Reconstruction of Human Motions Using Few Sensors

Jochen Tautges **Björn Krüger** **Arno Zinke** **Andreas Weber**
tautges kruegerb zinke weber
@cs.uni-bonn.de
Römerstraße 164
University of Bonn
53117 Bonn
phone: +49 (228) 73-4945

Abstract: In this paper we show that it is possible to reconstruct whole body motions from the data taken by as few as 1, 2, 4 inertial sensors, if a semantic pre-classification of the motion is available, and motion data bases can be used to synthesize missing degrees of freedom. We demonstrate some examples of reconstructed motions and compare them with a video of the take, and we use data of optical motion captures for numerical comparisons.

Although our work is mainly inspired by the wish to create easy-to-use and cheap motion capture systems for everyday settings, such a motion capture system is also an easy-to-use tracking system of whole body motions. Moreover, the underlying synthesis of motion out of low-dimensional control signals opens new ways for the control of avatars.

Keywords: Motion Capture and Tracking, Avatars, 3D Interaction

1 Introduction

Traditionally there is a distinction between tracking of certain parts of the human body—e.g. head, overall body position, one or two hands or the feet—and motion capturing, which is commonly understood as capturing whole body motions [Bis84, HW02, Val02, WF02, Fox05, MJKM04, HBS07, ZH07, VAV$^+$07, RNd$^+$07]. However, if rather sparse sets of markers resp. sensors can be used to capture resp. reconstruct human motion then there is a smooth transition between these two domains. Although the usability of a system for users greatly improves if only few markers or sensors have to be used to reconstruct their whole body motions the question of capturing, tracking resp. reconstructing human motion from few sensors has been addressed quite rarely. Chai and Hodgins [CH05] use the motions of a motion data base to generate realistically looking motions on the basis of video data of two cameras. They use only few markers (about 6–9) and track their 3D-positions by computer vision techniques, as well as the orientation of the body root. Searching for similar motions to the current frame they use principal component analysis to come up with a linear low-dimensional approximation of a motion, in which they synthesize the "most probable" next frame of the motion using information from the sensors, and motion priors generated from the motion data base.

As has already been noted by Chai and Hodgins the method is not only suitable for sparse sets of optical marker, but also for other low dimensional control signals, e.g. ones from acceleration sensors. However we do not know from realizations of such extensions. Liu et al. [LZWM06] use local linear models directly on sets of markers. For transitions between the local linear models they use random forest. In [LM06] they generalize the method to cases, in which several markers are not visible during a motion capture session.

Our contribution. In this paper we show that it is possible to reconstruct whole body motions from the data taken by as few as 1, 2, 4 inertial sensors for several classes of motions, if semantic pre-classifications of motions are available and motion data bases can be used to synthesize missing degrees of freedom. In contrast to previous approaches, which rely on linear models of the motions, we use multi-linear models and use an optimization approach to synthesize the full motions. We also extend the technique suggested in [CH05] to acceleration-based sensor signals, and compare the results of our approach to the outcome of this extension.

When comparing our reconstructed motions with ground-truth motions we realized that the established approaches to compute a distance between motions on the average error of local joint orientations can fail: Being purely pose-based the distance measure might fail to detect artifacts like directional flips or jitter (i.e. the distance between the original motion and the reconstructed motion is small, although the latter exhibits these artifacts). So we present a novel practical measure to compare the similarities of motions based on quantities represented in a global coordinate frame. Assuming a fixed skeleton topology our goal is a universal measure that both matches the human perception and is simple enough to be implemented in time critical environments.

2 Preliminaries

In this section we recall the use of multi-linear models for motion data representations [Vas02, RCO05, MK06, KTW07, KTMW08] that will be central to our approach.

2.1 Mathematical background

A *tensor* Θ of *order* $N \in \mathbb{N}$ and type $(d_1, d_2, \ldots, d_N) \in \mathbb{N}^N$ over the real number $\mathbb{R}$ is defined to be an element in $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$.

Intuitively, the tensor Θ represents $d = \prod_{i=1}^{N} d_i$ real numbers in a multidimensional array based on N indices. These indices are also referred to as the *modes* of the tensor Θ. A tensor Θ can be transformed by an N-mode singular value decomposition (N-mode SVD). The result of the decomposition is a *core tensor* Φ of the same size as Θ and associated orthonormal matrices $U_1, U_2, \ldots, U_N$. The matrices U_k are elements in $\mathbb{R}^{d_k \times d_k}$ where $k \in \{1, 2, \ldots, N\}$.

Mathematically this decomposition can be expressed in the following way:

$$\Theta = \Phi \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_N U_N. \tag{1}$$

This product is defined recursively, where the mode-k-multiplication $\times_k$ with U_k replaces each mode-k-vector v of $\Phi \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_{k-1} U_{k-1}$ for $k > 1$ (and Φ for $k = 1$) by the vector $U_k v$. A more detailed description of multi-linear algebra is given in [VBPP05].

2.2 Motion tensors

In our case the tensors are filled with motion data similar to [KTW07, KTMW08]. A frame is defined by the position of its root node p and quaternions $(q_1 \dots q_{31})$ describing the rotation of the skeleton segments. A motion is defined to be a sequence of frames. We build two motion tensors, one for the root positions and one for the rotational data, since the variance of these types of data is quite different: the values of unit quaternions are in the interval $[-1, 1]$ while the translational offset of the root position is not limited at all. The rotational data are stored in so called technical modes, that correspond to the structure of the underlying motion capture data, and so called natural modes, which correspond to properties of motions that typically appear in the context of a motion capture session. We call the technical modes *DOF mode*, *Joint mode* and *Frame mode* and the natural modes we used split up to *Style mode*, *Actors mode* and *Repetition mode*. A typical tensor with rotation data Θ_q has a dimension of $N = 6$ and size of $d = 4 \times 31 \times 70 \times 3 \times 5 \times 3$. Since only one node is considered in a tensor Θ_p storing the translation of the root node, this tensor does not need a Joint mode and its dimension reduces to $N = 5$. Identifying the root node's degrees of freedom with the axes in 3D space the size of the DOF mode in this case becomes 3.

2.3 Motion synthesis

As described in section 2.1 a data tensor Θ can be decomposed into a core tensor Φ and related matrices $U_1, \dots, U_N$. In this composition each matrix U_k corresponds to a specific mode and each row in a matrix U_k corresponds to a specific entry of this mode.

Instead of reconstructing the complete data tensor Θ (by mode-multiplying Φ with all matrices U_k) this representation also allows to directly reconstruct a single motion contained in the data tensor. This is done by first multiplying Φ with each matrix corresponding to a technical mode, and then multiplying the result with just one row of each matrix corresponding to a natural mode. Let t be the number of technical modes, n the number of natural modes and let u_k^i be the i-th row of matrix U_k. Reconstruction of a motion m then can be expressed in the following way:

$$m = \Phi \times_1 U_1 \dots \times_t U_t \times_{t+1} u_{t+1}^{i_1} \dots \times_{t+n} u_{t+n}^{i_n}. \tag{2}$$

While using a single row of each matrix $U_{t+1}, \dots, U_{t+n}$ always results in one of the original motions it is also possible to synthesize a new motion m_{new} by using linear combinations of matrix rows. This can be expressed mathematically in this way:

$$m_{\text{new}} = \Phi \times_1 U_1 ... \times_t U_t \times_{t+1} \lambda_{t+1} U_{t+1} ... \times_{t+n} \lambda_{t+n} U_{t+n}, \tag{3}$$

with

$$\lambda_k U_k = \left(\lambda_k^1 \dots \lambda_k^{d_k}\right) \begin{pmatrix} u_k^1 \\ \vdots \\ u_k^{d_k} \end{pmatrix} = \sum_{i=1}^{d_k} \lambda_k^i \, u_k^i =: x_k. \tag{4}$$

With this model in hand we are able to formulate an optimization problem based on the variables λ to synthesize a desired motion sequence. Since the size of the vectors λ depends on the size of the natural modes, the dimension of the optimization problem is 11 in our case.

3 Distance Measures for Comparing Motions

When comparing motions finding a distance measure matching the human perception of motion is a nontrivial task. A well established approach is to compute a distance based on the average error of local joint orientations [CH05]. However, such methods may be inappropriate if the global similarities of poses have to be computed since the hierarchical organization of a skeleton is completely neglected: An error at a parent joint affects also its children. Hence, a local error at a joint at the top of the skeleton hierarchy is likely to have a bigger impact on the global error than the same error at a lower level joint. As a consequence the resulting globally visible error may be not properly reflected by a distance measure based on local joint orientations. Moreover, using the L_2 norm on Euler Angles directly suffers from the problem of finding an adequate distance measure for this representation of rotations.

3.1 A novel distance measure

In this section we present a novel practical measure to compare the similarities of motions based on quantities represented in a global coordinate frame. Assuming a fixed skeleton topology our goal is a universal measure that both matches the human perception and is simple enough to be implemented in time critical environments.

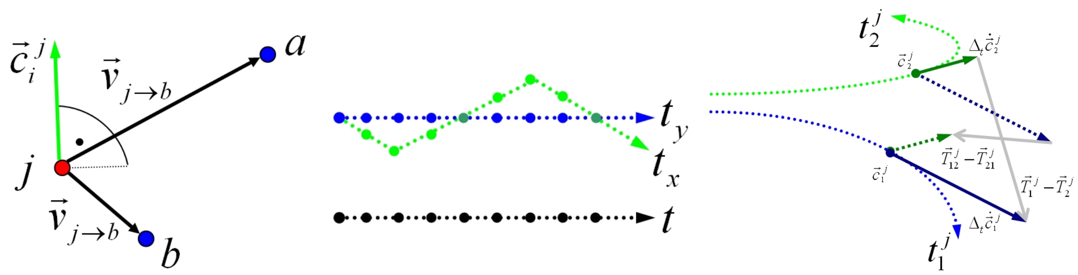


Figure 1: **Left:** Notation. **Middle:** Comparing two trajectories t_x and t_y to a reference t. Frames are indicated by dots. Note that t_y is just a shifted copy of t. Although the spatial distance is the same for t_x and t_y, t_x clearly differs from t which can be detected by comparing the local Taylor expansions of t_x and t. In this example a purely pose-based approach with frame-wise comparison fails. **Right:** Illustrating the meaning of T_1^j,T_2^j,T_{12}^j and T_{21}^j. In this example $D_{1,2}^j = \|T_1^j - T_2^j\|$.

The basic idea is to frame-wise compare the cross product $\vec{c}_i^{\,j}$ formed by a joint j and two of its child joints a and b (figure 1 left).

$$\vec{c}_i^{\,j}(a, b, f) = \vec{v}_{j\to a}(f) \times \vec{v}_{j\to b}(f) \tag{5}$$

Here, f denotes the frame of a motion i for which the cross product at a joint j is computed, $\vec{v}_{j\to a}$ the vector pointing from j to a and $\vec{v}_{j\to b}$ the vector pointing to b, respectively. Please note that $\vec{c}_i^{\,j}$ can be interpreted geometrically as the normal of the triangle spanned by $\vec{v}_{j\to a}$ and $\vec{v}_{j\to b}$ weighted by two times the area of this triangle. Hence, $\vec{c}_i^{\,j}$ characterizes the orientation and the relative angle of two connected bones. In the following the frame-based trajectory of $\vec{c}_i^{\,j}$ is denoted t_i^j.

Supposing that two different motions of a joint j (and its child joints) are given we use a local Taylor expansion of $\vec{c}_i^{\,j}$ to frame-wise describe the similarity between these two motions. For the two corresponding first-order Taylor expansions $\vec{T}_1^j$ and $\vec{T}_2^j$ around the frame f yields:

$$\vec{T}_1^j(f) = \vec{c}\,_1^j(a,b,f) + \Delta_t\,\dot{\vec{c}}\,_1^j(a,b,f) \tag{6}$$

and

$$\vec{T}_2^j(f) = \vec{c}\,_2^j(a,b,f) + \Delta_t\,\dot{\vec{c}}\,_2^j(a,b,f), \tag{7}$$

where Δ_t is a time step and $\dot{\vec{c}}\,_i^j$ is the time derivative of $\vec{c}\,_i^j$. Let moreover $\vec{T}_{12}$ and $\vec{T}_{21}$ be two functions of mixed terms of $\vec{T}_1$ and $\vec{T}_2$:

$$\vec{T}_{12}^j(f) = \vec{c}\,_1^j(a,b,f) + \Delta_t\,\dot{\vec{c}}\,_2^j(a,b,f), \tag{8}$$

$$\vec{T}_{21}^j(f) = \vec{c}\,_2^j(a,b,f) + \Delta_t\,\dot{\vec{c}}\,_1^j(a,b,f). \tag{9}$$

If the two trajectories are traversed in a similar manner $\vec{T}_1^j$, $\vec{T}_2^j$, $\vec{T}_{12}^j$ and $\vec{T}_{21}^j$ have to match. Consequently differences indicate local errors (see also figure 1 middle and right). Based on this observation our local distance measure $D_{1,2}^j$ with respect to a frame f computes as:

$$D_{1,2}^j(a,b,f) = \max(\|\vec{T}_1^j - \vec{T}_2^j\|, \|\vec{T}_{12}^j - \vec{T}_{21}^j\|), \tag{10}$$

which can be simplified to:

$$D_{1,2}^j(a,b,f) = C_{1,2}^j(a,b,f) + \dot{C}_{1,2}^j(a,b,f) \tag{11}$$

with

$$C_{1,2}^j(a,b,f) = \|\vec{c}\,_1^j - \vec{c}\,_2^j\| \tag{12}$$

and

$$\dot{C}_{1,2}^j(a,b,f) = \Delta_t\,\|\dot{\vec{c}}\,_1^j - \dot{\vec{c}}\,_2^j\|. \tag{13}$$

Setting the remaining free parameter Δ_t to

$$\frac{\|\vec{v}_{j\to a}\|\|\vec{v}_{j\to b}\|}{\|\dot{\vec{c}}\,_1^j\| + \|\dot{\vec{c}}\,_2^j\|} \tag{14}$$

scales $\dot{C}_{1,2}^j$ to the range of $C_{1,2}^j$. Now that a similarity measure for a single joint j and two children a and b can be computed we finally generalize this measure to a distance measure $\mathrm{D_{pv}}$ for an arbitrary set of joints by summing over all frames f, all joints j and child joints a,b according to

$$\mathrm{D_{pv}} = \sqrt{\sum_{f=1}^{d_2}\sum_{j=1}^{d_3} D^j(f)}, \tag{15}$$

with

$$D^j(f) = \sum_{a=1}^{s_j}\sum_{b=1}^{s_j}(1-\delta_{ab})\left(D_{1,2}^j(a,b,f)\right)^2. \tag{16}$$

Please note that the error at a joint is implicitly weighted by the length of its bones. This is a desirable property, since longer bones are very likely to dominate the perception of a motion. Moreover, subtle errors like flipped joints are detected by the proposed method. However, although $\mathrm{D_{pv}}$ is invariant under translation, rotating motions yields different results. This is a direct consequence of performing all computations with respect to a global coordinate frame.

4 Motion Data Base

For our approach we need a data base of motions that is semantically pre-classified. Using the category names such a semantic pre-classification is available in the commonly used CMU database [Car04]. However, the collection of motions contained in the CMU database is not sufficient for building a multi-linear model, since most motions are performed by one actor only without any stylistic variation.

For our purposes we found the data provided by HDM05 motion capture data base [MRC+07] more suitable. The HDM05 database consists of about 50 minutes of motion data, which are arranged into 64 different classes and styles. Each such motion class contains 10 to 50 different realizations of the same type of motion, covering a broad spectrum of semantically meaningful variations. The resulting *motion class database* contains $1\,457$ motion clips.

5 Reconstructing Motions from Few Sensors

5.1 A novel approach based on multi-linear representations of motions

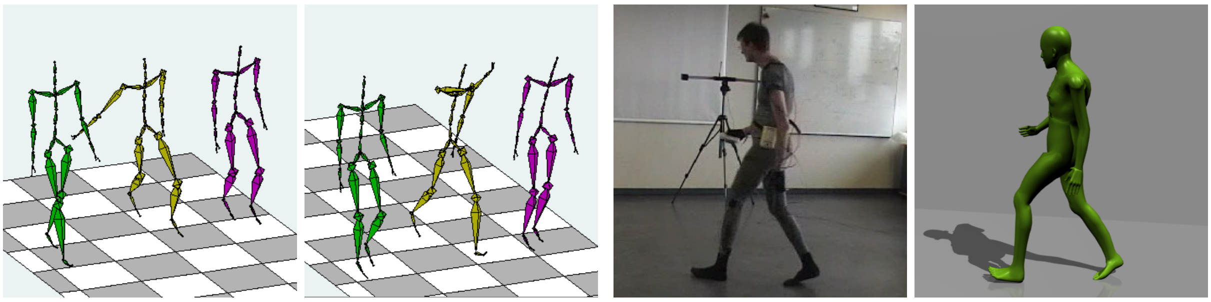
The multi-linear models give rise to morphable models of motions that can be used to synthesize motions in an easy way by using linear combinations, cf. equation 3.

For reconstructing motions given by only a few sensors we can formulate an optimization problem trying to find the synthetic motion that best fits the sensor data. In our case we create synthetic motions, calculate the accelerations and calculate the distance to the data we got from the acceleration sensors. Hence our synthetic motions depend on the vectors λ which give the weights to each row of the matrices U we have to find a configuration for x that best fits the sensor data.

In contrast to [CH05], in which only *a pose prior* is computed from the data base and the smoothness of transitions is ensured by a rather *ad hoc and non-physical smoothness term* involving the two prior poses only, we perform the optimization on a window of frames. In order to make use of the smoothness conditions contained in the motion data, we calculate distances between complete motions within the window instead of calculating distances between single frames. This is done by identifying acceleration vectors with points in 3D space and considering all points defined by a motion as one big point cloud. Mathematically, the distance between two motions m and m' (within the window) is given as

$$\mathrm{dist}(m, m') = \min_{\theta} \left(\sum_{f=1}^{d_2} \sum_{j=1}^{d_3} \|a_{f,j} - T_\theta a'_{f,j}\| \right) \tag{17}$$

where T is a linear transformation that rotates a point a about the (vertical) y-axis by θ degrees. This minimization problem has a closed-form solution that can be found in [KGP02]. The size of the Frame mode is denoted by d_2 and the one of the Joint mode by d_3. Reconstructing a motion m now means finding coefficients λ so that $\mathrm{dist}(m, m_{\mathrm{new}})$ is minimized with m_{new} defined as in equation 3. By applying equation 4 and optimizing in x we can speed up the optimization procedure by saving tensor multiplications and reducing the number of variables in case of truncated core tensors, cf. [KTW07, KTMW08]. Note that a solution in x can be easily transformed back to the more descriptive solution in λ that gives the weights for each component of a natural mode.



(a) (b)

Figure 2: a) Pictures of reconstruction based on the method of Chai and Hodgins using acceleration data as control signals directly. Original motion (green), reconstruction based on original control term (yellow) and reconstruction based on modified control term (purple).
b) Pictures of a walking motion. The left picture is from the video we took from the take, the right picture is a rendered picture of our reconstructed motion from five inertial sensors.

5.2 Extending the approach of Chai and Hodgins

We extended the technique described in [CH05] from marker positions to accelerations as control signal. Chai and Hodgins suggest to synthesize the reconstructed motion solving a minimization problem involving the following three terms:

1. A pose prior using the distribution of poses from a motion data-base.
2. A smoothness term ensuring smooth transitions between poses.
3. A control term measuring the distance of the control signal to the one computed from the reconstructed motion.

In the original version (according to [CH05]) the control term measures the Euclidean distance between poses. Thus, conceptually this control term involves distances of positions

$$D_c^{\mathrm{p}} = \|f(q) - c\|^2. \tag{18}$$

In principle also accelerations could be used as a control signal meaning that distances between accelerations have to be computed as e.g.

$$D_c^{\mathrm{a1}} = \|\ddot{f}(q) - \ddot{c}\|^2 \tag{19}$$

or

$$D_c^{\mathrm{a2}} = \frac{\|\ddot{f}(q) - \ddot{c}\|}{\|\ddot{f}(q)\| + \|\ddot{c}\|} \tag{20}$$

Unfortunately, none of these distance measures yields satisfying results, cf. Fig. 2a.

However, when using the position information from the previously reconstructed pose one can estimate a control position by doubly integrating the acceleration data for one time step. Since the position estimate bases on a very short time span between two frames the problem of velocity and position drifts that is otherwise an important issue when integrating acceleration data [WF02] can be neglected.

6 Results

We used the techniques described in the previous sections to reconstruct motions from just a few input signals in two different ways.

1. We used the sensor data from some XSens inertial sensors attached to the hands and feet of some subjects to reconstruct the performed motions. These motions are compared with a video that was taken during the capturing.
2. In order to numerically compare the outcome of the reconstructed motions, we use motions of the CMU motion database [Car04] and compute the accelerations of some body points. Those are then used to reconstruct the motions with our approach.

   Moreover, we compare the results of our approach with the techniques described in [CH05], cf. Sect. 5.2.

6.1 Reconstructions from few sensors

All motions that should be reconstructed out of our model were captured by an Xbus Master system from the company XSens `http://www.xsens.com`. The system consists of five sensors (called MTx) that provide measurement of drift-free 3D orientation as well as kinematic data: 3D acceleration, 3D rate of turn (rate gyro) and 3D earth-magnetic field. We refer to the supplemented video for results of this reconstruction. In Fig. 2b some screenshots are given as a reference.

6.2 Numerical comparisons

As described in section 4 we build our multi-linear model from the HDM05 motion database. To verify our results, we evaluate our reconstructions with motion data taken from the CMU database or from left-out motions from the HDM05 data-base, if no suitable motions are contained in the CMU database.

Moreover, we compare these results of the reconstructions of our novel approach with the results obtained by the technique of Chai and Hodgins [CH05]. As the acceleration data have to be derived from the position data in these examples taken from motion capture data, we use the position data directly in the control term instead of locally estimating them from acceleration data and position data of the previously synthesized pose, cf. Sect. 5.2. As this techniques uses PCA on a set of "similar poses" we will use the term "PCA based approach" in our tables.

In table 1 results of this comparison are shown. We used the same distance measure the authors suggest in [CH05], where the L_2-distance is calculated over the joint angles of the motions (which we will denote by $\mathrm{D_E}$). We refer to the supplemented video for renderings of the reconstructions.

Whereas for the walking motion both approaches give visually appealing results, if the data of accelerations of the two hands and feet are used, the PCA based approach gives strong artifacts if only two or one sensor is used (see also the screen shots given in Fig. 3 and Fig. 5).

Notice also the performance of the distance measures when comparing the numerical results with the ones of the renderings of the reconstructions. Whereas the novel distance measure $\mathrm{D_{pv}}$ introduced in Sect. 3.1 clearly identifies problematic cases, the joint angle based measure $\mathrm{D_E}$) fails

Table 1: Average reconstruction errors for sample motions from MoCap data for the PCA-based approach and our *Multi-linear Motion Model* (MMM). We give the average reconstruction errors using the novel distance measure defined in section 3.1 summing over all joints (denoted by D_{pv}), and the commonly used L_2-distance calculated over the joint angles (denoted by D_{E}).

	Walking				**Cartwheel**			
	Distance MMM		Distance PCA		Distance MMM		Distance PCA	
Regarded joints	D_{pv}	D_{E}	D_{pv}	D_{E}	D_{pv}	D_{E}	D_{pv}	D_{E}
footL	15.23	12.16	64.38	11.27	21.22	15.32	102.41	30.49
footR	17.63	12.02	43.64	13.39	40.34	15.83	98.20	39.11
handL	14.83	11.44	24.28	11.34	25.35	15.57	93.73	39.89
handR	14.75	10.13	24.59	12.73	50.42	17.63	91.06	37.26
footL, footR	15.32	8.18	17.37	9.93	26.50	15.11	79.36	33.91
footL, handL	23.41	5.55	24.06	10.29	24.81	15.37	88.38	36.23
footL, handR	14.47	10.18	45.85	16.17	41.49	16.19	90.22	31.45
footL, footR, handL	17.26	14.55	16.76	12.22	25.22	15.39	89.80	33.74
footL, handL, handR	14.50	10.64	22.89	10.40	23.45	16.59	88.13	30.62
footL, footR, handL, handR	14.82	10.15	21.75	13.95	29.69	15.29	93.45	36.00
footL, footR, handL, handR, shoulderR	14.98	10.45	25.68	16.13	29.62	15.29	91.35	31.98
footL, footR, handL, handR, kneeL, kneeR	14.54	10.67	23.70	16.91	25.27	15.26	88.32	33.75

to detect artifacts like directional flips or jitter. Moreover, perceptually similar motions get small distances within the novel distance measure D_{pv}.

7 Conclusion and Future Work

The results of this paper are a proof of concept that reconstructions of full body motions are possible from the control signals of very few acceleration sensors and using pre-classified motions. Whereas for some of our test motions a PCA based approach and our method already work quite well if 4 sensors are used on both hands and feet, for other motions our approach gives significant advantages. If only one or two sensors are used, our novel approach gives significant advantages for all of the tested sample motions.

In several applications there will be a priori knowledge about the kind of expected motions. For reconstructions of general motions that are a sequel of different kinds of motion we plan to combine our approach with classification techniques for motions such as *motion templates* [MR06]. We presume that by using indexing techniques for features we can come up with algorithms which require a preprocessing time being linear in the size of the data base thus overcoming the bottle-neck of quadratic preprocessing time (in the size of the underlying motion data base) to construct a neighbor graph of motions as in [CH05]. Also in future work we plan to compare the outcome of the linear techniques for dimensionality reduction as is done in [CH05] but also in our novel approach by non-linear techniques: we will consider the non-linear transformations obtained by inverse dynamics calculations but also *Gaussian latent variable methods* (GPVLM) that have been already used for “style-based inverse kinematics” [**?**].

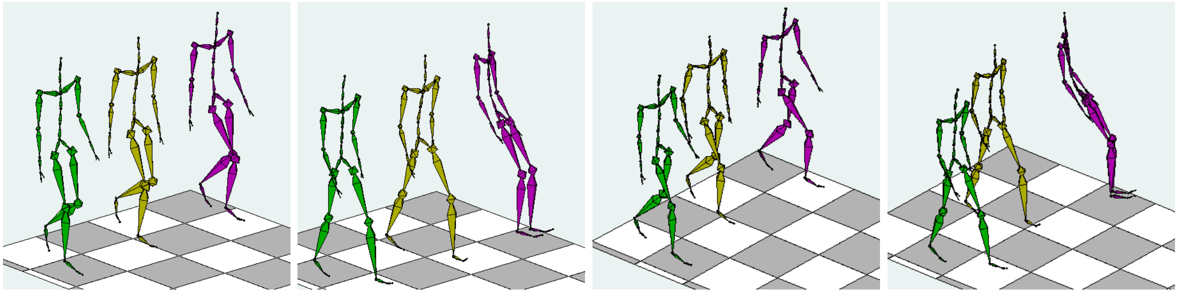


Figure 3: Screenshots from original motion (green), synthesized motion with our method (yellow) and synthesized motion with PCA based method (purple). Just two accelerations were used for reconstruction. In the right picture problems of the PCA method get visible that occur when using few markers.

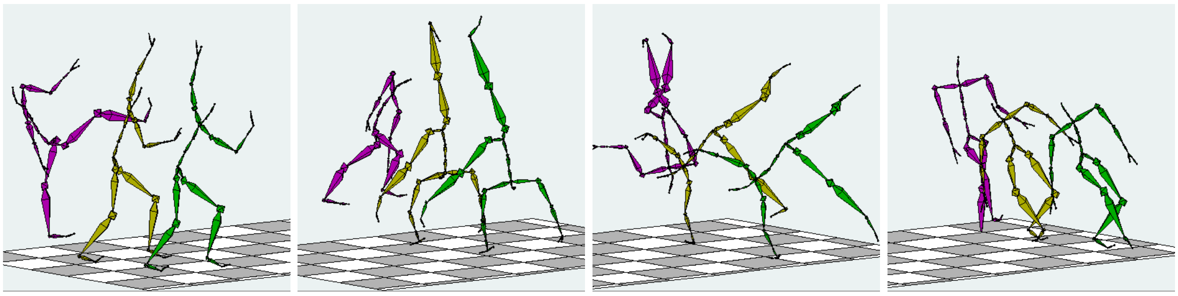


Figure 4: Some frames of an original cartwheel motion (green) and reconstructions (yellow our method, purple PCA based method). For both reconstructions we used acceleration data of the left foot and left hand only. One can see that the PCA based method shows massive errors in this case.

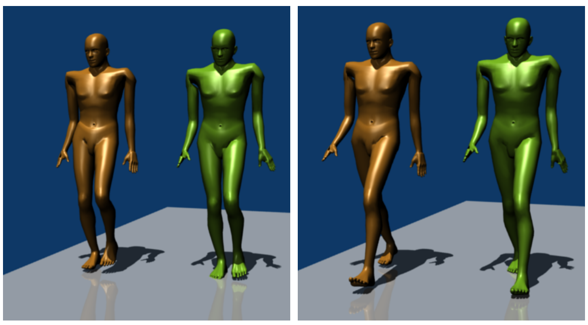


Figure 5: Comparison of two frames of an original walking motion (brown) and a reconstruction with our method (green). We only used the acceleration data of the left foot and the left hand to reconstuct these motions.

Our current implementation in Matlab is not suitable for any real-time requirements. Especially if one considers the very low latency requirements necessary for many tracking applications in VR and AR it might be necessary not only to use the parallel architectures of modern CPUs but also the parallel capabilities of modern GPUs as streaming processors to meet the requirements for on-line tracking. In addition to motion capturing, motion tracking, and motion puppetry our technique also seems to open the door to novel means of realistically synthesizing avatar motions from low dimensional control signals. In future work we will also explore this line of research.

Acknowledgements

The research is partially supported by *Deutsche Forschungsgemeinschaft* under grant We 1945/5-1. We are also grateful to the participants of the Dagstuhl seminar “Virtual Reality” for many inspiring discussions. Moreover, we thank the anonymous reviewers for several helpful comments.

References

[Bis84] Thomas Gary Bishop, *Self-tracker: a smart optical sensor on silicon (vlsi, graphics)*, Ph.D. thesis, University of North Carolina at Chapel Hill, 1984.

[Car04] Carnegie Mellon University Graphics Lab, *CMU Motion Capture Database*, 2004, `mocap.cs.cmu.edu`.

[CH05] Jinxiang Chai and Jessica K. Hodgins, *Performance animation from low-dimensional control signals*, ACM Trans. Graph. **24** (2005), no. 3, 686–696, SIGGRAPH 2005.

[Fox05] Eric Foxlin, *Pedestrian tracking with shoe-mounted inertial sensors*, IEEE Comput. Graph. Appl. **25** (2005), no. 6, 38–46.

[HBS07] T. Huynh, U. Blanke, and B. Schiele, *Scalable recognition of daily activities with wearable sensors*, Location- and Context-Awareness – Third International Symposium (LoCA 2007) (Oberpfaffenhofen, Germany) (J. Hightower, B. Schiele, and T. Strang, eds.), Lecture Notes in Computer Science, vol. 4718, Springer-Verlag, 2007, pp. 50–67.

[HW02] Mike Hazas and Andy Ward, *A novel broadband ultrasonic location system*, UbiComp ’02: Proceedings of the 4th international conference on Ubiquitous Computing (London, UK), Springer-Verlag, 2002, pp. 264–280.

[KGP02] Lucas Kovar, Michael Gleicher, and Frédéric Pighin, *Motion graphs*, ACM Transactions on Graphics **21** (2002), no. 3, 473–482, SIGGRAPH 2002.

[KTMW08] B. Krüger, J. Tautges, M. Müller, and A. Weber, *Multi-mode tensor representation of motion data*, Journal of Virtual Reality and Broadcasting (2008).

[KTW07] B. Krüger, J. Tautges, and A. Weber, *Multi-mode representation of motion data*, The 2nd International Conference on Computer Graphics Theory and Applications (GRAPP 2007) (Barcelona, Spain) (J. Braz, P.-P. Vzquez, and J. Madeiras Pereira, eds.), INSTICC Press, March 2007, pp. 21–29.

[LM06] G. Liu and L. McMillan, *Estimation of missing markers in human motion capture*, The Visual Computer **22** (2006), no. 9, 721–728.

[LZWM06] Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan, *Human motion estimation from a reduced marker set*, I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games (Redwood City, California), ACM Press, 2006, pp. 35–42.

[MJKM04] Nathan Miller, Odest C. Jenkins, Marcelo Kallmann, and Maja J. Matarić, *Motion capture from inertial sensing for untethered humanoid teleoperation*, Proceedings of the IEEE-RAS International Conference on Humanoid Robotics (Humanoids) (Santa Monica, CA), November 2004.

[MK06] Tomohiko Mukai and Shigeru Kuriyama, *Multilinear motion synthesis using geostatistics*, ACM SIGGRAPH / Eurographics Symposium on Computer Animation - Posters and Demos, 2006, pp. 21–22.

[MR06] Meinard Müller and Tido Röder, *Motion templates for automatic classification and retrieval of motion capture data*, SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM Press, 2006, pp. 137–146.

[MRC+07] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber, *Documentation: Mocap Database HDM05*, Computer Graphics Technical Report CG-2007-2, Universität Bonn, 2007, Data available at `http://www.mpi-inf.mpg.de/resources/HDM05/`.

[RCO05] Kalanov Rovshan, Jieun Cho, and Jun Ohya, *D-12-79 a study of synthesizing new human motions from sampled motions using tensor decomposition*, Proceedings of the IEICE General Conference **2005** (2005), no. 2, 229.

[RNd+07] Ramesh Raskar, Hideaki Nii, Bert deDecker, Yuki Hashimoto, Jay Summet, Dylan Moore, Yong Zhao, Jonathan Westhues, Paul Dietz, John Barnwell, Shree Nayar, Masahiko Inami, Philippe Bekaert, Michael Noland, Vlad Branzoi, and Erich Bruns, *Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators*, ACM Trans. Graph. **26** (2007), no. 3, 36.

[Val02] Nicholas Michael Vallidis, *Whisper: a spread spectrum approach to occlusion in acoustic tracking*, Ph.D. thesis, University of North Carolina at Chapel Hill, 2002.

[Vas02] M. Alex O. Vasilescu, *Human motion signatures: Analysis, synthesis, recognition*, Proc. Int. Conf. on Pattern Recognition (Quebec City, Canada), 2002, pp. 456–460.

[VAV+07] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović, *Practical motion capture in everyday surroundings*, SIGGRAPH '07: ACM SIGGRAPH 2007 papers (New York, NY, USA), ACM Press, 2007, p. 35.

[VBPP05] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović, *Face transfer with multilinear models*, ACM Trans. Graph. **24** (2005), no. 3, 426–433, SIGGRAPH 2005.

[WF02] Greg Welch and Eric Foxlin, *Motion tracking: No silver bullet, but a respectable arsenal*, IEEE Comput. Graph. Appl. **22** (2002), no. 6, 24–38.

[ZH07] H. Zhou and H. Hu, *Upper limb motion estimation from inertial measurements*, International Journal of Information Technology **13** (2007), no. 1.