3D Human Motion Sequences Synchronization Using Dense Matching Algorithm

Mikhail Mozerov[1], Ignasi Rius[1], Xavier Roca[1], and Jordi González[2]

[1] Computer Vision Center and Departament d'Informàtica
Universitat Autònoma de Barcelona, 08193 Cerdanyola, Spain
mozerov@cvc.uab.es
[2] Institut de Robòtica i Informàtica Industrial (UPC-CSIC), Edifici U
Parc Tecnològic de Barcelona. 08028, Spain

Abstract. This work solves the problem of synchronizing pre-recorded human motion sequences, which show different speeds and accelerations, by using a novel dense matching algorithm. The approach is based on the dynamic programming principle that allows finding an optimal solution very fast. Additionally, an optimal sequence is automatically selected from the input data set to be a time scale pattern for all other sequences. The synchronized motion sequences are used to learn a model of human motion for action recognition and full-body tracking purposes.

1 Introduction

Visual motion analysis of human motion remains one of the most challenging open problems from computer vision [4,10]. The number of related difficulties is wide ranging from shape and appearance changes, 2D-3D projection ambiguities and self and non-self occlusions among others. Many applications, such as action recognition or full-body 3D tracking, use high dimensional space models, and only a reduced number of the considered space components are directly observable from 2D images. As a result, incorporating *a priori* information on human motion into these applications is essential. Many action recognition and 3D body tracking works rely on proper models of human motion, which constrain the search space using a training data set of pre-recorded motions [3,6,8,9]. Consequently, it is highly desirable to extract useful information from the training set of motion. However, training sequences may be 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 put in correspondence postures from different sequences of the same action in order to perform useful statistical analysis to the raw training data. Therefore, a method for synchronizing the whole training set is required so that we can establish a mapping between postures from different sequences. Ning et al. proposed a method for normalizing the length of cyclic walking sequences using a self-correlation measure [6]. As a result, the training walking cycles are rescaled to last the same period of time and are aligned to the same phase. Then, a walking motion model is learnt as Gaussian distributions per each joint, which include constraints on human motion. The model is used to track a walking sequence of a 12 DOF body model using a particle filtering framework. However, unlike our approach, self-correlation is only suitable for cyclic motion sequences.

Similarly to our work, in [5] a variation of Dynamic Programming (DP) is used to match motion sequences acquired from a motion capture system. However, the overall approach is aimed at 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. They divided the body in 4 portions, and similarities are evaluated independently for each part. In contrast, our approach synchronizes motion sequences considering the whole body in the matching process. We also use a representation based on relative joint angles which is more suitable for human motion representation.

The DP approach has been widely used in the literature for stereo matching and image processing applications [1,7]. Such applications often demand fast calculations in real-time, robustness against image discontinuities and unambiguous matching. 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. Additionally, we automatically select from the training data the best pattern for time synchronization following a minimum global distance criterion.

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.

The remainder of 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. Experimental results with real 3D human motion data are presented and discussed in Section 4. Section 5 summarizes our conclusions.

2 Human Action Model

The motion sequences we want to synchronize have been acquired using a commercial Motion Capture system. A set of 19 reflective markers were placed on several characteristic points of the subject's body to obtain its absolute 3D positions. The body model employed is composed of twelve rigid body parts (hip, torso, shoulder, neck, two thighs, two legs, two arms and two forearms) and fifteen joints. These joints are structured in a hierarchical manner, constituting a kinematic tree, where the root is located at the hip. We use directional cosines to represent relative orientations of the limbs within the kinematic tree. The height of the pelvis is also modeled since it provides useful information for characterizing actions such as jumping or sitting. As a result, we represent a human body posture ψ using 37 parameters, i.e.

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

where u is the normalized height of the pelvis, and $\theta_l^x, \theta_l^y, \theta_l^z$ are the relative directional cosines for limb l, i.e. 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 doesn't 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, such representation generates a considerable redundancy of the vector space components. Indeed, we are using 3 parameters to determine only 2 DOF for each limb.

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

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

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 pre-recorded 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 $A_k = \left\{ \mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_{I_k} \right\}$.

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 Principal Component Analysis 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* [2]. 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 Eq. (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

$$\tilde{\mathbf{\psi}} = \left[\mathbf{e}_1, \ldots, \mathbf{e}_b \right]^T \left(\mathbf{\psi} - \bar{\mathbf{\psi}} \right), \tag{5}$$

where $\boldsymbol{\psi}$ refers to the original posture, $\tilde{\Psi}$ 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 $\bar{\boldsymbol{\psi}}$ 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 more suitable to describe human motion since we found that each dimension on the aSpace describes a natural mode of variation of human motion.

The projection of the training sequences into the *aSpace* will constitute 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 $\tilde{\boldsymbol{\Psi}}_i = \left\{\tilde{\boldsymbol{\psi}}_i^1,\ldots,\tilde{\boldsymbol{\psi}}_i^{F_i}\right\}$ such as

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

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

3 Synchronization Algorithm

Let us assume that any two considered signals correspond to the identical action, but one runs faster than another (e.g. Fig. 1. (a)). Under the assumption that the rates ratio of the compared actions is a constant, the two signals might be easily linearly synchronized in the following way

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

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*, $\tilde{\mathbf{x}}_{m,n}$ 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 Fig. 1. (b), shows that a simple normalization using Eq. (7) does not give us the needed signal fitting, and a nonlinear data synchronization method is needed. Further in the text we shall assume that the linear synchronization is done and all the periods T_n possess the same value *T*.

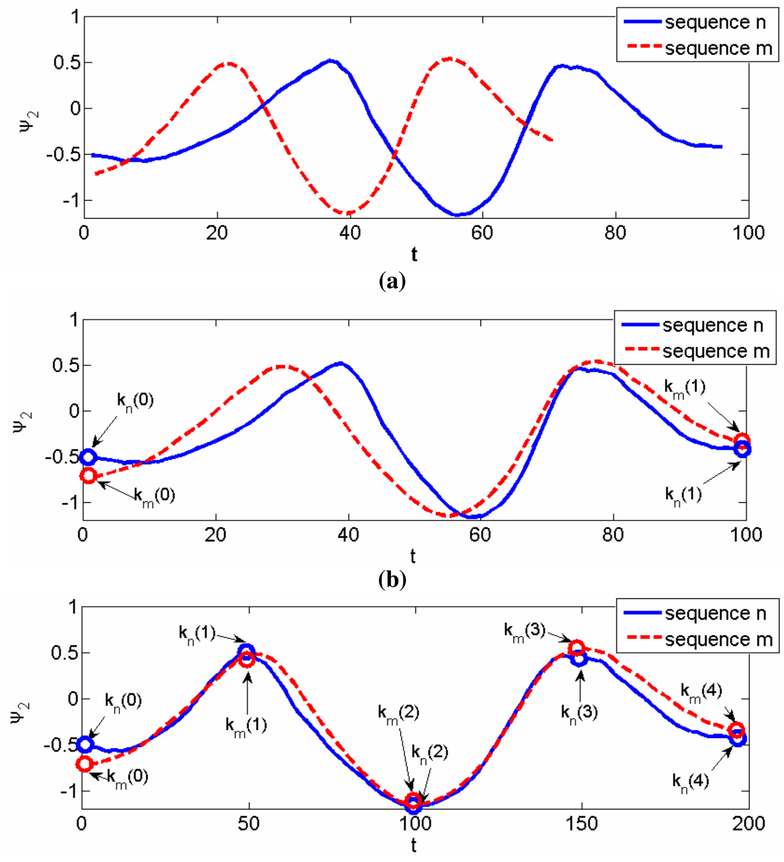
The nonlinear data synchronization should be done by

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

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 the literature the function *τ(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 *τ(t)* satisfies several useful constraints, that are

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



(a)

(b)

(c)

Fig. 1. **(a)** Non synchronized one-dimensional sequences. **(b)** Linearly synchronized sequences. **(c)** Synchronized sequences using a set of key-frames.

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 $\{t_n(0), \ldots t_n(p),..t_n(P\text{-}1)\}$ and $\{t_m(0), \ldots t_m(p),..t_m(P\text{-}1)\}$ 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 $\Delta_{\mathrm{n,m}}(t)$ 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

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

where $\|\bullet\|$ denotes one of possible vector distances, $D_{n,m}$ 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, Eq.(11) can be rewritten as

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

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

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

where index $p=\{0, \ldots, P\text{-}1\}$ 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 disparity space image (DSI) representation is used [1,7]. The DSI approach assumes that 2D DSI matrix has dimensions time $0 \leq p < P$, and disparity $-D \leq d \leq D$. Let $E(d, p)$ denote the DSI cost value assigned to matrix element (d, p) and calculated by

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

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 $\mathbf{x}_m$ i.e.

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

The discrete function $\Delta(p)$ coincides with the optimal path through the DSI trellis as it is shown in Fig. 2. 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

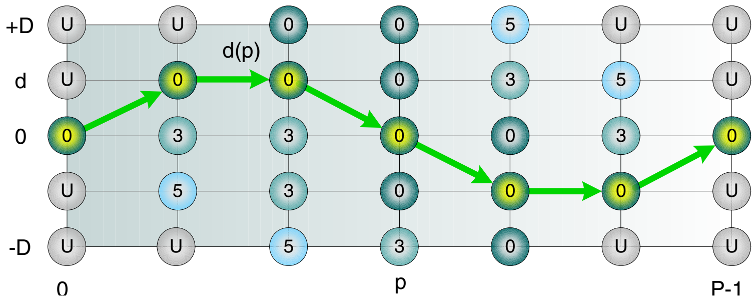


Fig. 2. The optimal path trough the DSI trellis

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

where the scope of the minimization parameter $k \in \{0,\pm 1\}$ is chosen in accordance with Eq. (13). By using the recurrence relation the minimal value of the objective function in Eq.(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

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

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 synchronize two arbitrary $\mathbf{x}_n(t)$ and $\mathbf{x}_m(t)$pre-recorded human motion sequences $\mathbf{x}_n$(t) and $\mathbf{x}_m(t)$ is now summarized as follows:

- Prepare a 2D DSI matrix, and set initial cost values E_0 using Eq. (14).
- Find the optimal path trough the DSI using recurrence Eqs. (16-17).
- Synchronize $\mathbf{x}_m(t)$ to the rate of $\mathbf{x}_n(t)$ using Eq.(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 Eq. (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.

4 Computer Experiments

The synchronization method has been tested with a training set consisting of 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 Fig.3.(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

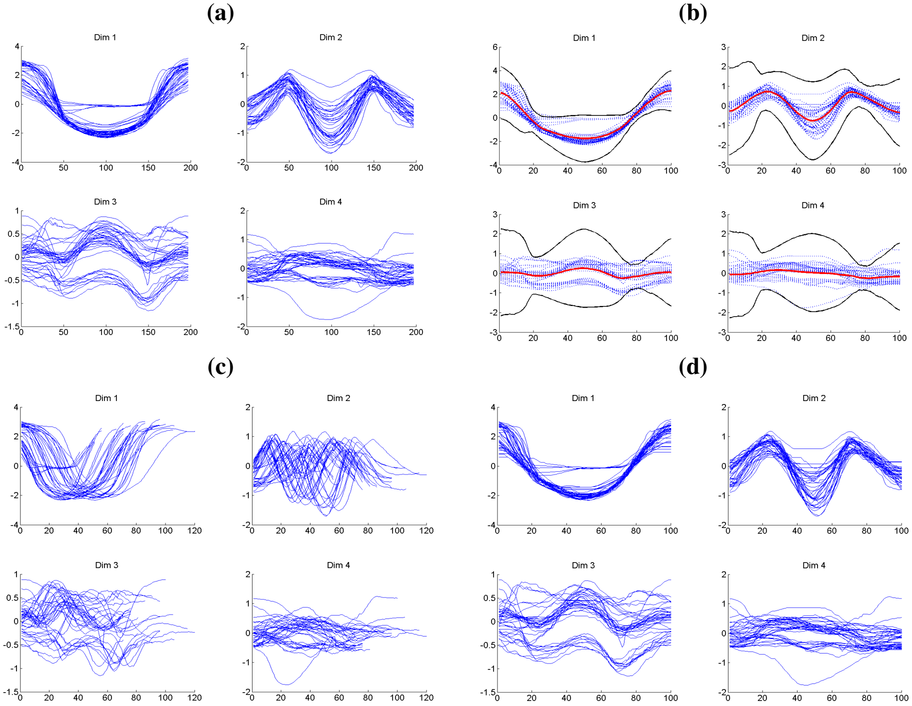
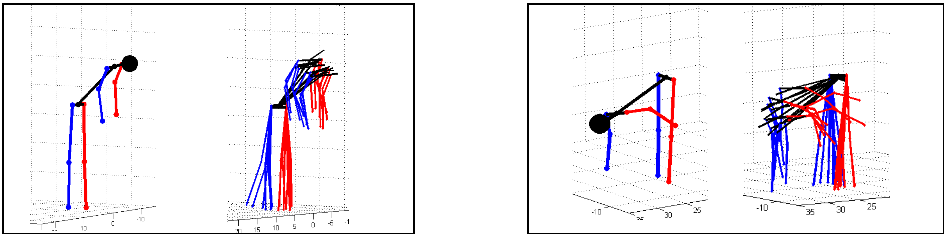


Fig. 3. **(a)** Non-synchronized training set. **(b)** Automatically-synchronized training set with the proposed approach. **(c)** Manually-synchronized training set with key-frames. **(d)** Learnt motion model for the bending action.



(a)

(b)

Fig. 4. (a) and (b) Mean learnt postures from the action corresponding to frames 10 and 40 (left). Sampled postures using the learnt corresponding variances (right).

database is 1.5 10^{-2} seconds and 0.6 seconds to synchronize the whole training set, which is illustrated in Fig.3.(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 Fig.3.(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 Fig.3.(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 Fig.3.(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 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 Fig.4 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.

5 Conclusion

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.

Acknowledgements

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, and J. Gonzàlez also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC.

References

1. Brown, M. Z., Burschka, D., and Hager, G. D.: Advances in computational stereo. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25 (2003) 993–1008
2. Gonzàlez, J., Varona, J., Roca, X., and Villanueva, J.J.: Analysis of human walking based on aSpaces. Lecture Notes in Computer Science, Vol. 3179. Springer-Verlag, Berlin Heidelberg New York (2004) 177–188
3. Grochow, K., Martin, S.L. , Hertzmann, A., and Popovic, Z.: Style-based inverse kinematics. ACM Transactions on Graphics (Proc. of SIGGRAPH 2004), Vol. 23 (2004) 522–531
4. Moeslund, T.B., and Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding, Vol. 81 (2001) 231–268
5. Nakazawa, A., Nakaoka, S., and Ikeuchi, K.: Matching and blending human motions using temporal scaleable dynamic programming. Proc. of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, (2004) 287–294
6. Ning, H., Tan, T., Wang, L., and Hu, W.: Kinematics-based tracking of human walking in monocular video sequences. Image and Vision Computing, Vol. 22 (2004) 429–441
7. Scharstein, D., and Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, Vol. 47 (2002) 7–42
8. Sidenbladh, H., Black, M.J., and Sigal, L.: Implicit probabilistic models of human motion for synthesis and tracking. Lecture Notes in Computer Science, Vol. 2350. Springer-Verlag, Berlin Heidelberg New York (2002) 784–800
9. Urtasun, R., Fleet, D.J., Hertzmann, A. and Fua, P.: Priors for people tracking from small training sets. Proc. IEEE International Conference on Computer Vision (ICCV05), Vol. 1 (2005) 403–410
10. Wang, L., Hu, W., and Tan, T.: Recent developments in human motion analysis. Pattern Recognition, Vol. 36 (2003) 585–601