Directing the Attention of a Wearable Camera by Pointing Gestures

Teófilo E. de Campos⋆ Walterio W. Mayol Cuevas† David W. Murray⋆

⋆ Department of Engineering Science
University of Oxford
Oxford OX1 3PJ, UK
http://www.robots.ox.ac.uk/ActiveVision

† Department of Computer Science
University of Bristol,
Woodland Road, Bristol BS8 1UB, UK
http://www.cs.bris.ac.uk/∼wmayol

Abstract

Wearable visual sensors provide views of the environment which are rich in information about the wearer's location, interactions and intentions. In the wearable domain, hand gesture recognition is the natural replacement for keyboard input. We describe a framework combining a coarse-to-fine method for shape detection and a 3D tracking method that can identify pointing gestures and estimate their direction. The low computational complexity of both methods allows a real-time implementation that is applied to estimate the user's focus of attention and to control fast redirections of gaze of a wearable active camera. Experiments have demonstrated a level of robustness of this system in long and noisy image sequences.

1 Introduction

Recent technology allows the implementation of robotic systems that are light enough to be worn without inconvenience to the user. This leads to a wide range of applications from assistive technologies to entertainment and portable communication. Wearable active cameras provide views of the environment which are rich in information about the wearer's location, interactions and intentions. But the images from them present severe challenges because neither the sensor nor its underlying "platform" is stationary. Compounding these difficulties, most researchers use cameras that are more or less rigidly mounted to one or other body part — head, shoulder, chest and hand have all been used — making the imagery highly dependent on posture.

Mayol [7] developed prototypes for a miniature wearable active camera, and argued that mounting it at the shoulder gives an optimum location measured against field of view, independence from the wearer's movements, and, important in wearable applications, social acceptability. The ability to redirect the camera also allows switching between sensing contexts: one context may be foccused on the manipulative space; another may be the horizon, aligned with gravity; and a third may be fixated on an independently moving object. Such devices require a range of sensing and perceptual modalities. In [6] inertial and visual cues are used to stabilise gaze by detecting user and image motion. In [14] slaving the device from head motion is investigated.

In the wearable domain, hand gesture recognition is a natural replacement for keyboard and mouse-based input. In [11], for example, a hat-mounted camera is used for a sign language recognition task, and interestingly performs better than a wall mounted one, while in [4] a bare hand is used as a cursor-and-click device for interacting with menus displayed on a head mounted display. Pointing gestures are the main form of non-verbal communication, presenting a major complement to speech in human to human communication [10].

In this paper, we are interested in using the view from the wearable camera to detect and track pointing gestures in order to determine the focus of attention and redirect the camera. In order to allow natural user interface, it is necessary to use real-time algorithms. To that end, we propose a coarse-to-fine method for shape detection that is invariant to translation and rotation, but retains the ability to identify position and orientation of the pointing hand. Using a cyclic finite state machine, this detection method is combined with a fast three-dimensional tracker that is able to refine the pose estimate and add depth information about the position and orientation of the hand. Such parameters enrich the ability of the wearable camera to perform a saccade to the pointed area in 3D.

This work was supported by a CAPES (Brazil) scholarship to TdC, by a CONACYT (Mexico) scholarship to WWMC and by the EPSRC (UK) grant GR/S97774.

2 The Wearable Camera System

The wearable active camera consists of a miniature camera mounted at the end of a serial chain of three motorised axes. As shown in Figure 1, the device is mounted on a collar and lies just above the shoulder of the wearer, its location was found optimal against a number of criteria. Full details about the device's kinematics and spatial layout are given in [7].

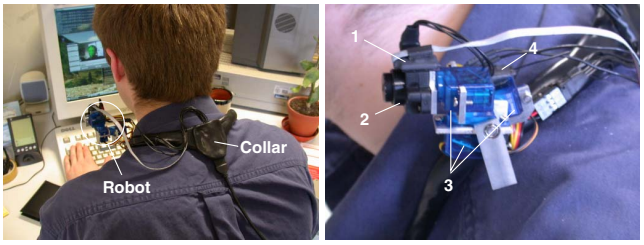


Figure 1. Wearable Visual Robot: (1) 2-axis accelerometer, (2) CMOS colour camera, (3) three motorised axes, (4) wireless video transmitter. The wearable interface box containing the data transceiver, microcontrollers and batteries is worn at the hip.

3 Locating Pointing Gestures Robustly

The hand detection algorithm is a coarse-to-fine matching method that is able to find the hand and also to estimate its pointing direction in the image plane without the need for scanning all the pixels. The tracking algorithm is essentially that of Harris [2], capable of recovering the rigid pose of the hand with information of depth and inclination, which enables the estimation of the pointing direction in the 3D world. The two processes are coupled in a finite state machine shown in Figure 2.

3.1 Preprocessing

The first step in detection consists of skin colour pixel segmentation. For this, a histogram-based classifier is used [3] in the chromatic channels of the YCrCb colour space (CrCb), which makes the method robust to brightness variations. Conversion from RGB to YCrCb is done by hardware in the camera. In the training process, skin colour samples were acquired from 134 facial images, most of which obtained from the Purdue University face detection database [5]. Background samples were acquired from images obtained with the wearable camera and other images obtained from the Internet. The CrCb colour space was populated

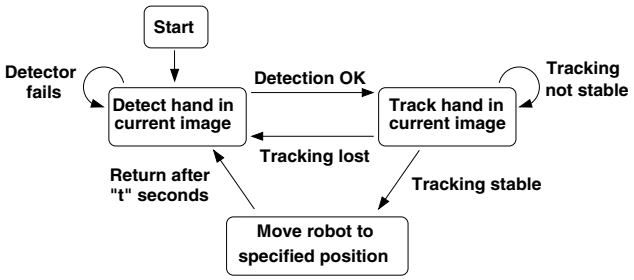


Figure 2. Finite state machine for our system.

with these samples and the likelihood of the classes was modelled using a 2D normalised histogram in this space. To model unknown colours, a third class with low likelihood was created, and this was labelled as background. Classification is done by maximum likelihood estimation.

If most of the background captured by the wearable camera is dark, then the automatic contrast normalisation of the camera can produce some saturated blobs in the hand image for Caucasian users, destroying the colour information in those regions. To finesse this problem, white saturated pixels were classified as skin. This reduces the false negative classification rate at a cost of increasing the false positive rate. But this is not critical because our method takes the global shape into account. If most of the hand silhouette is apparent, our method is able to locate it.

Noise is usually present in the images obtained from this wearable robot because an analogue wireless transmitter is used and the video is interlaced, so motion artifacts are often present. Such noise can be removed by applying a median filter with a 3×3 window, but this is not crucial to the performance of our hand pose estimation method. Figure 3 shows the results of this method for a challenging image.

3.2 Hand Shape Detection

Techniques for finding objects of a known shape include the use of 2D correlation, image moments, and specific spatial filters [1]. The first two methods work well when noise is small and when objects do not vary too much. But several kinds of distortion happen often in hand images: the hand can appear as a non-contiguous object due to occlusion and shadows; it can be in different orientations; other objects with similar colour and size can be present; and small variations in the hand shape can occur. To cope with these factors and with image noise generated by the wireless video transmission, a robust shape detector is needed.

Since the camera is located on the user's shoulder, the variation in the scale of the hand in the image is not expected to be very large, at least in the first frame of reference of the pointing gesture sequence. The detector uses

(a) (b) (c) (d) (e)

Figure 3. a) Original colour image; b) skin detection result; c) threshold result; d) combined (OR) image; e) filtered result.

the local shape descriptor presented in [8]. Given an image location, $r = 5$ rings with different radii centred at this location evaluate the skin classification value $\mathcal{C}$ of the image $\mathcal{I}(\cdot)$ at every π/K radians (in this experiment, $K = 32$), as shown in figure 4. A positive value ($\mathcal{C}(\mathcal{I}(i,j)) = 1$) in the curve indicates skin, a negative value ($\mathcal{C}(\mathcal{I}(i,j)) = -1$) indicates background. For rotation invariance, the descriptor builds a feature vector $\mathbf{x}$ where each element consists of a similarity measure between each possible pair of response curves, i.e.,

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

where $h_{m,n}$ is the similarity between the $2K$-dimensional curves m and n. Since the values of $\mathcal{C}(\mathcal{I}(\cdot))$ are either 1 or -1, $h_{m,n}$ is computed by

$$h_{m,n} = \frac{1}{2K} \sum_{k=1}^{2K} \mathcal{C}_{m,k} \mathcal{C}_{n,k} \ . \tag{2}$$

Note that $\mathbf{x}$ is invariant to rotation since it is a descriptor calculated with the shape itself, and invariant to column permutations.

A template $\overline{\mathbf{x}}$ is generated from a training image in which the user clicks on the metacarpophalangeal joint of the index finger and on the index finger tip[1]. This determines the centre and the orientation of the template, respectively. Upon application, a new sample vector $\mathbf{x}$ is compared with the template $\overline{\mathbf{x}}$ to determine the similarity $g(\mathbf{x}, \overline{\mathbf{x}})$ to the shape under search, determined by

$$g(\mathbf{x}, \overline{\mathbf{x}}) = \frac{1}{S} \sum_{i=1}^{S} x_i \overline{x}_i \ , \tag{3}$$

where $S = r!/2!(r-2)!$, where r is the number of rings used (here $S = 10$). Note that $g(\cdot) \in [0, 1]$. Thus, the detector is a function that tries to find the position of $\mathbf{x}'$ in the image $\mathcal{I}(\cdot)$, such that

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

[1]For the nomenclature of hand bones and joints, see [12].

The spacing between rings and the number of rings was determined experimentally. In practice, it was found that the spacing of 4 pixels between rings and the use of 5 rings (being 11 pixels the radius of the smallest circle) leads to the best trade-off between accuracy and computational effort for 144×192 images.

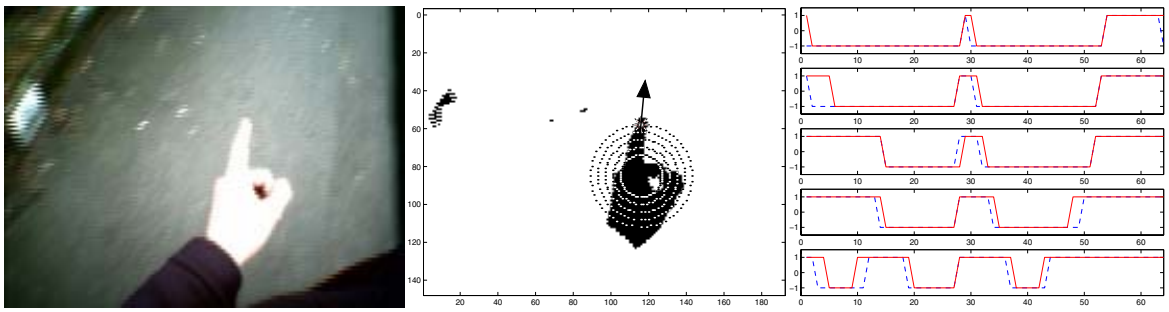
In order to speed up the detector, we propose a coarse-to-fine search method in terms of subsamples of image locations. In the first stage, a gross search is done and the similarity is evaluated only one time in each 27 pixels in the vertical and horizontal directions. Next, a fine search is done centred on all skin colour pixels in the neighbourhood of the best location found in the gross search. Once the position that maximises $g(\mathbf{x}, \overline{\mathbf{x}})$ is found, it is necessary to stipulate the orientation of the hand in the image plane. This is done by searching for the orientation θ of the template $\overline{\mathbf{x}}$ that maximises the similarities $h_{\overline{m},m'}$ between the rings that constitute the template and the located image descriptor.

To save computational time, the matching score $g(\mathbf{x}, \overline{\mathbf{x}})$ is evaluated before moving to a finer stage. If it falls below a threshold, it is considered that no pointing hand has been located in the image and the system waits for the next frame. The same happens after the finest search in order to decide whether to move to the tracking stage or not. Figure 5 shows that the detector functions under quite different and severe image noise.

4 Hand Tracking

The shape detector initialises three degrees of translational and rotational freedom that most affect image appearance. The other 3 DOFs are set to default values, and all are passed to an implementation of Harris' RAPiD tracker [2], which is able to refine the pose estimation. The idea is that the user tells the robot that (s)he is performing a pointing gesture by starting with the hand at a roughly standard distance from the camera. Next the user can adjust the depth of the pointing direction and this is identified by this tracker.

Since our aim is to track a single pointing gesture, a rigid model of the hand is enough. In order to reduce the computational cost, a simple planar model was used, so



(a) (b) (c)

Figure 4. Extracting and matching the shape descriptor: (a) outdoor view of the hand; (b) the shape detector locates the pointing gesturem, the direction is indicated by the arrow indicates; (c) the ring curves where the value 1 indicates skin area and -1 indicates background; blue dashed shows the template and solid red the current signal after best alignment.

self-occlusion handling is not necessary. This model comprises straight edges along which control points $\mathbf{X}^B$ are distributed in the model coordinate frame. Given the postulated pose, these are transformed to the camera frame $\mathbf{X}^C$ (Figure 6) and projected, via the known intrinsic calibration, into the image at $\mathbf{x}$.

4.1 Finger Edge Finder

As control points lie on edges, it is not possible to search for the actual correspondence $\mathbf{x}'$ and thence to update the pose by minimising a measure based on $|\mathbf{x}' - \mathbf{x}|$. Instead, as Harris explains [2], the distance minimised is that from the control point $\mathbf{x}$ to the image edge along the edge normal $\hat{\mathbf{n}}$, or, more efficiently, along the nearest cardinal direction $\hat{\mathbf{d}}$, as shown in Figure 7.

Since the images are binarised on skin colour, finding edges is trivial. But as the finger is narrow, some care has to be taken not merely to chose the edge closest to the control point. In Figure 7, for example, this would be a mismatch. The edge detector restricts the direction of the edge to be dependent on the searching direction. Our hand model is a polygon such that all the lines may lie in between hand and background pixels. Therefore, considering the clockwise direction, the search is performed from right to left. The first value change from 1 (skin) to 0 (background) is taken as the located edge. This also prevents the tracker fitting to background edges.

A novel implementation detail is that the size of path for edge searching $2L$ is set to a value that is proportional to the speed of the object projected in the image plane. Since projective geometry is used, the speed of the hand image is likely to be proportional to the proximity of the hand to the camera, so $L = K/{t_Z^{i-1}}_{BC}$, where ${t_Z^{i-1}}_{BC}$ is the distance between the camera and the hand in the previous frame of the video sequence. The constant K is set to $K = 5t^0_{ZBC}$, where t^0_{ZBC} is the default translation in depth that is used in the first iteration of the tracker after the detector is executed.

4.2 The RAPiD Tracker

The mathematics underlying the tracker is found in [2] and relies on the pose change being small enough such that the pose update, written as a product of the inter-frame time and the velocity screw

$$\delta\mathbf{s} = \delta\tau[v_x, v_y, v_z, \Omega_x, \Omega_y, \Omega_z]^\top$$

can be found from a linear system into which each control point i contributes a row

$$\begin{bmatrix} m_i \end{bmatrix} = \begin{bmatrix} \mathbf{a}_i \end{bmatrix} \delta\mathbf{s} \tag{5}$$

where both

$$m_i = \hat{\mathbf{n}}^\top(\mathbf{x}' - \mathbf{x}) \approx d\hat{\mathbf{n}}^\top\hat{\mathbf{d}},$$

and

$$\mathbf{a}_i = -\frac{1}{Z_i^C}\hat{\mathbf{n}}_i[\mathtt{I}_3 + \mathbf{x}_i[0\ 0\ 1]]\mathtt{G}_i$$

are determined from image measurement and current pose. The 3×6 matrix $\mathtt{G}$ depends on transformed depths, and is defined in [13]. The system is solved for $\delta\mathbf{s}$ using singular value decomposition. Because of the approximate nature of the linearization, it can be useful to iterate the solution within each image.

Figure 8 shows a skin colour segmented image overlapped by a projection of the five-line planar hand model showing the control points. Although the model used does not have a realistic appearance, our experiments have shown that modelling the finger as a triangle increases the motion

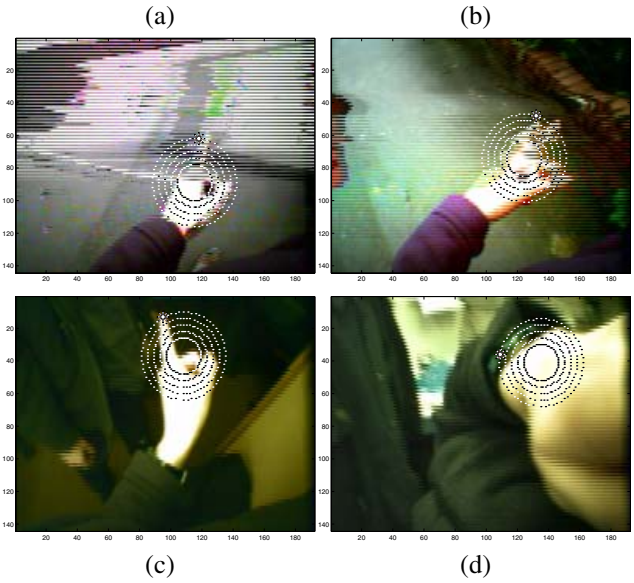


Figure 5. Challenging images: (a) outdoor noise image where hand is non-contiguous (finger striped), response value $g(\mathbf{x}, \overline{\mathbf{x}}) = 0.86$; (b) ghostly finger, $g(\mathbf{x}, \overline{\mathbf{x}}) = 0.85$; (c) indoor image with change in shape (sleeve retracted), $g(\mathbf{x}, \overline{\mathbf{x}}) = 0.81$; (d) Non gesturing hand, $g(\mathbf{x}, \overline{\mathbf{x}}) = 0.72$. The video noise in (a) and (b) is encountered at the limits of the wireless transmitter's range.

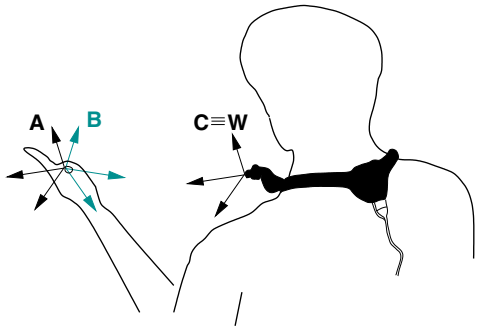


Figure 6. The hand and camera coordinate frames.

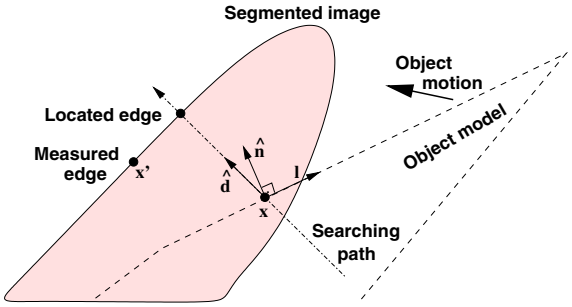


Figure 7. Searches are made from the projected control point x in the cardinal direction $\hat{\mathbf{d}}$ closest to the unit edge normal $\hat{\mathbf{n}}$.

constraints along the finger axis, also making it more robust to rotations in depth. This compensates the lack of edges on the wrist, which were not included because the user is not restricted to wear a long-sleeved shirt or a bracelet. The simplicity of this model speeds up projection calculations.

4.3 Monitoring Tracking

To monitor the tracking performance, the norm of the residual $||\mathbf{m}||$ before pose update could be used. But when an edge is not located, it is not included in the residual to avoid perturbation in the pose update computation, which this means that the value of $||\mathbf{m}||$ does not reflect the success of the tracker. We chose to used a cost function that depends on the actual distance between the located edges $\mathbf{r}$ and the projected lines $\mathbf{l}$ of the model after the pose update. Using homogeneous coordinates, each point can be modelled as a vector $\mathbf{x} = [x, y, 1]^\top$, and the lines are defined by $\mathbf{l} = \mathbf{x}_m{}' \times \mathbf{x}_n{}'$, which is equivalent to the following determinant:

$$\mathbf{l} = \begin{vmatrix} i & j & k \\ x'_m & y'_m & 1 \\ x'_n & y'_n & 1 \end{vmatrix}, \tag{6}$$

where $\mathbf{x}_m{}'$ and $\mathbf{x}_n{}'$ are two points that lie in $\mathbf{l}$. The distance $d_{p,q}$ between line $\mathbf{l}_p$ and point $\mathbf{r}_q$ can be written as:

$$d_{p,q} = \frac{\mathbf{r}_q^\top \mathbf{l}_p}{\sqrt{l_{x_p}^2 + l_{y_p}^2}} \tag{7}$$

Our cost function is defined by the sum of all the distances d between all the found edges and their respective lines:

$$\mathcal{C} = \frac{1}{\mathcal{W} D} \sum_{\forall p,q} d_{p,q}, \tag{8}$$

where D is the total number of control points in the whole model, and $\mathcal{W}$ is the worst case constant, defined by $\mathcal{W} = 2L$, which is the number of pixels in the path for searching edges. When no edge $\mathbf{r}_p$ is located in the searching path for a control point, $d_{p,q}$ is set to $\mathcal{W}$.

The cost function result is employed in order to determine if the tracker has lost the hand and the detector needs to be called. The function is also used to verify if the tracking results are good enough to be used to perform a camera movement toward the target. A second condition for that is

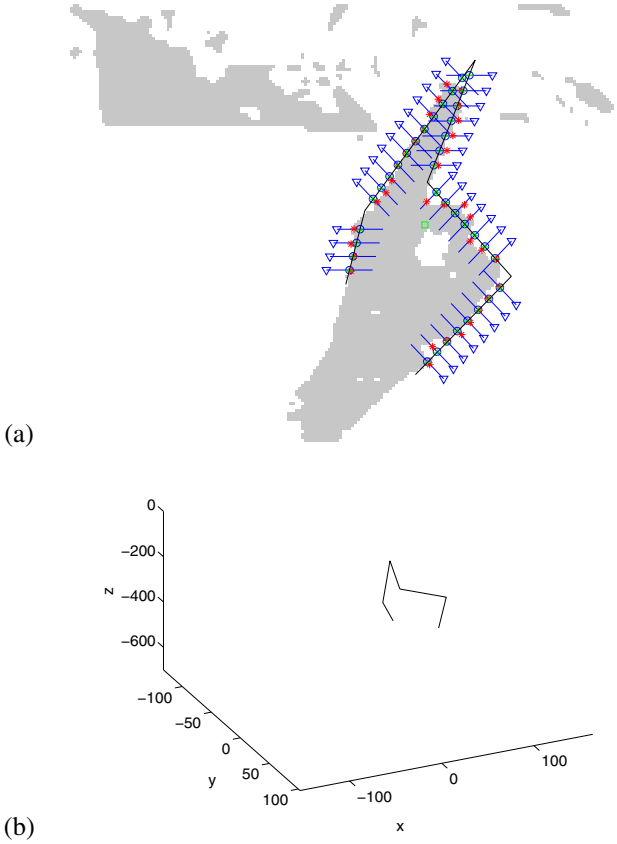


Figure 8. (a) Projection of our hand model (black line), search paths (segments with a triangle indicating the end of the search), control points (circles) and found edges ('*'). (b) Representation in the camera coordinate frame of the hand model projected in the image in (a). The units are in millimetres and the z axis is the camera axis.

the stability of the hand in the space. If the change of pose $||\delta \mathbf{s}||$ is below a given threshold for 1 second, the camera can be redirected to the target direction.

5 Results

The experiments described here were performed on a video sequence of 1104 frames grabbed from the wearable camera in an office environment with no illumination control and with a cluttered background. An approximate of the ground truth trajectory was generated from mouse clicks on three points of the hand. The Nelder-Mead nonlinear minimisation algorithm [9] was employed to recover the 3D pose of the hand from the mouse clicks.

The plots in figure 9 show the pose estimation results (thick curves) with time (in frames) in comparison with the ground true estimative (thin curves) for four degrees of freedom. When the cost function indicated a bad pose estimate, the hand detector was invoked. The circles illustrate the frames where this happened. Cost function results are shown in figure 10.

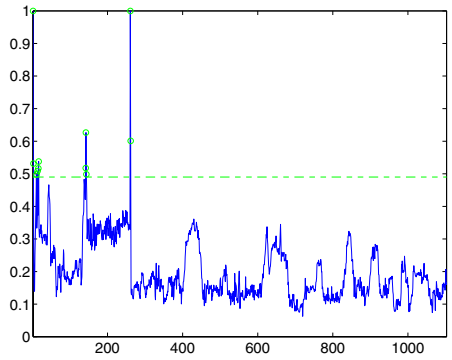


Figure 10. Cost function results for the experiment shown in Figure 9. The dashed line is the threshold used to indicate whether the tracker is lost, and the circles indicate when the hand detector was called.

These results show that the estimates of parameters parallel to the image plane (T_x, T_y and θ_z) are good match to the ground truth data, but the same is not observed for the depth parameters (e.g. T_z). However this does not necessarily imply that the pose estimated by the tracker lacked quality. In fact, the estimate of ground truth data was not reliable for depth parameters because we used only three mouse-clicked points in a single view without sub-pixel accuracy. It was difficult to choose more points to be clicked, as the hand texture is plain. A better estimate of the ground truth would be obtained if multiple views were available for the same sequence. The above can be verified in the video sequence that demonstrate the results, available from our web page (see title page).

The same video sequence was used to evaluate our application for redirecting the wearable camera by performing saccades. The results are plotted in figure 11, which, for clarity, shows only the estimated pose and the ground truth in the frames where the re-directing process was called. The wearable camera's saccade and location of object of interest is assumed to take 1s, after which the wearable camera moves back to detecting the hand. Figure 12 shows the cost function with time, indicating when the hand detector was invoked (circles) and when the redirecting process could be executed (asterisks).

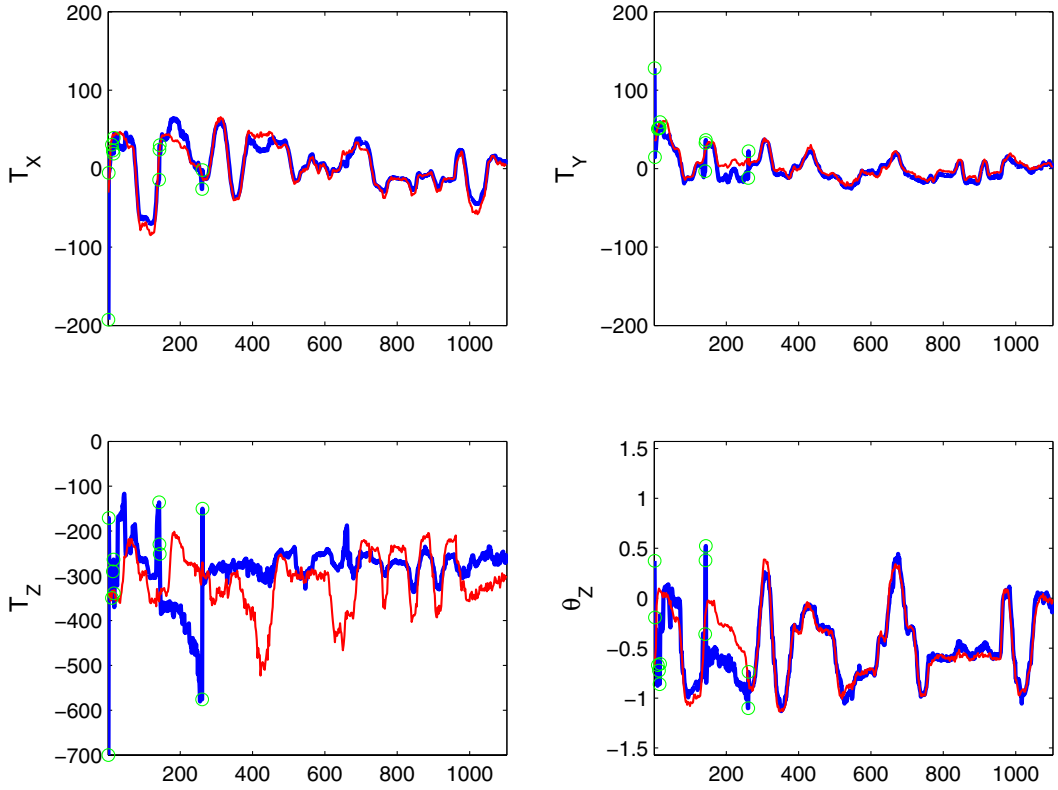


Figure 9. Results of the integrated system (thick curves) showing the detector calls (circles) and the ground truth estimate (thin curves). The space is measured in millimetres, angle in radians and the time in frames.

6 Conclusions

This paper presented a method for detecting and tracking a specific hand shape — pointing — with applications of estimating the focus of attention or controlling the gaze direction of a wearable active camera. This enhances user-robot interaction and enables the recognition of an important non-verbal communication gesture.

We combined a 2D shape detector and 3D tracker using a finite state machine. Criterion functions for both the detector and the tracker were used to automatically monitor their result in order to change the state in the finite state machine.

The detection method performed a coarse to fine search in the image using a simple shape descriptor that is invariant to rotation and a matching method to estimate orientation. It provided an initial estimate of the planar pose parameters. To refine the estimate and provide the depth parameters, we employed a 3D tracking method which uses control points on the edge of the hand silhouette.

Our experiments have shown that a simple rigid planar model of the hand lead to acceptable tracking results with low computational cost.

References

[1] L. da Fontoura Costa and R. M. Cesar-Jr. *Shape Analysis and Classification - Theory and Practice*. Image Processing. CRC Press, 2001.

[2] C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, pages 59–73, Cambridge, MA, USA, 1992. MIT Press.

[3] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, January 2002.

[4] T. Kurata, E. Okuma, M. Kourogi, and K. Sakaue. The hand mouse: GMM hand-color classification and mean shift tracking. In *Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time (in conjunction with ICCV)*, pages 119–124, Vancouiver, Canada, July 2001.

[5] A. M. Martinez and R. Benavente. The ar face database. Technical Report CVC 24, Purdue University, June 1998. Available at *http://rvl1.ecn.purdue.edu/∼aleix/ar.html*.

[6] W. Mayol, B. Tordoff, and D. Murray. Wearable visual robots. In *International Symposium on Wearable Computing*, Atlanta, GA, USA, 2000.

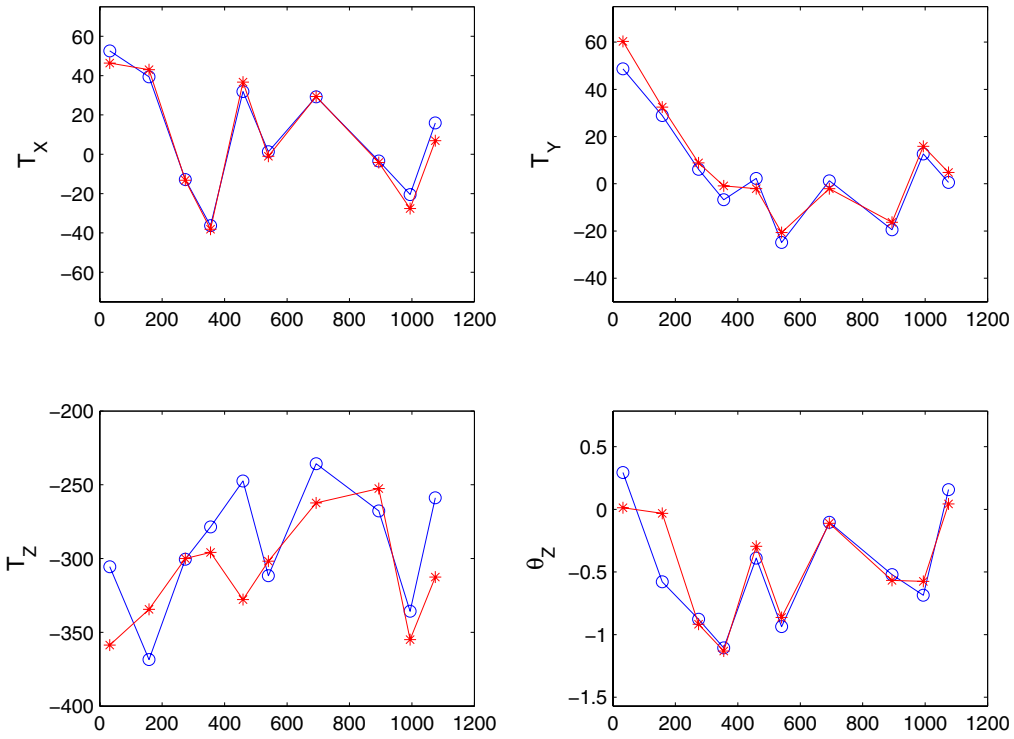


Figure 11. Pose estimations ('*') and estimated ground truth ('o') when the re-directing process was called.

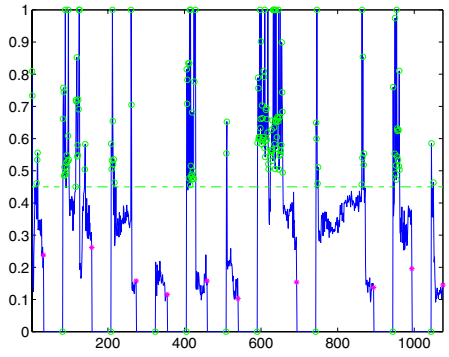


Figure 12. Cost function results for the experiment shown on Figure 11. Asterisks show when re-directing was called.

[7] W. W. Mayol-Cuevas. *Wearable Visual Robots*. PhD thesis, University of Oxford, Department of Engineering Science, Michaelmas 2004.

[8] W. W. Mayol-Cuevas, A. J. Davison, B. J. Tordoff, N. D. Molton, and D. W. Murray. Interaction between hand and wearable camera in 2d and 3d environments. In *Proc 15th British Machine Vision Conf, Kingston University, London*. British Machine Vision Association, September 2004.

[9] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965. Oxford University Press, *http://www3.oup.co.uk/computer_journal*.

[10] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:677–695, 1997.

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

[12] D. J. Sturman. *Whole-Hand Input*. PhD thesis, Media Arts and Science Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA, February 1992. *http://xenia.media.mit.edu/∼djs/thesis.ftp.html*.

[13] R. L. Thompson, I. D. Reid, L. A. Munoz, and D. W. Murray. Providing synthetic views for teleoperation using visual pose tracking in multiple cameras. *IEEE Transactions of Systems, Man and Cybernetics*, 31(1):43–54, 2001.

[14] B. J. Tordoff, W. W. Mayol, T. E. de Campos, and D. W. Murray. Head pose estimation for wearable robot control. In *Proc 13th British Machine Vision Conference*, Cardiff, Wales, September 2002. British Machine Vision Association.