

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/326307406>

Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging

Article in PLoS ONE · July 2018

DOI: 10.1371/journal.pone.0199744

CITATIONS

3

READS

193

3 authors:



Mickaël Tits

Centre d'Excellence en Technologies de l'Information et de la Communication

13 PUBLICATIONS 15 CITATIONS

[SEE PROFILE](#)



Joëlle Tilmanne

Université de Mons

51 PUBLICATIONS 355 CITATIONS

[SEE PROFILE](#)



Thierry Dutoit

Université de Mons

324 PUBLICATIONS 5,009 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Biomanufacturing [View project](#)



Expert gesture analysis through motion capture using statistical modeling and machine learning [View project](#)

RESEARCH ARTICLE

Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging

Mickaël Tits*, Joëlle Tilmanne, Thierry Dutoit

Numediart Institute, University of Mons, Mons, Belgium

* mickael.tits@umons.ac.be



OPEN ACCESS

Citation: Tits M, Tilmanne J, Dutoit T (2018) Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging. PLoS ONE 13(7): e0199744. <https://doi.org/10.1371/journal.pone.0199744>

Editor: Peter Andreas Federolf, University of Innsbruck, AUSTRIA

Received: February 8, 2018

Accepted: June 13, 2018

Published: July 10, 2018

Copyright: © 2018 Tits et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant files (data and code) are available from the github repository: <https://github.com/numediart/MocapRecovery>. All motion capture data files used for quantitative analysis are also available from the CMU and HDM05 databases: <http://mocap.cs.cmu.edu/> and <http://resources.mpi-inf.mpg.de/HDM05/>.

Funding: Mickaël Tits is funded through a PHD grant from the Fonds pour la Formation à la Recherche dans l'Industrie et l'Agriculture (FRIA, FRS-FNRS, FC 005499), Belgium, <http://www.fnrs.be/en/>. The Qualisys motion capture equipment

Abstract

Motion capture allows accurate recording of human motion, with applications in many fields, including entertainment, medicine, sports science and human computer interaction. A common difficulty with this technology is the occurrence of missing data, due to occlusions, or recording conditions. Various models have been proposed to estimate missing data. Some are based on interpolation, low-rank properties or inter-correlations. Others involve dataset matching or skeleton constraints. While the latter have the advantage of promoting a realistic motion estimation, they require prior knowledge of skeleton constraints, or the availability of a prerecorded dataset. In this article, we propose a probabilistic averaging method of several recovery models (referred to as *Probabilistic Model Averaging* (PMA) in this paper), based on the likelihoods of the distances between body points. This method has the advantage of being automatic, while allowing an efficient gap data recovery. To support and validate the proposed method, we use a set of four individual recovery models, based on linear/nonlinear regression in local coordinate systems. Finally, we propose two heuristic algorithms to enforce skeleton constraints in the reconstructed motion, which can be used on any individual recovery model. For validation purposes, random gaps were introduced into motion-capture sequences, and the effects of factors such as the number of simultaneous gaps, gap length and sequence duration were analyzed. Results show that the proposed probabilistic averaging method yields better recovery than (i) each of the four individual models and (ii) two recent state-of-the-art models, regardless of gap length, sequence duration and number of simultaneous gaps. Moreover, both of our heuristic skeleton-constraint algorithms significantly improve the recovery for 7 out of 8 tested motion-capture sequences ($p < 0.05$), for 10 simultaneous gaps of 5 seconds. The code is available for free download at: <https://github.com/numediart/MocapRecovery>.

Introduction

Motion capture (MoCap) is the process of recording motion data through any type of sensor. Because of the recent evolution of the digital world, motion capture techniques are increasingly

was funded by the Fonds de la Recherche Scientifique (FRS-FNRS, EQP U.N041.13), Belgium. This research project is supported by the European Regional Development Fund (ETR 1212 0000 3303). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

used for clinical purposes, sports science, as well as many applications in gaming, animation, and human computer interaction [1–6]. State-of-the-art MoCap systems rely on optical technologies, either marker-based (e.g. Vicon [7]), or markerless (e.g. Microsoft Kinect [8]). The former allow accurate tracking of optical markers (generally consisting of retro-reflective spheres) fixed on a suit. The latter use computer vision techniques like background suppression, silhouette extraction and skeleton reconstruction in RGB images or depth maps [9–11]. Besides their advantages and drawbacks, which is beyond the scope of this work, both techniques share an unavoidable issue: missing data. At recording time, if a marker or a body part is hidden from all cameras, its trajectory cannot be completely recorded, resulting in a gap in the MoCap data. Several issues may cause gaps, including occlusions, marker reflection quality, lighting condition, calibration or the limited area covered by the system. These gaps make it difficult and sometimes impossible to use the data [12–14]. A number of methods have already been proposed to address this issue, based on various techniques. One basic method is direct interpolation. From an incomplete trajectory of a marker, the coordinates over time can be interpolated using standard methods, such as linear, spline or monotone piecewise cubic interpolation [15], amongst others. Those methods are sufficient for small gaps (typically less than 0.5 second for human full-body motion [12]), but are ineffective for larger gaps. More advanced time-series interpolation methods have been proposed, based on linear dynamic systems [16], Gaussian process dynamic models [17], or Kalman filters [13].

Other methods are based on the fact that MoCap data generally consist of highly related trajectories of several markers, due to fixed bone length and to limited degrees of freedom in the skeleton. Expressing the incomplete trajectory using local coordinates, based on trajectories of three additional markers or based on a rigid body position and orientation, can be used to improve recovery [14]. Such coordinate transformation should reduce the variance of the trajectory representation, thereby easing the interpolation process. However, the three markers used for coordinate system transformation must have similar trajectories to the incomplete marker for the process to be efficient. This method thus highly depends on the number of complete marker trajectories available in the data.

Yet other methods for recovering missing data are based on human motion modeling, trained on a pre-recorded dataset. Liu and McMillan [12] trained a global linear model and a set of local linear models from a training set of MoCap data. The local models are defined using segmentation with probabilistic principal component analysis (PCA), and K-means clustering. They first used the global model to recover missing data, then, from the results, they assigned a local model to each frame using a Random Forest classifier. On the other hand, Chai and Hodgins [18] directly retrieved nearest neighbors of incomplete frames in a dataset, and trained a local linear model from these neighbors to recover the missing data. These methods are not fully automatic as they need a large dataset for the training of the models. Moreover, the data to recover must have the same marker disposition as the one used in these pre-trained models. It means that for a new type of MoCap data (with a different marker disposition), an entire dataset must be recorded to train a new model.

Finally, some methods are based on matrix transformation techniques, using PCA [19, 20], singular value thresholding (SVT) [21] or nonnegative matrix factorization (NMF) [22]. These methods consider the entire motion as a matrix, with columns representing 3D components of all marker trajectories, and allow the use of information based on linear relations between the columns to reconstruct a gap in the matrix. The transformations are all based on low-rank properties of MoCap data. A key point with these methods is that a low-rank model of motion is trained on the available data of the motion sequence itself, and does not require a training dataset. These methods are thus automatic and can be used on any MoCap data [23].

A drawback of all previous methods is that the recovered trajectories may not respect human body properties, including bones' fixed lengths. Motion animations may thus lead to unrealistic results. Yet, other methods are directly based on the body skeleton, forcing marker positions to respect these properties. These constraints were successfully applied to several previously mentioned methods. Li et al. [24] proposed the BoLeRO algorithm, combining skeleton constraints with linear dynamic systems. Tan et al. [25] proposed a skeleton constrained SVT algorithm. Peng et al. [22] adapted NMF to a hierarchical block-based skeleton structure model. However, such methods are generally significantly more computationally intensive as they are based on iterative optimization procedures. Moreover, they are often not automatic, as they are defined for a specific skeleton model based on a pre-defined marker set. Nonetheless, automatic procedures exist to estimate a skeleton structure in MoCap data [26, 27].

Each one of the methods mentioned so far has different advantages and drawbacks, possibly making them more or less effective according to different factors, including gap length, number of markers, motion speed and complexity, and total motion sequence duration. For instance, interpolation techniques are inherently independent of the duration of the entire motion sequence and of the number of markers, unlike matrix-based and machine learning based techniques. The latter indeed require training data based on the frames of the sequence without missing markers, to model the relationship between markers. The quality of the model thus depends on the size of the training set, i.e. sequence duration, and the number of markers. On the opposite, machine-learning techniques may be more robust to gap length or motion complexity than interpolation-based methods [12].

To the authors' knowledge, most previously proposed automatic MoCap data reconstruction methods are based on low-rank or temporal properties of motion, and use matrix operations to model human motion. Few papers focus on the use of machine-learning techniques such as linear and non-linear regression [28, 29] to model the motion of a missing marker. Moreover, no previous work known to the authors proposes the usage of ensemble learning to use likelihoods of different models and construct a more robust global model from the decisions of an ensemble of others [30, 31].

Therefore, the aim of this research is to propose a probabilistic averaging method that can be used with any ensemble of recovery models, and that enforces movement constraints. This method is referred to below as *Probabilistic Model Averaging* (PMA). The averaging process is based on the posterior likelihoods of the distances between the recovered body points and other markers. To validate the method, we used existing recovery models and developed four new regression-based recovery models, which were used as inputs to the proposed probabilistic averaging method.

Method

Fig 1 shows the overall approach of our data recovery method, which can be divided into different steps. First, parameters are extracted from each marker trajectory of the motion sequence. These parameters mainly represent relations between markers, and allow identification of related markers (termed as *reference markers* below) and of their distance distributions. Then, various recovery models are applied on the incomplete MoCap sequence, resulting in several candidate recovered sequences. For each candidate, a correction is applied to respect motion continuity (except for interpolation which inherently respects motion continuity). From all resulting individually recovered sequences, a weighted average is applied, in the spirit of ensemble learning systems [30]. Finally, a spacing constraint is applied on the recovered trajectory, enforcing plausibility of the distance with related markers.

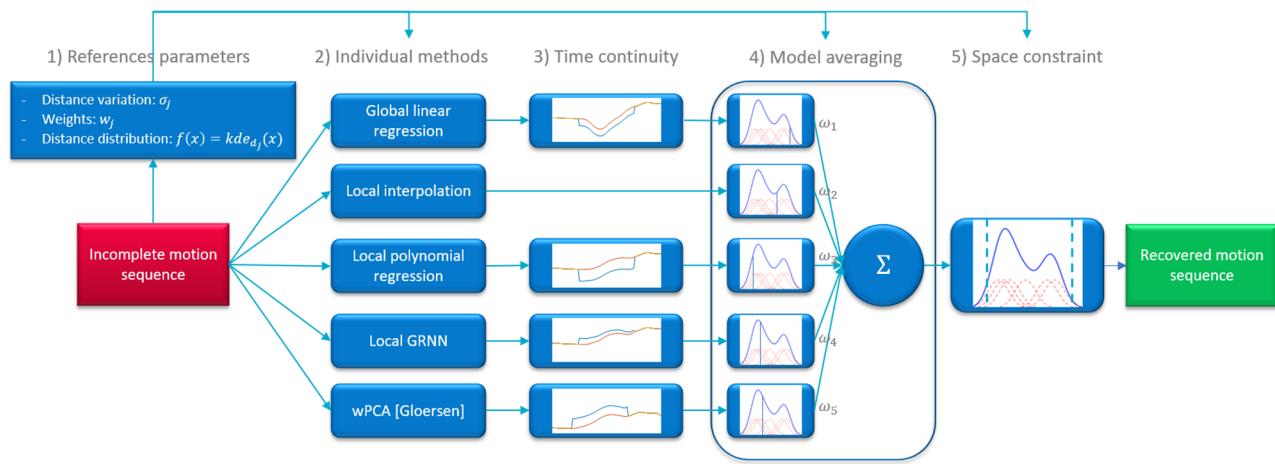


Fig 1. Block diagram of the proposed method. The overall process can be divided in five steps: 1) Extraction of marker trajectories parameters. 2) Individual recovery models. 3) Time constraint: trajectory continuity. 4) Distance-probability weighted averaging. 5) Spacing constraint: reference marker distance likelihoods.

<https://doi.org/10.1371/journal.pone.0199744.g001>

Note that it is good practice to center the motion sequence at the outset, by subtracting the mean position of all the markers available in each frame. This process makes it possible to reduce the component of global motion in the sequence, thus reducing motion variations. After the gap recovery process, the mean position is added back to translate the motion sequence to its original trajectories.

In the remainder of the paper, a motion sequence will be considered by a matrix representing all the trajectories of all the recorded markers during the entire sequence. This matrix has the dimension $N \times (3M)$, where N is the number of frames of the sequence, and M the number of recorded markers. A marker trajectory p_j ($j \in 1, \dots, M$) is represented as a matrix $N \times 3$.

The method was implemented with Matlab R2017a and the MoCap Toolbox [32]. The code is available for free download at: <https://github.com/numediar/MocapRecovery>.

Reference marks definition

All individual recovery models, as well as the proposed spacing constraint, depend on reference markers from which incomplete marker trajectories are to be recovered. We propose an automatic method to define such reference markers, based on inter-marker distance variations, inspired from skeleton-based methods where the intuition is based on the fact that joints of a skeleton have limited distance variations, due to bones' fixed lengths and limited degrees of freedom. We hypothesize that in many situations an incomplete marker trajectory can be recovered more effectively with methods using reference marker information, due to the close relation between the markers and their references. For instance, the wrist trajectory is more closely related to the elbow or shoulder trajectory than that of the foot. For each marker trajectory to be reconstructed, we can reorder related markers (referred to below as reference markers, or references) using distance standard variation. We denote by m the marker trajectory to recover, and p_j any other marker trajectory ($N - by - 3$). The distance variation is computed as the standard deviation of the Euclidean distance between m and p_j :

$$\sigma_j = \text{std}(\| (m - p_j) \|) \quad (1)$$

The marker p_j with the smallest σ_j is the marker most related to m , i.e. the best reference for reconstruction of m . All markers can then be sorted as potential references for reconstruction of m , according to their distance variation with it.

Table 1. Motion sequences used in the methods comparison.

File name	Type of motion	#markers	Duration (s)	#frames	fps
HDM_mm_01_02_03_120 (HDM1)	Locomotion on the spot	41	33.25	3990	120
HDM_mm_02_02_02_120 (HDM2)	Shelf (while walking)	41	70.83	8500	120
HDM_mm_03_02_01_120 (HDM3)	Kicking and punching	41	63.93	7671	120
HDM_mm_04_01_02_120 (HDM4)	Chair, table, floor	41	62.79	7535	120
HDM_bd_05_01_01_120 (HDM5)	Clapping and waving	41	57.94	6953	120
85_02 (CMU1)	Break dance (Jumptwist)	41	6.75	810	120
85_12 (CMU2)	Break dance	41	37.5	4499	120
135_02 (CMU3)	Martial art (Empi)	41	43.35	5202	120

Motion files from CMU and HDM05 MoCap databases used to test the methods.

<https://doi.org/10.1371/journal.pone.0199744.t001>

Model 1: Global linear regression

An incomplete marker trajectory can be modeled based on trajectories of all present markers as input information. In the present context of human motion data, we have a complex problem involving underlying non-linear relationships, with potentially important quadratic or circular components, due to angular motions of skeleton segments. However, for the method to be fully automatic, the model training data are limited to the incomplete sequence to recover itself, i.e. N frames minus the missing frames.

Because many input variables are considered for the design of the model ($3 \times M$), the model must be simple, to avoid overfitting. Accordingly, we selected linear regression [28] for that task. Assuming that marker with lower distance variation σ_j are more suited for predicting the missing marker trajectory, we defined a threshold θ on σ_j , and avoided markers with $\sigma_j > \theta$ to simplify the model. This way, only relevant markers (numbered M_X) are used to model the position of the incomplete marker. This threshold was experimentally set to 50 mm in this research, as it gave the best results for the dataset, consisting of eight motion sequences with 41 markers (see Results, Table 1). Intuitively, a larger threshold could be considered for data with fewer available markers.

Let m be missing between frames n_1 and n_2 . In other words, the rows from n_1+1 to n_2-1 of the matrix m must be recovered. A linear regression is performed on each axis i of m . Let $X = [p_j]_{j:\sigma_j < \theta, p_j \neq m}$ be the design matrix $N - by - (3 \times M_X)$, including all marker trajectories p_j with $\sigma_j < \theta$, except the incomplete marker trajectory itself m (in practice, we add an intercept column to the design matrix, all missing frames are excluded, and we use only markers always present during the gap of m to recover). Denote by $X(n)$ as the n -th row of X , m_i the i^{th} column (or spatial axis) of m ($i \in 1, 2, 3$), and $m_i(n)$ its n -th element. The missing part of m is recovered using the following equations:

$$\beta_i = \operatorname{argmin}_{b_i} \left(\sum_n (X(n) \cdot b_i - m_i(n))^2 \right)_{i=1,2,3, n \in \{1, \dots, N\} \setminus \{n_1+1, \dots, n_2-1\}} \quad (2)$$

$$\tilde{m} = [X \cdot \beta_i]_{1 \leq i \leq 3} \quad (3)$$

where β_i is the vector of regression coefficients of m_i extracted through least square error minimization, and \tilde{m} is the trajectory recovered with global linear regression. In practice, computation of \tilde{m} is needed only for the missing frames.

Model 2: Local interpolation

To simplify the modeling of the incomplete marker trajectory, skeletal motion properties involving strong relations between markers can be considered. A local coordinate system can be defined based on three references, and hence reduce the variance of the trajectory representation [14].

Our second algorithm performs a local interpolation, i.e. an interpolation performed in a local reference defined by three other markers (the references).

Denote by p_1, p_2 , and p_3 the first three reference markers (ordered by σ_j , see Eq 1), used to recover m . Define a local coordinate system based on these markers, with three orthonormal vectors (u_1, u_2, u_3) at each time (or frame) n (here n indicates the row of a matrix ($1 \leq n \leq N$). For instance, $p_2(n)$ is the n^{th} row of p_2 , i.e. a 1×3 vector):

$$v_1(n) = p_2(n) - p_1(n), \quad u_1(n) = \frac{v_1(n)}{\|v_1(n)\|} \quad (4)$$

$$v_2(n) = v_1(n) \times (p_3(n) - p_1(n)), \quad u_2(n) = \frac{v_2(n)}{\|v_2(n)\|} \quad (5)$$

$$u_3(n) = u_1(n) \times u_2(n) \quad (6)$$

where \times indicates the cross product. m can then be projected into the local coordinate system:

$$P = [u_1(n)^T \ u_2(n)^T \ u_3(n)^T] \quad (7)$$

$$m_l(n) = (m(n) - p_1(n)) \cdot P \quad (8)$$

P is the projection matrix, and m_l the (projected) local trajectory.

m_l can be interpolated with simple linear interpolation. The recovered local trajectory can then be projected back into the original coordinate system:

$$\tilde{m}_l(n) = \begin{cases} m_l(n_1) + \frac{n - n_1}{n_2 - n_1} \cdot (m_l(n_2) - m_l(n_1)), & \text{for } n = \{n_1 + 1, \dots, n_2 - 1\} \\ m_l(n), & \text{otherwise} \end{cases} \quad (9)$$

$$\tilde{m}(n) = \tilde{m}_l(n) \cdot P^{-1} + p_1(n) \quad (10)$$

The interpolation is possible under the condition that all three references are present at frames n_1 and n_2 . Also, if there are missing frames in a reference during the gap ($n \in \{n_1 + 1, \dots, n_2 - 1\}$), the incomplete marker trajectory m will only be partially recovered. In this case, we can iterate the process with other references on the residual gap (i.e. p_1, p_2, p_4 if p_3 was missing during the gap of m to fill, and so on) until m is completely recovered.

Model 3: Local polynomial regression

As just discussed, local interpolation takes advantage of markers relations by performing an interpolation in a local coordinate system. To further use that advantage, we can model and predict the position of the missing marker from its neighborhood (the local reference markers), using regression. As the number of input variables is much lower than for global regression, we can use a more complex model, able to model the non-linear relations between marker trajectories.

Our third algorithm is based on polynomial regression in the local coordinate system. First, the trajectory to recover is projected into a local coordinate system (m_l) defined by reference markers p_1 , p_2 , and p_3 (see Eq 8). For each local coordinate of the marker to recover, a polynomial regression is performed, using reference markers local coordinates as input variables. In practice, only three input local coordinates are useful: the origin of the system is located in p_1 ($p_1 = (0, 0, 0)$), the new x-axis passes through p_2 , giving $p_2 = (x_{p_2}, 0, 0)$, and the new y-axis is normal to the plane passing through p_1 , p_2 , and p_3 , giving $p_3 = (x_{p_3}, 0, z_{p_3})$. Finally, the input set is composed of three variables:

$$X^l = \{x_{p_2}, x_{p_3}, z_{p_3}\} \quad (11)$$

For polynomial regression, the input variable set X^l is extended to quadratic polynomials in the input variables, leading to a set of 9 variables:

$$X^q = \{X_1^l, X_2^l, X_3^l, X_1^{l^2}, X_2^{l^2}, X_3^{l^2}, X_1^l \cdot X_2^l, X_1^l \cdot X_3^l, X_2^l \cdot X_3^l\} \quad (12)$$

The regression model is trained on the frames of the motion sequence where all markers (m_l , p_1 , p_2 , p_3) are present. The trained model is then used to predict all missing values of m_l :

$$\beta_i = \operatorname{argmin}_{\beta_i} \left(\sum_n (X^q(n) \cdot \beta_i - m_i(n))^2 \right)_{i=\{1,2,3\}, n \in \{1, \dots, N\} \setminus \{n_1+1, \dots, n_2-1\}} \quad (13)$$

$$\tilde{m}_l = [X^q \cdot \beta_i]_{1 \leq i \leq 3} \quad (14)$$

Finally, the recovered local trajectory \tilde{m}_l can be projected back into the original coordinate system (see Eq 10). Like local interpolation, this method is processed iteratively on sorted references until the trajectory of \tilde{m} is completely recovered.

Model 4: Local generalized regression neural network

Generalized Regression Neural Network (GRNN) is a non-linear regression method, already used in various applications [33]. It is a variant of an artificial neural network, consisting of four layers: the input layer, a radial basis layer, a summation layer and the output layer. GRNN allows to estimate any arbitrarily complex function, given a sufficient number of observations (generating the radial basis kernels). Comparatively to standard neural networks, GRNN does not require an iterative training. Moreover, as the output of the model is bounded by the extrema of the training dataset, a GRNN can only give physically meaningful outputs [34]. It means for instance that a GRNN should not estimate marker positions with highly implausible distances.

The proposed algorithm applies a GRNN on local variables X^l , to model and predict the local trajectory m_l . The GRNN is thus trained with three input variables (in practice, each input variable is standardized by subtracting its mean and dividing it by its standard deviation). (X^l , see Eq 11) and three output variables (\tilde{m}_l), according to:

$$\tilde{m}_l(n) = \frac{\sum_k m_l(k) \cdot \exp\left(-\frac{\|X^l(n) - X^l(k)\|^2}{2s^2}\right)}{\sum_k \exp\left(-\frac{\|X^l(n) - X^l(k)\|^2}{2s^2}\right)}_{k=\{1, \dots, N\} \setminus \{n_1+1, \dots, n_2-1\}} \quad (15)$$

Parameter s determines the smoothness of the regression, and was experimentally set to 0.3 in this research (for standardized input variables), as it gave the best results for the dataset tested. Intuitively, a larger s could be chosen to recover slow motions, and a smaller one for sharp and fast motions.

Like the other local recovery models, local GRNN process is iterated on sorted marker references until the trajectory of \tilde{m} is completely recovered.

All these individual models can be used independently to recover a trajectory, leading to several candidates. We explain in the next sections how these candidates are further processed and combined to produce a more robust recovery.

Time constraint: Trajectory continuity

Human motion time series are limited by two major constraints:

- a spacing constraint, defined by limited ranges of motion and fixed bone lengths;
- trajectory continuity, due to body inertia.

All recovery techniques that are based on interpolation intrinsically respect the continuity constraint. However, this is not the case of predictive models. To enforce continuity on recovered data, we can add a linear correction ramp:

$$\delta_{n_1} = \tilde{m}(n_1) - m(n_1) \quad (16)$$

$$\delta_{n_2} = \tilde{m}(n_2) - m(n_2) \quad (17)$$

$$\delta(n) = \begin{cases} \delta_{n_1} + \frac{n - n_1}{n_2 - n_1} \cdot (\delta_{n_2} - \delta_{n_1}), & \text{for } n = \{n_1 + 1, \dots, n_2 - 1\} \\ (0 \ 0 \ 0), & \text{otherwise} \end{cases} \quad (18)$$

$$\check{m} = \tilde{m} - \delta \quad (19)$$

For each axis, we compute the difference between the real value and the predicted value at each border (δ_{n_1} and δ_{n_2}), and subtract from the recovered trajectory a linear ramp from δ_{n_1} to δ_{n_2} . An example of this correction is illustrated in Fig 2.

Probabilistic Model Averaging (PMA)

Depending on the context, each model can be more or less effective, making difficult the choice of the best model, and the development of a robust recovery method. To address this issue, we propose a model averaging method, based on the posterior likelihoods of the distances between the recovered body points and other markers. This method is inspired from Bayesian model averaging [31].

We estimate the *a posteriori* probability of each predicted location according to their distance to reference markers. For references p_1 , p_2 , and p_3 , we estimate the distance distribution with m throughout the entire motion sequence on non-missing frames, using kernel smoothing density estimation (“kde” [35], used with Silverman’s rule of thumb to choose the bandwidth of the kernel estimator [36]). For each recovery method k , a weight is computed:

$$d_j = \|p_j - m\| \quad (20)$$

$$f_j(x) = \text{kde}_{d_j}(x) \quad (21)$$

$$\check{d}_{jk}(n) = \|p_j(n) - \check{m}_k(n)\| \quad (22)$$

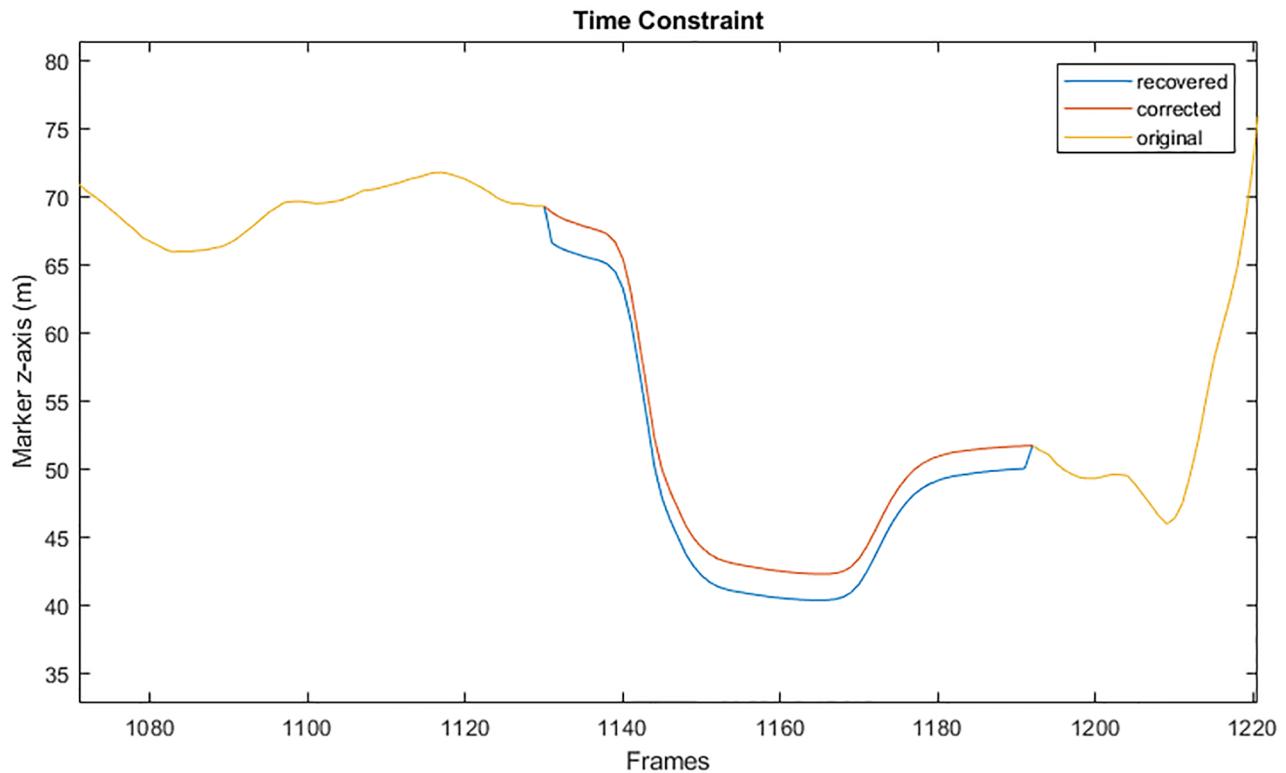


Fig 2. Trajectory continuity correction. The yellow curve shows incomplete data of a marker trajectory (m) on which a gap was introduced between frames 1130 and 1190 (only z-axis is shown). The blue curve represents the recovered data (\check{m}), and the red curve shows the corrected data using trajectory continuity constraint (\bar{m}) (see Eqs 16–19).

<https://doi.org/10.1371/journal.pone.0199744.g002>

$$\omega_k(n) = f_1(\check{d}_{1k}(n)) \cdot f_2(\check{d}_{2k}(n)) \cdot f_3(\check{d}_{3k}(n)) \quad (23)$$

Here ω_k is the weight of the trajectory \check{m}_k recovered with method k , and f_j is the estimated probability density function of the distance between m and the reference marker p_j .

We then compute a weighted average of all the recovered trajectories:

$$\bar{m}(n) = \frac{\sum_{k=1}^K \check{m}_k(n) \cdot \omega_k(n)}{\sum_{k=1}^K \omega_k(n)} \quad (24)$$

K is the number of individual models used for the recovery.

This process allows to give more importance to most likely recovered trajectories, according to their distance with other markers. In the remainder of the paper, we denote this method as *Probabilistic Model Averaging* (PMA).

Spacing constraint: Reference marker distance confidence interval

A final step is applied on the recovered trajectory. Knowing the probability density distribution of the distance $\|p_1 - m\|$, i.e. f_1 (see Eq 21), we can check if the distance of the recovered trajectory with p_1 respects the confidence interval:

$$CI = \{x : F_1(x) \in [0.05; 0.95]\}, \quad (25)$$

where F_1 is the cumulated probability density function estimation of the distance $\|p_1 - m\|$, i.e. $F_1(x) = \int_{-\infty}^x f_1(\xi) d\xi$. The limits of this interval correspond to two spheres centered on p_1 , with radii corresponding to:

$$r_1 = \arg_x(F_1(x) = 0.05) \quad (26)$$

$$R_1 = \arg_x(F_1(x) = 0.95) \quad (27)$$

If the recovered trajectory is outside these limits, it is projected onto the closest limit sphere. Fig 3 illustrates the projection of $\bar{m}(n)$ onto the limits of the confidence interval. In this example, the recovered frame $\bar{m}(n)$ is outside the confidence zone, as $x = \|p_1(n) - \bar{m}(n)\| > R_1$. The red arrow shows $\hat{m}(n)$, projection of the recovered frame $\bar{m}(n)$ onto the R_1 -radius sphere

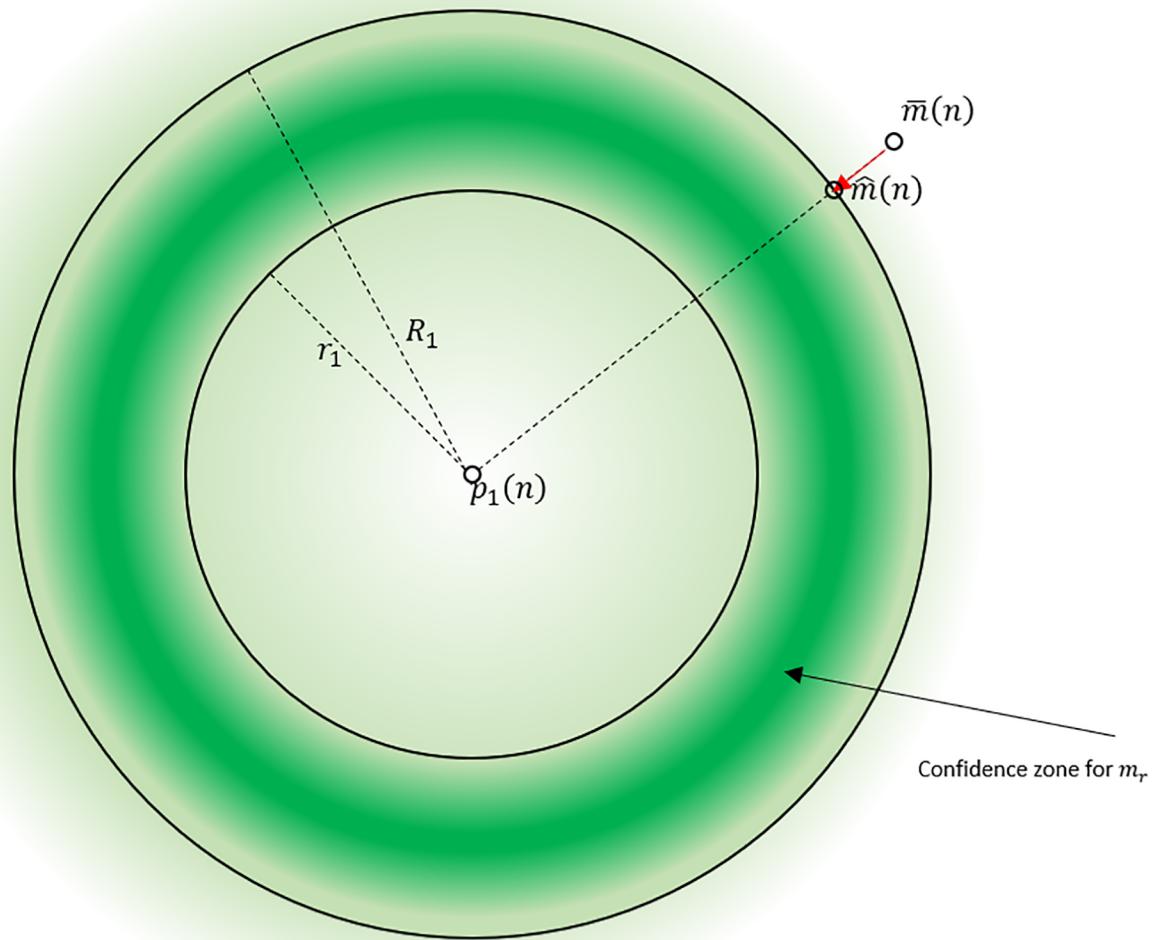


Fig 3. Reference distance soft constraints. The green intensity colormap indicates the probability of presence for the recovered frame. If the recovered frame $\bar{m}(n)$ is outside the confidence zone (delimited by spheres of radii r_1 and R_1), it is projected onto the closest point in this confidence zone ($\hat{m}(n)$).

<https://doi.org/10.1371/journal.pone.0199744.g003>

centered at p_1 , to fit the soft skeleton constraint. If the recovered frame is already in the confidence zone, no correction is applied: $\hat{m}(n) = \bar{m}(n)$. This process is thus used only if the recovered frame has low a posteriori probability, i.e. an unusual distance with its first reference marker p_1 .

A stricter version of the spacing constraint can be applied by recursively projecting the recovered point onto the CI obtained for several reference markers (e.g. the first three references p_1, p_2 and p_3), iteratively until the point is at a plausible distance of each reference.

Experiments

To validate our method, we tested each method used individually as well as the method combination with PMA. For such testing, we used the online CMU MoCap database (<http://mocap.cs.cmu.edu/>) [37] and the HDM05 database (<http://resources.mpi-inf.mpg.de/HDM05/>) [38]. They contain a high number of various motion sequences, and they have been used by much of the related work [12, 20, 22, 24, 25, 39]. Table 1 shows the motion sequences selected for the methods comparison. Motion sequences were selected to include a large variety of motions, in terms of complexity, type of motion, and duration.

The performance of the recovery method on a motion sequence may depend on different factors, including:

- The number of incomplete and complete marker trajectories
- The length of the gaps
- The duration of the sequence
- The complexity or periodicity of the motion

To analyze the performance of each method according to these factors, we introduced three concomitant gaps into our motion sequences, at random locations (uniformly distributed random markers and frames). We applied each method on these incomplete motion sequences, and extracted the recovery error for each method:

$$\epsilon = \frac{1}{g} \sum_{j=1}^g \frac{1}{n_{2_j} - (n_{1_j} + 1)} \sum_{n=n_{1_j}}^{n_{2_j}} \|\hat{m}_j(n) - m_j(n)\| \quad (28)$$

g is the number of random gaps created, n_{1_j} and n_{2_j} delimit the location (in frames) of the randomly introduced gap j , and m_j and \hat{m}_j are respectively the original and the recovered trajectories. We iterated this process 20 times with different random gap locations, and a mean recovery error was extracted from all iterations to estimate the general performance of each method. To analyze the influence of the duration of the sequence, fragments with different duration were extracted from each motion file.

Our method performances were compared to related work available online, namely the BoLeRo algorithm from Li et al. [24] (Matlab code available for download at: <https://github.com/lileicc/dynammo>) and the weighted PCA-based reconstruction method from Gloersen et al. [20] (Matlab code available for download at: <https://doi.org/10.1371/journal.pone.0152616>). In the sequel, these methods will be identified with the following numbers and acronyms:

1. Global Linear Regression (GLR)
2. Local Interpolation (LI) [14]

3. Local Polynomial Regression (LPR)
4. Local GRNN (LGRNN)
5. weighted PCA-based method (PCA) [20]
6. BoLeRo algorithm with soft bone constraints (BoLeRo) [24]

We used the soft bone constraints version of the BoLeRo algorithm, with 16 hidden dimensions as proposed by Li et al. [24]. The PCA-based method was used with the parameters proposed by their authors, using the consecutive reconstruction strategy for multiple gaps [20].

All results were obtained in MATLAB R2017a on a computer with Intel Core i7-4712HQ 2.3 GHz and 16 GB RAM running Windows 10.

Results

In this section, we present the results of the recovery on different simulated incomplete motion sequences. We analyze the influence of gap length, motion sequence duration, the number of incomplete marker trajectories, and the influence of the type of motion.

Gap length

[Fig 4](#) shows the results for two different motion sequences (respectively CMU1 and CMU3). In both cases, BoLeRo gives higher errors than all other methods. Moreover, the processing time, due to the iterative optimization process of the method, is significantly higher than others. For instance, the process duration is above the minute for filling three gaps of 2 seconds in the file CMU3, against less than a second for all other individual methods. For these reasons, and for better graphics readability, BoLeRo is left out in the remaining of the results.

On the right graphs, we can see results for all individual methods, except BoLeRo. Concerning the MoCap sequence CMU1 (top graph), [Fig 4](#) shows a clear separation of each method accuracy, where LGRNN seems to reach the best accuracy (20.2 mm mean error for three random gaps of 5 seconds). Accuracy of all methods seems to decrease with gap size. Concerning the MoCap sequence CMU3 (bottom graph), GLR seems to give the best results, with a mean recovery error of 12.7 mm for three random gaps of 5 seconds.

[Fig 4](#) also shows results of model averaging of several methods (dashed lines). In general, each combination of individual methods (all but 5 (PCA), all but 2 (LI), and all methods) seems to lead to an error comparable to that of the best individual methods in general. Our PMA method thus seems robust to gap size.

Motion sequence duration

Except for our most basic method based on interpolation (LI), each individual method performance may depend on the motion sequence duration. Indeed, more frames in the sequence means more information (more possible data variation), and more samples for model training.

To illustrate the influence of sequence duration on performance of gap recovery methods, fragments with different durations were extracted from each motion file. [Fig 5](#) shows the mean recovery error for different sequence durations, for different motion sequences. We can see on each graph that all methods follow similar patterns, showing that their performance highly depends on the specific motion. Nonetheless, for most graphs (except for HDM5), the mean recovery error seems to be higher for sequence durations of 5 seconds, and decreases for a sequence duration of 10 seconds. Beyond that duration, the recovery is not much improved. Concerning individual methods, LGRNN seems to be more robust to sequence duration,

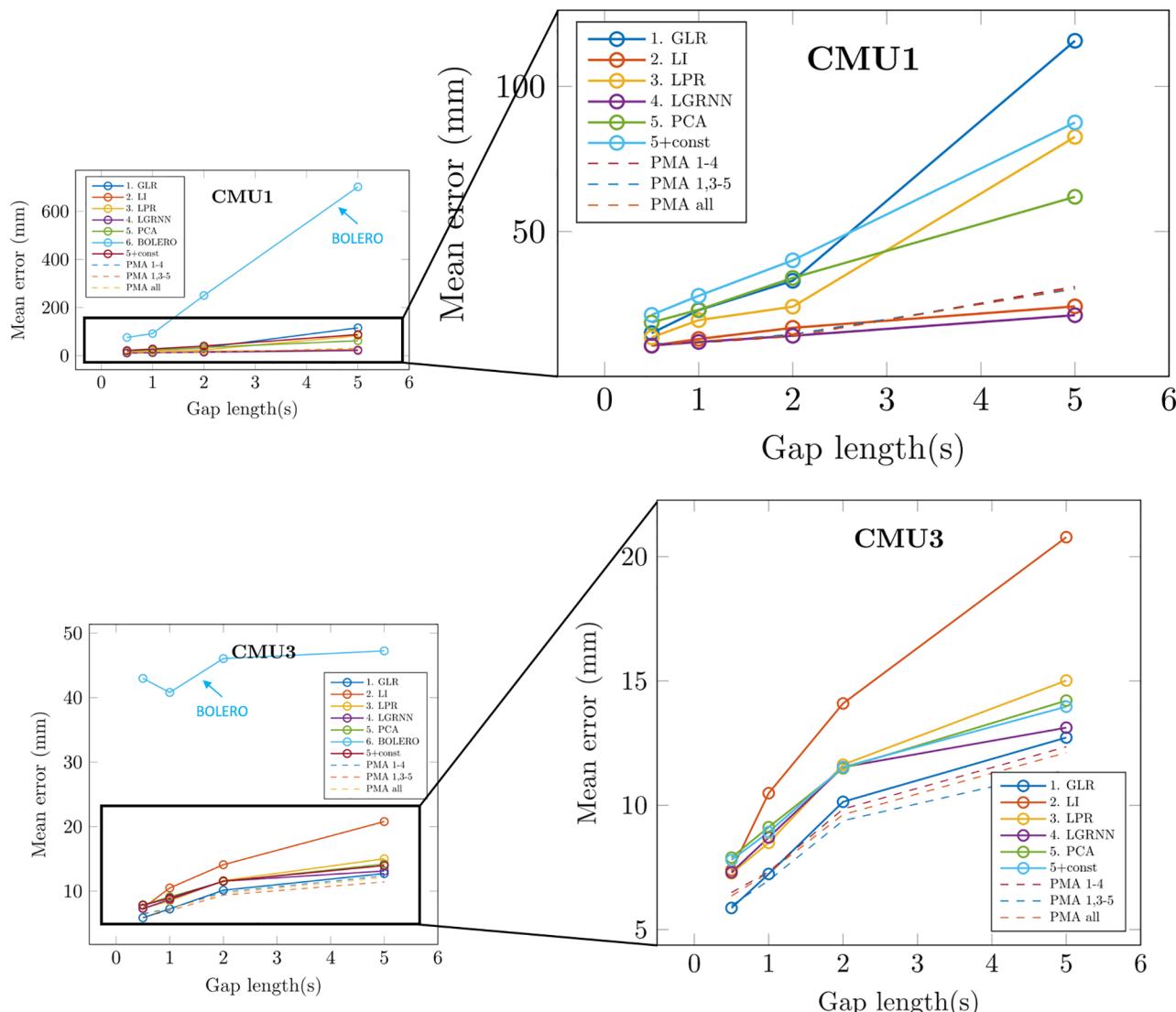


Fig 4. Mean recovery error for different gap sizes and gap recovery methods. Top: CMU1. Bottom: CMU3. Left: results including BoLeRo method. Right: results without BoLeRo method. Each point represents the mean of recovery errors, computed with 20 iterations, of three randomly created gaps of the same length (0.5, 1, 2 or 5 seconds). Solid lines show results for each individual method. Dashed lines show results for distance-probability averages of various combinations of individual methods.

<https://doi.org/10.1371/journal.pone.0199744.g004>

compared to other regression methods. For long durations, GLR seems to give the best results of all individual methods.

For all durations and all motion sequences, PMA effectively weights each individual method, hence providing optimal recovery in any context. The best combination is the averaging of all methods but LI (dark red dashed line). Our PMA method is thus robust to motion duration.

Number of concomitant gaps

Except for basic interpolation or dynamic filtering methods, the reconstruction quality of one marker trajectory depends on the presence of reference markers. If several markers are missing at the same time during the motion sequence, less information is available for reconstruction.

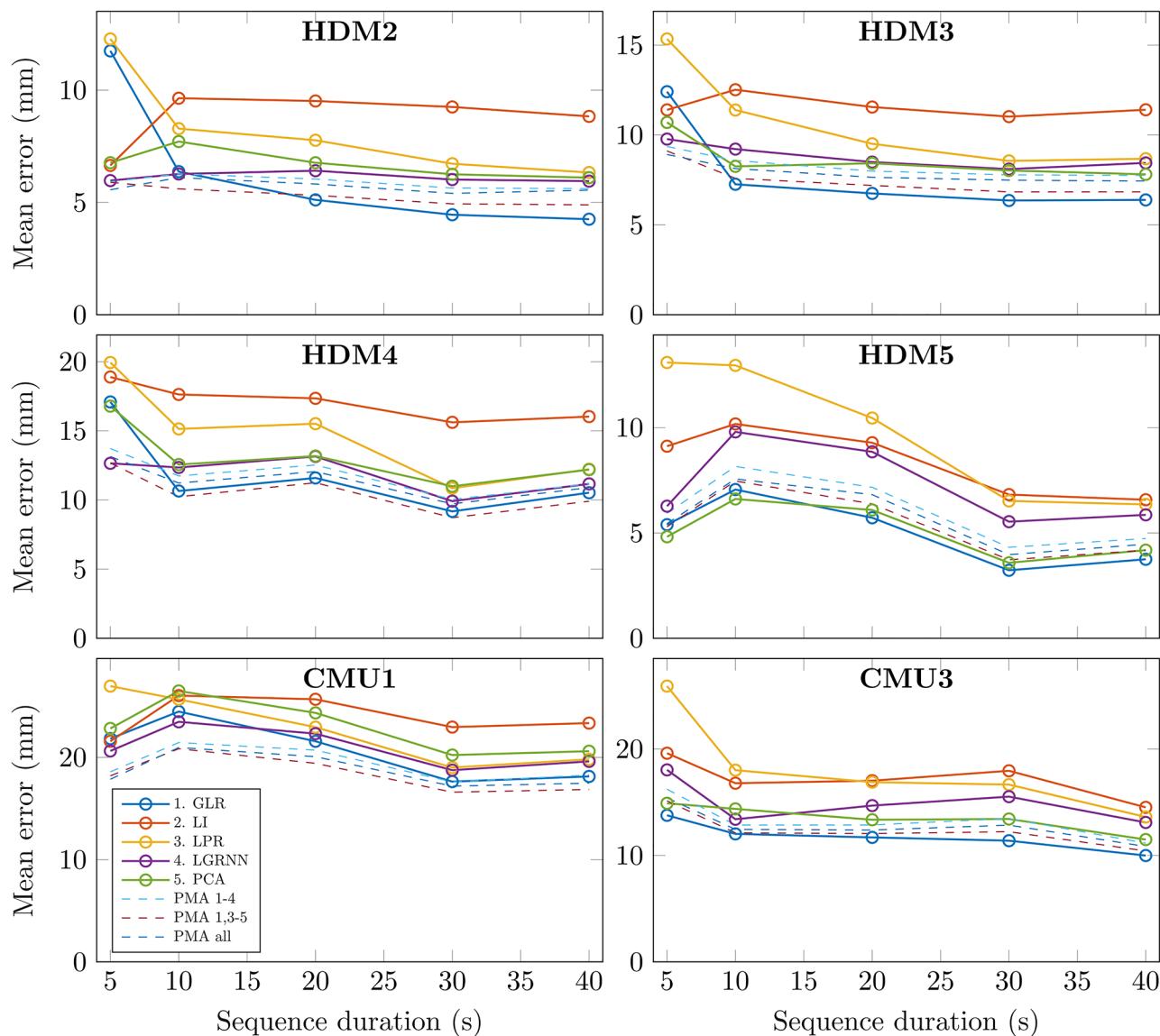


Fig 5. Mean recovery error for different sequence durations and gap recovery methods. To illustrate the influence of sequence duration on performance of gap recovery methods, fragments of different durations were extracted from each motion file. Each point represents the mean of the recovery errors computed on 20 iterations of three randomly created gaps of 1 second. Continuous lines show results for each individual method. Dashed lines show results for PMA with various methods combinations.

<https://doi.org/10.1371/journal.pone.0199744.g005>

According to the method, the quality of the reconstruction may be influenced differently. Fig 6 shows the mean recovery error of individual methods and their PMA combinations for different motion sequences, and for different numbers of markers missing at the same time (gaps of one second). We can see in general that for all motion sequences and for all methods, the recovery error grows with the number of concomitant gaps. Again, PMA generally give the best results.

Constraints effect

For all previous results, time and spacing constraints were applied for all individual methods and model averages. To verify the effectiveness of these constraints, PMA reconstruction was

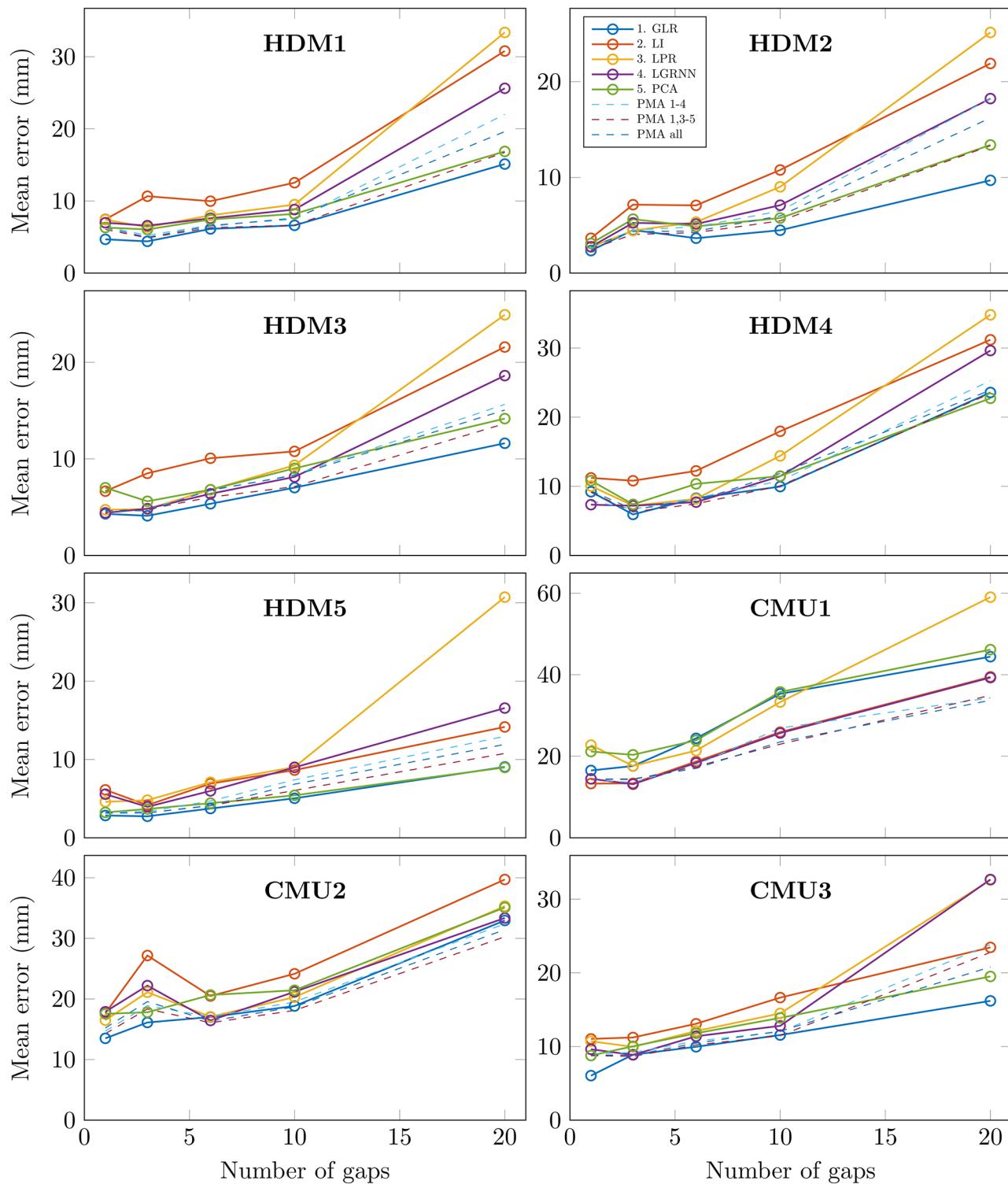


Fig 6. Mean recovery error for different numbers of missing markers and gap recovery methods. Each point represents the mean of recovery errors computed over 20 iterations of a number of randomly created gaps of 1 second (1, 3, 6, 10 or 20 gaps). Solid lines show results for each individual method. Dashed lines show results for distance-probability averages of various methods combinations.

<https://doi.org/10.1371/journal.pone.0199744.g006>

Table 2. Effect of constraints on mean recovery error (t-test, n = 200; conditions: 3 gaps of 1 seconds).

Sequence	$\bar{\epsilon}$, no constraint (mm)	$\bar{\epsilon}$, constraint (mm)	Difference (mm)	t-test p-value
HDM1	8.1	6.6	1.5	$p < 10e - 5$
HDM2	4.8	4.4	0.4	$p = 0.46$
HDM3	5.5	5.2	0.3	$p = 0.07$
HDM4	8.5	7.4	1.2	$p = 0.08$
HDM5	4.5	3.0	1.5	$p = 0.001$
CMU1	17.1	15.7	1.5	$p = 0.008$
CMU2	14.4	13.5	1.0	$p < 10e - 3$
CMU3	10.5	8.4	2.1	$p < 10e - 10$

Paired t-test (n = 200) on constraints effect on PMA for the reconstruction of 3 gaps of 1 second, introduced into different motion sequences. Individual methods 1 to 4 were used in this test.

<https://doi.org/10.1371/journal.pone.0199744.t002>

tested with and without constraints for each motion sequence, with 200 iterations of three gaps of one second. For each motion sequence, a paired t-test was performed on the mean recovery error of the 200 iterations with and without constraint, as shown in [Table 2](#).

Results show that for almost all tested motion sequences, PMA yields a significant improvement of the recovery method. The constraints did not improve the recovery for the sequence HDM2 (larger p-value), but this may be due to the fact that the recovery error is already low without constraint ($\bar{\epsilon} = 4.8\text{mm}$).

[Table 3](#) shows a similar analysis in a situation of low marker presence. In this case, 10 simultaneous gaps of 5 seconds were introduced into each motion sequence. We can see that in such situation, PMA's mean recovery error is much higher, and constraints always improve it significantly, up to 40mm for CMU1.

Synthesis—Mean results

As a synthesis, [Fig 7](#) shows the mean results of each method, obtained from the mean of the recovery errors on all the selected motion sequences. It can be seen that the various PMA combinations give more robust reconstruction regardless of the type of motion, the gap length (left graph), the duration of the motion sequence (center graph) and the number of incomplete marker trajectories (right graph). Among the individual methods, there is no clear difference of performance according to gap length. The local GRNN method seems more robust to gap

Table 3. Effect of constraints on the mean recovery error (t-test, n = 200; conditions: 10 gaps of 5 seconds).

Sequence	$\bar{\epsilon}$, no constraint (mm)	$\bar{\epsilon}$, constraint (mm)	Difference (mm)	t-test p-value
HDM1	40.9	13.5	27.4	$p < 10e - 10$
HDM2	14.4	8.4	6.0	$p < 10e - 3$
HDM3	32.2	10.8	21.4	$p < 10e - 10$
HDM4	29.8	16.1	13.7	$p < 10e - 8$
HDM5	25.0	8.6	16.4	$p < 10e - 10$
CMU1	80.8	39.1	41.7	$p < 10e - 10$
CMU2	27.7	26.2	1.5	$p = 0.16$
CMU3	22.1	19.6	2.6	$p = 0.02$

Paired t-test (n = 200) on constraints effect on PMA for the reconstruction of 10 simultaneous gaps of 5 seconds, introduced into different motion sequences. Individual methods 1 to 4 were used in this test.

<https://doi.org/10.1371/journal.pone.0199744.t003>

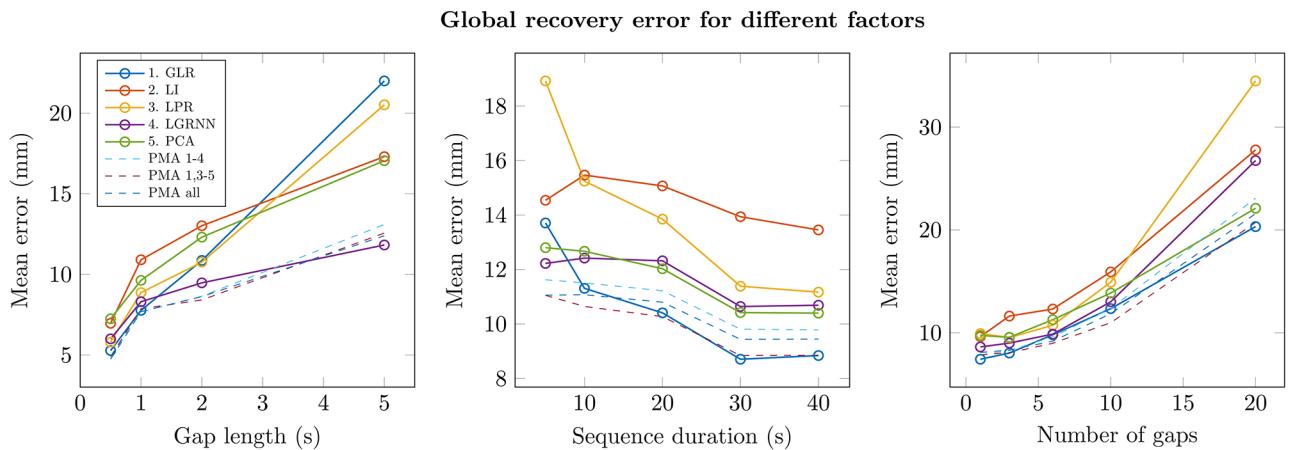


Fig 7. Mean recovery error for different recovery methods, for all test motion sequences. Left: different gap lengths (3 concomitant gaps, total sequence duration); Center: different motion durations (3 concomitant gaps of 1 second); Right: different numbers of concomitant gaps (gaps of 1 second, total sequence duration). Each point represents the mean of recovery errors computed over 20 iterations of a number of randomly created gaps. Solid lines show results for each individual method. Dashed lines show results for PMA with various individual methods combinations.

<https://doi.org/10.1371/journal.pone.0199744.g007>

length: it allows to recover three concomitant gaps of 5 seconds with a mean error of 10 mm. All other methods lead to a mean error above 15 mm. The local GRNN seems to be more robust to sequence duration. A duration of 5 seconds (with 41 markers, 120 fps) allows to train an effective model to reconstruct three concomitant gaps of 1 second with a mean error of 12 mm. However, for a longer sequence (40 seconds), GLR gives the best results, with a mean error of 9 mm. All individual methods are highly sensitive to the number of concomitant gaps, and thereby to the number of markers available to predict the missing trajectories. Finally, PMA systematically improves the gap recovery, independently of motion type, gap length, sequence duration or number of missing markers. As the local interpolation method seems to be the less effective, the best method combination is the averaging of methods 1, 3, 4 and 5.

Discussion

Our PMA method presents several advantages compared to the available state of the art. It is fully automatic and does not require any prior knowledge or any pre-trained model. It can be used on MoCap data recorded with any marker set. Graphical results show that PMA is robust to various factors, including gap length, sequence duration, the number of simultaneous gaps, and the type of motion. Additionally, the use of temporal and spacing constraints significantly improves the reconstruction, especially in challenging conditions (see Table 3).

Figs 4–7 show results where spacing and time constraints were applied to each individual method. These constraints may indeed be applied afterwards to any prediction method. The improvement of recovery after method combination is hence exclusively due to PMA, and confirms its effectiveness. In terms of quantitative results, no individual method shows better performance in general. All individual methods are more or less sensitive to the context, including motion type, gap length, sequence duration or the number of simultaneous missing markers. In contrast, PMA seems to take advantage of every individual method, improving the robustness of the recovery algorithm. Moreover, as averaging is based on distance with reference markers, PMA partly takes skeleton constraints into account.

Our methods could not be compared to some recently proposed methods, due to the unavailability of the code. However, the recent skeleton-constrained SVT method from Tan et al. (2015) [25] (not included in our study), based on both skeleton constraints and low rank

properties, achieved similar results to BoLeRo [24]: their improvement over BoLeRo mainly lies in execution time, as explained in [25]. Their constraint-fitting optimization method converges significantly faster than BoLeRo.

The effectiveness of the method proposed in the present study, including PMA and time and spacing constraints, is independent of the individual recovery model. It can theoretically be applied to any other set of individual recovery models in the future, possibly leading to better performance.

Limitations

The methods included in the present study rely on several parameters, including the threshold for reference marker selection in GLR, the smoothing parameter in LGRNN, parameters from w-PCA [20], as well as confidence interval thresholds for spacing constraints. All these parameters were experimentally chosen in this research, as they gave the best results for the dataset tested (see Table 1). The user should adapt these parameters for her/his own data if necessary. Optimal parameters could depend on MoCap data, including for instance the number of recorded markers, their particular placement, the complexity or speed of the motion, data accuracy, or noise due to marker vibrations or camera quality. For instance, a larger threshold for the linear regression model could be considered for MoCap data with fewer available markers, and the optimal smoothness parameter for LGRNN could depend on the smoothness of the motion itself.

For the validation of the proposed methods, gaps were introduced into motion sequences at random locations. It is possible that in some particular cases, a marker can be isolated, without any highly related reference. If this marker is missing, it could lead to a poor recovery. This issue thus depends on the placement of markers. It is hence relevant to consider this aspect when defining marker placement, to avoid isolated markers. On the other hand, markers placed too close to each other risk to be occluded simultaneously. A trade-off must thus be considered for their placement.

PMA has some limitations in comparison to a method such as BoLeRo [24]. In case of a blackout, i.e. when all markers disappear at the same time, a method based on a predictive filter such as Kalman filter can reconstruct an entire frame, and then use gradient descent or a similar optimization method to fit skeleton constraints, whereas PMA needs at least three present markers as references to evaluate distance probabilities. However, this is an extreme case, which can generally be avoided with an efficient use of the MoCap system.

Improvement prospects

Our methods could be improved in various ways. First, human motion is not a stationary process [40]. Each individual model might be made more efficient by giving more importance to motion data that are close to the gap to reconstruct. For instance, for each gap, a local model could be trained on a limited time window centered on that gap. The distance probabilities could also be locally defined on a time window. However, this would limit the number of available data for model training.

Secondly, we could use a more complex constraint fitting method, making use of a dynamic model such as the Kalman filter [41] to ensure trajectory continuity. Additionally, an optimization procedure could be used instead of projection for skeleton constraints fitting whenever the inter-marker distance is outside the confidence interval. However, this could significantly increase execution time.

Finally, an original interest in the distance variation density estimation is the possibility to assess the quality of the reconstruction. It could further be used as an indication to identify

and verify the most sensitive parts of the data, and possibly reject them and reprocess them with another configuration or method.

Processing time consideration

It is interesting to note that in human motion data, adjacent frames are very similar if the frame rate is high enough. In this case, data can be easily subsampled without losing much information for model training. This subsampling can drastically decrease computation time, either for computing reference weights ([Eq 1](#)), for individual model training, as well as for kernel smoothing density estimation ([Eq 21](#)).

Though it is not the initial goal of the proposed algorithm, each individual method based on regression, as well as their combination with PMA could be adapted for real-time purpose. Each individual model and distance distribution estimation can be trained on previously recorded data, and can be effective after a few seconds of recording. In this case, the time constraint would be limited to information about previous data. A Kalman filter would be appropriate for this task.

Conclusion

We have proposed an original automatic method, Probabilistic Model Averaging (PMA), for robust reconstruction of missing MoCap data. The robustness of our method relies on two major steps:

1. The weighted combination of several models, based on the posterior likelihoods of inter-marker distances.
2. The application of simple but effective constraints, enforcing trajectory continuity and plausible distance of reconstructed trajectories with related markers.

To support and validate our model-averaging method, several reconstruction methods based on regression and local coordinates were proposed, and were found to compete with state-of-the-art methods. Results show that PMA used with the constraints outperforms individual methods in various conditions, including various gap lengths, motion sequence durations and numbers of simultaneous gaps.

Our method has the advantage of being fully automatic. The algorithm is data-driven, and does not need any prior knowledge or any pre-trained model. Moreover, the model averaging and the proposed constraints are general and can be used with any other individual reconstruction method, leading to possible future improvement.

Supporting information

S1 Code. Matlab code. Matlab script containing a Matlab implementation of the proposed method. Note that the maintained version of the code can be found on the Github repository: <https://github.com/numediar/MocapRecovery>.

(M)

Author Contributions

Conceptualization: Mickaël Tits.

Data curation: Mickaël Tits.

Formal analysis: Mickaël Tits.

Investigation: Mickaël Tits.

Methodology: Mickaël Tits.

Project administration: Joëlle Tilmanne, Thierry Dutoit.

Resources: Mickaël Tits.

Software: Mickaël Tits.

Supervision: Joëlle Tilmanne, Thierry Dutoit.

Validation: Mickaël Tits.

Visualization: Mickaël Tits.

Writing – original draft: Mickaël Tits.

Writing – review & editing: Mickaël Tits, Joëlle Tilmanne.

References

1. Moeslund TB, Granum E. A survey of computer vision-based human motion capture. *Comput Vis Image Underst*. 2001; 81(3):231–268. <https://doi.org/10.1006/cviu.2000.0897>
2. Moeslund TB, Hilton A, Krüger V. A survey of advances in vision-based human motion capture and analysis. *Comput Vis Image Underst*. 2006; 104(2):90–126. <https://doi.org/10.1016/j.cviu.2006.08.002>
3. Multon F, France L, Cani-Gascuel MP, Debuinne G. Computer animation of human walking: a survey. *Comput Animat Virtual Worlds*. 1999; 10(1):39–54.
4. Jaimes A, Sebe N. Multimodal human–computer interaction: A survey. *Comput Vis Image Underst*. 2007; 108(1):116–134. <https://doi.org/10.1016/j.cviu.2006.10.019>
5. Rego P, Moreira PM, Reis LP. Serious games for rehabilitation: A survey and a classification towards a taxonomy. *Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on*. IEEE; 2010. p. 1–6.
6. Zhou H, Hu H. Human motion tracking for rehabilitation—A survey. *Biomed Signal Process Control*. 2008; 3(1):1–18. <https://doi.org/10.1016/j.bspc.2007.09.001>
7. Motion Capture Systems—VICON;. <https://www.vicon.com/>.
8. Zhang Z. Microsoft kinect sensor and its effect. *IEEE multimedia*. 2012; 19(2):4–10. <https://doi.org/10.1109/MMUL.2012.24>
9. Pfister A, West AM, Bronner S, Noah JA. Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis. *J Med Eng Technol*. 2014; 38(5):274–280. <https://doi.org/10.3109/03091902.2014.909540> PMID: 24878252
10. Ceseracciu E, Sawacha Z, Cobelli C. Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept. *PloS One*. 2014; 9(3): e87640. <https://doi.org/10.1371/journal.pone.0087640> PMID: 24595273
11. Mündermann L, Corazza S, Andriacchi TP. The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications. *J Neuroeng Rehabil*. 2006; 3(1):6. <https://doi.org/10.1186/1743-0003-3-6> PMID: 16539701
12. Liu G, McMillan L. Estimation of missing markers in human motion capture. *Vis Comput*. 2006; 22(9):721–728. <https://doi.org/10.1007/s00371-006-0080-9>
13. Aristidou A, Cameron J, Lasenby J. Predicting missing markers to drive real-time centre of rotation estimation. *Articulated Motion and Deformable Objects*. 2008; p. 238–247. https://doi.org/10.1007/978-3-540-70517-8_23
14. Howarth SJ, Callaghan JP. Quantitative assessment of the accuracy for three interpolation techniques in kinematic analysis of human movement. *Comput Methods Biomech Biomed Engin*. 2010; 13(6):847–855. <https://doi.org/10.1080/10255841003664701> PMID: 21153975
15. Fritsch FN, Carlson RE. Monotone piecewise cubic interpolation. *SIAM J Numer Anal*. 1980; 17(2):238–246. <https://doi.org/10.1137/0717021>
16. Li L, McCann J, Pollard NS, Faloutsos C. Dynammo: Mining and summarization of coevolving sequences with missing values. *KDD. ACM*; 2009. p. 507–516.

17. Wang JM, Fleet DJ, Hertzmann A. Gaussian process dynamical models for human motion. *IEEE Trans Pattern Anal Mach Intell.* 2008; 30(2):283–298. <https://doi.org/10.1109/TPAMI.2007.1167> PMID: 18084059
18. Chai J, Hodgins JK. Performance animation from low-dimensional control signals. *ACM Trans Graph.* vol. 24. ACM; 2005. p. 686–696.
19. Federolf PA. A novel approach to solve the “missing marker problem” in marker-based motion analysis that exploits the segment coordination patterns in multi-limb motion data. *PloS One.* 2013; 8(10): e78689. <https://doi.org/10.1371/journal.pone.0078689> PMID: 24205295
20. Gløersen Ø, Federolf P. Predicting Missing Marker Trajectories in Human Motion Data Using Marker Intercorrelations. *PloS One.* 2016; 11(3):e0152616. <https://doi.org/10.1371/journal.pone.0152616> PMID: 27031243
21. Lai RY, Yuen PC, Lee KK. Motion capture data completion and denoising by singular value thresholding. *Comput Graph Forum.* 2011;
22. Peng SJ, He GF, Liu X, Wang HZ. Hierarchical block-based incomplete human mocap data recovery using adaptive nonnegative matrix factorization. *Comput Graph.* 2015; 49:10–23. <https://doi.org/10.1016/j.cag.2015.04.004>
23. Feng Y, Xiao J, Zhuang Y, Yang X, Zhang JJ, Song R. Exploiting temporal stability and low-rank structure for motion capture data refinement. *Inf Sci (Ny).* 2014; 277:777–793. <https://doi.org/10.1016/j.ins.2014.03.013>
24. Li L, McCann J, Pollard N, Faloutsos C. Bolero: a principled technique for including bone length constraints in motion capture occlusion filling. *Comput Animat;* 2010. p. 179–188.
25. Tan CH, Hou J, Chau LP. Motion capture data recovery using skeleton constrained singular value thresholding. *Vis Comput.* 2015; 31(11):1521–1532. <https://doi.org/10.1007/s00371-014-1031-5>
26. Kirk AG, O'Brien JF, Forsyth DA. Skeletal parameter estimation from optical motion capture data. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit.* vol. 2. IEEE; 2005. p. 782–788.
27. De Aguiar E, Theobalt C, Seidel HP. Automatic learning of articulated skeletons from 3D marker trajectories. *Adv Vis Comput.* Springer; 2006. p. 485–494.
28. Seber GA, Lee AJ. Linear regression analysis. John Wiley & Sons; 2003.
29. Bates DM, Watts DG. Nonlinear regression analysis and its applications. vol. 2. Wiley Online Library;;
30. Dietterich TG. Ensemble methods in machine learning. International workshop on multiple classifier systems. Springer; 2000. p. 1–15.
31. Hoeting JA, Madigan D, Raftery AE, Volinsky CT. Bayesian model averaging: a tutorial. *Statistical science.* 1999; p. 382–401.
32. Burger B, Toiviainen P. MoCap Toolbox-A Matlab toolbox for computational analysis of movement data. 10th Sound and Music Computing Conference, SMC 2013, Stockholm, Sweden. Logos Verlag Berlin; 2013.
33. Specht DF. A general regression neural network. *IEEE Trans Neural Netw.* 1991; 2(6):568–576. <https://doi.org/10.1109/72.97934> PMID: 18282872
34. Cigizoglu HK, Alp M. Generalized regression neural network in modelling river sediment yield. *Adv Eng Softw.* 2006; 37(2):63–68. <https://doi.org/10.1016/j.advengsoft.2005.05.002>
35. Parzen E. On estimation of a probability density function and mode. *Annals of mathematical statistics.* 1962; 33(3):1065–1076. <https://doi.org/10.1214/aoms/1177704472>
36. Silverman BW. Density estimation for statistics and data analysis. vol. 26. CRC press; 1986.
37. Hodgins J. CMU graphics lab motion capture database;. <http://mocap.cs.cmu.edu/>.
38. Müller M, Röder T, Clausen M, Eberhardt B, Krüger B, Weber A. Documentation Mocap Database HDM05. Universität Bonn; 2007. CG-2007-2.
39. Tan CH, Hou J, Chau LP. Human motion capture data recovery using trajectory-based matrix completion. *Electron Lett.* 2013; 49(12):752–754. <https://doi.org/10.1049/el.2013.0442>
40. Chiari L, Della Croce U, Leardini A, Cappozzo A. Human movement analysis using stereophotogrammetry: Part 2: Instrumental errors. *Gait Posture.* 2005; 21(2):197–211. <https://doi.org/10.1016/j.gaitpost.2004.04.004> PMID: 15639399
41. Kalman RE. A new approach to linear filtering and prediction problems. *Journal of basic Engineering.* 1960; 82(1):35–45. <https://doi.org/10.1115/1.3662552>