SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2014, Vol. 57 072113:1–072113:11
doi: 10.1007/s11432-013-4898-2

Real-time control of human actions using inertial sensors

LIU HuaJun[1], HE FaZhi[1*], ZHU FuXi[1] & ZHU Qing[2]

[1]*School of Computer, Wuhan University, Wuhan* 430072*, China;*
[2]*School of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu* 610031*, China*

Received March 6, 2013; accepted March 26, 2013; published online June 20, 2013

Abstract Our study proposes a new local model to accurately control an avatar using six inertial sensors in real-time. Creating such a system to assist interactive control of a full-body avatar is challenging because control signals from our performance interfaces are usually inadequate to completely determine the whole body movement of human actors. We use a pre-captured motion database to construct a group of local regression models, which are used along with the control signals to synthesize whole body human movement. By synthesizing a variety of human movements based on actors' control in real-time, this study verifies the effectiveness of the proposed system. Compared with the previous models, our proposed model can synthesize more accurate results. Our system is suitable for common use because it is much cheaper than commercial motion capture systems.

Keywords avatars, motion capture/editing/synthesis, interaction techniques, game interaction, animation simulation

Citation Liu H J, He F Z, Zhu F X, et al. Real-time control of human actions using inertial sensors. Sci China Inf Sci, 2014, 57: 072113(11), doi: 10.1007/s11432-013-4898-2

1 Introduction

The ability to synthesize human action precisely in real-time can give a user/trainee the chance to control a virtual avatar using his/her own body movements, navigate the virtual world, or accomplish a virtual task. Such a system could also be used in real sports training, rehabilitation, and real-time control of game characters or robotic systems such as tele-operation. The challenge has already been partially solved by commercial motion capture (mocap) equipment, however, it is quite expensive for common use. Because the systems generally require the performer to wear skin-tight clothing along with no less than 40 retro-reflective markers, 18 magnetic or inertial sensors, or a full-body exoskeleton, they are cumbersome for actors.

Recently, major game console companies, including Microsoft, Sony, and Nintendo, have developed next generation hardware devices to capture the online performance of individual players. These control interfaces are suitable as performance interfaces because of their low cost and unobtrusiveness. However, control signals from these devices are often noisy and low-dimensional, and therefore cannot be used to control human movement accurately.

[*] Corresponding author (email: fzhe@whu.edu.cn)

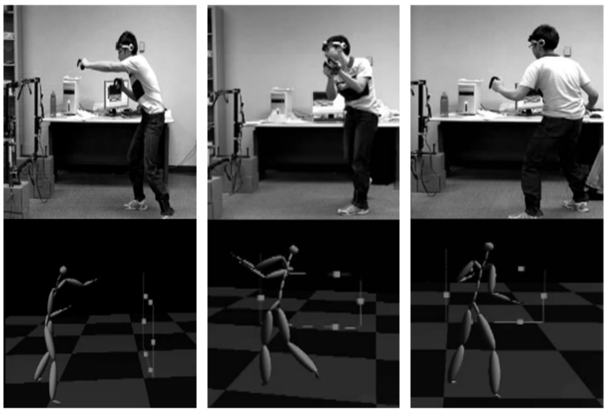

Figure 1 The actor wearing six sensors to accurately control a virtual character.

This study presents a new approach to performance animation that uses six inertial sensors to create a system to control an entire avatar accurately (see Figure 1). Our system employs inertial sensors because they are low-cost, compact, and highly accurate. However, constructing a performance animation interface is challenging because control signals from the equipment are fairly low-dimensional, and often inadequate for determining the full-body movement of actors. (Usually, more than fifty degrees of freedom (DOF) are used in representing a virtual human character.)

Our approach is to teach an online dynamic model using a pre-captured motion database and employ it to constrain the synthesized pose to look natural. The proposed model predicts the current pose $\boldsymbol{q}_t$ using its previous m poses $\boldsymbol{q}_{t-1},\ldots,\boldsymbol{q}_{t-m}$ using a group of mathematical functions. Generally, it is difficult to predict how people move because human action is highly nonlinear. Instead of teaching a global dynamic motion model, which is often not appropriate for modeling the nonlinear properties of complex human movements, this study proposes to construct online local regression models.

While running, we use the K-nearest neighbor search method to find the K closest sequences that are similar to the recently synthesized poses from the pre-captured database. These examples along with their subsequent poses are employed as training data to obtain a prediction function that finds the relationship between the previous m poses and the current pose. At each moment, our system produces a new local model for the next pose. The proposed model is effective for human motion because it takes the heterogeneity of a pre-captured database into full consideration. Using the constrained maximum a posteriori (MAP) inference approach, the problem of online motion reconstruction is formulated using a priori information from online local models together with a likelihood term imposed by the control signals.

2 Related work

• **Performance animation interfaces.** Commercial mocap equipment is one of the most popular technologies used to control a virtual character. The mocap system is based on passive/active optical, exoskeleton, and magnetic sensors, all of which are capable of carrying out real-time capture of an actor's movements. However, the systems are quite expensive when used by large numbers of actors. In addition, they are very complex, cumbersome and tedious, because they require the performer to wear skin-tight clothing along with more than 40 retro-reflective markers which must be carefully positioned, 18 magnetic or inertial sensors, or an exoskeleton.

Synthesizing total character motion using only a few sensors has already been well explored. To control a standing avatar in real-time, Badler [1] proposed the inverse kinematics (IK) method together with four magnetic sensors. To reduce the kinematic redundancy, a heuristic method was used in their approach. In contrast, our study adopts a data-driven approach. Compared to Badler's solution, Semwal [2] used eight magnetic sensors and added an analytic method to the IK technique. Using a foot pressure sensor, Yin [3] created a system that searches and duplicates motion from a prerecorded database. However, the method can barely reconstruct whole human movements for a narrow range of motions, and foot pressure sensors cannot provide enough information to synthesize an upper body action accurately. Using five cheap inertial sensors, Slyper's method [4] can control and synthesize upper-body movement. Different from their work, we could control whole body human actions using just six sensors. In addition, we created a group of time-varying local models to constrain and synthesize real-time human motion rather than search for the closest example and implement it. Recently, Ha [5] created a system, which used one foot pressure sensor and Sony PlayStation Move to reconstruct upper-body motion, which was similar to Slyper's method. However, our method can achieve not just upper-body but full-body control. Tautges [6] used accelerometers as signal providing equipment, and his method exceeded space limitations, but accelerometers cannot provide positional information, so the reconstructed results were not natural enough. Compared with our method in this study, their method cannot achieve accurate motion control. Liu [7] reached full-body human motion control using six inertial sensors, however, we used a more powerful model to enable dimensionality reduction and achieve better results. More recently, some researchers developed an accurate human motion control method using depth information provided by a depth camera [8,9], whereas, our interactive human motion control system is based on six inertial sensors. Unlike vision sensors, inertial sensors do not suffer from occlusion problems. In this study, we focus on marker-based approaches.

- **Data-driven animation.** A number of data-driven approaches were developed. Typically, we use three extremely different approaches: interpolating [10–12], motion graph [13–18] and a statistical modeling constraint approach [7, 19–25]. Because the first two approaches cannot achieve real-time requirements, we constructed a statistical dynamic model to predict the current pose using previous synthesized poses. To date, statistical motion models have been widely applied to synthesize realistic human motion. They can be used for inverse kinematics [20], interactively control human action using several retro-reflective markers [19], perturbations of natural-looking human motion [21], using manipulation interfaces to edit human motion [23], using the Gaussian process latent variable model (GPLVM) to synthesize human motion [25], performance animation using a global model [22], real-time motion control using a local principle component regression (PCR) model [7], building physically-valid motion models for human motion synthesis [24], and others.

Among the above-mentioned statistical models, ours is extremely similar to local models constructed in the subspace for online control of human motions [7,19], because all of them were built during runtime and based on training data which were close to the current example. Nevertheless, there is a very important difference. For regression learning approaches, the training data can be divided into two parts: input and output data. The principle component analysis (PCA) model used in [19] only focuses on the dimensionality reduction of the input data, and the PCR model used in [7] focuses on the dimensionality reduction of both the input and output data. However, these two models fail to recognize the projection relationship between the input and output training data. The model we propose in this study estimates an input-output projection with a linear combination of basis regression equations or functions, therefore, it can add more spatial-temporal relationships consistently in a pre-captured database than previous models. Our test in Section 7 shows that the proposed model can synthesize more natural-looking human actions than previous local pose models. In addition, the online local dynamic models make it easier to find suitable structures for high-dimensional global models.

3 Overview of performance interface

Our performance interface automatically transforms control inputs from six inertial sensors into realistic human actions by building sequential local models during runtime and then using them to interpret the performer's action (see Figure 2). Our performance interface includes the following components.

- **Calibration of the local coordinates for sensors and skeletal sizing.** A calibration step is implemented for two reasons: one is that different actors have different skeletal size; the other is that for the same user, the way he/she wears the sensors may vary, so our system needs to map the coordinates of each sensor to the control coordinates of the user's body. Thus, a new calibration approach is introduced,

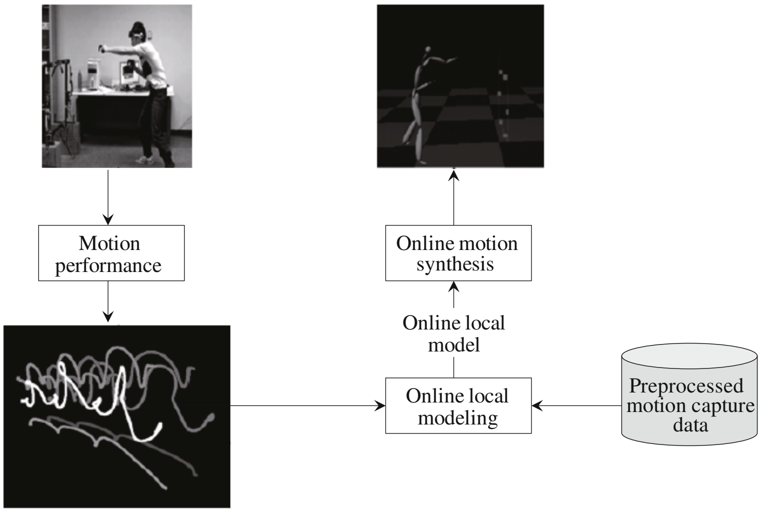

Figure 2 System overview. An actor wearing six inertial sensors performs the desired motions using the InterSense IS-900 system. The motion performance step automatically reconstructs the 3D orientation and position of each sensor for every time step in real-time. While running, the performance interface automatically transforms control signals into high quality human motion using a motion database.

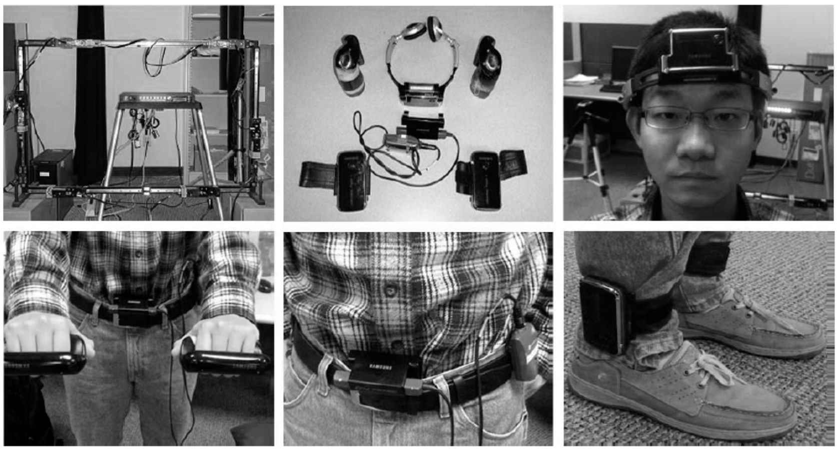

Figure 3 Sensors for our avatar control system.

which is robust to both different users and various sensor placements. Our performance interface requires the user to wear six inertial sensors on his/her head, center of torso, both hands and both ankles for performance-driven animation, as shown in Figure 3. By guiding the user to complete eight "calibration" poses, the calibration step can estimate the user's skeletal size and each sensor's local coordinates at the same time.

• **Online modeling of human dynamic behavior.** A novel statistical local model is presented for our online motion synthesis. Our performance interface uses sequential local linear models which are constructed from a pre-captured database to model various human actions on the fly. One advantage of modeling is that our proposed models have the ability to predict the movements of actors in local regions of the configuration space.

• **Online motion reconstruction.** While running, the actor performs the desired motion using six inertial sensors. The global 3D orientations and 3D positions of all sensors are recorded simultaneously using our performance interfaces, $[\boldsymbol{c}_1,\ldots,\boldsymbol{c}_t]$. This information is useful because it describes the trajectories of special points and vectors on the body of the avatar. By combining the current control signals $\boldsymbol{c}_t$ provided by the sensors and the constructed local probabilistic model based on previous m reconstructed poses $\tilde{Q} = [\tilde{\boldsymbol{q}}_{t-1}, \ldots, \tilde{\boldsymbol{q}}_{t-m}]$, our system can synthesize the user's pose $\boldsymbol{q}_t$ in a constrained MAP framework:

$$\max_{\boldsymbol{q}_t} \Pr\left(\boldsymbol{q}_t | \boldsymbol{c}_t, \tilde{Q}\right) \propto \max_{\boldsymbol{q}_t} \Pr(\boldsymbol{c}_t|\boldsymbol{q}_t) \cdot \Pr\left(\boldsymbol{q}_t|\tilde{Q}\right). \tag{1}$$

By applying the negative log to the posteriori distribution function $\Pr(\boldsymbol{q}_t|\boldsymbol{c}_t, \tilde{Q})$, we can convert the constrained MAP problem into an energy minimization problem:

$$\min_{\boldsymbol{q}_t} \underbrace{-\ln \Pr(\boldsymbol{c}_t|\boldsymbol{q}_t)}_{E_{\text{control}}} + \underbrace{-\ln \Pr\left(\boldsymbol{q}_t|\tilde{Q}\right)}_{E_{\text{prior}}}, \tag{2}$$

where E_{control} is the likelihood term that measures the extent to which the synthesized pose $\boldsymbol{q}_t$ matches the current signals $\boldsymbol{c}_t$, and E_{prior} is the prior term that describes the prior distribution of human motion. Conceptually, the prior term tests the naturalness of the reconstructed pose.

The calibration step is completed offline, however, the motion modeling and reconstruction process are executed online. In the following sections, we give a detailed description of these three components.

4 Calibration of local coordinates for skeletal size and sensors

An InterSense IS-900 system was used to record 3D orientation/position data of all inertial sensors (40 fps) in real-time from our performance interfaces. The IS-900 processes motion signals from a tracking device to compute 3-DOF orientation and position data, where the orientation data is integrated from magnetometers, gyroscopes, and accelerometers, and the position data is provided by ultrasonic sensors. The calibration step proposed above ensures that the performance interfaces are adequate for various sensor placements and for actors with different skeletal lengthes. Furthermore, the *skeletal size calibration* step aims to calculate the size of the actor's skeleton and the *sensors' local coordinates calibration* step computes each inertial sensor's local coordinates.

Eight "calibration" poses are used for calibration. The interface guides the user to perform the same pose as the green pose (target pose) which is shown on the screen, and the performance interface records the global orientation and location of inertial measurement sensors under these calibration poses (see the accompanying video). To reduce the ambiguity in the process of modeling a human skeleton, a low-dimensional eigen model is built based on data from a human skeleton. All skeletal data in our experiments are from the Carnegie Mellon University (CMU) motion capture database[1)] and are represented in the format of the Acclaim Skeleton File.

The skeletal size is represented by a vector $\boldsymbol{s}$ which records each bone's length. The vectors $\boldsymbol{o}_j$ and $\boldsymbol{p}_j$ which are related to the sensor's local coordinate systems, are used to represent the orientations and positions of the jth inertial measurement sensor. The vectors $\boldsymbol{q}_i,\ i = 1, \ldots, 8$ are used to represent the calibration poses. Hence, we can now solve the nonlinear optimization problem for our calibration step:

$$\arg \min_{\{\boldsymbol{o}_j\},\{\boldsymbol{p}_j\},\{\lambda_h\},\boldsymbol{s}} \sum_i \sum_j \left\| \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{o}_j; \boldsymbol{q}_i) - \boldsymbol{d}_i^j \right\|^2 + \left\| \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{p}_j; \boldsymbol{q}_i) - \boldsymbol{l}_i^j \right\|^2 + \alpha \left\| \boldsymbol{s} - \boldsymbol{e}_0 - \sum_{h=1}^{H} \lambda_h \boldsymbol{e}_h \right\|^2. \tag{3}$$

In the above formula, given the orientations $\boldsymbol{o}_j$ and positions $\boldsymbol{p}_j$ of the jth sensors, and the actor's skeletal size, $\boldsymbol{s}$, the forward kinematics (FK) function $\boldsymbol{f}$ calculates the calibration poses' joint angles $\boldsymbol{q}_i$, The vectors $\boldsymbol{d}_i^j$ and $\boldsymbol{l}_i^j$ are the recorded global directions and locations of the jth inertial sensor for the ith calibration pose. The scalar α weighs the importance of the skeletal model priors learned from the prerecorded data. The vector $\boldsymbol{e}_0$ is the mean value of skeletal model, and $\boldsymbol{e}_h,\ h = 1, \ldots, H$ are the eigen vectors. To calculate $\boldsymbol{o}_j$, $\boldsymbol{p}_j$, and $\boldsymbol{s}$, we run an optimization calculation using the Levenberg-Marquardt algorithm [14].

[1)] http://mocap.cs.cmu.edu.

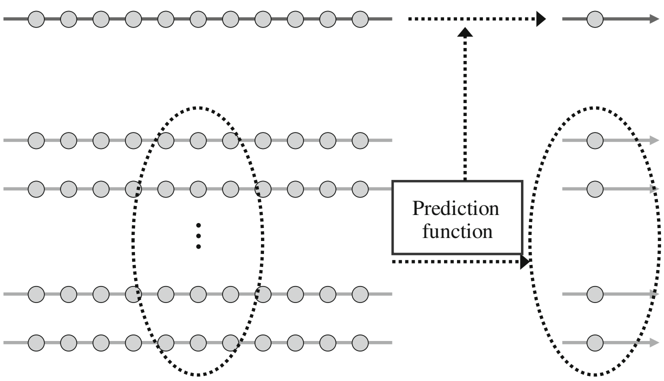

Figure 4 The key concept of our online local modeling. The points on the top line are recently reconstructed poses, and the other lines are the K motion examples from the motion capture database that are close to the recently reconstructed poses. We establish the relationships for these K motion examples and their subsequent poses to predict the next pose on the top line.

5 Online modeling of human dynamic behavior

The motion control problem is definitely challenging because the information from six inertial sensors attached to a user cannot fully constrain a full-body avatar's joint angles, because the control signals are of low-dimensionality while the full-body joint angles are of high-dimensionality. Our approach is to automatically build sequential online local regression models to adequately constrain the synthesized pose within the natural-looking solution space.

We assume human action can be represented by an m-order Markov chain, so the current pose $\boldsymbol{q}_t$ can be considered to depend only on previous m poses: $\Pr(\boldsymbol{q}_t|\boldsymbol{q}_{t-1},\ldots,\boldsymbol{q}_1) = \Pr(\boldsymbol{q}_t|\boldsymbol{q}_{t-1},\ldots,\boldsymbol{q}_{t-m})$. Nevertheless, modeling the dynamic behavior of human motion is difficult because human action is nonlinear, and a global dynamic model may not be sufficient to model complex movement. To solve this problem, sequential local regression models are constructed on the fly to predict how humans move.

To predict the current pose at frame t, the first step is to search the motion database captured in advance, and find the motion segments closest to the recently constructed motion segment $\tilde{Q} = [\tilde{\boldsymbol{q}}_{t-1},\ldots,\tilde{\boldsymbol{q}}_{t-m}]$. The K closest motion segments $[\boldsymbol{q}_{t_k-1},\ldots,\boldsymbol{q}_{t_k-m}]$, along with their subsequent poses q_{t_k}, $k=1,\ldots,K$, which are then used as training data to learn a prediction function $\boldsymbol{g}$ that maps the previous m poses to the current pose, as shown in Figure 4.

Suppose a linear relationship exists between an input joint angle vector $\boldsymbol{x} = [\boldsymbol{q}_{t-1},\ldots,\boldsymbol{q}_{t-m}]$ and an output joint angle vector $\boldsymbol{y} = \boldsymbol{q}_t$. For simplification, the function of the proposed model which is represented using linear regression is

$$\boldsymbol{y} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{x} + \beta_y, \tag{4}$$

where the input joint angle $\boldsymbol{x}$ is an $m \times D$-dimensional vector. D represents the dimension of the DOF for a human character and $\boldsymbol{y}$ is the joint angle value for the output. Regression coefficients $\boldsymbol{\alpha}$ are vectors, and β_y represents a homoscedastic noise variable, which is independent of vector $\boldsymbol{x}$. Moreover, given the K motion examples $\{(\boldsymbol{x}_k;y_k)\}$, $k=1,\ldots,K$, which are similar to the current synthesized poses, and by minimizing the expected error $E=\sum_{k=1}^{K}\|y_k-\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{x}_k\|^2$, we can obtain the coefficients $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}. \tag{5}$$

The row of the matrix $\boldsymbol{X}$ includes the input joint angle vectors $\boldsymbol{x}_k$, $k=1,\ldots,K$, and the K output joint angle values are stacked in vector $\boldsymbol{y}$.

In our implementation, first, we put input motion data $\boldsymbol{X}$ and output motion data $\boldsymbol{y}$ together, represented as $\boldsymbol{A} = [\boldsymbol{X}\boldsymbol{y}]$, and principle component analysis is then applied to $\boldsymbol{A}$. Thus, the eigenvectors can be extracted from the covariance matrix $\boldsymbol{C} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$. Therefore, the principal subspace contains the direction of the joint angle data distribution. When we perform dimensionality reduction, we prove that the directions existing in the input joint angle space have highly predictive values. In our implementation, by mapping input motion joint angle data as close as possible to the principal subspace, this subspace is directly used for regression operation. We can decompose the eigen vector matrix $\boldsymbol{U}$ into $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$, $\boldsymbol{U}^{\mathrm{T}} = [\boldsymbol{U}_x^{\mathrm{T}},\ \boldsymbol{U}_y^{\mathrm{T}}]$, where $\boldsymbol{U}_x$ represents the input joint angle space and $\boldsymbol{U}_y$ represents the output joint angle space. To obtain a mapping relationship from the input to output joint angle spaces, first, we minimize $\|\boldsymbol{x} - \boldsymbol{U}_x\boldsymbol{v}\|^2$ with respect to the value of eigen vector $\boldsymbol{v}$, and can then achieve $\boldsymbol{v} = (\boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{U}_x)^{-1}\boldsymbol{U}_x\boldsymbol{x}$, and the output $\boldsymbol{y} = \boldsymbol{U}_y\boldsymbol{v}$. Thus we can obtain the regression coefficients

$$\boldsymbol{\alpha} = \boldsymbol{U}_y\left(\boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{U}_x\right)^{-1}\boldsymbol{U}_x. \tag{6}$$

Because the matrix $\boldsymbol{U}$ is orthogonal, it means $\boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{U}_x + \boldsymbol{U}_y^{\mathrm{T}}\boldsymbol{U}_y = 1$, and the invertible square matrixes $\boldsymbol{E}$ and $\boldsymbol{S}$ have the feature: $(\boldsymbol{E} + \boldsymbol{USU}^{\mathrm{T}})^{-1} = \boldsymbol{E}^{-1} - \boldsymbol{E}^{-1}\boldsymbol{U}(\boldsymbol{S}^{-1} + \boldsymbol{U}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{U})^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{E}^{-1}$, so we can obtain a more easy-to-calculate appraoch for the coefficients $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \boldsymbol{U}_x\left(\boldsymbol{U}_y^{\mathrm{T}} - \boldsymbol{U}_y^{\mathrm{T}}\left(\boldsymbol{U}_y\boldsymbol{U}_y^{\mathrm{T}} - \mathbf{I}\right)^{-1}\boldsymbol{U}_y\boldsymbol{U}_y^{\mathrm{T}}\right). \tag{7}$$

Suppose there exists a Gaussian distributed noise variable β_y, its standard deviation σ can be estimated by $\boldsymbol{y}_k - \boldsymbol{\alpha x}_k,\ k = 1, \ldots, K$. In our experiment, a predicted function for each DOF of the synthesized pose is constructed, therefore, to predict the dth DOF of the pose, we can describe our local regression model as

$$q_{t,d} = \boldsymbol{\alpha}_d^{\mathrm{T}}\tilde{\boldsymbol{Q}} + N(0, \sigma_d), \tag{8}$$

where $q_{t,d}$, σ_d are scalars, $q_{t,d}$ represents the dth DOF of the tth frame pose, and σ_d represents the standard deviation for the dth prediction function. $\boldsymbol{\alpha}_d$, $\tilde{\boldsymbol{Q}}$ are vectors, $\boldsymbol{\alpha}_d$ are the regression coefficients for dth DOF, and $\tilde{\boldsymbol{Q}}$ are the reconstructed poses before current synthesized pose.

6 Online motion synthesis

In this section, we solve the problem of how to synthesize sequential poses from the control information provided by six inertial sensors. During runtime, our performance animation system automatically combines the control signals and online local regression models, and synthesizes a performer's poses frame by frame.

6.1 Control stability

The stability of our control system is very important. However, the control signals $\boldsymbol{c}_t$ provided by inertial sensors might change because of Gaussian noise. Suppose σ is the standard normal distribution, the control term of sensors can be defined as

$$E_{\text{control}} = -\ln \Pr(\boldsymbol{c}_t|\boldsymbol{q}_t) \propto \frac{\|\boldsymbol{f}(\boldsymbol{q}_t; \tilde{\boldsymbol{s}}, \boldsymbol{L}) - \boldsymbol{c}_t\|^2}{2\pi\sigma^2}, \tag{9}$$

where $\boldsymbol{q}_t$, $\tilde{\boldsymbol{s}}$, $\boldsymbol{L}$, $\boldsymbol{c}_t$ are vectors, $\boldsymbol{q}_t$ is the synthesized pose, $\tilde{\boldsymbol{s}}$ is the avatar's skeletal size, $\boldsymbol{L}$ are the inertial sensors' local coordinates, and $\boldsymbol{c}_t$ are the observation data provided by sensors. The FK function $\boldsymbol{f}$ calculates the global coordinate values for the current pose.

There exist outliers in control signals provided by inertial sensors, especially for the positional data. Because of the ultrasonic sensors and the occlusion problems, the positional data may be destroyed by outliers, missing data and error accumulation. Focusing on these problems, we adopt the Lorentzian robust estimator to filter the noise data. Thus the matching cost term can be defined as follows:

$$\rho(e) = \log\left(1 + \frac{e^2}{2\sigma^2}\right), \tag{10}$$

where e is the distance value between the predicted signals and the observation signals, and the parameter σ is for the robust estimator.

6.2 Motion priors

We use the prior term to constrain the synthesized motion to meet the probabilistic distribution up to similar motion data in the local region. In our animation system, the prior term is defined as follows:

$$\Pr(\boldsymbol{q}_t|\tilde{\boldsymbol{Q}}) \propto \prod_{d=1}^{D} \exp\left(-\frac{(q_{t,d}-\boldsymbol{\alpha}_d^{\mathrm{T}}\tilde{\boldsymbol{Q}})^2}{2\pi\sigma_d^2}\right). \tag{11}$$

where $q_{t,d}, d=1,\ldots,D$ is the dth DOF of the current pose $\boldsymbol{q}_t$. $\boldsymbol{\alpha}_d$ and σ_d are the regression coefficients and standard deviation of the dth prediction model. The vector $\tilde{\boldsymbol{Q}}$ sequentially records the m previous synthesized poses $[\tilde{\boldsymbol{q}}_{t-1},\ldots,\tilde{\boldsymbol{q}}_{t-m}]$.

We can obtain the following energy formulation by minimizing the negative log of $\Pr(\boldsymbol{q}_t|\tilde{\boldsymbol{Q}})$:

$$E_{\text{prior}} = \sum_d \frac{(q_{t,d}-\boldsymbol{\alpha}_d^{\mathrm{T}}\tilde{\boldsymbol{Q}})^2}{2\pi\sigma_d^2}. \tag{12}$$

6.3 Implementation details

We adopted gradient-based optimization using the Levenberg-Marquardt method[2] for the objective function which is defined in (2), and used the most similar motion example which already exists in the database to initialize the optimization. Because of a good initialization, the optimization converged quickly. The computational efficiency of our animation system mainly relies on the searching scope in the motion databases, so we accelerated the process for the K nearest neighbor search with a similar strategy in [19]. Our system was able to reach an average frame rate of 37 fps real-time synthesis.

7 Results and evaluation

The database we captured includes five full-body behavior movements: golf swing (2537 frames), basketball (6582), boxing (29852), walking (20866) and running (5772). All of them were recorded using a Vicon mocap system[3] which has a frame rate of 120 fps. To match the inertial sensor's frame rate, the original mocap data were downsampled to 40 fps.

We verified the effectiveness of our proposed approach on various movements based on a large motion database and evaluated the reconstructed results with ground-truth data.

- **Testing on performance control and evaluation.** We used all the control signals provided by six inertial sensors to control a full-body avatar in real-time (please watch the video). The video also shows the performance comparison between our method and the IK method. Because the inertial sensor can provide two types of data (orientation and position), we used the position data for center of torso, head and both ankles, and used orientation data for both hands. The results show that without prior data, we cannot achieve real-time control based only on the IK technique. In addition, we used the leave-one-out error evaluation method to evaluate the quality of the synthesized actions. Figure 5 shows the average synthesis errors. The errors were calculated by degrees per joint angle per frame, using the average distance between the motion captured by the Vicon mocap system and the synthesized motion. We considered three types of control information: 1) 3D position and 3D orientation; 2) only 3D orientation; 3) only 3D position. We found that if we used both position and orientation constraints, the reconstruction errors were the lowest in the above three combinations of information. In addition, compared with 3D orientation information, 3D position information was more useful and had better compliance with the constraints in motion reconstruction.
- **Calibration for skeleton and local coordinates.** Our system needs to be robust for different users and various sensor wearing styles, so we used an average skeleton size which was calculated using different skeleton files from the CMU mocap database. It was used as the standard subject for skeleton

[2] Lourakis M I A. Levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms In C/C++. 2009.

[3] http://www.vicon.com.

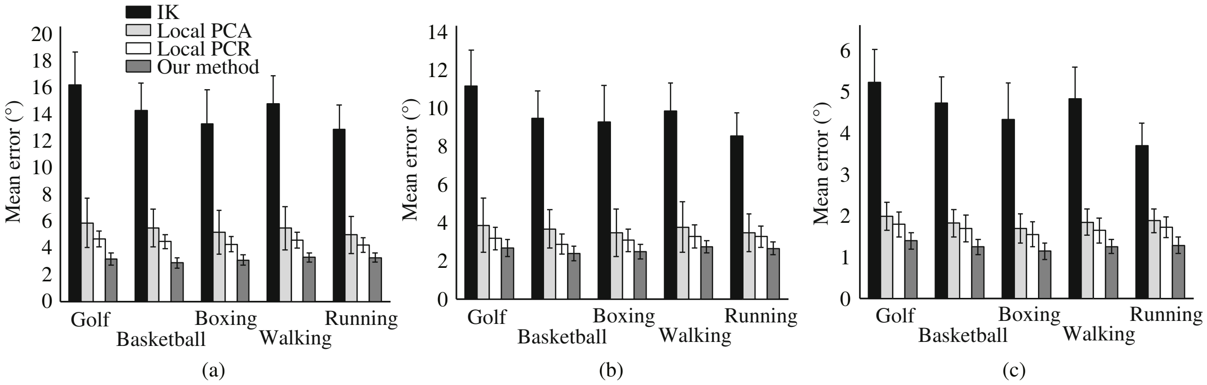

Figure 5 Comparisons with inverse kinematics, local principle component analysis and local principle component regression algorithms. (a) Motion synthesis using only orientation signals provided by all the inertial sensors; (b) motion synthesis using only position signals provided by all the inertial sensors; (c) motion synthesis using both orientation and position signals provided by all the inertial sensors. The bars from left to right are mean error from IK, local PCA, local PCR and our method.

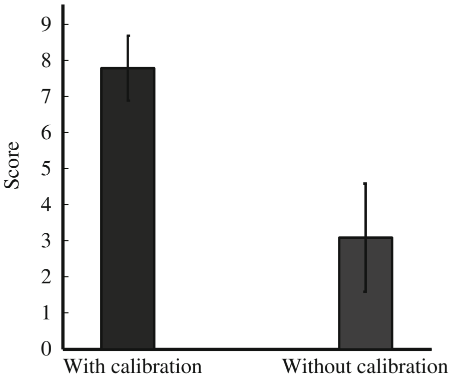

Figure 6 Study of the comparison between calibration and no calibration by the user. Score 9 means most realistic, and score 0 means least realistic.

Table 1 Skeleton size comparison for calibration

Skeleton size	Femur	Tibia	Back	Neck	Head	Clavicle	Humerus	Radius	Wrist
Standard Subject	7.23	7.54	7.89	4.21	1.93	3.89	6.57	4.02	1.85
Calibration data 1	6.59	7.38	7.39	3.97	2.48	3.73	5.56	3.06	1.51
Ground truth 1	6.53	7.41	7.42	3.94	2.59	3.75	5.51	3.09	1.54
Calibration data 2	6.58	6.89	7.03	3.45	1.76	3.49	5.04	2.85	1.37
Ground truth 2	6.57	6.81	6.99	3.39	1.72	3.47	4.99	2.71	1.35

calibration. We tested different users, and Table 1 shows the calibration results for several skeletons of two different users. We found that, after our calibration step, the user's skeleton size was close to his/her ground truth data captured by the Vicon mocap system. After the calibration process, we also obtained local coordinates for each sensor. We asked sixteen users to provide a score (1–9) for the online synthesized motions, without telling them whether we calibrated or not. The users were chosen from undergraduate students with little experience of 3D animation. We tested different human movements. Figure 6 shows the study of the comparisons of motion quality with and without calibration by the user. We found that users usually chose the motion after calibration as the better one, and give it a high score. The results from the user study told us that our calibration step is important and useful for the quality of our online motion control.

- **Comparisons with previous algorithms.** To test its performance, we compared the IK techniques,

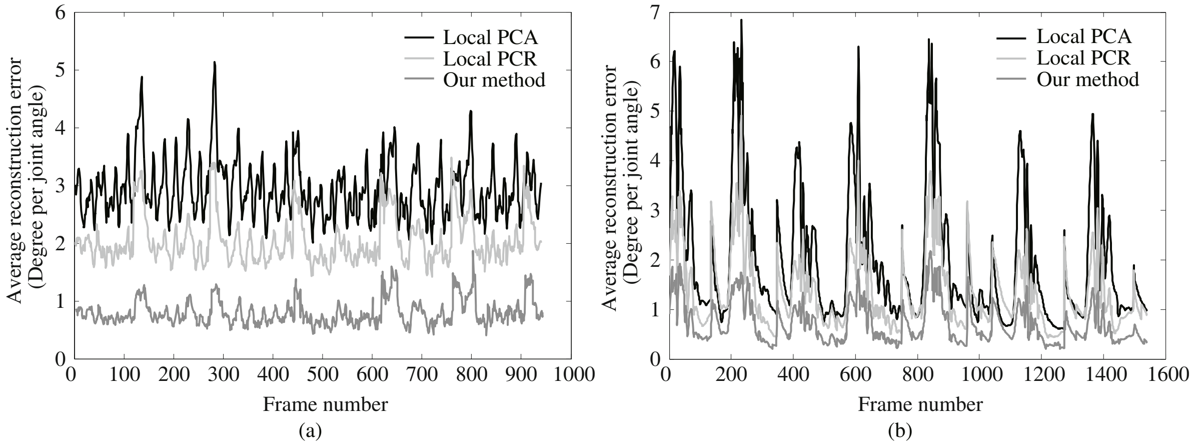

Figure 7 Frame-by-frame comparison for one testing sequence. (a) Walking motion; (b) boxing motion. The lines from top to bottom are reconstruction errors from local PCA, local PCR and our method.

Table 2 Comparison of the average reconstruction errors for different methods and different databases

	65609 poses	1.1 M poses
IK	4.76	4.76
LPCA	1.97	1.42
LPCR	1.75	1.19
Our method	1.43	0.87

local PCA models in [19] and local PCR models in [7] with the proposed model in our study. Figure 5 shows the standard deviations and mean errors of the reconstruction errors for various movements (golf swing, basketball, boxing, walking and running). Figure 7 shows the frame-by-frame comparison of reconstruction errors for single test data between the local PCA model, local PCR model and our proposed model. The assessment results indicated the synthesis results using our proposed method were better than the results created by the other two methods.

- **Different information from sensors.** The video of our study analyzed four combinations of input signals from the inertial sensors. The results told us that the more constraints used, the smaller the reconstruction errors. It is not surprising that when the total information from all sensors was used, the reconstruction errors were the smallest.
- **Testing on different databases.** Table 2 gives the average reconstruction errors of five different actions from four algorithms for two different training databases. One database has 65,609 poses based on five captured motion sequences, and the other has 1.1 M poses downloaded from the CMU database. The reconstruction errors were calculated using both 3D orientation and position constraints from six inertial sensors. We found that when the size of the training database was increased, the reconstruction error reduced. By testing on different databases, we also verified the benefits of the proposed model.

8 Conclusion

In this study, a new local model was introduced for real-time control of a virtual person using only six initial sensors. The proposed method, which was based on a data-driven approach, was to use several nearest motion examples to construct sequential online local regression models for online motion synthesis. There was one limitation for the proposed method, which was that the motion data needed to be previously prepared for online search. However, the proposed model demonstrated better performance over previous models, and our performance interface which used only six sensors was much cheaper and less intrusive for full-body avatar control.

Acknowledgements

This work was supported by National Science and Technology Support Program (Grant No. 2012BAH35B02), National Natural Science Foundation of China (Grant No. 61070078), and Fundamental Research Funds for Central University. Particular thanks to the kind guidance of Prof. Jinxiang Chai in Texas A&M University.

References

1 Badler N I, Hollick M, Granieri J. Realtime control of a virtual human using minimal sensors. Presence, 1993, 2: 82–86

2 Semwal S, Hightower R, Stansfield S. Mapping algorithms for real-time control of an avatar using eight sensors. Presence, 1998, 7: 1–21

3 Yin K, Pai D K. FootSee: an interactive animation system. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, 2003. 329–338

4 Slyper R, Hodgins J. Action capture with accelerometers. In: Proceedings of 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Dublin, 2008. 193–199

5 Ha S, Bai Y, Liu C. Human motion reconstruction from force sensors. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Vancouver, 2011. 129–138

6 Tautges J, Zinke A, Kruger B, et al. Motion reconstruction using sparse accelerometer data. ACM Trans Graph, 2011, 30: 18

7 Liu H J, Wei X L, Chai J X, et al. Realtime human motion control with a small number of inertial sensors. In: Proceedings of the 2011 Symposium on Interactive 3D Graphics and Games. New York: ACM, 2011. 133–140

8 Shotton J, Fitzgibbon A, Cook M, et al. Real-time human pose recognition in parts from single depth images. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2011. 1297–1304

9 Wei X L, Zhang P Z, Chai J X. Accurate realtime full-body motion capture using a single depth camera. ACM Trans Graph, 2012, 31: 188

10 Kovar L, Gleicher M. Automated extraction and parameterization of motions in large data sets. ACM Trans Graph, 2004, 23: 559–568

11 Kwon T, Shin S Y. Motion modeling for online locomotion synthesis. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, 2005. 29–38

12 Mukai T, Kuriyama S. Geostatistical motion interpolation. ACM Trans Graph, 2005, 24: 1062–1070

13 Heck R, Gleicher M. Parametric motion graphs. In: Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games. New York: ACM, 2007. 129–136

14 Kovar L, Gleicher M, Pighin F. Motion graphs. ACM Trans Graph, 2002, 21: 473–482

15 Lee Y, Wampler K, Bernstein G, et al. Motion fields for interactive character locomotion. ACM Trans Graph, 2010, 29: 1–8

16 Levine S, Wang J, Haraux A, et al. Continuous character control with low-dimensional embeddings. ACM Trans Graph, 2012, 31: 28

17 Min J Y, Chai J X. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. ACM Trans Graph, 2012, 31: 153

18 Safonova A, Hodgins J K. Construction and optimal search of interpolated motion graphs. ACM Trans Graph, 2007, 26: 108

19 Chai J X, Hodgins J. Performance animation from low-dimensional control signals. ACM Trans Graph, 2005, 24: 686–696

20 Grochow K, Martin S L, Hertzmann A, et al. Style-based inverse kinematics. ACM Trans Graph, 2004, 23: 522–531

21 Lau M, Chai J X, Xu Y Q, et al. Face poser: interactive modeling of 3D facial expressions using facial priors. ACM Trans Graph, 2009, 29: 3

22 Liu H J, He F Z, Cai X T, et al. Performance-based control interfaces using mixture of factor analyzers. Visual Comput, 2011, 27: 595–603

23 Min J Y, Chen Y L, Chai J X. Interactive generation of human animation with deformable motion models. ACM Trans Graph, 2009, 29: 9

24 Wei X L, Min J Y, Chai J X. Physically valid statistical models for human motion generation. ACM Trans Graph, 2011, 30: 19

25 Ye Y, Liu C. Synthesis of responsive motion using a dynamic model. Comput Graph Forum, 2010, 29: 555–562