estimate $\mathtt{H}_i$ is noisy, there is no guarantee that $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ found as above will be orthogonal (which they are required to be as they are columns of a rotation matrix). We determine the closest rotation matrix, and hence the overall scale for the translation, using singular value decomposition.

The rotation matrix and translation vector calculated in this way specify the transformation of the camera from the frame of reference of an object's canonical database image. We can apply this transformation in reverse to place the object in the frame of reference of the camera; and then apply a further transformation determined by the camera's current position relative to the world coordinate frame defined by the SLAM map to determine the position of the object in world coordinates.

IV. Single camera SLAM and wearables

Single camera SLAM in which the camera is allowed to move generally in 3D is a challenge because neither single nor multiple views of a single point yield depth when the motion is unknown. Information comes from points collectively, but as processing has to be completed in a fixed time a limit must be imposed on the feature map size[1]. As explained below, we use the object's location to inject a set of 3D measurements into the map.

In the extended Kalman Filter formulation of Davison [13], the state $\boldsymbol{\chi}$ comprises two parts, $\boldsymbol{X}_i$, the fixed 3D locations of map features, and $\boldsymbol{c}_t = (\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\omega})$, the time-dependent camera position, orientation, translational velocity and angular velocity, all defined in a fixed world frame. The associated covariance $\mathtt{P}$ is fully populated:

$$\boldsymbol{\chi} = \begin{bmatrix}\boldsymbol{c}_t \\ \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_n\end{bmatrix} \quad \mathtt{P} = \begin{bmatrix}\mathtt{P}_{cc} & \mathtt{P}_{cX_1} & \cdots & \mathtt{P}_{cX_n} \\ \mathtt{P}_{X_1c} & \mathtt{P}_{X_1X_1} & \cdots & \mathtt{P}_{X_1X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{P}_{X_nc} & \mathtt{P}_{X_nX_1} & \cdots & \mathtt{P}_{X_nX_n}\end{bmatrix} . \tag{5}$$

The state prediction, measurement process, and update cycle are quite standard, and assume constant translational and angular velocities in the world frame. In our implementation the $\boldsymbol{X}_i$ will come from objects and from "standard" features.

A. The choice of "standard" features for SLAM

The need for robustness against viewpoint changes during SLAM is no less than that during recognition, and ideally the same feature detector would be used throughout. Se *et al.* [19] have implemented trinocular visual SLAM using SIFT, but the time required to locate and match even the few features needed in our work would be well in excess of the inter-frame time. While object detection does not *need* to occur every frame, in order to maintain correct matching the underlying localization process really must.

In Section VI we note that a number of detectors which are faster than SIFT might be considered for deployment throughout. However, for this implementation, "standard" features for (potential) insertion into the 3D map are detected with the Shi-Tomasi saliency operator [20], and features that are eventually inserted are stored with an 11×11 pixel appearance template. Given a predicted camera position, each feature $\boldsymbol{X}_i$ is projected into the new image along with its associated uncertainty region derived at the 3σ limit from the prior covariance projected into the image. Searches are made within the region for correspondence using normalised sum-of-squared difference correlation.

[1]For some applications of wearables, such a constraint is quite compatible with a restricted workspace around a user.

B. Adding and managing "standard" features

For standard features, i.e. those not derived from recognition, Davison's original method of feature initialization [13] has been used in experiments reported here. The more recent method [21] can initialize features over much wider depth ranges than the particle method, by using inverse depth in the state, effectively adopting a disparity-based representation.

The map management criteria aim to restrict the number of features observable in any one view to be compatible with both the desired localization accuracy and maintaining video frame-rate on portable CPUs. A feature is *predicted* to be observable from some particular viewpoint if its projection into a model of the camera lies well within the image, and if both angular and range differences between the viewpoint and the feature's initial viewpoint are not so great as to make the image template valueless. New features are added if the number of those actually observable falls to five or less, and a feature is deleted if its long-term ratio of actual observability after search to predicted observability falls below 1:2.

C. Adding recognized object locations to the SLAM map

We now turn to the key aspect of the paper: adding recognized objects to the SLAM map. A number of methods for achieving this can be envisaged. In this work we choose the straightforward but effective approach of using the recovered 3D position of the planar object to define 3D point measurements.

However, we do *not* use the feature positions themselves. Instead we use points on the object's boundary — and use the minimum of three to define the plane whilst avoiding overburdening the SLAM process. For example, for the rectangular pictures used in experiments, three of the four corners are used, as indicated in Fig. 2.

There are several benefits in this approach. First, no additional mechanism is required in the SLAM process. Provided reasonable values are supplied for the (typically much lower) 3D error in these points, constraints on the scene will propagate properly through the covariance matrix. Secondly, and perhaps most importantly, is that we do not rely on any particular SIFT features being re-measured over time. In effect, the underlying planar structure is allowing implicit measurement of the chosen three boundary points, much reducing the likelihood of mismatching which would otherwise damage the map. Thirdly, for graphical augmentation, the boundary points provide a convenient representation of the extent of the object.

V. Implementation and Experimental Evaluation

The detection, localization and SLAM methods have been implemented in C++ and run on a single 3.2 GHz Pentium 4 CPU. Including operating system overheads, monocular SLAM with around 20 point features takes approximately 10 ms on a 320×240 image, leaving some 20 ms per frame to pursue detection. SIFT detects around 300 keypoints and takes some 500 ms per frame to complete on average, and matching one object takes on average 70 ms. While SLAM runs at 30 Hz this unoptimized detection can run at around

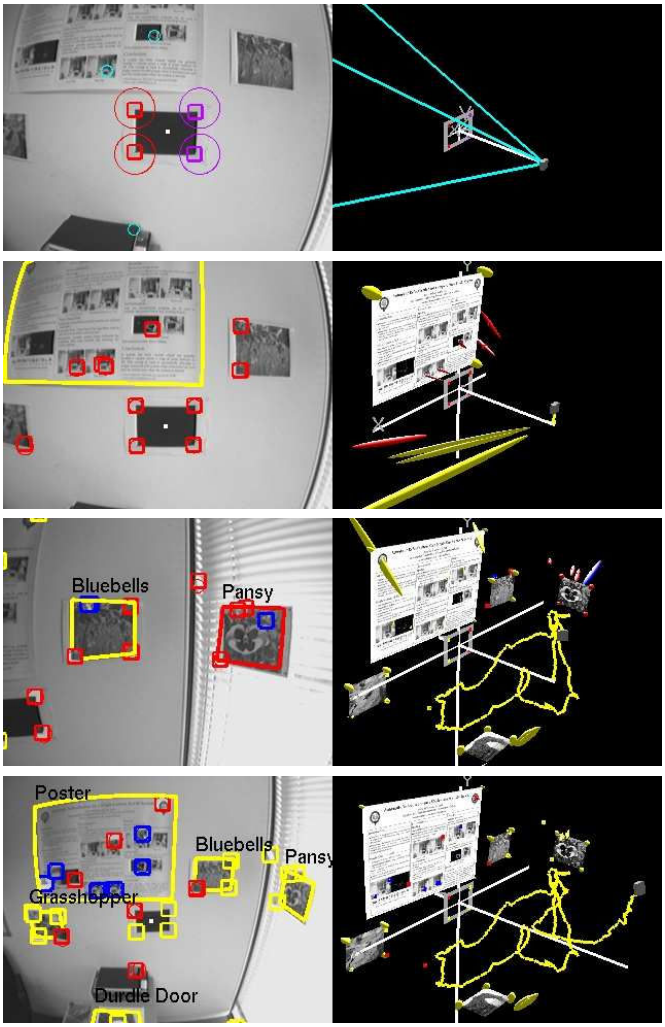


Fig. 3. From top to bottom (the camera view is shown on the left and the map on the right): The initial image with calibration plate visible in the map; The first object is detected; More objects are detected and added to the map; All objects have been detected and successfully localized.

1 Hz at best. These timings will of course vary with the number of objects in the database. With larger databases a faster searching method such as KD-trees would be required. To allow the storing of the results, the experiments reported here were run on a pre-recorded sequence, where the SIFT processing and matching was run only every 30th frame.

In the tests of the system reported here, a database of five planar objects was used. The database was created by running SIFT on each object image to generate the keypoints and measuring the metric sizes of the objects. The size of the images and the number of keypoints generated for each object in the database is given in Table I.

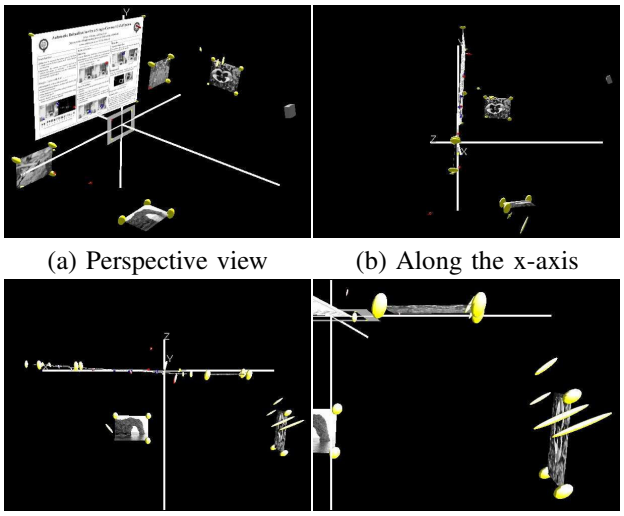
Fig. 3 shows the evolution of processing, from initial calibration of the SLAM system to a time when there are five recognized planar objects in the SLAM map. The 2D views show overlaid identities and extents of the objects, typical of that which would be useful to the user of a wearable or hand-held camera. The views on the right show the evolution of the 3D map with textures marking the recognized areas.

Fig. 4 shows various views around a particular 3D map

TABLE I

DATABASE OBJECTS, KEYPOINTS, AND THE SIZES

Object label	No. of keypoints	Image size	Metric Size (m)
Bluebells	761	320×256	0.248×0.198
Durdle Door	981	320×227	0.264×0.198
Grasshopper	469	320×256	0.248×0.198
Pansy	676	320×256	0.248×0.198
Poster	1240	320×240	0.841×0.594
Total	4127		



(a) Perspective view (b) Along the x-axis

(c) Along the y-axis (d) Two of the objects

Fig. 4. (a) View of the whole 3D map. (b,c) Individually recognized and located planar objects on the XY wall are recovered as coplanar to within map error, and (d) the right angles between two walls and the floor are also recovered faithfully. See Table II.

in which there are five picture objects, three of which (poster, bluebells, grasshopper) should be coplanar with the calibration plate (and hence in the XY-plane), and the other two (pansy, Durdle Door) are mutually orthogonal in the ZY and XZ planes respectively. It can be seen that all of the objects are in their respective planes to within experimental error. Table II shows the angles between the planes recovered from the SLAM map.

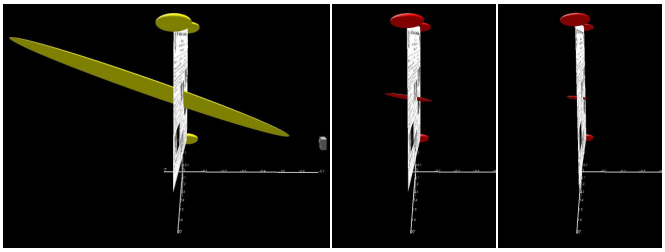
Object points localized via object detection and localization benefit map building because they can be inserted into the map with a higher level of accuracy using just a single measurement. This same level of accuracy is only achieved with "standard" interest point features when observed multiple times from different angles. To illustrate this assertion, the leftmost image in Fig. 5 shows an interest point feature which has just been initialised on the surface of an object which has also just been detected. The highly elongated ellipse clearly shows that it is not well localized in depth, compared with the object points' uncertainty ellipses. The middle and rightmost images show the scene 30 and 90 frames later. The detected object points have just been measured for a second and fourth time, respectively, whereas the interest point has been measured in every intermediate frame. Only by the 90th frame is the interest feature point as well localized in depth as the object points.

Over time, with enough observations, the difference in the uncertainty of interest points compared to those of objects is

TABLE II

ANGLES BETWEEN THE CALIBRATION PLATE AND THE OBJECTS

Object label	Actual angle	Measured angle	Error
Bluebells	0	5.9	± 10
Durdle Door	90	95.2	± 10
Grasshopper	0	5.0	± 5
Pansy	90	92.7	± 4
Poster	0	4.7	± 5



(a) Frame 0 (b) Frame 30 (c) Frame 90

Fig. 5. A "standard" interest feature point on the surface of an object. The 3σ ellipse lengths are 138cm and 12cm respectively when first initialized (a). By frame 90, (c), they are both as well localized in depth (9.2cm).

small. The advantage of using object measurements occurs when only a limited number of measurements are made or the view does not change enough for features to become well localized – a common occurance for wearable cameras.

The monocular SLAM system takes time to initialise new features. If during this time features that are being tracked fail to be measured, for example due to occulsion or moving out of shot, the system can loose track of its position. Being able to measure objects as well as interest points also makes the system more robust to such tracking failure, as demonstrated in the accompanying video.

Although the map looks well-formed in terms of coplanarity and angles, the error covariance for the 3D measurements is not fully characterized. It depends, of course, on the image covariance and its tortuous propagation through the estimation of homographies and their decomposition, and a subsequent Euclidean transformation. Here, as an expedient, we have assumed that the error covariance is diagonal and have examined the effect of inserting a single 3D feature on the performance of the EKF when already running "standard" monocular SLAM. Tuning the performance to the size of the covariance suggest that the lateral and depth errors are of order 10 mm and 20 mm respectively.

VI. DISCUSSION

This paper has presented the combination of methods of point-based recognition and localization with those of point-based monocular visual SLAM to identify and to recover the 3D geometry of objects, and then insert them as 3D measurements into the map for updating by an extended Kalman filter. The method has been demonstrated using planar objects. By fitting a higher geometrical entity to visual data, the measurements entered into the SLAM map are sparse, robust to partial occlusion, less likely to be incorrect through mismatching, and more accurate.

There are a number of ways the approach can be developed. On the more geometrical side, provided care is taken

when specifying allowed acceleration noise in the EKF, it is possible to initialize the monocular SLAM (particularly that using inverse depth [21]) without calibrating the scale, and then to allow identified objects to provide resolution of the depth/speed scaling ambiguity.

Using salient features for wide-view matching has been used in SLAM to assist in loop closure [22], [23]. In large scale problems, establishing a graph of identified objects with additional sparse geometrical information about them (for example, a single point location and surface normal for a planar object) would allow a more rapid search for potential locations for detailed matching. Much the same approach can be taken after mid-loop tracking failure.

Another area for development is the addition of objects to the database during motion, determining their size from what is known about the world already. Here a more unified and faster approach to feature detection would be useful. The very fast detector based on randomised trees [24] is able to detect features at video rates in 640×480 images, and in our experience is able to handle large out-of-plane rotation. Its scalability to multiple objects is however unexplored. Another very recent detector of promise is the "speeded up robust feature" (SURF) detector [25]. However, when the database is of a realistic size, it seems unlikely that any object identifier will be able to run sufficiently quickly, particularly if it wastes time re-identifying areas of the image already examined. This can be offset by searching for further objects intelligently by only searching unchecked areas. Re-detecting an object is also quicker since only the area of the image where it is expected needs checking.

The focus of the work has been on the geometrical benefits of recognition, and little has been said of availability of meaningful labels. In [9] map features were hand-labelled to allow a remote operator to command an active wearable to fixate on successive objects while continuing to map. By adding auditory feedback to the present system we intend to explore the control of the wearer of an active camera, of the sort alluded to earlier in the paper.

VII. Acknowledgements

This work was supported by UK Engineering and Physical Science Research Council (grants GR/S97774 and EP/D037077). The authors are grateful to David Lowe for the SIFT source code, and for the insightful conversations with members of the Active Vision Lab.

References

[1] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, Phoenix AZ, 1999*, volume VI, pages 3037–3040, 1999.

[2] H. Aoki, B. Schiele, and A. Pentland. Realtime personal positioning system for a wearable computers. In *Proc 3rd IEEE Int Symp on Wearable Computing, San Francisco CA, Oct 18-19, 1999*, pages 37–43, 1999.

[3] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[4] A. Vardy, J. Robinson, and L. Cheng. The WristCam as input device. In *Proc 3rd IEEE Int Symp on Wearable Computing, San Francisco CA, Oct 18-19, 1999*, pages 199–202, 1999.

[5] W. W. Mayol, B. J. Tordoff, and D. W. Murray. Designing a miniature wearable visual robot. In *Proc Int Conf on Robotics and Automation, Washington DC, May 11-15, 2002*, pages 3725–3730, 2002.

[6] T. Kurata, N. Sakata, M. Kourogi, H. Kuzuoku, and M. Billinghurst. Remote collaboration using a shoulder-worn active camera/laser. In *Proc 8th IEEE Int Symp on Wearable Computing, Arlington VA, Oct 31 - Nov 3, 2004*, pages 62–69, 2004.

[7] J. Farringdon and Y. Oni. Visual uugmented memory. In *Proc 4th IEEE Int Symp on Wearable Computing, Atlanta GA, Oct 16-17, 2000*, pages 167–168, 2000.

[8] T. Kawamura, Y. Kono, and M. Kidode. Nice2CU: managing a person's augmented memory. In *Proc 7th IEEE Int Symp on Wearable Computing, White Plains NY, Oct 21-23, 2003*, 2003.

[9] W. W. Mayol, A. J. Davison, B. J. Tordoff, and D. W. Murray. Applying active vision and slam to wearables. In P. Dario and R. Chatila, editors, *Robotics Research: The Eleventh International Symposium, Siena Italy, October 19-21, 2003 (Springer Tracts in Advanced Robotics*, volume 15, pages 325–334. Springer, 2005.

[10] E. Foxlin. Generalized architecture for simultaneous localization, auto-calibration and map-building. In *Proc IEEE/RSJ Conf on Intelligent Robots and Systems, Lausanne, Switzerland, October 2-4, 2002*, pages 527–533, 2002.

[11] M. Kourogi, T. Kurata, and K. Sakaue. A panorama-based method of personal positioning and orientation and its real-time applications for wearable computers. In *Proc 5th IEEE Int Symp on Wearable Computing, Oct 2001*, pages 107–114, 2001.

[12] A. J. Davison, W. W. Mayol, and D. W. Murray. Real-time localisation and mapping with wearable active vision. In *Proc IEEE Int Symp on Mixed and Augmented Reality, Tokyo, Oct 2003*, 2003.

[13] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc 9th Int Conf on Computer Vision, Nice France, Oct 13-16, 2003*, 2003.

[14] A. Rottmann, O. Martínez-Mozos, C. Stachniss, and W. Burgard. Semantic place classification of indoor environments with mobile robots using boosting. In *AAAI2005*, pages 1306–1311, 2005.

[15] H. I. Posner, D.Schröter, and P. M. Newman. Using scene similarity for place labeling. In *Proc 10th Int Symp on Experimental Robotics, Rio de Janeiro, July 6-10*, 2006.

[16] S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery Montemerlo, C. Deepayan, and W. Burgard. A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 20(3):433–442, 2003.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

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

[19] S. Se, D.G. Lowe, and J.J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–760, 2002.

[20] J. Shi and C. Tomasi. Good features to track. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, Seattle WA, June 21-23, 1994*, pages 593–600, 1994.

[21] J. M. M. Montiel, J. Civera, and A. J. Davison. Unified inverse depth parametrization for monocular SLAM. In *Proc Conf on Robotics: Science and Systems, Philadelphia PA, Aug 16-19*, 2006.

[22] P. M. Newman and K. Ho. SLAM-loop closing with visually salient features. In *Proc Int Conf on Robotics and Automation, Barcelona, Apr 18-22, 2005*, pages 644–651, 2005.

[23] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkmann. A framework for vision based bearing only SLAM. In *Proc Int Conf on Robotics and Automation, Barcelona, Apr 18-22, 2005*, 2005.

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

[25] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc 9th European Conf on Computer Vision, Graz, May 7-13*, pages 404–417, 2006.