Interaction between hand and wearable camera in 2D and 3D environments

W.W. Mayol, A.J. Davison, B.J. Tordoff, N.D. Molton and D.W. Murray
Department of Engineering Science, University of Oxford
Parks Road, Oxford OX1 3PJ, UK

Abstract

This paper is concerned with allowing the user of a wearable, portable, vision system to interact with the visual information using hand movements and gestures. Two example scenarios are explored. The first, in 2D, uses the wearer's hand to both guide an active wearable camera and to highlight objects of interest using a grasping vector. The second is based in 3D, and builds on earlier work which recovers 3D scene structure at video-rate, allowing real-time purposive redirection of the camera to any scene point. Here, a range of hand gestures are used to highlight and select 3D points within the structure and in this instance used to insert 3D graphical objects into the scene. Structure recovery, gesture recognition, scene annotation and augmentation are achieved in parallel and at video-rate.

Keywords: wearable computing, active vision, gesture determination, SLAM

1 Introduction

The emphasis of much work in wearable *computing* is, reasonably enough, on the creation of wearable *computers*. But to advance beyond the weaving of processors, memory and buses into this or that garment requires the development of sensing and interfaces that provide "first-person" feedback on the interaction between wearer and environment. Visual sensing fits comfortably into the wearable computing paradigm. It offers many non-intrusive ways of recovering the cues that allow humans to follow action, to react to events, to navigate, to touch, to grasp, and so on. Moreover, mobile CPU power has now grown such that serious video rate applications have become feasible.

In [9] we began to combine techniques from active vision with wearable computing, producing the lightweight, controllable wearable camera pictured in Figure 1. The degree of independence from the wearer of an active wearable (which is of course absent in a body-fixed camera of the sort used by [12, 8, 1]) allows it to address three sets of frames. The first relates to the wearer's activity, say sensing the manipulation space in front of the wearer's chest, or pointing where the wearer's head is pointing. The second set includes frames aligned with the static surroundings — the camera might wish, for example, to maintain alignment with the vertical; and last, the frame attached to an independent moving object, particularly relevant to object tracking.

The visual routines initially developed for the active wearable camera concentrated on stabilisation and 2D tracking, and might be described as purely reactive, in that the wearable camera merely responded to change in the image and scene [9]. However, in [10] it was shown how a wearable active camera might gain greater autonomy by controlling fixation and saccade (gaze holding and switching) within a 3D representation of its surroundings. That 3D structure was recovered in real-time using a single camera simultaneous localisation and mapping (SLAM) method [3], also recovering the camera

BMVC 2004 doi:10.5244/C.18.78

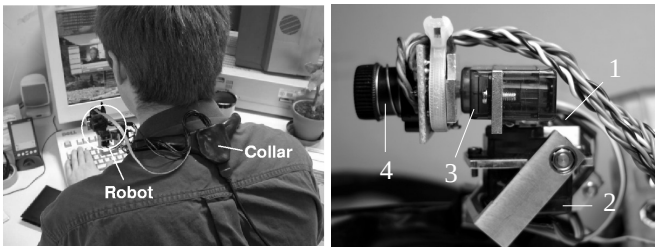

Figure 1: The wearable camera has three motorised axes, and is worn on the shoulder. The labels are 1) tilt axis. 2) pan axis. 3) cyclotorsion axis. 4) colour image sensor. The interface containing the data transceiver, microcontrollers and battery is worn at the hip.

position over time. This allows the active wearable camera to fixate and stabilise with respect to any 3D point, real or virtual, using the device's inverse kinematics rather than relying on image-based tracking.

Whilst all previous work has focused on fully automatic control of the camera, in this paper we introduce interaction, allowing the user's hands to guide the wearable camera in two ways. The first is rooted in two dimensions, and uses hand-detection to control the wearable's direction of gaze, and grasp-detection to indicate objects of interest. The second method uses the same hand-detection scheme, but now selects points within a simultaneously recovered 3D structure. Although demonstrated here initialising graphical objects in the structure, a longer term goal is to allow easier annotation of structural points. Annotating 3D points with, for example, object identity would make for more natural interaction between wearer and wearable, or with a remote collaborator if used as a teleoperated device. There is little realistic prospect of achieving this automatically using the rather sparse visual representation that the complexity of SLAM imposes, and so an expedient is to use annotation by hand, but with the hand and its configuration detected automatically.

Section 2 describes the method of hand detection and grasp determination, and is followed in Section 3 by examples of the method being applied in two dimensions. Section 4 gives an overview of the method of structure recovery using SLAM, and an experimental example of the method of interaction between wearable and 3D structure is given in Section 5. The paper closes with conclusions and a brief discussion of future possibilities.

2 Detection of hand and gesture

Two dimensional techniques for finding objects of a known shape include the use of image patch correlation, image moments or error minimisation with morphable models. In the case of hands from a wearable camera, in [12], a hat-mounted camera and a feature vector based in image moments is used for a sign language recognition task while in [7] a mixture of Gaussians and a mean-shift tracker is used for detecting the wearer's hand within a cursor-and-click interface. With a mobile camera we need to be robust to a greater number sources of shape distortion — such as non-contiguity due to occlusion and colour misclassification, orientation changes, and variations in basic hand shape — and also be able to initialise automatically. We have therefore devised a more specialised detector.

2.1 Preprocessing

The first step in detection consists of skin colour segmentation. The camera provides images with separate luminance (Y) and colour (UV) channels, so that building a 2D colour histogram in the UV space provides a degree of illumination invariance. The histogram bins are populated from training exemplars of various classes C_i, allowing the conditional probabilities $p(c|C_i)$ that an arbitrary pixel colour $c(x,y)$ originated from class C_i to be determined. The classification likelihoods for this colour are then

$$p(C_i|c) = p(c|C_i)p(C_i)/\sum_{j=1}^{N} p(c|C_j)p(C_j) \tag{1}$$

and classification determined by "winner takes all". Here we adopt the simplest case of just two classes, skin and background, and assume uniform priors. After classifying image pixels, high frequency noise is removed by spatial filtering, and the skin image $S(x,y) = +1, -1$, for skin and background respectively, passed on for shape detection.

Although UV space is relatively robust to lighting changes and to variation in skin colour, classification errors are reduced, and hence process time saved, by colour balancing to the current user, lighting and camera for a particular run. The maximum M in a colour channel k is determined from a view of the skin under the current conditions, and, denoting the peak on that colour channel in the training set as $M_T(k)$, the channel value for each measured pixel value is corrected by scaling,

$$c'(k) = (M_T(k)/M(k))\ c(k)\ . \tag{2}$$

2.2 Hand Shape Detection

The detector exploits the general shape of a hand — a solid mass of skin colour for the palm and narrow promontories of colour for the fingers. Setting the detector at $\mathbf{x} = (x,y)$ generates a feature vector γ by sampling the skin image every π/K radians around each of a set of $N = 5$ concentric rings centred at $\mathbf{x}$ and each doubling in radius, as shown in Figure 2(b). The radius of the smallest ring is optimised experimentally as is the value of K, and the entire detector scaled thereafter by $\sqrt{A_t/A_0}$ where A_t and A_0 are the areas of skin in the current and fiducial images.

A gesture template $\overline{\gamma}$ is generated from training imagery in which the wearer aligns the principal direction of the gesture with the image's y-axis and then selects a class label. The feature vector γ is compared with a template $\overline{\gamma}$ to determine its similarity

$$g(\gamma,\overline{\gamma}) = \gamma \cdot \overline{\gamma}/L\ , \tag{3}$$

where $L = 2KN$ is the length of the templates, giving $0 \leq g(\cdot) \leq 1$. The aim is to find the optimal position and orientation of the detector

$$\mathbf{x}', \phi' = \arg\max_{\mathbf{x},\phi}\ g(\gamma,\overline{\gamma}), \tag{4}$$

along with its identity if multiple detectors are deployed. A further threshold τ is applied to $g(\cdot)$ to decide if there is enough support for class membership.

In practise, the result of the skin classifier is often sufficiently clean for the optimum $\mathbf{x}'$ to lie close to the centre of mass of the skin image, and only ϕ has to be optimised. When this is not the case, to avoid a costly dependence on ϕ, a rotationally invariant feature vector is derived by computing the difference between the resulting response curves

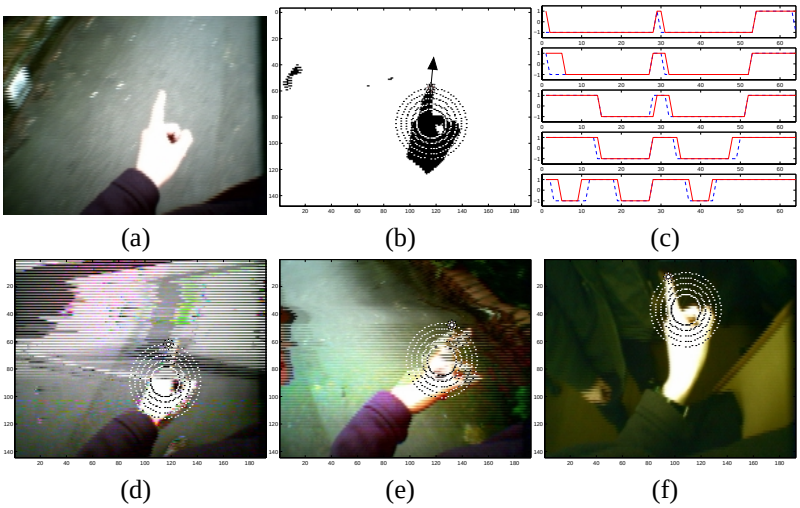

Figure 2: a) View of the hand. b) the shape detector locates the pointing gesture c) the ring curves where high response means hand area and low response background; dashed shows the template and solid the current signal after best alignment. (d-f) Challenging images: with shatter, noise interlacing and changed hand shaped (sleeve retracted). The responses were $g = 0.86, 0.85, 0.81$ respectively.

(Figure 2c). A vector $\mathbf{h}$ of similarities between the ring signals is calculated (here for $N = 5$)

$$\mathbf{h} = [h_{1,2}, h_{1,3}, h_{1,4}, h_{1,5}, h_{2,3}, h_{2,4}, h_{2,5}, h_{3,4}, h_{3,5}, h_{4,5}] \tag{5}$$

where $h_{l,m}$ is the similarity between the $2K$-dimensional ring signals l and m defined as

$$h_{l,m} = \sum_{k=1}^{2K} R_l(k) R_m(k) \tag{6}$$

where $R_l(k)$ is the k-th entry in the l-th ring. Note that $\mathbf{h}$ is invariant to rotation since it is a descriptor calculated with the shape itself, and thus invariant to column permutations. The position $\mathbf{x}$ is first optimised by search from the centre of mass, and then, if required, the original rotationally sensitive detector used to optimise ϕ. Figure 2(d,e,f) shows that the detector functions under quite different sources of image noise, and still produces relatively high values of g. When there is no hand in the current image, the response g is much lower, and thresholding is satisfactory (at least empirically).

2.3 Accuracy in orientation and classification

Accuracy in pointing orientation was measured by aiming the gesture in Figure 2(a) to 13 targets with angular spacings of 5°. An average of 10 aimings per target were recorded without providing any feedback from the system to the wearer. Ground truth was gathered by manually measuring in the image the difference between the target and the pointing line computed by the algorithm. The resulting mean error in aiming precision was $4.6 \pm 2.2^\circ$.

For testing the gesture identification performance, the ten numerals of the American Sign Language (ASL) were trained from a single example for each class. Receiver operator characteristic (ROC) curves were then computed and plotted in Figure 3, summarising

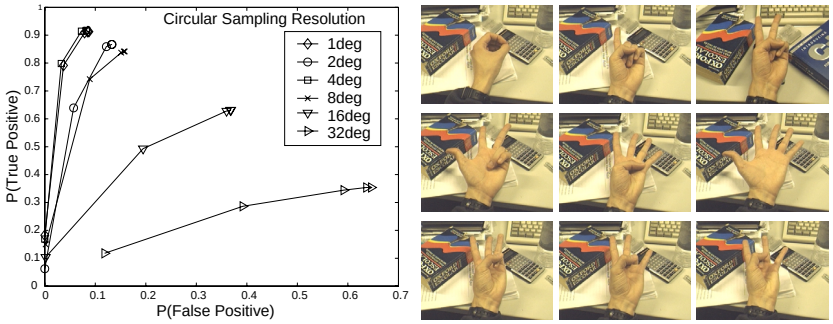

Figure 3: ROC curves for different circular sampling densities and variation of the membership threshold used in the classification of the 10 numerals of ASL, 9 of which are shown on the right.

classification performance of the more than 1100 frames evenly sampled from a seven minute long sequence of continuous gesturing under background clutter, gesture in plane rotation, scale variation and mild occlusion. The different curves correspond to different circular sampling resolutions used to generate the templates as the membership threshold τ is varied from 0.1 to 0.9. Peak performance is reached with a circular sampling resolution of at least $4^\circ (K = 90)$ and $\tau > 0.7$. Overall probability for a true positive (sensitivity) is then grater than 0.91 with a 0.92 probability to not mis-classify (specificity).

3 Implementation and experiment in 2D

3.1 Implementation

The algorithms have been implemented in C++ and run on a laptop with a 1.7GHz Intel Centrino processor. As well as running the skin detection and gesture recognition processes and later the the SLAM process concurrently, it runs a text-to-speech (TTS) synthesiser [2] that provides a feedback on the state of the system to the wearer. The visual algorithms and TTS server run as independent processes.

To improve the accuracy of the hand gesture recognition, the raw recognition results at each timestep are fed into a simple inference engine. A modal filter is used to select the gesture which is most consistently recognised over recent time-steps and the resulting estimate tested against a state-transition table.

The wearable active camera used for these experiments has a horizontal field of view of about $42^\circ$and transmits 320×240 YUV images at 30Hz over an IEEE1394 interface. An identical camera but with a wider view lens of about 80°horizontal was also tested. This latter device permits a better disambiguation of the wearer's motion, and allows the wearer's hand to operate closer to the image sensor.

3.2 Interactions in 2D

Using the active wearable camera and colour-based detection alone allows a simple but practical mode of visual selection. Figure 4 shows stills cut from the wearable camera's imagery as it tracks the hand reaching for a ball. If both hands are present, the midpoint of the two hands is followed instead. Additional colour classes in the segmentation allow simple object recognition, and voice feedback advises the wearer of the current state.

To illustrate the utility of the hand and grasp detection, Figure 5 shows an example using two templates, one trained for a grasping hand and the other for a closed hand. When

Figure 4: The wearable active camera follows the hand and recognises objects near to them as a cue for activity identification. The labels indicate the output from the TTS.

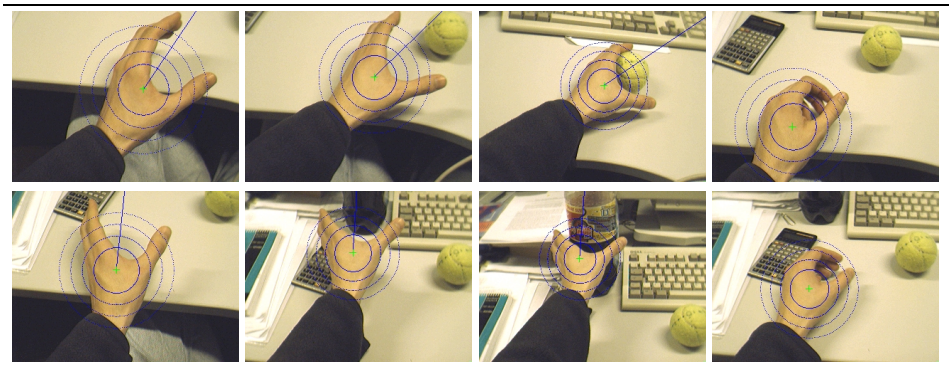

Figure 5: The wearable active camera keeps the grasping hand within view. Notice how the hand changes shape due to projection but the grasping vector (shown by the blue line) is still computable. A video is available from this paper's entry at www.robots.ox.ac.uk/ActiveVision/Papers

the hand is closed the centroid is used to redirect the camera. With the hand open, the direction of grasp is also computed. This might be used to drive the wearable's attention to an object, and has, as shown later, been used to select items of interest. The fourth frame shows the hand close after the move, and the grasp direction absent.

4 Wearable camera position and local 3D structure

One way of increasing the potential for autonomous operation of an active or passive wearable camera is to enable it to determine its location in an environment which is unknown a priori, and to re-localise itself within that environment at a later time.

Methods of real-time localisation have been addressed in wearable computing, but those using vision have used inertial sensing combined with imagery either from premeasured targets [6] or from custom fiducial markers [5]. In [10] visual sensing alone was used, and the single camera SLAM method [3] used general point features and required neither fabricated features nor a known map. Single camera SLAM is challenging

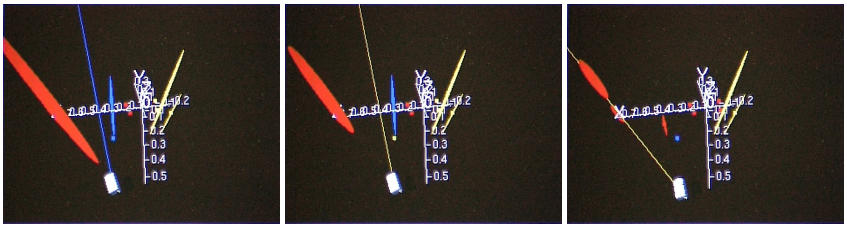

Figure 6: 3D models of the scene from a sequence showing the camera movement and evolving scene structure. Ellipses represent the scene points and their uncertainties.

because views of a single point fail to reveal depth when the camera motion is unknown. Multiple points are required, but as processing has to be completed in a fixed time some limit must be imposed on the size of the feature map. Here it is natural to restrict the workspace to a few cubic metres around the wearer. But the very proximity of camera and scene introduces a further difficulty: small motions introduce relatively large changes of viewpoint and occlusion is more likely.

The single EKF used is quite standard [3] and the briefest details suffice here. The state vector contains $\mathbf{X}_i$, the fixed 3D locations of map features i, and $\mathbf{c}(t) = (\mathbf{t}, \mathbf{q}, \mathbf{v}, \omega)$, the time dependent camera position, orientation, translational velocity and angular velocity, all referred to a fixed world frame. The update model is one of constant translational and angular velocities, allows for control input, and assumes that in each time step, white, Gaussian distributed impulses in velocity and angular velocity of $\mathbf{n} = \Delta t\,(\mathbf{a}, \alpha)$ are applied to the camera. These components are assumed uncorrelated so that the covariance matrix $\mathtt{P}_n$ of the noise vector $\mathbf{n}$ is diagonal.

New features for proposed addition to the 3D map are detected initially with the Shi-Tomasi saliency operator [11], and features actually added to the map are stored augmented by a 15×15 pixel appearance template, as in [4]. Given a predicted camera position, each feature $\mathbf{X}_i$ is projected into the new image along with its associated uncertainty region derived at the 3σ limit from the innovation covariance matrix $\mathtt{S}_i$, and searches made within the new image for correspondence using zero normalised cross correlation. In a fixed-time system, it pays to search first for the feature with highest $\mathtt{S}_i$ as this give greatest scope for reducing the uncertainty in the map as a whole.

Figure 6 shows 3D models of the evolving scene structure and camera position for a few frames from on-line sequence. When a point is first observed its error ellipse is infinitely extended along the line of sight and its error distribution is handled by factored sampling for the first few frames, before being entered into the map.

Map management criteria aim to keep the number of features observable in any one view to a number compatible with both the desired localisation accuracy and maintaining a 30Hz frame-rate. A feature is predicted to be observable from a 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 small enough for the image template to be meaningful. Features are added if the number of those actually observable falls to 5 or less, and a feature is deleted if its long-term ratio of actual observability after search to predicted observability falls below 1:2. Experience with this scheme indicates that it allows for the necessary degree of ephemeral clutter in the scene without introducing mismatches.

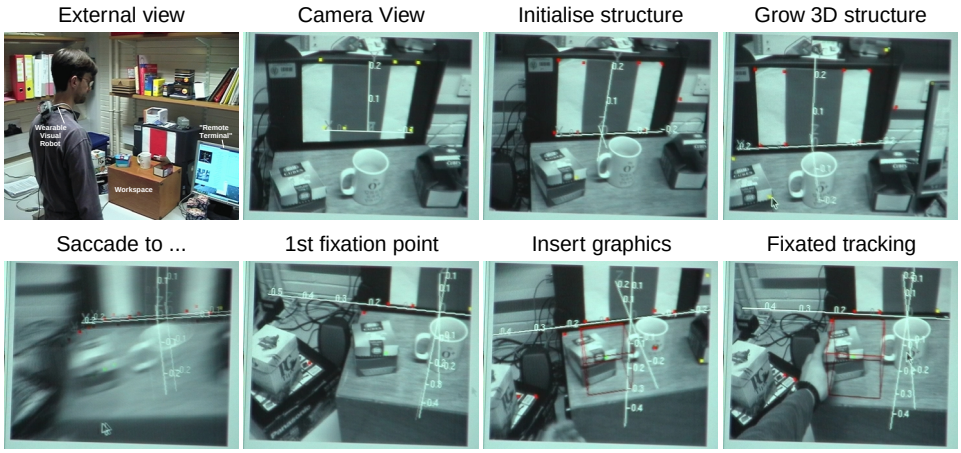

Figure 7: Saccade and fixation within a 3D structure recovered on the fly. Top left shows the experimental setup, and the remaining frames follow the camera's view of initialisation and tracking of features and saccades between them during continuous wearer motion. A video is available from this paper's entry at www.robots.ox.ac.uk/ActiveVision/Papers

5 Interaction in 3D

5.1 Active control of the wearable

The SLAM procedure just described is applicable to any single camera – a hand-held camera, a static wearable camera, robot-mounted or otherwise [3]. But for an *active* wearable camera, knowledge of its own position within the 3D scene allows control of the camera in two ways: (i) saccades, or controlled redirections of gaze between known 3D locations, even when the target location is out of the current field of view; and (ii) extended periods of fixation on a 3D location during wearer motion. Because SLAM recovers the 3D camera and point positions over time, the active wearable can fixate on any 3D point, real or virtual, using the device's inverse kinematics.

Figure 7 shows stills taken from an extended sequence viewed through the wearable camera as the wearer moves in front of a workbench scene [10]. The SLAM process determines 3D location and structure, and a remote operator determines when to redirect the wearable camera's gaze and when to use it to fixate on an object by clicking on one of two graphical displays. The first shows a three-dimensional reconstruction of the estimated locations of the camera and features, and the other shows the view from the camera augmented with a world coordinate frame and other graphics, including virtual objects.

5.2 Hand interaction with the 3D structure

Interaction during the previous experiment was with a remote operator, not the wearer, and required clicking on projections of 3D points on the screen using a mouse. Interestingly, operators found it difficult to click reliably on objects in the rapidly-moving camera view, much preferring to use the stable 3D view in the world coordinate frame for this task.

The alternative approach is for the wearer rather than the remote operator to annotate the scene. Unlike a remote operator, the wearer does not (and perhaps should not) have easy access to mouse and keyboard, and so may wish to use hand gestures alone.

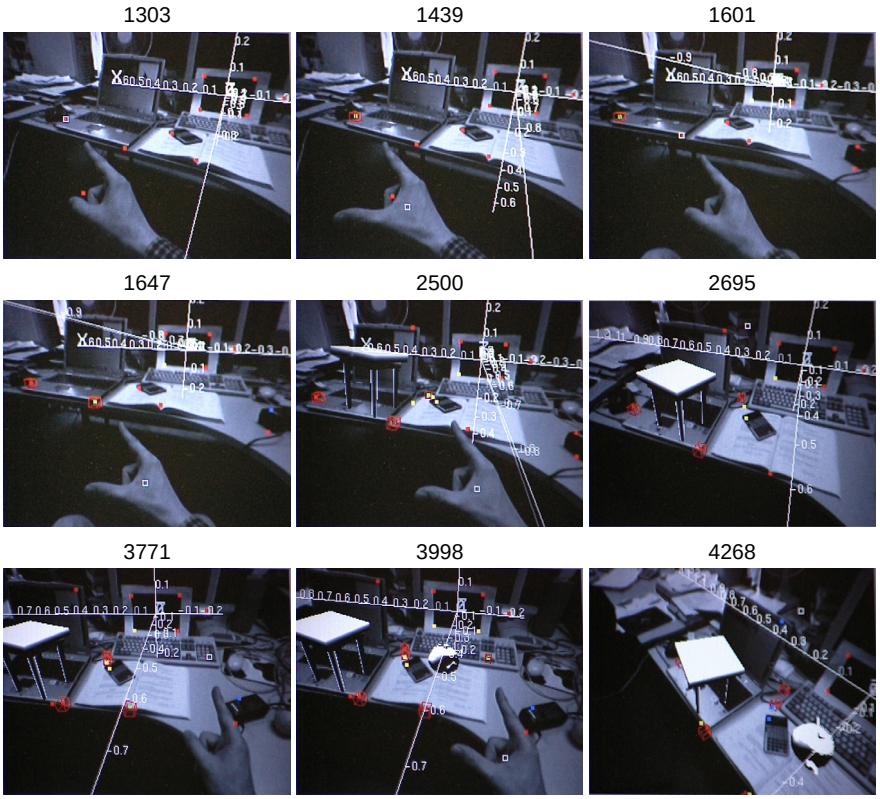

Figure 8: Frames from a sequence as the wearer annotates features and inserts virtual objects (a table and the GL teapot) on the fly. A video is available from this paper's entry at www.robots.ox.ac.uk/ActiveVision/Papers

This mode of operation is more natural, and helps avoid the need for "small" versions of desktop hardware which hinders many wearable computing applications.

The hand detection method has been implemented to run concurrently with simultaneous localisation and mapping. Here three gesture templates are used, a pointing one to move a selecting cursor, an "L" gesture to accept a selection, and an open hand to reject a selection. Figure 8 show stills taken from a sequence where the wearer points in the scene to select 3D points already mapped by the SLAM process. Here the points selected are not used to drive fixation or saccade, but are used instead to allow 3D graphical objects to be inserted into the scene.

At frame 1303 the first of three points to define a plane is highlighted by the pointing gesture and at frame 1439 it is accepted by recognising the L gesture. Further acceptances occur at frames 1647 and 2500. With the plane defined, at frame 2500 and onwards a small table appears inserted in the 3D map. Note that the inserted graphic is rigidly connected to the structure as the wearable camera moves around in the following frames. Further pointing and L gesture conjure up a teapot at frame 3998 and onwards.

6 Discussion and Conclusions

This paper has described methods of enabling the user of a wearable vision system to interact with the visual information as it is recovered, by making hand movements and gestures.

A method of hand detection was described which determines the similarity between a colour signature determined by training and that detected in the current image. Using multiple concentric rings allows both solid and intermittent regions of skin colour to be detected, indicating the configuration of palm and fingers.

Two applications areas have been explored. The first was rooted in two dimensions, and uses the hand both to guide an active wearable camera and to highlight objects of interest. Although not explored here, the grasping vector could be used to direct movements of the camera, or of an assistive robotic device, or to initialise object recognition.

The second application was based in 3D, and builds on earlier work which demonstrated scene structure recovery whilst making purposive movements of the camera. In this case the hand is not used to guide the camera, but rather to select 3D points within the structure, as it continues to be built up. The selected points are used here to insert 3D graphical objects into the scene. Our longer term aim however is to allow the wearer to point to and to annotate the structure with (possibly oral) object labels, so that the wearer and system can interact at a cognitive rather than geometrical level.

Acknowledgements

This work was supported by UK EPSRC Grants GR/N03266 and GR/R89080/01, by an EPSRC Advanced Research Fellowship to AJD and by a Mexican CONACYT scholarship to WWM.

References

[1] 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*, pp. 37–43, 1999.

[2] A. Black and K. Lenzo. Flite: a small fast run-time synthesis engine, ISCA, *4th Speech Synthesis Workshop*, Scotland. 2001

[3] 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.

[4] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.

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

[6] 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*, pp. 107–114, 2001.

[7] T. Kurata, T. Okuma, M. Kourogi and K. Sakaue. "The Hand Mouse: GMM Hand Color Classification and Mean Shift Tracking", *In Proc Second Int Workshop RATFG-RTS*. Canada, pp.119–124. 2001.

[8] S. Mann. WearCam (the wearable camera): Personal imaging systems for long-term use in wearable tetherless computer-mediated reality and personal photo/videographic memory prosthesis. In *Proc 2nd IEEE Int Symp on Wearable Computing, Pittsburgh PA, Oct 19-20, 1998*, pp. 124–131, 1998.

[9] W.W. Mayol, B. Tordoff, and D.W. Murray. Wearable visual robots. In *Proc 4th IEEE Int Symp on Wearable Computing, Atlanta GA, Oct 16-17, 2000*, pp. 95–102, 2000.

[10] W.W. Mayol, A.J. Davison, B.J. Tordoff and D.W. Murray. Applying Active Vision and SLAM to Wearables. Robotics Research The 11th International Symposium. Ed. P. Dario and R. Chatila. Springer Tracts in Advanced Robotics. 2004.

[11] 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*, pp. 593–600, 1994.

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