Human Body Gesture Recognition using Adapted Auxiliary Particle Filtering

A.Oikonomopoulos

Computing Department
Imperial College London
London, UK

M.Pantic

Computing Department
Imperial College London
London, UK

Abstract

In this paper we propose a tracking scheme specifically tailored for tracking human body parts in cluttered scenes. We model the background and the human skin using Gaussian Mixture Models and we combine these estimates to localize the features to be tracked. We further use these estimates to determine the pixels which belong to the background and those which belong to the subject's skin and we incorporate this information in the observation model of the used tracking scheme. For handling self-occlusion (i.e., when one body part occludes another), we incorporate the information about the direction of the observed motion into the propagation model of the used tracking scheme. We demonstrate that the proposed method outperforms the conventional Condensation and Auxiliary Particle Filtering when the hands and the head are the tracked body features. For the purposes of human body gesture recognition, we use a variant of the Longest Common Subsequence algorithm (LCSS) in order to acquire a distance measure between the acquired trajectories and we use this measure in order to define new kernels for a Relevance Vector Machine (RVM) classification scheme. We present results on real image sequences from a small database depicting people performing 15 aerobic exercises.

1. Introduction

Vision-based analysis of hand and body gestures is nowadays one of the most active fields of computer vision, due to its practical importance for security (video surveillance, monitoring), natural multimodal interfaces, augmented reality, smart rooms, object-based video compression and driver assistance.. Furthermore, it offers the key to ambient intelligence, anticipatory interfaces, and human computing, through the ability to unobtrusively sense certain behavioral cues of the users and to adapt to their typical behavioral patterns and the context in which they act. Tremendous amount of work has been done in the field in recent years [1],[2].

In order to obtain a semantic description of the content of a scene, we do not need to use all the available information. What is happening in a scene can be determined by monitoring the temporal transitions of the scene's non-static elements. The main objective of tracking is to estimate the state x_k (e.g. position, pose) given all the measurements $z_{1:k}$ up to the current time instant k. In a probabilistic framework, this translates in the construction of the a posteriori probability $p(x_k|z_{1:k})$. Theoretically, the optimal solution in case of Gaussian noise in the measurements is given by the Kalman filter [3], which yields the posterior being also Gaussian. However, in nonlinear and non-Gaussian state estimation problems Kalman filters can be significantly off. To overcome the limitations of Kalman filtering, the so-called Condensation algorithm was proposed [4], [5]. The main idea behind condensation is to maintain a set of possible solutions called particles. By maintaining a set of solutions instead of a single estimate as is done by Kalman filtering, particle filters are more robust to missing and inaccurate data. The major drawback of the classic Condensation algorithm, however, is that a large amount of particles might be wasted because they are propagated into areas with small likelihood. In order to overcome this problem, a number of variants to the original algorithm have been proposed, having as a common characteristic the goal of achieving a more optimal allocation of new particles, through the use of kernels [6], [7], orientation histograms [8] or special transformations like Mean Shift [9].

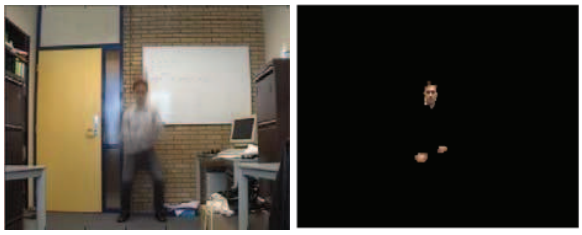
Despite the improvement in the tracking performance of the previous methods, the inherent problem of the classic condensation algorithm, that is, the propagation of particles in areas of small likelihood is not sufficiently addressed. In order to effectively deal with this issue, the Auxiliary Particle Filtering (APF) algorithm was proposed by Pitt and Shephard [10]. The APF algorithm operates in two steps. At first, particles are propagated and their likelihood is evaluated. Subsequently, the algorithm chooses again and propagates the particles according to the likelihood of the previous step. Since the introduction of the APF algorithm, a number of variants have been proposed in order to address different issues. In [11] a modified APF tracking scheme is proposed for the tracking of deformable facial features,

978-1-4244-1696-7/07/$25.00 ©2007 IEEE.

like mouth and eye corners. The method uses an invariant color distance that incorporates a shape deformation term as an observation model to deal with the deformations of the face. In order to take into account spatial constraints between tracked points, the particle filter with factorized likelihoods is proposed in [12], where the spatial constraints between different facial features are pre-learned and the proposed scheme tracks constellations of points instead of a single point, by taking into account these constraints.

The creation of trajectory-based representations of actions via feature tracking has been used in extend. In [13], the spatiotemporal curvatures of the trajectories of moving objects, such as the hands, are used in order to represent human actions. The local maxima of these curvatures are view-invariant and are used for image sequence alignment and matching of the actions. Trajectories that represent human activities in moving camera environments are matched in [14], by relaxing standard geometrical constraints, like constant speed and constant acceleration. Three-dimensional representations have also been extensively studied for human action recognition. In [15] human actions are treated as three-dimensional shapes in the space-time volume. The method utilizes properties of the solution to the Poisson equation to extract space-time features such as local space-time saliency, action dynamics, shape structure and orientation, while spectral clustering is used in order to group similar actions. In [16], long video sequences are segmented in the time domain by detecting single events in them. The detection is completely unsupervised, since it is done without any prior knowledge of the types of events, their models, or their temporal extent. The method can be used for event-based indexing even when only one short example-clip is available.

In this paper we automatically select body features by combining background and skin color cues. We model the background of the sequences using Gaussian Mixture Models (GMM). We learn the parameters of the mixture using EM and the first 24 frames of the sequences as training data. We obtain a skin color model in a similar way, using a set of training images consisting of human faces. We use the detected features to initialize an adapted Auxiliary Particle Filtering tracking scheme. We incorporate skin and background cues to the observation model to ensure that only skin patches belonging to the foreground are tracked. We use an adapted motion model to predict the location of particles at the next time instant. We compare our results with the ones acquired using Condensation and simple Auxiliary Particle Filtering (APF). We examine the effectiveness of our motion model by comparing our results with the ones acquired by an APF that uses only the enhanced observation model. We use LCSS to acquire a distance measure between trajectories and define kernels for an RVM classification scheme. Our results show the superiority of the



(a) (b)

Figure 1: (a) Estimated background for one of the sequences in our database and (b) the segmented hands and head by combining.

representations acquired by the proposed tracker.

The remainder of this paper is organized as follows: in section 2 we describe the automatic localization process that we followed for the selection of body features. In section 3 we analyze the proposed observation and motion models that we incorporated in the classic APF tracking scheme. In section 4 we describe the recognition process that we followed in order to compare the trajectories derived from section 3. In section 5 our experimental results are given and in section 6 our final conclusions are drawn.

2 Feature Localization

Similar to [17] we use a mixture of Gaussian distributions in order to model the skin. The parameters of the Gaussians are estimated using EM. The model is trained on approximately 700 frontal facial images from the FERET database [18]. To construct the model, we convert our training images to the nRGB colorspace. We implement the background estimation algorithm of [19] in order to determine background pixels. The recent history of each pixel position is modeled by a mixture of K Gaussians. The covariance matrix Σ is assumed to be diagonal, meaning that the RGB values of the pixels are assumed to be uncorrelated. We use the first 24 frames (i.e. 1 second) of our sequences and EM in order to estimate the parameters of the mixture. In Fig. 1(a), the estimated background model for a sequence where the subject is raising both of its hands is given. The final result of the segmentation for one of the sequences in our dataset is illustrated in Fig. 1(b).

3 Feature Tracking

Recently, particle filtering tracking schemes [4], [10], have been successfully used [11] to track the state of a temporal event given a set of noisy observations. Its ability to maintain multiple solutions makes it particularly attractive when the noise in the observations is not Gaussian and makes it

robust to missing or inaccurate data.

Let us denote by c the template containing the color information in a rectangular window centered at each point to be tracked, by α the unknown location of the feature at the current time instant and by $Y = \{y^1, \ldots, y^-, y\}$ the observations up to the current time instant. The main idea of the particle filtering is to maintain a particle based representation of the a posteriori probability $p(\alpha|Y)$ of the state α given all the observations Y up to the current time instance. The distribution $p(\alpha|Y)$ is represented by a set of pairs (s_k, π_k) such that if s_k is chosen with probability equal to π_k, then it is as if s_k was drawn from $p(\alpha|Y)$. Suppose that we have a particle based representation of the density $p(\alpha^-|Y^-)$, that is we have a collection of K particles and their corresponding weights (i.e. (s_k^-, π_k^-)). Then, the Auxiliary Particle Filtering can be summarized as follows:

1. Propagate all particles s_k^- via the transition probability $p(\alpha|\alpha^-)$ in order to arrive at a set of K particles μ_k.
2. Evaluate the likelihood associated with each particle μ_k, that is let $\lambda_k = p(y|\mu_k; c)$.
3. Draw K particles s_k^- from the probability density, represented by the collection $(s_k^-, \lambda_k \pi_k^-)$. In this way, the auxiliary particle filter favors particles with high λ_k.
4. Propagate each particle s_k^- with the transition probability $p(\alpha|\alpha^-)$ in order to arrive at a collection of K particles ${s_k}'$.
5. Assign a weight ${\pi_k}'$ to each particle as follows,

$$ {w_k}' = \frac{p(y|{s_k}'; c)}{\lambda_k}, \quad {\pi_k}' = \frac{{w_k}'}{\sum_j w_j} \tag{1} $$

In this work we use for the definition of $p(y|\mu_k; c)$ the observation model described in [11] along with background and skin color information. Furthermore, we utilize an adapted motion model in order estimate the most probable position of the particles in the next state and deal with the problem of self-occlusion.

3.1 Observation Model

Shadows and color variations affect the accuracy of the feature localization process of section 2. The tracker templates may contain a considerable portion of background pixels, misleading the tracking process. Furthermore, since we want to track body parts consisting of skin pixels, we want to ensure that our tracker will follow such regions. To deal with these issues, we enhance the observation model of our tracking scheme using background and skin color cues.

Let us denote by $\{N(\boldsymbol{\Sigma^b}, \boldsymbol{\mu^b})\}$ the set of background Gaussians with the largest prior weights per pixel and by $N(\boldsymbol{\Sigma^s}, \boldsymbol{\mu^s})$ the dominant Gaussian in the skin model mixture. Then the observation model that we use is defined as:

$$ p(y|s_k; c, \{N(\boldsymbol{\Sigma^b}, \boldsymbol{\mu^b})\}^k, N(\boldsymbol{\Sigma^s}, \boldsymbol{\mu^s})) = \\ \alpha \cdot p(y|s_k; c) \cdot p(y|s_k; \{N(\boldsymbol{\Sigma^b}, \boldsymbol{\mu^b})\}^k) \cdot p(y|s_k; N(\boldsymbol{\Sigma^s}, \boldsymbol{\mu^s})), \tag{2} $$

where α is a normalization term and $\{N(\boldsymbol{\Sigma^b}, \boldsymbol{\mu^b})\}^k$ denotes the set of Gaussians that describe the image patch defined by s_k. For the first term of the product in eq. 2 we use the invariant color distance of [11]. We use the inverse sigmoid function in order to define $p(y|s_k; \{N(\boldsymbol{\Sigma^b}, \boldsymbol{\mu^b})\}^k)$:

$$ p(y|s_k; \{N(\boldsymbol{\Sigma^b}, \boldsymbol{\mu^b})\}^k) = 1 - \frac{1}{1 + e^{-r(d_k^b - q)}}, \tag{3} $$

where d_k^b is the average Mahalanobis distance of the M pixels in the patch defined by s_k to their corresponding background Gaussians and r, q are parameters that define the steepness and the middle point of the sigmoid respectively. The likelihood $p(y|s_k; N(\boldsymbol{\Sigma^s}, \boldsymbol{\mu^s}))$ is defined in a similar way:

$$ p(y|s_k; N(\boldsymbol{\Sigma^s}, \boldsymbol{\mu^s})) = \frac{1}{1 + e^{-r(d_k^s - q)}}, \tag{4} $$

where d_k^s is the average Mahalanobis distance of the pixels in the patch from the skin distribution.

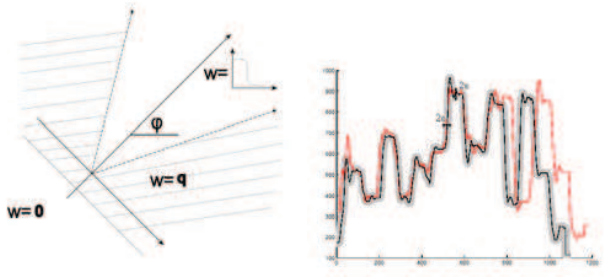
3.2 Motion Model

We incorporate an adapted motion model in order to predict the location of the tracker particles in the next time instant. Let us denote by ϕ the mean direction of motion obtained by considering a window of previous tracked states and by g the estimated displacement of the particles in the next time instant, calculated as the difference in the average locations of the tracked states in two previous time windows of N frames. A schematic representation of the motion model used is given in Fig. 2(a). As can be seen, our model assigns a large weight if the particle is located inside the clear region, which defines the dominant direction of motion. The reason for assigning a small weight q to the dashed region in the figure is to give the model some tolerance in case the motion changes direction abruptly.

4 Recognition

4.1 Longest Common Subsequence (LCSS) Algorithm

Using the analysis of the previous sections, we represent a given image sequence by a set of trajectories, where each trajectory is initialized at the points automatically selected using the procedures of section 2. Formally, an image sequence is represented by a set of trajectories $\{A_i\}, i =$



(a) (b)

Figure 2: (a) Schematic representation of the proposed motion model and (b) the notion of the LCSS matching.

$1 \dots K$, where K is the number of trajectories that consist the set. In this work, $K = 3$, since we only track the hands and head of the subjects performing the actions. Each trajectory is defined as $A_i = ((x_{i,n}, y_{i,n}), \dots)$, $n = 1 \dots N$, where $x_{i,n}, y_{i,n}$ are spatiotemporal coordinates and N is the number of samples that consist A_i. Let us define another trajectory set $\{B_i\}, i = 1 \dots K$ representing a different image sequence. Similar to $\{A_i\}$, the trajectories in $\{B_i\}$ are defined as $B_i = ((x_{i,m}, y_{i,m}), \dots)$, $m = 1 \dots M$, where M is the number of samples that consist $\{B_j\}$. We use a variant of the LCSS algorithm presented at [21], in order to compare the two sets. Before we proceed with the comparison, we transform the x and y coordinates of the consisting trajectories so that they have zero mean and we align the two sets in time using Dynamic Time Warping [20]. Let us define the function $Head(A_i) = ((x_{i,n}, y_{i,n})), n = 1 \dots N-1$, that is, the individual trajectory A_i reduced by one sample. Then, according to the LCSS algorithm, the distance between individual trajectories A_i and B_j is given by:

$$d_L(A_i, B_j) = \begin{cases} 0, & if\ A_i\ or\ B_i\ is\ empty \\ d_e((x_{i,n}, y_{i,n}), (x_{i,m}, y_{i,m})) \\ +d_L(Head(A_i), Head(B_i)), \\ if\ \ |x_{i,n} - x_{i,m}| < \varepsilon\ and\ \ |y_{i,n} - y_{i,m}| < \varepsilon \\ \max(d_L(Head(A_i), B_i), d_L(A_i, Head(B_i))) + p, \\ otherwise \end{cases}, \tag{5}$$

where d_e is the Euclidean distance, ϵ is the matching threshold and p is a penalty cost in case of mismatch. The notion of the LCSS distance of eq. 5 is depicted in Fig. 2(b).

Subsequently, the distance between sets $\{A_i\}$ and $\{B_j\}$ is defined as follows:

$$D_L(\{A_i\}, \{B_j\}) = \frac{1}{K} \sum_i d_L(A_i, B_i) \tag{6}$$

that is, the average LCSS distance between the trajectories of sets $\{A_i\}$ and $\{B_i\}$.

4.2 Relevance Vector Machine Classifier

We propose a classification scheme based on Relevance Vector Machines [22] in order to classify given examples of human actions. A Relevance Vector Machine (RVM) is a probabilistic sparse kernel model identical in functional form to the Support Vector Machines (SVM). In their simplest form, Relevance Vector Machines attempt to find a hyperplane defined as a weighted combination of a few Relevance Vectors that separate samples of two different classes. In contrast to SVM, predictions in RVM are probabilistic. Given a dataset of N input-target pairs $\{(F_n, l_n), 1 \leq n \leq N\}$, an RVM learns functional mappings of the form:

$$y(F) = \sum_{n=1}^{N} w_n K(F, F_n) + w_0\,, \tag{7}$$

where $\{w_n\}$ are the model weights and $K(.,.)$ is a Kernel function. Gaussian or Radial Basis Functions have been extensively used as kernels in RVM. In our case, we use as a kernel a Gaussian Radial Basis Function defined by the distance measure of eq. 6. That is,

$$K(F, F_n) = e^{-\frac{D_L(F, F_n)^2}{2\eta}}\,, \tag{8}$$

where η is the Kernel width. RVM performs classification by predicting the posterior probability of class membership given the input F. The posterior is given by wrapping eq. 7 in a sigmoid function, that is:

$$p(l|F) = \frac{1}{1 + e^{-y(F)}} \tag{9}$$

In the two class problem, a sample F is classified to the class $l \in [0, 1]$, that maximizes the conditional probability $p(l|F)$. For L different classes, L different classifiers are trained and a given example F is classified to the class for which the conditional distribution $p_i(l|F)$,$1 \leq i \leq L$ is maximized, that is:

$$Class(F) = \arg\max_i (p_i(l|F))\,. \tag{10}$$

5 Experimental Results

For the evaluation of the proposed method, we use aerobic exercises as a test domain. Our dataset consists of 15 different aerobic exercises, performed twice by five different subjects, leading to a set of 150 sequences. To provide ground truth for our experiments, every 5^{th} frame is manually labeled by a human operator. We use the following distance metric as a criterion for success:

$$m_e = \sum_{i=1}^{n} h_i, \tag{11}$$

where $h_i = 1$ if the euclidean distance of the computed point exceeds a predefined threshold and 0 otherwise. We

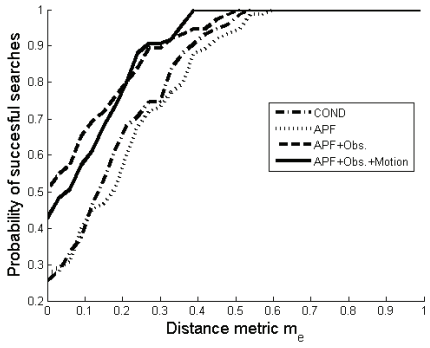


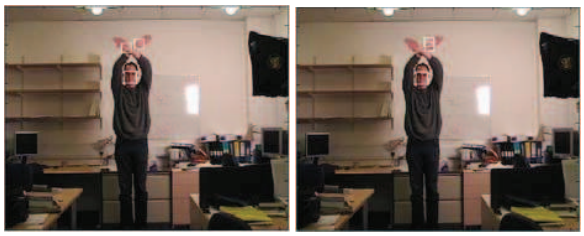
Figure 3: Cumulative distribution of measure m_e.

set the threshold equal to the size of the template used for the tracking of the point. In other words, we consider that an error has been performed if the automatically selected point is sufficiently far from the ground truth.

To illustrate the effectiveness of the proposed observation and motion models, we present in Fig. 3 the comparative results of our proposed tracking scheme with 3 different trackers, including Condensation, a simple APF and an APF that just utilizes the proposed observation model. As can be seen from the figure, the probability of successful searches achieved by the trackers incorporating the proposed observation model is always higher than the one achieved by Condensation and simple APF. In particular, the probability that the error will be less than 20% is almost 0.79 for the proposed trackers, almost 0.65 for Condensation and falls to almost 0.5 for the APF. This result highlights the performance improvement provided by the adapted observation model. Another interesting observation from Fig. 3 is that the Condensation algorithm outperforms the simple APF, even though the latter was designed in order to overcome one of the major drawbacks of Condensation, that is, a large number of particles being propagated in areas with small likelihood. This is due to the fact that, contrary to condensation, the simple APF does not utilize any motion model, and the particles are thrown with equal probability around the point at the current time instant.

Another interesting observation from Fig. 3 is that the condensation algorithm outperforms the simple auxiliary particle filter, even though the latter was designed in order to overcome one of the major drawbacks of condensation, that is, a large number of particles being propagated in areas with small likelihood. This is due to the fact that, contrary to condensation, the simple auxiliary filter does not utilize any motion model, and the particles are thrown with equal probability around the point at the current time instant.

In order to further illustrate the effect of our proposed motion model, we present in Fig. 4 two tracking instances



(a) (b)

Figure 4: Tracking instances acquired using (a) both the proposed observation and motion models and (b) only the proposed observation model.

acquired using both of the proposed observation and motion models (Fig. 4(a)) and just the proposed observation model (Fig. 4(b)). As can be seen from the figure, the tracker using both models effectively distinguishes the two hands. On the contrary, tracking of the left hand is lost in the case where only the observation model is used, since the new configuration also fits the utilized observation model.

In order to classify a test example using the Relevance Vector Machines, we constructed 15 different classifiers, one for each class, and we calculated for each test example F the conditional probability $p_i(l|F), 1 \leq i \leq 15$. Each example was assigned to the class for which the corresponding classifier provided the maximum conditional probability, as depicted in eq. 10. We followed a leave-one-out subject cross validation scheme, that is, for estimating each of the $p_i(l|F)$, an RVM is trained by leaving out the example F as well as all other instances of the same exercise that were performed by the subject from F. We performed this experiment for all of the trackers under comparison. Our classification results are depicted in Table. 5. As can be seen from the table, incorporating our proposed observation model lead to an increase of almost 5% in classification performance comparing to condensation and classic APF, while the additional incorporation of the proposed motion model resulted in an increase of almost 10%, leading us to the conclusion that the motion model plays a far more important role than the observation model.

6 Conclusions

In this work we introduced an adapted Auxiliary Particle Filter in order to track a number of selected body features. We combined background and skin color models to automatically select the features in the first frames of our sequences and we initialized a tracking scheme at the selected locations. We used the same models to enhance the observation model of the tracker. In this way, we favored particles located on foreground skin regions in the scene. In addition,

Class Labels	COND. R/P	APF R/P.	APF+Obs. R/P	APF+Obs.+Motion R/P
1	0.9 / 1	0.9 / 0.75	1/1	1/1
2	0.2 / 0.14	0 / 0	0.6 / 0.5	0.6 / 0.86
3	0.9 / 0.81	1 / 1	1 / 0.76	1 / 0.83
4	0.5 / 0.71	0.6 / 0.75	0.7 / 0.7	0.8 / 0.89
5	0.6 / 0.85	0.9 / 0.81	0.7 / 1	0.3 / 0.5
6	0.9 / 0.81	0.8 / 0.72	0.7 / 0.58	0.8 / 0.67
7	0.4 / 0.33	0.4 / 0.23	0.6 / 0.5	0.9 / 0.82
8	1 / 0.9	1 / 0.9	1 / 0.9	1 / 0.77
9	1 / 0.9	1 / 0.77	1 / 0.9	1 / 1
10	0.6 / 0.35	0.2 / 0.2	0.4 / 0.4	0.6 / 0.43
11	0.5 / 1	0.3 / 0.6	0.4 / 0.8	0.5 / 0.83
12	0.8 / 0.72	0.7 / 0.78	0.7 / 0.78	0.8 / 0.72
13	0.5 / 0.83	0.8 / 1	0.5 / 1	0.7 / 0.87
14	1 / 0.83	1 / 1	1 / 0.83	1 / 0.9
15	0.4 / 0.67	0.6 / 0.5	0.7 / 0.64	0.6 / 0.6
Total	0.68	0.68	0.73	0.77

Table 1: Recall and Precision rates for the compared trackers.

we proposed an adapted motion model to predict the location of new particles at the next time instant. We showed that our proposed tracking scheme outperforms the Condensation and Auxiliary Particle Filtering algorithms when the hands and the head are the tracked body features. We used a variant of the LCSS algorithm to acquire a distance measure for our trajectory representations and we used this measure in order to define a kernel for the RVM classifier that was used for recognition. Finally, we presented classification results on real image sequences that illustrated the superiority of the representations acquired by the proposed tracking scheme.

References

[1] J. J. Wang and S. Singh, "Video analysis of human dynamics - A survey," *Real Time Imaging*, Vol. 9 No. 5, pp. 321-346, 2003.

[2] L. Wang, W. Hu and T. Tan, "Recent Developments in Human Motion Analysis," *Pattern Recognition*, Vol. 36 No. 3, pp. 585-601, 2003.

[3] Y. Bar-Shalom, T. Fortmann, "Tracking and Data Association," *Academic Press* (1988)

[4] M. Isard and A. Blake, "Condensation conditional density propagation for visual tracking," *International Journal of Computer Vision*, Vol. 29, No. 1, pp. 5-28, 1998.

[5] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," *European Conf. on Computer Vision*, Vol. 29, No. 1, pp. 893-908, 1998.

[6] J. Schmidt, J. Fritsch and B. Kwolek, "Kernel particle filter for real-time 3D body tracking in monocular color images," *Automatic Face and Gesture Recognition*, pp. 567-572, 2006.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking", *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol. 25, No. 5, pp. 564-577, 2003.

[8] C. Yang, R. Duraiswami and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," *Proc. IEEE Int. Conf. Computer Vision*, vol. 1, pp. 212-219, 2005.

[9] C. Shan, Y. Wei T. Tan and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," *Automatic Face and Gesture Recognition*, Vol. 1, pp. 669-674, 2004.

[10] M. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filtering," *J. American Statistical Association*, Vol. 94, pp. 590-, 1999.

[11] I. Patras and M. Pantic, "Tracking deformable motion," *IEEE International Conference on Systems, Man and Cybernetics*, pp. 1066-1071, 2005.

[12] I. Patras and M. Pantic, "Particle Filtering with Factorized Likelihoods for Tracking Facial Features," *Proc. IEEE Intl Conf. on Face and Gesture Recognition*, pp. 97-102, 2004.

[13] C. Rao, A. Yilmaz and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, Vol. 50, No. 2, pp. 203-226, 2002.

[14] A. Yilma and M. Shah, "Recognizing human actions in videos by uncalibrated moving cameras," *Proc. IEEE Int. Conf. Computer Vision*, Vol. 1, pp. 150-157, 2005.

[15] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes," *Proc. IEEE Int. Conf. Computer Vision*, Vol. 2, pp. 1395-1402, 2005.

[16] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 123-130, 2001.

[17] Q. Zhu, K.-T. Cheng and C.-T. Wu, "A unified adaptive approach to accurate skin detection," *Proc. IEEE Int. Conference on Image Processing*, Vol. 2, pp. 1189-1192, 2004.

[18] The feret database, http://www.itl.nist.gov/iad/humanid/feret.

[19] C. Stauffer, "Adaptive background mixture models for real-time tracking," *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.

[20] C. Myers and L. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected word recognition," *The Bell System Technical Journal*, Vol. 60, No. 7, pp. 1389-1409, 1981.

[21] M. Vlachos, G. Kollios and D. Gunopulos, "Discovering similar multidimensional trajectories," *Proc. International Conference on Data Engineering*, pp. 673-684, 2002.

[22] M. Tipping, "The Relevance Vector Machine," *Advances in Neural Information Processing Systems*, pp. 652-658, 1999.