A Hybrid Framework for 3D Human Motion Tracking

Bingbing Ni, Ashraf Ali Kassim, Stefan Winkler

***Abstract*—In this paper, we present a hybrid framework for articulated 3D human motion tracking from multiple synchronized cameras with potential uses in surveillance systems. Although the recovery of 3D motion provides richer information for event understanding, existing methods based on either deterministic search or stochastic sampling lack robustness or efficiency.**

We therefore propose a hybrid sample-and-refine framework that combines both stochastic sampling and deterministic optimization to achieve a good compromise between efficiency and robustness. Similar motion patterns are used to learn a compact low-dimensional representation of the motion statistics. Sampling in a low-dimensional space is implemented during tracking, which reduces the number of particles drastically. We also incorporate a local optimization method based on simulated physical force/moment into our framework, which further improves the optimality of the tracking.

Experimental results on several real human motion sequences show the accuracy and robustness of our method, which also has a higher sampling efficiency than most particle filtering based methods.

***Index Terms*—Articulated 3D human motion tracking, particle filter, vector quantization principal component analysis, simulated physical force/moment**

I. Introduction

MULTIPLE view based, marker-less articulated human motion tracking has attracted a growing interest in recent years, primarily because of a large number of potential applications such as motion capture, human computer interaction, virtual reality, smart surveillance systems etc. However, most existing systems track the target as 2D blobs, from which only coarse behavior information can be extracted, e.g., walking, running, etc. Other systems such as [1]–[3] use a 2D image analysis approach to detect certain events, e.g., shaking hands, falling down, fighting. Additional tracking information such as articulated 3D motion, which is investigated in this paper, can help to obtain a more detailed understanding of human actions and interactions. However, due to the high dimensionality of human body motion, the 3D tracking problem is inherently difficult. A variety of approaches have been proposed – see [4], [5] for comprehensive surveys.

Gavrilla and Davis [6] are among the first to address the problem of tracking articulated 3D human motion by multiple synchronized images. They project a kinematic 3D human model onto each image plane and define the tracking problem as searching for the best fit between the projected model and the image contours.

Yamamoto et al. [7] carry out tracking by estimating the increment of the body pose vector between two successive images. They obtain the pose vector increment by solving a set of linear equations, which relate the image flow estimated from each view to the motion parameters of the articulated objects. Bregler and Malik [8] model the 3D human model by twists and exponential maps to perform a local search for pose estimation. Kehl and Gool [9] propose a method which takes reconstructed human voxel data as system input, and they develop a stochastic meta descent (SMD) optimization algorithm to perform human motion tracking.

Given the human model and the observed scenes, e.g., image contours or 3D reconstructions, the tracking problem could also be formulated as a registration problem. Delamarre and Faugeras [10] propose a method which creates forces between the 3D human model and the detected image contours of the moving person to align them. They also apply this concept directly to the 3D domain [11], where the physical forces are generated between the human model and the densely reconstructed 3D points of the scene. The human pose vector is updated by recursively solving a set of dynamics equations.

Kakadiaris and Metaxas [12] develop a similar framework for tracking the motion of human body parts from one or multiple cameras based on information extracted from the occluding contours. Tracking is based on applying forces generated by the displacement of the occluding contours and the model parts. They also adopt an extended Kalman filter (EKF) to predict the motion between consecutive frames. Their method is capable of auto-selecting a view point which gives the most important tracking information.

For many of the systems cited above, tracking robustness remains an issue since global optima are not guaranteed by these gradient-based or force-based optimization procedures. Such methods may easily be trapped in local minima for a long motion sequence due to error accumulation [7] and are sensitive to image noise, foreground segmentation errors, self-occlusion, etc. To address the robustness problem, a large number of algorithms based on stochastic sampling have been proposed, including particle filters [13]–[16] and related sampling-based approaches such as unscented Kalman filter [17], [18], belief propagation [19], Markov network [20] etc. These sampling-based techniques provide a promising probabilistic framework, which can handle high dimensionality, nonlinear and non-Gaussian problems, while avoiding complex analytical computations. However, in order to approximate the underlying posterior density for a high-dimensional problem

Manuscript received October 30, 2007; revised March 10, 2008.

B. Ni and A. A. Kassim are with the Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore. Corresponding author: Ashraf Ali Kassim (E-mail: eleashra@nus.edu.sg).

S. Winkler is now with Symmetricom, San Jose, CA 95131, USA.

like human motion tracking, a large number of particles are needed, which is impractical and makes the evaluation of the likelihood function very time-consuming.

There exist some methods which improve the classical particle filter method in terms of computational efficiency. Deutscher et al. [14] propose an annealed particle filter approach for full body motion tracking. Using a well designed simulated annealing strategy, they reduce the number of particles from 10^4 to 10^2. Lee et al. [16] introduce a framework that integrates analytical inference into the particle filtering scheme to reduce the computational load as well as to auto-initialize and auto-recover from tracking failure. Han and Huang [19] propose a dynamic belief propagation framework that accelerates the articulated tracking algorithm by adaptively selecting the search space based on the prediction of human motion dynamics.

Recent success on learning probabilistic models from a small training set has made it possible to further improve the Bayesian tracking of human motion. Following the idea of statistical modeling of images for texture synthesis, Sidenbladh et al. [21] describe an approach that models the appearance of articulated 3D objects into a linear subspace via weighted principal component analysis (PCA). This generative appearance model improves the performance of their particle filter tracking framework. Nonlinear dimensionality reduction techniques such as locally linear embedding (LLE) are also exploited to model the dynamic appearance of human motion [22]. Given a training database of human motion capture data, the statistics of human motion dynamics can be modeled directly. Gall et al. [15] present a method that integrates prior statistical information about the pose configurations into the general model of particle filter and therefore reduce the number of particles required. Sidenbladh et al. [23] further propose a method that represents the implicit empirical distribution of fixed length motion sequences data in a low dimensional space by PCA. Therefore tracking is equivalent to searching the best matching sequence example in the training database.

In our work, the reconstructed human surface points and normals from multi-view images are analyzed to enable identification of the human pose in each frame. Prior information is based on a small set of training samples of motion capture data. Our approach is significantly different from those presented in [15], [23]. Instead of modeling the low dimensional subspace of the fixed length motion segments [23] or learning the probability distribution of human motion in the original high-dimensional space [15], the proposed algorithm attempts to directly model the pose configuration statistics in a lower-dimensional space using a method from [24] for dimensionality reduction. Due to this compact representation, our stochastic sampling can be conducted in a much lower-dimensional space resulting in a much smaller number of required particles compared to [15].

The method proposed in this paper is essentially a hybrid tracking framework, which combines stochastic sampling using a particle filter with a deterministic searching algorithm based on simulated physical force/moment based 3D registration [25]. This sample-and-refine strategy [26] helps us achieve more efficient sampling and more accurate tracking. An additional contribution of our work is the time series extension of vector quantization principal component analysis (VQPCA) [24] to model the statistics of human motion in a compact way. Combined with our modified particle filter, this significantly improves the sampling efficiency.

Qualitative and quantitative experimental results show that the proposed framework achieves a good compromise between accuracy, robustness and tracking efficiency. The limitation of our method is the need for a training step, in which motion patterns similar to the ones of interest have to be learned by the system.

The paper is organized as follows: Section II describes our human model. Section III outlines the 3D human reconstruction method used. Section IV describes our probabilistic modeling of human motion in detail. Section V explains the hybrid tracking framework including the modified particle filtering scheme and the proposed local optimization algorithm. Section VI shows and discusses experimental results for several human motion sequences, and Section VII compares the performance of our framework with two other methods. Section VIII concludes the paper.

II. 3D Human Model

For a computationally efficient representation of the human body, we use a simple cylinder model similar to [27], which is shown in Fig. 1. The torso can be regarded as a degenerate cylinder since it has an elliptical cross-section. For each part except the torso, a local coordinate frame is defined with the origin at the base of the cylinder. These origins also correspond to the center of rotation of each body part. The global coordinate system originates at the center of the torso. The body parts and corresponding parameters are indexed from 0 to 9.

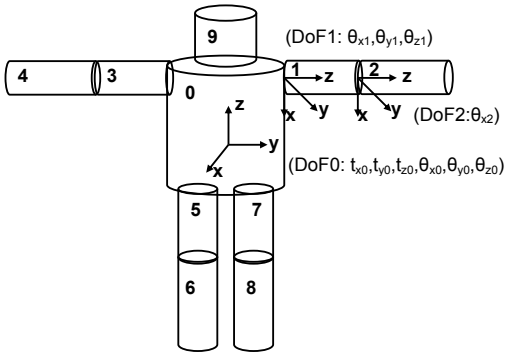


Fig. 1. 3D human model.

Human kinematic knowledge is employed as a prior to define the degrees of freedom (DoF) for our human model. We incorporate 25 DoF: 3 DoF for upper arms, legs and head (rotating about their X, Y and Z axes), 1 DoF for lower arms and legs (they are only allowed to rotate about their X axes), and 6 DoF for the torso (global translation and rotation). With these definitions, the entire 3D pose of the body is determined by a 25D pose vector $\mathbf{x} = (t_{0x}, t_{0y}, t_{0z}, \theta_{0x}, \theta_{0y}, \theta_{0z}, \theta_{1x}, \theta_{1y}, \theta_{1z}, \theta_{2x}, ...)^T$, which contains the joint angles of shoulders, elbows, hips, and knees, plus the global position and orientation of the torso.

To further constrain this high-dimensional solution space as well as to eliminate the ambiguity during tracking, two types of motion constraints are imposed:

1) ***Kinematic constraints:*** The connectivity between adjacent body parts as well as the length constancy of the body parts are enforced through kinematic constraints. Each body part is only allowed to move according to its DoF (e.g., the lower arms are only allowed to rotate about their X axes).
2) ***Joint angle limits:*** For real human motion, the joint angles between adjacent body parts are limited to a certain range (e.g., the elbow can only rotate by 135 degrees around its X axis). This constraint further reduces the solution space.

In Section V we show how our simulated physical force/moment based local optimization algorithm automatically incorporates the above constraints.

III. 3D Reconstruction of the Human Body

The inputs to our tracking framework are the sparsely reconstructed human surface points and surface normals, which can be obtained via a standard 3D reconstruction algorithm given multiple synchronized images and camera calibration parameters. Segmented human silhouettes can be computed by the foreground detection method [28] provided with the background statistics.

We adopt the well-known visual hull method as described in [29], [30] to reconstruct the 3D scene points as well as their surface normal vectors. These surface reconstruction points are obtained by intersecting the viewing cones from each view, and their corresponding normals are given by the cross product between the viewing lines and the tangent to the image silhouette.

IV. Probabilistic Model of Human Motion

Given a sequence of training motion data $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_T$ in temporal order, our goal is to model the low-dimensional statistical representation of the spatial-temporal structure of the human motion. PCA [31] is a very popular technique to deal with this dimensionality reduction problem by providing a sequence of best linear approximations to a given high-dimensional dataset. However, the human motion statistics are nonlinear and multimodal, which violates the basic PCA assumption of global linearity. Some manifold learning techniques have been developed to deal with the nonlinear dimensionality reduction problems (e.g., LLE [32], Laplacian eigenmap [33], ISOMAP [34]), but these involve the use of the original high-dimensional datasets. Therefore they are unable to deal with novel inputs and not ready for applications such as tracking.

In this work, we adopt the technique of VQPCA [24] by developing a time series extension of the basic VQPCA learning algorithm to model human motion. VQPCA is a non-parametric, nonlinear dimensionality reduction technique, which is suitable for modeling nonlinear data structures like human motion data and also provides the orthogonal projection basis flexible enough to handle novel inputs. The basic idea of VQPCA is to first partition the data space, in this case the motion space, into disjoint Voronoi regions using vector quantization and then perform local PCA about each cluster center. This technique is suitable for probabilistic modeling of human motion, where the whole motion sequence is partitioned into states $\{s|s \in (q_1, q_2, ...q_M)\}$, all pose vectors $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_T$ are considered as observations of these states, and the transition probabilities between states/regions are also defined. In VQPCA, each state naturally corresponds to a Voronoi region, and thus local PCA can be performed within each region, given that its statistical distribution can be approximated by a single Gaussian distribution. In parallel, state transition probabilities are modeled using hidden Markov models (HMM).

We use an expectation-maximization (EM) [35] framework to simultaneously partition the motion data, perform the subspace learning and estimate the transition probabilities as follows:

1) ***Initialization:***
All training data $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_T$ are partitioned into M Voronoi regions $R_1, R_2, ...R_M$, which correspond to M states $q_1, q_2, ...q_M$ of human motion. Each region R_i is modeled by a center $\mathbf{r}_i$, which is randomly selected from the training data, and a $n \times n$ covariance matrix Σ_i, which is initialized to be identity; n is the dimensionality of the pose vector $\mathbf{x}_t$. $\mathbf{e}_1^{(i)}, ...\mathbf{e}_n^{(i)}$ are n eigenvectors of Σ_i; the first m eigenvectors compose the linear basis $P_i = (\mathbf{e}_1^{(i)}, ...\mathbf{e}_m^{(i)})$, which projects the original n-dimensional vector $\mathbf{x}_t$ to its m-dimensional subspace $\mathbf{z}_t$. $\Lambda_i = P_i^T \Sigma_i P_i$ is the associated covariance matrix in the m-dimensional space. As in HMM, the state transition probabilities $\{a_{ij}|i, j = 1, 2, ...M\}$ are initialized to be equal, e.g., $1/M^2$. The choice of M is important in the training phase – if M is too small, the underlying non-Gaussian distribution would not be well approximated; if M is too large, there over-fitting could occur. Therefore, we try out several different values of M and choose the one that still produces a reasonably low average reconstruction error on the training data.
2) ***Expectation:***
As in the HMM, the forward probabilities $\{\alpha_t(j)|j = 1, 2, ...M\}$ and backward probabilities $\{\beta_t(j)|j = 1, 2, ...M\}$ are recursively updated as follows:

$$\alpha_t(j) = p(\mathbf{x}_{1:t}, s_t = q_j|\lambda) \quad (1)$$

$$= b_j(\mathbf{x}_t) \sum_{i=1}^{M} \alpha_{t-1}(i) a_{ij} \quad (2)$$

$$\alpha_1(j) = \pi_j b_j(\mathbf{x}_1) \quad (3)$$

$$\beta_t(i) = p(\mathbf{x}_{t+1:T}|s_t = q_i, \lambda) \quad (4)$$

$$= [\sum_{j=1}^{M} a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j)] \quad (5)$$

$$\beta_T(i) = 1 \quad (6)$$

Here $b_j(\mathbf{x}_t)$ is the likelihood of being in state q_j given the observation $\mathbf{x}_t$ at time instant t. In our work, we assume it to be a single Gaussian density, i.e., $b_j(\mathbf{x}_t) \sim$

$\phi(\mathbf{r}_j, \Sigma_j)$. π_j is the initial probability of state q_j, which is assumed to be equal for all states. $\mathbf{x}_{1:t}$ denotes the observation sequence of motion vector $(\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_t)$. λ represents the parameters of the HMM model, i.e., $\lambda = (a_{ij}, \pi_j, \mathbf{r}_j, \Sigma_j | i, j = 1, 2, ...M)$.

The probability of being in state j at time instant t is given by:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{p(\mathbf{x}_{1:T}|\lambda)} \tag{7}$$

Here $p(\mathbf{x}_{1:T}|\lambda)$ is the probability of observing the sequence $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_T$ given λ, which can be considered a normalization constant.

3) ***Maximization:***

All training data are partitioned into M regions $R_1, R_2, ...R_M$ according to their reconstruction distance $d(\mathbf{x}_t, \mathbf{r}_i)$:

$$R_i = \{\mathbf{x}_t | d(\mathbf{x}_t, \mathbf{r}_i) \leq d(\mathbf{x}_t, \mathbf{r}_j); \forall j \neq i\} \tag{8}$$

$$d(\mathbf{x}_t, \mathbf{r}_i) = \|\mathbf{x}_t - \mathbf{r}_i - \sum_{j=1}^{m} z_j \mathbf{e}_j^{(i)}\|^2 \tag{9}$$

$$= (\mathbf{x}_t - \mathbf{r}_i)^T T_i T_i^T (\mathbf{x}_t - \mathbf{r}_i) \tag{10}$$

Here T_i is composed of the trailing eigenvectors of Σ_i, i.e., $T_i = (\mathbf{e}_{m+1}^{(i)}, ...\mathbf{e}_n^{(i)})$.

The generalized centroid $\mathbf{r}_i$ of the region R_i is updated via minimizing the cost function:

$$\mathbf{r}_i = \arg\min_{\mathbf{r}} \frac{1}{N_i} \sum_{\mathbf{x}_t \in R_i} (\mathbf{x}_t - \mathbf{r}_i)^T T_i T_i^T (\mathbf{x}_t - \mathbf{r}_i) \tag{11}$$

There exist several solutions to the above equation; according to [24], a convenient choice is:

$$\mathbf{r}_i = \overline{\mathbf{x}_t} = \frac{1}{N_i} \sum_{\mathbf{x}_t \in R_i} \mathbf{x}_t \tag{12}$$

and

$$\Sigma_i = \frac{1}{N_i} \sum_{\mathbf{x}_t \in R_i} (\mathbf{x}_t - \mathbf{r}_i)(\mathbf{x}_t - \mathbf{r}_i)^T \tag{13}$$

The projection matrix P_i is then updated by the m eigenvectors corresponding to the m largest eigenvalues of the new covariance matrix Σ_i:

$$P_i = (\mathbf{e}_1^{(i)}, \mathbf{e}_2^{(i)}, ...\mathbf{e}_m^{(i)}) \tag{14}$$

Finally, the transition probabilities are updated as:

$$\xi_t(i,j) = p(s_t = q_i, s_{t+1} = q_j | \mathbf{x}_{1:T}, \lambda) \tag{15}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j)}{P(\mathbf{x}_{1:T}|\lambda)} \tag{16}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{17}$$

The above steps are iterated until there is no significant change in the average reconstruction error. We then get a set of Voronoi regions $(R_1, R_2, ...R_M)$ and their corresponding linear subspace basis $(P_1, P_2, ...P_M)$. To encode a novel input $\mathbf{x}$, we assign it to its corresponding region R_i according to the reconstruction distance, then project it to its local linear subspace:

$$\mathbf{z} = (z_1, z_2, ...z_m)^T = (\mathbf{e}_1^{(i)} \cdot (\mathbf{x} - \mathbf{r}_i), ..., \mathbf{e}_m^{(i)} \cdot (\mathbf{x} - \mathbf{r}_i))^T \tag{18}$$

and reconstruction is done as:

$$\widehat{\mathbf{x}} = \mathbf{r}_i + \sum_{j=1}^{m} z_j \mathbf{e}_j^{(i)} \tag{19}$$

V. Hybrid Human Motion Tracking Framework

A. Modified Particle Filtering

In the Bayesian framework, the task of human motion tracking can be formulated as inferring maximum a posterior (MAP) of the joint probability $p(\mathbf{x}_t, s_t | I_{1:t})$ given the image observation sequence $I_{1:T} = (I_1, I_2, ...I_t)$. Provided with the previous estimation of the density $p(\mathbf{x}_{t-1}, s_{t-1} | I_{1:t-1})$, inferring the posterior density of the current frame is therefore expressed as:

$$p(\mathbf{x}_t, s_t | I_{1:t}) = \kappa p(I_t | \mathbf{x}_t, s_t) \times \int p(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) \times p(\mathbf{x}_{t-1}, s_{t-1} | I_{1:t-1}) d\mathbf{x}_{t-1} ds_{t-1} \tag{20}$$

where κ is a normalization constant. $p(I_t | \mathbf{x}_t, s_t)$ is a likelihood term, which measures the probability of observing I_t given the motion state s_t and pose vector $\mathbf{x}_t$. The detailed definition of the likelihood function will be given in Section V-B. $p(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1})$ models the transition probability of the motion dynamics.

Although it would be impossible to develop an analytical expression of the posterior density, we can approximate it by a set of weighted pose and state vectors called *particles*. Each particle i is denoted as $(s_t^{(i)}, \mathbf{x}_t^{(i)}, \omega_t^{(i)})$, representing the motion state, pose vector and the associated likelihood weight. In a particle filter scheme, these weighted particles are propagated throughout consecutive frames via sequential importance sampling [36]. We define the importance density as:

$$q(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) = p(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) \tag{21}$$

Further factorization of this expression gives:

$$p(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) = p(s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) \times p(\mathbf{x}_t | s_t, \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) \tag{22}$$

The second term can be simplified as:

$$p(\mathbf{x}_t | s_t, \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) = p(\mathbf{x}_t | s_t, \mathbf{x}_{t-1}, s_{t-1}) \tag{23}$$

if we assume that the transition probability is independent of the observation sequence. Also, assuming first a order Markov chain, the first term is trivially equal to the state transition probability, i.e.,

$$p(s_t | \mathbf{x}_{t-1}, s_{t-1}, I_{1:t-1}) = p(s_t = q_j | s_{t-1} = q_i) = a_{ij} \tag{24}$$

Two important features distinguish our sampling and filtering scheme from the general particle filter algorithm.

First, special treatment is adopted when sampling from $p(\mathbf{x}_t|s_t, \mathbf{x}_{t-1}, s_{t-1})$ to achieve our idea of sampling in a low-dimensional space. For each selected particle i, we first locally project it into the linear subspace of its Voronoi region R_j with $(\mathbf{r}_j, \Sigma_j, P_j = (\mathbf{e}_1^{(j)}, \mathbf{e}_2^{(j)}, ...\mathbf{e}_m^{(j)}), \Lambda_j = P_j^T \Sigma_j P_j)$ (for notational simplicity, we ignore the region index j in the rest of the paper), i.e.,

$$\mathbf{z}_{t-1}^{(i)} = (\mathbf{e}_1 \cdot (\mathbf{x}_{t-1}^{(i)} - \mathbf{r}), \mathbf{e}_2 \cdot (\mathbf{x}_{t-1}^{(i)} - \mathbf{r}), ..., \mathbf{e}_m \cdot (\mathbf{x}_{t-1}^{(i)} - \mathbf{r}))^T \quad (25)$$

Then new samples are drawn from the low-dimensional space according to the Gaussian density $\mathbf{z}_t^{(i)} \sim (\mathbf{z}_{t-1}^{(i)}, \Lambda)$, where $\mathbf{z}_t^{(i)}$ is a m-dimensional vector ($m \ll n$) in the linear subspace of Voronoi region R. This sampling scheme drastically reduces the number of required particles according to the theory of particle filters [37], i.e., based on a lower bound of the number of required particles: $N \geq D_{min}/\alpha^d$, where N is the number of particles required, d is the dimension of the sampling space, D_{min}, α are constants and $\alpha \ll 1$. Obviously, for a low dimension d, much fewer samples are needed.

Later, to evaluate the likelihood, each low-dimensional sample $\mathbf{z}_t^{(i)}$ is synthesized into the original dimension according to its Voronoi region R, i.e.,

$$\widehat{\mathbf{x}_t^{(i)}} = \mathbf{r} + \sum_{j=1}^{m} z_{tj}^{(i)} \mathbf{e}_j \quad (26)$$

The second distinguishing feature is our sample-and-refine strategy, where each new particle is optimized to its nearby local peak of the posterior density after sampling by our proposed simulated physical force/moment based optimization scheme (see Section V-B below). This strategy further imrpoves the performance of our tracking algorithm.

The overall framework of our modified particle filter can be summarized as follows:

1) ***Objective:***
From a set of particles $\{s_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \omega_{t-1}^{(i)}, i = 1, 2...N\}$ at time step $t-1$, construct a new sample set $\{s_t^{(i)}, \mathbf{x}_t^{(i)}, \omega_t^{(i)}, i = 1, 2...N\}$ to represent the posterior density of the current frame.
2) ***Selection:*** Randomly draw a sample $(s_{t-1}^{\prime(i)}, \mathbf{x}_{t-1}^{\prime(i)}) \sim \{s_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \omega_{t-1}^{(i)}, i = 1, 2...N\}$, according to the weights $\{\omega_{t-1}^{(i)}, i = 1, 2...N\}$;
3) ***Sampling:***
For each particle $(s_{t-1}^{\prime(i)} = q_L, \mathbf{x}_{t-1}^{\prime(i)})$
   (a) Draw a new state according to the state transition probability, i.e., $s_t^{(i)} \sim \{a_{L1}, a_{L2}, ...a_{LM}\}$. The new state is denoted as: $s_t^{(i)} = q_C$, which corresponds to Voronoi region R with $(\mathbf{r}, \Sigma, P = (\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_m), \Lambda = P^T \Sigma P)$.
   (b) Project $\mathbf{x}_{t-1}^{\prime(i)}$ into the linear subspace of region R, i.e.,

$$\mathbf{z}_{t-1}^{\prime(i)} = (\mathbf{e}_1 \cdot (\mathbf{x}_{t-1}^{\prime(i)} - \mathbf{r}), \mathbf{e}_2 \cdot (\mathbf{x}_{t-1}^{\prime(i)} - \mathbf{r}), ..., \mathbf{e}_m \cdot (\mathbf{x}_{t-1}^{\prime(i)} - \mathbf{r}))^T \quad (27)$$

   (c) Draw a new low-dimensional sample according to the Gaussian density i.e., $\mathbf{z}_t^{(i)} \sim \phi(\mathbf{z}_{t-1}^{\prime(i)}, \Lambda)$;
   (d) Reconstruct $\widehat{\mathbf{x}_t^{(i)}}$ in the original dimension space, i.e.,

$$\widehat{\mathbf{x}_t^{(i)}} = \mathbf{r} + \sum_{j=1}^{m} z_{tj}^{(i)} \mathbf{e}_j \quad (28)$$

4) ***Refinement:***
For each particle $(s_t^{(i)} = q_C, \widehat{\mathbf{x}_t^{(i)}})$, perform the refinement by our simulated physical force/moment algorithm on $\mathbf{x}_t^{(i)}$.
5) ***Weighting:***
Evaluate the likelihood weight of each particle according to the likelihood function $\omega_t^{(i)} \propto p(I_t|\mathbf{x}_t^{(i)}, s_t^{(i)})$ proposed in Section V-B.
Now a new set of particles which approximate the posterior density of time t are obtained, i.e., $\{s_t^{(i)}, \mathbf{x}_t^{(i)}, \omega_t^{(i)}, i = 1, 2...N\}$. Using the idea of sampling in the linear subspace, we significantly reduce the number of particles required.

B. Local Optimization Using Simulated Physical Force/Moment

Given a set of 3D reconstruction points with surface normals for each frame and a 3D human model composed of a set of connected body parts, our likelihood function is designed as:

$$p(I_t|\mathbf{x}_t, s_t) = \kappa e^{-D/\sigma^2} \quad (29)$$

where κ is some normalization constant, σ is the variance. The distance term D is the average distance between 3D scene points $\mathbf{p}_i$ and the corresponding points on the 3D model $\mathbf{p}_i'$:

$$D = \frac{1}{S} \sum_i d(\mathbf{p}_i, \mathbf{p}_i') \quad (30)$$

S is the number of scene points, and

$$d(\mathbf{p}_i, \mathbf{p}_i') = \rho \left(\left(m_d(\mathbf{n}_{\mathbf{p}_i}, \mathbf{n}_{\mathbf{p}_i'}) \|\overrightarrow{\mathbf{p}_i \mathbf{p}_i'}\| \right)^2 \right) \quad (31)$$

Here $\|\overrightarrow{\mathbf{p}_i \mathbf{p}_i'}\|$ denotes the Euclidean distance between a scene point $\mathbf{p}_i$ and its corresponding point (i.e., the closest) on the model $\mathbf{p}_i'$. It is modulated by a factor $m_d(\mathbf{n}_{\mathbf{p}_i}, \mathbf{n}_{\mathbf{p}_i'})$ that considers the alignment of the surface normals $\mathbf{n}_{\mathbf{p}_i}$ and $\mathbf{n}_{\mathbf{p}_i'}$ between the model and the scene reconstruction: $m_d(\mathbf{n}_1, \mathbf{n}_2) = 1 - \epsilon \cos(\mathbf{n}_1, \mathbf{n}_2), 0 < \epsilon < 1$. The distance is further embedded into a robust function $\rho(x)$ (a truncated quadratic function) to suppress the effect of outliers.

As mentioned before, our proposed sampling scheme can significantly reduce the number of required particles. However, a small number of particles might not be dense enough to capture each local peak of the posterior distribution, which in turn degenerates the optimality of tracking, i.e., the global minimum is missed. Therefore, our local optimization task is to find a suitable refinement of each sampled pose vector $\mathbf{x}_t^{(i)}$ to minimize the distance term D. In other words, we move these particles onto their adjacent local peaks of the posterior distribution. Our local optimization method is based on the well-known iterative closest points (ICP) concept [38], which can coarsely align a model with scene points in an iterative manner.

While the original version of the ICP algorithm is only designed for rigid models, we introduced an articulated extension called *simulated physical force/moment based registration algorithm* [25] to deal with articulated 3D human model and scene fitting. Although this method is similar to the tracking method proposed in [11], there are several differences: our distance function is different as it uses a modulation term to account for the surface alignment and a robust function to get rid of the outliers; furthermore, we propose a simple and efficient hierarchical model pose updating scheme instead of recursively solving a set of dynamics equations.

Suppose a displacement between a scene point and its closest point on the model creates a simulated physical force. This force generates two effects on the model: translational velocity and angular moment to pull/rotate the model into alignment with the 3D scene point. Fig. 2 illustrates the force/moment created by the displacement between a scene point and its closest point on the model.

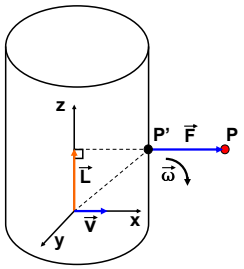


Fig. 2. The physical force F created by the displacement $P'P$; v and ω are corresponding translation and rotation vectors created by the physical force/moment.

The force can be expressed as:

$$\overrightarrow{F} = \rho(m_F(\mathbf{n}_{\mathbf{p}_i}, \mathbf{n}_{\mathbf{p}'_i})\|\mathbf{p}_i\mathbf{p}'_i\|)\overrightarrow{\mathbf{a}_{\mathbf{p}_i\mathbf{p}'_i}} \tag{32}$$

Here $\overrightarrow{\mathbf{a}_{\mathbf{p}_i\mathbf{p}'_i}}$ is the unit vector pointing to $\overrightarrow{\mathbf{p}_i\mathbf{p}'_i}$; $m_F(\mathbf{n}_{\mathbf{p}_i}, \mathbf{n}_{\mathbf{p}'_i})$ is a modulation factor that accounts for the surface alignment between the model and the scene: $m_F(\mathbf{n}_1, \mathbf{n}_2) = \cos(\mathbf{n}_1, \mathbf{n}_2)$.

The moment is given by:

$$\overrightarrow{M} = \overrightarrow{F} \times \overrightarrow{L} \tag{33}$$

where $\overrightarrow{L}$ is the vertical distance from the force $\overrightarrow{F}$ to the rotation center. The translation and rotation vectors generated are proportional to the physical force/moment:

$$\overrightarrow{v} = \rho\overrightarrow{F} \tag{34}$$

$$\overrightarrow{\omega} = \lambda\overrightarrow{M} \tag{35}$$

where ρ and λ are small coefficients.

As in the ICP, we iteratively compute the closest points and then update the model pose according to the estimated transform. During each iteration step, the displacements between all 3D scene points and the model are calculated, and all forces and moments are summed up, resulting in a translation and a rotation vector to align the model with the 3D scene points:

$$(\delta t_x, \delta t_x, \delta t_z)^T = \sum_i \overrightarrow{v_i} = \sum_i \rho\overrightarrow{F_i} \tag{36}$$

$$(\delta\theta_x, \delta\theta_x, \delta\theta_z)^T = \sum_i \overrightarrow{\omega_i} = \sum_i \lambda\overrightarrow{M_i} \tag{37}$$

Here $\overrightarrow{F_i}$ and $\overrightarrow{M_i}$ are the simulated physical force and moment created by the scene point p_i. With enough iterations, the misalignment between the model and the 3D scene points will be minimized, and the overall physical force/moment will be balanced, indicating convergence.

Given a set of articulated cylinder model parts, we start with assigning each scene point to its closest model part. However, instead of applying the above method to each body part independently, we adopt a hierarchical approach from our earlier work [25] for applying the transform to the human model, which is based on the following intuition: Suppose a physical force is applied to the right lower arm of the model; this force will not only create the angular moment for the right lower arm to rotate around the elbow, but will also contribute to the right upper arm's rotation about the shoulder, as well as the global rotation and translation of the torso. Our hierarchical updating approach is consistent with this observation. The human model will be treated as a hierarchical tree with its root at the torso, as illustrated in Fig. 3.

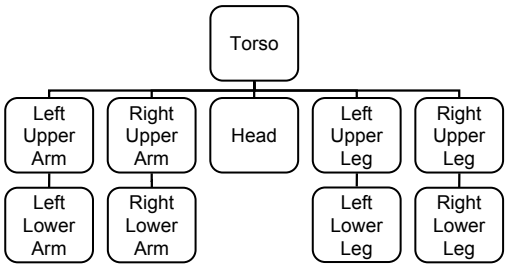


Fig. 3. Human model hierarchy tree.

When estimating the transform associated with a certain body part, the physical forces applied to all body parts in its group will be integrated. For example, when calculating the global translation and rotation $(\delta t_{0x}, \delta t_{0y}, \delta t_{0z}, \delta\theta_{0x}, \delta\theta_{0y}, \delta\theta_{0z})^T$, the forces applied to all body parts will be calculated as follows:

$$\begin{pmatrix} \delta t_{0x} \\ \delta t_{0y} \\ \delta t_{0z} \end{pmatrix} = \sum_j \sum_{i}^{\overrightarrow{F_i} \in part(j)} \lambda_{j0}\overrightarrow{F_i} \tag{38}$$

$$\begin{pmatrix} \delta\theta_{0x} \\ \delta\theta_{0y} \\ \delta\theta_{0z} \end{pmatrix} = \sum_j \sum_{i}^{\overrightarrow{M_i} \in part(j)} \rho_{j0}\overrightarrow{M_i} \tag{39}$$

where λ_{j0} and ρ_{j0} $(j = 0, 1, ...9)$ are weights, $part(j)$ denotes the j model part.

Similarly, when estimating the rotation $(\delta\theta_{1x}, \delta\theta_{1y}, \delta\theta_{1z})^T$ of the right upper arm about the right shoulder (there would be no translation for the right upper arm as defined by its DoF), the physical forces applied to the right upper arm and right lower arm will be counted:

$$\begin{pmatrix} \delta\theta_{1x} \\ \delta\theta_{1y} \\ \delta\theta_{1z} \end{pmatrix} = \sum_{i}^{\overrightarrow{M_i} \in part(1)} \rho_{11}\overrightarrow{M_i} + \sum_{i}^{\overrightarrow{M_i} \in part(2)} \rho_{21}\overrightarrow{M_i} \tag{40}$$

We further concatenate the transform vectors estimated for each body part to obtain the pose increment vector $\delta\mathbf{x} = (\delta t_{0x}, \delta t_{0y}, \delta t_{0z}, \delta\theta_{0x}, \delta\theta_{0y}, \delta\theta_{0z}, \delta\theta_{1x}, \delta\theta_{1y}, \delta\theta_{1z}, \delta\theta_{2x}...)^T$.

Obviously the degrees of freedom of each body part are preserved and the articulated structure of the human model is maintained implicitly in this updating scheme.

Furthermore, the kinematic constraints and joint angle limits can be incorporated in this framework automatically. Given the original pose vector $\mathbf{x}$ and the updating vector $\delta\mathbf{x}$ generated by our registration algorithm, we can clamp the new pose vector to avoid the violation of any constraint by the following inequality:

$$\mathbf{x}_{lb} \preceq \mathbf{x} + \delta\mathbf{x} \preceq \mathbf{x}_{ub} \tag{41}$$

where $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are the lower and upper bounds of the joint angles.

VI. Experimental Results

We validate the proposed framework using the real human motion database HumanEva [39]. This database contains human motion videos captured by seven synchronized digital cameras with a resolution of 640×480 pixels surrounding the scene. The ground truth of the human motion is also provided. We show results for three motion sequences here, namely walking, jogging, and boxing.

The size of training and test sets in terms of frames for each sequence are summarized in Table. I. In the training phase, the ground truth of the training set is used as input into our algorithm, and we output the partitions of all the Voronoi regions with the associated local projection basis. The input motion data are 27D while our algorithm learns a low-dimensional 3D representation.

TABLE I
Datasets used in our experiments.

Motion	Training size	Testing size
Walking	100 frames	433 frames
Jogging	100 frames	393 frames
Boxing	100 frames	377 frames

As mentioned in Section IV, we select the number of Voronoi regions M according to the criterion that ensures the average reconstruction error for the training data is below some threshold δ. The results for different values of M are shown in Fig. 4. It can be observed that at $M = 20$ the average reconstruction error is already quite low.

In the tracking phase, the inputs are synchronized frames from each of the 7 views. The 3D human surface reconstruction points as well as the corresponding surface normals are computed via the method described in Section III. This reconstruction is then input into our tracking algorithm. The output of the algorithm is the estimated body pose, i.e., global position, orientation and joint angles of our human model.

Some examples of tracking results are shown in Fig. 5. It can be observed that the estimated human pose closely matches the ground truth.

Fig. 6 compares the estimated joint angles with the ground truth for all frames in the three sequences. Despite the obvious differences in motion regularity, the estimated joint angles accurately follow the ground truth data.

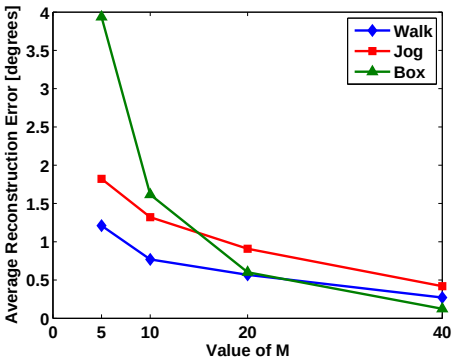


Fig. 4. Average reconstruction error vs. number of Voronoi regions.

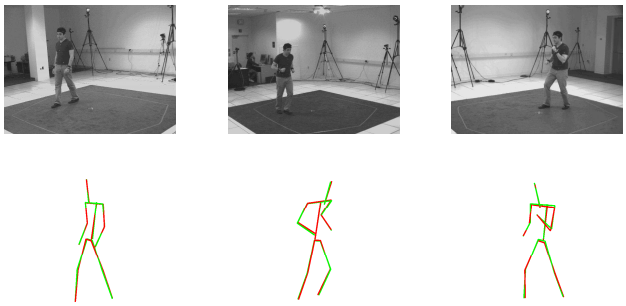


Fig. 5. Examples of the tracking results for a frame from each motion sequence. The top row shows one of the captured views; the bottom row shows the corresponding tracking results. The estimated human pose is shown in green, the ground truth pose in red.

Fig. 7 shows the root mean squared error (RMSE) of joint angles and joint positions averaged over all frames of each sequence. The increase of errors in the jogging and boxing sequences is mainly due to the more difficult poses – in those sequences, the arms are sometimes closely coupled with the torso, which makes the reconstruction more noisy and introduces more tracking errors.

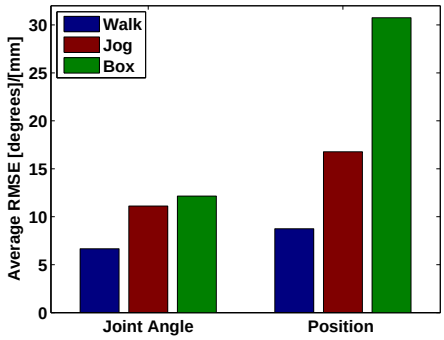
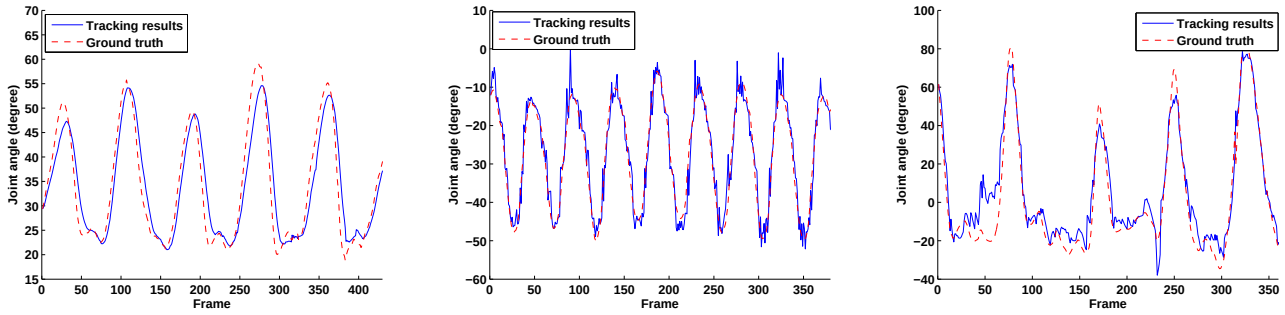


Fig. 7. Average RMSE of joint angle and position for each sequence.

We also investigate the influence of the number of Voronoi regions on tracking performance by varying the value of M



(a) Left elbow, walking (b) Left knee, jogging (c) Right shoulder, boxing

Fig. 6. Comparison between estimated joint angles and ground truth.

from 5 to 80. The result is shown in Fig. 8. As expected, the average RMSE decreases with increasing M – when more regions are used, the underlying distribution of human motion statistics is more accurately approximated, which reduces the tracking errors. For small M, the error reduction is large, but beyond $M = 20$ the curves flatten and the gains become insignificant. A good choice of M seems to be in the range of 10 to 20.

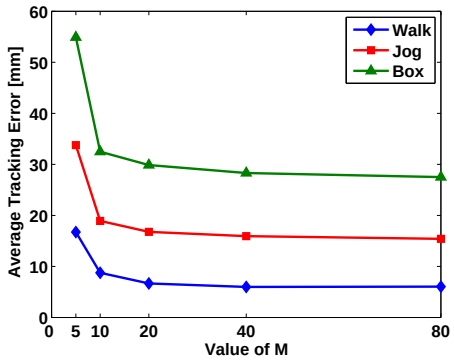


Fig. 8. Average RMSE vs. number of Voronoi regions.

VII. Comparisons and Discussions

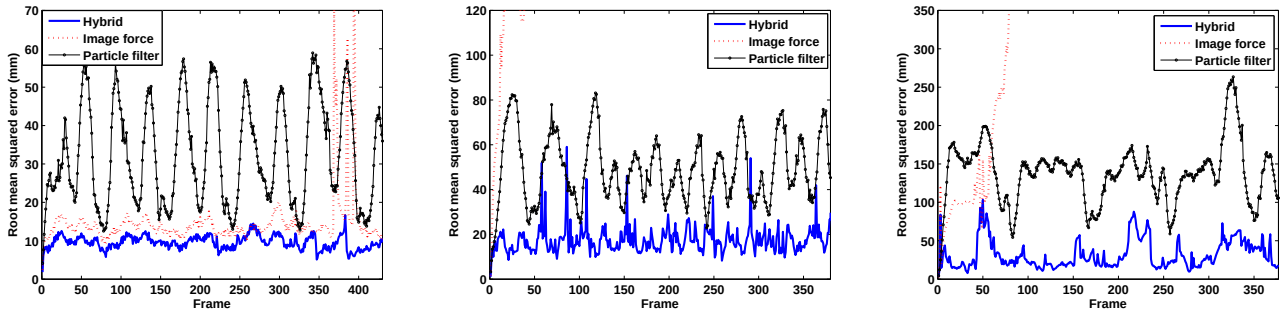
In this section, we compare the performance of our algorithm in terms of tracking accuracy, robustness, and efficiency with a deterministic image force-based method as well as a stochastic sampling based method (particle filter). The reason for selecting these two methods for comparison is that they are state-of-the-art methods in their categories. The two methods are implemented on the same MATLAB platform and evaluated using the same motion sequences (walking, jogging and boxing) as our hybrid method; likewise, the system inputs are the same 3D reconstructions as our method. The implementation details of the different methods are as follows:

- Our proposed hybrid tracking method with 40 particles and $M = 20$. Sampling is performed on the reduced low-dimensional space (3D).
- An image force/ICP based method is similar to [11] (i.e., local optimization based method).
- A particle filter based tracking algorithm is similar to [14], [15]. Sampling is performed in the original high-dimensional space (33D). The likelihood function is defined according to Eq. (29). 1000 particles are used.

The results are shown in Fig. 9, where we compare the RMSE of the joint positions estimated by different methods on the test sequences. It is clear that our hybrid method consistently achieves higher tracking accuracy throughout all test sequences, while the image force based method and particle filter based method generally have higher errors. The image force based method performs well in easy cases such as walking; however, in cases where the body parts have sharp joint angles, e.g., jogging and boxing, it gives unreliable results. For the particle filter based method, 1000 particles are still not adequate to track the mode of the posterior density, which leads to large errors. To achieve good results, many more particles would be required for this method, perhaps 10^4. In contrast, our method takes the advantages of both sampling and local optimization to approximate the MAP while using fewer particles.

In terms of tracking robustness, our hybrid method and the particle filter based algorithm track the entire sequences successfully, while the image force based algorithm quickly loses track in the boxing and jogging sequences. This is mainly due to the issue of error accumulation, which is a theoretic limitation of the deterministic searching method. Fortunately, the usage of sampling in our method successfully eliminates this effect.

The comparisons of tracking efficiency are summarized in Table II. The particle filter based method is very slow in practice due to the large number of particles, and still its tracking accuracy is only mediocre. The image force based method is fast, but usually gives unreliable results, e.g., tracking is lost in the jogging and boxing sequences. Our method achieves a good compromise between both efficiency and accuracy, thanks to the scheme of sampling in the low-dimensional space and the combination of both sampling and optimization. The only limitation is the need for explicit training with similar motion patterns.



(a) Walking (b) Jogging (c) Boxing

Fig. 9. RMSE comparison of the joint positions.

TABLE II
SUMMARY OF THE TRACKING PERFORMANCE OF DIFFERENT METHODS.

Motion		Walking	Jogging	Boxing
No. of particles	Hybrid	40	40	40
	Image force	1	1	1
	Particle filter	1000	1000	1000
Time per frame (s)	Hybrid	15	14	17
	Image force	3	2	2
	Particle filter	250	256	271
RMSE (mm)	Hybrid	8.57	16.77	30.75
	Image force	13.56	666.56	587.75
	Particle filter	32.45	48.82	137.69

VIII. CONCLUSIONS

Tracking full body human motion is a challenging task given its high dimensionality. Current tracking methods suffer from either a robustness problem or inefficiency. In this paper we proposed a novel tracking framework which achieves a good compromise between accuracy, robustness and efficiency. Prior information about human motion statistics is encoded into a compact representation by a subspace learning algorithm (VQPCA), which performs sampling in a low-dimensional space, thus reducing the number of particles. We also introduced a sample-and-refine framework, which combines the concept of both particle filtering and simulated physical force based registration. This new framework further improves tracking robustness and accuracy.

Quantitative experimental results on several real motion sequences show the high accuracy achieved by our method, while comparisons demonstrate the robustness of our method as well as a much higher sampling efficiency compared to other methods. A limitation is that our method is not able to deal with unfamiliar motions since a training set is always needed.

REFERENCES

[1] S. Park and J. Aggarwal, "A hierarchical Bayesian network for event recognition of human actions and interactions," *Multimedia Systems*, vol. 10, no. 2, pp. 164–179, August 2004.

[2] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[3] D. Ramanan, D. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 65–81, January 2007.

[4] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[5] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[6] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: A multi-view approach," in *Proc. International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, June 1996, pp. 73–80.

[7] M. Yamamoto, A. Sato, S. Kawada, T. Kondo, and Y. Osaki, "Incremental tracking of human actions from multiple views," in *Proc. International Conference on Computer Vision and Pattern Recognition*, 1998, pp. 2–7.

[8] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *Proc. International Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, 1998, pp. 8–15.

[9] R. Kehl and L. V. Gool, "Markerless tracking of complex human motions from multiple views," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 190–209, November 2006.

[10] Q. Delamarre and O. Faugeras, "3D articulated models and multi-view tracking with silhouettes," in *Proc. International Conference on Computer Vision*, vol. 2, Corfu, Greece, September 1999, pp. 716–721.

[11] ——, "3D articulated models and multi-view tracking with physical forces," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 328–357, March 2001.

[12] L. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1453–1459, 2000.

[13] 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.

[14] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2000, pp. 126–133.

[15] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel, "Learning for multi-view 3D tracking in the context of particle filters," in *Proc. Second International Symposium on Advances in Visual Computing*, Lake Tahoe, NV, USA, November 2006, pp. 59–69.

[16] M. Lee, I. Cohen, and S. Jung, "Particle filter with analytical inference for human body tracking," in *Proc. Workshop on Motion and Video Computing*, 2002, pp. 159–165.

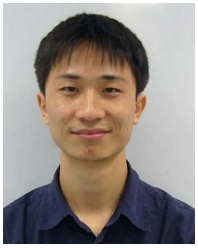
[17] B. Stenger, P. Mendonca, and R. Cipolla, "Model-based 3D tracking of an articulated hand," in *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 02, Kauaii, USA, 2001, pp. 310–318.

[18] J. Ziegler, K. Nickel, and R. Stiefelhagen, "Tracking of the articulated upper body on multi-view stereo image sequences," in *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 1, New York, NY, USA, July 2006, pp. 774–781.

[19] T. Han and T. Huang, "Articulated body tracking using dynamic belief propagation," in *Proc. IEEE International Workshop on Human-computer Interaction*, Beijing, China, October 2005, pp. 26–35.

[20] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic Markov network," in *Proc. International Conference on Computer Vision*, vol. 2, Nice, France, October 2003, pp. 1094–1101.

[21] H. Sidenbladh, F. Torre, and M. Black, "A framework for modeling the appearance of 3D articulated figures," in *Proc. International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000, pp. 368–375.
[22] H. Lim, O. I. Camps, M. Sznaier, and V. I. Morariu, "Dynamic appearance modeling for human tracking," in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 751–757.
[23] H. Sidenbladh, M. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *Proc. European Conference on Computer Vision*, vol. 1, Copenhagen, Denmark, May 2002, pp. 784–800.
[24] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, 1997.
[25] B. Ni, S. Winkler, and A. Kassim, "Articulated object registration using simulated physical force/moment for 3D human motion tracking," in *Proc. 2nd Workshop on Human Motion Understanding, Modeling, Capture and Animation in conjunction with ICCV 2007*, 2007, pp. 212–224.
[26] C. Sminchisescu and B. Triggs, "Covariance scaled sampling for monocular 3D body tracking," *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 01, pp. 447–454, 2001.
[27] H. Sidenbladh, M. Black, and D. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *Proc. European Conference on Computer Vision*, 2000, pp. 702–718.
[28] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 02, Fort Collins, CO, USA, 1999, pp. 246–252.
[29] J. Franco and E. Boyer, "Exact polyhedral visual hulls," in *Proc. British Machine Vision Conference*, 2003, pp. 329–338.
[30] M. Niskanen, E. Boyer, and R. Horaud, "Articulated motion capture from 3-D points and normals," in *Proc. British Machine Vision Conference*, vol. 1, 2005, pp. 439–448.
[31] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.
[32] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[33] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
[34] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[35] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
[36] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
[37] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. European Conference on Computer Vision*, vol. 2, Dublin, Ireland, June 2000, pp. 3–19.
[38] P. Besl and H. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
[39] L. Sigal and M. J. Black, "HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion," Brown University, Tech. Rep. CS-06-08, 2006.



Bingbing Ni received his B.Eng. degree in Electrical Engineering from Shanghai Jiao Tong University (SJTU), China in 2005. He is currently a Ph.D. candidate in Electrical and Computer Engineering at the National University of Singapore (NUS). His research interests are in the areas of computer vision and machine learning.

Ashraf A. Kassim received his B.Eng. (First Class Honors) and M.Eng. degrees in Electrical Engineering from the National University of Singapore (NUS) in 1985 and 1987, respectively. From 1986 to 1988, he worked on machine vision systems at Texas Instruments. He went on to obtain his Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, in 1993. Since 1993, he has been with the Electrical and Computer Engineering Department at NUS, where he is currently an Associate Professor and Vice Dean of Engineering Faculty. Dr Kassim's research interests include image analysis, machine vision, video/image processing and compression.

Stefan Winkler holds an M.Sc. degree in Electrical Engineering from the University of Technology in Vienna, Austria, and a Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Dr. Winkler is currently Principal Technologist for Symmetricom's QoE Assurance Division. Prior to that, he was Chief Scientist of Genista Corporation, which he co-founded in 2001. He has also held assistant professor positions at the National University of Singapore (NUS) and the University of Lausanne, Switzerland. Dr. Winkler has published more than 50 papers and is the author of the book "Digital Video Quality." His interests include visual perception, media quality, computer vision, and human-computer interaction.