

## Research Article

# Articulated Human Motion Tracking Using Sequential Immune Genetic Algorithm

**Yi Li and Zhengxing Sun**

*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China*

Correspondence should be addressed to Zhengxing Sun; [szx@nju.edu.cn](mailto:szx@nju.edu.cn)

Received 5 October 2012; Accepted 3 December 2012

Academic Editor: Baozhen Yao

Copyright © 2013 Y. Li and Z. Sun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We formulate human motion tracking as a high-dimensional constrained optimization problem. A novel generative method is proposed for human motion tracking in the framework of evolutionary computation. The main contribution is that we introduce immune genetic algorithm (IGA) for pose optimization in latent space of human motion. Firstly, we perform human motion analysis in the learnt latent space of human motion. As the latent space is low dimensional and contains the prior knowledge of human motion, it makes pose analysis more efficient and accurate. Then, in the search strategy, we apply IGA for pose optimization. Compared with genetic algorithm and other evolutionary methods, its main advantage is the ability to use the prior knowledge of human motion. We design an IGA-based method to estimate human pose from static images for initialization of motion tracking. And we propose a sequential IGA (S-IGA) algorithm for motion tracking by incorporating the temporal continuity information into the traditional IGA. Experimental results on different videos of different motion types show that our IGA-based pose estimation method can be used for initialization of motion tracking. The S-IGA-based motion tracking method can achieve accurate and stable tracking of 3D human motion.

## 1. Introduction

Tracking articulated 3D human motion from video is an important problem in computer vision which has many potential applications, such as virtual character animation, human computer interface, intelligent visual surveillance, and biometrics. Despite having been attacked by many researchers, this challenging problem is still long standing because of the difficulties conducted mainly by the complicated nature of 3D human motion, self-occlusions, and high-dimensional search space.

In the previous work, two main classes of motion tracking approaches can be identified: discriminative approaches and generative approaches [1]. Discriminative methods attempt to learn a direct mapping from image features to 3D pose using training data. The mapping is often approximated using nearest neighbor [2], regression models [3] or mixture of regressors [4]. Discriminative approaches are effective and fast. However, they need a large training database and are limited to fixed classes of motion. Moreover, the inherent one-to-many mapping from 2D images to 3D poses is difficult

to learn accurately. In contrast, generative methods exploit the fact that although the mapping from visual features to poses is complex and multimodal, the reverse mapping is often well posed. Therefore, pose recovery is tackled by optimizing an object function that encodes the pose-feature correspondence [5] or by sampling posterior pose probabilities [6]. Compared with discriminative methods, generative methods are usually more accurate. However, generative methods are generally computationally expensive because one has to perform complex search over the high-dimensional pose state space in order to locate the peaks of the observation likelihood. Moreover, prediction model and initialization are also the bottlenecks of the approach in the tracking scenario. In this work, we focus on recovering 3D human pose within the generative framework.

In general, high-dimensional state space and search strategy are two main problems in generative approaches. High-dimensional pose state space makes pose analysis computationally expensive or even infeasible. Despite the high dimensionality of the configuration space, many human motion activities lie intrinsically on low-dimensional latent

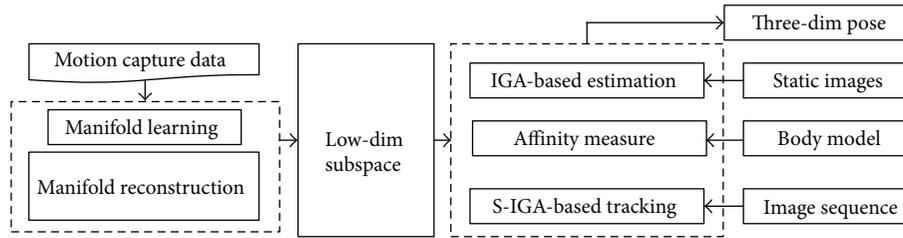


FIGURE 1: The framework of our approach.

space [7, 8]. Motivated by this observation, we use ISOMAP, a nonlinear dimensionality reduction method, to learn the low-dimensional latent space of pose state, by which the aim of both reducing dimensionality and extracting the prior knowledge of human motion are achieved simultaneously. On the other hand, search strategy, in general how to track in the low-dimensional latent space, is another important problem. The search strategy should suit for the characteristics of the subspace and be global, optimal, and convergent. Although considerable work has already been done, a more effective search strategy is still intensively needed for robust visual tracking. In our opinion, motion prior knowledge has great influence on the search strategy, which can aid in performing more stable tracking. Compared with the previous methods, extracting the prior knowledge and introducing it in the designing of search strategy are of particular interests to us.

In this paper, we propose a novel generative approach in the framework of evolutionary computation, by which we try to widen the bottlenecks mentioned above with effective search strategy embedded in the extracted state subspace. The framework of our approach is illustrated in Figure 1. Firstly, we use ISOMAP to learn this latent space. Then we propose a manifold reconstruction method to establish the inverse mapping, which enables pose analysis in this latent space. As the latent space is low dimensional and contains the prior knowledge of human motion, it makes pose analysis more efficient and accurate. In the search strategy we introduce immune genetic algorithm (IGA) for pose optimization. Details of the implementations, such as encoding and initialization, computation of affinity, and genetic and immunity operators, are designed. We propose an IGA-based method for pose estimation, which can be used for initialization of motion tracking. In order to make IGA suitable for human motion tracking, a sequential IGA (S-IGA) framework is proposed by incorporating the temporal continuity information into the traditional IGA. Experimental results on different motion types and different image sequences demonstrate our methods.

The rest of the paper is organized as follows. Section 2 gives an introduction to the related works. Section 3 gives a description of how the latent space is learnt. In Section 4, we give a detailed description of how we apply IGA for pose optimization in the latent space. We then show how to apply IGA-based pose optimization algorithm for pose estimation and tracking in Section 5. Section 6 contains experimental results and comparison with other tracking algorithms. The conclusions and possible extension for future work are given in Section 7.

## 2. Related Works

There has been a great deal of prior works on human motion analysis from video [1, 8, 9]. Here we focus our survey on the most related research on generative methods. In generative human motion tracking methods, the high-dimensional pose state space is the most significant problem. There are several possible strategies for reducing the dimensionality of the configuration space, including using motion models [10], hierarchical search [11], and dimensionality reduction [5, 12]. Motion models are often derived from training data of a single class of movement. Although they can aid in performing more stable tracking, this comes at the cost of putting a strong restriction on the poses that can be recovered. Another way to constrain the configuration space is to perform a hierarchical search. For example, John et al. [11] proposed a hierarchical particle swarm optimization method to search the best pose hierarchically. An inherent problem with this approach is the need to estimate accurately the position and orientation of the initial body segment (typically the torso), as a wrong pose estimate for the initial segment can distort the pose estimates for subsequent limbs. Nowadays, dimensionality reduction has become the most widely used methods. For example, Urtasun et al. [12] construct a differentiable objective function based on the Principle Component Analysis (PCA) of motion capture data and then find the poses of all frames simultaneously by optimizing a function in low-dimensional space. However, this method needs many example sequences of data to perform PCA, and all of these sequences must keep the same length and same phase by interpolating and aligning. Zhao and Liu [5] use PCA to learn the low-dimensional state space of human pose and perform pose analysis in the latent space. However since the mapping between the original pose space and the latent space is in general nonlinear, linear PCA is inadequate. Nonlinear dimensionality reduction methods have also been used. For example, Sminchisescu and Jepson [13] use spectral embedding to learn the embedding which is modeled as a Gaussian mixture model. Radial Basis Functions (RBFs) are learned for inverse mapping. A linear dynamical model is used for tracking. Elgammal and Lee [14] learn view-based representations of activity manifolds using nonlinear dimensionality reduction method (LLE). Then, the nonlinear mappings from the embedding space into both visual input space and 3D pose space are learnt using the generalized radial basis function. Although nonlinear dimensionality reduction methods can learn this nonlinear mapping, they are not invertible. The smooth inverse mapping is still a not

well-solved problem. In this paper, we use ISOMAP [15], a nonlinear dimensionality reduction method, to learn the low dimensionality subspace of a specific activity. And then, based on the intrinsic executive mechanism of ISOMAP, a manifold reconstruction method is proposed to generate smooth mappings between the subspace and the original space. This enables us to perform human motion tracking in the learned subspace.

Search strategy is another key research problem of pose tracking in the generative framework. They are typically tackled using either deterministic methods or stochastic methods. Deterministic methods usually involve a gradient descent search to minimize a cost function [16]. Although these methods are usually computationally efficient, they easily become trapped in local minima. In contrast, stochastic methods introduce some stochastic factors into the searching process in order to have a higher probability of reaching the global optimum of the cost function. Particle filter [6] is the most widely studied stochastic method which is based on Monte Carlo sampling. Although, in theory, particle filter is very suitable for tracking, it needs a large number of particles to approximate the posterior distributions, and it tends to suffer from sample impoverishment, so that the final particle sets cannot represent the true distributions. Therefore, many improvements on the traditional particle filter have been proposed. For example, Deutscher et al. [17] introduced the annealed particle filter which combines a deterministic annealing approach with stochastic sampling to reduce the number of samples required. At each time step the particle set is refined through a series of annealing cycles with decreasing temperature to approximate the local maxima in the fitness function. In Krzeszowski et al. [18], a particle swarm optimization algorithm is utilized in the particle filter to shift the particles toward more promising configurations of the human model. Compared with the deterministic counterparts, stochastic methods are usually more robust, but they suffer a large computational load, especially in high-dimensional state space. In recent years, evolutionary computing methods, such as genetic algorithm [5] and particle swarm optimization [11, 18, 19], have received increasing attention. For example, Zhao and Liu [5] proposed an annealed genetic algorithm to track human motion in compact base space, where the base space is learned using PCA. John et al. [11] proposed a hierarchical particle swarm optimization (HPSO) algorithm for articulated human tracking. Their comparative experimental results show that HPSO is more accurate than particle filter. However, based on our experimental results, due to the high-dimensional pose state space and imperfect image observations, HPSO may deviate from the pose state space and result in inaccurate tracking. Evolutionary algorithms are all good searching algorithms with an iterative process of generation and test. Two operators, crossover and mutation, give each individual the chance of optimization and ensure the evolutionary tendency with the select mechanism of survival of the fittest. However, the two operators change individuals randomly and indirectly under some conditions. Therefore, they not only give individuals the evolutionary chance but also cause certain degeneracy. Recently, immune algorithms have been another

hotspot succeeding genetic algorithm and particle swarm optimization for its success in solving pattern recognition and optimization problems. Its main advantage, compared with GA and PSO, is it has the ability to use the prior knowledge of problem by vaccination and immune selection [20]. In this paper, we apply immune genetic algorithm (IGA) [20], a novel immune method, for pose optimization. We propose an IGA-based method for pose estimation from monocular images. In order to make IGA suitable for pose tracking, we propose a sequential IGA (S-IGA) algorithm by incorporating the temporal continuity information into the traditional IGA. To the best of our knowledge, the proposed algorithm is new in the human motion tracking literature.

### 3. Learning the Latent Space of Human Motion

Tracking in a low-dimensional latent space requires three components [8]. First, a mapping between original pose space and low-dimensional subspace must be learned. Second, an inverse mapping must be defined. Third, how tracking within the low-dimensional space occurs must be defined. In this section, we first learn the low-dimensional subspace using ISOMAP [15]. Then, we propose a manifold reconstruction method to establish the mappings between high- and low-dimensional states.

*3.1. ISOMAP-Based Latent Space Learning.* We describe the human body as a kinematic tree consisting of rigid limbs that are linked by joints. Every joint contains a number of degrees of freedom (DOF), indicating in how many directions the joint can move. All DOF in the body model together form the pose representation. In this paper, the pose is described by a 66D vector  $x = \{x_r, x_k\}$ , where 3D vector  $x_r$  represents the root joint rotations and 63D vector  $x_k$  represents the body joints rotations. Apart from the kinematic structure, the human shape is also modeled. Each rigid limb of the body is fleshed out using conic sections with elliptical cross-sections (see Figure 2). Human shape will be used to compute the likelihood function (see Section 4.2).

Since the mapping between the original pose space and latent space is in general nonlinear, linear PCA is inadequate. So we use ISOMAP to learn the nonlinear mapping. We extract the subspace using motion capture data obtained from the CMU database [21].

As for a special activity, such as walking, running, jumping, and so forth, the original pose state space has no relation with the global motion. Different from the previous methods of learning different manifolds for the same activity (such as walking) of different views, we filter out the rotations of root joint ( $x_r$ ) and represent the pose using the rotations of body joints ( $x_k$ ) only. Assuming  $\{x_i \mid x_i \in X, i = 1, \dots, l\}$  is a given sequence of motion capture data corresponding to one motion type, where  $x_i = (x_k)_i$ ,  $i$  is the frame index,  $l$  is the number of total frames, and  $X$  is the original pose state space, the subspace  $Y$  is extracted by ISOMAP as follows.

- (1) *Construct Neighborhood Graph.* Define the graph  $G$  over all data points (in our method the data point is one frame in motion sequence) by connecting point

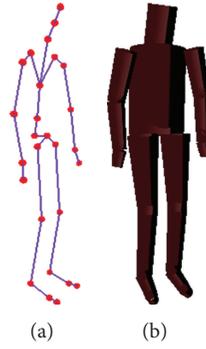


FIGURE 2: (a) The 3D human skeleton model. (b) The shape model.

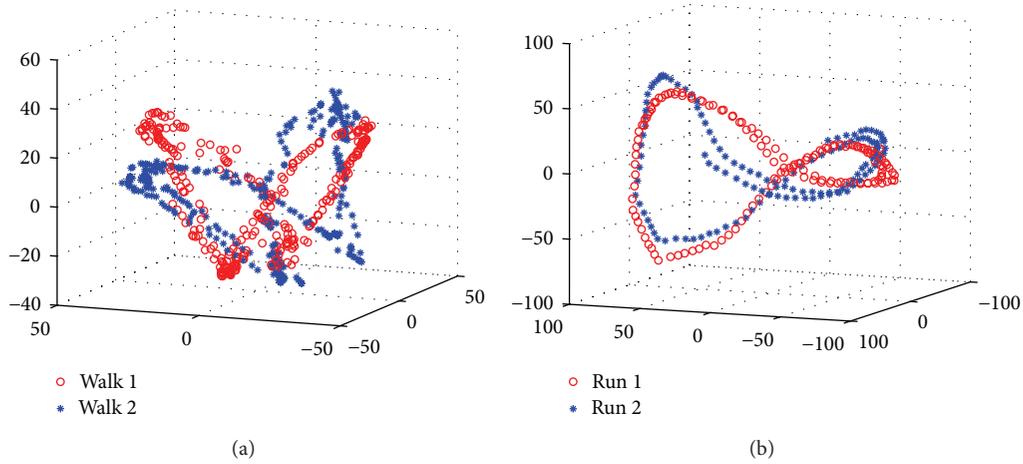


FIGURE 3: ISOMAP-based dimensionality reduction results. (a), (b) are manifolds of two sequences of walking and running in 3D subspace, respectively.

$x_i$  and  $x_j$  if  $d_x(x_i, x_j) < \varepsilon$ . Set edge length to be  $d_x(x_i, x_j)$ . Moreover,

$$d_x(x_i, x_j) = \frac{\sum_{m=1}^M \|(x_i)_m - (x_j)_m\|}{M}, \quad (1)$$

where  $M$  is the dimensionality of pose state space;  $M = 63$  here.

- (2) *Compute Shortest Paths.* Initialize  $d_G(x_i, x_j) = d_x(x_i, x_j)$  if  $x_i$  and  $x_j$  are linked by an edge;  $d_G(x_i, x_j) = \infty$  otherwise. Then for each value of  $i = 1, 2, \dots, l$  in turn, replace all entries  $d_G(x_i, x_j)$  by  $\min\{d_G(x_i, x_j), d_G(x_i, x_t) + d_G(x_t, x_j)\}$ . The matrix of final values  $D_G = \{d_G(x_i, x_j)\}$  will contain the shortest path distances between all pairs of points in  $G$ .
- (3) *Construct  $d$ -Dimensional Embedding.* Let  $\lambda_p$  be the  $p$ th eigenvalue (in decreasing order) of the matrix  $\tau(D_G)$  and  $v_p^i$  the  $i$ th component of the  $p$ th eigenvector. Then set the  $p$ th component of the  $d$ -dimensional coordinate vector  $y_i$  to be equal to  $\sqrt{\lambda_p} v_p^i$ .

The subspace  $Y$  learned by ISOMAP is shown in Figure 3. Actually, similar low-dimensional subspace can be

extracted from the training sequences that belong to the same type of motions but performed by different subjects. And the training sequences corresponding to different types of motions produce different subspaces. For example, experiments demonstrate that different walking sequences generate similar manifolds in the 3D subspace, which is different from that of running motion.

ISOMAP cannot only reduce the dimensionality of high-dimensional input space, but also find meaningful low-dim structures hidden behind their high-dim observations. In doing so, infeasible solutions, namely, the absurd poses, can be avoided naturally during optimization, which will make pose tracking in this subspace more efficient and accurate.

*3.2. Mapping between High- and Low-Dimensional States.* Traditional ISOMAP can only learn the mapping from the original pose space to the latent space but not the inverse mapping. However, in order to track human motion in the low-dimensional manifold, the inverse mapping is required. Based on the intrinsic executive mechanism of ISOMAP, we proposed an ISOMAP-based manifold reconstruction method to establish the mapping between high- and low-dimensional states.

**Input:** The training data set  $\{x_i \mid x_i \in X, i = 1, \dots, l\}$ .

**Output:** The mappings  $g : X \rightarrow Y, y = g(x)$  and  $f : Y \rightarrow X, x = f(y)$ .

Step 1: (preparing)

(1) Using the ISOMAP algorithm to compute the low-dim vector  $\{y_i \mid y_i \in Y, i = 1, \dots, l\}$  for the original input vector  $\{x_i \mid x_i \in X, i = 1, \dots, l\}$ ;

(2) Construct the following matrixes:

$X_i = (x_{i_1} - x_i, \dots, x_{i_{l_i}} - x_i) \in R^{n \times l_i}, Y_i = (y_{i_1} - y_i, \dots, y_{i_{l_i}} - y_i) \in R^{d \times l_i}$ , where  $\{x_{i_j} \mid j = 1, \dots, l_i\}$  is the  $\varepsilon$ -neighbor of  $x_i$ .

(3) Compute  $Q_i = X_i Y_i^T (Y_i Y_i^T)^G \in R^{n \times d}$ , where  $(Y_i Y_i^T)^G$  is the generalized inverse matrix of  $Y_i Y_i^T$ .

Step 2: (manifold reconstruction)

(1) Mapping from original space to latent space:  $g : X \rightarrow Y, y = g(x)$ .

Given a high-dim pose vector  $x_0$ , the corresponding low-dim vector  $y$  can be computed as:

(1.1) Find the nearest neighbor of  $x_0$  in  $\{x_i \mid i = 1, \dots, l\}$ , set it to be  $x_s$ ;

(1.2) The low-dim vector correspondence to  $x_0$  can be formulated as:

$$y = \tilde{g}(x_0) = y_s + Q_s^T (x_0 - x_s);$$

(2) Mapping from latent space to original space:  $f : Y \rightarrow X, x = f(y)$ .

Given a low-dim pose vector  $y_0$ , the corresponding high-dim vector  $x$  can be computed as:

(2.1) Find the nearest neighbor of  $y_0$  in  $\{y_i \mid i = 1, \dots, l\}$ , represented as  $y_s$ ;

(2.2) The high-dim vector correspondence to  $y_0$  can be formulated as:

$$x = \tilde{f}(y_0) = x_s + Q_s (y_0 - y_s).$$

ALGORITHM 1: ISOMAP-based manifold reconstruction.

Suppose the pose state space to be  $X \subset R^n$  and the low-dim state space to be  $Y \subset R^d$ . Denote the mapping as  $f : Y \rightarrow X, x = f(y)$  and  $g : X \rightarrow Y, y = g(x)$ , where  $x, y$  are the high- and low-dimensional vectors, respectively. The set of input instances is  $\{x_i \mid x_i \in X, i = 1, \dots, l\}$ , and their corresponding points in the embedding space learned by ISOMAP are  $\{y_i \mid y_i \in Y, i = 1, \dots, l\}$ . Assume  $\{x_{i_j} \mid j = 1, \dots, l_i\}$  are the  $\varepsilon$ -neighbors of point  $x_i$ , where  $l_i$  is the number of neighbors. And their corresponding points in the embedding space are  $\{y_{i_j} \mid j = 1, \dots, l_i\}$ . Then our ISOMAP-based manifold reconstruction method can be described as in Algorithm 1.

Using the ISOMAP-based manifold reconstruction method, we can generate smooth mapping between the original pose space and the latent space, which enables us to track human pose in the latent space. In the following section, we will show how tracking within the latent space occurs.

## 4. Immune Genetic Algorithm for Pose Optimization

We formulate pose estimation as a constrained optimization problem and solve it using immune genetic algorithm. In this section, we first give a brief introduction to IGA. Then, we design an IGA-based method for pose optimization.

**4.1. Immune Genetic Algorithm.** In IGA, the idea of immunity is mainly realized through two steps based on reasonably selecting vaccines, that is, a vaccination and an immune selection, of which the former is used for raising fitness and the latter is for preventing the deterioration. A very clear overview of IGA, from immunology and engineering points of view, is presented in [20].

**4.1.1. Immunological Terms.** In order to describe the IGA clearly, some immunological terms will be given first [22].

**Antigen.** In immunology, an antigen is any substance that causes immune system to produce antibodies against it. In this paper, IGA is used for optimization:

$$\text{Minimize } f(x), \quad (2)$$

where  $x = [x_1, x_2, \dots, x_D] \in F$ ,  $F$  is the feasible region,  $D$  is the number of problematic parameter, and the antigen is defined as the objective function  $f(x)$ .

**Antibody and Antibody Population.** In this paper, an antibody is a representation of a candidate solution of an antigen. The antibody  $\vec{a} = [a_1, a_2, \dots, a_D]$  is the coding of variable  $x$ , denoted by  $a = e(x)$ , and  $x$  is called the decoding of antibody  $a$ , expressed as  $x = e^{-1}(a)$ . The representation of antibody  $a$  varies with antigen and can be binary string, real number sequence, symbolic sequence, and characteristic sequence. In this study, we adopt real-coded representation, that is,  $a = e(x) = x$ .

An antibody population,

$$A = \{a_1, a_2, \dots, a_N\}, \quad a \in F, \quad 1 \leq i \leq N, \quad (3)$$

is an  $N$ -dimensional group of antibody  $a$ , where the positive integer  $N$  is the size of antibody population  $A$ .

**Affinity.** In immunology, affinity is the fitness measurement for an antibody. For the optimization problem, the affinity,  $\text{Affinity}(a) \geq 0$ , is a mapping of the objective function  $f(x)$  for a given antibody  $a$ .

**4.1.2. Description of IGA.** In this paper, we use the IGA for optimization task. The flow chart of IGA is shown in Figure 4.

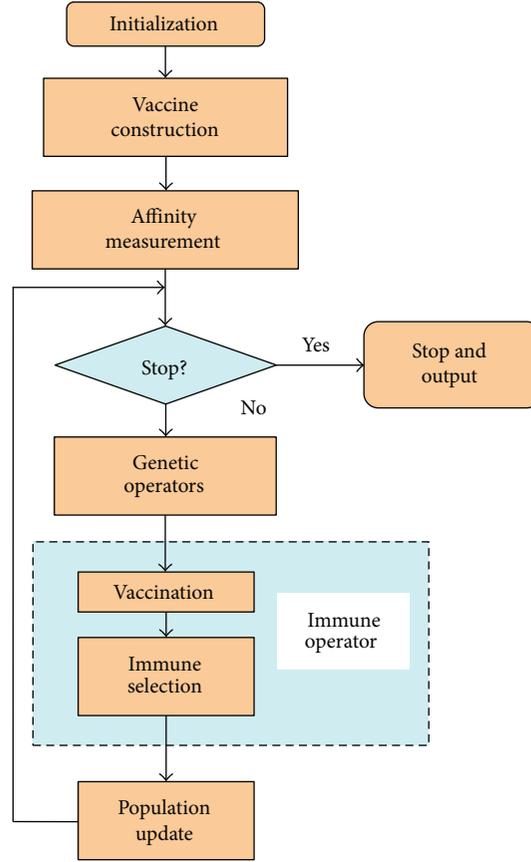


FIGURE 4: Flow chart of IGA for pose optimization.

The main steps of our modified IGA can be summarized as follows.

*Step 1.* Initialization: randomly generate the initial antibody population  $A(k)$ ; set  $k = 0$ .

*Step 2.* Vaccine construction: abstract vaccines according to the prior knowledge.

*Step 3.* Evaluation: calculate the affinities of all antibodies in  $A(k)$ .

*Step 4.* Termination test: if termination criterion is satisfied, export the antibody having the highest affinity in  $A(k)$  as the output of the algorithm and stop the algorithm; otherwise, continues.

*Step 5.* Genetic operators: perform genetic operators on the  $k$ th parent  $A(k)$  and obtain the results  $B(k)$ .

*Step 6.* Vaccination: perform vaccination on  $B(k)$  and obtain the results  $C(k)$ .

*Step 7.* Immune selection: perform immune selection on  $C(k)$  and obtain the next parent  $A(k + 1)$ , then go to Step 3.

In general, the IGA algorithm is to be implemented as the following evolution process:

$$A(k) \xrightarrow{T_g^p} B(k) \xrightarrow{T_v^p} C(k) \xrightarrow{T_s^p} A(k+1), \quad (4)$$

where  $A(k)$ ,  $B(k)$ , and  $C(k)$  are the antibody populations during different periods in a single evolution generation,  $k$  is the iterative step.  $T_g^p$ ,  $T_v^p$ , and  $T_s^p$  are the genetic, vaccination, and immune selection operators, respectively.

*4.2. Apply IGA for Pose Optimization.* In this section, we apply IGA for human pose optimization. Some details of our implementations are discussed below.

*4.2.1. Encoding and Initialization.* In IGA, each antibody represents a potential solution in the search space. For our problem, we perform human motion analysis in the latent space. So an antibody is corresponding to a pose vector in the latent space. In this paper, we represent the full 3D pose vector as  $x = \{x_r, y\}$ , where 3D vector  $x_r = (r_x, r_y, r_z)$  represents the root joint rotations,  $y = (y_1, \dots, y_d)$  corresponds to the pose vector in latent space; we set  $d = 6$  here. So  $x$  is a 9-dimensional vector. We use real encodings. We represent the

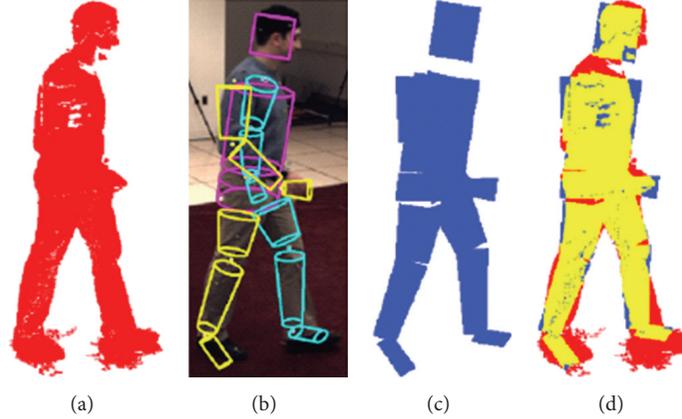


FIGURE 5: Silhouette-based affinity measurement, a bidirectional likelihood version [23].

TABLE 1: The genetic operators in IGA.

Operator	Example
Exchange	$y = (y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow y' = (y_1, y_6, y_3, y_4, y_5, y_2)$
Segment reversion	$y = (y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow y' = (y_1, y_6, y_5, y_4, y_3, y_2)$
Segment shift	$y = (y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow y' = (y_1, y_6, y_2, y_3, y_4, y_5)$
Point mutation	$y = (y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow y' = (y_1, y_2, y'_3, y_4, y_5, y_6)$
Segment mutation	$y = (y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow y' = (y_1, y'_2, y'_3, y'_4, y'_5, y'_6)$

antibody population as  $A = \{x_1, x_2, \dots, x_N\}$ , where  $N$  is the size of population.

**4.2.2. Computation of Affinity.** For each antibody, an affinity measure needs to be computed to estimate how well a given antibody (pose) matches the observed images. Here we use the bidirectional likelihood proposed by [23]. Let  $M^b$  represent the binary silhouette map for the body model and  $M^f$  the image foreground. We seek to minimize the non-overlapping regions, red and blue, therefore maximizing the Yellow region (see Figure 5). The size of each region can be computed by summing over all image pixels  $p$  using

$$\begin{aligned}
 R &= \sum_p (M^f(p)(1 - M^b(p))), \\
 B &= \sum_p (M^b(p)(1 - M^f(p))), \\
 Y &= \sum_p (M^f(p)M^b(p)).
 \end{aligned} \tag{5}$$

Then the objective function of candidate pose  $x$  with regard to image  $z$  can be calculated as

$$f(x) = (1 - \gamma) \frac{B}{B + Y} + \gamma \frac{R}{R + Y}, \tag{6}$$

where  $\gamma$  is the weight. We set  $\gamma = 0.5$  in this paper.

Affinity is the fitness measurement for an antibody. As defined above, the affinity,  $\text{Affinity}(x)$ , for a given antibody  $x$

is a linear mapping of the objective function  $f(x)$ . Therefore, we define the affinity of antibody  $x$  as

$$\begin{aligned}
 \text{Affinity}(x) &= S^* \exp(-f(x)) \\
 &= S^* \exp\left(-\left((1 - \gamma) \frac{B}{B + Y} + \gamma \frac{R}{R + Y}\right)\right),
 \end{aligned} \tag{7}$$

where  $S$  is a positive constant; we set  $S = 100$  in this paper.

**4.2.3. Genetic Operators.** We design five genetic operators, which are executed orderly in IGA. We introduce the operators by evolving an example antibody  $x = (x_r, y) = (x_r, y_1, y_2, y_3, y_4, y_5, y_6)$ . The new antibody generated by the operators is denoted as  $x' = (x'_r, y')$ . Assuming the positions generated randomly are numbers 2 and 6 or 3 (for point mutation operator) of  $y = (y_1, y_2, y_3, y_4, y_5, y_6)$ , for example, the five operators are illustrated in Table 1. The application order of the genetic operators in the algorithm is just as that listed in Table 1.

The genetic operators were represented as  $T_g^p$ . We perform genetic operator on the  $k$ th parent  $A(k)$  and obtain the results  $B(k)$ .

**4.2.4. Vaccine Construction and Immune Operators.** Genetic operators give each antibody the chance of optimization and ensure the evolutionary tendency with the select mechanism of survival of the fittest. However, it changes individuals randomly and indirectly under some conditions. Therefore, they not only give individuals the evolutionary chance but also cause certain degeneracy. In IGA, the idea of immunity is mainly realized through two steps based on reasonably

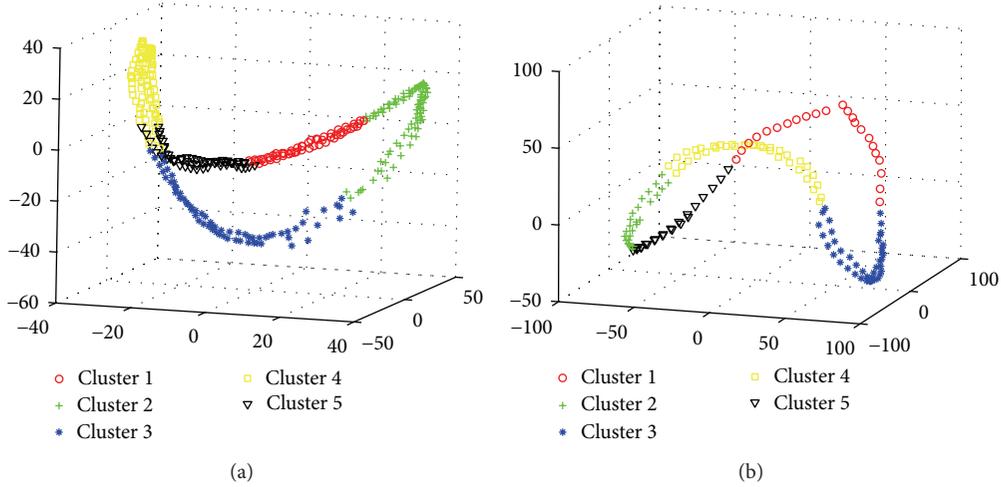


FIGURE 6: Clustering of human pose in 3D subspace, where (a), (b) represent the results of walking and running data, respectively.

selecting vaccines, that is, a vaccination and an immune selection, of which the former is used for raising fitness and the latter is for preventing the deterioration.

In this section, we extract the prior knowledge of human motion and construct two vaccines. Then we design the vaccination and immune selection operators.

(1) *Vaccine Construction.* A vaccine is abstracted from the prior knowledge of the pending problem. Human pose in subspace is located on a manifold structure but not the whole subspace. Actually, pose subspace is a compact space. We constrained the subspace of human motion and construct two vaccines for our human pose estimation problem.

(i) *Vaccine 1.* Every dimensionality  $y_i$  of subspace pose  $y = (y_1, y_2, \dots, y_d)$  should be distributed in a scope as  $\min(y_i) < y_i < \max(y_i)$ , where the bound  $\max(y_i)$  and  $\min(y_i)$  are learned from the motion training data.

(ii) *Vaccine 2.* Vaccine 2 is motivated by the fact that every generated pose should locate on the manifold. Based on the consistency of human motion, we partition the manifolds into different subparts with  $k$ -means clustering, where the number of class  $c$  is 5 in this paper (see Figure 6). For each class  $c_i$ ,  $i = 1, \dots, 5$ , we assume the poses in it is of Gaussian distribution, described as follows:

$$p_i(Y) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} e^{(-1/2)(Y-\mu_i)^T \Sigma_i^{-1} (Y-\mu_i)}, \quad (8)$$

$$i = 1, \dots, 5,$$

where,  $\mu_i$  is the mean vector,  $\Sigma_i$  is the covariance matrix,  $d = 6$  is the dimensionality of the pose subspace. Then the vaccine 2 can be described as for all  $y \in Y$ ,  $\exists i$ , such that  $p_i(y) > \chi$ , where  $i = 1, \dots, 5$ .

(2) *Vaccination.* A vaccination means the course of modifying the genes of an individual  $x = (x_r, y)$  on some bits in accordance with prior knowledge so as to gain higher fitness with greater probability. For an antibody  $x' = (x'_r, y')$ ,  $x' \in B(k)$  generated using genetic operators, we perform vaccination operator on  $x'$  to generate a new antibody  $x'' = (x''_r, y'')$ .

*Inoculation of Vaccine 1.* Vaccine 1 indicates that every dimensionality  $y'_i$  of subspace pose  $y' = (y'_1, y'_2, \dots, y'_6)$  should be distributed in a scope. When it moves out of this scope, we set it to be the border. The process can be formulated as

$$\begin{aligned} & \text{if } (y'_i < \min(y_i)), \\ & y'_i = \min(y_i); \quad \text{if } (y'_i > \max(y_i)), \\ & y'_i = \max(y_i), \quad \text{where } i = 1, \dots, 6. \end{aligned} \quad (9)$$

*Inoculation of Vaccine 2.* Vaccine 2 indicates that every pose  $y' = (y'_1, y'_2, \dots, y'_d)$  should locate on the manifold. If a pose  $y'$  does not locate on the manifold, that is,  $p_i(y') < \chi$ , for  $i = 1, \dots, 5$ , we first calculate to which class it is most likely to belong, suppose it to be  $c_i$ . Then, we set  $y'$  to be a random antibody in this class.

The vaccination operator was represented as  $T_v^p$ . We perform vaccination operator on the  $k$ th parent  $B(k)$  and obtain the results  $C(k)$ .

(3) *Immune Selection.* This operation is accomplished by the following three components. The first one is the immune test, that is, testing the antibodies. If the affinity is better than that of the parent, we add it to a temporal population  $D(k)$ ,  $D(k) = \{x'' \mid \text{Affinity}(x'') > \text{Affinity}(x)\}$ . The second one is the annealing selection, that is, if the affinity is worse than

**Input:** total number of antibodies  $N$ , maximum number of generations  $K$ .

**Steps:**

(1) **Initialization:** generate the initial population with  $N$  antibodies. The initial population can be represented as  $A(0) = \{x_1(0), x_2(0), \dots, x_N(0)\}$ .

(2) **Repeat:**

**While** ( $k < K$ ) **do**

(2.1) Genetic operator: Perform genetic operators  $T_g^p$  on the  $k$ th parent  $A(k)$  and obtain the results  $B(k)$ ;

(2.2) Vaccination: Perform vaccination  $T_v^p$  on  $B(k)$  and obtain  $C(k)$ ;

(2.3) Immune selection: Perform immune selection  $T_s^p$  on  $C(k)$  and obtain the next generation population  $A(k+1)$ ;

(2.4)  $k = k + 1$ .

**End while**

(3) **Output:** output the population  $A(K) = \{x_1(K), x_2(K), \dots, x_N(K)\}$ .

ALGORITHM 2: IGA-based pose optimization algorithm.

that of the parent, select an individual in the present population  $C(k)$  to join in the temporal population  $D(k)$  with the probability as follows:

$$P(x'') = \frac{\text{Affinity}(x'') * e^{\text{Affinity}(x'')/T_k}}{\sum_{i=1}^N \text{Affinity}(x''_i) * e^{\text{Affinity}(x''_i)/T_k}}, \quad (10)$$

where  $\text{Affinity}(x'')$  is the affinity of the individual  $x''$  and  $T_k$  is the temperature controlled series tending towards 0. The third one is the next population generation. We design a hybrid  $(\mu + \lambda)$  evolutionary strategy to generate the new generation  $A(k+1)$ .  $(\mu + \lambda)$  evolutionary strategy means selecting the first  $N$  individuals from the current population  $A(k)$  (with the size of  $N$ ) and temporal population  $D(k)$  (with the size of  $M$ ) to compose a new parent population  $A(k+1)$ .

**4.3. IGA-Based Human Pose Optimization.** Based on the description above, the IGA-based pose optimization algorithm can be described as in Algorithm 2.

We will show how to apply IGA-based pose optimization algorithm for pose estimation and tracking in the next section.

## 5. Sequential Immune Genetic Algorithm for Pose Tracking

In tracking applications, the data is typically a time sequence, and hence the task is essentially a dynamic optimization problem which distinguishes it from traditional optimization problems. In tracking situation, the previous estimation results can be used to cut the current search space. From the Bayes' view, we can formulate the pose tracking problem as

$$p(x_t | z_t) \propto p(z_t | x_t) p(x_t | x_{t-1}), \quad (11)$$

where  $\{x_t | t = 1, 2, \dots, T\}$  and  $\{z_t | t = 1, 2, \dots, T\}$  represent temporal states and observations, respectively. How to

determine the conditional distribution  $p(x_t | x_{t-1})$  effectively is the core problem for 3D human pose tracking.

In this paper, we proposed a sequential IGA- (S-IGA)-based framework for human motion tracking. The flowchart of the S-IGA framework is shown in Figure 7. First, we perform human pose estimation on the first frame of the video as initialization for tracking. Then, the previous converged antibodies at time  $t$  are randomly propagated as initial antibodies for the next time (frame)  $t+1$ . Finally, we perform IGA-based pose optimization on current antibodies. The individual with best affinity is used to approximate the tracking result of time  $t+1$ , and the converged antibodies are used to initial the next frame. There are three major stages in the S-IGA framework: automatically initialization, next frame propagation, and IGA-based optimization.

**5.1. Pose Estimation for Initialization of Motion Tracking.** Initialization is an important problem of human motion tracking. How to begin the tracking process from a good starting point sometimes is an intractable problem. We achieve the automatic initialization by determining the pose of the first frame using the IGA-based human pose estimation algorithm, which can be described as follows.

Pose estimation is the process to estimate articulated human pose from a single image which can be formulated as an optimization process. We apply IGA for pose estimation. For clarity, we redefine the full 3D pose vector as  $x = \{x_r, y\}$ , where  $x_r$  is the global motion of human body with respect to the camera and  $y$  is the pose vector in state subspace. We perform the state posterior inference by optimizing the affinity function. The optimal pose can be represented as

$$x = \arg \max_x (\text{Affinity}(x)). \quad (12)$$

We maximize the search efficiency by embedding the global search capability of IGA into the local conditions of state subspace.

The global motion of human body is very important for its visual appearance in an image and is also critical in disambiguating the left-right confusion. Determining this motion

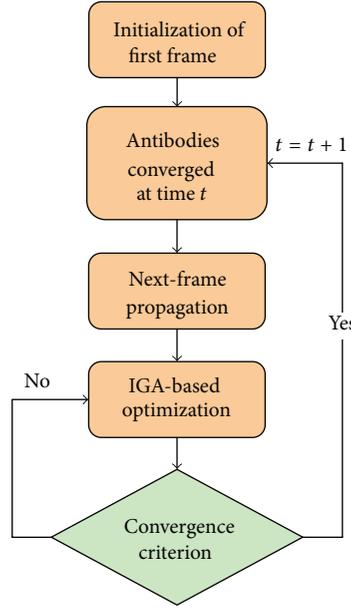


FIGURE 7: Overview of the sequential IGA.

accurately makes our method viewpoint invariant. With the aim of both cutting the search space and determining the motion direction roughly, we incorporated the global motion process step [5] into the framework of IGA. The global motion process can be summarized as follows. (1) In state vector  $x = \{x_r, y\}$ , the global motion  $x_r = (r_x, r_y, r_z)$  include the rotation of the full body about the coordinate axes  $X$ ,  $Y$ , and  $Z$ , respectively. In the first round of state evolution ( $k = 1$ ), we only actually search the optimal solutions of global motion. Other state components ( $y$ ) are taken as one of the clustering centers  $y_{\tilde{c}_i}$ ,  $i = 1, \dots, 5$ , randomly. The variance domain  $\min(x_r)$  and  $\max(x_r)$  of  $x_r$  is computed by storing the  $N'$  best antibodies.  $N'$  is determined empirically according to the threshold value of affinity. (2) In the rest rounds of state evolution, the antibody is evolved normally as described in Section 4. In doing so, we can get the coarse scopes of global motion in the first round of state evolution, and the fine tuning of these parameters can be achieved in the followed evolution rounds.

Based on the proposed IGA pose optimization algorithm, the antibody with the highest affinity in population  $A(K) = \{x_1(K), x_2(K), \dots, x_N(K)\}$  will be selected to be the optimal pose. Figure 8 is the process of pose estimation, where (a) is one frame of input video, (b) is the initialized poses, (c), and (d) are results with 10, 40 times of iteration, respectively. We can see that the poses generated by our initialization method can cover the whole walking pose state space, and the poses become convergent with times of iteration increase.

**5.2. Next-Frame Propagation.** Next-frame propagation is the key stage in the S-IGA framework which aims to find the dynamic model  $p(x_t | x_{t-1})$ . In this paper, we design a randomly propagation method. The randomly propagation method is actually a first-order Gauss-Markov dynamical model. Given the converged antibodies  $x_{i,t-1}(K)$  at frame  $t-1$ ,

the antibodies in the next frame  $t$  are initialized by sampling a Gaussian distribution centered in the current best antibodies. Consider,

$$x_{i,t}(0) \sim p(x_t | x_{t-1}) = N(x_{i,t-1}(K), \Sigma), \quad (13)$$

where  $x_{i,t}(0)$  are the initial antibodies at time  $t$ ,  $x_{i,t-1}(K)$  are the converged antibodies at time  $t-1$ ,  $i = 1, \dots, N$ , and  $\Sigma$  is the covariance matrix of Gaussian distribution. Low value  $\Sigma$  will promote temporal consistency but is likely to lose the diversity. We set it empirically according to the motion type and speed. S-IGA propagates only a minimal amount of information between frames and does not incorporate any motion model. Although randomly propagation is simple, it is sufficient because it is only used to produce an initial value for a subsequent search for the optimal state.

We do not incorporate any learnt constant motion model here, which is motivated by two considerations.

- (1) **Generality:** many prior motion models are derived from training data. A possible weakness of these motion models is that the ability to accurately represent the space of realizable human movements generally depends significantly on the amount of available training data. This comes as a cost of putting a strong restriction on the poses that can be recovered. Therefore, we do not use any constant learnt motion models here.
- (2) **The effectiveness of our IGA pose optimization algorithm,** which can explore efficiently large portions of the search space starting from the initial distribution of antibodies.

Actually, the S-IGA framework is a “sample-and-refine” search strategy. Firstly, the initial antibodies are sampled for the transition distribution as  $x_{i,t}(0) \sim N(x_{i,t-1}(K), \Sigma)$ . Then

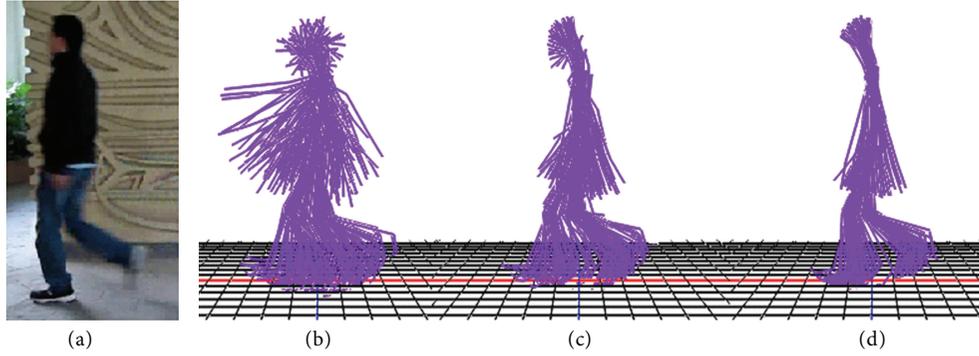


FIGURE 8: The process of human pose estimation, where (a) is a frame of input video, (b) is the initialized poses, and (c), (d) are results with different times of iteration, respectively.

- (1) **Initialization:** perform human pose estimation on the first frame of the video, output the converged antibodies  $\{x_{1,t}(K), x_{2,t}(K), \dots, x_{N,t}(K)\}$ , where  $t = 0$ ;
- (2) **for**  $t = 1 : T$  **do**
- (3) Next-frame propagation: randomly propagate the antibodies to enhance their diversities according to the following transition model:  

$$x_{i,t}(0) \sim N(x_{i,t-1}(K), \Sigma), i = 1, \dots, N;$$
- (4) IGA-based pose optimization: using Algorithm 2 to optimize the initial antibodies:  

$$\{x_{1,t}(0), x_{2,t}(0), \dots, x_{N,t}(0)\};$$
- (5) Check the convergence criterion: if satisfied, the converged antibodies are used to initial the next frame;
- (6) The individual with best affinity in population  $\{x_{1,t}(K), x_{2,t}(K), \dots, x_{N,t}(K)\}$  is used to approximate the tracking result of time  $t$ ;
- (7) end for.

ALGORITHM 3: S-IGA-based motion tracking algorithm.

the antibodies are updated according to the newest observations in each IGA iteration. Through the IGA iteration, the antibodies are moved towards the region where the likelihood of observation has larger values and are finally relocated to the dominant modes of the likelihood. And in a Bayesian inference view, the IGA iterations are essentially a multi-layer importance sampling strategy which incorporates the new observations into a sampling stage and thus avoids the sample impoverishment problem suffered by the particle filter [6].

*5.3. Sequential Immune Genetic Algorithm-Based Pose Tracking.* Based on the designing above, we can formulate our sequential IGA for pose tracking as in Algorithm 3.

## 6. Experimental Results

### 6.1. Experimental Data and Evaluation Measures

*Experimental Data.* The data for latent space training is from CMU Database [21]. We quantitatively evaluate our method on synthesized image sequences as in [3]. We also give experimental results on real image sequences from [24], CMU Database [21], and HumanEva [23].

*Evaluation Measures.* In this paper, we use the evaluation measures proposed in [23]. The average error over all joint angles (in degrees) is defined as

$$D(x, \hat{x}) = \frac{\sum_{m=1}^M \|x_m - \hat{x}_m\|}{M}, \quad (14)$$

where  $x = (x_1, x_2, \dots, x_M)$  and  $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_M)$  are the ground truth pose and the estimated pose date, respectively. For the sequence of  $T$  frames, the average performance and the standard deviation of the performance are computed using the following:

$$\mu_{\text{seq}} = \frac{1}{T} \sum_{t=1}^T D(x_t, \hat{x}_t), \quad (15)$$

$$\sigma_{\text{seq}} = \sqrt{\frac{1}{T} \sum_{t=1}^T [D(x_t, \hat{x}_t) - \mu_{\text{seq}}]^2}.$$

*6.2. The Convergence of IGA.* It is understood that the number of antibodies  $N$  and iteration times  $K$  will affect the

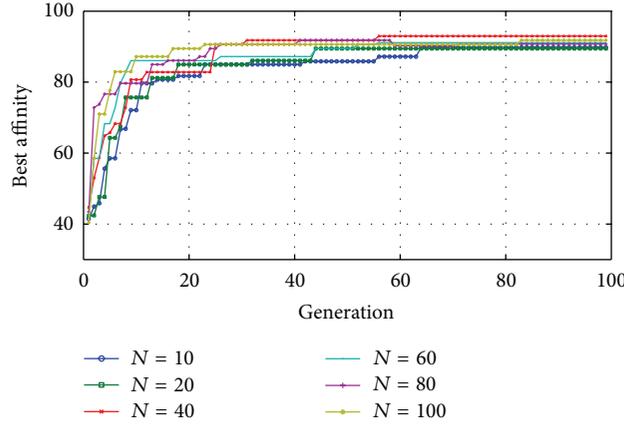


FIGURE 9: The convergence process.

convergence. We take pose estimation experiment on a single image and report the affinities of the best antibody during iteration. Figure 9 demonstrates the convergence process. Different lines represent different numbers of antibodies used. The  $x$ -axis is the times of iteration while the  $y$ -axis is the affinity value. As shown in Figure 9, the affinities will converge as the times of iteration increase. The experimental results demonstrate that our IGA-based pose optimization is convergent.

We have ascertained experimentally that higher numbers of  $N$  and  $K$  will achieve better results. However, in order to deal with the tradeoff of computational time and accuracy, we set  $N = 40$ ,  $K = 60$ .

**6.3. IGA-Based Pose Estimation Results.** We test our IGA-based pose estimation method on three image sequences, including one straight walk sequence [24], one turning walk sequence [23], and one run sequence [21]. The purpose is to test the capability of the method to cope with limb occlusion, left-right ambiguity, and view-point problems, which are the main challenges that a pose estimation method has to deal with. As mentioned in Section 3, we first learn the subspace of walking and running. To extract the motion subspace of walking, a data set consisting of motion capture data of a single subject was used. The total number of frame is 316. For running subspace learning, a data set with 186 frames was used. It was found that the different subjects and different frame numbers can produce generally identical subspace. So the learned subspaces are also used in the tracking experiments.

For pose estimation on a single image, the parameters of IGA are set as  $N = 40$  and  $K = 60$  to deal with the trade-off of computational time and accuracy. We test our IGA-based pose estimation method on 100 frames of images for all three types of motions, and the mean errors of joint angle are reported, which are shown in Figure 10. From Figure 10 we can see that, except for some particular joints, the mean errors of most joints for three sequences are less than 5 degrees. The mean errors of some joint angles are larger than

others because they have wider range of variation or less observability related to 2D image features. Our results are competitive with others reported in the related literatures.

Table 2 shows the ground truth and estimated values of some joint angles in an example frame. Three values in each cell are the rotation angles of the joints around  $X$ ,  $Y$ , and  $Z$  axes, respectively. The values come from a frame on the level of average error. Actually, other frames show generally the similar comparison results. From Table 2 we can see that estimated joint angles are close to the ground truth data. The experiment results demonstrate that our IGA-based pose estimation method is effective to analyze articulated human pose from a single image.

The results on real images are shown in Figure 11. From the above experiment results, we can see that, on most of the frames, the occlusion and left-right confusion problems are tackled by searching the optimal pose in the extracted subspace because the prior knowledge about motions is contained in this subspace. And the pose estimator is view invariant, mainly because of the viewpoint-independent manifold learning and special step for searching the global motion. In addition, the experiment results on walking and running sequence demonstrate that our algorithm is efficient for different types of motions. Actually, our method can be generalized to any other types of motions as long as the corresponding subspace can be properly extracted from training data.

**6.4. S-IGA-Based Pose Tracking Results.** We demonstrate our tracking algorithm on walking and running image sequences. And then we compare S-IGA quantitatively with other tracking methods and include particle filter (PF) method [6], particle swarm optimization (PSO) [11], and pose tracking in linear subspace using annealing genetic algorithm (PCA + GA) [5].

As suggested in [6], for a human model with DOF between 6 and 12, PF needs about 1000 particle to run. And in [17], PF used 4000 particles for a 29 DOF human model. While in [11], 7200 particles are used for a 31 DOF human

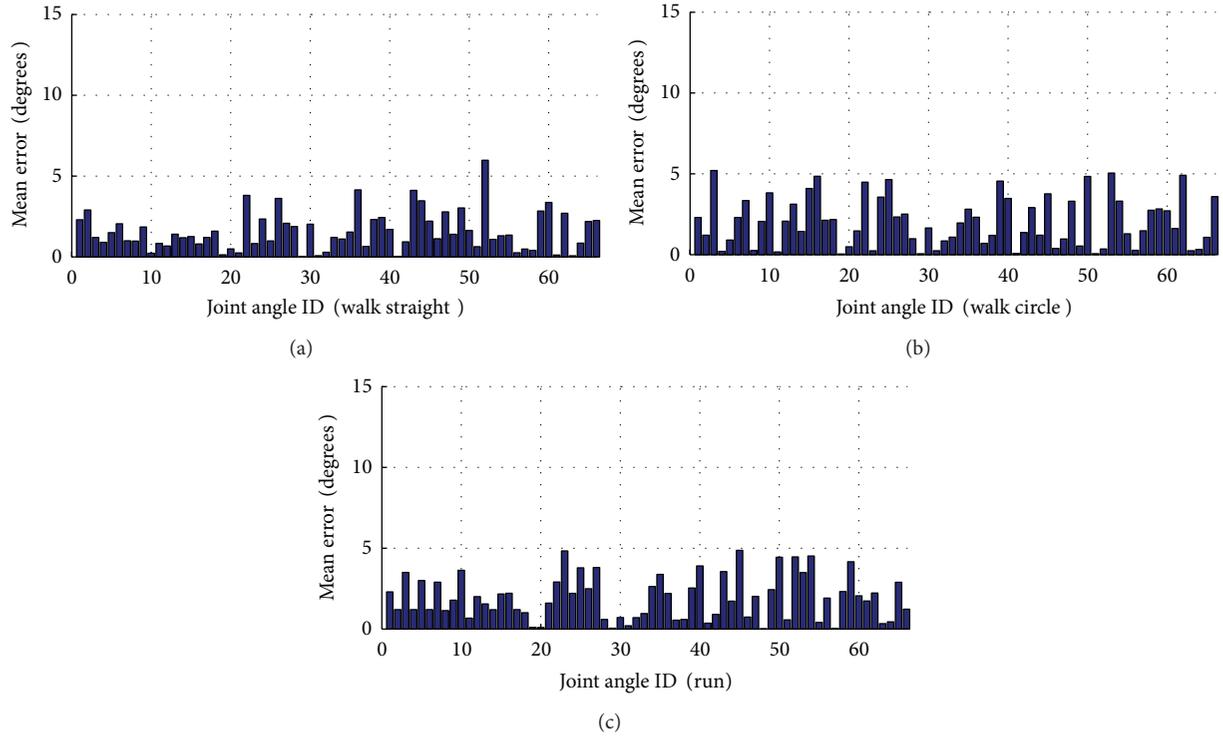


FIGURE 10: The mean errors of individual joint angle for different sequences.

TABLE 2: Ground truth ( $T$ ) and estimated ( $E$ ) results of some joint angles for different motions.

		Joint angles			
		L Femur	R Femur	L Knee	R Knee
Walk straight	$T$	(-23.235, 47.366, 13.754)	(-1.237, 6.456, 25.356)	(-3.245, 50.782, 4.567)	(-1.982, 30.425, 3.904)
	$E$	(-20.967, 43.459, 8.351)	(-0.923, 4.535, 26.429)	(-3.024, 4.368, 8.546)	(0.673, 30.456, 5.336)
Walk in circle	$T$	(-15.324, 50.339, 8.479)	(-0.923, 3.546, 20.764)	(-4.234, 59.436, 7.451)	(-1.590, 28.904, 2.405)
	$E$	(-16.847, 48.837, 5.435)	(-0.456, -0.345, 25.763)	(-3.458, 60.348, 5.345)	(0.890, 34.941, -1.234)
Run	$T$	(-10.213, 43.225, 10.863)	(0.456, 6.433, 24.567)	(-0.932, 49.687, 8.891)	(-0.379, 34.227, 7.904)
	$E$	(-10.763, 46.678, 15.304)	(1.023, 5.645, 31.566)	(-0.983, 42.684, 6.894)	(0.374, 36.679, 2.570)

model. In this paper, the human model in the original space is with 66 DOF; we set the particles size to be 12000 for PF. While in IGA, the quantitative results of experiments show that IGA with 40 antibodies yields results, under similar testing conditions, more accurate than PF available to us. For motion tracking, the iteration time is set to be  $K = 20$ . Thus, the number of likelihood evaluations for a single image would be 800 at most, which is much less than 4000 for GA [5] (size of population is 100, iteration time is 40), 7200 for PSO [13], and 12000 for PF.

We first use IGA-based pose estimation method to analyze human pose on the first image of the video for initialization, where the parameters are set as  $N = 40$ ,  $K = 60$  for careful search of the state space in initialization. While on the following frames, we set the iteration times to be  $K = 20$ . It is mainly because our next-frame propagation strategy can produce a compact antibodies population for optimization.

And in our experiment, we set  $\Sigma = 0.01$  for straight walking sequences and  $\Sigma = 0.02$  for running sequences.

The mean errors of different methods over all joint angles of the test sequences are shown in Figure 12. And Table 3 is the statistics of the average errors and the standard deviations. From Figure 12 and Table 3, we can see that our method achieve better results. The average errors and the standard deviations over all frames are near  $3^\circ$  and  $1^\circ$ , respectively, in general. It also can be found that the change of mean error of our method in whole sequence is small, which indicates that our method can achieve stable tracking of 3D human pose.

Figure 13 is the tracking results on walking and running image sequences, respectively. From the above experimental results we can see that our IGA-based pose estimation method can successfully be used for initialization of tracking. Acutely, our IGA-based pose estimation method is also used for initialization of PF in our experiments. Experimental

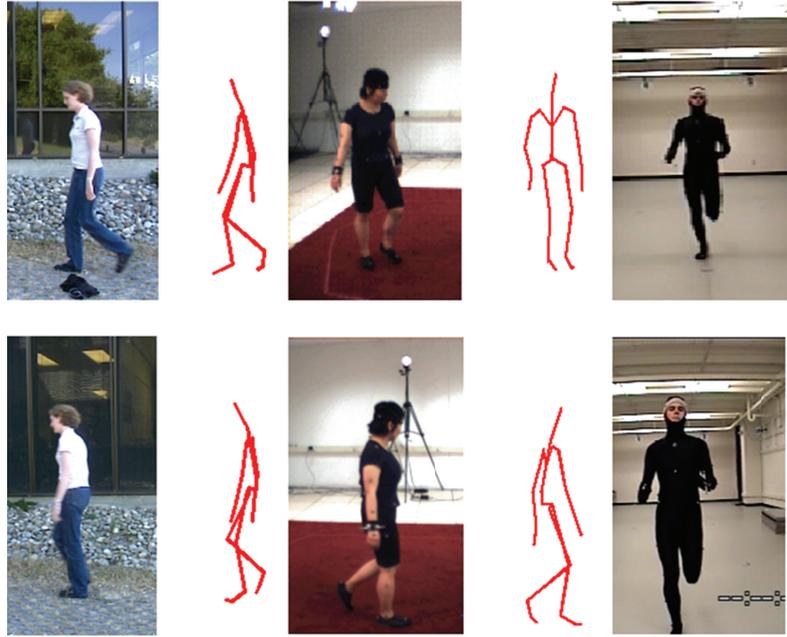


FIGURE 11: Pose estimation results on different image sequences.

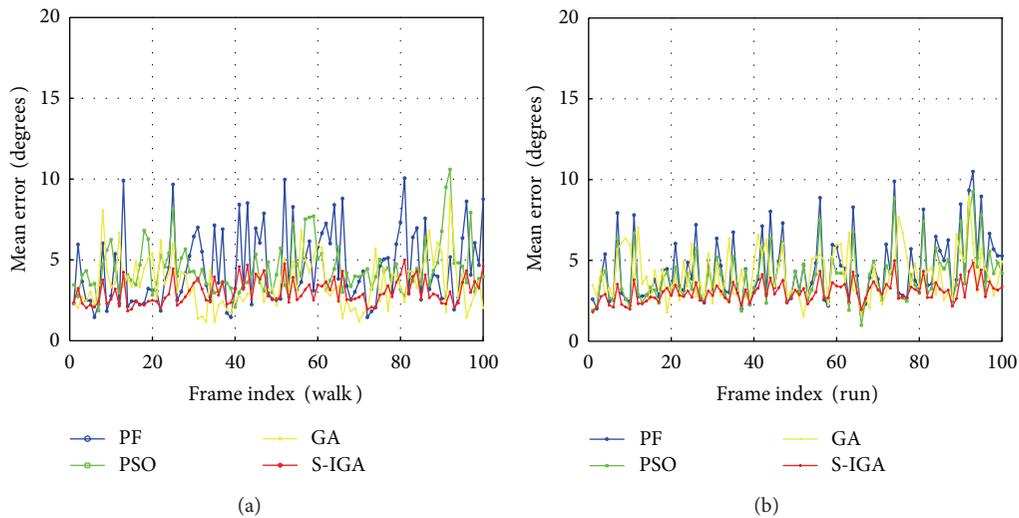


FIGURE 12: Comparison of different tracking methods.

results on different types of motion sequence show that S-IGA has good performance even without any learnt constant motion models, which demonstrate our next-frame propagation strategy is effective to generate initial distribution of antibodies for the next frame.

Experimental results demonstrate that our S-IGA-based tracking method can achieve accurate and stable tracking of 3D human motion. However, our method has some drawbacks as discussed below. Firstly, though pose optimization in the latent space makes our method more effective and accurate, it makes our method not suitable for more complicated motion analysis. So in our future work, we will extend our algorithm to cover a wider class of human motions

and explore switch mechanism between different subspaces. Secondly, in generative tracking approaches, the time taken by an algorithm depends mostly on the number of likelihood evaluations. In our IGA pose optimization method, the time complexity would be  $O(NK)$ , which makes it cannot work for real time applications. In addition, our method is dependent on the silhouette detection from video. But human silhouette detection from video is difficult especially in uncontrolled environment. More robust human silhouette detection method and more sophisticated image likelihood function will be considered in our future work.

Recently, Gaussian Process Latent Variable Models (GPLVM [25]) has been another widely studied latent space

TABLE 3: Results of different tracking methods.

	Walking		Running	
	Mean error	Standard deviations	Mean error	Standard deviations
PF	4.5113	2.3217	4.4669	2.0188
PSO	4.4369	1.5181	4.3949	0.9821
GA	3.5705	1.5651	4.1494	1.4779
S-IGA	3.0626	0.8345	3.0455	0.6370

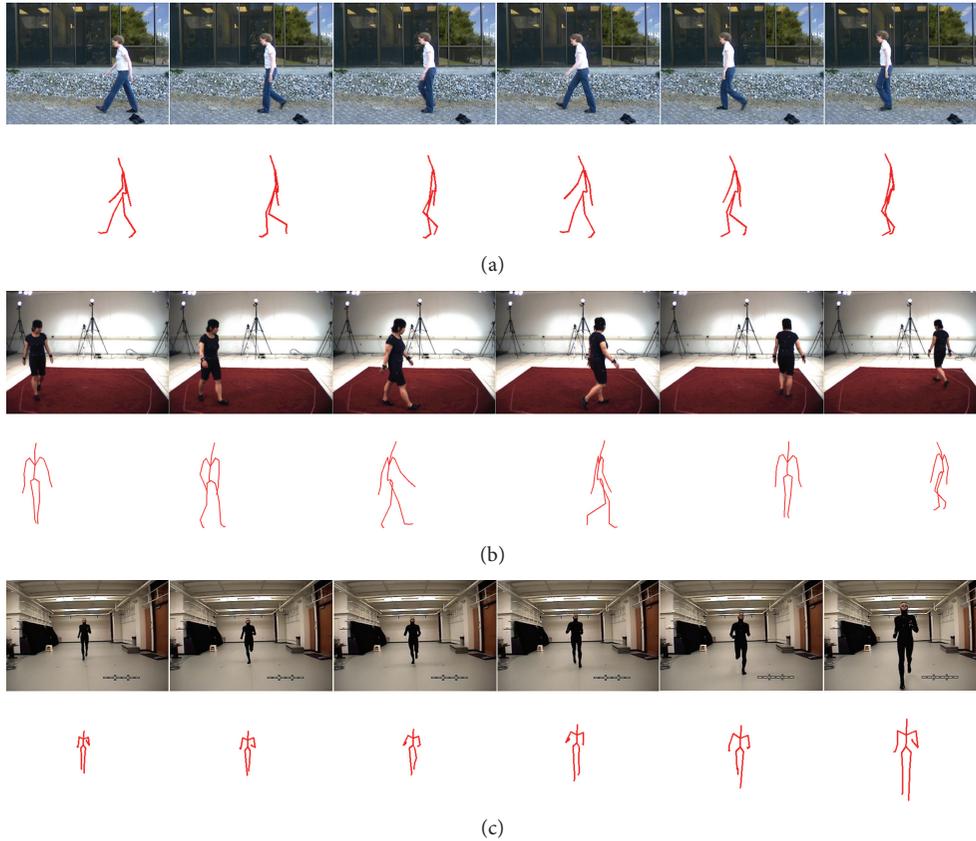


FIGURE 13: Human tracking results on real image sequences, where (a) is results on a subject walking straight (the data is from [24]) (b) is results on a subject walking in circle (the data is from HumanEva [21]), and (c) is results on a subject running (the data is from CMU Mocap database [23]).

learning method for human motion tracking. Compared with manifold learning method (ISOMAP), GPLVM could building the inverse mapping easily. However, GPLVM cannot work well on small training dataset and high-dimensional data. So in our future work, we will study how to apply GPLVM for motion tracking effectively. And more, studies on motion tracking using evolutionary computing methods are still limited. In our future work, we will consider to apply other evolutionary computing methods for motion tracking.

## 7. Conclusions

We presented a novel generative approach to reconstruct 3D human pose from a single monocular image as well as from monocular image sequences. The main contribution is to optimize human pose in learnt latent human motion

space. Pose analysis in the latent space learnt using ISOMAP happens to be more efficient and accurate. In the search strategy, we apply the immune genetic algorithm for pose estimation. A sequential IGA framework is proposed for pose tracking by incorporating the temporal continuity information into the traditional IGA. Compared with GA and PSO, IGA has the ability to use the prior knowledge of human motion. Experiment results on different motion types and image sequences demonstrated that our IGA-based method for pose estimation is effective to deal with occlusion, left-right ambiguity, and the viewpoint problem. The sequential IGA method can achieve stable and accurate pose tracking. Quantitative experiments compared with other state-of-art methods show that our methods achieve better results.

In the future work, we will extend our algorithm to cover a wider class of human motions and explore switch mechanism

between different subspaces. In addition, we will also consider more sophisticated image likelihood and how to reduce the computation time. How to apply other evolutionary computation methods for human motion tracking will also be considered in our future work.

## Acknowledgments

This work is supported by the National High Technology Research and Development Program of China (2007AA01Z334), National Natural Science Foundation of China (61272219, 61021062, and 61100110), Program for New Century Excellent Talents in University of China (NCET-04-04605), Natural Science Foundation of Jiangsu Province (BK2010375), and Key Technology R&D Program of Jiangsu Province (BY2012190, BE2010072, and BE2011058).

## References

- [1] C. Sminchisescu, "3D human motion analysis in monocular video, techniques and challenges," in *Human Motion Understanding, Modeling, Capture and Animation*, R. Kleete, D. Metaxas, and B. Rosenhahn, Eds., Springer, New York, NY, USA, 2007.
- [2] G. Mori and J. Malik, "Recovering 3D human body configurations using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1052–1062, 2006.
- [3] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.
- [4] A. Agarwal and B. Triggs, "Monocular human motion capture with a mixture of regressors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPRW '05)*, p. 72, San Diego, Calif, USA, June 2005.
- [5] X. Zhao and Y. Liu, "Generative tracking of 3D human motion by hierarchical annealed genetic algorithm," *Pattern Recognition*, vol. 41, no. 8, pp. 2470–2483, 2008.
- [6] 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.
- [7] L. Raskin, M. Rudzsky, and E. Rivlin, "Dimensionality reduction using a Gaussian process annealed particle filter for tracking and classification of articulated body motions," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 503–519, 2011.
- [8] R. Poppe, "Vision-based human motion analysis: an overview," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.
- [9] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.
- [10] C. S. Lee and A. Elgammal, "Coupled visual and kinematic manifold models for tracking," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 118–139, 2010.
- [11] V. John, E. Trucco, and S. Ivekovic, "Markerless human articulated tracking using hierarchical particle swarm optimisation," *Image and Vision Computing*, vol. 28, no. 11, pp. 1530–1547, 2010.
- [12] R. Urtasun, D. J. Fleet, and P. Fua, "Monocular 3-D tracking of the golf swing," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 932–938, San Diego, Calif, USA, June 2005.
- [13] C. Sminchisescu and A. Jepson, "Generative modeling for continuous non-linearly embedded visual inference," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, pp. 759–766, Alberta, Canada, July 2004.
- [14] A. Elgammal and C. S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. 681–688, Washington, DC, USA, July 2004.
- [15] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [16] N. R. Howe, M. E. Leventon, and W. T. Freeman, "Bayesian Reconstruction of 3D human motion from single-camera video," in *Proceedings of Neural Information Processing Systems (NIPS '00)*, pp. 820–826, Denver, Colo, USA, 2000.
- [17] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, pp. 126–133, Hilton Head, SC, USA, June 2000.
- [18] T. Krzeszowski, B. Kwolek, and K. Wojciechowski, "Articulated body motion tracking by combined particles swarm optimization and particle filtering," in *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG '10), Part I*, vol. 6374 of *Lecture Notes in Computer Science*, pp. 147–154, 2010.
- [19] X. Zhang, W. Hu, S. Maybank, and L. Xi, "Sequential particle swarm optimization for visual tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 23–28, Anchorage, Alaska, USA, 2008.
- [20] W. Lei and J. Licheng, "Immune evolutionary algorithms," in *Proceedings of the International Conference of Signal Process (ICSP '00)*, pp. 1655–1662, Beijing, China, August 2000.
- [21] CMU Database, <http://mocap.cs.cmu.edu/>.
- [22] M. Gong, L. Jiao, and L. Zhang, "Baldwinian learning in clonal selection algorithm for optimization," *Information Sciences*, vol. 180, no. 8, pp. 1218–1236, 2010.
- [23] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 4–27, 2010.
- [24] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie, "Learning and tracking cyclic human motion," in *Proceedings of Neural Information Processing Systems (NIPS '01)*, pp. 894–900, Vancouver, Canada, December 2001.
- [25] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua, "Priors for people tracking from small training sets," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 403–410, Beijing, China, October 2005.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

