Video-rate recognition and localization for wearable cameras

R O Castle, D J Gawley, G Klein, and D W Murray
Department of Engineering Science, University of Oxford, UK
[bob,djg,gk,dwm]@robots.ox.ac.uk

Abstract

Using simultaneous localization and mapping to determine the 3D surroundings and pose of a wearable or hand-held camera provides the geometrical foundation for several capabilities of value to an autonomous wearable vision system. The one explored here is the ability to incorporate recognized objects into the map of the surroundings and refer to them. Established methods for feature cluster recognition are used to identify and localize known planar objects, and their geometry is incorporated into the map of the surrounds using a minimalist representation. Continued measurement of these mapped objects improves both the accuracy of estimated maps and the robustness of the tracking system. In the context of wearable (or hand-held) 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.

1 Introduction

Three principal threads run through research into wearable computing. The first is the creation of strata of portable and genuinely wearable hardware, and the second is the development of unobtrusive and socially acceptable sensors and interfaces to gather data and feed information back to the user. These two alone allow a degree of environmental and self monitoring by the user, or monitoring of the user by a remote operator. The last thread involves the exploration of perceptual modalities which can assess the user's environment, the user's relationship to it and activities within it, and thence augment the user's capabilities by offering contextually pertinent advice.

Work in the first thread has been greatly aided by the inexorable increase in the integration of electronic components. Wearability, however, requires account to be taken of human factors which must still be determined empirically over several cycles of design, build, and test. The hardware series from Smailagic, Sieworek and coworkers at CMU (e.g. [24]) and more recently by Tröster and colleagues at ETH (e.g. [1]) are ones where the design methodology is particularly clear. Within the second thread, sensors can be grouped into those sensing the wearer or sensing the surroundings. A raft of physiological signals has been used to determine muscle, brain and heart activity, skin conductance, respiration, blood pressure, and body temperature; and accelerometers and motion-sensitive textiles have measured user activity [17] [22] [20] [31]. Outward looking environmental sensors include those for ambient quantities like noise, or temperature (e.g. [5]); those giving just the user's position (e.g. [2]); and those giving a more fine-grained understanding of the surroundings (e.g. [27] [11] [19] [30]). Of this last group, it is visual sensing

BMVC 2007 doi:10.5244/C.21.112

that provides the strongest first-person perspective on the surroundings. The breadth of information available from imagery (or potentially so) makes it the "must-have" sensor for the third research thread.

Key to providing a wearable camera system with a greater degree of autonomy are the abilities both to locate the camera in the environment and to determine what is where around it [18]. In [18] it was demonstrated that once a partial 3D map of the camera's environment is established, locations can be selected for the camera to fixate upon, counteracting movements of the wearer, while continuing both to accumulate scene structure and to determine the camera pose. Whilst that process demonstrated the ability of a wearable system to direct attention independently of the wearer's movements, the 3D structure had no intrinsic significance to the wearable system. The fixated 3D points *may* have been related to known objects, but the semantic link between point and object had to be established by the wearer.

In [4] this limitation was removed by using learned appearance models to recognize objects in the scene. As well as permitting graphical augmentation of the recovered map, it was found that incorporating the recognized objects' geometry into the map improved the robustness and accuracy of localization. A minimal representation of the objects was used, which had the benefit of causing minimal disruption in the underlying localization mechanism, meaning they could run independently.

In this paper we advance the method in [4] by proposing a novel implementation with a solution to providing synchronization between the localization process, which runs regularly at video rate, and the recognition process which takes both a considerably longer and variable time. We adapt the method of delayed decision making of Leonard and Rikoski [12], a method developed to enable the initialization of features using data from multiple steps. We demonstrate the method working on a localized desk top environment, showing that a spatially-aware dialogue can be established between the wearable system and its wearer.

The following two sections briefly review the methods of establishing the camera position and map using monocular SLAM, and object identification using SIFT features. Section 4 describes the new method of organizing the combination of localization and recognition, and Section 5 gives an experimental evaluation. The paper closes with remarks on current work to combine fixation with recognition.

2 MonoSLAM

In contrast to batch methods of structure from motion recovery, simultaneous localization and mapping [26] [14] [28] places emphasis on continual recovery of the state of the camera and structure, and on maintaining information on the correlation between state members — not only to allow re-matching after neglect, but also to allow uncertainty to be reduced throughout the camera and map state vector when loop closing occurs.

Early applications were based on the extended Kalman filter [25] using landmarks in the sensor data, whether sonar [13] [14] or visual [3] [7]. The quadratic computational complexity of the EKF has made finding other methods to handle large-scale maps a major concern (e.g. [9] [10] [15] [29]), and EKF-SLAM is no longer used in its general form in field robotics. However, it remains well-suited to wearable vision using sparse landmark points. First, for wearables, sparseness of representation is no hindrance to navigation — one can rely on the wearer to get around. Secondly, the need to impose a limit on the

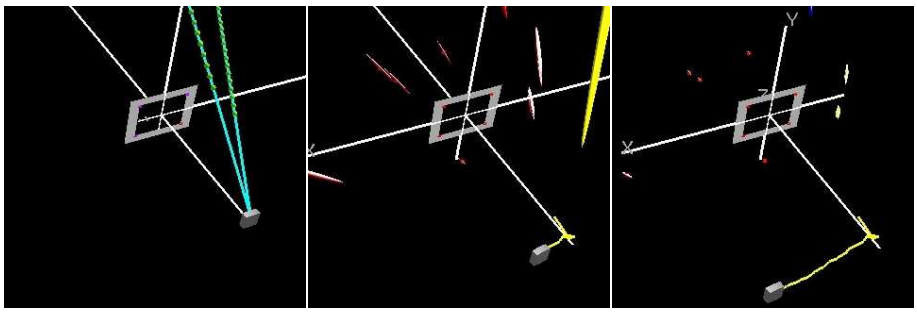


Figure 1: Typical initialization and evolution of structure and camera track in monoSLAM.

growth of the feature map in order to maintain video-rate performance is quite compatible with the notion of a local "workspace" of fixed volume around the wearer. Thirdly, such points may be annotated or recognized as points or objects of intrinsic interest to the wearer. Sparseness does make for fragility however. In monoSLAM with unconstrained camera motion, depth is not recovered from a single view or multiple views of a single point. Information comes from all points collectively, but, as processing has to be completed in a fixed time, a limit must be imposed on the feature map size.

In this paper we use the EKF monoSLAM formulation of Davison [6], [8]. The state is $\mathcal{X} = [\boldsymbol{c}, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]$ where the $\boldsymbol{X}$ are 3D locations of map features, and $\boldsymbol{c} = [\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\omega}]$ is the camera position, orientation, translational velocity and angular velocity, all defined in the world frame. The usual non-linear state update equation, $\mathcal{X}_{k+1} = \boldsymbol{f}(\mathcal{X}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k$, from time-step k to $k+1$ is assumed, where $\boldsymbol{u}_k$ is a control input, and $\boldsymbol{e}_k$ is an uncorrelated zero-mean Gaussian noise sequence. Here, as there is no source of odometry, the control input is taken to be zero. In the update, the 3D positions are assumed to be static, but the camera's state is updated according to a constant velocity model. The projections of the scene points are assumed to be related to the state at time-step k by $\boldsymbol{m}_k = \boldsymbol{h}(\mathcal{X}_k) + \boldsymbol{d}_k$ where $\boldsymbol{d}_k$ is an uncorrelated zero mean Gaussian noise sequence. The standard EKF update of the state and fully populated covariance matrix is followed.

For this implementation, "standard" [1] features for (potential) insertion into the 3D map are detected with the Shi-Tomasi saliency operator [23], and features that are eventually inserted are stored with an 11×11 pixel appearance template. Active search for correspondence is made within the predicted match region using normalized sum-of-squared difference correlation. Standard features are initialized using an inverse depth representation, using the state representation of Montiel *et al.* [21]. Figure 1 shows a typical view of recovered structure and camera track from monoSLAM that underpins the recognition process discussed below.

3 Object detection and identification

The aim now is to detect and identify known objects in the scene and to determine their location in the world frame from just a single image, while maintaining frame-rate operation. The location of a detected object will serve as an extra measurement for the SLAM

[1] The description standard merely distinguishes features used for SLAM from those used for recognition.

process. To unify recognition and localization, a point-based representation is adopted throughout, and ideally the same point features would be used for both purposes. However, Shi-Tomasi is insufficiently discriminating for recognition, making necessary a more robust method, invariant to scale and orientation changes. For this we adopt Lowe's SIFT [16], which is known to perform well, but is too computationally expensive for frame-rate operation.

3.1 The object database

The database includes at present only planar objects. To construct an entry, an image of the object is captured and, after correcting for radial distortion, SIFT descriptors $\boldsymbol{\sigma}^i$ and their positions $\boldsymbol{x}^i$, $i = 1...I$ are computed. The image need not be fronto-parallel, and so the homography $\mathtt{H}$ between the scene and image is found by choosing $n \geq 4$ image points whose corresponding scene points $\boldsymbol{X} = [X, Y, 1]^\top$ can be located in a object-based Euclidean plane. The database entry

$$\mathcal{O}_j = \{\mathcal{I}_\mathrm{R}, \{\boldsymbol{\sigma}^i, \boldsymbol{X}^i = \mathtt{H}^{-1}\boldsymbol{x}^i\}_{i=1...I}, \{\boldsymbol{X}^k_\mathrm{B}\}_{k=1...K}, \{k_1, k_2, k_3 \in 1...K\}\}_j$$

contains (i) the image $\mathcal{I}_\mathrm{R}$ of the object rectified by the homography so that it appears as a fronto-parallel view, (ii) the list of SIFT descriptors and their scene locations, (iii) the locations of several scene boundary points $\boldsymbol{X}^k_\mathrm{B}$ to define the object extent, and (iv), as explained later, the indices of three boundary points flagged for use in the SLAM map.

3.2 Object detection and localization

During a run, a video frame is selected at regular intervals and SIFT features are extracted. The detected feature locations are corrected for radial distortion[2], and are then matched to the stored keypoints of the known objects. Candidate matching descriptors are found using a pre-computed kd-tree based method [16] to search the database. If the number of matched points from any given object's database entry to the current image is greater than a threshold, 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 $\boldsymbol{X}$ and currently observed image points $\boldsymbol{x}$ are related by a plane-to-plane homography $\boldsymbol{x} = \mathtt{H}'\boldsymbol{X}$. RANSAC is used to estimate the homography $\mathtt{H}'$ and, if a sufficiently large consensus set is found, we infer that the database object is visible in the current frame.

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 $Z = 0$, and 3D homogeneous points on the object are $\boldsymbol{X}^{(4\times1)} = [X, Y, 0, 1]^\top$. In any view, the projection can therefore be written in terms of extrinsic and intrinsic parameters as $\boldsymbol{x} = \mathtt{K}[\mathtt{R}|\boldsymbol{t}]\boldsymbol{X}^{(4\times1)}$. Hence $\boldsymbol{x} = \mathtt{K}\mathtt{A}\boldsymbol{X}$, where $\mathtt{A} = [\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}]$ contains the translation $\boldsymbol{t}$ and the first two columns of the rotation matrix $\mathtt{R}$, all modulo a scaling factor. Using the homography already computed as the output of RANSAC and assuming known camera calibration $\mathtt{K}$, $\begin{bmatrix}\boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t}\end{bmatrix} = \mathtt{K}^{-1}\mathtt{H}'$, again up to scale. Because the estimate $\mathtt{H}'$ 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). The closest rotation matrix, and hence the overall scale for the translation, is determined using singular value decomposition.

[2] This is faster than undistorting the whole image, and the distortion is not significant enough to effect SIFT.

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 apply this transformation in reverse to place the object in the frame of reference of the camera at the time the image was selected; and then apply a further transformation determined by the camera's pose at the time of capture relative to the world coordinate frame defined by the SLAM map to derive the position of the object in world coordinates.

3.3 Adding recognized object locations to the SLAM map

A number of methods for adding objects to the 3D map can be envisaged. The straightforward, but certainly effective, approach used here is to allow the recovered 3D position of the planar object to define 3D point measurements. The feature positions themselves are not entered, but instead the three points $\boldsymbol{X}_{\mathrm{B}}^{k}$, $k = k_1, k_2, k_3$ from the object's boundary designated in the object database entry are used. For example, for the rectangular pictures used in experiments, three of the four corners are inserted into the map. The benefits in this approach are, firstly, 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, there is no reliance on any particular SIFT features being re-measured over time. Thirdly, the boundary points provide a convenient representation of the extent of the object for graphical augmentation.

4 A novel implementation with delayed object insertion

The detection, localization, and SLAM methods have been re-implemented to take advantage of the capabilities of a dual core processor (2.13 GHz Intel Pentium Core 2 Duo). Including operating system overheads, monoSLAM, executing on one core with around 20 point features, takes approximately 10 ms for a 640×480 image, leaving some 20 ms per frame to perform any further computation. Object detection and localization is run in a separate thread on the second core, continuously grabbing and processing frames.

For a typical frame, SIFT detects around 500 keypoints and takes on average 700 ms to complete. Matching against a database of 16 objects containing 3.2×10^4 features takes around 100 ms. While the point based SLAM runs at 30 Hz the object detection runs at around 1.5 Hz at best. These timings will of course vary with the size of the database, the number of features found in a frame, and the number of objects found in the scene.

4.1 Delayed object insertion

Because object detection takes a variable amount of time, and because it runs much more slowly than SLAM, the process must be done in the background — that is, it must always defer to the needs of monoSLAM to run at frame-rate. A mechanism is required to permit measurement updates using recognized objects at *whatever* time the detection and recognition processes manage to complete processing a frame.

We use the delayed decision making proposed by Leonard and Rikoski [12]. Suppose the single camera SLAM system runs as normal, and that at some time step k the object recognition and object localization module described in §3.2 is able to start processing. At this point the current state vector is augmented by the camera pose, $\boldsymbol{s} = [\boldsymbol{t}, \boldsymbol{q}]$,

$$\mathcal{X}^{\mathrm{A}} = \begin{bmatrix} \boldsymbol{c} & \boldsymbol{s} & \boldsymbol{X}_1 & \cdots & \boldsymbol{X}_n \end{bmatrix}^{\top} , \tag{1}$$

No.	Object label	No. of keypoints	Image Size	Metric Size (m)
1	Colosseum	2562	480×640	0.198×0.264
2	Durdle Door	3026	600×480	0.246×0.198
3	Grasshopper	1362	600×480	0.246×0.198
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
14	Multiple View Geometry	1245	446×637	0.174×0.247
15	Pansy	940	600×480	0.246×0.198
16	Pots of Fire	596	480×640	0.198×0.264
	Total	31910		

Table 1: Database objects, keypoints, and the sizes.

initialized to the current pose value $\boldsymbol{s}_k$. The covariance matrix is similarly augmented

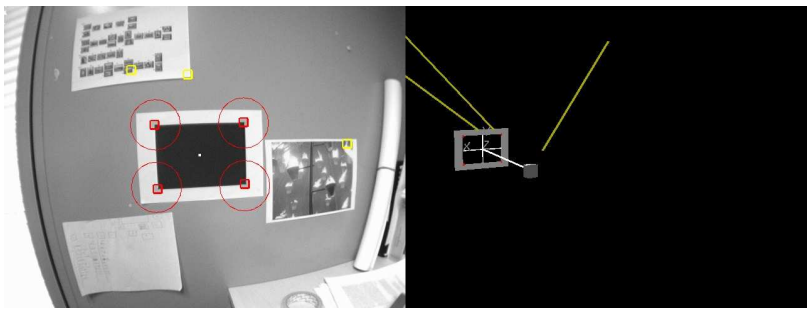
$$\mathtt{P}^{\mathrm{A}} = \begin{bmatrix} \mathtt{P}_{cc} & \mathtt{P}_{cs} & \mathtt{P}_{cX_1} & \cdots & \mathtt{P}_{cX_n} \\ \mathtt{P}_{sc} & \mathtt{P}_{ss} & \mathtt{P}_{sX_1} & \cdots & \mathtt{P}_{sX_n} \\ \mathtt{P}_{X_1c} & \mathtt{P}_{X_1s} & \mathtt{P}_{X_1X_1} & \cdots & \mathtt{P}_{X_1X_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathtt{P}_{X_nc} & \mathtt{P}_{X_ns} & \mathtt{P}_{X_nX_1} & \cdots & \mathtt{P}_{X_nX_n} \end{bmatrix}, \tag{2}$$

where $\mathtt{P}_{sc} = \mathtt{P}_{cc}[\partial \boldsymbol{s}/\partial \boldsymbol{c}]^\top$. After the saved camera pose has been added to the state, its value can no longer be directly measured. However, the correlation values contained in $\mathtt{P}^A$, between this saved pose and other elements of the state, enable its value to be updated as EKF updates continue. Therefore, as the state continues to be updated, the saved pose will be refined such that it remains consistent with newer state estimates. Once the object detection and localization completes, say n frames later, the updated saved camera pose $\boldsymbol{s}_{k+n}$ is used to determine the position of any recognized objects in the world, rather than $\boldsymbol{s}_k$. Then the saved pose is deleted from the state vector and covariance matrix. Although only one saved state is used here, the mechanism allows for multiple detection processes to start and finish at different times, were further processors available.

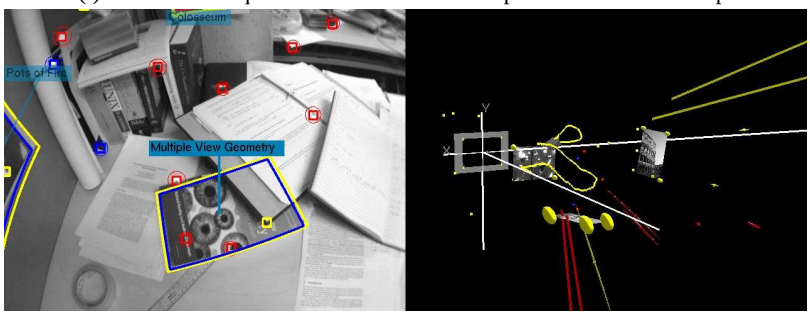
Using a saved camera pose to calculate the location of objects relies on the monoSLAM system maintaining a good estimate of the camera pose and trajectory during the intervening frames. In [4] it was shown that the inclusion of recognized object locations improved the quality of the map, and examples of object localization rescuing a failing SLAM process have been observed. However, this cannot be relied upon owing to the varying and relatively long time between object measurements. This delayed insertion method provides a faster and less complex update compared with the alternative of rolling back the EKF, inserting the measurement, and then rolling forward by recalculating all measurements from the frame the object detection was performed on.

5 Experimental evaluation

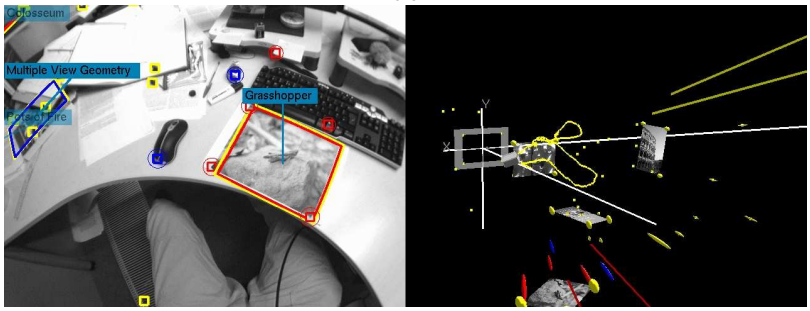
In the tests of the system reported here, a database of 16 planar objects with a total of 31,910 features was used (a sample of which is shown in Table 1), but only a subset of these objects appear in the scene. The database was created by running SIFT on each object image to generate the keypoints and measuring the metric sizes of the objects.



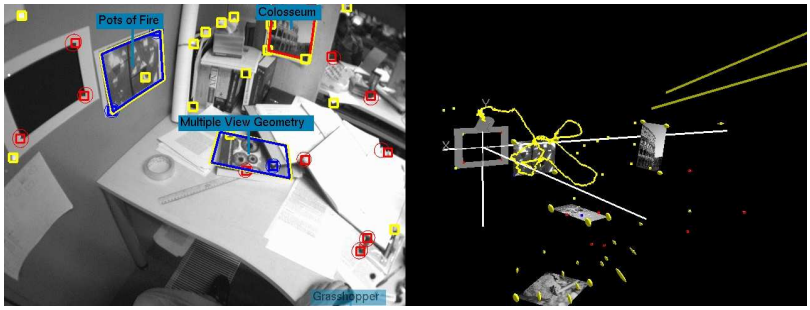
(i) Start of the sequence with the calibration plate visible in the map.



(ii) Two objects already initialized (Pots of Fire and Colosseum) and Multiple View Geometry just initialized.

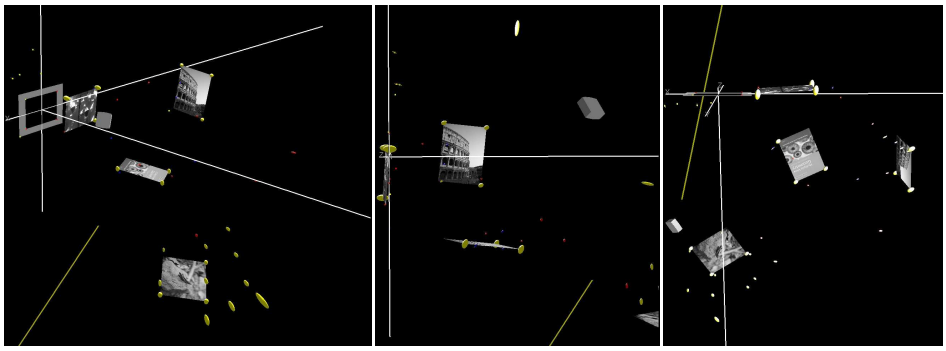


(iii) Final object (Grasshopper) located.



(iv) All objects have been detected and successfully localized.

Figure 2: The sequence runs from top to bottom with the camera view shown on the left and the map on the right.



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

Figure 3: (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. See Table 2.

Object label	Actual angle (°)	Measured angle (°)	Error (°)
Colosseum	90	91.2	±6
Grasshopper	90	84.7	±3
Multiple View Geometry	90	87.1	±3
Pots of Fire	0	5.0	±9

Table 2: Angles between the calibration plate and the objects.

Fig. 2 shows the evolution of processing, from initial calibration of the SLAM system to a time when there are four recognized planar objects in the SLAM map. The 2D views show the automatically generated 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 recognized objects represented by their database image.

Fig. 3 shows various views around a particular 3D map in which there are four picture objects, one of which (Pots of Fire) should be coplanar with the calibration plate (and hence in the XY-plane), two of which (Multiple View Geometry, Grasshopper) are in the XZ plane and the final object (Colosseum) is in the ZY plane. It can be seen that all of the objects are in their respective planes to within experimental error. Table 2 shows the angles between the planes recovered from the SLAM map. 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.

6 Discussion

This paper has described a system able to detect and recognize planar objects using appearance-based methods and to insert both their geometry and identity into a map — a map which is initialized and updated by an underlying monocular SLAM process which runs at fixed frame-rate using for the most part more cheaply-computed features. In particular here, the variable and relatively slow rate of delivery of geometry from the recognition process has been properly accommodated in SLAM's statistical framework using

Leonard and Rikoski's method of delayed decision-making, which inserts a temporary "place-holder" location in the state and covariance. This is updated during the time the recognition takes to complete, and is then deleted once it has been used to calculate the geometry of recognized objects. The paper demonstrates the system working in a desk top environment, providing automatic feedback on location and identity to the user.

Two avenues of application are being explored, one in the area of hand-held cameras, the second using an active wearable camera. With input from a hand-held camera, the system has no direct control over what imagery is captured. We are exploring guiding the user to different parts of the scene to search for new or already discovered objects using directional feedback provided on screen and by auditory instruction. Street frontages and art galleries are areas where the use of planarity is not a particular constraint to experimentation. When an active wearable camera supplies the imagery, the system has some autonomy to explore the world itself. As mentioned in the introduction, in [18] 3D point positions in the map were hand-labelled to allow a remote operator to command an active wearable to fixate on objects of interest while continuing to map. This method can now be automated to command the system to locate and fixate upon particular objects, without intervention of the wearer. Another avenue of exploration is that of extending the method to non-planar objects. There seems no fundamental impediment to doing so.

7 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 insightful conversations with members of the Active Vision Laboratory.

References

[1] U. Anliker, J. Beutel, M. Dyer, R. Enzler, P. Lukowicz, L. Thiele, and G. Tröster. A systematic approach to the design of distributed wearable systems. *IEEE Trans on Computers*, 53(8):1017–1033, 2004.

[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] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[4] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proc Int Conf on Robotics and Automation, Rome, Italy, April 10-14, 2007*, pages 4102–4107, 2007.

[5] 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*, volume 6, pages 3037–3040, 1999.

[6] 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.

[7] A.J. Davison and D.W. Murray. Mobile robot localisation using active vision. In *Proc 5th European Conf on Computer Vision, Freiburg, Germany, May*, pages 809–825. Springer-Verlag, 1998.

[8] A.J. Davison, I.D. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted for publication, 2007.

[9] M.W.M.G. Dissanayake, P.M. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[10] Joss Knight, Andrew Davison, and Ian Reid. Towards constant time SLAM using postponement. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems, Maui, HI*, volume 1, pages 406–412. IEEE Computer Society Press, October 2001.

[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] J. Leonard and R. Rikoski. Incorporation of delayed decision making into stochastic mapping. In *D. Rus and S. Singh, editors, Experimental Robotics VII, Lecture Notes in Control and Information Sciences. SpringerVerlag*, 2001.

[13] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, Boston MA, 1992.

[14] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(8):286–298, 1992.

[15] J.J. Leonard and P.M. Newman. Consistent, convergent and constant-time SLAM. In *Int Joint Conference on Artificial Intelligence, 2003*, volume 18, pages 1143–1150. Morgan Kaufmann, 2003.

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

[17] S. Mann, D. Chen, and S. Saman. HI-Cam: intelligent biofeedback signal processing. In *Proc 5th IEEE Int Symp on Wearable Computing, Oct 2001*, pages 178–179, 2001.

[18] 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, *International Symposium on Robotics Research, Siena, Italy, October 19-21, 2003*, volume 15, pages 325–334. Springer, 2003.

[19] W.W. Mayol, B.J. Tordoff, and D.W. Murray. Wearabale visual robots. *Personal and Ubiquitous Computing*, 6:37–48, 2002.

[20] J. Meyer, P. Lukowicz, and G. Tröster. Textile pressure sensor for muscle activity and motion detection. In *Proc 10th IEEE Int Symp on Wearable Computers, Montreux, Switzerland, Oct 11-14, 2006*, 2006.

[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] D. Raskovic, T. Martin, and E. Jovanov. Medical monitoring applications for wearable computing. *The Computer Journal*, 47(4):495–504, 2004.

[23] 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.

[24] A. Smailagic and D. Siewiorek. System level design as applied to CMU wearable computers. *Journal of VLSI Signal Processing Systems*, 21(3), 1999.

[25] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[26] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[27] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Proc 2nd IEEE Int Symp on Wearable Computing, Pittsburgh PA, Oct 19-20, 1998*, pages 50–57, 1998.

[28] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge MA, 2005.

[29] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[30] J.A. Ward, P. Lukowicz, G. Tröster, and T.E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, 2006.

[31] T. Westeyn, P. Presti, and T. Starner. ActionGSR: A combination galvanic skin response-accelerometer for physiological measurements in active environments. In *Proc 10th IEEE Int Symp on Wearable Computers, Montreux, Switzerland, Oct 11-14, 2006*, pages 129–130, 2006.