AIDIA – Adaptive Interface for Display InterAction

Björn Stenger* Thomas Woodley† Tae-Kyun Kim†
Carlos Hernández* Roberto Cipolla†
* Computer Vision Group † Dept. of Engineering
Toshiba Research Europe University of Cambridge

Abstract

This paper presents a vision-based system for interaction with a display via hand pointing. An attention mechanism based on face and hand detection allows users in the camera's field of view to take control of the interface. Face recognition is used for identification and customisation. The system allows the user to control the screen pointer by tracking their fist. On-screen items can be selected using one of four activation mechanisms. Current sample applications include browsing image and video collections as well as viewing a gallery of 3D objects. In experiments we demonstrate the performance of the vision components in challenging conditions and compare it to that of other systems.

1 Introduction

This paper presents a vision-based interface using a single camera on top of a display, as shown in Fig.1. Such a system allows touch-free input at a distance and has several uses in practice: virtual remote control for a TV or for other home appliances, gaming, or browsing public information terminals in museums or window shops. Here we present a complete system which integrates (a) an attention mechanism for initiating the interaction, (b) face recognition for user identification and customisation (in terms of content and functionality) and (c) fist tracking for moving a pointer and recognition of hand gestures such as a 'thumb up' or a 'shake' gesture for item selection.

For face recognition we make use of the video data by matching image sets, which has been shown to be significantly more robust than single image matching [10]. Adaptation is a key element for recognition under changing conditions and improves the recognition rate by integrating new training data. The system therefore includes a scheme to update the face manifold representation online.

The hand tracking problem is challenging due to several factors, including motion blur, distraction from background objects, and appearance variation due to pose and lighting changes. This is illustrated in Fig.2, showing examples of image regions around the hand taken from the test sequences. In order to handle such variation the proposed hand tracker switches dynamically between different cues based on confidence estimates. In addition to tracking, automatic initialisation is required to find the hand at the beginning and after loss of track. This may occur regularly, for example every time the hand is outside the camera's view. The proposed system thus integrates an off-line trained detector to initialise and update the trackers to avoid drift.

(a) (b) (c)

Figure 1: **Gesture interface.** *(a) face detection is performed during the attention phase, interaction is initiated by hand detection, (b) set up of camera mounted on top of the screen and multiple users in the field of view, (c) a sample application for the inspection of 3D models.*

In the following section we give an overview of prior work on hand tracking in the context of this work. Section 2 explains the attention mechanism that allows a user to initiate the interaction. The face recognition component is described in Section 3 and the fist tracker in Section 4. Experiments in Section 5 demonstrate the performance of the face recognition and hand tracking components.

1.1 Previous work

A large number of vision-based gesture interfaces have been proposed, only some of which are concerned with our particular setting of having a single camera pointing towards a scene of possibly multiple people. In this paper we focus on single camera systems, although a stereo setup or time-of-flight sensors present valid alternatives. We give a brief overview of prior art while highlighting some of the limitations.

Freeman and Weissman [6] introduced a system for television remote control by hand motion where a hand template is tracked based on correlating local orientations. It uses a hand template for detection and tracking and includes background subtraction. The tracker works when the hand moves slowly, but edge features tend to be unstable when motion blur occurs. Bretzner et al. [4] used multi-scale blob detection of colour features in order to detect an open hand pose with possibly some of the fingers extended, corresponding to different input commands. A simple 2D shape model is used for tracking with a particle filter. The method requires a skin colour prior, which is obtained by manually labelling 30 frames. An interface based on tracking multiple skin coloured regions was proposed in [1]. Again, the skin colour model is obtained by manually labelling skin regions, but the colour model is adapted during tracking. We observed that trackers which use only colour features struggle in our setting, in particular if the hand moves in front of the face, if the user wears short sleeves, or if there are objects of similar colour in the background. An active camera system for hand tracking by finding regions of high motion and skin colour probability was proposed in [12]. The Viterbi algorithm is used to find a temporal path connecting local maxima of a likelihood function that combines these two cues. A spatial prior is used to associate blobs to hand and face. A restriction of the system is that it performs search over a single scale only, requiring the user to be at a fixed distance to the camera. Kölsch and Turk [11] presented a multi-cue tracker that combines colour and many short tracks of local features under 'flocking' constraints. The colour model is automatically initialised from hand detection. Although the method was shown

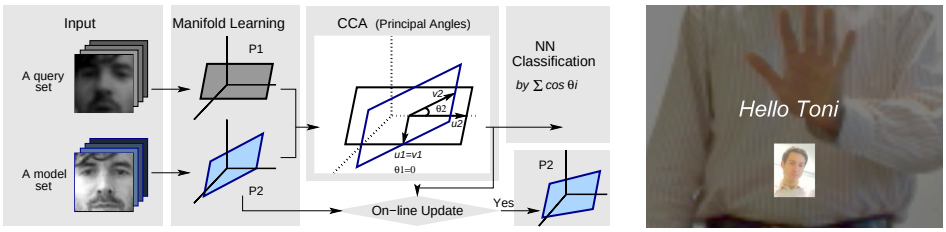
Figure 2: **Appearance variation of hand regions.** *Shown are cropped hand regions from test sequences. Motion blur, changing pose and other skin coloured objects make tracking challenging.*

for top-view tracking, it is general enough to work for frontal views. However, it struggles with rapid hand motion and skin coloured background objects. The system in [21] used a trained detector followed by optical flow tracking. Tracking based on optical flow alone has difficulties coping with rapid hand motion as well as moving background objects. Ike et al. [9] presented a real-time system for gesture control that detects three different hand poses independently in each frame. Due to the high computation requirement it was implemented on a multi-core processor. We compared with five of the above systems and present the results in Section 5.

To summarise, no complete system meets the requirements of robust tracking, cleanly handling initialisation and tracking failure, working for both slow and rapid motion, handling multiple scales, using a single CCD camera and being sufficiently fast to run on a standard PC.

2 Visual attention mechanism

One goal of this work is being able to set up the system in an arbitrary environment, such as the living room, or a public space, where multiple people may be within the camera's view. For some periods there may be no interaction at all until one person initiates the interaction in order to achieve a specific task. In AIDIA this works as follows: Initially the system performs face detection using a boosted detector [18]. Multiple detections are associated over time by minimising the sum of distances of detections between two frames with the Hungarian algorithm. Once a face is detected the user is prompted to show an open hand gesture within the area below their face, see Fig.1a. This also works for multiple users in the scene. The rectangular input regions below the face detections are ordered according to scale, giving easier access to users who are closer to the camera. The first detection of an open hand triggers the face recognition step: Detected face regions are stored during the attention phase and the image set of the person who activated the system is passed to the recognition component. At this point the user may register in the database or, if they have used the system before, they can choose to update their face model with the new data. Recognition prompts a personalised greeting message to be displayed (see Fig.3b) and the content can be customised according to the user's profile. Subsequently the hand tracker becomes active and allows the user to browse the content by selecting items from a menu that is overlaid on the screen. Note that the scale of the face detection is used to define the size of the interaction area while the centre of the interaction area is set to the location of the open hand detection. This means that the range of motion remains constant for different distances to the camera.



(a) (b)

Figure 3: **Face recognition by matching image sets.** *(a) The similarity between manifolds is computed as the sum of principal angles and is used for NN classification. Once a query set has been classified, it can be included in the model by on-line updating the existing manifold. (b) Screenshot after recognition.*

3 Face recognition by matching image sets

This section describes the face recognition component of the system. While the number of users may be small in our system, the appearance variation may be large due to pose and illumination changes. Our recognition component uses image sets for matching, which are captured during the attention phase. The image set can capture appearance changes and provide more evidence on face identity than a single image alone. No temporal coherence is used as this may actually decrease recognition performance [25].

Generally, there are three types of approaches to image set (or vector set) matching: aggregation of multiple nearest neighbour vector-matches [5], probability-density based methods [22], and manifold-based methods [24]. Taking the latter approach, we match manifolds using canonical correlations. Canonical Correlation Analysis (CCA) (also called canonical or principal angles) [24] compares manifolds by measuring the angles between them (see Fig.3a). Canonical correlations, which are cosines of principal angles between any two d-dimensional linear manifolds $\mathscr{L}_1$ and $\mathscr{L}_2$, are defined as

$$\cos\theta_i = \max_{\mathbf{u}_i \in \mathscr{L}_1} \max_{\mathbf{v}_i \in \mathscr{L}_2} \mathbf{u}_i^T \mathbf{v}_i, \qquad i = 1, ..., d, \tag{1}$$

subject to $\mathbf{u}_i^T \mathbf{u}_i = \mathbf{v}_i^T \mathbf{v}_i = 1$, $\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = 0$, $i \neq j$. If $\mathbf{P}_1, \mathbf{P}_2$ denote the basis matrices of the two manifolds (see Section 3.1), canonical correlations are conveniently obtained as singular values of $\mathbf{P}_1^T \mathbf{P}_2$, only taking $O(d^3)$. CCA has the following nice properties: (a) It allows interpolation of the vectors in each set when finding maximum correlations, thus being more robust to data variation and noise, and (b) the low-dimensional manifold representation allows matching that is both time and memory efficient.

The manifold angle is a natural extension of prior manifold-based face recognition methods. When a single face image is given as an input, there is a standard way to classify it by manifolds: by measuring the distance of the face vector to each manifold and picking the closest one. When classifying a manifold instead of a single vector, angles between manifolds become a reasonable distance measurement. Experimental comparison with other vector-set classification methods advocates the canonical correlation method [10]. Since Hotelling [8], CCA has received increasing attention and recently Bach and Jordan [2] have proposed a probabilistic interpretation, and Wolf and Shashua [24] proposed a kernel version. Kim et al. [10] proposed discriminative manifold learning for CCA, resulting in better performance than other CCA-based methods.

3.1 On-line manifold learning

While most existing recognition systems rely on a single off-line training phase, it is desirable to include new data when it becomes available. Therefore the face recognition component includes a method for user-interactive updating of the manifolds.

We will first explain how to learn the discriminative manifold for CCA, i.e. the basis matrix $\mathbf{P}_i$ in Eqn. 1. Recalling that the canonical vectors represent the directions of most similar data variations the of two sets, it is ideal to represent each set by the manifold that maximally represents the respective class data while minimising the variance of other class data:

$$\max_{arg\mathbf{P}_i} \frac{\mathbf{P}_i^T \mathbf{S}_i \mathbf{P}_i}{\mathbf{P}_i^T \mathbf{S}_T \mathbf{P}_i}, \quad i = 1,...,C \tag{2}$$

where $\mathbf{S}_i, \mathbf{S}_T$ denote the covariance matrices of the *i*-th class and the total data. The basis matrix of *i*-th class model $\mathbf{P}_i$, is obtained as the generalised eigen-solution.

It is too inefficient in terms of time and memory to run the batch-computation of the manifold whenever new data is added. Instead, the two covariance matrices are first eigen-decomposed as $\mathbf{S}_i = \mathbf{Q}_i \Lambda_i \mathbf{Q}_i^T, \mathbf{S}_T = \mathbf{Q}_T \Lambda_T \mathbf{Q}_T^T$, where $\mathbf{Q}, \Lambda$ are the eigenvector and eigenvalue matrix, respectively, corresponding to the first few eigenvectors. The manifold is then updated by separately updating the eigen-components and then computing the manifold only by the new eigen-models. Owing to its linearity, the method of Hall et al. [7] can be applied: $\mathbf{Q}_i, \Lambda_i, \mathbf{Q}_T, \Lambda_T$ are updated and $\mathbf{P}_i$ is computed by SVD of $(\sqrt{\Lambda_T}\mathbf{Q}_i)^{-1}\mathbf{Q}_i\sqrt{\Lambda_i}$.

4 Hand tracking

For initialisation, a detector for a fist pose is trained off-line using the method of Mita et al. [18]. It is applied within a region of interest I obtained during the attention phase, which constrains the valid region of the hand tracker. Due to the distinctive appearance of the frontal fist region a single image patch is tracked using normalised cross-correlation (NCC) [14]. The patch is selected as a smaller subregion of the hand in order to discount background regions. NCC tracking is accurate and works for slow hand motion within a limited range of motion. However, It can only deal with minor appearance variation, and rapid motion leading to strong motion blur is also problematic. The idea therefore is to start with NCC tracking and in case of failure apply a second tracker as a fall-back strategy. The second tracker uses different feature spaces, namely colour and motion (CM tracker). Colour models for the foreground region and the surrounding background region are obtained from the detector and are represented by 32-bin RGB histograms. The motion model is represented as histograms of the absolute differences between consecutive frames. The CM tracker detects scale space maxima of a likelihood function that uses both cues. First a colour likelihood map is computed for each location in the image region of interest $p(x|\text{col}), x \in I$. Similarly a motion likelihood map $p(x|\text{mot}), x \in I$ is obtained. The likelihood function combines three terms as a sum and is based on [12], however, here the functions are smoothed by Gaussians with a variance depending on the size of the previously detected hand. The likelihood function is defined as

$$p(\text{hand}|x) \propto w_c\, p(x|\text{col}) + w_m\, p(x|\text{mot}) + (1 - w_c - w_m)\, p(x|\text{col})\, p(x|\text{mot}), \tag{3}$$

where w_c and w_m are weights that are determined through experiments on a validation set (in our case $w_c = w_m = 0.1$). Scale space maxima of this function are found with a

'box filter' [23], which is an efficient approximation to the Laplacian. The three terms in Eqn. 3 allow tracking in different scenarios: e.g. if there is no other skin coloured object in the background, the colour likelihood is discriminative enough. Rapid motion leads to peaks in the motion likelihood function. The third term gives high values to objects that are moving and are skin coloured. The terms could be combined in a more principled way, but in practice this formulation turns out to be quite efficient. Since the CM tracker essentially models the shape as a simple blob, it can handle large variations in pose. Both trackers return a confidence value, which is the NCC correlation score and the filter output, respectively.

The complete tracking algorithm proceeds as follows: After detection the NCC tracker is active. If it returns a confidence value below a threshold θ_{NCC} tracking continues with the CM tracker. At every kth frame, the fist detector is applied in the local neighbourhood and, if successful, NCC tracking resumes with a new template. Thus, trackers (and corresponding features) are switched online. Tracking is stopped when the confidence value of the CM tracker is below a threshold θ_{CM}. A Kalman filter is used to combine the estimates with a constant velocity dynamic model. The approach taken in this paper is to efficiently but densely sample the likelihood values around the estimated location.

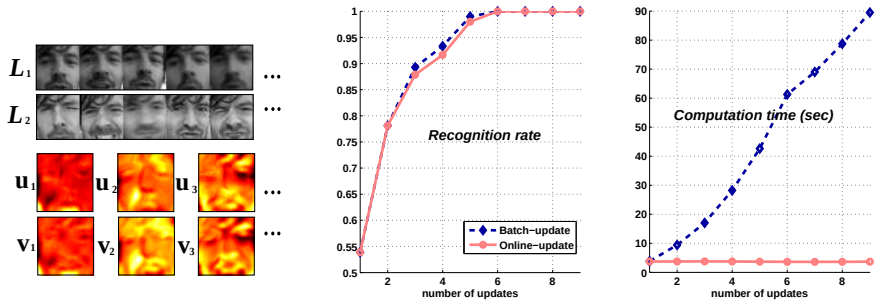
Related tracking methods can be found in the extensive literature on multi-cue tracking [3, 13, 20]. The benefit of these approaches is increased robustness when different cues have different failure modes and therefore complement each other. The most common idea is to run several trackers in parallel and subsequently combine their output, by either selecting between them [3] or by probabilistically merging them [13, 20]. In contrast, the proposed tracker switches between trackers (and corresponding features) entirely, therefore not requiring trackers to run simultaneously. We further note that our tracker is tightly integrated with a detector. Indeed, local detection together with a strategy to link up missing detections through time is a viable solution such as in the system of Ike et al. [9]. Even though localisation is not as precise as with NCC and less pose variation is handled compared to the CM tracker, it allows handling multiple scales and updating the tracking template. Note that the idea of running a tracker and a detector in tandem has previously been used to build tracking systems that work over arbitrary time periods, e.g. the system in [11]. Similarly, detector output has been integrated directly in the observation model [15].

4.1 Selection mechanisms

In order to activate a screen icon a selection mechanism equivalent to a mouse click needs to be defined. Solutions that have previously been proposed include changing hand pose, finger or thumb extension and simply hovering over an icon for a short time period [4, 6, 9, 11, 16, 21]. We have implemented these by training separate detectors, see Fig.4, (a) an open hand detector, (b) a thumb up detector, and (c) hovering over an icon for a short period of time (0.5 seconds). Additionally, we propose the following method: (d) detecting a quick left-right shake gesture. The shake gesture is

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

Figure 4: ***Different gestures for selection:*** *(a) open hand pose, (b) thumb up pose, (c) hovering for a short time period and (d) a shake gesture.*



(a) (b)

Figure 5: **Face recognition experiments.** *(a) Example input sets and canonical vectors* $\mathbf{u}, \mathbf{v}$ *computed. (b) (left) Accuracy improvement of the on-line and batch-method for the number of updates. (right) Computational time of the two methods.*

detected by recording the hand motion over a sliding window of 20 frames and classifying this vector. In experiments LDA and k-nearest neighbour classifiers were tested, but the most reliable results were obtained by computing the distance to the closest positive training example (among a small set of 75 examples) and threshold this value. Only one of the four selection mechanisms is used at any time, according to the user's preference.

5 Results

This section presents quantitative results on the face recognition algorithm as well as the hand tracking algorithm.

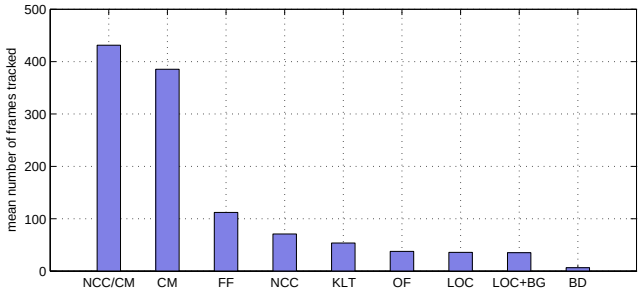
5.1 Face recognition experiments

We have evaluated the face recognition performance using a data set containing 5 people (10 sequences per person, 50 frames per sequence). The 10 sets were collected at different times, days and places and leading to appearance variation. The input dimension was set to 40×40 and the manifold dimension to 10.

Fig.5a shows example inputs and the canonical vectors computed by CCA. The canonical vectors in each pair $\mathbf{u}_i, \mathbf{v}_i$ are visually similar despite the large appearance changes across the two sets. As shown in Fig.5b, the method achieved perfect recognition results after updating with 6 image sets. The on-line method requires significantly lower computation time than the batch-solution when increasing the amount of training data. In the experiment one set per person was added to the model at each stage and all remaining data was used as query during each update. 5-fold cross validation was performed by random data partitioning.

5.2 Hand tracking experiments

The robustness of different hand tracking algorithms was evaluated on a set of 10 labelled sequences of 500 frames each (size 320×240, recorded at 30fps), measured as the mean



Algorithm	NCC/CM	CM	FF	NCC	KLT	OF	LOC	LOC + BG	BD
Mean frames tracked	431.3	385.4	112.0	70.9	53.8	37.8	35.9	35.2	6.5

Figure 6: **Hand tracker evaluation.** *Comparative results showing the mean number of consecutively tracked frames over 10 sequences of 500 frames. The NCC/CM tracker is the most robust.*

number of successfully tracked frames. After loss of track (defined by a scale-normalised distance being above a threshold) trackers are re-initialised at the next detection within the sequence. This allows a realistic assessment of the performance over the complete data set. To reduce the bias introduced by the finite number of frames (a failure close to the end may lead to a very short track) the last measurement before the end of the sequence is discarded if at least one tracking failures has occurred previously. The trackers that were compared against have been used in other hand tracking systems and include: local orientation correlation (LOC) [6], flocks of features tracking (FF) [11], optical flow tracking using templates on a regular grid (OF) and local feature tracking, KLT-tracker (KLT) [21], and boosted detection (BD) [9, 11, 17, 19, 21]. The performance of the individual components, the CM and NCC tracker, was also measured. The results are shown in Fig.6. The proposed NCC/CM tracker performs best and loses track in only two of the ten sequences. This is due to the CM component locking onto other coloured objects, in one case the user's arm, in the other case the moving hand of another person. In both cases the CM tracker's confidence value drops below the confidence threshold after a few frames and the tracker re-initialises by global detection. The CM tracker comes second in terms of robustness, however, it is much less precise during slow hand motion. The FF tracker can handle slow motion, but struggles with strong motion blur. It can also be distracted by other skin coloured regions with salient features such as the face. The regular block-based optical flow algorithm showed to be more robust than the KLT tracker, but both had difficulties handling rapid hand motion. Somewhat surprisingly the NCC tracker is more robust than the LOC tracker. A background estimation step used in [6] does not change the performance much (9 different updating weights were tested), which is likely due to the fact that the background appearance changes occasionally in the test sequences. The performance of the boosted detector is lowest in terms of our definition of robustness as consecutively tracked frames. The average number of detections on the data set is 242, but it varies significantly across the sequences. On some sequences there are very few detections due to larger pose changes.

Fig.7 shows some typical results on one of the test sequences, comparing the individual trackers as well as the frame-by-frame detector output. The NCC tracker loses track during rapid motion while the CM tracker is robust, but not always accurate (see the two rightmost frames). The frame-by-frame detector does not fire in several frames. The best

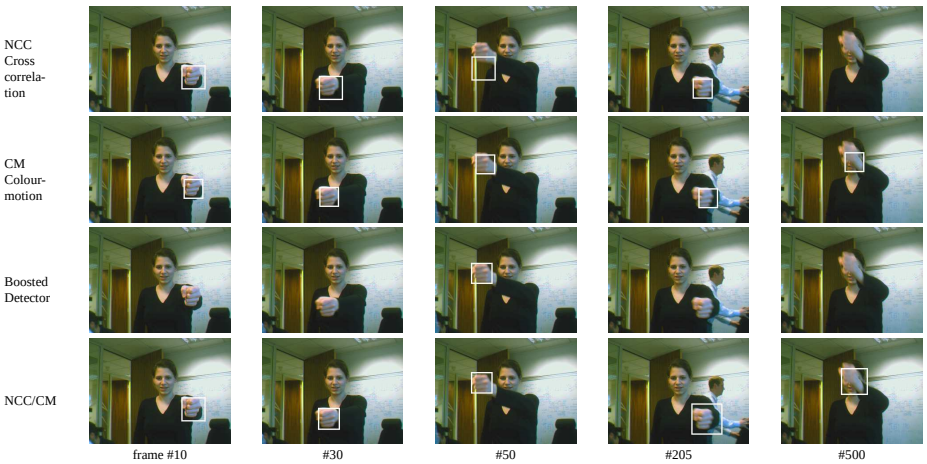


Figure 7: **Comparison of individual trackers with combined NCC/CM tracker.** *This figure shows snapshots of a sequence and results of the NCC tracker, the CM tracker, frame-by-frame detection and the proposed NCC/CM tracker.*

results are obtained with the combined NCC/CM tracker. The switching behaviour of the NCC/CM tracker is illustrated in Fig.8. During this sequence the light is turned off and on. Switches between components allows the tracker it to handle track successfully by updating its object representation.

6 Conclusions

We have presented a gesture interface by tracking a pointing fist with a single camera facing the user. The system includes an attention mechanism that allows one user at a time to be in control. Face recognition is employed for customising the interface. To increase the recognition performance under changing conditions the face model can be updated using efficient online learning. For fist tracking, we proposed a multi-cue method that switches trackers over time and is updated continually by an off-line trained detector. In experiments on ten hand pointing sequences our method outperformed other algorithms proposed for hand tracking such as local orientation correlation tracking, flocks-of-features tracking and optical flow tracking. So far the system has been tried by approximately 100 people within public exhibition settings. The main failure modes were found to be false fist detections, leading to incorrect adaptation of the colour model, as well as the CM tracker's reliance on colour and motion cues alone. Future work will address improving feature selection as well as the performance of the fist detector in order to handle larger appearance variation.

References

[1] A. A. Argyros and M. I. A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *Proc. ECCV*, pages 368–379, May 2004.

[2] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

[3] V. Badrinarayanan, P. Pérez, F. Le Clerc, and L. Oisel. Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues. In *Proc. ICCV*, Rio de Janeiro, Brazil, October 2007.

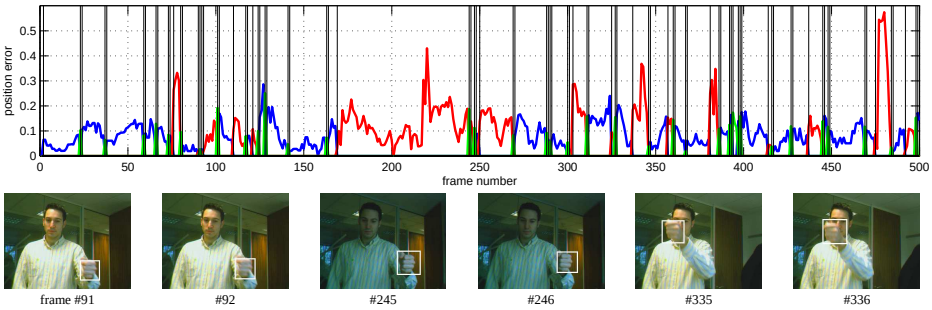


Figure 8: **Switching trackers over time.** *This figure shows the tracker's switching behaviour, colours in the plot indicate the component at each frame (blue=NCC, red=CM, green=detector). During this sequence the light was turned off and on. Example frames where transitions occur are shown below (first and third pair from NCC to CM due to motion blur, middle pair from CM to NCC via local detection).*

[4] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Proc. Face and Gesture*, pages 423–428, Washington, DC, 2002.

[5] M. R. Everingham, J. Sivic, and A. Zisserman. Hello! My name is... Buffy – Automatic naming of characters in TV video. In *Proc. BMVC*, pages 889–908, 2006.

[6] W. T. Freeman and C. D. Weissman. Television control by hand gestures. In *Intl. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, June 1995.

[7] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *Trans. PAMI*, 22(9):1042–1049, 2000.

[8] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(34):321–372, 1936.

[9] T. Ike, N. Kishikawa, and B. Stenger. A real-time hand gesture interface implemented on a multi-core processor. In *Proc. Machine Vision Applications*, pages 9–12, Tokyo, Japan, May 2007.

[10] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *Trans. PAMI*, 29(6):1005–1018, 2007.

[11] M. Kölsch and M. Turk. Fast 2D hand tracking with flocks of features and multi-cue integration. In *Workshop on Real-Time Vision for HCI*, Washington, DC, July 2004.

[12] N. Krahnstoever, E. Schapira, S. Kettebekov, and R. Sharma. Multimodal human-computer interaction for crisis management systems. In *Proc. WACV*, pages 203–207, Orlando, FL, December 2002.

[13] I. Leichter, M. Lindenbaum, and E. Rivlin. A generalized framework for combining visual trackers – the black boxes approach. *Int. Journal of Computer Vision*, 67(2):91–110, 2006.

[14] J. P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123, 1995.

[15] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *Proc. CVPR*, Minneapolis, MN, June 2007.

[16] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. ECCV*, volume 2, pages 3–19, Dublin, Ireland, June 2000.

[17] A. Micilotta, E. Ong, and R. Bowden. Real-time upper body detection and 3D pose estimation in monoscopic images. In *Proc. ECCV*, volume 3, pages 139–150, Graz, Austria, May 2006.

[18] T. Mita, T. Kaneko, B. Stenger, and O. Hori. Discriminative feature co-occurrence selection for object detection. *Trans. PAMI*, 30(7):1257–1269, July 2008.

[19] E.-J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Intl. Conf. Autom. Face and Gesture Recognition*, pages 889–894, Seoul, Korea, May 2004.

[20] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, March 2004.

[21] P. Robertson, R. Laddaga, and M. Van Kleek. Virtual mouse vision based interface. In *Intl. Conf. on Intelligent User Interfaces*, pages 177–183, Funchal, Portugal, January 2004.

[22] G. Shakhnarovich, J. W. Fisher, and T. Darrel. Face recognition from long-term observations. In *Proc. ECCV*, volume 3, pages 851–868, 2002.

[23] B. Stenger. Template-based hand pose recognition using multiple cues. In *Proc. ACCV*, pages 551–560, Hyderabad, India, January 2006.

[24] L. Wolf and A. Shashua. Kernel principal angles for classification machines with applications to image sequence interpretation. In *Proc. CVPR*, pages 635–640, 2003.

[25] S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91(1-2):214–245, 2003.