*Volume 23 (2004), number 4 pp. 799–812* COMPUTER GRAPHICS *forum*

# Style-Based Motion Synthesis†

Raquel Urtasun[1], Pascal Glardon[2], Ronan Boulic[2], Daniel Thalmann[2] and Pascal Fua[1]

[1] Computer Vision Lab, EPFL, CH-1015 Lausanne, Switzerland
[2] Virtual Reality Lab, EPFL, CH-1015 Lausanne, Switzerland

**Abstract**

*Representing motions as linear sums of principal components has become a widely accepted animation technique. While powerful, the simplest version of this approach is not particularly well suited to modeling the specific style of an individual whose motion had not yet been recorded when building the database: it would take an expert to adjust the PCA weights to obtain a motion style that is indistinguishable from his. Consequently, when realism is required, the current practice is to perform a full motion capture session each time a new person must be considered. In this paper, we extend the PCA approach so that this requirement can be drastically reduced: for whole classes of cyclic and noncyclic motions such as walking, running or jumping, it is enough to observe the newcomer moving only once at a particular speed or jumping a particular distance using either an optical motion capture system or a simple pair of synchronized video cameras. This one observation is used to compute a set of principal component weights that best approximates the motion and to extrapolate in real-time realistic animations of the same person walking or running at different speeds, and jumping a different distance.*

**Keywords:** animation, computer vision, human body tracking, motion synthesis, motion capture, motion models.

**ACM CCS:** I.3.7 [Computer Graphics] Three-Dimensional Graphics and Realism: Animation.

## 1. Introduction

Representing motions as linear sums of principal components has become a widely accepted animation technique [1–4]. These principal components are computed by motion capturing as many people as possible performing a specific activity, representing each motion as a temporally quantized vector of joint angles, and performing a Principal Component Analysis (PCA) [5] on the resulting database of motion vectors. Linear combinations of these vectors can then be considered as valid motions and used to produce new animations.

While powerful, the simplest version of this approach is not particularly well suited to modeling the specific style of an individual whose motion had not yet been recorded when building the database: it would take an expert to adjust the PCA weights to obtain a motion style that is indistinguishable from his. Consequently, when realism is required, the current practice is to perform a full motion capture session each time a new person must be considered.

In this paper, we show that the PCA approach can be extended so that this requirement can be drastically reduced: For whole classes of cyclic and noncyclic motions such as walking, running or jumping, it is enough to observe the newcomer walking or running only once at a particular speed or jumping a particular distance using either an optical motion capture system or a simple pair of synchronized video cameras. This one observation is used to compute a set of principal component weights that best approximates the motion and to extrapolate in real-time realistic animations of the same person moving at different speeds or jumping at different distances. This has an important advantage over traditional blending approaches that simply rely on a linear combination of the captured data to create new styles [4,6,7]: extrapolation allows us to reach a comparatively larger subspace of physically correct motions. Furthermore, unlike techniques such as the one described in [8], our approach does not need fine parameter settings for initialization purposes. Our

[†] This work was supported in part by the Swiss National Science Foundation.

Submitted July 2004
Revised October 2004
Accepted November 2004

animations are produced in real-time, with potential changes of physical motion properties such as walking speed or jumping distance.

We first validated our approach for both cyclical and noncyclical motions by exclusively using reliable optical motion capture data: we built a walking database by capturing nine people walking at speeds ranging from 3 to 7 km/h, a running database by capturing five people running at speeds ranging from to 6 to 12 km/h, and a jumping database by capturing jumps ranging from 40 to 120 cm in length for four different people. Given a PCA decomposition and a new captured motion that had not been used to perform this decomposition, we project it into PCA space and compute Mahalanobis distances to database motions corresponding to the same speed or jump length. These can then be used to synthesize new motions corresponding to different speeds or jumping lengths and we have verified that these synthesized motions and the actual ones that we have also recorded are both statistically and visually close.

We then replaced the optical motion capture data for the new person by stereo imagery acquired with a cheap and commercially available device [9]. To this end, we take advantage of a technique that we developed in previous work [10] and that lets us track the motion by minimizing a differentiable objective function whose state variables are the PCA weights. This step replaces the projection into PCA space discussed above and allows us again to speed-up or slow-down the motion while preserving the style.

Note that, even though the database we used for validation purposes is specific to three kinds of motions, the approach itself is completely generic. Transposed onto a production set, it has great labor-saving potential: the actors' motion need only be captured once to generate a whole range of realistic and personalized animations, thus sparing the need for time-consuming motion capture sessions and expensive gear.

In the remainder of this paper, we will first discuss related work, introduce our motion models and show how they can be used to capture the motion from synchronized videos. We will then introduce our approach to computing PCA weights for observed motions that are not part of the initial motion database and using them to extrapolate new ones. Finally, we will validate it using both optical motion capture and video data.

## 2. Related Work

The literature on walking and running animation is so rich that a full article would be necessary to discuss the advances for walking alone since the last major review of the field [11]. Three main classes of approach can nevertheless be distinguished.

**Inverse kinematics.** This involves specifying at each key time the corresponding key positions of some joints and obtaining the joint angles according to biomechanical data information. This can be done efficiently [12,13] but there is no guarantee of physical realism and this often leads to overly mechanical movements.

**Inverse dynamics.** These techniques look for the correct forces and torques to apply to joints to reach a given position. This produces smooth results but may involve postures that are not humanly feasible. It therefore becomes necessary to check and potentially correct these postures by applying appropriate constraints [14–16].

**Motion capture and editing.** New motions are typically created by blending and interpolation. Composite motions can then be obtained by combining several captured ones. Comparable methods presented in [17–20] do this by connecting them into a directed graph. Its edges represent motion clips and nodes are potential connecting points between clips. The user can generate new motions by moving along an optimized path in the graph.

To synthesize a motion that closely resembles that of a specific person, as opposed to a generic virtual human, using motion capture data is clearly the favored approach because there is no easy way to set the parameters for either Inverse Kinematics or Inverse Dynamics to achieve the desired goal. The latter class of techniques is therefore the most widely used. However, they are not usually designed to allow the modification of intrinsic properties of the database motions, which is the issue we address in this paper and discuss in more details below.

### 2.1. Motion Editing

Constraint-based techniques, discussed and classified in [21], alter an original motion while preserving some specific geometric features. Between them, space-time constraint [22,23] or physically based approaches [14,24] provide effective tools to interactively manipulate a motion clip by changing some important properties of the movement. While performing almost in real-time, these methods are appropriate for slight modifications of the motion but not to introduce stylistic variations, mainly because they are difficult to formulate as mathematical constraints.

A motion can be treated as a time-varying signal. Signal processing techniques have therefore been developed both to edit the complete motion by varying the frequency bands of the signal [25], or the Fourier coefficients [20], and to randomize the original motion [26]. However, controlling the randomization is far from straightforward and may yield unpredictable results that can be physically impossible.

Blending or interpolation are the typical approaches to generating new motions. For example, Ashraf and Wong [27] interpolate walking and running motions in 3D space where the axes correspond to significant parameters of a locomotion cycle. The method uses bilinear interpolation to synthesize a new motion given four motions having different values for each of the three dimensions. This approach, however, is limited to a small number of input motions and parameters.

Multivariate interpolation can be used to solve this problem. In [6], multivariate interpolation is performed on a wide range of motion capture data. Motions are defined by B-Spline coefficients and manually classified according to their characteristics, which yields a parameter vector for each motion. Similar motions are time-normalized using a time warping process that structurally aligns them. New ones are then generated by applying polynomial and Radial Basis Function interpolation between the B-Spline coefficients. Kovar and Gleicher [7] present a general method allowing motions generation from a various input motions. They introduce registration curves that ensure a consistent time-warping and root alignment and apply physical constraints to produce blended motions.

While powerful, these methods are highly dependent on the weights values the animators assign to a set of input motions. Thus, determining a good combination of these weights becomes difficult for the creation of a very specific motion. To enhance this control, a possible alternative is presented in [28]. A motion is modified interactively by an animator manipulating a refletive device whose motions are captured by an optical system and transferred to a virtual character.

The approaches presented above have the disadvantages that the newly generated motion cannot be retargeted to subjects of different sizes. Park *et al.* [29] propose a locomotion generation, adaptable to any target character, based on the motion interpolation of [6,30]. A motion retargeting based on the approach introduced by Shin *et al.* [31] provides a real-time animation framework. However, even if stylistic variations were incorporated into this approach, generating the motions corresponding to a specific person will involve the same problems as before.

### 2.2. Principal Component Analysis

The methods discussed above suffer from a number of limitations: First, there is no intuitive way to create a motion with specific characteristics. Second, they do not provide for extrapolation. As a result, to create a whole range of motions such as those of a specific athlete running at varying speeds, one must perform a full motion capture session of that athlete actually running at a number of different speeds. This can prove cumbersome and the technique we advocate in this paper takes advantage of Principal Components Analysis (PCA) to alleviate this problem by giving our system an extrapolation capability.

PCA [2,3,32] has recently been extensively used in motion synthesis. It has also been used to compress keyframed animation data [1] and to emphasize similarities between instances of objects such as heads [33,34] to deform them for example by changing their apparent age or gender. Unfortunately for walking, running and jumping motions, the PCA weights have no obvious direct interpretation. More sophisticated blending techniques are required. In earlier work [4], we have used hierarchical structures to isolate specific motion parameters. It can be used to extrapolate new motions but is not designed to reproduce the specific style of an individual whose motions are not in the motion database. In theory, it could be modeled as a weighted sum of database motions, which would require finding the right weights. However, in practice, finding the weights by hand is very difficult and that is precisely what the technique proposed in this paper let's us do from a single example for each new style. Therefore, our earlier approach and the one presented here address different problems and are essentially complementary.

While our intention is quite similar to [8], by automatically extrapolating stylistic animations, we enhance the motion creation process by offering a separate control of its physical parameters (speed for walking, length for jumping) and the ability to retarget the motion to virtual humans of different sizes. Moreover, we provide a mathematical framework that does not depend on a specific parameterization and fullfils real-time constraints.

## 3. Models for Motion Synthesis and Analysis

In this section, we introduce the motion models we use both to synthesize walking, running and jumping animations and to capture such motions from video sequences.

### 3.1. Walking and Running

To build walking models, we used a Vicon optical motion capture system [35] to capture nine people walking at speeds ranging from 3 to 7 km/h by increments of 0.5 km/h on a treadmill. Similarly, to build a running model, we captured five people running at speeds ranging from 6 to 12 km/h by increments of 1 km/h. The data were segmented into cycles and sampled at regular time intervals using quaternion spherical interpolation [36] so that each example can be treated as $N = 100$ samples of a motion starting at normalized time 0 and ending at normalized time 1.

As different subjects may be shorter or taller, size normalization is required. Murray [37] has shown that, for adults, relative angles for the hip, knee and ankle in the sagittal plane have very similar trajectories for the same value of normalized speed $V$, obtained by dividing the walking velocity $v$ by the hip joint height $H$ that represents the leg length. In our previous work [4] we have generalized this approach to running and jumping. The input motions of our database have been normalized by dividing the translation values of the humanoid root by the leg length of the captured subject.

An example is then represented by an *angular motion vector* $\Psi$ of dimension $N * NDofs$, where $NDofs = 78$ is the number of angular degrees of freedom in the body model.

$\Psi$ is a line vector of the form

$$\Psi = [\psi_{\mu_1}, \ldots, \psi_{\mu_N}], \quad 0 \leq \mu_i < 1, \tag{1}$$

where $\psi_{\mu_i}$ represent the joint angles at normalized time $\mu_i$. The posture at a given time $0 \leq \mu_t \leq 1$ is estimated by interpolating the values of the $\psi_{\mu_i}$ corresponding to postures immediately before and after $\mu_t$.

This process produces $M = 324$ (9 subjects, 9 speeds, 4 cycles) angular motion vectors for walking and 140 (5 subjects, 7 speeds, 4 cycles) for running. We form their covariance matrix and compute its eigenvectors $\Theta_{1 \leq i \leq M}$ by Singular Value Decomposition. Assuming our set of examples to be representative, any other motion vector $\Psi$ can be approximated as a weighted sum of the mean motion $\Theta_0$ and the $\Theta_i$:

$$\Psi \approx \Theta_0 + \sum_{i=1}^{m} \alpha_i \Theta_i, \tag{2}$$

where $\alpha_i$ are scalar coefficients that characterize the motion and $m \leq M$. $m$ controls the percentage of the database that can be represented in this manner. This percentage is defined as

$$\sigma = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{M} \lambda_i}, \tag{3}$$

where $\lambda_i$ are the eigenvalues corresponding to the $\Theta_i$ eigenvectors. It is depicted by Figure 1(a) as a function of $m$ for the runnning database. It is taken to be 0.9 for the walking and 0.95 for the running databases, given $m \simeq 10$. The posture at time $\mu_t$ is computed by interpolating the components of the $\Psi$ vector of Equation (2) as discussed above.

The top row of Figure 1 depicts the first three $\alpha_i$ components of the original running motion vectors when expressed in terms of the $\Theta_i$ eigenvectors. Note that the vectors corresponding to specific subjects tend to cluster. The walking database exhibits the same clustering behavior but in higher dimension as depicted in Figure 2. This is due to the fact that the inter-variability between subjects is smaller than for a running motion.

### 3.2. Jumping

To build the jumping database, we also used a Vicon optical motion capture system to capture four people jumping distances ranging from 40 to 120 cm by increments of 40 cm. The data were segmented into key-events, such as start and end of the jump, and sampled at regular time intervals using spherical interpolation, so that each example has $N = 100$ samples. The same procedure as in the case of the walking and running motions can then be applied with $M = 48$ (4 subjects, 3 jumps, 4 trials). The vectors of the jumping database corresponding to specific subjects tend also to cluster, validating the proposed approach, as depicted in the bottom row of Figure 1.

## 4. Motion Generation

In this section, we show how to extrapolate from a motion that is captured after the motion database of Section 3 has been built. This is a two-step process: First, we project the new motion into the PCA space and measure its Mahalanobis distance to each recorded motion corresponding to the same speed or jump length. The generated motion is then taken to be a weighted average of motions at the target speed with the weights being inversely proportional to those distances.

More precisely, let $\Psi^{p,s}$ be the motion vector of Equation 1 corresponding to database person $p$, where $s$ represents either the speed or the jump length. In the remainder of the paper we will refer to $s$ as the *motion parameter*. Each one of these vectors can be approximated by its projection in PCA space $\hat{\Psi}^{p,s}$ computed as

$$\widehat{\Psi}^{p,s} = \Theta_0 + \sum_i \alpha_i^{p,s} \Theta_i, \tag{4}$$

$$\alpha_i^{p,s} = \left(\Psi^{p,s} - \Theta_0\right) \cdot \Theta_i, \tag{5}$$

where $\Theta_i$ are the principal component vectors of Equation (2), and $\alpha_i^{p,s}$ are the scalar coefficients that characterize each motion.

Let $Y^{x,s_1}$ be a vector characterized by motion parameter $s$, corresponding to a motion performed by person $x$ who has not been captured before. The length is arbitrary and we wish to extrapolate motion $Y^{x,s_2}$ of the same person moving with motion parameter $s_2 \neq s_1$ from it. As before, if the motion is cyclic we break $Y^{x,s_1}$ into cycles and perform quaternion spherical interpolation [36] to produce a set of $\Psi^{x,s_1}$ motion vectors of the same dimension as the principal component vectors. If the motion is noncyclic, we identify the start and end of the key-events of the motion and use the resulting vector of taking the frames in between such key-events in the same way as a cycle in a cyclic motion. By projecting one of these cycles into PCA space, we can compute a set of $\alpha_i^{x,s_1}$ weights analogous to those of Equation (5).

Because the influence of each $\Theta_i$ principal component vector is proportional to the corresponding $\lambda_i$ eigenvalue, we use the normalized Mahalanobis instead of the Euclidean distance to compare motion vectors. For each $p$ in the database, we therefore take the distance between $\Psi^{x,s_1}$ and $\Psi^{p,s_1}$ to be

$$d_m(\Psi^{x,s_1}, \Psi^{p,s_1}) = \sqrt{\frac{\sum_i \lambda_i^2 \left(\alpha_i^{x,s_1} - \alpha_i^{p,s_1}\right)^2}{\sum_i \lambda_i^2}}. \tag{6}$$

Kovar [17] defined a more realistic distance function in terms of the distances between meshes. However the cost of using such distance is prohibitive in our context since we would need to evaluate $P * N$ distances between meshes, where $P$ is the total number of subjects and $N$ the number of frames. The distance proposed in this paper requires only $m * P$ norm evaluations, where $m$ is the number of eigenvalues, allowing a real-time animation of multiple subjects. The only

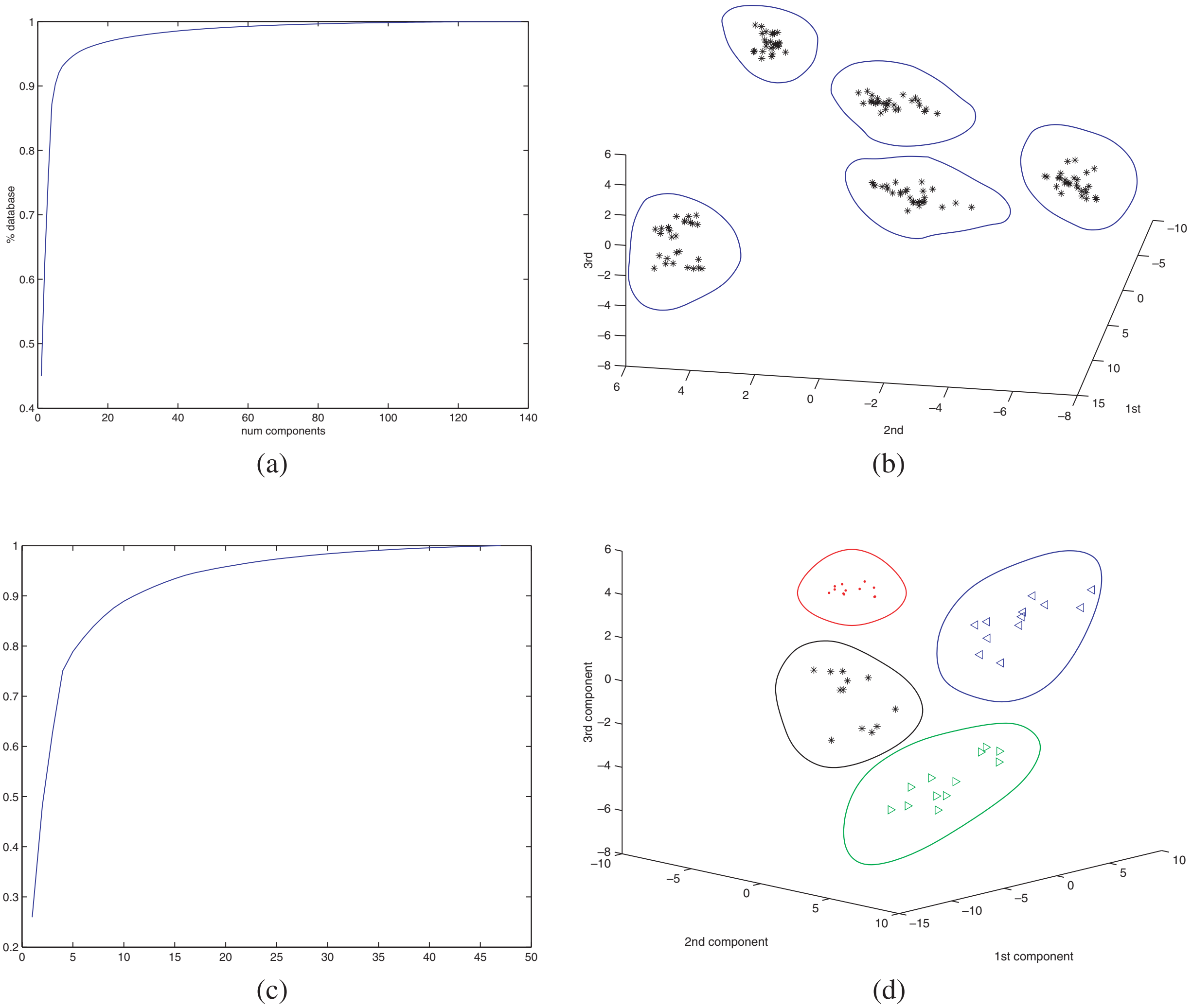

**Figure 1:** *The motion database. (a) Percentage of the database that can be generated with a given number of eigenvalues for the running database. (b) Clustering behavior of the first three $\alpha_i$ coefficients of Equation (2) for the 140 motion vectors measured for five subjects running at speeds ranging from 6 to 12 km/h. They form relatively compact clusters in 3D space that can be used for recognition purposes. (c) Percentage of the database that can be generated with a given number of eigenvalues for the jumping database. (d) Clustering behavior of the first three components for 48 motions vectors measured for four subjects jumping distances ranging from 40 to 120 cm. They cluster in 3D.*

preprocessing needed is the generation of the PCA database, which takes a few seconds.

The weights are then taken to be the normalized inverse of these distances:

$$w^{x,p} = \frac{\left[d_m(\Psi^{x,s_1}, \Psi^{p,s_1})\right]^{-1}}{\sum_q [d_m(\Psi^{x,s_1}, \Psi^{q,s_1})]^{-1}}. \tag{7}$$

This completes the interpolation step of our motion synthesis scheme and we are now ready to generate a new motion. If $s_2$ is one of the motion parameters recorded in the database, we can simply take the new motion $\widetilde{\Psi}_{s_1}^{x,s_2}$ to be an extrapolation of $\Psi^{x,s_1}$:

$$\begin{aligned} \widetilde{\Psi}_{s_1}^{x,s_2} &= \Theta_0 + \sum_i \widetilde{\alpha}_i^{x,s_2} \Theta_i, \\ \widetilde{\alpha}_i^{x,s_2} &= \sum_p w^{x,p} \alpha_i^{p,s_2}. \end{aligned} \tag{8}$$

Otherwise, to produce smooth transitions between motion parameters, we take the $\widetilde{\alpha}_i^{x,s_2}$ coefficients to be Cubic Spline Interpolations of those computed as described above.

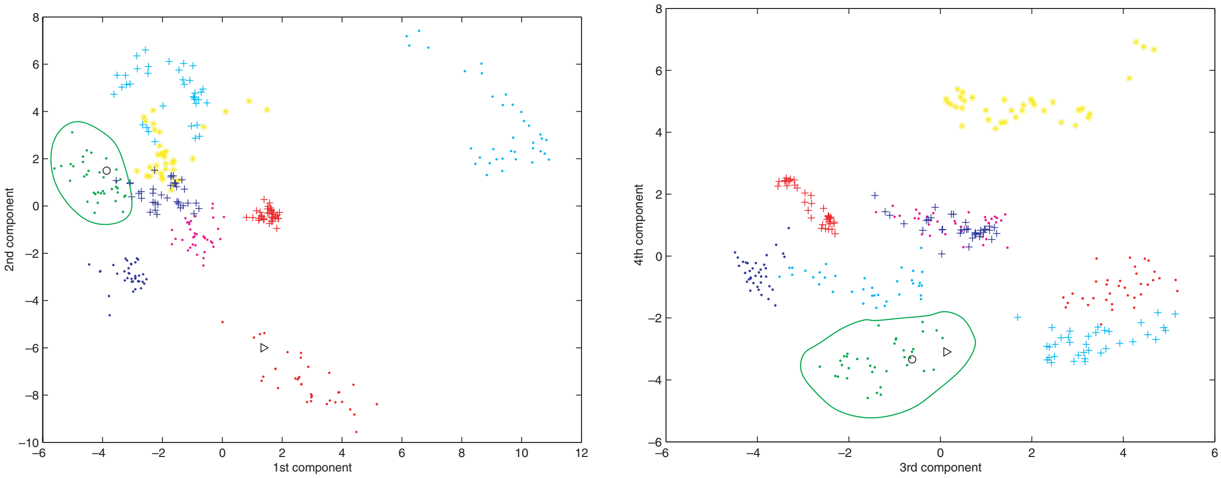

**Figure 2:** *Clustering behavior of the first four PCA components for the walking database. Coefficients corresponding to different subjects are depicted with different colors and symbols. The black circle represents the tracking results for the woman of Figure 10, while the man of Figure 11 is shown as a black triangle. Note that the female subject coefficients are inside her cluster, while the male ones do not belong to any given cluster since his motion is not part of the database.*

## 5. Validation

In this section we use a *cross-validation* framework to validate statistically and visually our approach. For each database subject $p$ in turn, we remove all the $\Psi^{p,s}$ motion vectors that correspond to him or her and perform a new PCA. For any two motion parameters $s_1$ and $s_2$, we can then use the procedure of Section 4 to synthesize $\widetilde{\Psi}^{p,s_2}_{s_1}$ from $\Psi^{p,s_1}$ and compare it to $\widehat{\Psi}^{p,s_2}$, the actual projection of the recorded motion. Ideally, the Mahalanobis distances of these two motions should be zero for all $p$, $s_1$ and $s_2$.

In practice, this can of course never be exactly true. As discussed in Section 3, recall that the database contains, for each subject, several motion cycles at the same speed and that they are never exactly similar to one another.

### 5.1. Animation Results

We now show running and walking animation results obtained by synthetically varying the speed of a motion captured at *one* single speed. These animations are visualized using the time and space normalization described in [4], which allows smooth transitions between motions and adaptation to different human sizes.

To perform size normalization, the root node translation is simply multiplied by the hip joint height $H_i$ of the subject to animate. Recall from Section 3.1 that all input motions have been resampled to a fixed number of frames, $N$. A time normalization stage establishes a correspondence between the elapsed time $\Delta\, t$ and the normalized time $\mu_t$, $0 \leq \mu_t \leq 1$ of Equation (1). Given $F$, the cycle frequency, defined as an adapted version of the Inman law [4], the current normalized time $\mu_t$ is used to defined the frame $f$ to display as:

$$\mu_t = \mu_{t-1} + \Delta t F, \tag{9}$$

$$f = \mu_t N. \tag{10}$$

Figure 3 depicts running at speeds increasing from 6 to 12 km/h. For comparison purposes, we superpose the results with an animation obtained by interpolating the actual motion capture data at *all* the relevant speeds. Note that the two synthetic characters are superposed almost perfectly. To highlight the quality of the results, in the bottom of Figure 3, we superpose two animations corresponding to two *different* women. There the differences are obvious. The same phenomenon can be seen in Figure 4 where we plot the alpha coefficients as a function of speed.

Figure 5 depicts a similar behavior for walking at speeds ranging from 3 to 7 km/h. Again, as can be seen in the top row, the motions generated from a single example and those interpolated using a whole set of examples match very well, except for small discrepancies of the arm motion. As will be discussed below, this can be ascribed to the fact that people do not perform a motion twice in exactly the same fashion.

The same principle can then be applied to jumping motion, but instead of parametrizing as a function of the speed, we parametrize as a function of the jump length. Figure 6 depicts two jumps of 40 and 120 cm generated from a single example of 80 cm. Note that the interpolated results and the original sequence again superpose very well. Once more

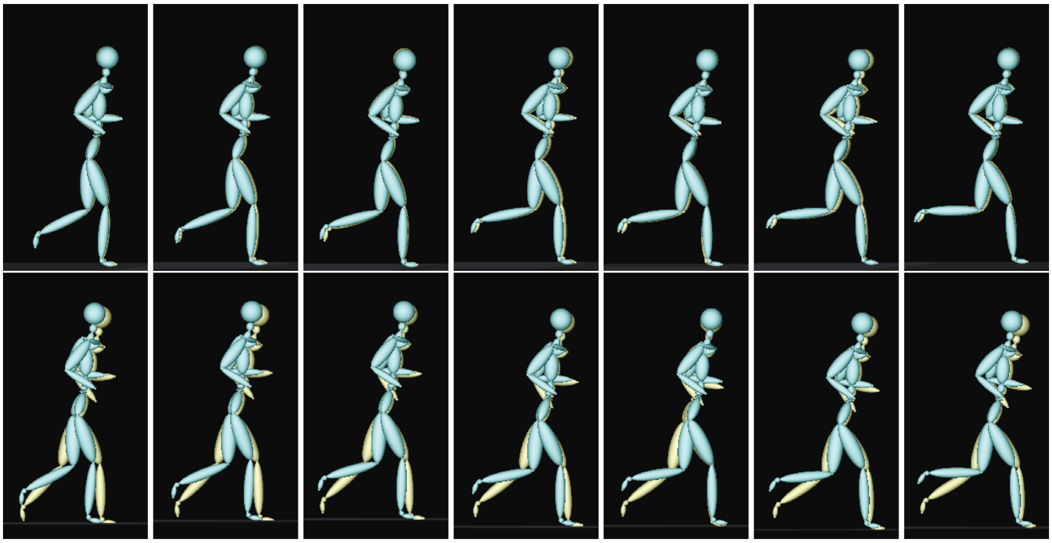

**Figure 3:** *Running at speeds increasing from 6 to 12 km/h.* ***Top row:*** *Superposition of the synthesized motion generated from a single optical motion capture at 6 km/h, in yellow, to an animation observed in the actual motion capture data at all the relevant speeds, in blue. Note that the two synthesized characters are superposed almost perfectly.* ***Bottom row:*** *Superposition of the animations corresponding to two different women. Note the big differences compared to the results in the top row.*

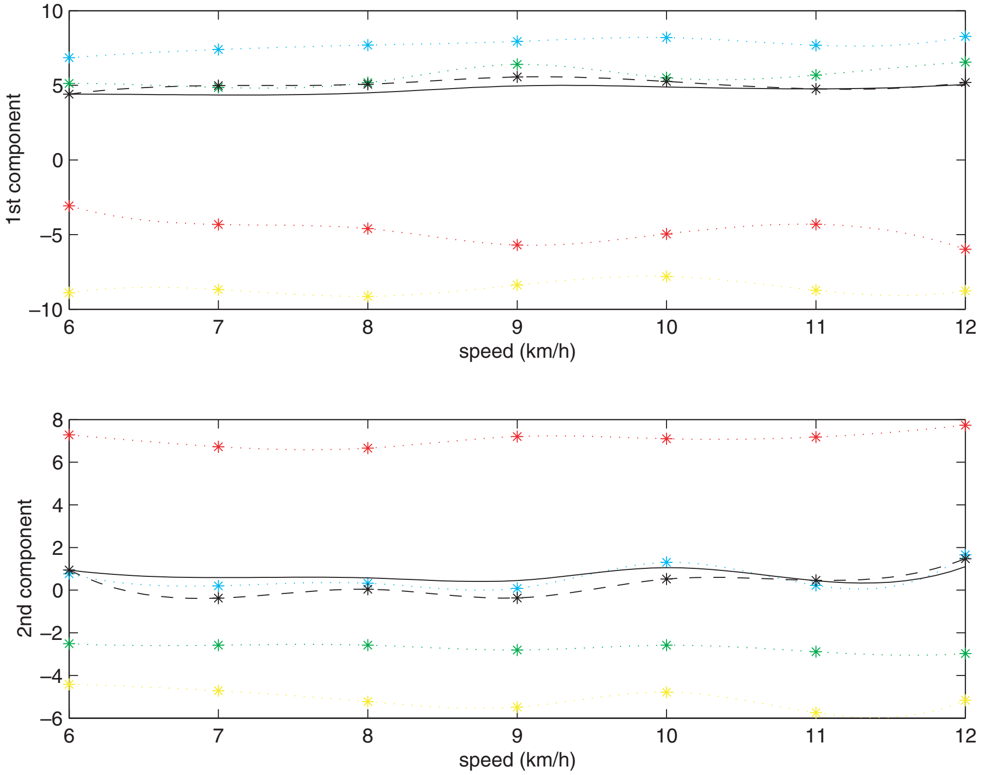

**Figure 4:** *Cubic spline interpolations of the first two components as a function of the speed for the running database. The synthesized motion depicted in Figure 3 is generated from a single optical motion at 6 km/h and is shown in solid black. The original captured motion is depicted in dashed black while the database subjects are shown in different dotted colors.*

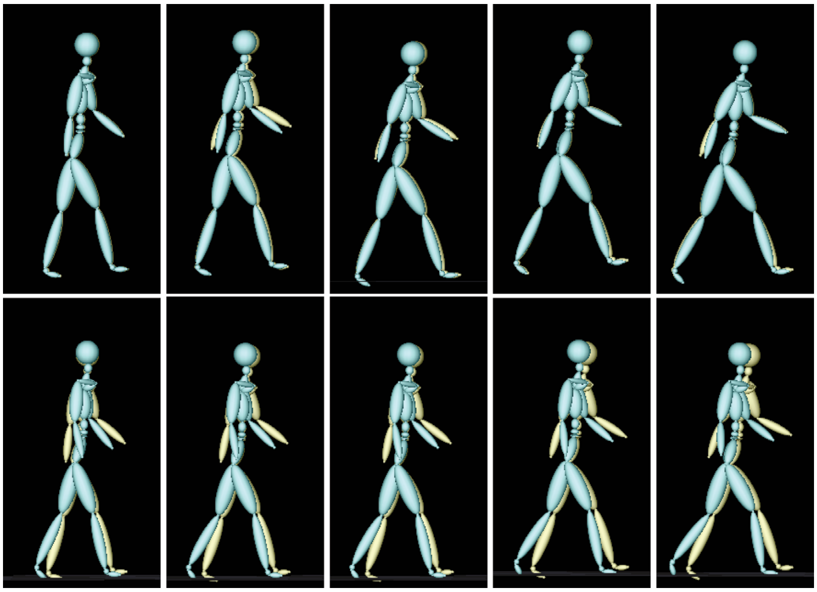

**Figure 5:** *Walking at speeds ranging from 3 to 7 km/h.* ***Top row:*** *Superposition of the motions generated from a single example, in yellow, to those observed in the optical motion capture, in blue. They matched very well except from small discrepancies of the arm motion, that can be ascribed to the fact that people do not perform a motion twice exactly in the same fashion.* ***Bottom row:*** *Superposition of the animation corresponding to two different subjects. Note the big differences compared to the results in the top row. Note that the motion captured has not been corrected. This is the reason why the left leg is penetrating into the floor.*

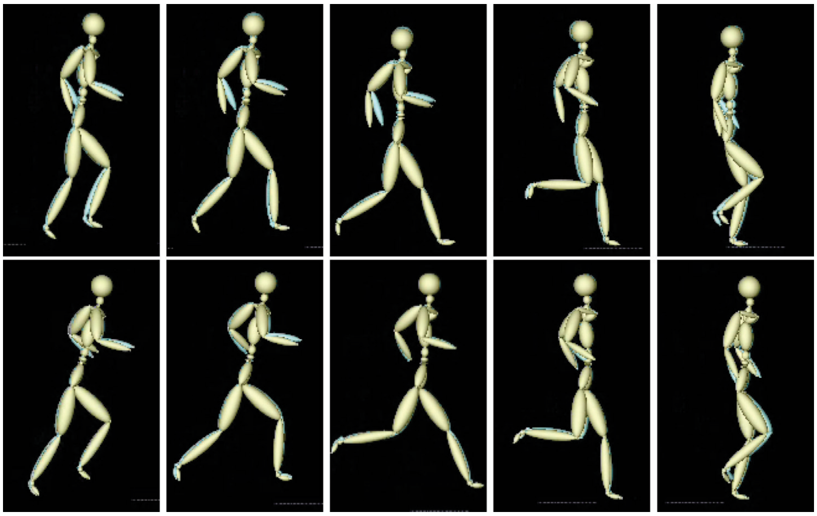

**Figure 6:** *Motion generation from a jump of length 80 cm.* ***Top row:*** *40 cm jump.* ***Bottom row:*** *120 cm jump. We superpose motions generated from a single example (80 cm), in yellow, to those observed by the optical motion capture, in blue. They match very well except for small discrepancies in the arms.*

small discrepancies appear at the arm level. Figure 7 depicts the entire sequence of a 120 cm jump extrapolated from an original 80 cm one. Note that both superpose well during the whole sequence.

The physical correctness of the results could be further improved by correcting the output animations by using standard inverse kinematic techniques to avoid the foot penetration into the floor or sliding effects.

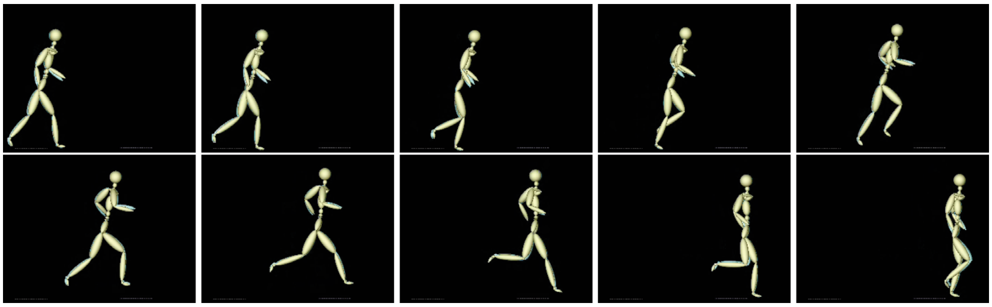

**Figure 7:** *Superposition of the 120 cm jumping motion generated from a single example of length 80 cm, in yellow, to those observed by the optical motion capture, in blue.* ***Top row:*** *Frames 1–40.* ***Bottom row:*** *Frames 41–80.*

### 5.2. Statistical Validation

We now introduce the statistical cross-validation framework we use to validate the results shown above. To this end, we define the following quantitative measures:

- **Interpolation error:** The average over all subjects and pairs of speeds of the normalized Mahalanobis distance between the recorded and synthesized motion vectors discussed above.
- **Intra-variability:** The dispersion between different realization by the same subject of the same motion at the same speed. It is taken to be the mean over all subjects and all speeds of the normalized Mahalanobis distance of the $\widehat{\Psi}^{p,s}$ motion vectors corresponding to different cycles.
- **Inter-variability:** The dispersion between different clusters belonging to different subjects. It is calculated as the mean over all subjects of the distance between motion vectors corresponding to different cycles and speeds.

In Figure 8, we give the Interpolation Errors, Intra-variability and Inter-variability values for our walking, running and jumping databases. Because we perform cross-validation, we take each person out of the databases in turn and therefore list as many values as there are subjects in each one. Note that the interpolation error is consistently larger than the intra-variability but much smaller than the inter-variability. In other words, our motion generation scheme, while not perfect, nevertheless yields motions that are close enough to those of their rightful owner to be associated with him or her rather to anybody else.

The inter-variablity of jumping is bigger than the one of the walking or running because it is more difficult for a subject to perform a jump twice in the same fashion, and to control the jump length while behaving normaly. The interpolation error of the walking database is smaller than the others because the number of subjects in larger. This is also the reason why the inter-variability is smaller: The larger the number of subjects the smaller the distance between clusters.

### 5.3. Stylistic Extrapolation

The motion generation technique presented in Section 4 can now be used to generate walking styles completely different from the standard ones that form the walking database. The principle is the same as before: The new stylized motion is projected into the motion database. Weights are then computed based on the Mahalanobis distance and used to create the same style at a different speed. Figure 9 depicts a sneaking motion at 7 km/h generated by using a single example at 4.5 km/h and the standard walking database. Note that the subject bends her back and increases the step size when the speed increases, which is very realistic since that is what humans do while accelerating the motion in order to keep their balance.

## 6. From Video to Animation

In this section, we show that we can replace the optical motion capture data we have used so far by synchronized video-sequences acquired using an inexpensive commercial product [9]. In the remainder of this section, we first outline briefly the Computer-Vision algorithm [10] we use to extract the PCA coefficients of Equation (2) from the images. We then show that they are accurate enough to produce valid and realistic motions used as input by our motion generation scheme.

### 6.1. Inferring the PCA Coefficients From Video

Most recent tracking techniques presented in the Computer Vision literature rely on probabilistic methods to increase robustness [38–42]. While effective, such approaches require large amount of computation. In previous work [10], we developed an approach that relies on the motion models of Section 3 to formulate the tracking problem as the one of minimizing differentiable objective functions. The body model is composed of implicit surfaces attached to an articulated skeleton. Each primitive defines a field function and the skin is taken to be a level set of the sum of these fields.

| | sub_1 | sub_2 | sub_3 | sub_4 | sub_5 |
|---|---|---|---|---|---|
| Intra Variability | 0.22 | 0.21 | 0.22 | 0.19 | 0.20 |
| Inter Variability | 3.17 | 2.45 | 3.19 | 3.05 | 3.97 |
| Interp Error | 0.38 | 0.78 | 0.92 | 0.34 | 0.51 |

| | sub_1 | sub_2 | sub_3 | sub_4 | sub_5 | sub_6 | sub_7 | sub_8 | sub_9 |
|---|---|---|---|---|---|---|---|---|---|
| Intra Variability | 0.24 | 0.21 | 0.21 | 0.22 | 0.23 | 0.23 | 0.24 | 0.23 | 0.20 |
| Inter Variability | 1.17 | 1.16 | 1.10 | 1.22 | 1.13 | 1.24 | 1.20 | 1.19 | 1.13 |
| Interp Error | 0.35 | 0.62 | 0.76 | 0.47 | 0.35 | 0.63 | 0.49 | 0.55 | 0.60 |

| | sub_1 | sub_2 | sub_3 | sub_4 |
|---|---|---|---|---|
| Intra Variability | 0.50 | 0.63 | 0.52 | 0.63 |
| Inter Variability | 1.82 | 1.88 | 1.73 | 1.97 |
| Interp Error | 0.93 | 0.73 | 0.72 | 0.87 |

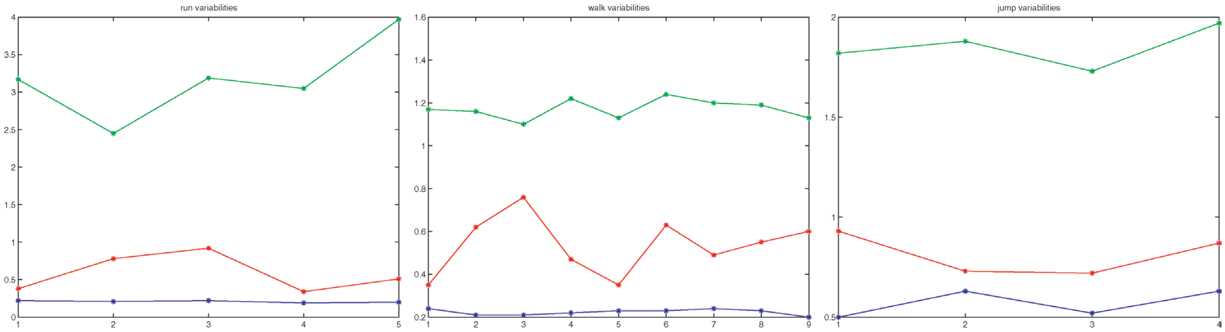

**Figure 8:** *Interpolation Errors, Intra-variability and Inter-variability values.* ***Top row:*** *Running.* ***Second row:*** *Walking.* ***Third row:*** *Jumping.* ***Bottom row:*** *Graphic depiction of the above tables. Note that the interpolation error is consistently larger than the intra-variability but much smaller than the inter-variability in the three databases.*

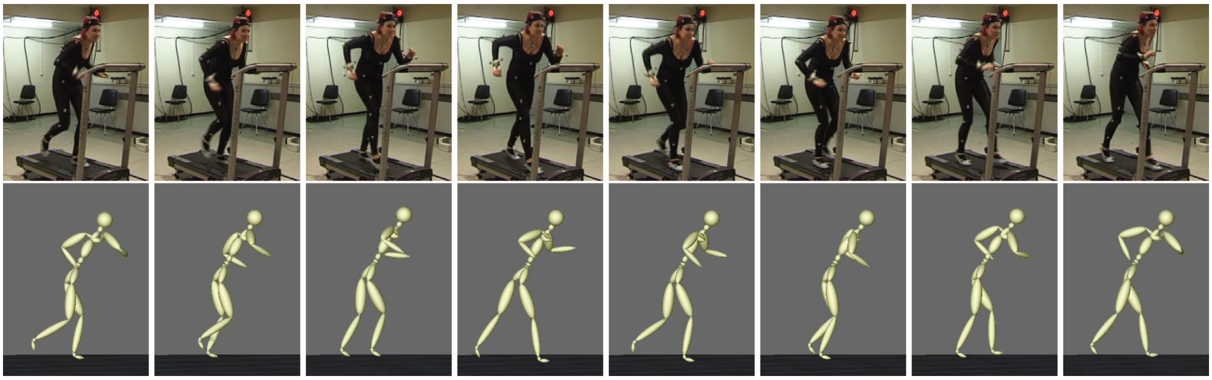

**Figure 9:** *Generation of stylized motion.* ***Top row:*** *Original sneaking walking at 4.5 km/h.* ***Bottom row:*** *Walking with a sneaking style at 7km/h generated using only one example of such style at 4.5 km/h. Note that the database used is composed only of normal walking styles.*

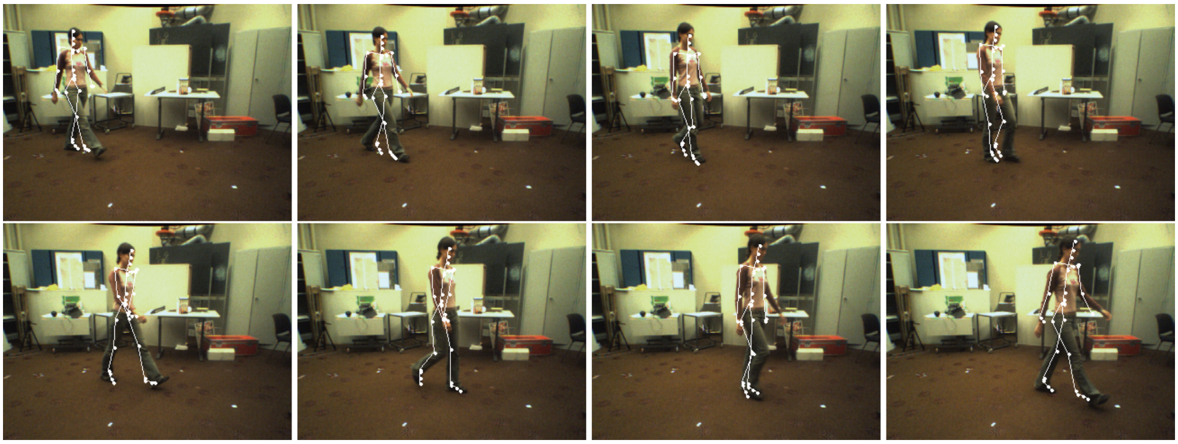

**Figure 10:** *3D motion recovered from a video-sequence of a woman whose motion was also recorded in the database. It is displayed as a stick figure projected into the original low-resolution video-sequence. Corresponding animation results are shown in Figure 12.*

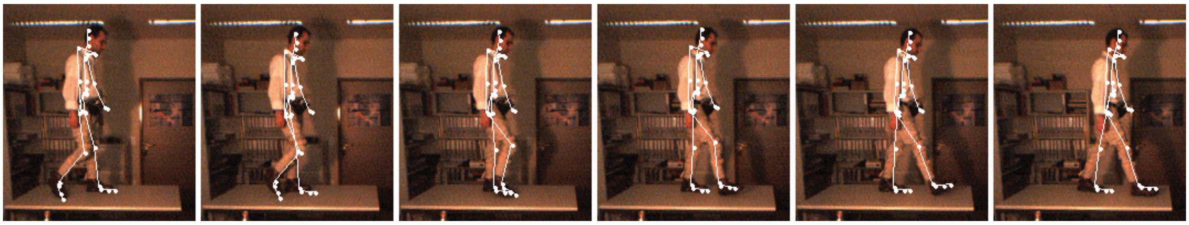

**Figure 11:** *Tracking the walking motion of a man whose motion was not recorded in the database. Animation results for this subject are shown in Figure 13.*

Defining surfaces in this manner let's us define a distance function of data points to the model that is easy to evaluate and differentiable. The structure of these objective functions is rich enough to take advantage of standard deterministic optimization methods and, thus, reduce the computational costs.

Given a $T$ frames video sequence in which the motion of a subject remains relatively steady such as those of Figures 10 and 11, the entire motion can be completely described by the angular motion vector of Equation (2) and, for each frame, a six-dimensional vector $G_t$ that defines the position and orientation of the root body model node with respect to an absolute referential. We therefore take the state vector $\phi$ to be

$$\phi = [G_1, \ldots, G_T, \mu_1, \ldots, \mu_T, \alpha_1, \ldots, \alpha_m], \qquad (11)$$

where $\mu_t$ are the normalized times assigned to each frame and which must also be treated as optimization variables.

In the sequences of Figures 10 and 11, the images we show were acquired using one of the three synchronized cameras used to compute clouds of 3D points via correlation-based stereo. The $\phi$ state vector, and thus the motion, were recovered by minimizing in the least-squares sense the distance of the body model to those clouds in *all* frames simultaneously [10]. Note the good performance of the tracking even though the images are of low resolution. Figure 2 shows the first four components recovered by tracking the motion of Figure 10 in the walking database. Note that the recovered coefficients fall squarely in the cluster corresponding to the subject.

### 6.2. Synthesized Animation

The set of coefficients recovered by the computer vision algorithm is used in the same manner as in Section 4 to generate animations of the subjects shown in Figure 10 and 11 walking at different speeds.

Figure 12 depicts a side view of the optical motions captured from 3 to 7 km/h (yellow) superposed on the synthesized motion extrapolated from the video-sequence of Figure 10 where the person was walking at a speed of 5 km/h. For the legs the correspondance is almost perfect. The small differences in the arms stem from three different reasons: the intra-variability of the walking motion, the low resolution of the images in which the arms contain less than 10 pixels, and the fact that the optically captured motions were performed

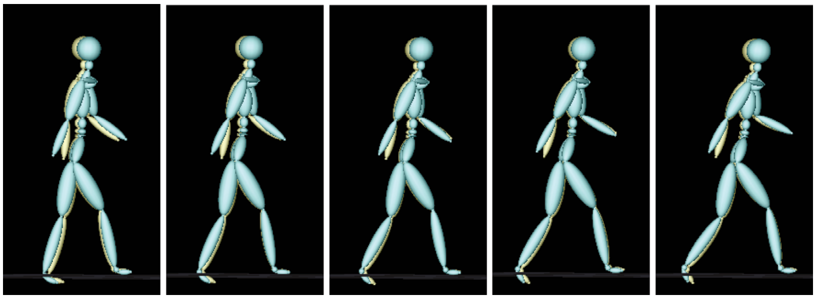

**Figure 12:** *Walking at speeds from 3 to 7 km/h. Superposition of the video of Figure 10, in blue, on a whole set of optical motion capture examples, in yellow. The leg motion matches almost perfectly, which results in the lower body of the two figures being almost perfectly superposed. Small discrepancies in the arms are due to the intra-variability of the motion, the low resolution of the video, and the fact that the optical motions were performed using a treadmill. Note that the motion capture data has not been cleaned-up, which is why the left leg is penetrating the floor.*

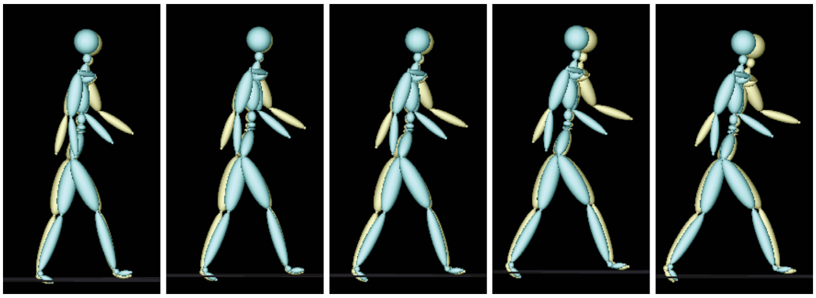

**Figure 13:** *Side view of the original and synthesized motion from 3 to 7 km/h for the subject of the video sequence of Figure 11. The subject is compared to its closest neighbor in the database according to the Mahalanobis distance. The original motion capture data is represented in blue, and the synthesized one in yellow. They are quite different, which goes to show that our approach can generate a wide range of realistic styles.*

using a treadmill which is not the case for the video-sequence. Note that the motion capture data has not been corrected, which is why the left leg is penetrating into the floor.

Finally, we show how our synthetization framework can generate movements from a large space by tracking a subject that is not part of the database depicted in Figure 11 at 3 km/h and by synthesizing his movement from 3 to 7 km/h. The motion remains natural and physically possible even though it is *different* from all the recorded motions. In Figure 13, we highlight this difference by superposing the generated motion to the closest one in the database, which is clearly dissimilar. This was to be expected because, as shown in Figure 2, the corresponding alpha coefficients do not match any of the clusters that correspond to specific individuals.

## 7. Conclusion

We have presented a real-time motion generation technique that allows us to generate the motion of a particular individual performing parameterized displacement activities. More specifically, we have investigated the case of walking, running and jumping. The first two are cyclical and parametrized by speed. The third one is noncyclical and parametrized in terms of jump length. Given one single example, we can modify the speed, length or body size while preserving the individual's specific style. The required example can be obtained using either a sophisticated optical motion capture system or a much simpler set of synchronized cameras. While we have only validated our approach in the case of three specific cyclic and noncyclic motions, we believe it to be fully generic and applicable to a whole range of activities.

Currently, the crucial limitation of the method comes from the fact that we have not investigated the curvilinear motion patterns that would be required by a complete system to blend the straight line motion sequences we synthesize. We did not include such motions into our database because they cannot be captured on a treadmill and therefore require a more complex experimental setup than the one we have. However, since they are also controlled by a well-identified parameter, namely the radius of curvature, we believe them to be amenable to our approach. Of course, increasing our repertoire of motions could result in nonlinearities, which may require us to replace PCA by more sophisticated statistical tools such as Isomap [43] or Gaussian Processes [44], that allow the same kind of treatment for nonlinear models.

Another area that requires further investigation is the combination of our method with inverse kinematics technique to clean up the artifacts, such as foot sliding, which can be observed in some of our results. The simplest approach would be to use Inverse Kinematics in a post-processing step. A more ambitious approach would be to detect foot support phases in real-time and enforce them with an IK solver [13].

## References

1. W. Müller and M. Alexa. Representing animations by principal components. In *Computer Graphics Forum (Eurographics 2000)*,M. Gross and F. R. A. Hopgood (eds.), vol. 19(3), pp. 411–426, 2000.

2. N. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision* 2(5):371–387, 2002.

3. I. S. Lim and D. Thalmann. Construction of animation models out of captured data. In *Procedings of the IEEE Conference of Multimedia and Expo*, August 2002.

4. P. Glardon, R. Boulic and D. Thalmann. A coherent locomotion engine extrapolating beyond experimental data. In *Proceedings of CASA*, 2004.

5. I. T. Jolliffe. *Principal Component Analysis*. Springer series in statistics. Springer-Verlag, 1986.

6. C. Rose, M. F. Cohen and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18(5):32–41, 1998.

7. L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *ACM Symposium on Computer Animation*, pp. 214–224, July 2003.

8. M. Brand and A. Hertzmann. Style machines. *Computer Graphics, SIGGRAPH Proceedings*, 183–192, July 2000.

9. Digiclops Stereo Vision camera system. Point Grey Research Inc., Vancouver, Canada. Available at http://www.ptgrey.com/products/digiclops/.

10. R. Urtasun and P. Fua. 3D human body tracking using deterministic temporal motion models. In *European Conference on Computer Vision*, Parague, Czech Republic, May 2004.

11. F. Multon, L. France, M. Cani-Gascuel and G. Debunne. Computer animation of human walking: A survey. *Journal of Visualization and Computer Animation* 10(1):39–54, 1999.

12. M. Girard. Interactive design of 3-d computer-animated legged animal motion. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics*, ACM Press, pp. 131–150, 1987.

13. P. Baerlocher and R. Boulic. An inverse kinematics architecture for enforcing an arbitrary number of strict priority levels. *The Visual Computer*, Springer Verlag, 20(6), pp. 402–417, 2004.

14. C. Liu and Z. Popovic. Synthesis of complex dynamic character motion from simple animations. In *Computer Graphics, SIGGRAPH Proceedings*, 2002.

15. J. K. Hodgins, W. L. Wooten, D. C. Brogan and J. F. O'Brien. Animating human athletics. *Computer Graphics, SIGGRAPH Proceedings* 71–78, 1995.

16. W. L. Wooten and J. K. Hodgins. Simulating leaping, tumbling, landing and balancing humans. In *Proceedings of IEEE International Conference on Robotics and Automation*, April 2000.

17. L. Kovar, M. Gleicher and F. Pighin. Motion graphs. In *Computer Graphics, SIGGRAPH Proceedings*, pp. 473–482, July 2002.

18. J. Lee, J. Chai, P. Reitsma, J. Hodgins and N. Pollard. Interactive control of avatars animated with human motion data. *Computer Graphics, SIGGRAPH Proceedings*, 2002.

19. D. Forsyth and O. Arikan. Interactive motion generation from examples. *Computer Graphics, SIGGRAPH Proceedings* 21:3, 2002.

20. M. Unuma, K. Anjyo and R. Takeuchi. Fourier principles for emotion-based human figure. *Computer Graphics, SIGGRAPH Proceedings* 91–96, August 1995.

21. M. Gleicher. Comparing constraint-based motion editing methods. *Graphical Models*, 63(2):107–134, March 2001.

22. M. Gleicher. Motion editing with spacetime constraints. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pp. 139–148, April 1997.

23. J. Lee and S.Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Computer Graphics, SIGGRAPH Proceedings*, pp. 39–48, August 1999.

24. Z. Popovic and A. Witkin. Physically Based Motion Transformation. *Computer Graphics, SIGGRAPH Proceedings*, 11–20, 1999.

25. A. Bruderlin and L. Williams. Motion signal processing. *Computer Graphics, SIGGRAPH Proceedings*, 97–104. August 1995.

26. K. Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics* 1(1), March 1995.

27. A. Golam and K.C. Wong. Dynamic time warp based framespace interpolation for motion editing. In *Graphics Interface*, pp. 45–52, May 2000.

28. M. Dontcheva, G. Yngve and Z. Popovic. Layered acting for character animation. *Computer Graphics, SIGGRAPH Proceedings*, 2003.

29. S. Park, H. J. Shin and S. Shin. On-line locomotion generation based on motion blending. In *ACM Symposium on Computer Animation*, 2002.

30. P.-P. J. Sloan, C. F. Rose and M. F. Cohen. Shape by example. In *Symposium on Interactive 3D Graphics*, pp. 135–144, 2001.

31. H. J. Shin, J. Lee, S. Y. Shin and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Transactions on Graphics* 20(2):67–94, 2001.

32. V. Blanz, C. Basso, T. Poggio and T. Vetter. Reanimating faces in images and video. In *Eurographics*, Granada, Spain, September 2003.

33. V. Blanz and T. Vetter. A morphable model for the synthesis of 3–D faces. In *Computer Graphics, SIGGRAPH Proceedings*, Los Angeles, CA, August 1999, pp. 187–194.

34. M. Dimitrijevic, S. Ilic and P. Fua. Accurate face models from uncalibrated and ill-lit video sequences. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, June 2004.

35. Vicon Optical Motion Catpure system. Vicon Motion System Ltd., Oxford, UK. Available at http://www.vicon.com/.

36. K. Shoemake. Animating rotation with quaternion curves. *Computer Graphics, SIGGRAPH Proceedings*, 19:245–254, 1985.

37. M. P. Murray. Gait as a total pattern of movement. *American Journal of Physical Medicine* 46(1):290–333, 1967.

38. M. Isard and A. Blake. CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision* 29(1):5–28, August 1998.

39. J. Deutscher, A. Blake and I. Reid. Articulated body motion capture by annealed particle filtering. In *Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, 2000, pp. 2126–2133.

40. A. J. Davison, J. Deutscher and I. D. Reid. Markerless motion capture of complex full-body movement for character animation. In *Eurographics Workshop on Computer Animation and Simulation*, Springer-Verlag LNCS, 2001.

41. K. Choo and D. Fleet. People tracking using hybrid monte carlo filtering. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.

42. C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3D human tracking. In *Conference on Computer Vision and Pattern Recognition*, vol. I, Madison, WI, June 2003.

43. J. Tenenbaum, V. de Silva and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323, 2000.

44. K. Grochow, S.L. Martin, A. Hertzmann and Z. Popovic. Style-based inverse kinematics. *Computer Graphics, SIGGRAPH Proceedings* 522–531, April 1997.