

Data-Driven Completion of Motion Capture Data

Jan Baumann¹, Björn Krüger², Arno Zinke³ and Andreas Weber²

¹Fraunhofer FKIE, Wachtberg

²Bonn University, Institute of Computer Science II

³GfaR mbH, Bonn

email: {baumannj,kruegerb,zinke,weber}@cs.uni-bonn.de

Abstract

We present a data-driven method for completion of corrupted marker-based motion capture data. Our novel approach is especially suitable for challenging cases, e.g. if complete marker sets of multiple body parts are missing over a long period of time. Without the need for extensive preprocessing we are able to fix missing markers across different actors and motion styles. Our approach can be used for incrementally increasing prior-databases, as the underlying search technique for similar motions scales well to huge databases.

Keywords: motion capture, data cleaning, data driven methods

1. Introduction

Optical motion capture is the standard technique for creating realistic human motions in computer animation: Multiple cameras are used to track markers which are attached to an actor's body. Finally, 3D trajectories of the individual markers are reconstructed from the two dimensional images by triangulation techniques. Using fitting techniques, skeleton abstractions may be computed.

Although the topic of cleanup of motion capture data is a classical one and various cleaning techniques are available in commercial mocap system software, the problem is far from being solved and has obtained renewed attention in the last years [LMPF10, LC10, XFH11].

If gaps in several markers occur for a longer period of time—a scenario quite common if closely interacting actors are captured simultaneously or interaction with the environment is essential and hence occlusion of several markers over longer time periods occur—all of the existing approaches have major limitations, especially if no previously captured motions of *the same actor* which are similar to the one to be cleaned are available.

In this paper we present a general framework for data-driven filling of gaps in marker-based mocap data. The novel approach can handle challenging cases, especially if complete marker sets of multiple body parts are missing over a long period of time. Without the need for extensive preprocessing we are able to fix missing markers across different actors and motion styles. The results agree with human intuition and key features of the original input motion are greatly retained.

2. Related Work

Rudimentary gap filling is available in commercial software systems like Vicon IQ or Blade [Vic]. These methods rely on simple interpolation techniques, such as linear and spline interpolation of marker trajectories and thus fail if curvature changes sign. Moreover, such simple methods do not account for correlated motion of markers (e.g. for markers attached to the same body segment). For this reason, the above mentioned software systems also provide methods to recover a missing marker from a group of other markers if a rigid relationship between both the marker and the group may be assumed. However, this requires at least three other markers or joint positions relative to the missing marker's segment to be present in the gap.

Herda et al. [HFP*00] developed a skeleton based marker tracking and reconstruction technique to infer the positions of missing markers by using kinematic information provided by the underlying skeleton and the markers' positional data from previous frames that are attached to the same bone.

This method is applicable to short time occlusions of single markers, but it fails if entire segments are occluded for extended periods of time.

Kalman filters have been used in [DU03] to predict the trajectories of missing markers. However, Kalman filter based approaches fail when markers are missing for an extended time period or are missing entirely.

Li et al. [LMPF10] propose a method for occlusion filling of marker data by learning a linear dynamical system that respects inter-marker distances. However, this method relies on the existence of other markers on the same segment to make inter-marker distance measurements possible at all.

A data-driven method, which uses a piecewise linear modeling approach, was proposed by Liu and McMillan [LM06] for estimating missing markers.

Recently, additional data-driven methods for cleaning motion capture data have been proposed [LC10, XFH11]. Lou and Chai [LC10] were able to filter corrupted motion data by learning a series of spatial-temporal filter bases from prerecorded motion data. Using their filtering approach in a nonlinear optimization framework they were able to reduce noise, remove outliers and fill gaps while keeping the spatial-temporal patterns of the filtered human motion intact. Their method requires building of a training database in a time consuming pre-processing step which exclusively contains motions similar to the motion to be cleaned. Thus, in contrast to our method, it cannot handle different motion styles simultaneously without expensive preprocessing. Xiao et al. [XFH11] devised a method for filling gaps by representing incomplete poses by a linear combination of a few poses from a training set. Their approach as well as the work in [LC10] requires the training mocap data to be cleaned and to contain similar motion patterns (of the same actor) as the input motion. Moreover, the robustness of their approach to additional unrelated data in the training set was not discussed.

The problem of pose and motion reconstruction from sparse markers has also been the topic of various papers. In [GMHP04] and its accompanying video, the authors show the reconstruction of motion from sparse marker data. Although the results of their method are visually appealing, it largely depends on a specifically learned model that fails to capture the natural diversity of human motion. In [CH05], Chai and Hodgins show how to transform the positions of a small number of markers to full body poses. They construct a *neighbor graph* with the poses of the prior database as vertices. In a preprocessing step, an edge between two poses is added to the graph if the poses are near each other. This limits the NN-search to poses already in the database and can give only approximate results if the query pose is not contained within the available example motions. Due to its quadratic preprocessing time, it does not scale well with respect to the size of the database. Moreover, in the optimization step, the synthesized motion depends on the positional

information contained in the prior database while completely ignoring the temporal evolution (e.g. velocities and accelerations) of the local model. This might be an issue at turning points in the motion's trajectory. In our method, we incorporate this additional information yielding smooth and natural results. Krüger et al. [KTWZ10] improve on the method presented in [CH05] by using a kd-tree for determining the neighborhood of a query pose resulting in exact neighborhoods for arbitrary query poses.

Since our method is data-driven, it uses motions from a mocap database to construct a prior-database. Currently, the largest freely available database is the Carnegie Mellon University mocap database [Car04], which contains 2605 trials in 6 categories and 23 subcategories. Another large database, the HDM05 library [MRC*07], was recorded at the Hochschule der Medien in Stuttgart and contains more than three hours of systematically recorded and well documented motion capture data. Both of these databases provide the data in c3d as well as asf/amc data format.

3. Overview

Our approach takes advantage of data driven techniques. For that reason we need a mocap-database containing motions which are comparable to the clip to be processed by our method. One fundamental assumption of the proposed method is that all poses contained in the database as well as the motion to be cleaned share the same marker set. Furthermore, we assume that valid markers — i.e. the set of markers that are assumed to contain reliable positional information — are given for each frame of the input motion to be completed.

In a preprocessing step all mocap data from the prior-database is first normalized with respect to global position and orientation. Based on normalized positional data of valid markers we then build an efficient spatial indexing structure (kd-tree). In addition, linear marker velocities as well as accelerations are stored. These quantities are required for prior-based motion synthesis. For more details regarding this preprocessing step we refer to Sect. 4.1.

Subsequently, missing markers are synthesized for a given motion clip using nonlinear optimization. To this end similar examples from the database are retrieved by kd-tree based nearest neighbor search. These examples serve as priors to drive the synthesis process as discussed in Section 4.2. The whole pipeline of the proposed method is sketched in Fig. 1.

4. Workflow

4.1. Preprocessing

Our method is inspired by the solution to the *pose matching* problem presented by Krüger et al. [KTWZ10]. Here, the key idea was to analyze similarity of poses by employing

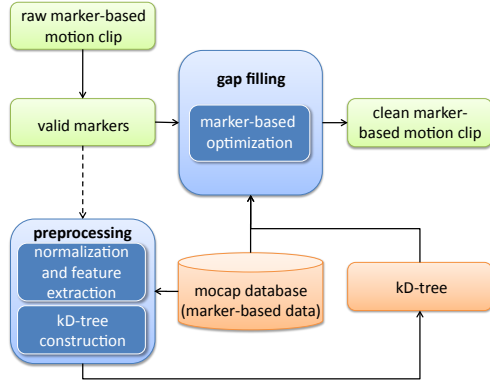


Figure 1: Workflow of the proposed method.

kd-tree based k -nearest-neighbor-search in dedicated feature spaces. Please note that this approach requires normalizing all poses with respect to global orientation and position. As — in contrast to Krüger et al. [KTWZ10] — no skeleton representation but point cloud data is given these quantities are estimated by exploiting rigid connectivity between valid markers.

Let \mathbf{x} be a pose vector involving M markers, where components are given by positional marker data. We then search for the k nearest neighboring poses $(\mathbf{y}_i), i \in \{1, \dots, k\}$ using a subset of all markers. The actual choice of the subset is based on two different criteria. First of all we consider only reliable markers where the placement is well-defined according to the markerset, such as knee and elbow markers for the standard markerset we use. We will formalize this first criterion by a static bitvector $(\tilde{\mathbf{m}} = \tilde{m}_i), i \in \{1, \dots, M\}$ that determines if a marker is suitable for k -nn search (one) or not (zero). Second, we only consider markers that are valid according to the capturing logic. In particular we assume that marker data is available. Such markers are indicated by another bitvector $\mathbf{m} = (m_i), i \in \{1, \dots, M\}$. In contrast to $\tilde{\mathbf{m}}$, which is independent of the motion to be cleaned, the bitvector \mathbf{m} is computed per gap.

Once we have selected viable markers, i.e. markers with $\tilde{m}_i = m_i = 1$, their respective coordinates form a vector space that is used for building a kd-tree from all motion data included in the database. As missing markers may depend on the actual motion, this kd-tree is build from scratch for each cleaning process. Please note, that building this tree takes only a few seconds even for the largest currently available databases and thus not resembles a bottleneck of our method.

4.2. Gap Filling

We perform a search for k nearest neighbor poses for each pose that requires to be cleaned by our technique. To this end each of the given frames is normalized with respect to position and orientation, similar to the data in the knowledge

Table 1: List of markers usable as features. To determine reliable markers that are (if valid) suitable for k -nearest-neighbor-search, a bitvector $\tilde{\mathbf{m}}$ is used. Here, for each marker of the markerset, a component indicates if a marker is usable (1) or not reliable (0).

Label	\tilde{m}	Label	\tilde{m}	Label	\tilde{m}
LFHD	1	LWRB	1	LKNE	1
RFHD	1	LFIN	1	LSHN	0
LBHD	1	RSHO	0	LANK	1
RBHD	1	RUPA	0	LHEE	1
C7	1	RELB	1	LTOE	0
T10	0	RFRM	0	LMT5	0
CLAV	0	RWRA	1	RTHI	0
STRN	1	RWRB	1	RKNE	1
RBAC	0	RFIN	1	RSHN	0
LSHO	0	LFWT	0	RANK	1
LUPA	0	RFWT	0	RHEE	1
LELB	1	LBWT	0	RTOE	0
LFRM	0	RBWT	0	RMT5	0
LWRA	1	LTHI	0		

base. We retrieve a set $(\mathbf{y}_i), i = [1..k]$ of k nearest neighbors that can be used for the data-driven gap filling procedure.

The gap filling procedure employs prior-driven optimization to synthesize the positional data of missing markers. We use an energy minimization formulation which is frequently used in data driven computer animation. Our specific choice of the energy terms to be minimized most closely resembles the one used in [TZK*11]. Here, the objective function is consisting of three different terms: pose E_{pose} and motion priors E_{smooth} and E_{motion} enforcing positions, acceleration and velocities of the missing markers to be comparable to examples retrieved from the database.

$$\mathbf{x}_{\text{best}} = \underset{\mathbf{x}}{\operatorname{argmin}} (E_{\text{pose}}(\mathbf{x}) + E_{\text{motion}}(\mathbf{x}) + E_{\text{smooth}}(\mathbf{x})) \quad (1)$$

4.2.1. Prior Terms

Linear velocities and accelerations have been previously computed for all motion clips contained in the mocap database used for cleaning. Let $(\mathbf{y}_i), i = [1..k]$ be the poses retrieved from the database by k -nearest-neighbor search and $(\mathbf{v}_i), i = [1..k]$ and $(\alpha_i), i = [1..k]$ the respective velocities and accelerations and let $\mathbf{v}(\mathbf{x})$ and $\alpha(\mathbf{x})$ be the velocity and acceleration of a given pose. We then use kernel regression for each of the prior terms along the lines of [TZK*11] considering only markers that are assumed to be invalid:

$$E_{\text{pose}}(\mathbf{x}) = \sum_{i=1}^k (\tilde{\mathbf{m}} \circ (\mathbf{y}_i - \mathbf{x}))^2, \quad (2)$$

$$E_{\text{motion}}(\mathbf{x}) = \sum_{i=1}^k (\tilde{\mathbf{m}} \circ (\mathbf{v}_i(\mathbf{y}_i) - \mathbf{v}(\mathbf{x})) \cdot \Delta t)^2, \quad (3)$$

$$E_{\text{smooth}}(\mathbf{x}) = \sum_{i=1}^k (\bar{\mathbf{m}} \circ (\alpha_i(\mathbf{y}_i) - \alpha(\mathbf{x})) \cdot \Delta t^2)^2 \quad (4)$$

with $\bar{\mathbf{m}}$ denoting the component wise inversion of the bitvector \mathbf{m} . Please note that for all the above priors only markers that are assumed to be invalid are considered by Hadamard vector multiplication.

4.3. Optimization Procedure

The objective function (1) is minimized using gradient descent. To improve efficiency, only a subset of all frames is considered during optimization. This includes frames with the highest associated costs as well as neighboring frames indirectly affecting reconstruction results through temporal derivatives occurring in motion and smoothness priors. We refer to this as *scheduling*.

To improve the robustness of our method and to speed up the process of optimization, we employ a multi-resolution approach, where the optimization takes place on subsequently higher resolutions of the motion to be cleaned, starting with the lowest. This requires resampling the motion to a predefined number of lower resolutions. When the error on a certain resolution cannot be improved by at least a certain threshold (that we set to 1%), the algorithm upsamples the results and switches to the next higher resolution. Given the number of resolutions n and the highest resolution r_{max} , we calculate lower resolutions r_i by

$$r_i = \frac{r_{\text{max}}}{2^i}. \quad (5)$$

For every possible resolution, positions, velocities and accelerations have to be precomputed in the prior-database. Moreover, separate kd-trees have to be created. Please note that the memory requirements of the multiscale approach is bounded by twice the original data.

5. Results

In our tests, in order to evaluate the effectiveness of our method, we took originally artifact-free mocap data and discarded certain markers or sets of markers representing body segments for various time spans. The reconstruction results were analyzed both visually and numerically. Besides synthetic test cases our method was also employed on data containing real gaps.

For a visual comparison we refer to the accompanying video, showing:

1. Examples of real gaps in original marker data taken from the HDM05 database [MRC*07].
2. Reconstruction of a motion with missing left arm markers.
3. Gap-filled Cartwheel motion with leg markers missing.
4. Comparison of databases according to section 5.1.2.

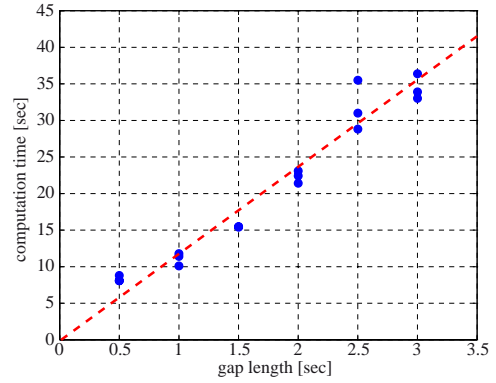


Figure 2: Scatter plot of computation times for various examples with respect to the length of the filled gap.

5. Example of a running motion that was presented and reconstructed in [LMPF10]. For this example the complete CMU database [Car04] was used as prior-database.
6. Comparison of reconstructions of a walking motion based on [KTWZ10] and our method.
7. Reconstructions of gaps found in real mocap data.

In Fig. 2 we give the computation times for various examples in dependency of the length of the filled gaps. As had to be expected, the computation time scales linearly with the length of the gaps. There are certain variations with respect to the used motion classes and numbers of missing markers, but these effects yield much smaller variations than the primary dependency on the gap length.

The computation times are obtained using a single threaded implementation on a Dual Core 3 GHz PC with 8GB of memory. Roughly speaking the computation times are about 10 times the length of the longest gaps for this implementation. Hence it is already practical for interactive applications even without having performed code optimization and using multi-threading.

Our experiments show that our method is able to fill gaps in motions ranging from a single marker missing to multiple body segments missing for up to several seconds.

5.1. Evaluation on synthetic examples

In this section we report on a series of tests on synthetic examples. We evaluate several aspects of the proposed method. For this reason we removed markers from intact motion sequences to compare our results with ground truth data. We computed all results on motions resampled to 30 Hz.

5.1.1. Tests on single missing markers

We systematically removed each marker for three test motions taken from the HDM05 database. The test motions are:

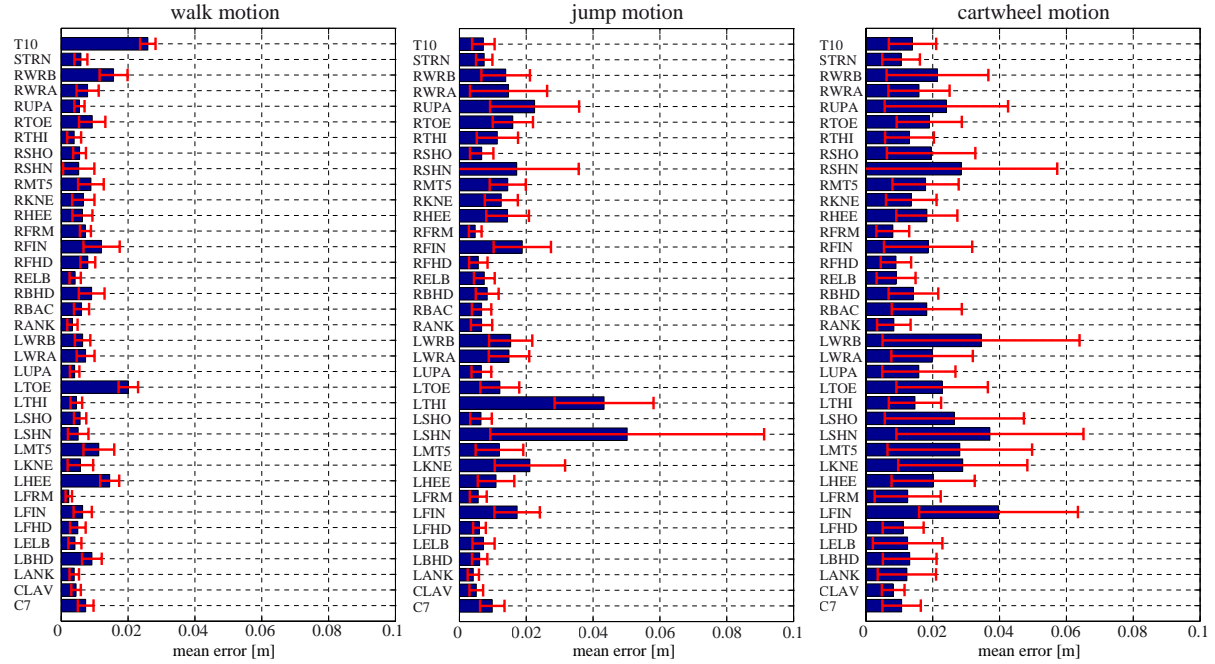


Figure 3: Mean distances (blue bars) between original marker trajectories and synthesized trajectories and the corresponding standard deviations (red lines) are shown for three testing motions, where the indicated marker was removed.

1. walk: HDM_bd_01_01_01_120.c3d, frames 650 – 1100
2. jump: HDM_tr_01_05_01_120.c3d, frames 1000 – 1350
3. cartwheel: HDM_tr_05_03_03_120.c3d, frames 2550 – 3000

The database used for these experiments included all motions from the HDM05 database except the whole take the test sequence was taken from. Figure 3 shows the results of this test. The mean distance between the original and the synthesized markers as well as the standard deviation of this distance is presented. As can be seen on the left of Figure 3, the walking motion gives very good results, showing a mean of 0.77 cm over all examples. For the more complex jump and cartwheel motions the means are 1.27 cm and 1.81 cm, respectively.

5.1.2. Tests on groups of missing markers

On the three motions that were used on the single marker tests we performed tests where several groups of markers were removed simultaneously. For each test, we removed the six markers of the segments of the left arm. For the walk and jump motion, these markers will not be in contact with the ground, whereas for the cartwheel motion a contact of the left arm with the ground occurs. For these tests we regarded the following scenarios

1. The prior-database does not contain motions of the actor of the test motion.
2. The prior-database contains motions of various performers, including motions of the actor (other than the test motion).
3. Only motions of the actor were included in the prior-database.

The results of the tests are summarized in Fig. 4. If motions of the actor are not contained in the prior-database, the average reconstruction errors are more than twice as high as in the other cases for all three examples. However, the reconstruction results are still good, with a mean error ranging from 2.5 cm for the walking motion to 5 cm for the cartwheel motion. Moreover, the reconstructed motions have a high visual fidelity (see accompanying video).

In another test suite to estimate the influence of properties of the performing actor, we performed left-out tests for any of the five actors performing walking motions in the HDM05 database. Again, we reconstructed removed marker positions for the left arm. For this experiment we used the takes HDM_**_01_01_01_120.C3D which were performed by each actor for this test.

5.2. Comparison with previous work

We compared our results with the motion reconstruction technique described by Krüger et al. [KTWZ10] which is

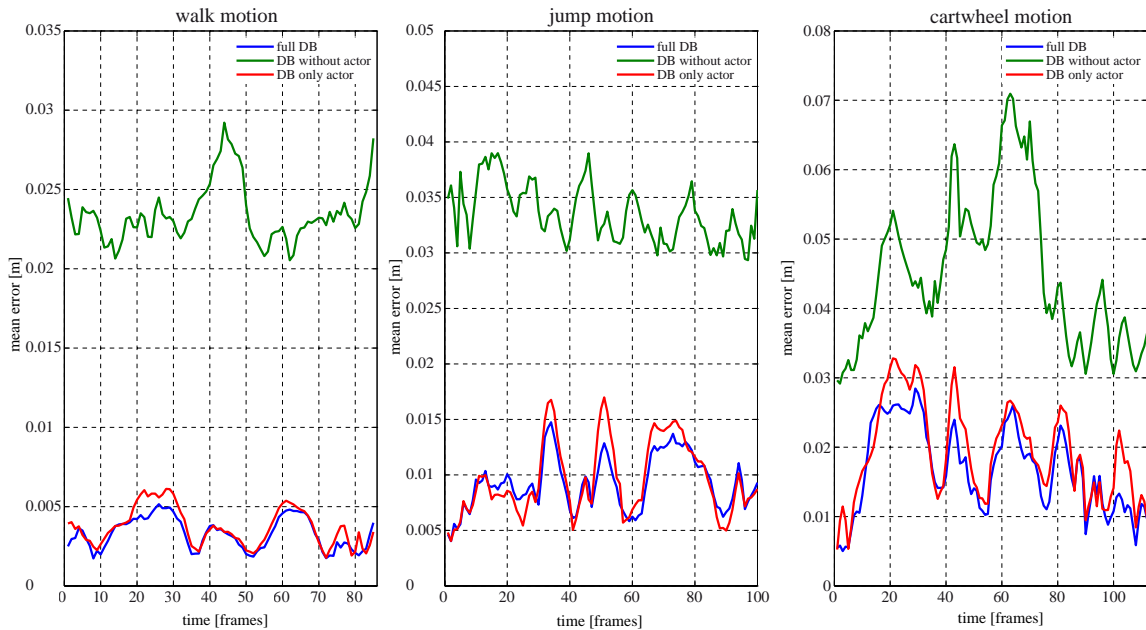


Figure 4: Mean distances over all markers for three motion sequences. The distances are presented for three different databases: The full database, excluding only the test motion (blue), a database where the actor was completely removed (green) and a database where only motions from the actor were included (red).

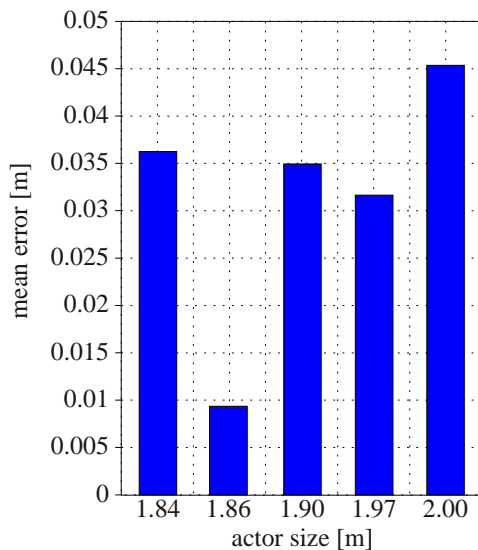


Figure 5: Results of test with actors of different sizes. We plot the mean reconstruction error for a walking sequence (HDM_**_01_01_01_120.C3D) versus the body size of the actors. The actors were not included in the database for this experiment.

an extension, based on fast similarity searches, of the technique originally described by Chai and Hodgins [CH05]. For this comparison we regarded six scenarios of missing data: left arm, right arm, both arms, left leg, right leg and both legs. Since our method only reconstructs missing marker positions, a skeleton was fitted to the ground truth and reconstructed marker data using the method of de Aguiar et al. [dATS06]. To make the marker fitted skeleton comparable to the standard asf/amc skeleton used in the HDM05 library, a set of joints was selected as the intersecting set of joints of both skeleton topologies.

A standard error measure to compare different reconstructions is to calculate the average distance between this set of joints of the reconstructed and the ground truth motion for every frame. The mean reconstruction error is then calculated as the mean over all frames. The numeric results of both reconstructions are given in table 2.

Our approach uses multi resolution optimization combined with scheduling and a more sophisticated prior model to produce natural looking results while Krüger et al. use ad-hoc smoothness and framewise optimization combined with a kd-tree nearest neighbor search. Although in most cases our method proves to be numerically better using the standard error measure, the difference in the visually perceived quality of the results can be seen in the accompanying video.

Table 2: Results for motion reconstructions based on the method presented in this paper (our), compared to reconstructions based on the method of Krüger et al. [KTWZ10]. The table gives the mean reconstruction error in centimeters.

motion	method	scenario					
		left arm	right arm	both arms	left leg	right leg	both legs
HDM_bd_01-01_01_120 Frames: 650 – 1100	our	0.64	0.59	1.23	0.88	0.82	2.47
	KTWZ10	1.00	1.56	2.29	1.35	1.48	4.55
HDM_bd_01-02_03_120 Frames: 450 – 750	our	1.20	1.18	3.63	1.60	2.20	4.00
	KTWZ10	1.81	2.43	4.71	1.63	2.19	4.62
HDM_bk_01-01_03_120 Frames: 6300 – 6600	our	1.29	1.47	4.80	1.27	1.61	4.24
	KTWZ10	2.05	2.44	5.91	1.86	2.65	6.26
HDM_mm_02-03_02_120 Frames: 450 – 750	our	1.00	3.50	10.20	0.95	1.11	2.00
	KTWZ10	2.89	4.46	12.07	0.97	1.37	1.85
HDM_dg_01-06_01_120 Frames: 1000 – 1300	our	1.20	0.94	1.81	1.70	1.58	3.56
	KTWZ10	1.68	1.26	3.69	1.42	1.59	4.68

6. Conclusion and Future Work

We have presented a data driven method for filling large gaps in marker based mocap data. Our method works well even for large gaps from the perspective of required computational resources as well as quality of results—provided that there are sufficiently similar motions available in the prior-database. The basic mechanism can be extended to other cleaning and reconstruction tasks, such as optimal skeleton fitting and correcting marker-mislabelings. These extensions will be one topic of future work.

In contrast to previous approaches we can keep all available and cleaned motion capture data in our prior-database, and our approach scales well to huge prior-databases. The quality of our gap filling methods depends on the similarity of data contained in the prior-database and we obtain somewhat better results if motions of the performing actor of the clip to be cleaned are already contained in the prior-database. Nevertheless, our method also works quite well if such data is not available. In our approach it is possible in principle to incorporate model knowledge about skeleton constraints and contact constraints. Using a good algorithmic heuristic to estimate contact constraints from motion data—e.g. the method presented in [LCB06]—the contact information can be incorporated into the search and all defined constraints can be incorporated into the optimization procedure. We presume that such information is useful in all settings and might be crucial if for a gap-filling the information of body segments such as lower-body parts or upper body parts only are considered. Such restrictions to body parts allow an extension of the notion of “similar motion” to ones being similar for body parts only.

In our future work we will explore the algorithmic techniques and will perform empirical investigations for incremental extension of the prior-databases: cleaned motion clips can be incrementally added to the prior-database potentially allowing a step-wise extension of the expressibility of the prior-database. With such extensions motions which

could not be handled by an original prior-database might become tractable by the newly added clips.

The scenario of missing markers on entire body segments for longer periods of capture-time is a common challenge even for single user capture using practical low-cost equipment such as the KINECT. The integration of our algorithms into a capturing and processing pipe-line for such low-cost devices will be a topic of our future work.

7. Acknowledgements

Jan Baumann was partially supported by *Deutsche Forschungsgemeinschaft* under Grant WE 1945/5-1.

References

- [Car04] CARNEGIE MELLON UNIVERSITY GRAPHICS LAB: CMU Motion Capture Database, 2004. mocap.cs.cmu.edu. 2, 4
- [CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. *ACM Trans. Graph.* 24 (July 2005), 686–696. 2, 6
- [dATS06] DE AGUIAR E., THEOBALT C., SEIDEL H.-P.: Automatic learning of articulated skeletons from 3d marker trajectories. In *ISVC (2006)* (2006). 6
- [DU03] DORFMÜLLER-ULHAAS K.: *Robust Optical User Motion Tracking Using a Kalman Filter*. Tech. rep., Universitätsbibliothek der Universität Augsburg, Universitätsstr. 22, 86159 Augsburg, 2003. 2
- [GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based inverse kinematics. *ACM Transactions on Graphics* 23, 3 (2004), 522–531. SIGGRAPH 2004. 2
- [HFP*00] HERDA L., FUA P., PLANKERS R., BOULIC R., THALMANN D.: Skeleton-based motion capture for robust reconstruction of human motion. In *Computer Animation* (Philadelphia, USA, 2000). 1
- [KTWZ10] KRÜGER B., TAUTGES J., WEBER A., ZINKE A.: Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*

- (Aire-la-Ville, Switzerland, Switzerland, July 2010), SCA '10, Eurographics Association, pp. 1–10. [2](#), [3](#), [4](#), [5](#), [7](#)
- [LC10] LOU H., CHAI J.: Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics 16* (2010), 870–879. [1](#), [2](#)
- [LCB06] LE CALLENNEC B., BOULIC R.: Robust kinematic constraint detection for motion data. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (sept 2006). [7](#)
- [LM06] LIU G., McMILLAN L.: Estimation of missing markers in human motion capture. *The Visual Computer 22*, 9 (2006), 721–728. [2](#)
- [LMPF10] LI L., McCANN J., POLLARD N., FALOUTSOS C.: Bolero: a principled technique for including bone length constraints in motion capture occlusion filling. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2010), SCA '10, Eurographics Association, pp. 179–188. [1](#), [2](#), [4](#)
- [MRC*07] MÜLLER M., RÖDER T., CLAUSEN M., EBERHARDT B., KRÜGER B., WEBER A.: *Documentation: Mocap Database HDM05*. Computer Graphics Technical Report CG-2007-2, Universität Bonn, May 2007. www.mpi-inf.mpg.de/resources/HDM05. [2](#), [4](#)
- [TZK*11] TAUTGES J., ZINKE A., KRÜGER B., BAUMANN J., WEBER A., HELTEN T., MÜLLER M., SEIDEL H.-P., EBERHARDT B.: Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.* 30 (May 2011), 18:1–18:12. [3](#)
- [Vic] VICON MOTION CAPTURE SYSTEMS.: <http://www.vicon.com/>. [1](#)
- [XFH11] XIAO J., FENG Y., HU W.: Predicting missing markers in human motion capture using l1-sparse representation. *Computer Animation and Virtual Worlds 22*, 2-3 (2011), 221–228. [1](#), [2](#)