Modelling and Recognising Spatio-Temporal Hand Gestures with an Uncalibrated Camera

Kaustubh Srikrishna Patwardhan
Dept. of Electrical Engineering, IIT Bombay, Powai, Mumbai - 400 076, INDIA
kaustubhp@gmail.com

Sumantra Dutta Roy
Dept. of Electrical Engg., IIT Delhi, Hauz Khas, New Delhi - 110 016, INDIA
sumantra@cse.iitd.ac.in

Abstract

Tracking highly articulated 3-D objects such as human hands with an uncalibrated camera is not an easy task. In addition to changes in position, such objects change their shape and appearance with time. We present a robust tracker for such cases which works well in spite of cases of other similar moving objects, and background clutter. Our hand gesture analysis system is based on such a tracker. The system can recognise gestures involving the same hand shapes following different trajectories, and vice versa.

Keywords: Dynamic Space-Time Hand Gestures, Predictive EigenTracker

1. INTRODUCTION

We present a novel appearance-based tracker capable of tracking objects that change both their position as well as shape, size and appearance. Our gesture analysis system analyses dynamic space-time gestures based on the output of this robust tracker. We also address the problem of effective selection of a gesture set for a particular task.

Having a calibrated camera system with multiple cameras helps in getting 3-D information, which can disambiguate between many otherwise difficult situations. However, it is well-accepted in the field of computer vision and image analysis that calibration is a particularly cumbersome process, and requires a lot of precision and accuracy. Systems often try to do away with calibrated cameras, and the success of a method is often gauged by the relative number of constraints it can do away with. Since the 1990s, there has been a trend towards maximising the amount of useful information that can be extracted from single off-the-shelf uncalibrated cameras. This enables the creation of more applications which can use simple systems without any cumbersome set-up required, or the restrictive requirement of the presence of a trained operator in computer vision and image analysis. An example of such an application is the panorama mode in digital cameras. The mathematics behind a panorama dictate that the camera should be rotated exactly about its optical centre, which will related corresponding two points in two images by a 3×3 non-singular matrix (Szeliski, 1994). In practical terms, for far-away objects and the camera held at an arm's length and rotated slowly, the approximations hold good, and one gets a fairly visually acceptable output.

Pavlovic and co-workers review existing hand gesture recognition techniques (Pavlovic, Sharma, and Huang, 1997). They point out that 3-D based approaches are restrictive, and apart from a few (such as a 27-degree-of-freedom parameter estimation system), such methods are not very general, and as research has progressed in the direction of systems with least restrictive requirements, that use simpler and far more robust features and algorithms. Many approaches focus primarily on motion/trajectory information e.g., (Yeasin and Chaudhuri, 2000), (Min, Yoon,

Soh, Yang, and Ejima, 1999), or shape information e.g., (Triesch and Malsburg, 2002). In this work, we consider gestures which cannot be differentiated on the basis of shape, or trajectory information alone. In other words, we consider gestures with the same shape traversing different trajectories, and different shapes traversing the same trajectories. Our system does not have any explicit feature detection step, unlike other systems which are limited by the the restrictions of any one particular feature detector. In our case, we use an eigenspace-based approach - using pixel information from the entire image, obliterating an intermediate (and possibly error-prone) feature-detection step. The eigenspace models the visual appearance of an object. Using significant eigenvectors to approximate the general appearance of an object is consistent with Gibson's notion of visual invariants (Gibson, 1979). Further, our method is robust to common hand shape deformations, which often make other systems restrictive: rotation, translation, scale and shear. For the temporal modelling and recognition, most systems use Finite State Machines (FSMs) e.g., (Yeasin and Chaudhuri, 2000), or the more general Hidden Markov Models (HMMs) e.g., (Nam and Wohn, 1997), (Kapuscinski and Wysocki, 2001), (Min, Yoon, Soh, Yang, and Ejima, 1999), (Ng and Ranganath, 2002). Our method does not have the extensive training requirement of HMMs, and is easily adaptable to a given gesture set. We also address the issue of designing a set of gestures for a particular task. To the best of our knowledge, no other work in the literature except our prior work (Patwardhan and Dutta Roy, 2004), (Patwardhan and Dutta Roy, 2007), address these issues, or proposes solutions to these.

A gesture analysis system with as few restrictive assumptions as possible - is very natural for human-computer interaction. Speech-based systems have their restrictions in terms of the learning - the training set. Gestures (especially hand gestures) are a very natural mode of communication in most cultures across the globe. Hearing- and speech-impaired individuals use sign language to communicate - while a real-time computer vision-based general sign language interpretation and analysis system is perhaps too much to ask for, our gesture analysis system that is relatively general and addresses two important problems as mentioned above. First, the use of both shape and trajectory information, and second, issues concerning the construction of a gesture set. To control a system using visual input using an off-the-shelf uncalibrated camera - is an important application which we explore in the experimental results section.

The organisation of the rest of the paper is as follows. We first introduce a robust and efficient tracker in Section 2, which tracks objects with changing appearances across cluttered backgrounds, with other (possibly similar) objects moving about. This section also develops a representation for the changing appearance and trajectory of a moving object being tracked. Section 3 builds up on the above representation of a moving object being tracked - the tracker output, and deals with the issue of constructing a gesture set for a particular application, which increases the accuracy of the gesture recognition system. We present representative experimental results on a sample application of our gesture analysis system: controlling a Winamp©-like audio player using a simple off-the-shelf uncalibrated camera. This is in Section 4. Section 5 presents some discussions, and concludes the paper.

2. A ROBUST AND EFFICIENT APPEARANCE-BASED TRACKER

Tracking a moving object across video frames is a difficult task for many reasons. The first is like any problem in computer vision, we are given 2-D images of 3-D objects with 3-D motions. Second, the background could be cluttered - an assumption of a constant or a fixed background is not applicable in many real-world scenarios. Third, there could be more than one object moving in the scene, and worse still, many of them could be similar to the object of interest to be tracked. Lastly, the moving object could also change its appearance as it moves across video frames.

Unlike other visual trackers, Black and Jepson's EigenTracker (Black and Jepson, 1998) tracks objects which change both their position as well as appearance. They pose the problem as estimation of the eigenspace reconstruction coefficients $\mathbf{s}$, modulo a deformation (modelled as an affine transformation with coefficients $\mathbf{a}$), which minimises a robust error function between the parametrised image $\mathbf{I}$ (indexed by its pixel location $\mathbf{x}$. i.e., $\mathbf{x}$ is a pixel in the image) and the

ALGORITHM PREDICTIVE_EIGENTRACKER

1. INITIALISATION: Delineate object of interest
2. Consider sample set $\{\mathbf{s}_{t-1}^{(i)}, \pi_{t-1}^{(i)}\}, 1 \leq i \leq N$, starting at $t = 1$

REPEAT FOR ALL frames:

3. SELECT sample set for prediction using $\pi_{t-1}^{(i)}$
4. PREDICT new sample set: seed values for non-linear optimisation
5. Get MEASUREMENT $\pi_t^{(i)}$, optimising affine params $\mathbf{a}$ & recons coeffs $\mathbf{s}$
6. Tracker OUTPUT: the sample $\mathbf{s}_t^{(j)} \equiv$ least recons error
7. IF recons error $\in (T_1, T_2]$ THEN update eigenspace
   ELSE IF recons error $> T_2$ THEN construct eigenspace afresh

FIGURE 1: Our Predictive EigenTracker: An Overview (Details in Sec. 2)

reconstructed one $\mathbf{Uc}$ (where $\mathbf{U}$ is the matrix of the most significant eigenvectors):

$$arg\ min_{\forall \mathbf{x}}, \rho(\mathbf{I}(\mathbf{x} + \mathbf{f}(\mathbf{x}, \mathbf{a})) - [\mathbf{Us}](\mathbf{x}), \sigma) \tag{1}$$

Here, $\rho(x, \sigma) = x^2/(x^2 + \sigma^2)$ is the robust error function, and σ is a scale parameter (Black and Jepson, 1998). The robust error function (also sometimes referred to as an M-estimator) is one that minimises the effect of outliers - which would otherwise affect a least-squares problem. The distortion (the change in the position for a pixel $\mathbf{x}$) is modelled as a 2-D affine transform:

$$\mathbf{f}(\mathbf{x}, \mathbf{a}) = \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \mathbf{x} \tag{2}$$

There are several disadvantages of the original EigenTracker formulation. First, the non-linear optimisation is time-consuming, and needs a good seed point. Typically, the optimisation does not work if motion between frames is more than 2-3 pixels. The original EigenTracker assumes that all appearances of the object to be tracker are learnt in an off-line phase, of which the eigenvectors corresponding to the top eigenvalues are considered. In general, it is not possible to learn all possible appearances of highly articulated objects offline. We propose an efficient appearance-based tracker called a Predictive EigenTracker (Sec. 2).

Our Predictive EigenTracker (Gupta, Mittal, Patwardhan, Dutta Roy, Chaudhury, and Banerjee, 2004) enhances the capability of the original EigenTracker (Black and Jepson, 1998) in three ways. We propose a Particle filtering/CONDENSATION (Isard and Blake, 1998)-based predictive framework (which can work with any distribution). This speeds up the search for the object of interest, and speeds up the non-linear optimisation with a good seed point. We learn and track unknown views of an object *on the fly* with an efficient on-line eigenspace update mechanism. Fig. 1 gives an overview of the Predictive EigenTracker. We use a six-element state vector $\mathbf{X}_t$ with a second order AR model for state/process dynamics: $\mathbf{X}_t = \mathbf{D}_2\mathbf{X}_{t-2} + \mathbf{D}_1\mathbf{X}_{t-1} + \mathbf{w}_t$, where t represents time, $\mathbf{D}_i$ are 6×6 matrices, and $\mathbf{w}_t$ is a zero-mean, white, Gaussian random vector. We emphasise here that the particular motion model chosen for the experiments here (in our case, the second-order AR model) does not constrain the formulation in any way - this can work with any model which is better suited to the dynamics of the motion being tracked. In the absence of any prior knowledge about the dynamics, one often uses a random-walk model with a large entries on the covariance matrix diagonal (Dutta Roy, Tran, Davis, and Sreenivasa Vikram, 2008). The six-element state vector can be the affine coefficients a_i, or the coordinates of the 3 points defining the bounding parallelogram.

The tracker uses a set of N samples $\{\mathbf{s}_{t-1}^{(i)}, \pi_{t-1}^{(i)}\}$, $1 \leq i \leq N$, where samples $\mathbf{s}_{t-1}^{(i)}$ are drawn from $P(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})$, the state distribution given all observations $\mathbf{Z}_1 \ldots \mathbf{Z}_{t-1}$ (6-element observation vectors) thus far. ($P(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})$ comes from the state/process dynamics model - for instance, the second order AR model in our experiments.) We use a combination of skin colour and motion cues to perform *fully automatic initialisation* (Step 1 in Fig. 1). The algorithm next selects a new sample set according to $\pi_{t-1}^{(i)}$, and uses the State/Process Dynamics Model to predict the new sample set. (Steps 3 and 4 in Fig. 1). *These serve as the seed values for the basic EigenTracker's*

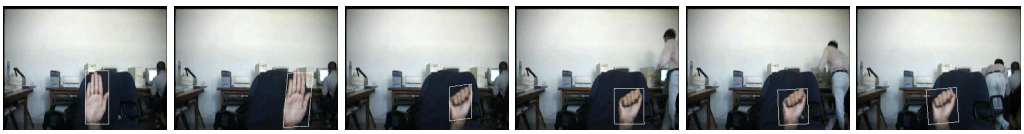

FIGURE 2: Successful tracking of a highly articulated shape in clutter, with other moving objects - with our predictive EigenTracker (Details in Sec. 2).

non-linear optimisation (Eqn. 1). The optimisation finds the affine coefficients $\mathbf{a}$ and eigenspace reconstruction coefficients $\mathbf{s}$ corresponding to the least reconstruction error. We formulate the observation/measurement $P(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{s}_t^{(i)})$ as being proportional to the negative exponential of the above reconstruction error. (This is fairly standard in any method based on particle filtering - to have Gaussians around observations (Isard and Blake, 1998).) We declare the sample with the least reconstruction error as the tracker output. Fig. 2 shows a sample of successful tracking using our predictive EigenTracker.

3. MODELLING DYNAMIC SPACE-TIME GESTURES

The output of our Predictive EigenTracker is a set of eigenspace reconstruction coefficients $\mathbf{c}$ (shape parameters) and affine transformation coefficients $\mathbf{a}$ (motion parameters, or equivalently, the state vectors $\mathbf{X_t}$). A large reconstruction error (Fig. 1) indicates a new shape of the gesticulating hand. We formulate a particular gesture $\mathcal{G}_i^k$ as having k shape-trajectory vector pairs. *We model a particular gesture $\mathcal{G}_i^k$ as an $m-$ dimensional vector of a sequence of shape and trajectory coefficients, $\mathbf{g}_i^k = [\mathbf{s}_{i_1}\mathbf{t}_{i_1} \ldots \mathbf{s}_{i_k}\mathbf{t}_{i_k}]^T$. Thus, $\mathcal{G} = \bigcup_{i,k} \mathcal{G}_i^k$ represents the gesture vocabulary, or set.* For each set of k shape-trajectory coefficient pair vectors $\mathbf{g}_i^k$, we compute the mean gesture vector $\overline{\mathbf{g}_i^k}$, and covariance matrix $\mathbf{\Sigma}_i^k$. Given a query gesture $\mathcal{G}_j^k$, we compute the Mahalanobis distance d_{ij}^k of its vector representation $\mathbf{g}_j^k$ from that of all gestures $\mathcal{G}_i^k$ with k shape-trajectory coefficient pairs:

$$d_{ij}^k = \left[(\mathbf{g}_j^k - \overline{\mathbf{g}_i^k})^T (\mathbf{\Sigma}_\mathbf{i}^\mathbf{k})^{-1} (\mathbf{g}_j^k - \overline{\mathbf{g}_i^k})\right]^{\frac{1}{2}}, \ \forall \mathcal{G}_i^k \in \mathcal{G} \tag{3}$$

The probability of the given gesture $\mathcal{G}_j^k$ being one of the gestures in the given set $\mathcal{G}_i^k$ is given by $p_{ij}^k = \frac{exp(-d_{ij}^k)}{\sum_i exp(-d_{ij}^k)}$. *For high recognition accuracy, gestures should be so chosen that the gesture-classes are well-separated in gesture-space, and the intra-class distance is small.* This also puts an upper bound on the accuracy of the recognition system for a particular chosen set of gestures, and their representation. Our earlier works (Patwardhan and Dutta Roy, 2004), (Patwardhan and Dutta Roy, 2007) explain many of these ideas in detail.

It is important to note that the formulation is independent of the relative speed of performing the gesture. There is no normalisation (using dynamic time warping/dynamic programming, or otherwise) on the number of video frames corresponding to a gesture, nor on any constancy of the speed of movement of the hand during the gesture.

4. EXPERIMENTS WITH A REPRESENTATIVE GESTURE SET

We have chosen a representative gesture set (for controlling an audio player such as Winamp©). Fig. 3 shows the eight gestures in our set. Clearly, the gestures cannot be differentiated on the basis of shape or trajectory information alone. *We have chosen 4 basic hand shapes to be as far apart in appearance-space as possible, and the same goes for the gestures themselves.* In our representative experimental setup, each gesture consists of two different hand shapes, requiring two epoch changes in the tracking phase. For the trajectories in our sample gesture set, we use a least-squares linear approximation. *Gesture pairs {2, 6}, {3, 4}, and {7, 8} involve identical hand shapes (in order) and differ only in the hand trajectories. Conversely, gesture pairs {1, 5}, {2, 3}, and {4, 6} have different hand shapes trace identical trajectories* (Fig. 3). In our set, we take the 5 most significant eigenvalues (they correspond to above 90% of the total energy). We represent

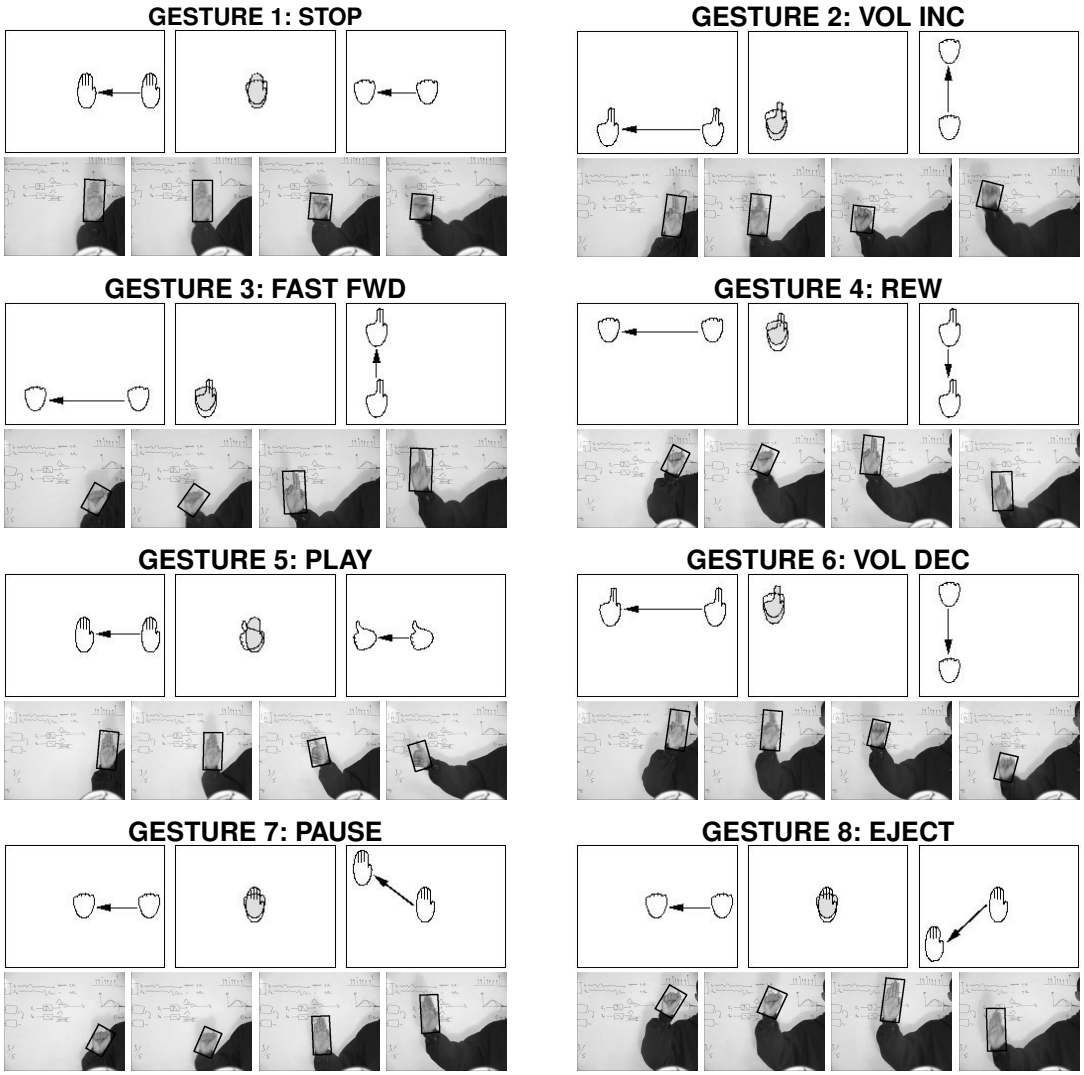

FIGURE 3: A sample gesture set for controlling an audio player such as Winamp©: the gestures cannot be distinguished on the basis of shape or trajectory alone. The top row for each gives a schematic, the bottom one shows representative tracker output frames.

each gesture by a 14 element vector, with 5 (shape) + 2 (trajectory) parameters corresponding to each epoch. We have tested the gesture recognition performance of this framework using 64 gestures present in the training set, and 16 additional gestures which were not used during the training phase. Table 1 lists the Mahalanobis distances of a set of 9 gestures (not used for training) from the template gestures.

5. DISCUSSION AND CONCLUSIONS

Gestures are a common mode for human-human interaction, and are a possible natural mode for human-computer interaction as well - and this is not just for physically challenged people with speech and hearing impairment. This work presents a vision-guided gesture analysis system with minimum assumptions about the vision sensor - something we feel will be more natural from a human-computer interaction point of view. We base our system on a robust appearance-based visual tracker, which can successfully track human hands (among other objects) in spite of cluttered backgrounds. The gesture recognition formulation also takes into account the modelling aspect. We address the issue of formulating a set of gestures for a particular application - which will maximise recognition accuracy. Planned extensions of this work include applying this framework to two-handed gestures, possibly using our robust two-hand tracker (Barhate,

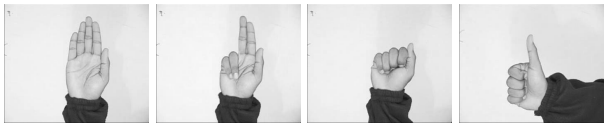

FIGURE 4: The basic hand shapes themselves are significantly different in appearance (Sec. 4).

	GES. 1	GES. 2	GES. 3	GES. 4	GES. 5	GES. 6	GES. 7	GES. 8
1	$\mathbf{0.98}$	1.6×10^8	5.9×10^7	2.7×10^7	9.4×10^7	1.0×10^8	8.6×10^7	1.6×10^8
2	2.1×10^8	$\mathbf{0.26}$	2.1×10^7	4.4×10^7	1.6×10^9	2.9×10^7	1.5×10^7	2.1×10^8
3	3.0×10^8	5.1×10^7	$\mathbf{0.88}$	4.6×10^7	1.8×10^9	3.8×10^7	5.4×10^7	1.7×10^8
4	3.2×10^8	6.5×10^7	6.0×10^7	$\mathbf{0.77}$	1.8×10^9	5.4×10^6	6.5×10^7	1.6×10^8
5	5.9×10^7	1.8×10^8	7.4×10^7	6.9×10^7	$\mathbf{0.95}$	1.1×10^8	9.5×10^7	1.3×10^8
6	4.2×10^8	4.4×10^7	4.7×10^7	4.7×10^6	2.2×10^9	$\mathbf{0.99}$	7.1×10^7	2.9×10^8
7	3.6×10^7	8.9×10^7	4.5×10^7	1.0×10^8	6.8×10^8	1.3×10^8	$\mathbf{0.88}$	2.6×10^8
8	7.7×10^7	9.2×10^7	1.3×10^8	4.2×10^7	6.2×10^8	6.5×10^7	1.2×10^8	$\mathbf{0.97}$

TABLE 1: Mahalanobis distance of Set 9 gestures (not used for training) from template gestures. The inter-gesture distances are orders of magnitude apart.

Patwardhan, Dutta Roy, Chaudhuri, and Chaudhury, 2004), which models all possible cases of hand-hand interactions.

References

Barhate, K. A., Patwardhan, K. S., Dutta Roy, S., Chaudhuri, S., Chaudhury, S., 2004. Robust Shape Based Two Hand Tracker. In: Proc. IEEE International Conference on Image Processing (ICIP). pp. 1017 – 1020.

Black, M. J., Jepson, A. D., 1998. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. International Journal of Computer Vision 26 (1), 63 – 84.

Dutta Roy, S., Tran, S. D., Davis, L. S., Sreenivasa Vikram, B., 2008. Multi-Resolution Tracking in Space and Time. In: Proc. IAPR- and IEEE-sponsored Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP). pp. 352 – 358.

Gibson, J., 1979. The Ecological Approach to Visual Perception. Lawrence Erlbaum Associates.

Gupta, N., Mittal, P., Patwardhan, K. S., Dutta Roy, S., Chaudhury, S., Banerjee, S., 2004. Online Predictive Appearance-based Tracking. In: Proc. IEEE International Conference on Image Processing (ICIP). pp. 1041 – 1044.

Isard, M., Blake, A., 1998. CONDENSATION - Conditional Density Propagation For Visual Tracking. International Journal of Computer Vision 28 (1), 5 – 28.

Kapuscinski, T., Wysocki, M., 2001. Hand Gesture Recognition for Man-Machine Interface. In: Proc. Second International Workshop on Robot Motion and Control. pp. 91 – 96.

Min, B., Yoon, H., Soh, J., Yang, Y., Ejima, T., 1999. Visual Recognition of Static/Dynamic Gesture: Gesture Driven Editing System. Journal of Visual Languages and Computing 10, 291 – 309.

Nam, Y., Wohn, K., 1997. Recognition of Hand Gestures, with 3D, Non-linear Arm movement. Pattern Recognition Letters 18, 105 – 113.

Ng, C. W., Ranganath, S., 2002. Real-Time Gesture Recognition System and Application. Image and Vision Computing 20, 993 – 1007.

Patwardhan, K. S., Dutta Roy, S., 2004. Dynamic Hand Gesture Recognition using Predictive EigenTracker. In: Proc. IAPR- and IEEE-sponsored Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP). pp. 675 – 680.

Patwardhan, K. S., Dutta Roy, S., 2007. Hand gesture modeling and recognition involving changing shapes and trajectories, using a predictive eigentracker. Pattern Recognition Letters 28 (3), 329 – 334.

Pavlovic, V. I., Sharma, R., Huang, T. S., July 1997. Visual Interpretation of Hand Gestures for Human-Computer Interaction. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7), 677 – 695.

Szeliski, R., May 1994. Image Mosaicing for Tele-Reality Applications. Tech. Rep. CRL 94/2, Cambridge Research Laboratory.

Triesch, J., Malsburg, C., 2002. Classification of Hand Postures against Complex Backgrounds using Elastic Graph Matching. Image and Vision Computing 20, 937 – 943.

Yeasin, M., Chaudhuri, S., 2000. Visual Understanding of Dynamic Hand Gestures. Pattern Recognition 33, 1805 – 1817.