A Locality-based Neural Solver for Optical Motion Capture

XIAOYU PAN and BOWEN ZHENG, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab, China
XINWEI JIANG, GUANGLONG XU, XIANLI GU, and JINGXIANG LI, Tencent Games Digital Content Technology Center, China
QILONG KOU, Tencent Technology (Shenzhen) Co., LTD, China
HE WANG, University College London (UCL), United Kingdom
TIANJIA SHAO and KUN ZHOU, State Key Lab of CAD&CG, Zhejiang University, China
XIAOGANG JIN*, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab, China

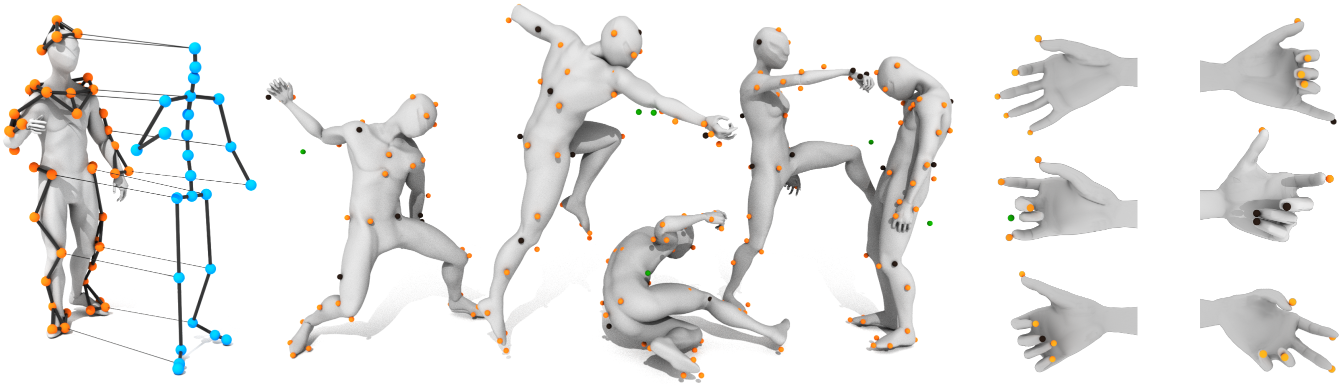

Fig. 1. Given raw MoCap data, our method constructs a heterogeneous graph neural network (left) to solve large body motions (middle) and fine multi-fingered hand motions (right). Our method is robust to occlusions (black balls) and outliers (green balls).

We present a novel locality-based learning method for cleaning and solving optical motion capture data. Given noisy marker data, we propose a new heterogeneous graph neural network which treats markers and joints as different types of nodes, and uses graph convolution operations to extract the local features of markers and joints and transform them to clean motions. To deal with anomaly markers (e.g. occluded or with big tracking errors), the key insight is that a marker's motion shows strong correlations with the motions of its immediate neighboring markers but less so with other markers, a.k.a. *locality*, which enables us to efficiently fill missing markers (e.g. due to occlusion). Additionally, we also identify marker outliers due to tracking errors by investigating their acceleration profiles. Finally, we propose a training regime based on representation learning and data augmentation, by training the model on data with masking. The masking schemes aim to mimic the occluded and noisy markers often observed in the real data. Finally, we show that our method achieves high accuracy on multiple metrics across various datasets. Extensive comparison shows our method outperforms state-of-the-art methods in terms of prediction accuracy of occluded marker position error by approximately 20%, which leads to a further error reduction on the reconstructed joint rotations and positions by 30%. The code and data for this paper are available at github.com/localmocap/LocalMoCap.

CCS Concepts: • **Computing methodologies** → **Motion capture**; **Neural networks**.

Additional Key Words and Phrases: Motion Capture, Motion Capture Marker Cleaning, Motion Capture Solving, Machine Learning

ACM Reference Format:
Xiaoyu Pan, Bowen Zheng, Xinwei Jiang, Guanglong Xu, Xianli Gu, Jingxiang Li, Qilong Kou, He Wang, Tianjia Shao, Kun Zhou, and Xiaogang Jin. 2023. A Locality-based Neural Solver for Optical Motion Capture. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH ASIA'23 Conference Proceedings), December 12–15, 2023, Sydney, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/xxx.xxxx

1 INTRODUCTION

Motion capture (MoCap) is a popular industry solution for giving virtual characters realistic motions. It captures the motions of real-world actors using a variety of sensors. Because of its precision and flexibility, optical motion capture is still the industry standard for film and video game production to capture body and hand motions with highly coordinated movement. Even with high-fidelity and

[*] Corresponding author.

SIGGRAPH ASIA'23 Conference Proceedings, December 12–15, 2023, Sydney, Australia
© 2023 Association for Computing Machinery.
ACM ISBN xxx.xxxx…$15.00
https://doi.org/xxx.xxxx

expensive motion capture equipment, raw data is inevitably contaminated by occlusions, position errors, and mislabelling. Despite the development of tools for cleaning and solving MoCap data, greater accuracy still requires manual fixing and adjustment. When it comes to complex motions like hand motions, even experienced animators find this process time-consuming.

Numerous efforts in the past have been made to automatically clean MoCap data and solve for motions. Early methods [Aristidou and Lasenby 2013; Burke and Lasenby 2016; Feng et al. 2014; Herda et al. 2000; Kirk et al. 2004; Li et al. 2009, 2010; Liu et al. 2014; Zordan and Van Der Horst 2003] heavily rely on rules abstracted from empirical observations and hand-crafted features. These methods can produce satisfactory results with specific patterns and noises via careful hand-tuning, but they constantly suffer from poor generalization onto real-world data, which contains complex noises, e.g. simultaneous occlusions of multiple markers, and varying duration of occlusions. Recently, data-driven methods [Chen et al. 2021; Ghorbani and Black 2021; Holden 2018; Pavllo et al. 2018; Perepichka et al. 2019; Tits et al. 2018; Wang et al. 2016; Xiao et al. 2015] have been employed to address the aforementioned limitations. For instance, MoCap-Solver [Chen et al. 2021] solves motions and reconstructs clean markers by separately encoding motion, marker configuration, and skeleton using different neural networks, and achieves the state-of-the-art results. However, dealing with noisy marker data with outliers and large simultaneous occlusions remains difficult. This is especially true for hand markers, which often have subtle motions involving frequent marker occlusions.

Overall, we observe that the state-of-the-art data-driven methods [Chen et al. 2021; Holden 2018] have three main limitations. First, they solve for motion and reconstruct clean markers using skinning functions, which ignore the complexity of marker motions (e.g., sliding across the body surface) and may introduce additional errors due to motion solving errors. Second, they often ignore the fine-grained correlations between markers, e.g. using one single fully connected network structure to encoder all markers in the same way, resulting in inaccurate solving results for certain large body motions and fine multi-fingered hand movements (see Figs. 9 and 10). Third, all methods have to assume/model data noise, where a common strategy is to use random sampling per frame. However, modeling noise data with per-frame random sampling does not account for long gaps caused by complex and occlusion-intensive movements and varying occlusion probability of different markers, potentially lowering solve accuracy on real data.

Three major challenges must be overcome in order to accurately solve for motions. (1) The ability to deal with complex occlusions that include simultaneous occlusions and occlusions of varying duration. Although neural-based methods can be used to learn from real-world data, using networks alone makes it difficult to learn complex patterns of marker motions, limiting its use for efficiently filling occluded marker positions. (2) Modeling the relationships between markers in a neural network. A critical task in quantifying such relationships is to establish links between markers with strong correlations while avoiding links to other markers. Networks with fully connected structures do not explicitly model such relationships as they encode all markers in the same way. (3) Creating realistic large-scale synthetic MoCap data with long occlusion gaps. Due to the high cost of MoCap systems, obtaining real noise data is difficult, necessitating data augmentation to generate synthetic noises. Actual MoCap sequences frequently have occlusion gaps ranging from 40 to 100 frames. The per-frame random sample method fails to generate occlusion gaps longer than eight frames (see Fig. 4), resulting in a mismatch between the synthetic and real-data distributions.

We present a data-driven method for cleaning optical MoCap data and solving for body and hand motions. A key aspect of our approach is leveraging marker locality by identifying neighbor markers with stable distances between them across the entire motion. Using this locality as a prior, we can obtain an initial estimate of occluded markers' positions using the distance matrix between neighbor markers via Euclidean distance matrix optimization, which reduces the difficulty of network learning intricate marker motions and significantly improves accuracy. To quantify the relationships between markers, we connect markers and related joints and construct edges between neighbor markers and parent joints to form a heterogeneous graph. We perform convolution operations to extract local features of markers and joints, which enables the network to accurately solve motions. In addition, we augment the dataset with data distributions similar to real data by sampling occlusion gaps of varying lengths for different markers based on actual occlusion probabilities and occlusion gap distributions. In summary, our paper makes the following three contributions:

- A robust MoCap data occlusion filling method that integrates locality and learned priors to handle simultaneous occlusions and occlusions of varying durations.
- A heterogeneous graph neural network that solves motions by explicitly modeling the relationships between neighbor markers, allowing it to solve accurate large body motions as well as fine multi-fingered hand motions.
- A novel method for augmenting motion capture datasets that takes into account the statistical features of marker occlusion and generates data with a distribution similar to the real data.

2 RELATED WORK

Human pose estimation is an important and ongoing research topic in computer vision and computer graphics. While marker-free, image-based solutions have yielded promising results, marker-based motion capture remains popular due to its accuracy and flexibility [Taheri et al. 2020]. MoCap systems' high-fidelity body motion data [CMU 2000; Mahmood et al. 2019; Sigal et al. 2010; Trumble et al. 2017] is useful in a variety of applications [Zheng et al. 2020], including action recognition [Hua et al. 2023; Yan et al. 2018], action prediction [Cao et al. 2020; Cui et al. 2021], motion synthesis [Chen et al. 2023; Tevet et al. 2022] and image-based pose estimation [Munea et al. 2020; Varol et al. 2017]. However, raw MoCap data is inevitably contaminated with errors.

To solve motions from imperfect MoCap data, various methods for cleaning noises and solving motions have been proposed, which can be broadly classified into hand-crafted prior-based methods and data-driven models. The former are based on hard-coded empirical rules such as spatiotemporal continuity and bone-length consistency, which are implemented using skeleton templates [Herda et al. 2000; Kirk et al. 2004; Zordan and Van Der Horst 2003], Kalman filters [Aristidou and Lasenby 2013; Burke and Lasenby 2016; Li et al. 2009, 2010] and low-rank matrix completion [Feng et al. 2014; Liu et al. 2014]. However, the priors used in these methods make assumptions about the data and noise distributions, limiting their ability to handle real-world data with more complex noises.

Data-driven methods learn from a large database to acquire intrinsic knowledge of MoCap data, such as KD-tree [Baumann et al. 2011; Tautges et al. 2011], local PCA [Chai and Hodgins 2005; Liu and McMillan 2006], self-similarity [Aristidou et al. 2018], sparse encoding [Wang et al. 2016; Xiao et al. 2015], and model averaging [Tits et al. 2018]. With the advancement of deep-learning, a number of neural-based methods have emerged [Chen et al. 2021; Ghorbani and Black 2021; Holden 2018; Pavllo et al. 2018; Perepichka et al. 2019]. SOMA [Ghorbani and Black 2021] uses a transformer-based network to automatically label the marker point cloud. While SOMA assigns unlabeled markers to specific body parts (such as the inner left wrist and right elbow), our method solves motions with labeled markers. The majority of works focus on repairing occluded markers and solving motions. [Pavllo et al. 2018] uses auto-encoder-based models to recover occluded hand markers and solve hand motions. [Perepichka et al. 2019] presents a missing marker completion method by comparing motions generated by commercial software with those generated by a neural solver, the accuracy of these methods is determined by the neural solver. DeepMerf [Madadi et al. 2021] fixes occluded markers with a denoising autoencoder and employs a cascading network to regress SMPL body parameters [Pavlakos et al. 2019] from joint positions estimated with an attention model. However, this method is based on the SMPL model and is difficult to apply to characters with different skeleton topologies. Holden [Holden 2018] solves skeletal motions from MoCap data for characters with arbitrary skeletons using a simple forward neural network with residual blocks. Their method solves the motion and skeleton frame by frame and necessitates smoothing as a post-processing step for temporal motion continuity. MoCap-Solver [Chen et al. 2021] solves motions in a temporal window by encoding body shapes, marker distributions, and motions separately to ensure temporal continuity. Their method reconstructs clean markers based on skinning functions with solved motions, which ignores the complexity of marker motions (e.g. sliding over the body surface), and may induce additional errors due to motion solving errors. In contrast, our method fills the occlusion by incorporating hand-crafted priors and learned intrinsic priors, which accurately reconstruct occluded markers. Furthermore, previous methods [Chen et al. 2021; Holden 2018] do not explicitly quantify the local feature of markers, which negatively affects solving result of certain large body motions and fine multi-fingered hand motions. Unlike them, our method extracts local features by constructing a heterogeneous graph that distinguishes markers and joints as different types of nodes and performing graph convolution operations on it, which significantly improves solving accuracy.

3 METHOD

Given a sequence of raw body and hand marker data $M_{raw} \in \mathbb{R}^{T\times|M|\times 3}$, i.e., the positions of markers M in a temporal window of T frames, and their occlusion status $O \in [0,1]^{T\times|M|}$, which is a binary mask indicating the marker visibility, our method aims to solve the body and hand motions $Y \in \mathbb{R}^{|J|\times 9+3}$, which consists of the body's global translation and the rotation of every joint J, and the underlying skeleton $S \in \mathbb{R}^{|J|\times 3}$. In this section, we will first explain the neighbor markers and then present our pre-process method and the motion solving network. At the end, we describe the data augmentation algorithm.

Our method leverages spatial locality based on neighbor markers $\mathcal{N}(M)$. To find neighbor markers $\mathcal{N}(M_i)$ for marker M_i, we first compute the variances of pairwise distances between markers across the motions and choose K markers with the lowest variances. The stable distances between M_i and its neighbor markers help to efficiently solve for occluded marker positions. Furthermore, the neighbor markers can better represent motions of the corresponding body parts, assisting the motion solving networks in accurately solving body and hand motions.

Initially, the raw data M_{raw} is subjected to a cleaning process that includes occlusion gap filling and outlier removal, resulting in refined data M_{clean}. This step contributes to enhancing accuracy of the motion solving network. The occlusion gap filling process is a two-step process in which we first calculate an approximate value X_{EDM} of occluded markers based on their distances to neighbor markers, then fine-tune it using a bidirectional Long-Short Time Memory (biLSTM) network. Outliers are detected by identifying abnormal values in markers' acceleration curves and replaced by simple spline interpolation. To reduce training difficulty, we split body and hand markers and remove their global transformations by explicitly calculating local coordinate systems of wrist and waist markers. The resultant aligned clean markers are denoted by $M_{clean,local}$.

Next, we use heterogeneous graph neural networks to solve the motion frame-by-frame, as shown in Fig. 3. We train three networks with similar structures to separately solve body and hand motions. We begin by constructing a heterogeneous graph $\mathcal{G}$ comprised of the marker graph $\mathcal{G}_M$, whose nodes and edges represent marker and their spatial adjacency, and the joint graph $\mathcal{G}_J$, whose nodes and edges represent joints and bones, respectively. The connections between two graphs are based on spatial proximity of markers and joints. Our network employs both intra-graph and inter-graph convolution components, with the former operating within subgraphs and the latter transferring information between them. The network first extracts the markers' features from the marker graph $\mathcal{G}_M$ using a stack of intra-graph convolution layers, and then feeds these features into two branches: a global branch composed of residual blocks that extract global motion features, and a local branch composed of inter-graph convolution layers that extract local features of marker and joint nodes, representing the motion pattern of the body part. To obtain body motions, the features extracted by local and global branches are concatenated and fed into a stack of intra-graph convolution layers operating on joint graph $\mathcal{G}_J$.

Our approach generates synthetic data for data augmentation. We simulate occlusion gaps by randomly sampling gaps based on the occlusion probability of markers $\hat{p}_{occ,i}$ and the gap length distribution $\hat{g}_l$. Furthermore, we shift markers to simulate outliers based on given shifting probabilities and intensities p_{shift} and σ_{shift}.

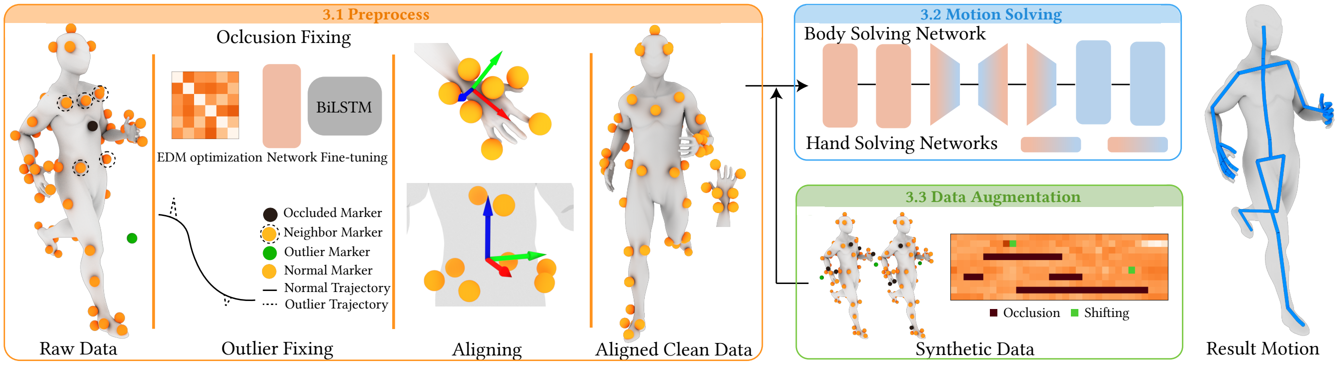

Fig. 2. The pipeline of our method consists of three main modules. Left: Given the raw MoCap data, we clean the data by fixing occluded markers with the EDM algorithm with a bi-directional neural network and identifying outliers by locating the peaks on the marker acceleration curve, and remove the global transformation of body and hand markers by explicitly calculating their local coordinates using waist and wrist markers. Top right: Using the aligned clean body markers, we solve the body and hand motions separately using heterogeneous graph neural networks, whose details are shown in Fig. 3. Bottom right: We augment our dataset by randomly sampling gaps and adding shifting based on actual MoCap data.

3.1 Preprocess

Real-world optical motion capture data is frequently plagued by occlusions and outliers, making it difficult for the solver to solve motions accurately. Furthermore, the global transformations of body and hand reduces the accuracy of the neural motion solver. To mitigate these issues, we preprocess the data by filling occlusions and remove outliers, and align markers to remove the global transformation, thereby facilitating more effective network training.

Marker occlusion is a common problem in optical motion capture systems caused by body occlusion or marker dropping. Occlusion gap filling is a highly non-linear problem due to the complexity of body motion and the uncertainty of the lengths and quantity of occlusion gaps. To address this issue, we propose a recurrent neural network-based approach. In contrast to previous learning-based methods, we use the Euclidean distance matrix optimization (EDM) algorithm to obtain an initial value for the network by solving the position of occluded markers based on the distance matrix of their neighbor markers. This approach is based on the observation that during motion, the distances between neighbor markers vary slightly. As a result, we calculate the position of the occluded marker based on its distances to neighbor markers in its visible frames.

Specifically, given a marker $M_i \in \mathbb{R}^3$ that is occluded between frames s and e, we begin by identifying K visible neighbor markers $M_j \in \mathcal{N}(M_i)$ between these frames. We then compute distance matrices $D^s_i, D^e_i \in \mathbb{R}^{(1+K)\times(1+K)}$ for M_i and its neighbor markers at the start and end of the occlusion gap, respectively. We use linear interpolation to estimate the distance matrix at frame t within this interval: $D^t_i = \frac{(e-t)D^s_i + (t-s)D^e_i}{e-s}$. Formally, D^s_i and D^e_i lie in a space of symmetric positive definite matrices where interpolation should follow geodesics. In practice, we found that linear interpolation provides a great approximation of the distance matrix at frame $t \in [s, e]$, as the distance between M_i and neighbor markers vary only slightly. Furthermore, instead of treating neighboring markers as rigid bodies by using fixed distance matrices, the interpolated matrix reflects subtle change of distances between markers due to markers sliding over the skin surface or dynamic effects of soft body tissues. Following that, we employ an EDM optimization method [Zhou et al. 2020] to compute occluded marker's position $M^t_{i,EDM}$. Formally, the EDM optimization algorithm is defined as follows:

$$\min_{M^t_{EDM,i}} f(M^t_{EDM,i}, M^t_j) = \sum_{M_j \in \mathcal{N}(M_i)} \left| ||M^t_{EDM,i} - M^t_j||^2 - D^t_i[i,j] \right|. \tag{1}$$

To solve Eq. 1, the optimization is divided into two stages, where we first compute an initial guess on a set of marker positions $P^t_i \in \mathbb{R}^{(K+1)\times 3}$ that satisfies D^t_i, then align P^t_i with $\mathcal{N}(M_i)$ by a rigid transformation. To calculate P^t_i, where one marker position corresponds to the occluded marker and the others correspond to its neighbor markers, we compute their initial positions using the Multi-Dimensional Scaling method [Torgerson 1952] as follows:

$$P^t_i = [\mathbf{p}_1, ..., \mathbf{p}_r] = diag(\sqrt{\lambda_1}, ..., \sqrt{\lambda_r})[\mathbf{x}_1, ..., \mathbf{x}_r]^T, \tag{2}$$

where $\mathbf{p}_i$ represents the points in P^t_i, the eigenvalues λ_i and corresponding eigenvectors $\mathbf{x}_i$ are from eigen decomposition of the MDS embedding $-\frac{1}{2}(JD^i_tJ)$, where $J = I - \frac{1}{K+1}\mathbf{1}\mathbf{1}^T$ is the centering matrix with I being the identity matrix and $\mathbf{1}$ being vector of all ones and r is the rank of JDJ.

Next, we align the initial guess P^t_i to $\mathcal{N}(M_i)$ using rigid transformation by solving the well-known Rotation Orthogonal Procrusts Problem [Zhang et al. 2010]. For simplicity, we use a closed-form solution by Singular Value Decomposition (SVD).

However, the distances between markers do not fully represent the complex motion patterns of markers. Next, we fine-tune the EDM result using a recurrent neural network that learns from a large dataset. The network is made up of an intra-convolution layer that extracts markers' local features and a biLSTM layer that uses temporal information. In the following section, we will go over the intra-convolution layer in more detail. The network takes as input the aligned marker positions and their occlusion status, denoted as $[M^t_{EDM,local,i}, 0]$ for occluded markers and $[M^t_{clean,local,i}, 1]$ for visible markers. The network estimates the offset$M^t_{off,i}$ from the EDM results to the ground truth using the following loss function:

$$\mathcal{L}_{occ} = \sum ||(1-O^t_i) \odot (\hat{M}^t_{clean,local,i} - (M^t_{EDM,local,i} + M^t_{off,i}))||, \tag{3}$$

where $\hat{M}^t_{clean,local,i}$ is ground truth, and $\odot$ is element-wise product.

Another critical issue that can affect the quality of optical MoCap data is outlier. A common method for reducing such jittery movements in data is to apply a filter to the entire sequence, such as the Savitzsky-Golay filter [Savitzky and Golay 1964] used in [Holden 2018], which may reduce the fidelity of outlier-free parts. To reduce the affection to these parts, we identify the outliers and only fix the frames near where the outlier appears. In contrast to normal markers, which have smoother movement patterns, outliers have sudden jumps or accelerations due to labeling errors or other sources of noise. As a result, we use movement smoothness as a criterion to identify outliers, which are defined as markers with acceleration peaks that exceed a certain threshold. We remove the marker sequence around the outlier and fill the gap with spline interpolation to improve motion fidelity.

To remove global transformation, we use a straightforward yet effective method for aligning body and hand markers to the local space. Similar with [Chen et al. 2021], we select markers around rigid body parts, such as the waist and wrist. However, we observe that even these markers are not rigid bodies and exhibit slight non-rigid deformations, potentially causing instability in the rigid body alignment. To tackle this issue, we explicitly compute local coordinate systems using relationships between reference markers. A more detailed explanation can be found in the supplementary material.

3.2 Motion Solving

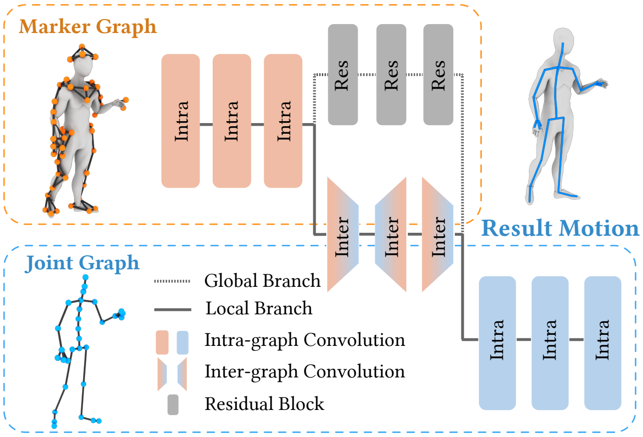

Fig. 3. Our solving network structure. The network operates on a heterogeneous graph comprising the marker graph and joint graph. It first extracts the local features of markers, which are subsequently transferred to joint features and finally transformed into motion.

We construct a heterogeneous graph $\mathcal{G} = (\mathcal{G}_M, \mathcal{G}_J, \mathcal{A}_{MJ})$, using the markers M and joints J. This graph comprises the marker graph $\mathcal{G}_M$, joint graph $\mathcal{G}_J$, and the adjacency matrix $\mathcal{A}_{MJ}$ representing the edges between the two subgraphs. The marker graph is denoted as $\mathcal{G}_M = (\mathcal{V}_M, \mathcal{E}_M)$, where $\mathcal{V}_M$ is the set of graph nodes indicating markers, and $\mathcal{E}_M$ denotes its edges connecting neighbor markers (see the second paragraph of Section 3). In the joint graph $\mathcal{G}_J = (\mathcal{V}_J, \mathcal{E}_J)$, each node in $\mathcal{V}_J$ represents a joint and edges in $\mathcal{E}_J$ are identical to the connections between joints. The connections are dictated by the target skeleton structure. We construct the adjacency matrix $\mathcal{A}_{MJ}$ based on the distance between joints and markers at T-pose, where $\mathcal{V}_{M,i}$ and $\mathcal{V}_{J,i}$ are connected if their corresponding marker and joints' distance is below a threshold. To represent the global transformation, we add an additional node to $\mathcal{G}_J$ that connects to every other nodes $\mathcal{V}$ in both $\mathcal{G}_J$ and $\mathcal{G}_M$.

The inputs of our network are positions of aligned markers and their visibility at frame t: $[M^t_{clean,local}, O^t]$, which are mapped to their corresponding nodes $\mathcal{V}_M$ in $\mathcal{G}_M$. The output of the network comprises features of joint nodes $\mathcal{V}_J$ in $\mathcal{G}_J$. Specifically, the output of $\mathcal{V}_{J,i}$ contains the rotation matrix $Y^t_i \in \mathbb{R}^9$ and the joint offset $S^t_i \in \mathbb{R}^3$. Since the network does not guarantee orthogonality, we use Gram-Schmidt orthogonalization as a post-process for the rotation matrix. The first three dimensions of the translation node represent the global translation. To ensure bone length consistency, the final joint offset is computed as the average of estimated joint offsets across the motions, denoted by $S_i = mean(S^1_i, ..., S^T_i)$.

Our convolution operations are built based on the skeletal convolution in [Aberman et al. 2020]. To gather information from neighboring nodes, the operation employs a learned filter. There are some distinctions between intra- and inter-convolutions: the former aggregates the local features of the nodes' neighbors of the same type, whereas the latter aggregates the local features of their neighbors of a different type. A marker node in our scenario gathers information from neighboring marker nodes in intra-convolution and aggregates features of linked joint nodes in inter-convolution. The convolution operation is illustrated below:

$$f'_{\mathcal{V}_i} = \sum_{\mathcal{V}_j \in \mathcal{N}(\mathcal{V}_i)} W_{ij} f_{\mathcal{V}_j} + b_i, \tag{4}$$

where $f_{\mathcal{V}_i}$ and $f'_{\mathcal{V}_i}$ are the input and output features of node $\mathcal{V}_i$, respectively. $\mathcal{N}(\mathcal{V}_i)$ is the set of its neighbors, and W_{ij}, b_i are the learned filters and biases, respectively. It should be noted that unlike traditional convolution operations, our convolution operation employs different filters and biases for different edges.

We train the network using the following loss with three terms:

$$\mathcal{L}_{solving} = \lambda_M(Y - \hat{Y}) + \lambda_S(S - \hat{S}) + \lambda_{FK}(FK(Y, S) - FK(\hat{Y}, \hat{S})), \tag{5}$$

where the first two terms push the estimated motion and skeleton Y, S to their ground truths $\hat{Y}, \hat{S}$, and $FK()$ is the forward kinematic function computing global positions of joints according to the motion and skeleton, and $\lambda_M, \lambda_S, \lambda_{FK}$ are predefined weight factors.

3.3 Data Augmentation

One of the most crucial tasks for our networks is to learn to deal with complex occlusions. However, obtaining such noise data is hard due to the high cost of optical MoCap systems and the low frequency of

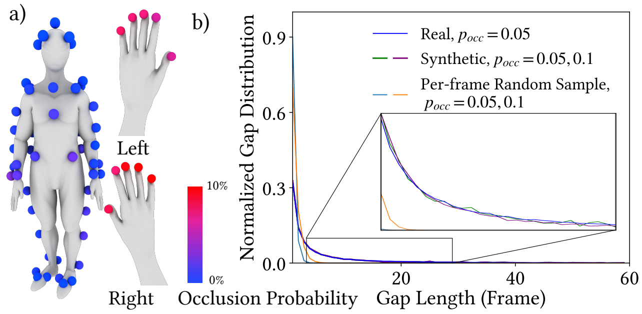

Fig. 4. (a) Occlusion probabilities and (b) gap length distribution of the real MoCap data. Markers on the right side of body tend to have higher occlusion probabilities than those on the left side. The distributions of gap length follow a heavy-tailed distribution.

noise occurrences in modern MoCap scenarios. For instance, our real MoCap dataset has an overall occlusion probability of only 5%. Consequently, there is a limited amount of data available, which makes it easy for the networks to overfit. To address this issue, we draw inspiration from masked autoencoders [He et al. 2022] and feed the network with strategically perturbed data. We apply a training regime based on representation learning and data augmentation by simulating shifting and occlusion gaps based on their spatial and temporal patterns. Notably, markers located at different positions on the body exhibit varying occlusion probabilities $p_{occ,i}$ due to differences in motion patterns. For instance, hand markers have higher occlusion probabilities than that of other body parts, as they are most densely placed and the motions of hand are the most complex. Furthermore, because the motion and capture situation is continuous, occlusions usually last for a period of time rather than occurring haphazardly in single frames. The distribution of occlusion gap lengths follows a heavy-tailed distribution that is consistent across different markers. To simplify the process, we use the average gap length distribution of all markers when generating synthetic data. We illustrate the occlusion probabilities $\hat{p}_{occ,i}$ and the occlusion gap length distribution $\hat{g}_l$ collected from real data on Fig. 4. We also illustrate the gap length distribution of our method and per-frame random sample methods [Chen et al. 2021; Holden 2018]. Our method generates gaps that are closest to the real distribution, whereas the per-frame random sample method fails to generate gaps longer than 6 frames.

Given the overall occlusion probability p_{occ} and total sequence length L, we first compute the gap number $N_{l,i}$, which represents the number of gaps with length l for marker M_i, using the gap length distribution and marker occlusion probabilities as follows:

$$N_{l,i} = \lfloor \frac{L}{l} \frac{l\hat{g}_l}{\sum(l\hat{g}_l)} p_{occ} \frac{\hat{p}_{occ,i}}{\sum \hat{p}_{occ,i}}) \rfloor, \tag{6}$$

where $\frac{l\hat{g}_l}{\sum(l\hat{g}_l)}$ indicates gap possibility weighted by gap length, $p_{occ} \frac{\hat{p}_{occ,i}}{\sum \hat{p}_{occ,i}}$ represents weighted occlusion probability for marker M_i, and $\lfloor \cdot \rfloor$ represents floor operator.

Using the calculated occlusion gap numbers $N_{l,i}$, we randomly select segments from longest to shortest without overlapping and set them to occluded. Selecting segments randomly or from shortest to longest may lead to insufficient space for occlusion gaps. After that, we add shifts to markers on randomly sampled frames by applying offsets with the uniform distribution $o \sim U(-\sigma_{shift}, \sigma_{shift})$.

4 RESULTS AND EXPERIMENTS

We conduct our experiments on two distinct datasets: a real dataset and a synthetic dataset, each featuring characters with different marker configurations and skeleton structures. The real dataset was captured in a game studio, and the synthetic dataset is generated by driving the SMPL+H body [Romero et al. 2017] using the body motions from the CMU MoCap dataset [CMU 2000] and the hand motions from the GRAB [Taheri et al. 2020] dataset. We use 6 neighbor markers with the lowest distance variances for occlusion fixing and 3 for motion solving networks. The supplementary material contains additional information about our dataset, network architectures, hyper-parameters, and other implementation details.

4.1 Evaluation

We present the results of our method for unseen motions in Fig. 5. Our method successfully handles diverse motions, including large body movements and fine hand motions. Additionally, our method is robust to marker occlusions and outliers.

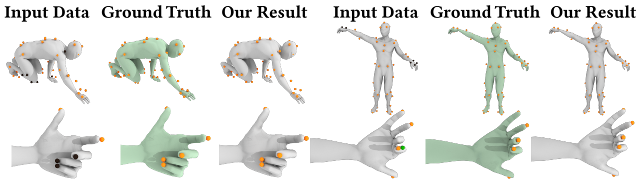

Fig. 5. Our method can accurately solve diverse unseen body motions and fine multi-fingered hand motions. Our method is robust to occlusions (black ball) and outliers (green ball).

Following that, we assess the efficacy of our data augmentation method on the real dataset. We randomly sample the training data, selecting 20%, 40%, ..., 100% to train the occlusion fixing and solving networks, and then test them using the same testing set. The quantitative results are shown in Fig. 6. The losses decrease as the data scale increases because the training set contains more diverse data. We also run experiments that augment training data by adding corrupted data from sampled data to the training set. We compare our data augmentation method with the per-frame random sample method used in [Chen et al. 2021; Holden 2018]. Our method improves network performance for marker occlusion fixing because it can generate long occlusion gaps with a data distribution similar to real data. The per-frame random sample method degrades the network's performance on real data, as it fails to generate long occlusion gaps. In terms of solving network, our data augmentation method improves network performance by 10% when only a small amount of training data is available. The per-frame random sample method, on the other hand, does not significantly improve network performance.

On the real dataset, we tested different selection criteria and numbers of neighbor markers. The results show that choosing markers

Table 1. Comparison with other methods and ablations for our method.

		Optimization-based Methods (Vicon for Real Mosh++ for Synthetic)		[Holden 2018]		MoCap-Solver		Ours (w/o EDM initial value)		Ours (w/o splitting body & hand)		Ours (w/o marker convolution)		Ours	
		Body	Hand	Body	Hand	Body	Hand	Body	Hand	Body	Hand	Body	Hand	Body	Hand
Real	JOE (°)	4.21	1.37	2.54	0.80	1.89	0.58	1.27	0.58	1.31	1.17	1.78	0.57	**1.22**	**0.45**
	JPE (cm)	1.75	0.49	1.02	0.27	0.89	0.20	0.76	0.21	0.85	0.55	0.85	0.18	**0.61**	**0.15**
	OMPE (cm)	5.32	2.01	3.62	1.58	3.27	1.46	3.83	1.71	2.76	1.76	2.60	1.15	**2.55**	**1.11**
Synthetic	JOE (°)	5.72	6.37	3.53	2.21	2.76	1.72	2.47	1.78	2.25	3.41	2.65	1.70	**2.15**	**1.59**
	JPE (cm)	2.02	0.79	1.37	0.27	1.08	0.21	1.12	0.23	1.17	0.53	1.09	0.20	**1.00**	**0.18**
	OMPE (cm)	7.20	1.25	4.58	0.58	4.28	0.57	5.22	0.83	4.02	1.18	**3.66**	0.38	3.67	**0.35**

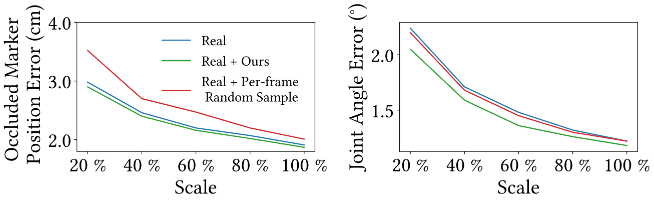

Fig. 6. Occluded marker position error and joint angle error vs. training data scale. We also test two data augmentation methods: "Real" denotes real dataset without augmentation, whereas "Real + Ours" denotes the same dataset supplemented with data generated using our augmentation method.

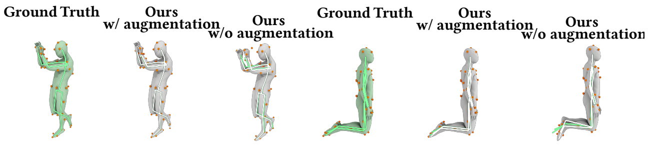

Fig. 7. Qualitative comparisons of motions solved using our network trained with 20% of training set with and without data augmentation. We render the skeletons and overlay the ground truth's skeleton onto ours.

with the smallest distance variances as neighbor markers outperforms choosing those with the smallest mean distances. Our approach yields the best results with six markers for occlusion filling and three markers for the network resolution. Please refer to supplementary materia for an in-depth qualitative analysis and discussion.

We conduct three ablation studies to validate the efficacy of our method's modules. The first study does not use the EDM optimization for the occlusion fixing module, and the second does not separate the body and hand markers and solves their motions in a single network. Finally, the third uses residual modules rather than performing intra-convolutions on the marker graph $\mathcal{G}_M$. The quantitative results are shown on the right side of Table 1. For occlusion fixing, the initial value given by EDM improves marker position accuracy by approximately 40%. As a result of more accurate marker positions, the solving network estimates motions closer to the ground truth with errors reduced by approximately 4% for the body and 20% for the hands (where markers have higher occlusion probabilities). Solving whole-body motions with a single network negatively affects the accuracy of both hand and body motions, which justifies the necessity of splitting body and hand markers and solving them using separate networks, which improves hand motion accuracy in terms of angle error by 60% and body motion accuracy by 5%. Solving motions by first extracting markers' local features on the marker graph $\mathcal{G}_M$ significantly improves performance of the solving network, reducing errors in terms of angle differences by approximately 30% for body motions and 15% for hand motions.

4.2 Comparison with Prior Methods

We compare our method with two neural-based approaches: [Holden 2018] and MoCap-Solver [Chen et al. 2021] on the two datasets. Additionally, we compare our method with two optimization-based approaches: Vicon [Vicon 2023] for the real dataset and Mosh++ [Mahmood et al. 2019] for the synthetic dataset. The neural-based approaches are trained with the same dataset and evaluated using the same unseen data. To ensure a fair comparison, the body and hand markers are aligned using the same strategy and trained using separate networks for neural approaches. We use three metrics to quantitatively evaluate the reconstructed marker sequence and the solved motions, namely OMPE (Occluded Marker Position Error), JOE (Joint Orient Error), and JPE (Joint Position Error). The first one is the distance between the estimated occluded marker position and the ground truth. The latter two represent different aspects of motion solving accuracy: JOE is the angle difference for joints and JPE is the global joint position difference.

On the left of Table 1, we list the quantitative results for comparison methods. Our method outperforms the baseline methods for all metrics. For body motions and markers, JOE and JPE are about 30% lower, while OMPE is 20% lower than the best of them. For hand, all metrics are approximately 20% lower than the state-of-the-art methods. Furthermore, our method achieves lower average errors and generates fewer large errors compared with other methods. We present two detailed comparisons of body and hand motions in Fig. 8, where our method tends to generate motions with the highest accuracy. Optimization based methods [Mahmood et al. 2019; Vicon 2023] lack the learned prior knowledge for the occluded markers and motions, which fails to accurately solve motions when large numbers of markers are occluded. For two reasons, our method outperforms other neural-based methods [Chen et al. 2021; Holden 2018]. First, our method better reconstructs the positions of the occluded markers, providing a better input to the solving network.

Second, our solving network extracts local features of each marker and joint, which aids in generating motions that are close to the ground truth. We present additional qualitative comparisons of large body motions on Fig. 9 and fine hand motions on Fig. 10.

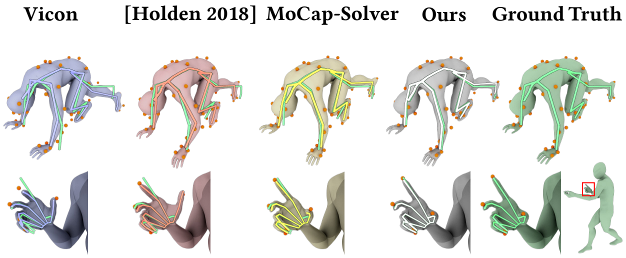

Fig. 8. Qualitative comparisons of motions solved using various methods. Top / bottom row: bodies with large motions and hands with fine motions, with the hand marked using red boxes and enlarged for easier comparison. Our method generates motions that are closest to the ground truth.

5 CONCLUSION, LIMITATIONS, AND FUTURE WORK

We present a learning-based method that leverages locality to clean optical MoCap data and solve body and hand motions. We have evaluated it on diverse and complex benchmarks and highlighted improved performance over prior baseline methods. Compared with previous methods, our method better recovers the occluded markers' positions and more accurately solves the body and hand motions. Additionally, our method generates synthetic MoCap data with distributions close to real data.

Our method has limitations. First, our data augmentation method solely considers the spatiotemporal distribution of occlusions. Incorporating other occlusion and noise patterns, such as simultaneous occlusion, varying shifting intensities of markers on different parts of body, and mislabeling between close markers, may yield a distribution more closely resembling real data. Second, our method is unable to detect subtle marker swaps, such as those between the index and middle fingers. Detecting and correcting such exchanges is a worthwhile future endeavor.

ACKNOWLEDGMENTS

Xiaogang Jin was supported by Key R&D Program of Zhejiang (No. 2023C01047) and the National Natural Science Foundation of China (Grant Nos. 62036010, 61972344). We thank Hua Zhang, Yuchen Liu, Qishen Liu, Cheng Ge, and other colleagues in NExT Studios for preparing the training data and rendering the video.

REFERENCES

2000. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware Networks for Deep Motion Retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

Andreas Aristidou, Daniel Cohen-Or, Jessica K Hodgins, and Ariel Shamir. 2018. Self-similarity Analysis for Motion Capture Cleaning. In *Computer graphics forum (CGF)*, Vol. 37. 297–309.

Andreas Aristidou and Joan Lasenby. 2013. Real-time marker prediction and CoR estimation in optical motion capture. *The Visual Computer* 29 (2013), 7–26.

Jan Baumann, Björn Krüger, Arno Zinke, and Andreas Weber. 2011. Data-Driven Completion of Motion Capture Data. *VRIPHYS 2011 - 8th Workshop on Virtual Reality Interactions and Physical Simulations*, 111–118.

Michael Burke and Joan Lasenby. 2016. Estimating Missing Marker Positions using Low Dimensional Kalman Smoothing. *Journal of biomechanics* 49, 9 (2016), 1854–1858.

Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. 2020. Long-term human motion prediction with scene context. In *ECCV 2020*. 387–404.

Jinxiang Chai and Jessica K Hodgins. 2005. Performance Animation from Low-dimensional Control Signals. In *ACM SIGGRAPH 2005 Papers*. 686–696.

Kang Chen, Yupan Wang, Song-Hai Zhang, Sen-Zhe Xu, Weidong Zhang, and Shi-Min Hu. 2021. MoCap-Solver: A Neural Solver for Optical Motion Capture Data. *ACM Transactions on Graphics (TOG)* 40, 4, Article 84 (2021), 11 pages.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. 2023. Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 18000–18010.

Qiongjie Cui, Huaijiang Sun, Yue Kong, and Xiaoning Sun. 2021. Deep Human Dynamics Prior. In *Proceedings of the 29th ACM International Conference on Multimedia (MM)* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 4371–4379. https://doi.org/10.1145/3474085.3475581

Yinfu Feng, Jun Xiao, Yueting Zhuang, Xiaosong Yang, Jian J Zhang, and Rong Song. 2014. Exploiting Temporal Stability and Low-rank Structure for Motion Capture Data Refinement. *Information Sciences* 277 (2014), 777–793.

Nima Ghorbani and Michael J Black. 2021. Soma: Solving optical marker-based mocap automatically. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 11117–11126.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked Autoencoders are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16000–16009.

Lorna Herda, Pascal Fua, Ralf Plankers, Ronan Boulic, and Daniel Thalmann. 2000. Skeleton-based Motion Capture for Robust Reconstruction of Human Motion. In *Proceedings Computer Animation 2000*. IEEE, 77–83.

Daniel Holden. 2018. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Yilei Hua, Wenhan Wu, Ce Zheng, Aidong Lu, Mengyuan Liu, Chen Chen, and Shiqian Wu. 2023. Part Aware Contrastive Learning for Self-Supervised Action Recognition. *International Joint Conference on Artificial Intelligence (IJCAI)* (2023).

Adam Kirk, James F O'Brien, and David A Forsyth. 2004. Skeletal Parameter Estimation from Optical Motion Capture Data. In *ACM SIGGRAPH 2004 Sketches*. 29.

Lei Li, James McCann, Nancy S. Pollard, and Christos Faloutsos. 2009. DynaMMo: Mining and Summarization of Coevolving Sequences with Missing Values. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Association for Computing Machinery, 507–516.

Lei Li, James McCann, Nancy S. Pollard, and Christos Faloutsos. 2010. BoLeRO: a principled technique for including bone length constraints in motion capture occlusion filling. In *Symposium on Computer Animation (SCA)*. 179–188.

Guodong Liu and Leonard McMillan. 2006. Estimation of Missing Markers in Human Motion Capture. *The Visual Computer (TVC)* 22 (2006), 721–728.

Xin Liu, Yiu-ming Cheung, Shu-Juan Peng, Zhen Cui, Bineng Zhong, and Ji-Xiang Du. 2014. Automatic Motion Capture Data Denoising via Filtered Subspace Clustering and Low Rank Matrix Approximation. *Signal processing* 105 (2014), 350–362.

Meysam Madadi, Hugo Bertiche, and Sergio Escalera. 2021. Deep Unsupervised 3D Human Body Reconstruction from a Sparse Set of Landmarks. *International Journal of Computer Vision (IJCV)* 129, 8 (2021), 2499–2512.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *The IEEE International Conference on Computer Vision (ICCV)*. 5442–5451.

Tewodros Legesse Munea, Yalew Zelalem Jembre, Halefom Tekle Weldegebriel, Longbiao Chen, Chenxi Huang, and Chenhui Yang. 2020. The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation. *IEEE Access* 8 (2020), 133330–133348.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 10975–10985.

Dario Pavllo, Thibault Porssut, Bruno Herbelin, and Ronan Boulic. 2018. Real-time Marker-based Finger Tracking with Neural Networks. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 651–652.

Maksym Perepichka, Daniel Holden, Sudhir P Mudur, and Tiberiu Popa. 2019. Robust Marker Trajectory Repair for MoCap using Kinematic Reference. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG)*. 1–10.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.

Leonid Sigal, Alexandru O Balan, and Michael J Black. 2010. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)* 87, 1-2 (2010), 4–27.

Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. 2020. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *European Conference on Computer Vision (ECCV)*. 581–600.

Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. 2011. Motion Reconstruction Using Sparse Accelerometer Data. *ACM Transactions on Graphics (TOG)* 30, 3, Article 18 (2011), 12 pages.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2022. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Mickaël Tits, Joëlle Tilmanne, and Thierry Dutoit. 2018. Robust and Automatic Motion-capture Data Recovery using Soft Skeleton Constraints and Model Averaging. *PloS one* 13 (2018), 1–21.

Warren S Torgerson. 1952. Multidimensional Scaling: I. Theory and method. *Psychometrika* 17, 4 (1952), 401–419.

Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. 2017. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference (BMVC)*. 1–13.

Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2017. Learning from Synthetic Humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vicon. 2023. *Vicon Shogun*.

Zhao Wang, Shuang Liu, Rongqiang Qian, Tao Jiang, Xiaosong Yang, and Jian J Zhang. 2016. Human Motion Data Refinement Unitizing Structural Sparsity and Spatial-temporal information. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*. 975–982.

Jun Xiao, Yinfu Feng, Mingming Ji, Xiaosong Yang, Jian J Zhang, and Yueting Zhuang. 2015. Sparse Motion Bases Selection for Human Motion Denoising. *Signal Processing* 110 (2015), 108–122.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*.

Lei Zhang, Ligang Liu, Craig Gotsman, and Steven J Gortler. 2010. An As-rigid-as-possible Approach to Sensor Network Localization. *ACM Transactions on Sensor Networks (TOSN)* 6, 4 (2010), 1–21.

Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. 2020. Deep learning-based human pose estimation: A survey. *Comput. Surveys* (2020).

Shenglong Zhou, Naihua Xiu, and Hou-Duo Qi. 2020. Robust Euclidean embedding via EDM optimization. *Mathematical Programming Computation* 12 (2020), 337–387.

Victor Brian Zordan and Nicholas C Van Der Horst. 2003. Mapping Optical Motion Capture Data to Skeletal Motion Using a Physical Model. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 245–250.

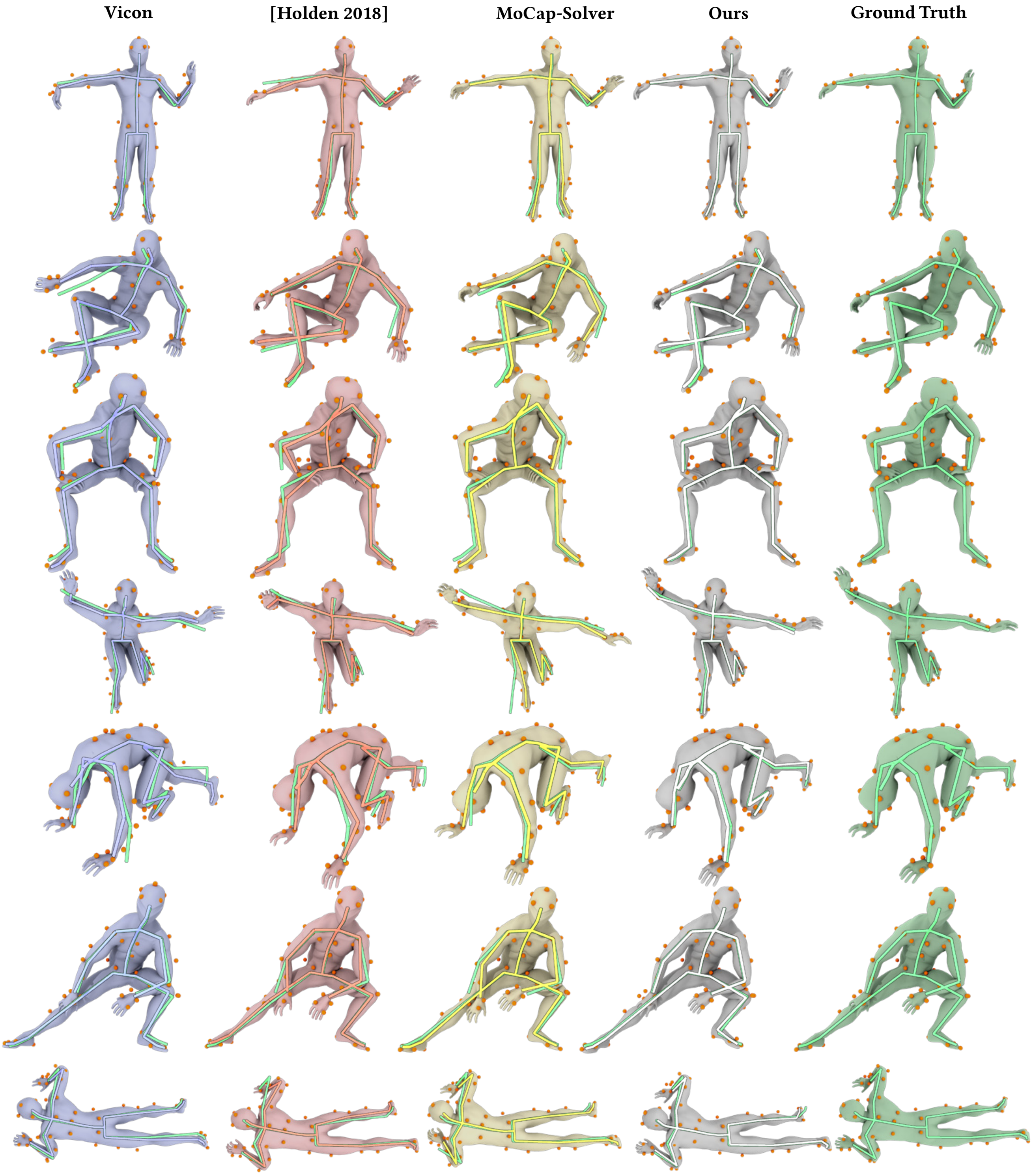

Fig. 9. Additional comparison of body motions solved by different methods. To compare with the ground truth, we render the skeletons and overlay the ground truth's skeleton onto those generated by solving methods.

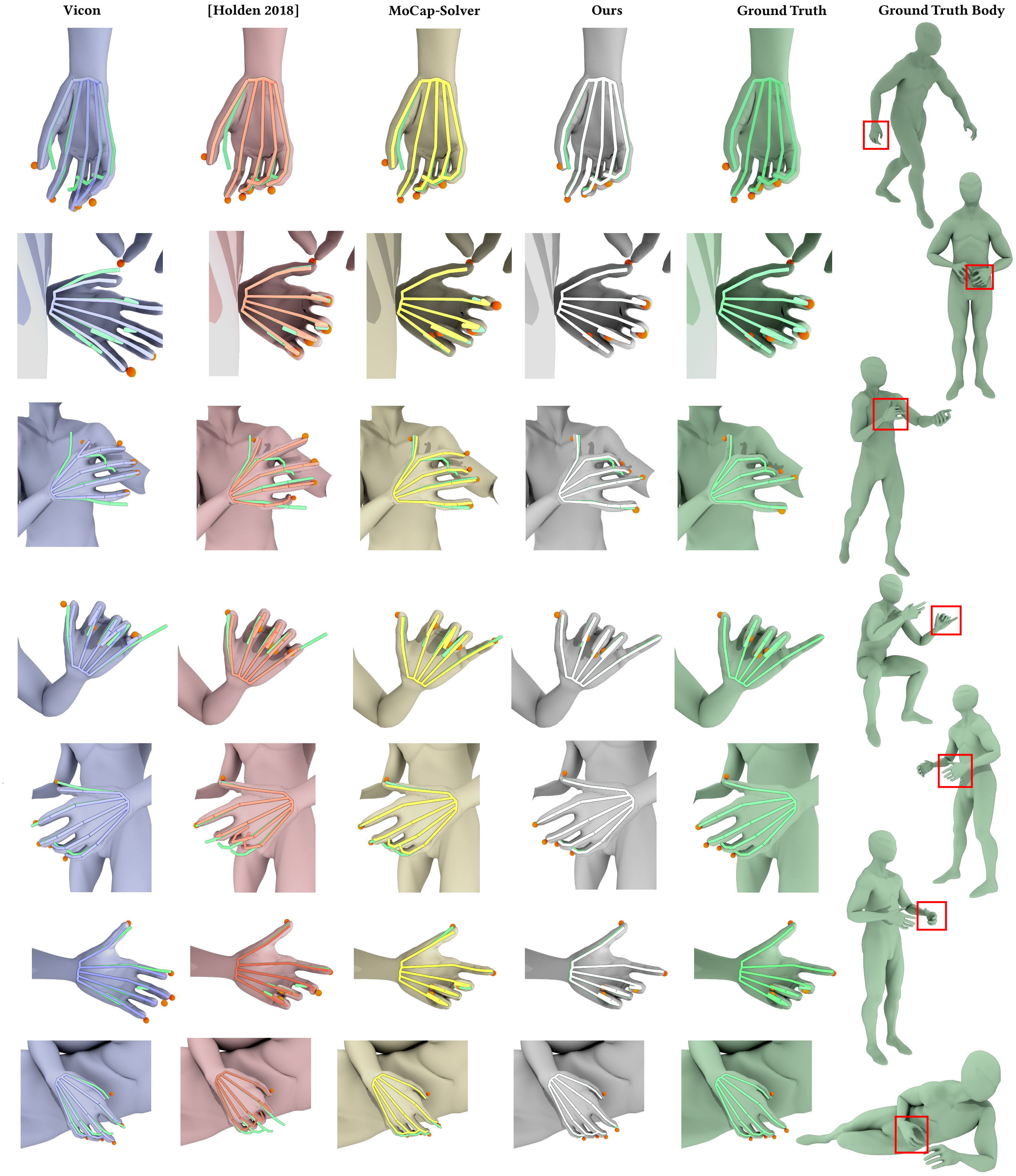

Fig. 10. Additional comparison of hand motions solved by different methods. To avoid the interference of body motions, we use the same body motions for different methods. To compare with the ground truth, we render the skeletons and overlay the ground truth's skeleton onto those generated by solving methods.