ROBUST TWO HAND TRACKER USING PREDICTIVE EIGENTRACKING

K. A. Barhate[1], *K. S. Patwardhan*[1], *S. Dutta Roy*[1], *S. Chaudhuri*[1], *S. Chaudhury*[2]

[1]Dept of Electrical Engineering
IIT Bombay, Mumbai 400076

[2]Dept of Electrical Engineering
IIT Delhi, New Delhi 110016

ABSTRACT

This paper presents a robust shape-based on-line tracker for simultaneously tracking the motion of both hands, that is robust to cases of background clutter, other moving objects, occlusions of one hand by the other, and a wide range of illumination variations. The tracker is based on an on-line predictive EigenTracking framework. This framework allows efficient tracking of articulate objects, which change in appearance across views. We show results of successful tracking across a wide variety of possible hand motions, and illumination conditions.

1. INTRODUCTION

A gesture interface using hand postures attempts to make the communication between the user and the computer more natural and intuitive. Use of two hands in such a communication process is an obvious choice. Due to the presence of two hands mutual occlusions are possible and a method to handle these must be devised. Utsumi and Ohya [1] propose a method to track the 3-D position, posture and shape of human hands from multiple view points, using a 'best view point' selection mechanism. In most cases, having multiple synchronized cameras (even if they are uncalibrated) is often not feasible.

Mammen *et al.* [2] estimate the occluded observation elements in terms of non-occluded ones and their predicted values. The authors however do not consider all possible cases of inter-hand occlusions, or identify the respective hands before and after it. Peterfreund [3] presents a Kalman filter-based active contour model (snake) to track non-rigid objects such as hands. The work uses an optical flow-based detection method to deal with occlusions and image clutter. The method rejects measurements that are inconsistent with previous estimates of image motion. This may not be true for all cases of mutual occlusions.

Extensive research work has been carried out in the area of tracking people during occlusion that could be considered analogous to the problem of two hand tracking. In their work on probabilistic framework for segmenting people under occlusion, Elgammal and Davis [4] demonstrate use of the segmentation result obtained prior to occlusion, to conduct occlusion reasoning for recovering relative depth information. Additionally, they use this depth information in the same segmentation framework. McKenna *et al.* [5] define visibility index as the ratio between the number of pixels visible of each person during occlusion to the expected number of pixels for that person when isolated. They use the visibility index to deduce which person is in front during occlusion. Haritaoglu *et al.* [6] track people during occlusion by keeping track of their heads, based on the silhouettes of the foreground regions corresponding to the group. The system fails if the heads of the people involved are not visible at some point during occlusion. Weber *et al.* [7] use a ground plane constraint to reason about occlusion between cars. This implies however, that the tracker would fail if the ground plane is not visible either due to clutter or occlusion by other objects, or because the reference points are out of view. Further, we note that the above systems make domain-specific assumptions about features of objects being tracked (people, cars) which may not hold for the case of two hands.

Sherrah and Gong [8] use a Bayesian network to track multiple interacting body parts like faces and hands, during occlusion. Shamaie and Sutherland [9] approach the occlusion problem in a two hand tracker by modeling the spatial synchronization in bimanual movements by the position and temporal synchronization using the velocity and acceleration of each hand.

Most of the above systems would fail in case the moving objects change their appearance substantially. An EigenTracker [10] has the ability to track objects simultaneously undergoing image motions and changes in view. One of the main lacunae of the EigenTracker is the absence of a predictive framework. The predictive EigenTracker of Gupta *et al.* [11] removes this restriction, allowing for faster and more reliable tracking. Instead of a compute-intensive (albeit offline) learning phase for each object to be tracked, the authors use efficient eigenspace updates to track any unknown object, and learn its appearance on the fly.

In this paper, we have used two such predictive EigenTrackers, one for each hand. We utilize the ability of an EigenTracker to track hands based on appearance, to identify the left and right hands after occlusion. Without an EigenTracker, such identification is very difficult for unadorned hands, Our algorithm can handle all possible cases of occlusions, just as in [9]. We achieve robustness to a wide range of illumination variations using a neural network-based colour constancy algorithm.

2. TWO HAND TRACKER

We use motion and skin colour cues [11] to infer the position of hands in a given frame. Figure 1 gives an overview of our two-hand predictive EigenTracker. Our system automatically segments out the regions of interest *i.e.*, the two hands (Details in Section 2.2). We model the motion of each hand by a six element affine vector. This model takes into account the effects of rotation, translation, scaling and shear - commonly observed changes in hand shapes. Six affine parameters imply a parallelogram bounding window, which offers a tighter fit to the object being tracked, than a rectangular window. One can use the coordinates of three image points as elements of the state vector. Alternatively, the six affine parameters themselves can serve the purpose. *i.e.,* $\mathbf{X} = [a_0, a_1, a_2, a_3, a_4, a_5]^T$. The 2-D affine transformation is given as

$$\mathbf{f}(\mathbf{p}, \mathbf{X}) = \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \mathbf{p} \qquad (1)$$

where **p** is a position vector of a point in two dimensions..

TWO HAND PREDICTIVE EIGENTRACKER

```
A. Delineate moving objects of
     interest i.e., the two hands
B. REPEAT FOR ALL frames:
1.   Obtain image MEASUREMENT
     optimizing affine parameters $\mathbf{a}$
     & reconstruction coefficients $\mathbf{c}$
2.   ESTIMATE new affine parameters
     for both hands using output of
     step 1
3.   FOR EACH hand:
     IF reconstruction error $\in (T_1, T_2]$
     THEN update eigenspace
4.   IF reconstruction error for ANY
     hand very large THEN construct
     eigenspace afresh
C. Once occlusion begins:
1.   Stop Eigenspace update for both
     hands
2.   Determine which edges of the
     two hands are observable
3.   Derive the unobservable edges
     from the observable ones
4.   Update the translation params.
     of the affine vector
D. When occlusion is declared over:
1.   If ANY recons.  error v.  large
     THEN swap the bounding windows
2.   Construct Eigenspace afresh
```

Figure 1: Predictive EigenTracking Algorithm for two hands: An Overview.

A commonly used state dynamics model is the second order Auto Regressive (AR) process (t denotes time): $\mathbf{X}_t = \mathbf{A}_2\mathbf{X}_{t-2} + \mathbf{A}_1\mathbf{X}_{t-1} + \mathbf{W}_t$ (Step B.2 in Figure 1). The particular form of the model will depend on the application - constant velocity, random walk model, etc. We use a CONDENSATION-based framework [12] for propagation of state densities across frames. We model the measurement as: $\mathbf{Z}_t = \mathbf{B}\mathbf{X}_t + \mathbf{F}_t$ where $\mathbf{X_t}$ is the state vector at time t and $\mathbf{Z_t}$ is the observation vector (Step B.1 in Figure 1). $\mathbf{A}_2, \mathbf{A}_1, \mathbf{B}$ are coefficient matrices and $\mathbf{W}_t$ and $\mathbf{F}_t$ are assumed to be zero mean, white Gaussian noise vectors. We estimate the required parameters for a particular form

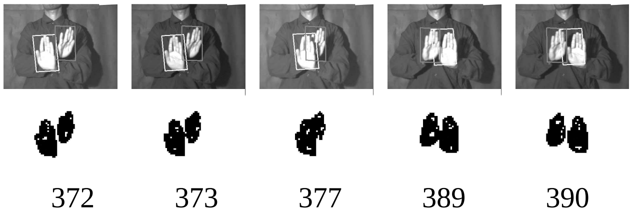


Figure 2: Detection of occlusion start and end. Images in upper row are the input images while the images in the lower row shows the segmented hands based on skin colour.

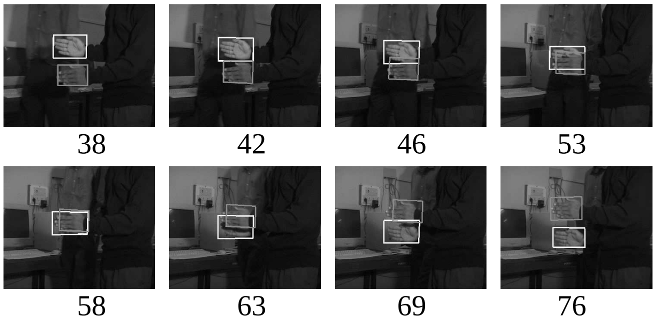


Figure 3: Occlusion begins at frame 42 and ends at frame 69. Note that the bounding window is a parallelogram. All videos corresponding to this paper: `http://www.ee.iitb.ac.in/∼sumantra/ncc04`

of the above models (depending on its suitability for an application), from a large number of representative observations.

In any sequence, hands often undergo considerable changes in appearance across frames. *Thus, one needs to learn and update the relevant eigenspaces, on the fly.* As in [11], we use an efficient SVD update algorithm of Chandrasekaran *et al.* [13]. Depending on the subsequent reconstruction error, the system takes a decision on updating the eigenspace, if required (Steps B.3 and B.4 in Figure 1).

In every frame the tracker checks for overlap of the two skin coloured blobs corresponding to two hands. Figure 2 shows an example of *occlusion*: the overlapping beginning at frame 373, till frame 389. We cannot update the appearance model of hands during occlusion. The following section describes our occlusion handling strategy.

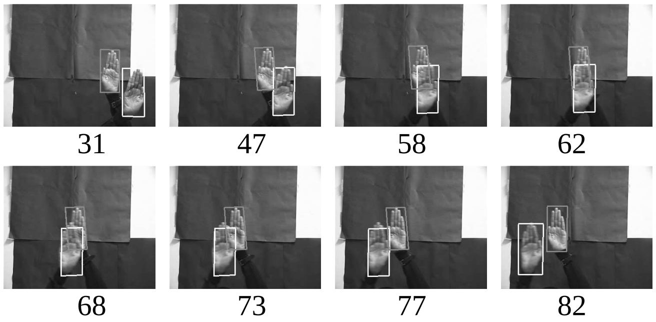


Figure 4: Both hands moving in the same direction but with different velocities. Occlusion begins at frame 47 and ends at frame 82.

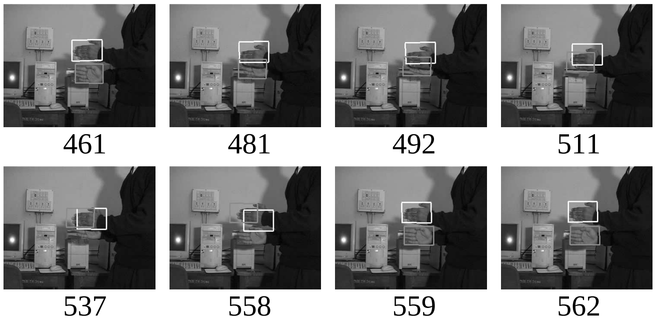
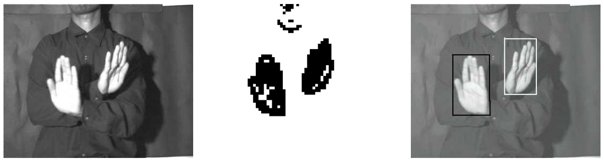


Figure 5: Hands go back to their original position after occlusion. Occlusion begins at frame 481 and ends at frame 559.

2.1. Occlusion Handling

Two EigenTrackers can not be used as such, to track overlapping objects. For the occlusion phase, the system uses a heuristic of not taking any measurements. However, this may lead to inaccurate tracking during this phase. For cases when the hands are not too tilted, we can take measurements from the bounding extents of the detected skin blobs. *This makes measurements possible even during the occlusion phase, thus making the tracker even more accurate.* If a pair of opposite boundaries is visible, we update the corresponding difference variable (height or width). If only one boundary is visible, we use the corresponding difference variable to estimate position of the other. If both are unobservable, we use the second order AR process to estimate their positions. Once all the boundaries are estimated in a frame, we update the translation parameters of the affine vector (we leave the other affine parameters unchanged). Step C in Figure 1 summarizes our occlusion handling strategy. Figure 3 shows re-



(a) (b) (c)

Figure 6: Applying skin colour and motion cues to (a) gives (b). Selecting the two largest area blobs in (b) segments out the two hands in (c).

sults of successful tracking of hands moving in opposite directions during occlusion.

During the time the hands overlap, their motion models may or may not change. A *Collision* occurs if the hands change their direction of motion during occlusion [9]. Our tracker successfully tracks hands during not just ordinary cases of occlusion, it works for collisions as well. After a collision hands may get wrongly identified, because of a change in the underlying motion model. We use the hand appearance models developed prior to the collision, to identify the left and right hands, after collision. Figure 5 shows results of successful tracking when the hands approach each other from opposite directions, and change the direction of motion during occlusion, to return to their starting positions.

2.2. Automatic Tracker Initialization

Initializing trackers is a challenging problem because of clutter, other moving objects, and the possibility of misclassifying the region of interest. Our tracker performs fully automatic initialization of both hands. The initialization method combines skin-colour and motion cues. Further, since the initial hand shape is not predefined, accurate initialization of the tracker helps us to create an appearance model of the observed hand shape. Among the several moving skin coloured blobs in a frame, we select the two largest ones as the hands (Figure 6).

2.3. Use of Color Constancy for Robust Tracking

If a gesture sequence is performed in a poorly illuminated environment (as in the upper part of Figure 7), the skin colour detection algorithm fails to identify skin coloured regions. To make our tracker robust to different illumination conditions, we apply a colour correction algorithm [14] to the input frames before the tracker processes them. This algorithm aims at estimating the transformation of an image taken under poor illumination conditions, to canonical illumination conditions. Our system considers canonical conditions as those used in the training phase for skin colour detection [15]. We use a neural network implementing the back-propagation learning rule to perform the transformation. We train it using a skin colour palette under unknown illumination, and a similar palette under known illumination conditions. We use such a transformation for the frames in the upper part of Figure 7. The lower sequence in Figure 7 shows results of successful tracking for the corresponding transformed frames.

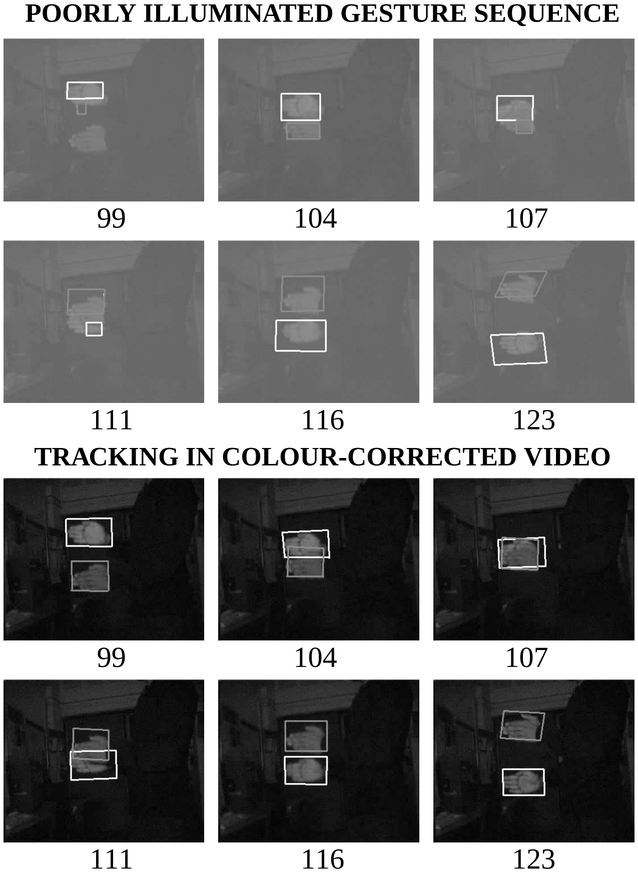

POORLY ILLUMINATED GESTURE SEQUENCE
99
104
107
111
116
123
TRACKING IN COLOUR-CORRECTED VIDEO
99
104
107
111
116
123

Figure 7: The use of colour correction for enhanced tracking. The contrast has actually been enhanced in the first set of frames, for clarity. Occlusion begins at frame 104 and ends at frame 116.

3. CONCLUSIONS

This paper presents a two hand shape-based on-line tracker. The predictive EigenTracking framework allows articulated objects with changing shape (hands, here) to be efficiently tracked. The system is robust to cases of background clutter, other moving objects, and mutual hand occlusions. The EigenTracking framework allows us to identify the left and right hand correctly, after occlusion. For certain cases of hand motion, we propose a framework to take measurements even during occlusion, thus enhancing tracking accuracy. The use of colour constancy makes it robust to poor illumination conditions, as well. We show results of successful tracking for a large number of sequences.

Acknowledgment

Funding support of the Naval Research Board, Government of India, is gratefully acknowledged.

REFERENCES

[1] A. Utsumi and J. Ohya, "Multiple Hand Gesture Tracking using Mutliple Cameras," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp. 473 – 478.

[2] J. Mammen, S. Chaudhuri, and T. Agrawal, "Tracking of both hands by estimation of erroneous observations," in *Proc. British Machine Vision Conference (BMVC)*, 2001.

[3] N. Peterfreund, "Robust Tracking of Position and Velocity with Kalman Snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 564 – 569, June 1999.

[4] A. Elgammal and L. Davis, "Probabilistic Framework for Segmenting People Under Occlusion," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 145 – 152.

[5] S. J. McKenna, S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Tracking Groups of People," *Computer Vision and Image Understanding*, vol. 80, pp. 42 – 56, 2000.

[6] I. Haritaoglu, D. Harwood, and L. S. Davis, "Multiple people detection and tracking using silhouettes," in *IEEE International Workshop on Visual Surveillance*, 1999.

[7] D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," in *Proc. European Conference on Computer Vision (ECCV)*, 1994, pp. 189 – 196.

[8] J. Sherrah and S. Gong, "Resolving Visual Uncertanity and Occlusion through Probabilistic Reasoning," in *Proc. British Machine Vision Conference (BMVC)*, 2000.

[9] A. Shamaie and A. Sutherland, "A dynamic model for real-time tracking of hands in bimanual movements," in *5th International Gesture Workshop, Geneva*, April 2003.

[10] M. J. Black and A. D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63 – 84, 1998.

[11] N. Gupta, P. Mittal, S. Dutta Roy, S. Chaudhury, and S. Banerjee, "A Predictive Scheme for Appearance-based Hand Tracking," in *Proc. National Conference on Communications (NCC)*, 2002, pp. 513 – 522.

[12] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation For Visual Tracking," *International Journal of Computer Vision*, vol. 28, no. 1, pp. 5 – 28, 1998.

[13] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang, "An Eigenspace Update Algorithm for Image Analysis," *Graphical Models and Image Processing*, vol. 59, no. 5, pp. 321 – 332, September 1997.

[14] A. Nayak and S. Chaudhuri, "Self-induced Color Correction for Skin Tracking Under Varying Illumination," in *Proc. International Conference on Image Proccesing*, September 2003.

[15] R. Kjeldsen and J. Kender, "Finding Skin in Color Images," in *Proc. Intl. Conf. on Automatic Face and Gesture Recognition*, 1996, pp. 312 – 317.