Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 507247, 10 pages
doi:10.1155/2010/507247

Research Article

Nonlinear Synchronization for Automatic Learning of 3D Pose Variability in Human Motion Sequences

M. Mozerov, I. Rius, X. Roca, and J. González

Computer Vision Center and Departament d'Informàtica, Universitat Autònoma de Barcelona, Campus UAB, Edifici O, 08193 Cerdanyola, Spain

Correspondence should be addressed to M. Mozerov, mozerov@cvc.uab.es

Received 1 May 2009; Revised 31 July 2009; Accepted 2 September 2009

Academic Editor: João Manuel R. S. Tavares

A dense matching algorithm that solves the problem of synchronizing prerecorded human motion sequences, which show different speeds and accelerations, is proposed. The approach is based on minimization of MRF energy and solves the problem by using Dynamic Programming. Additionally, an optimal sequence is automatically selected from the input dataset to be a time-scale pattern for all other sequences. The paper utilizes an action specific model which automatically learns the variability of 3D human postures observed in a set of training sequences. The model is trained using the public CMU motion capture dataset for the walking action, and a mean walking performance is automatically learnt. Additionally, statistics about the observed variability of the postures and motion direction are also computed at each time step. The synchronized motion sequences are used to learn a model of human motion for action recognition and full-body tracking purposes.

1. Introduction

Analysis of human motion in activities remains one of the most challenging open problems in computer vision [1–3].

The nature of the open problems and techniques used in human motion analysis approaches strongly depends on the goal of the final application. Hence, most approaches oriented to surveillance demand performing activity recognition tasks in real-time dealing with illumination changes and low-resolution images. Thus, they require robust techniques with a low computational cost, and mostly, they tend to use simple models and fast algorithms to achieve effective segmentation and recognition tasks in real-time.

In contrast, approaches focused on 3D tracking and reconstruction require to deal with a more detailed representation about the current posture that the human body exhibits [4–6]. The aim of full body tracking is to recover the body motion parameters from image sequences dealing with 2D projection ambiguities, occlusion of body parts, and loose fitting clothes among others.

Many action recognition and 3D body tracking works rely on proper models of human motion, which constrain the search space using a training dataset of prerecorded motions [7–10]. Consequently, it is highly desirable to extract useful information from the training set of motion. Traditional treatment suffers from problems inadequate modeling of nonlinear dynamics: training sequences may be acquired under very different conditions, showing different durations, velocities, and accelerations during the performance of an action. As a result, it is difficult to collect useful statistics from the raw training data, and a method for synchronizing the whole training set is required. Similarly to our work, in [11] a variation of DP is used to match motion sequences acquired from a motion capture system. However, the overall approach is aimed to the optimization of a posterior key-frame search algorithm. Then, the output from this process is used for synthesizing realistic human motion by blending the training set.

The DP approach has been widely used in literature for stereo matching and image processing applications [12–14]. Such applications often demand fast calculations in real-time, robustness against image discontinuities, and unambiguous matching.

The DP technique is a core of the dynamic time warping (DTW) method. Dynamic time warping is often used in speech recognition to determine if two waveforms

represent the same spoken phrase [15]. In addition to speech recognition, dynamic time warping has also been found useful in many other disciplines, including data mining, gesture recognition, robotics, manufacturing, and medicine [16].

Initially DTW method was developed for the one-dimensional signal processing (in speech recognition, e.g.). So, for this kind of the signal the Euclidean distance minimization with a weak constraint (the derivative of the synchronization path is constrained) works very well. In our case the dimensionality of the signal is up to 37D and weak constraint does not yield satisfactory robustness due to the noise and the signal complexity. We propose to minimize a composite distance that consists of two terms: a distance itself and a smoothness term. Such kind of a distance has the same meaning of the energy in MRF optimization techniques.

The MRF energy minimization approach shows the perfect performance in stereo matching and segmentation. Likewise, we present a dense matching algorithm based on DP, which is used to synchronize human motion sequences of the same action class in the presence of different speeds and accelerations. The algorithm finds an optimal solution in real-time.

We introduce a median sequences or the best pattern for time synchronization, which is another contribution of this work. The median sequence is automatically selected from the training data following a minimum global distance criterion among other candidates of the same class.

We present an action-specific model of human motion suitable for many applications, that has been successfully used for full body tracking [4, 5, 17]. In this paper, we explore and extend its capabilities for gait analysis and recognition tasks. Our action-specific model is trained with 3D motion capture data for the walking action from the CMU Graphics Lab Motion capture database. In our work, human postures are represented by means of a full body 3D model composed of 12 limbs. Limbs' orientations are represented within the kinematic tree using their direction cosines [18]. As a result, we avoid singularities and abrupt changes due to the representation. Moreover, near configurations of the body limbs account for near positions in our representation at the expense of extra parameters to be included in the model. Then, PCA is applied to the training data to perform dimensionality reduction over the highly correlated input data. As a result, we obtain a lower-dimensional representation of human postures which is more suitable to describe human motion, since we found that each dimension on the PCA space describes a natural mode of variation of human motion. Additionally, the main modes of variation of human gait are naturally represented by means of the principal components found. This leads to a coarse-to-fine representation of human motion which relates the precision of the model with its complexity in a natural way and makes it suitable for different kinds of applications which demand more or less complexity in the model.

The synchronized version of the training set is utilized to learn an action-specific model of human motion. The observed variances from the synchronized postures of the training set are computed to determine which human postures can be feasible during the performance of a particular action. This knowledge is subsequently used in a particle filter tracking framework to prune those predictions which are not likely to be found in that action.

This paper is organized as follows. Section 2 explains the principles of human action modeling. In Section 3 we introduce a new dense matching algorithm for human motion sequences synchronization. Section 4 shows some examples of data base syncronisation. Section 5 describes the action specific model and explains the procedure for learning its parameters from the synchronized training set. Section 6 summarizes our conclusions.

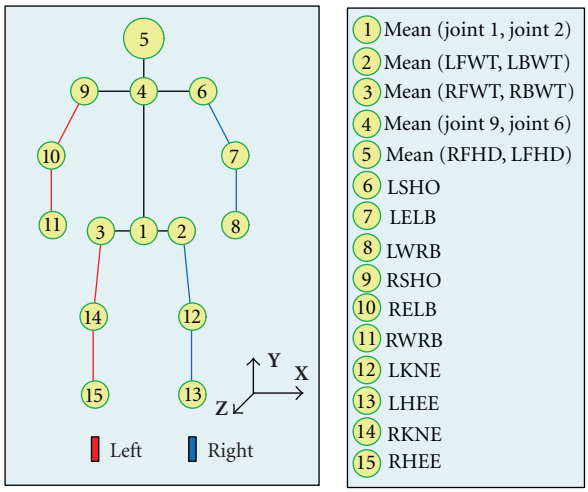
2. Human Action Model

The body model employed in our work is composed of twelve rigid body parts (hip, torso, shoulder, neck, two thighs, two legs, two arms, and two forearms) and fifteen joints; see Figure 1(a). These joints are structured in a hierarchical manner, constituting a kinematic tree, where the root is located at the hip. However, postures in the CMU database are represented using the **XYZ** position of each marker that was placed to the subject in an absolute world coordinates system. Therefore, we must select some principal markers in order to make the input motion capture data usable according to our human body representation. Figure 1(b) relates the absolute position of each joint from our human body model with the markers' used in the CMU database. For instance, in order to compute the position of joint 5 (head) in our representation, we should compute the mean position between the RFHD and LFHD markers from the CMU database, which correspond to the markers placed on each side of the head. Notice that our model considers the left and the right parts of the hip and the torso as a unique limb, and therefore we require a unique segment per each. Hence, we compute the position of joints 1 and 4 (hip and neck joints) as the mean between the previously computed joints 2 and 3, and 6 and 9, respectively.

We use directional cosines to represent relative orientations of the limbs within the kinematic tree [18]. As a result, we represent a human body posture Ψ using 37 parameters, that is,

$$\boldsymbol{\psi} = \left\{u, \theta_1^x, \theta_1^y, \theta_1^z, \ldots, \theta_{12}^x, \theta_{12}^y, \theta_{12}^z\right\}, \tag{1}$$

where u is the normalized height of the pelvis, and θ_l^x, θ_l^y, θ_l^z are the relative directional cosines for limb l, that is, the cosine of the angle between a limb l and each axis x, y, and z, respectively. Directional cosines constitute a good representation method for body modeling, since it does not lead to discontinuities, in contrast to other methods such as Euler angles or spherical coordinates. Additionally, unlike quaternion, they have a direct geometric interpretation. However, given that we are using 3 parameters to determine only 2 DOFs for each limb, such representation generates a considerable redundancy of the vector space components. Therefore, we aim to find a more compact representation of the original data to avoid redundancy.



(a) (b)

Figure 1: (a) Details of the human body model used; (b) the relationship to the marker set employed in the CMU database.

Let us introduce a particular performance of an action. A performance $\mathbf{\Psi}_i$ consists of a time-ordered sequence of postures

$$\mathbf{\Psi}_i = \left\{\boldsymbol{\psi}_i^1, \ldots, \boldsymbol{\psi}_i^{F_i}\right\}, \tag{2}$$

where i is an index indicating the number of performance, and F_i is the total number of postures that constitute the performance $\mathbf{\Psi}_i$. We assume that each two consecutive postures are separated by a time interval δf, which depends on the frame rate of the prerecorded input sequences; thus the duration of a particular performance is $T_i = \delta f F_i$. Finally, an action A_k is defined by all the I_k performances that belong to that action $A_k = \{\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_{I_k}\}$.

As we mentioned above, the original vector space is redundant. Additionally, the human body motion is intrinsically constrained, and these natural constraints lead to highly correlated data in the original space. Therefore, we aim to find a more compact representation of the original data to avoid redundancy. To do this, we consider a set of performances corresponding to a particular action A_k and perform the Principal Component Analysis (PCA) to all the postures that belong to that action. Eventually, the following eigenvector decomposition equation has to be solved:

$$\lambda_j \mathbf{e}_j = \mathbf{\Sigma}_k \mathbf{e}_j, \tag{3}$$

where $\mathbf{\Sigma}_k$ stands for the 37×37 covariance matrix calculated with all the postures of action A_k. As a result, each eigenvector $\mathbf{e}_j$ corresponds to a mode of variation of human motion, and its corresponding eigenvalue λ_j is related to the variance specified by the eigenvector. In our case, each eigenvector reflects a natural mode of variation of human gait. To perform dimensionality reduction over the original data, we consider only the first b eigenvectors that span the new representation space for this action, hereafter *aSpace* [16]. We assume that the overall variance of a new space approximately equals to the overall variance of the unreduced space:

$$\lambda_S = \sum_{j=1}^{b} \lambda_j \approx \sum_{j=1}^{b} \lambda_j + \varepsilon_b = \sum_{j=1}^{37} \lambda_j, \tag{4}$$

where ε_b is the *aSpace* approximation error.

Consequently, we use (4) to find the smallest number b of eigenvalues, which provide an appropriate approximation of the original data, and human postures are projected into the *aSpace* by

$$\widetilde{\boldsymbol{\psi}} = [\mathbf{e}_1, \ldots, \mathbf{e}_b]^T (\boldsymbol{\psi} - \overline{\boldsymbol{\psi}}), \tag{5}$$

where $\boldsymbol{\psi}$ refers to the original posture, $\widetilde{\boldsymbol{\psi}}$ denotes the lower-dimensional version of the posture represented using the *aSpace*, $[\mathbf{e}_1, \ldots, \mathbf{e}_b]$ is the *aSpace* transformation matrix that correspond to the first b selected eigenvectors, and $\overline{\boldsymbol{\psi}}$ is the posture mean value that is formed by averaging all postures, which are assumed to be transformed into the *aSpace*. As a result, we obtain a lower-dimensional representation of human postures which is more suitable to describe human motion, since we found that each dimension on the PCA space describes a natural mode of variation of human motion [16]. Choosing different values for b lead to models of more or less complexity in terms of their dimensionality. Hence, while the *gross-motion* (mainly, the motion of the torso, legs, and arms in low resolution) is explained by the very first eigenvectors, subtle motions in the PCA space representation require more eigenvectors to be considered. In other words, the initial 37-dimensional parametric space becomes the restricted b-dimensional parametric space.

The projection of the training sequences into the *aSpace* constitutes the input for our sequence synchronization algorithm. Hereafter, we consider a multidimensional signal $\mathbf{x}_i(t)$ as an interpolated expansion of each training sequence $\widetilde{\mathbf{\Psi}}_i = \{\widetilde{\boldsymbol{\psi}}_i^1, \ldots, \widetilde{\boldsymbol{\psi}}_i^{F_i}\}$ such as

$$\widetilde{\boldsymbol{\psi}}_i^f = \mathbf{x}_i(t) \quad \text{if } t = (f-1)\delta f;\ f = 1, \ldots, F_i, \tag{6}$$

where the time domain of each action performance $\mathbf{x}_i(t)$ is $[0, \mathrm{T}_i)$.

3. Synchronization Algorithm

As stated before, the training sequences are acquired under very different conditions, showing different durations, velocities, and accelerations during the performance of a particular action. As a result, it is difficult to perform useful statistical analysis to the raw training set, since we cannot put in correspondence postures from different cycles of the same action. Therefore, a method for synchronizing the whole training set is required so that we can establish a mapping between postures from different cycles.

Let us assume that any two considered signals correspond to the identical action, but one runs faster than another (e.g., Figure 2(a)). Under the assumption that the rates ratio of

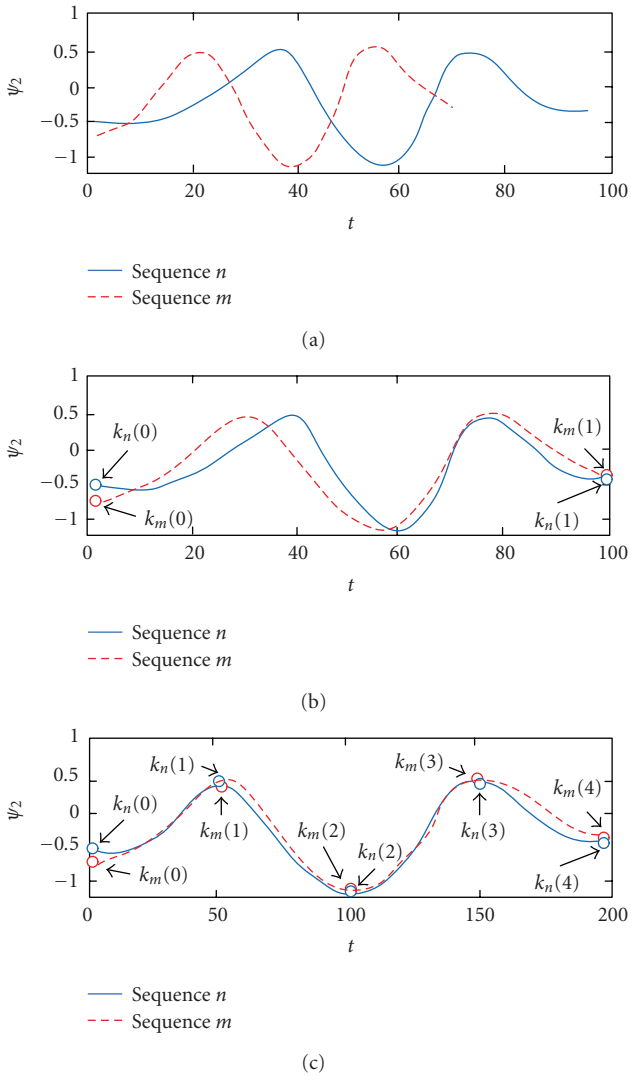


FIGURE 2: (a) Non synchronized one-dimensional sequences. (b) Linearly synchronized sequences. (c) Synchronized sequences using a set of key-frames.

the compared actions is a constant, the two signals might be easily linearly synchronized in the following way:

$$\mathbf{x}_n(t) \approx \mathbf{x}_{n,m}(t) = \mathbf{x}_m(\alpha t); \quad \alpha = \frac{T_m}{T_n}, \tag{7}$$

where $\mathbf{x}_n$ and $\mathbf{x}_m$ are the two compared multidimensional signals, T_n and T_m are the periods of the action performances n and m, and $\widetilde{\mathbf{x}}_{m,n}$ is linearly normalized version of $\mathbf{x}_m$; hence $T_n = T_{m,n}$.

Unfortunately, in our research we rarely, if ever, have a constant rate ratio α. An example, which is illustrated in Figure 2(b), shows that a simple normalization using (7) does not give us the needed signal fitting, and a nonlinear data synchronization method is needed. Further in the text we will assume that the linear synchronization is done and all the periods T_n possess the same value T.

The nonlinear data synchronization should be done by

$$\mathbf{x}_n(t) \approx \mathbf{x}_{n,m}(t) = \mathbf{x}_m(\tau); \quad \tau(t) = \int_0^t \alpha(t)dt, \tag{8}$$

where $\mathbf{x}_{n,m}(t)$ is the best synchronized version of the action $\mathbf{x}_m(t)$ to the action $\mathbf{x}_n(t)$. In literature the function $\tau(t)$ is usually referred to as the distance-time function. It is not an apt turn of phrase indeed, and we suggest naming it as the rate-to-rate synchronization function instead.

The rate-to-rate synchronization function $\tau(t)$ satisfies several useful constraints, that are

$$\tau(0) = 0; \qquad \tau(T) = T; \qquad \tau(t_k) \geq \tau(t_l) \quad \text{if } t_k > t_l. \tag{9}$$

One common approach for building the function $\tau(t)$ is based on a key-frame model. This model assumes that the compared signals $\mathbf{x}_n$ and $\mathbf{x}_m$ have similar sets of singular points, that are $\{t_n(0), \ldots, t_n(p), \ldots t_n(P-1)\}$ and $\{t_m(0), \ldots, t_m(p), \ldots, t_m(P-1)\}$ with the matching condition $t_n(p) = t_m(p)$. The aim is to detect and match these singular points; thus the signals $\mathbf{x}_n$ and $\mathbf{x}_m$ are synchronized. However, the singularity detection is an intricate problem itself, and to avoid the singularity detection stage we propose a dense matching. In this case a time interval $t_n(p+1) - t_n(p)$ is constant, and in general $t_n(p) \neq t_m(p)$.

The function $\tau(t)$ can be represented as $\tau(t) = t(1 + \Delta_{n,m}(t))$. In this case, the sought function $\Delta_{n,m}(t)$ might synchronize two signals $\mathbf{x}_n$ and $\mathbf{x}_m$ by

$$\mathbf{x}_n(t) \approx \mathbf{x}_m(t + \Delta_{n,m}(t)t). \tag{10}$$

Let us introduce a formal measure of synchronization of two signals by

$$D_{n,m} = \int_0^T ||\mathbf{x}_n(t) - \mathbf{x}_m(t + \Delta_{n,m}(t)t)||dt + \mu \int_0^T \left|\left| \frac{d\Delta_{n,m}(t)}{dt} \right|\right| dt, \tag{11}$$

where $\|\bullet\|$ denotes one of possible vector distances, and $D_{n,m}$ is referred to as the synchronization distance that consists of two parts, where the first integral represents the functional distance between the two signals, and the second integral is a regularization term, which expresses desirable smoothness constraints of the solution. The proposed distance function is simple and makes intuitive sense. It is natural to assume that the compared signals are synchronized better when the synchronization distance between them is minimal. Thus, the sought function $\Delta_{n,m}(t)$ should minimize the synchronization distance between matched signals.

In the case of a discrete time representation, (11) can be rewritten as

$$\begin{aligned} D_{n,m} = &\sum_{i=0}^{<P} \left| \mathbf{x}_n(i\delta t) - \mathbf{x}_m\left((i + \Delta_{n,m}(i))\delta t\right) \right|^2 \\ &+ \mu \sum_{i=0}^{<P-1} \left| \Delta_{n,m}(i+1) - \Delta_{n,m}(i) \right|, \end{aligned} \tag{12}$$

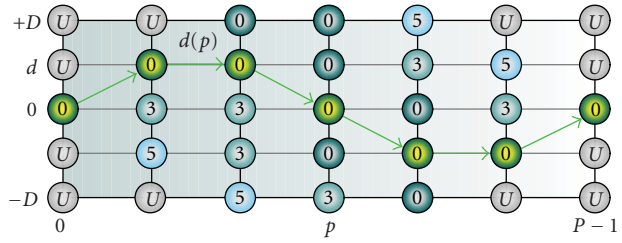


Figure 3: The optimal path through the DSI trellis.

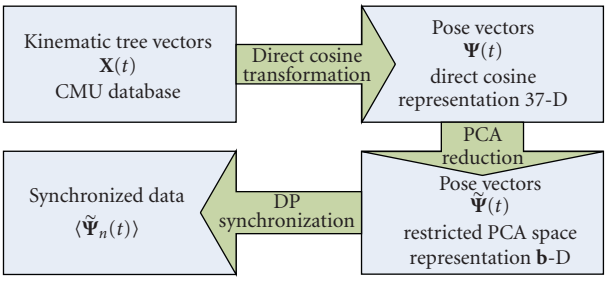


Figure 4: Flowchart of the synchronization method that is based on DP and PCA approaches.

where δt is a time sampling interval. Equation (9) implies

$$|\Delta_{n,m}(p+1) - \Delta_{n,m}(p)| \le 1, \tag{13}$$

where index $p = \{0, \ldots, P-1\}$ satisfies $\delta t P = T$.

The synchronization problem is similar to the matching problem of two epipolar lines in a stereo image. In the case of the stereo image processing the parameter $\Delta(t)$ is called disparity. For stereo matching a DSI representation is used. The DSI approach assumes that 2D DSI matrix has dimensions time $0 \le p < P$, and disparity $-D \le d \le D$. Let $E(d, p)$ denote the DSI cost value assigned to matrix element (d, p) and calculated by

$$E_{n,m}(p,d) = |\mathbf{x}_n(p\delta t) - \mathbf{x}_m(p\delta t + d\delta t)|^2. \tag{14}$$

Now we formulate an optimization problem as follows: find the time-disparity function $\Delta_{n,m}(p)$, which minimizes the synchronization distance between the compared signals $\mathbf{x}_n$ and $\mathbf{x}_m$, that is,

$$\Delta_{n,m}(p) = \arg\min_{d} \sum_{i=0}^{<P} E_{n,m}(i, d(i)) + \mu \sum_{i=0}^{<P-1} |d(i+1) - d(i)|. \tag{15}$$

The discrete function $\Delta(p)$ coincides with the optimal path through the DSI trellis as it is shown in Figure 3. Here term "optimal" means that the sum of the cost values along this path plus the weighted length of the path is minimal among all other possible paths.

The optimal path problem can be easily solved by using the method of dynamic programming. The method consists of step-by-step control and optimization that is given by a recurrence relation:

$$S(p,d) = E(p,d) + \min_{k \in 0, \pm 1} \{S(p-1, d+k) + \mu |d+k|\},$$
$$S(0,d) = E(0,d), \tag{16}$$

where the scope of the minimization parameter $k \in \{0, \pm 1\}$ is chosen in accordance with (13). By using the recurrence relation the minimal value of the objective function in (15) can be found at the last step of optimization. Next, the algorithm works in reverse order and recovers a sequence of optimal steps (using the lookup table $K(p,d)$ of the stored values of the index k in the recurrence relation (16)) and eventually the optimal path by

$$d(p-1) = d(p) + K(p, d(p)),$$
$$d(P-1) = 0, \tag{17}$$
$$\Delta(p) = d(p).$$

Now the synchronized version of $\mathbf{x}_m(t)$ might be easily calculated by

$$\mathbf{x}_{n,m}(p\delta t) = \mathbf{x}_m(p\delta t + \Delta_{n,m}(p)\delta t). \tag{18}$$

Here we assume that n is the number of the base rate sequences and m is the number of sequences to be synchronized.

The dense matching algorithm that synchronizes two arbitrary $\mathbf{x}_n(t)$ and $\mathbf{x}_m(t)$ prerecorded human motion sequences $\mathbf{x}_n(t)$ and $\mathbf{x}_m(t)$ is now summarized as follows.

(i) Prepare a 2D DSI matrix, and set initial cost values E_0 using (14).

(ii) Find the optimal path through the DSI using recurrence equations (16)-(17).

(iii) Synchronize $\mathbf{x}_m(t)$ to the rate of $\mathbf{x}_n(t)$ using (18).

Our algorithm assumes that a particular sequence is chosen to be a time scale pattern for all other sequences. It is obvious that an arbitrary choice among the training set is not a reasonable solution, and now we aim to find a statistically proven rule that is able to make an optimal choice according to some appropriate criterion. Note that each synchronized pair of sequences (n,m) has its own synchronization distance calculated by (12). Then the full synchronization of all the sequences relative to the pattern sequences n has its own global distance:

$$C_n = \sum_{m \in A_k} C_{n,m}. \tag{19}$$

We propose to choose the synchronizing pattern sequence with minimal global distance. In statistical sense such signal can be considered as a median value over all the performances that belong to the set of A_k or can be referred to as "median" sequence.

The flowchart of the synchronization method that is based on DP and PCA approaches is illustrated in Figure 4.

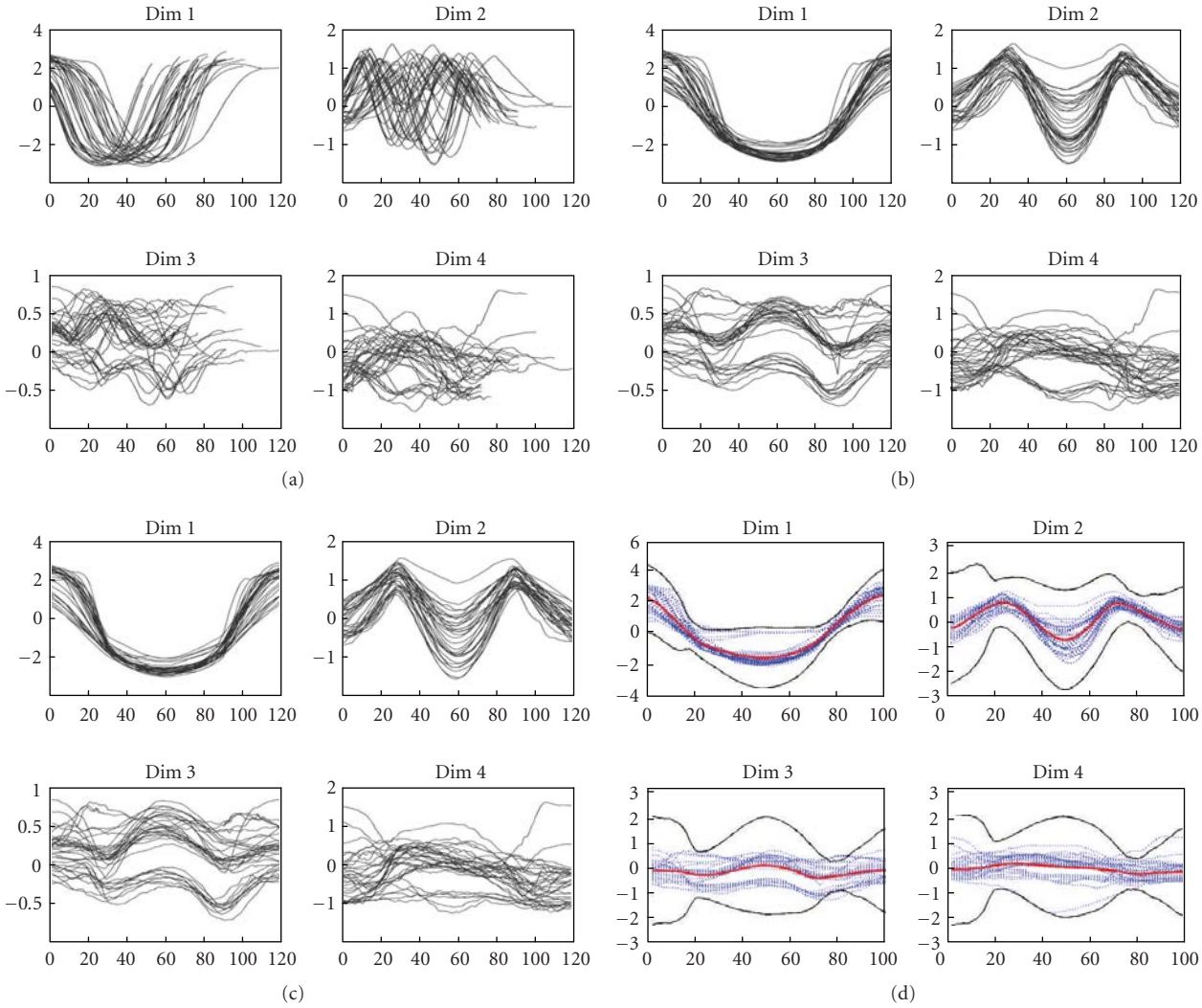


Figure 5: (a) Nonsynchronized training set. (b) Automatically synchronized training set with the proposed approach. (c) Manually synchronized training set with key-frames. (d) Learnt motion model for the bending action.

4. Results of Synchronization

The synchronization method has been tested with many different training sets. In this section we demonstrate our result using 40 performances of a bending action. To build the *aSpace* representation, we choose the first 16 eigenvectors that captured 95% of the original data. The first 4 dimensions within the *aSpace* of the training sequences are illustrated in Figure 5(a). All the performances have different durations with 100 frames on average. The observed initial data shows different durations, speeds, and accelerations between the sequences. Such a mistiming makes very difficult to learn any common pattern from the data. The proposed synchronization algorithm was coded in C++ and run with a 3 GHz Pentium D processor. The time needed for synchronizing two arbitrary sequences taken from our database is 1.5×10^{-2} seconds and 0.6 seconds to synchronize the whole training set, which is illustrated in Figure 5(b).

To prove the correctness of our approach, we manually synchronized the same training set by selecting a set of 5 key-frames in each sequence by hand following a maximum curvature subjective criterion. Then, the training set was resampled; so each sequence had the same number of frames between each key-frame. In Figure 5(c), the first 4 dimensions within the *aSpace* of the resulting manually synchronized sequences are shown. We might observe that the results are very similar to the ones obtained with the proposed automatic synchronization method. The synchronized training set from Figure 5(b) has been used to learn an action-specific model of human motion for the bending action. The model learns a mean-performance for the synchronized training set and its observed variance at each posture. In Figure 5(d) the learnt action model for the bending action is plotted. The mean-performance corresponds to the solid red line while the black solid line depicts ± 3 times the learnt standard deviation at each

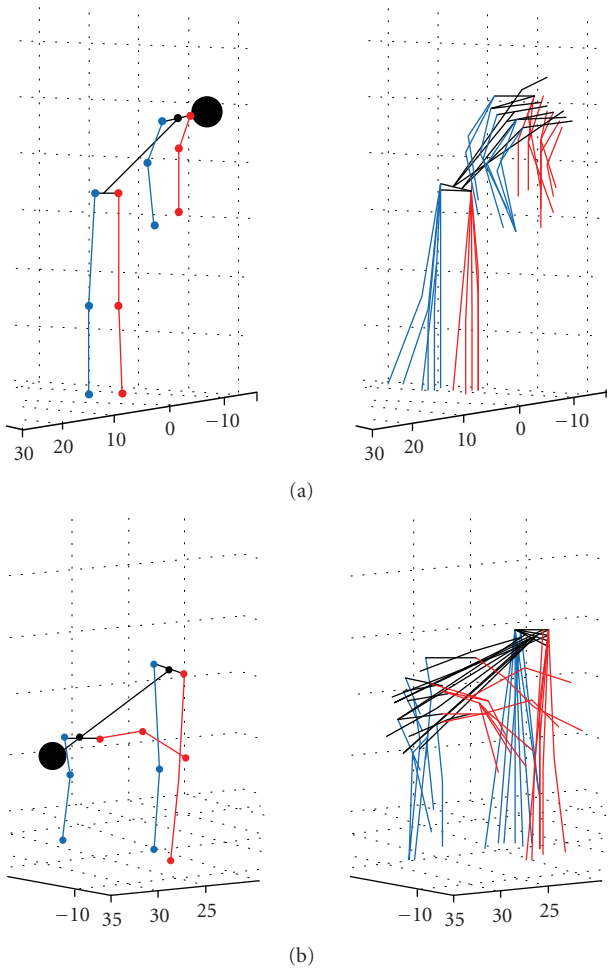


FIGURE 6: (a) and (b) Mean learnt postures from the action corresponding to frames 10 and 40 (left). Sampled postures using the learnt corresponding variances (right).

synchronized posture. The input training sequence set is depicted as dashed blue lines.

This motion model can be used in a particle filter framework as a priori knowledge on human motion. The learnt model would predict for the next time step only those postures which are feasible during the performance of a particular action. In other words, only those human postures which lie within the learnt variance boundaries from the mean performance are accepted by the motion model. In Figure 6 we show two postures corresponding to frames 10 and 40 from the learnt mean performance and a random set of accepted postures by the action model. We might observe that for each selected mean posture, only similar and meaningful postures are generated.

Additionally, to prove the advantage of our approach with respect to DTW we applied our algorithm with the cut objective function (without smoothness term), which is coincide with the DTW algorithm. In this case the synchronization process was not satisfactory: some selected mean postures were completely outliers or nonsimilar to any meaningful posture. It means that the smoothness factor μ in (12) and (16) plays an important role. To find an optimal value of this parameter a visual criterion has been used (the manual synchronization that had been done before yields such a visual estimation technique). However, as a rule of thumb the parameter can be set equal to the mean value of the error term $E(i,d)$:

$$\mu = |\Lambda|^{-1} \sum_{i,d\in\Lambda} E(i,d), \tag{20}$$

where Λ is a domain of i and d indexes and $|\Lambda|$ is the cardinality (or the number of elements) of the domain.

5. Learning the Motion Model

Once all the sequences share the same time pattern, we learn an action specific model which is accurate without loosing generality and suitable for many applications. In this section we consider the waking action and its model is useful for gait analysis, gait recognition, and tracking. Thus, we want to learn where the postures lie in the space used for representation, how they change over time as the action goes by, and what characteristics the different performances have in common which can be exploited for enabling the aforementioned tasks. In other words, we aim to characterize the shape of the synchronized version of the training set for the walking action in the PCA-like space. The process is as follows.

First, we extract from the training set $\widehat{A}_k = \{\widehat{\psi}_1, \ldots, \widehat{\psi}_{I_k}\}$ a mean representation of the action by computing the mean performance $\overline{\Psi}^{A_k} = \{\overline{\psi}^1, \ldots, \overline{\psi}^F\}$ where each mean posture $\overline{\psi}^t$ is defined as

$$\overline{\psi}^t = \sum_{i=1}^{I_k} \frac{\widehat{\psi}_i^t}{I_k}, \quad t = 1, \ldots, F. \tag{21}$$

I_k is the number of training performances for the action $\widehat{A}_k$, $\widehat{\psi}_i^t$ corresponds to the tth posture from the ith training performance, and finally, F denotes the total number of postures of each synchronized performance.

Then, we want to quantify how much the training performances $\widehat{\Psi}_i$ vary from the computed mean performance $\overline{\Psi}^{A_k}$ of (21). Therefore, for each time step t, we compute the standard deviation σ^t of all the postures $\overline{\psi}^t$ that share the same time stamp t, that is,

$$\sigma^t = \sqrt{\frac{1}{I_k} \sum_{i=1}^{I_k} \left(\widehat{\psi}_i^t - \overline{\psi}^t\right)^2}. \tag{22}$$

Figure 7 shows the learned mean performance $\overline{\Psi}^{A_k}$ (red solid line) and ± 3 times the computed standard deviation σ^t (dashed black line) for the walking action. We used $b = 6$ dimensions for building the PCA space representation explaining the 93% of total variation of training data.

On the other hand, we are also interested in characterizing the temporal evolution of the action. Therefore, we compute the main direction of the motion $\overline{\mathbf{v}}_t$ for each

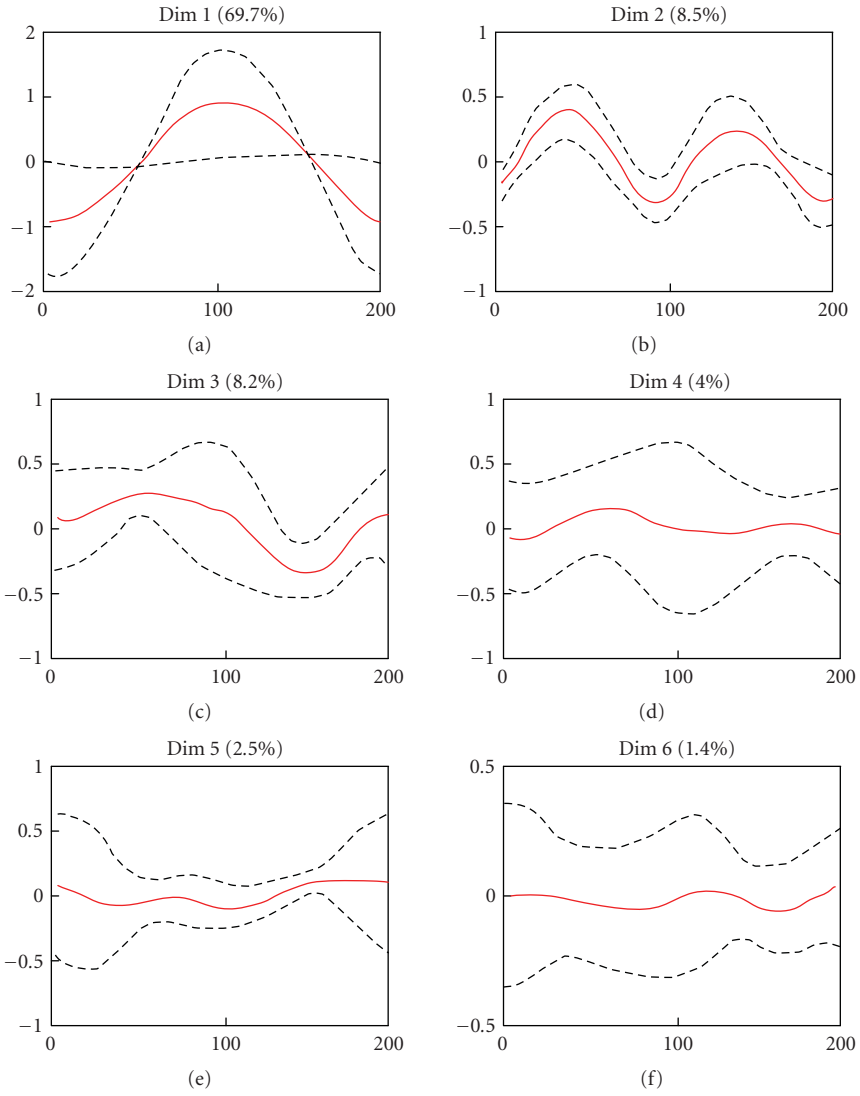


FIGURE 7: Learned mean performance $\overline{\Psi}^{A_k}$ and standard deviation σ^t for the walking action.

subsequence of d postures from the mean performance $\overline{\Psi}^{A_k}$, that is,

$$\mathbf{v}_t = \frac{1}{d}\sum_{j=t}^{t-d+1} \frac{\overline{\psi}^j - \overline{\psi}^{j-1}}{\left\|\overline{\psi}^j - \overline{\psi}^{j-1}\right\|}; \qquad \overline{\mathbf{v}}_t = \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|}, \tag{23}$$

where $\overline{\mathbf{v}}_t$ is a unitary vector representing the observed direction of motion averaged from the last d postures at a particular time step t. In Figure 8 the first 3 dimensions of the mean performance are plotted together with the direction vectors computed in (23).

Each black arrow corresponds to the unitary vector $\overline{\mathbf{v}}_t$ computed at time t, scaled for visualization purposes. Hence, each vector encodes the mean observed motion's direction from time $t - d$ to time t, where d stands for the length of the motion window considered. Additionally, selected postures from the mean performance have been sampled at times $t = 1, 30, 55, 72, 100, 150$, and 168 and overlaid in the graphic.

As a result, the action model Γ^{A_k} is defined by

$$\Gamma^{A_k} = \left\{\Omega^{A_k}, \overline{\Psi}^{A_k}, \sigma_t, \overline{\mathbf{v}}_t\right\}, \quad t = 1, \ldots, F, \tag{24}$$

where Ω^{A_k} is the PCA space definition for action A_k, $\overline{\Psi}^{A_k}$ is the mean performance, and σ_t and $\overline{\mathbf{v}}_t$ correspond to the computed standard deviation and mean direction of motion at each time step t, respectively.

Finally, to handle the cyclic nature of the waking action, we concatenate the last postures in each cycle with the initial postures of the most close performance according to a Euclidean distance criterion within the PCA space. Additionally, the first and last $d/2$ postures from the mean performance (where d is the length of the considered subsequences) are resampled using cubic spline interpolation in order to soft the transition between walking cycles. As a result, we are able to compute σ_t, $\overline{\mathbf{v}}_t$ for the last postures of a full walking cycle.

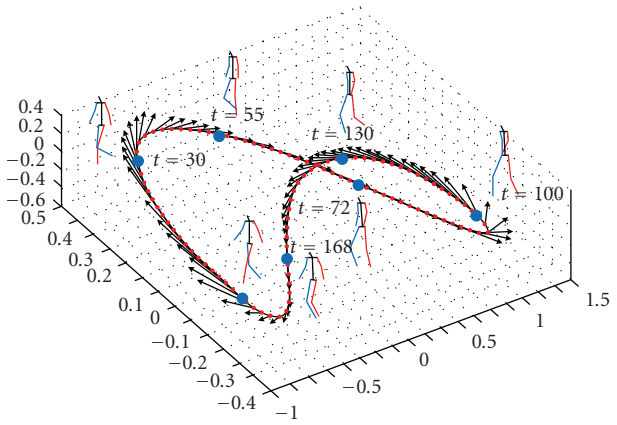


FIGURE 8: Sampled postures at different time steps, and learnt direction vectors $\bar{\mathbf{v}}_t$ from the mean performance for the walking action.

6. Conclusions and Future Work

In this paper, a novel dense matching algorithm for human motion sequences synchronization has been proposed. The technique utilizes dynamic programming and can be used in real-time applications. We also introduce the definition of the median sequence that is used to choose a time-scale pattern for all other sequences. The synchronized motion sequences are utilized to learn a model of human motion and to extract signal statistics. We have presented an action-specific model suitable for gait analysis, gait identification and tracking applications. The model is tested for the walking action and is automatically learnt from the public CMU motion capture database. As a result, we learnt the parameters of our action model which characterize the pose variability observed within a set of walking performances used for training.

The resulting action model consists of a representative manifold for the action, namely, the mean performance, the standard deviation from the mean performance. The action model can be used to classify which postures belong to the action or not. Moreover, the tradeoff between accuracy and generality of the model can be tuned using more or less dimensions for building the PCA space representation of human postures. Hence, using this coarse-to-fine representation, the main modes of variation correspond to meaningful natural motion modes. Thus, for example, we found that the main modes of variation for the walking action obtained from PCA explain the combined motion of both the legs and the arms, while in the bending action they mainly correspond to the motion of the torso.

Future research lines rely on obtaining the joint positions directly from image sequences. Previously, the action model has been successfully used in a probabilistic tracking framework for estimating the parameters of our 3D model from a sequence of 2D images. In [5] the action model improved the efficiency of the tracking algorithm by constraining the space of possible solutions only to the most feasible postures while performing a particular action, thus avoiding estimating postures which are not likely to occur during an action. However, we need to develop robust image-based likelihood measures which evaluate the predictions from our action model according to the measurements obtained from images. Work based on extracting the image edges and the silhouette from the tracked subject is currently in progress. Hence, the pursued objective is to learn a piecewise linear model which evaluates the fitness of segmented edges and silhouettes to the 2D projection of the stick figure from our human body model. Methods for estimating the 6DOF of the human body within the scene, namely, 3D translation and orientation, also need to be improved.

Acknowledgments

This work has been supported by EC Grant IST-027110 for the HERMES project and by the Spanish MEC under projects TIC2003-08865 and DPI-2004-5414. M. Mozerov acknowledges the support of the Ramon y Cajal research program, MEC, Spain.

References

[1] 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.

[2] 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.

[3] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[4] I. Rius, J. Varona, J. Gonzalez, and J. J. Villanueva, "Action spaces for efficient Bayesian tracking of human motion," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 1, pp. 472–475, 2006.

[5] I. Rius, J. Varona, X. Roca, and J. Gonzàlez, "Posture constraints for bayesian human motion tracking," in *Proceedings of the 4th International Conference on Articulated Motion and Deformable Objects (AMDO '06)*, vol. 4069, pp. 414–423, Port d'Andratx, Spain, July 2006.

[6] L. Sigal and M. J. Black, "Measure locally, reason globally: occlusion-sensitive articulated pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 2041–2048, 2006.

[7] J. Gonzàlez, J. Varona, X. Roca, and J. J. Villanueva, "Analysis of human walking based on *aSpaces*," in *Articulated Motion and Deformable Objects*, vol. 3179 of *Lecture Notes in Computer Science*, pp. 177–188, Springer, Berlin, Germany, 2004.

[8] T. J. Roberts, S. J. McKenna, and I. W. Ricketts, "Adaptive learning of statistical appearance models for 3D human tracking," in *Proceedings of the British Machine Vision Conference (BMVC '02)*, pp. 121–165, Cardiff, UK, September 2002.

[9] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *Proceedings of the 7th European Conference on Computer Vision Copenhagen (ECCV '02)*, vol. 2350 of *Lecture Notes in Computer Science*, pp. 784–800, Springer, Copenhagen, Denmark, May 2002.

[10] 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)*, vol. 1, pp. 403–410, October 2005.

[11] A. Nakazawa, S. Nakaoka, and K. Ikeuchi, "Matching and blending human motions using temporal scaleable dynamic programming," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, vol. 1, pp. 287–294, Sendai, Japan, September-October 2004.

[12] Y. Bilu, P. K. Agarwal, and R. Kolodny, "Faster algorithms for optimal multiple sequence alignment based on pairwise comparisons," in *Proceedings of the IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB '06)*, vol. 3, pp. 408–422, October-December 2006.

[13] M. Gong and Y.-H. Yang, "Real-time stereo matching using orthogonal reliability-based dynamic programming," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 879–884, 2007.

[14] J. L. Williams, J. W. Fisher III, and A. S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4300–4311, 2007.

[15] J. Kruskall and M. Liberman, "The symmetric time warping problem: from continuous to discrete," in *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pp. 125–161, Madison-Wesley, Reading, Mass, USA, 1983.

[16] E. Keogh and M. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 1st SIAM International Conference on Data Mining*, pp. 1–12, Chicago, Ill, USA, 2001.

[17] I. Rius, D. Rowe, J. Gonzalez, and F. Xavier Roca, "3D action modeling and reconstruction for 2D human body tracking," in *Proceedings of the 3rd International Conference on Advances in Patten Recognition (ICAPR '05)*, vol. 3687, pp. 146–154, Bath, UK, August 2005.

[18] V. M. Zatsiorsky, *Kinematics of Human Motion*, chapter 1, Human Kinematics, 1998.