Low-rank completion for motion capture data recovery: Approaches, constraints, and algorithms

Downloaded from: https://research.chalmers.se, 2026-01-24 08:14 UTC

Citation for the original published paper (version of record):

Mohaoui, S., Dmytryshyn, A. (2026). Low-rank completion for motion capture data recovery: Approaches, constraints, and algorithms. Computer Science Review, 60. http://dx.doi.org/10.1016/j.cosrev.2025.100878

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Computer Science Review 60 (2026) 100878

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Low-rank completion for motion capture data recovery: Approaches, constraints, and algorithms

Souad Mohaoui [a,*], Andrii Dmytryshyn [b, a]

[a] School of Science and Technology, Örebro University, Örebro, Sweden
[b] Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden

ARTICLE INFO

Keywords:
Human motion recovery
Motion capture data
Missing markers
Low-rank prior
Matrix completion
Optimization algorithms
Tensor decomposition

ABSTRACT

Motion capture (MoCap) systems are indispensable tools across fields such as biomechanics, computer animation, human-robot interaction, and clinical gait analysis, owing to their ability to accurately record and analyze human movement in 3D space. Marker-based systems use reflective markers attached to subjects and video recordings to track human movement. The tracking requires markers to be detected in the video, which is not always possible due to occlusions, sensor failures, and limited camera coverage. These issues create gaps in recorded trajectories, compromising data integrity and making the motion difficult to utilize in practical applications. Therefore, a wide range of MoCap data completion techniques has been proposed to reconstruct missing trajectories while preserving the realism and dynamics of human movement. Human motion data exhibits a low-rank property due to the inherent repetitive nature of human movement as well as the correlations between joints and markers, enforced by the skeletal structure and biomechanical constraints. Low-rank completion techniques exploit this property to reconstruct missing marker positions. This paper reviews state-of-the-art low-rank completion methods for MoCap data completion, focusing specifically on optimization-based low-rank methods. These optimization approaches directly address the missing data completion problem through optimization formulations. We examine two main aspects: kinematic priors, which embed anatomical constraints, joint dependencies, and motion smoothness, and low-rank priors, which exploit inter-marker correlations through matrix and tensor formulations. We further evaluate optimization algorithms for solving these completion problems, such as alternating minimization, proximal algorithms, ADMM, and hybrid schemes, as well as the datasets and tools commonly used in the literature.

1. Introduction

The automatic capture and analysis of human motion is a rapidly evolving field, driven by its broad applications and the complexity of human movement [1–10]. Motion capture (MoCap) technology enables detailed recording of complex movement patterns by tracking the position and orientation of the human body in 3D space. The precise recording of human motion provides significant information for clinical rehabilitation, athletic performance optimization, realistic character animation, and virtual reality experiences.

The marker-based system employs specialized cameras, sensors, and wearable devices to track the movement of markers attached to key joints, such as the wrists, elbows, knees, and ankles. These systems capture the position and orientation of the markers in three-dimensional space, enabling software to reconstruct the subject's skeleton and posture. Although marker-based motion capture is widely used for its high precision, it still faces limitations such as occlusions, mismatched markers, or poor lighting conditions that lead to incomplete data. When a marker is blocked from all camera views during a recording, its trajectory is lost, creating gaps in the data. These gaps disrupt the continuity of the captured motion and pose a significant challenge to creating accurate and seamless representations of human movement (Fig. 1).

The problem of filling these gaps is a specific example of the broader missing data completion problem that appears in many data-driven fields. Although advanced MoCap systems and software offer increasingly high-quality data capture, gaps and noise remain challenging issues that require sophisticated computational solutions. The problem of recovering missing marker data is central to improving the usability and quality of MoCap recordings. Inaccurate or incomplete motion data can propagate errors in practical analyses, such as motion tracking, gait analysis, or rehabilitation assessments, which may ultimately compromise the performance of applications relying on MoCap data.

* Corresponding author.
Email addresses: souad.mohaoui@oru.se (S. Mohaoui), andrii@chalmers.se (A. Dmytryshyn).

https://doi.org/10.1016/j.cosrev.2025.100878
Received 18 June 2025; Received in revised form 13 October 2025; Accepted 3 December 2025
Available online 7 January 2026
1574-0137/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

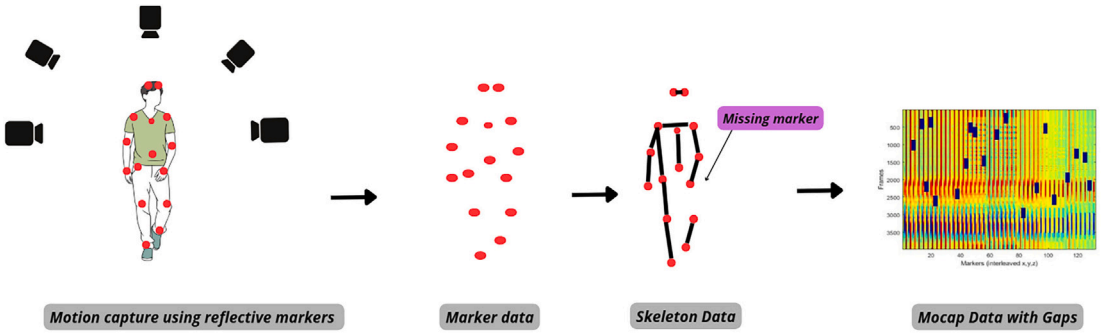

Fig. 1. Illustration of the motion capture (MoCap) process. Multiple cameras record reflective markers placed on the body to obtain the skeletal data. Occlusions or tracking errors can cause missing markers, leading to MoCap sequences with spatial and temporal gaps.

A promising solution to the missing data problem lies in exploiting the low-rank structure inherent in human motion. Although human movements are complex, they often exhibit repetitive patterns and coordinated joint movements throughout time and between individuals, allowing the data to be effectively approximated by low-rank matrices or tensors. Low-rank recovery methods exploit this underlying structure to fill in missing values via an optimization problem, which can be further enhanced by incorporating biomechanical characteristics of human movement as constraints.

Missing data represents one of the most common challenges in modern data science, affecting every domain from medical imaging and sensor networks to recommendation systems and social network analysis. The problem of incomplete observations stems from diverse factors including sensor failures, storage limitations, network interruptions, and measurement costs, creating scenarios where traditional statistical methods that assume complete data become inadequate or entirely inapplicable. Recent advances in low-rank recovery theory have introduced approaches to missing data problems across multiple domains by exploiting the observation that many high-dimensional datasets naturally exhibit low-dimensional structure.

Matrix completion methods, grounded in compressed sensing and nuclear norm relaxation, have demonstrated remarkable success in applications ranging from collaborative filtering in recommendation systems to image recovery and human motion reconstruction. In hyperspectral imaging, structured low-rank matrix factorization has been shown to effectively exploit spatial–spectral correlations for high-quality image restoration under severe data loss [11]. In wireless communications, matrix completion and sensing approaches have played a critical role in channel estimation: from handling array-inherent impairments in mmWave systems [12] to recent advances in hybrid MIMO channel estimation using low-rank matrix sensing [13]. Beyond imaging and communications, matrix completion has also been employed in computational biology, where weighted-constraint formulations improve haplotype estimation accuracy from incomplete sequencing data [14]. In recommendation systems and social networks, collaborative filtering enhanced with graph information [15] and inductive matrix completion leveraging graph neural networks [16] have extended low-rank recovery to relational and dynamic data. Despite these successes, fundamental challenges remain, as highlighted by recent complexity-theoretic results on the hardness of matrix completion [17], which underscore the importance of developing scalable, domain-aware algorithms.

While matrix completion has proven effective for two-dimensional data, many real-world applications involve inherently multi-dimensional structures that are more naturally modeled as tensors. Tensor completion methods extend low-rank recovery ideas to higher-order data by exploiting correlations across spatial, temporal, and spectral dimensions, using both tensor decomposition and tensor rank minimization approaches. In medical imaging, low-rank tensor models have enabled accelerated dynamic and quantitative MRI reconstruction, including multiparametric mapping with MR multitasking and non-Cartesian MRF reconstruction [18–21]. Tensor approaches leveraging smoothness and sparsity have improved multidimensional data recovery and robust color image reconstruction [22–27]. In hyperspectral imaging and remote sensing, low-rank tensor optimization with plug-and-play priors, fibered rank constraints, and sparsity-regularized methods has achieved superior image restoration and anomaly detection [28–35]. Recent advanced formulations, including Fourier low-rank and sparse tensor completion [36], low-rank reduced biquaternion tensor ring decomposition [37], preconditioned Riemannian optimization in tensor-train format [38], multilayer sparsity-based tensor decomposition [39], and t-Schatten-p norm minimization [40,41], have significantly improved scalability and reconstruction accuracy. Collectively, these works illustrate the growing importance of tensor-based low-rank recovery as a natural extension of matrix completion, enabling accurate reconstruction in complex high-dimensional, multi-modal data across medical imaging, remote sensing, and computational imaging applications.

The broad success of low-rank methods across imaging, communications, and remote sensing demonstrates the versatility of low-dimensional priors in recovering structured high-dimensional data. These domains share fundamental similarities with MoCap data completion in that missing entries can be inferred by exploiting redundancy and correlation across multiple modes. However, motion capture introduces unique challenges that distinguish it from image or signal recovery tasks. First, MoCap trajectories are constrained by biomechanical structure: fixed bone lengths, joint angle limits, and articulated skeleton kinematics must be preserved, whereas pixel arrays in images or voxels in MRI lack such rigid physical constraints. Second, missing data patterns in MoCap are often structured, arising from marker occlusions or tracking errors that span entire trajectories over long temporal ranges, rather than the pixel-wise or patch-wise missing patterns common in image recovery (see Fig. 5). Third, human motion exhibits strong nonlinearities and abrupt dynamics (e.g., impacts, rapid direction changes) that can weaken the low-rank assumption, whereas image and video sequences typically display smoother correlations.

Martini et al. [42] presented a survey of denoising and completion filters for motion-capture and human-pose-estimation systems. The survey paper spans general-purpose filters, dimensionality-reduction techniques, and deep learning architectures. Within this framework, low-rank and matrix-completion methods are identified as key dimensionality-reduction approaches for reconstructing incomplete MoCap sequences. However, their discussion remains at a general level, primarily outlining the role of low-rank priors among other refinement strategies. This paper aims to provide a comprehensive review of low-rank methods for MoCap data completion. It covers the evolution of low-rank matrix recovery techniques, starting with early approaches based on nuclear norm minimization, followed by more advanced

methods that incorporate various priors such as kinematic constraints, temporal continuity, and other domain-specific knowledge. We discuss approaches that rely on motion properties and skeleton information, as well as those centered on introducing low-rank priors. The integration of kinematic constraints directly into the completion process, referred to as MoCap data priors, aims to refine the low-rank matrix completion problem. Moreover, the analysis of various low-rank priors provides valuable insights into the diverse strategies employed in MoCap data recovery.

The remainder of this paper is structured as follows. Section 2 reviews existing approaches for handling missing markers in motion capture data. Section 3 presents the background and theoretical foundation, including notation, definitions, and problem statement. Section 4 explores the kinematic priors in the completion of MoCap. Section 5 examines various low-rank priors used to model the low-rank property of the MoCap data problem. Section 6 discusses matrix-based low-rank completion techniques, while Section 7 focuses on tensor decomposition methods for MoCap data completion. Section 8 reviews the key minimization algorithms used to solve these completion problems. Section 9 provides information about datasets and tools commonly used in motion capture research. Section 10 discusses the algorithmic complexity, robustness, and practical considerations of the proposed methods. Finally, Section 11 concludes by identifying current challenges and discussing relevant future research directions in this field.

2. Related works

The landscape of motion capture completion encompasses three primary methodological paradigms that differ fundamentally in their approach to missing data reconstruction. Understanding these distinctions is essential for contextualizing the focus on optimization-based low-rank methods within the broader field. Interpolation methods operate under temporal continuity assumptions, estimating missing values through relationships with neighboring time points. These approaches assume that motion exhibits smooth temporal transitions and reconstruct missing data by fitting curves or applying filters to available neighboring markers. While computationally efficient and suitable for real-time applications, interpolation methods are fundamentally limited to short gaps and fail when motion exhibits rapid changes or discontinuities (Fig. 6).

Data-driven methods learn reconstruction patterns from external training datasets, applying learned models to predict missing values based on similarity matching or statistical modeling. These approaches can handle complex motion patterns by leveraging prior knowledge encoded in training corpora, but introduce dependency on the quality and representativeness of training data. They suffer from the well-documented out-of-sample problem when test sequences diverge from training distributions, limiting their applicability to novel or unusual motions.

Low-rank methods exploit the intrinsic low-dimensional structure of human motion, operating on the fundamental principle that motion data naturally resides in low-dimensional subspaces due to biomechanical constraints and kinematic relationships. This category encompasses various approaches that leverage the mathematical property that human motion matrices have significantly fewer degrees of freedom than their ambient dimensions would suggest. Within this paradigm, two primary methodological streams have emerged: classical PCA-based approaches that decompose motion data into principal components for reconstruction, and modern optimization-based frameworks that directly formulate completion as constrained optimization problems. Optimization-based low-rank methods exploit the intrinsic low-dimensional structure of human motion through constrained optimization frameworks that directly minimize rank-based objectives while satisfying data fidelity constraints. This paradigm operates on the principle that human motion naturally resides in low-dimensional subspaces due to biomechanical constraints and kinematic relationships. Unlike interpolation methods, they can handle arbitrary gap patterns and durations. Unlike data-driven approaches, they require no external training data and provide theoretical recovery guarantees, making them particularly suitable for robust motion completion across diverse scenarios and novel activities.

2.1. Interpolation methods

Interpolation is one of the simplest techniques to estimate missing marker positions in MoCap data. These methods assume that the motion is continuous and that the missing data can be predicted based on the available neighboring markers. Various interpolation techniques have been explored in the literature, including linear interpolation [43,44], spline interpolation [45,46], Kalman filtering [44,47,48], and local linear models [49]. These methods are often effective for short-duration gaps and are computationally inexpensive. They typically rely on the assumption that motion continuity holds across neighboring markers, making them suitable for small gaps (lasting no more than a few seconds) or a limited number of frames. However, interpolation methods have limitations. They often require manual intervention to ensure the proper selection of neighboring markers, and their accuracy decreases for longer gaps or complex movements. Interpolation techniques also struggle to handle large, unpredictable gaps that arise due to occlusions or sensor failures. As a result, while these methods can be used in real time for minor data recovery tasks, they are not scalable for more complex or long-term MoCap data recovery [50].

2.2. Data-driven methods

A. database matching approaches. These approaches address the missing marker problem by using a database of similar motion sequences to reconstruct missing entries. These methods have demonstrated success in reconstructing high-quality human motion from sparse marker data. However, data-driven techniques require a pre-collected set of markers and pre-trained classifiers to predict missing values, which makes them dependent on having a well-established training dataset. A key limitation of these methods is their vulnerability to the “out-of-sample” problem, where the motion data diverges from the pre-trained models [51,52]. This restricts their flexibility and makes them less applicable in real-world scenarios where motion data can vary widely.

B. skeleton-driven methods. Skeleton-driven approaches use kinematic or bone-length constraints to guide the recovery of missing markers [53–55]. These methods rely on the accurate placement of markers on the human body. However, minor variations in marker placement between different subjects can significantly affect the performance and generalization of these approaches. Although these methods can be effective when marker placement is precise, their applicability in more dynamic and uncontrolled environments remains limited.

C. neural network–based methods. In recent years, deep learning has been increasingly applied to the motion capture data completion problem. Neural architectures address completion by learning motion priors directly from data rather than relying on handcrafted rank or smoothness constraints. Early work by Kucherenko et al. [56] demonstrated that recurrent and feed-forward networks can reconstruct missing markers by capturing temporal continuity, achieving superior performance over interpolation in long-gap scenarios but requiring large amounts of training data. Bidirectional recurrent autoencoders further improved robustness by propagating information from both past and future frames, while U-net–inspired Bi-LSTM architectures have been introduced to address long-term occlusions using adaptive loss functions [57].

More advanced approaches have incorporated attention mechanisms and generative modeling. In [58], authors proposed a bi-directional attention network that selectively emphasizes informative temporal contexts. Generative adversarial networks (GANs), where a convolutional recurrent generator reconstructs occluded skeletons and an LSTM-based discriminator enforces temporal consistency, are introduced in [59]. Graph neural networks (GNNs) explicitly represent the skeleton as a spatio-temporal graph, with joints as nodes and bones as edges.

A temporal-structural awareness GNN has been introduced in [60]; it captures both connectivity and motion dynamics to achieve state-of-the-art completion accuracy. In [61], the authors proposed a denoising graph autoencoder that treats missing joints as structured noise and reconstructs them using Laplacian smoothing and sharpening, improving robustness under occlusion and noisy capture conditions.

Despite these advances, neural network approaches face several limitations. They typically require large, diverse, and clean training datasets to generalize effectively, while low-rank optimization methods can be applied directly to corrupted input without pretraining. Their performance is also sensitive to dataset bias, often degrading when test motions deviate from training distributions. Furthermore, complex recurrent, attention-based, or graph models introduce substantial computational overhead, which may hinder real-time deployment.

2.3. Sparse representation methods

Sparse representation techniques assume that incomplete MoCap data can be sparsely represented as a linear combination of a few poses in an appropriate basis or dictionary [62–64]. These methods focus on finding the sparsest representation of motion data within a learned dictionary, where the reconstruction process is driven by mathematical optimization of sparsity constraints rather than similarity retrieval. However, they require extensive training data and dictionary learning, which can limit their flexibility in different scenarios. This requirement for training data restricts the applicability of these methods, especially in dynamic environments where data may not conform to predefined models.

2.4. Low-rank methods

2.4.1. PCA-based approaches

Principal component analysis (PCA)-based methods are widely used to address the missing marker problem by capturing the dominant components of human motion [49,65–67]. These methods decompose motion data into orthogonal basis vectors, also known as principal components, and reconstruct missing markers by projecting available data onto the lower-dimensional subspace defined by the most significant components. These methods exploit linear or nonlinear correlations in the motion matrix to estimate missing values based on the PCA-based methods can be classified into three primary reconstruction methodologies based on their core algorithmic strategies.

Basic PCA. A dual-PCA coordinate transformation approach was introduced in [68] that exploits correlation patterns between body segments in human motion data. The method constructs two principal component spaces: one from complete frames and another from the same frames with missing marker coordinates set to zero. Coordinate transformations between these spaces enable reconstruction of missing markers, with spatial weighting applied to neighboring markers for improved accuracy. While computationally efficient and requiring no external datasets, this approach is limited to single missing markers and exhibits degraded performance with multiple simultaneous gaps or non-repetitive movement patterns. The work in [66] extended eigen-space methods by incorporating training-based learning through precomputed motion-specific principal components from representative datasets. Rather than relying solely on current test sequences, their PCA-based algorithms leverage precomputed eigen-spaces from training samples to enable robust recovery through three distinct approaches: filling gaps in marker trajectories, recovering entire missing time frames within motion sequences, and handling complex motion patterns with unpredictable or non-repetitive movement characteristics.

Multi-scale PCA. Authors in [49] developed a hierarchical approach that combines global and local PCA modeling. The method first segments training motion sequences using probabilistic PCA, constructs a hierarchy of local linear models through divisive clustering, and utilizes random forest classifiers to identify the most appropriate local model for each frame during estimation. This piecewise linear framework enables robust recovery of missing markers even during extended occlusion periods, handling scenarios where traditional interpolation methods become ineffective, particularly when markers are missing at sequence boundaries or when significant portions of markers are absent simultaneously. This approach effectively handles diverse motion types and arbitrary missing marker sets but requires extensive training data and careful parameter tuning.

Weighted PCA. The study in [65] introduced a PCA-based method that exploits marker correlations through adaptive weighting strategies based on spatial proximity between markers. The method employs a dual PCA approach, performing principal component analysis on both complete data and gap-filled matrices to create transformation bases that reflect the kinematic chain structure of human bodies. Higher weights are assigned to biomechanically related markers using Gaussian functions. The authors implemented two reconstruction strategies for handling multiple gaps: simultaneous reconstruction which processes all corrupted trajectories together using only complete time frames, and consecutive reconstruction which selectively includes or excludes corrupted trajectories based on proximity criteria.

2.4.2. Optimization-based low-rank methods

This paper provides a comprehensive review of motion capture data recovery methods that explicitly exploit the low-rank structure through optimization-based formulations. These approaches differ fundamentally from other paradigms. Interpolation methods, such as linear or spline interpolation, operate on the principle of local estimation, using neighboring known values to predict missing entries under the assumption of smooth transitions. Interpolation approaches are inherently limited to short gaps and perform poorly when motion characteristics vary over longer temporal windows. Data-driven methods, while more sophisticated, rely on pre-collected motion databases or learned priors, which makes them powerful for known motion types but vulnerable to the out-of-sample problem when novel movements are encountered. PCA-based methods similarly depend on pre-learned subspaces from complete training data, which constrains their applicability to motions well represented in the training set (Table 1).

In contrast, optimization-based low-rank methods formulate motion completion directly as a constrained optimization problem by explicitly leveraging the inherent low-dimensional structure of human motion [69–77]. Rather than relying on local predictions or external databases, these methods recover missing data by minimizing an objective function subject to low-rank constraints. This framework naturally accommodates biomechanical and kinematic priors, provides theoretical recovery guarantees under certain conditions, and avoids the dependency on large, clean training datasets. Since direct rank minimization is non-convex and computationally intractable, convex relaxations and regularization techniques such as nuclear norm minimization, Schatten-p norms, and matrix factorization are employed. These optimization-based approaches represent the state-of-the-art in motion capture data completion and provide a principled foundation for integrating domain-specific knowledge about human motion.

3. Background and theoretical foundations

Low-rank property is a fundamental concept in various fields, including computer vision, dimensionality reduction, data analysis, and machine learning [78,78–87]. It plays a crucial role in reducing the complexity and noise in the data while preserving relevant information. The rank of a matrix refers to the maximum number of linearly independent columns/rows of that matrix. Mathematically, a data matrix is said to have a low-rank property when its rank is significantly smaller than its dimensions. This means that the data can be approximated by a matrix of a lower rank than the original data. The low-rank structure is adept

Table 1
Classification of low-rank approaches for motion capture data completion.

Reference	Low-Rank prior	Kinematic constraints	Optimization	Additional priors
Nuclear norm methods				
([69], 2011)	Nuclear norm	None	SVT	None
([90], 2013)	Nuclear norm	Motion trajectory	SVT	None
([73], 2014)	Nuclear norm	Smoothness	ALM	None
([75], 2015)	Nuclear norm	Bone length	SVT	None
([72], 2017)	Truncated nuclear norm	Smoothness	ALM	None
([109], 2019)	Nuclear norm	Isometry & smoothness	ALM	Sparsity
([71], 2020)	Nuclear norm	Frequency smoothness	ALM	Sparsity
([107], 2022)	Nuclear norm	Smoothness	FPPA	Sparsity
([91], 2024)	Approximate nuclear norm	Bone length & Smoothness	ADMM	Mixed corruption handling
([108], 2025)	Nuclear norm	Smoothness	ADMM	Hierarchical fusion
Schatten p-norm methods				
([70], 2018)	Truncated Schatten p-norm	Bone length & smoothness	ADMM	None
([74], 2019)	Weighted Schatten p	Bone length & smoothness	ADMM	Nonlocal similarity
([111], 2022)	Powered Schatten p-norm	Temporal smoothness	ADMM	Discrete subspace clustering
([108], 2025)	Schatten p-norm	Smoothness	ADMM	Hierarchical fusion
Other low-rank methods				
([76], 2015)	Matrix factorization	None	Inexact ALM	Hierarchical & Nonnegativity
([110], 2018)	Nonlinear low-rank	Bone length & smoothness	ALM	Kernel learning
Tensor decomposition methods				
([77], 2024)	CP decomposition	Smoothness	BCD	Sparsity
([112], 2025)	Tucker decomposition	Temporal continuity	Proximal BCD	None

Abbreviations: SVT = singular value thresholding; ALM = augmented Lagrange multiplier; ADMM = alternating direction method of multipliers; BCD = block coordinate descent; FPPA = fixed point proximal algorithm.

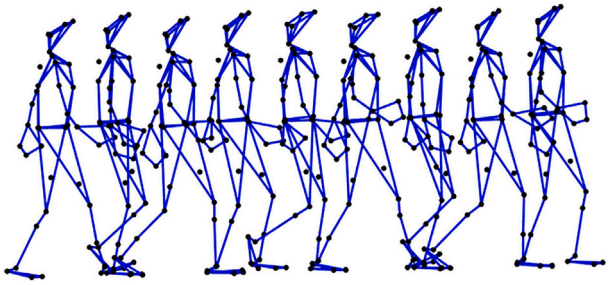

Fig. 2. Walking skeleton frames of walking sequence 07_01.c3d from CMU database.

at capturing the patterns present in the data. Thus, it has become a key to addressing the problem of reconstructing missing data [88,89].

One significant application of low-rank matrix completion is the gap-filling problem caused by missing markers in human motion capture. The low-rank property for human motion recovery can be justified by the observed similarity between adjacent frames in motion capture data, e.g., Fig. 2. A key trait of human motion is that the entire motion is characterized by the movements of joints interconnected by rigid and inflexible bones. Thus, the motion of one joint is inherently correlated with that of others. To check whether a given motion matrix $\mathbf{X}$ has a low-rank structure, we analyze the spectrum (singular values) of $\mathbf{X}$ or $\mathbf{X}^T\mathbf{X}$. We demonstrate in Fig. 3 the logarithmic scale of singular values for four[1] distinct C3D files from the CMU MoCap database.[2] We can observe that the motion matrices exhibit a decay in the singular values, indicating that the motion matrices possess a low-rank characteristic.

3.1. Basic notations and definitions

Tensors are denoted using Euler script (e.g., $\mathcal{X}$), matrices are represented by bold uppercase letters (e.g., $\mathbf{X}$), vectors by bold lowercase letters (e.g., $\mathbf{x}$), and scalars by standard lowercase letters (e.g., x). $\mathbf{x}_i$ refers to the i-th column of the matrix $\mathbf{X}$. The order of a tensor corresponds to the number of its dimensions. More generally, an N-order tensor $\mathcal{X}$ with dimensions $I_1 \times I_2 \times \cdots \times I_N$ is an N-dimensional array in $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The indexing conventions are as follows: the element at position $(i_1, i_2, \ldots, i_N)$ in a tensor $\mathcal{X}$ is written as $\mathcal{X}_{i_1,i_2,\ldots,i_N}$, and the matrix entry at (i_1, i_2) is represented as $\mathbf{X}_{i_1,i_2}$. For a positive integer N, we use the notation $[N] := \{1, 2, \ldots, N\}$. The symbol "$\circ$" represents the vector's outer product. The n-mode tensor-matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{A}$ and is of the size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. The mode-$n$ unfolding of $\mathcal{X}$ is defined as the matrix $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times M}$, with $M = \prod_{j \neq n}^{N} I_j$, obtained by converting a tensor to a matrix according to the mode-n where $\mathcal{X}_{i_1 \ldots i_N}$ corresponds to $\mathbf{X}_{i_n,s}$, with

$$s = 1 + \sum_{j \neq n}^{N} (i_j - 1)\, S_j \quad \text{and} \quad S_j = \prod_{k \neq n}^{j-1} I_k.$$

3.2. Matrix representation of mocap data

Motion capture data is recorded as a series of motion frames that represent highly articulated movements. A captured motion is composed of a sequence of frames (poses), and each frame is characterized by the positions of markers placed on specific body parts or joints. Each joint has 3D coordinates $[x, y, z]$.

Frame-based representation. The captured motion sequence can be represented as an $m \times n$ matrix M where m corresponds to the total number of frames in the motion capture sequence, $n = 3p$ represents the number of position coordinates for all markers or joints, and thus each row of the matrix M corresponds to a frame in the sequence:

$$\mathbf{M} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n], \tag{1}$$

where each frame $\mathbf{f}_i$ for $i = 1, \ldots, n$ is defined either as:

$$\mathbf{f}_i = \left[x_{i,1} y_{i,1} z_{i,1} x_{i,2} y_{i,2} z_{i,2} \ldots x_{i,p} y_{i,p} z_{i,p}\right]^T, \tag{2}$$

[1] 07_01.c3d, 85_02.c3d, 85_12.c3d, and 135_02.c3d.
[2] http://mocap.cs.cmu.edu/

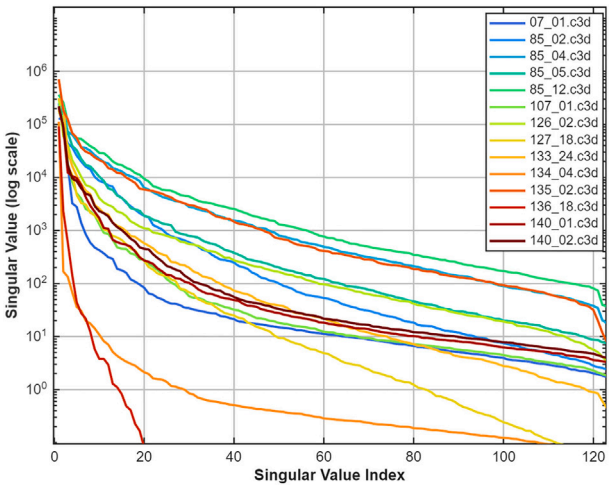

Fig. 3. Singular values (log scale) of motion capture sequences from multiple subjects and activity categories from CMU dataset. The analyzed motions include: **Subject #7:** walk; **Subject #85:** jumps, flips, and breakdance (02: *JumpTwist*, 04: *FancyFootWork*, 05:*HandStandKicks*, 12: *LongSequenceGood*); **Subject #107:** walking with obstacles; **Subject #126:** swimming (02: *BackStroke*); **Subject #127:** action–adventure motions, running, jumping, ducking, rolling (17: *RunStopRun*); **Subject #133:** baby–styled walk (24: *WalkZigZag*); **Subject #134:** skateboard motion (04: *Motorcycle*); **Subject #135:** martial arts walk (02: *Empi*); **Subject #136:** stylized or "weird" walks (18: *Flamingo*); **Subject #140:** getting up from the ground (01–02: *Run*). These sequences cover diverse motion styles, illustrating the low-rank structure of motion capture data across both periodic and complex non-periodic movements.

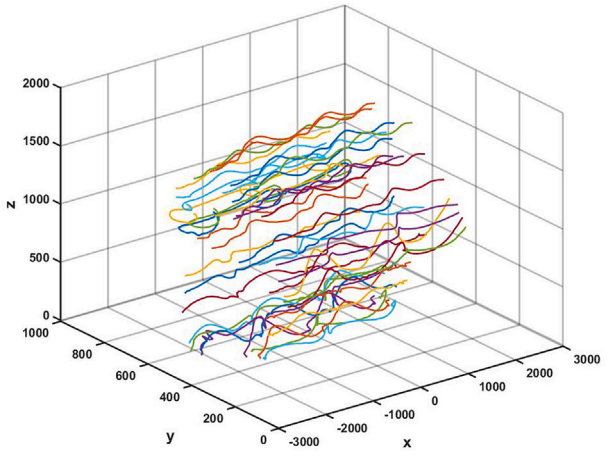

Fig. 4. 3D marker trajectories of the walking motion sequence 07_01.c3d (316 frames) from the CMU dataset.

or as:

$$\mathbf{f}_i = [x_{i,1}x_{i,2}\dots x_{i,p}y_{i,1}y_{i,2}\dots y_{i,p}z_{i,1}z_{i,2}\dots z_{i,p}]^T. \tag{3}$$

As shown in Fig. 4, the trajectories of all the joints are inherently correlated. Due to the high data acquisition speeds of the Mocap system, the frames exhibit mutual consistency in the motion data. Thus, the frame-based representation of Mocap data reveals the redundancies in marker positions across frames, which intuitively implies the low-rank property.

Trajectory-based representation. To further enhance the reliability of the low-rank prior, a trajectory-based representation has been introduced in [90]. The trajectories of different joints are correlated over a short time window (see Fig. 6). Thus, the motion data is structured into a trajectory-based representation that takes advantage of the repetitive patterns inherent in human motion in small periods. Therefore, the motion data is arranged as a sequence of trajectory segments:

$$\mathbf{M_t} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_k], \quad \text{where} \quad k = d \times n/l$$

where l is the length of a trajectory segment and each $\mathbf{t}$ is the trajectory segment of a joint from frame j to frame $j + l - 1$ given as

$$\mathbf{t}_i = \left[x_{i,j}x_{i,j+1}\dots x_{i,j+l-1}y_{i,j}y_{i,j+1}\dots y_{i,j+l-1}z_{i,j}z_{i,j+1}\dots z_{i,j+l-1}\right]^T.$$

3.3. Tensor representation

MoCap data can naturally be modeled as a third-order tensor $\mathcal{M} \in \mathbb{R}^{m\times p\times 3}$ where the dimensions correspond to time (frames), joints (markers), and spatial coordinates (3D coordinates), respectively, as demonstrated by the 3D trajectories in Fig. 4.

$$\mathcal{M} = [\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z],$$

where $\mathbf{M}_x$, $\mathbf{M}_y$, and $\mathbf{M}_z \in \mathbb{R}^{m\times p}$ represent the components of the MoCap data along the x, y, and z axes, respectively.

3.4. Problem statement

In motion capture systems, markers are placed on an object to track its movement. However, these markers can become temporarily invisible to cameras during recording sessions due to occlusions, self-occlusions, poor lighting conditions, or technical failures. When a marker disappears from the camera's field of view, its 3D position data becomes unavailable, creating gaps in the trajectories. These gaps significantly degrade the quality of motion data and pose challenges for practical applications that require continuous and complete motion information.

When a marker is occluded, the three coordinates (x, y, z) are typically missing simultaneously, creating gaps in temporal blocks (consecutive frames) rather than as isolated values. Thus, the challenge of gap-filling differs significantly from the conventional missing data problem studied in other applications such as image processing as illustrated in Fig. 5. Recovering these missing markers presents significant challenges due to the complexity of human movement and the intricate interdependencies between joints. These challenges increase as gaps become longer or when multiple markers disappear concurrently. Several key factors critically influence the performance of gap-filling methods:

- *Gap length and duration:* The number of consecutive missing frames per marker. This can also be measured as the time duration based on the Mocap sampling frequency (missing frames = duration × frequency). Longer gaps are typically more challenging to fill accurately.
- *Number of missing markers:* The total number of missing markers within a sequence directly impacts completion accuracy. As this number increases, the available spatial-temporal information decreases, creating a more challenging reconstruction problem.
- *Gap distribution and multiplicity:* The pattern and frequency of gaps significantly influence completion effectiveness. Randomly distributed gaps are more challenging to predict than those following systematic patterns. Additionally, when multiple gaps co-occur within a single marker trajectory, the complexity increases substantially, limiting how effectively spatial-temporal correlations can be exploited for accurate trajectory reconstruction.
- *Motion complexity:* The complexity of motion significantly affects completion performance. Highly dynamic or complex movements (e.g., dancing, boxing) present greater challenges than simpler, repetitive motions (e.g., walking). Complex motions typically involve unpredictable trajectories, variable speeds, and intricate coordination patterns that are more difficult to predict accurately.

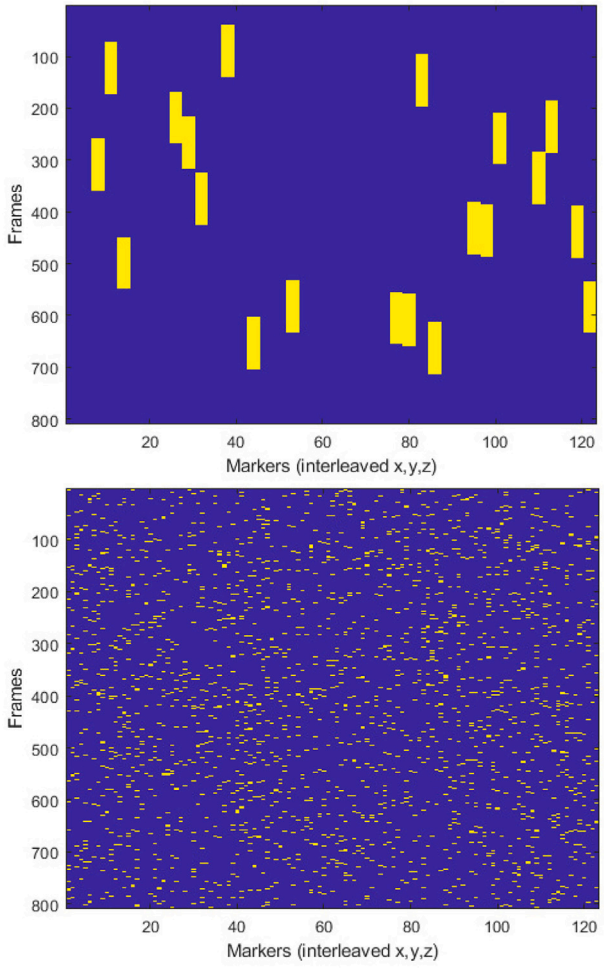

Fig. 5. Illustration of different missing data patterns. (a) Structured gaps typical in MoCap data (e.g., continuous marker occlusions). (b) Randomly missing entries, as commonly modeled in image or pixel-based data.

The MoCap data completion problem can be formulated using either matrix or tensor representations. In the matrix setting, the motion data $\mathbf{M} \in \mathbb{R}^{m\times n}$ typically represents m time frames and n marker coordinates, with missing entries. Let Ω be the index set of observed entries, where the values of elements $\mathbf{M}_{ij}, (i,j) \in \Omega$ are known. The low-rank matrix completion problem can then be formally expressed as:

$$\begin{aligned} &\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \quad \operatorname{rank}(\mathbf{X}), \\ &\text{s.t} \;\; \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}), \end{aligned} \tag{4}$$

where $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^{m\times n}$ is the orthogonal projection onto the subspace of matrices that vanish outside of Ω, $(i,j) \in \Omega$ if and only if $\mathbf{M}_{ij}$ is observed. $\mathcal{P}_\Omega(\mathbf{X})$ is defined by

$$\mathcal{P}_\Omega(\mathbf{X}) = \begin{cases} \mathbf{X}_{i,j} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Alternatively, this projection can be expressed as $\mathcal{P}_\Omega(\mathbf{X}) = \mathcal{H} \circ \mathbf{X}$. Where $\mathcal{H} \in \{0,1\}^{m\times n}$ is a binary matrix that refers to the mask matrix defined as

$$\mathcal{H} = \begin{cases} 1 & \text{if } \mathbf{M}_{i,j} \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

While the matrix formulation has been widely studied, MoCap data inherently possesses a multi-dimensional structure that can be better preserved using tensor representations. In this case, the motion data can be represented as a 3-way tensor $\mathcal{T} \in \mathbb{R}^{m\times n\times p}$, where the dimensions might represent time, markers, and 3D coordinates. The core objective remains similar to matrix completion: recovering missing entries while maintaining the underlying low-dimensional structure of the data as follows

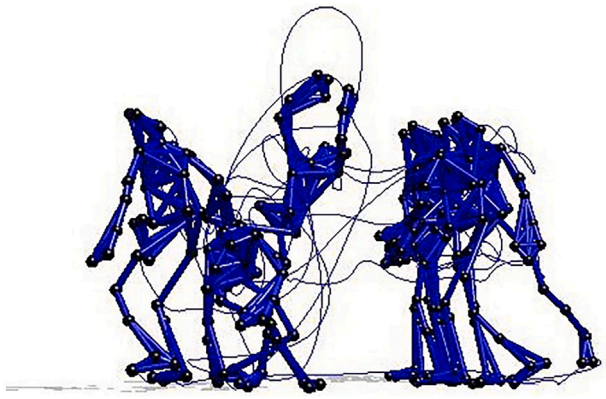

Fig. 6. Trajectories of selected markers across consecutive frames of a motion sequence.

$$\begin{aligned} &\min_{\mathcal{X}\in\mathbb{R}^{m\times n\times p}} \quad \operatorname{rank}(\mathcal{X}), \\ &\text{s.t} \;\; \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \end{aligned} \tag{7}$$

where $\mathcal{P}_\Omega : \mathbb{R}^{m\times n\times p} \longrightarrow \mathbb{R}^{m\times n\times p}$ is the orthogonal projection operator defined as:

$$\mathcal{P}_\Omega(\mathcal{X}) = \begin{cases} \mathcal{X}_{i_1,i_2,i_3} & \text{if } (i_1,i_2,i_3) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Both matrix and tensor completion problems aim to find the lowest-rank structure that matches the observed entries. However, these problems are generally NP-hard, leading to various computational approaches and relaxations.

4. Kinematic priors in MoCap completion

Human motion follows inherent biomechanical rules and patterns that can be used to improve the accuracy of the completion problem. These kinematic constraints serve as side information that can be integrated into the low-rank completion framework. The integration of motion priors acts as a regularization mechanism, guiding the low-rank optimization towards solutions that align with the expected characteristics of human motion. These priors restrict the solution space to physically valid motions that preserve natural motion characteristics, leading to more realistic and biomechanically plausible results.

4.1. Skeleton constraints

The skeleton constraints ensure that the completion process follows the skeleton's structure as shown in Fig. 7. This helps prevent unrealistic features such as incorrect bone lengths in the recovered results. Here, we discuss the three skeleton constraints used in the MoCap completion problem.

A. bone length constraint. This constraint ensures that the distances between adjacent joints are preserved during the recovery process. Typically, an articulated skeleton is represented as a tree structure, where each node corresponds to a skeletal joint, and the edges connecting nodes represent bones. For each missing joint, the set $\mathcal{E}$ contains its adjacent edges. Each edge $e_i \in \mathcal{E}, i = 1,\ldots,p$ has a length d_i and a

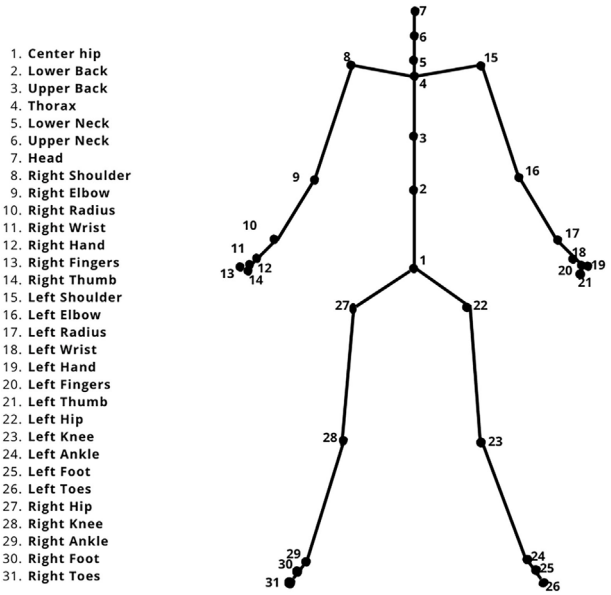

Fig. 7. Skeletal representation of the human body with 31 labeled joints.

connected joint pair $(\alpha, \beta)_i$. Therefore, for each e_i, a matrix $\mathbf{C}_i \in \mathbb{R}^{3\times n}$ is defined to extract the inter-joint distance of $(\alpha, \beta)_i$ from $\mathbf{x}$ i.e.,

$$\mathbf{C}_i\mathbf{x} = [x\ y\ z]_\alpha^T - [x\ y\ z]_\beta^T,$$

where $\mathbf{x} = Vec(\mathbf{X})$ and Vec maps a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ to a vector $\mathbf{x} \in \mathbb{R}^n$, with $n = m \times n$. For example, the matrix $\mathbf{C}_i$ can be given as:

$$\mathbf{C}_i = [0, \dots, \mathbf{I}_{\alpha_i}, 0, 0, -\mathbf{I}_{\beta_i}, 0, \dots, 0],$$

where $\mathbf{I}_{\alpha_i}$, and $\mathbf{I}_{\beta_i}$ are the 3×3 identity matrices. Thus, to preserve the skeleton constraints in the completion algorithm, the bone length constraint

$$\|\mathbf{C}_i\mathbf{x}\|_2 \le d_i, \text{ for } i = 1, \dots, p, \tag{9}$$

could be integrated into the minimization function of the completion problem. The distance d_i is computed by averaging the inter-joint distances of bone e_i across all frames where both connected joints are observable.

B. total bone length constraint. The fundamental principle underlying skeletal motion capture is that bone lengths remain constant throughout movement, as the human skeleton maintains its rigid structure. Let $\mathbf{f}_i$ be the ith frame:

$$\begin{aligned}\mathbf{f}_i &= [x_{i,1}\, y_{i,1}\, z_{i,1}\, x_{i,2}\, y_{i,2}\, z_{i,2} \cdots x_{i,d}\, y_{i,d}\, z_{i,d}]^T,\\ &= \left[\mathbf{c}_{i,1}^T, \mathbf{c}_{i,2}^T, \dots, \mathbf{c}_{i,j}^T, \dots, \mathbf{c}_{id}^T\right]^T,\end{aligned}$$

where $\mathbf{c}_{i,j} = [x_{i,j}, y_{i,j}, z_{i,j}]$ is the 3D coordinates of the jth joint in the ith frame. The relative coordinates $\mathbf{c}_{i,j}^r$ of the j-th joint in the i-th frame are defined as:

$$\mathbf{c}_{i,j}^r = \frac{\mathbf{c}_{i,j} - \mathbf{c}_{i,p(j)}}{l_j},$$

where $\mathbf{c}_{i,p(j)}$ is the coordinates of the parent joint of joint j in frame i, and l_j is the bone length connecting joint j to its parent joint. The bone-length error function $K\left(\mathbf{f}_i\right)$ of the i-th frame in $\mathbf{X}$ is defined as:

$$K\left(\mathbf{f}_i\right) = \sum_{j=1}^{m}\left(\left(\mathbf{c}_{i,j}^r\right)^T\mathbf{c}_{i,j}^r - 1\right)^2.$$

Since the bone-length error of each frame in $\mathbf{X}$ is independent, the total bone-length error can be formulated as:

$$g(\mathbf{X}) = \sum_{i=1}^{n} K\left(\mathbf{f}_i\right), \tag{10}$$

where $g(\mathbf{X}) = 0$ if and only if no bone length is wrong in the recovered results. An equivalent formulation using absolute bone lengths is called the isometry constraint:

$$\Theta_{\text{iso}}(\mathbf{X}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\left\|\mathbf{c}_{i,j} - \mathbf{c}_{i,p(j)}\right\|_2^2 - l_j^2\right)^2. \tag{11}$$

Both the relative and absolute formulations enforce the same underlying constraint: preserving the rigid skeletal structure by maintaining constant bone lengths throughout the motion sequence.

An alternative formulation of bone length preservation has been proposed in [91], where the constraint considers all connected joint pairs in the skeletal structure:

$$\mathbf{C}_{ph}(\mathbf{X}) = \sum_{i=1}^{n}\sum_{e_{j,k}\in\mathcal{E}}\left(\left\|\mathbf{c}_{i,j} - \mathbf{c}_{i,k}\right\|_2^2 - l_{j,k}^2\right)^2, \tag{12}$$

where $\mathcal{E}$ represents the set of all bones connecting joint pairs (j, k) in the skeletal structure. Unlike the isometry constraint in Eq. (11) which enforces bone lengths only between joints and their hierarchical parents, this formulation encompasses all anatomically connected joint pairs, potentially including sibling joints, grandparent–grandchild relationships, or other structural connections beyond the strict parent-child hierarchy.

4.2. Smoothness prior

In human movement, transitions between different positions or poses are typically continuous. Moreover, the trajectory of a moving body part rarely involves sudden or jerky movements (see Fig. 4). Thus, human motion data often has a smooth structure. This smoothness refers to the continuous and gradual transitions in the position, velocity, and acceleration of body parts during movement. Therefore, methods that capture and model the smoothness property provide a more natural reconstruction of human motion.

A. difference operators. The smoothness of the MoCap matrix $\mathbf{X}$ in the temporal direction can be expressed using difference operators of various orders. These operators enforce trajectory smoothness by minimizing the differences between adjacent data points, where smaller differences imply smoother transitions.

First-order difference operator. The first-order difference operator, also known as the Total Variation (TV) regularizer, enforces smoothness by penalizing large gradients between consecutive time points while preserving important discontinuities. For a given vector $\mathbf{x} \in \mathbb{R}^m$, the TV operator is defined as

$$\|\mathbf{x}\|_{\text{TV}} = \left(\sum_{i=1}^{m-1}|x_i - x_{i+1}|^2\right)^{\frac{1}{2}} = \|\mathbf{D}_{tv}\mathbf{x}\|_2,$$

where $\mathbf{D}_{tv}$ is the first-order difference matrix:

$$\mathbf{D}_{tv} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}.$$

Second-order difference operator. The second-order difference operator provides enhanced smoothness regularization by penalizing the curvature (second derivative) of trajectories, which corresponds to constraining acceleration rather than velocity. This approach enforces smoother transitions that are more consistent with natural human motion patterns. The operator is defined as

$$\Theta(\mathbf{X}) = \|\mathbf{X}\mathbf{D}_h^T\|_F^2, \tag{13}$$

where the matrix $\mathbf{D}_h$ is a tridiagonal square matrix defined by

$$(\mathbf{D}_h)_{i,j} = \begin{cases} \frac{2}{h_{i-1}(h_{i-1}+h_i)} & \text{if } j = i-1, \\ \frac{-2}{h_{i-1}h_i} & \text{if } j = i, \\ \frac{2}{h_i(h_{i-1}+h_i)} & \text{if } j = i+1, \\ 0 & \text{otherwise.} \end{cases}$$

for $\forall i, 2 \leqslant i \leqslant n-1$, where n is the number of elements in $\mathbf{X}_{r,:}$ and h_i represents the step between consecutive data points $\mathbf{X}_{r,i}$ and $\mathbf{X}_{r,i+1}$. Assuming repeating border elements $(\mathbf{X}_{r,0} = \mathbf{X}_{r,A}$ and $\mathbf{X}_{r,n} = \mathbf{X}_{r,n+1})$ gives

$$-(\mathbf{D}_h)_{1,1} = (\mathbf{D}_h)_{1,2} = \frac{1}{h_1^2}, \quad \text{and} \quad (\mathbf{D}_h)_{n,n-1} = -(\mathbf{D}_h)_{n,n} = \frac{1}{h_{n-1}^2}.$$

When assuming that the data is equally spaced with a constant interval of $h_i = 1, \forall i$, the divided difference matrix $\mathbf{D}_h$ becomes

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}.$$

The matrix $\mathbf{D}$ focuses on the immediate neighborhood of each data point, thus providing a local analysis of temporal smoothness. By considering boundary conditions (repeating border elements), the matrix $\mathbf{D}$ accounts for the stability at the edges of the data sequence. This ensures that the analysis of smoothness is consistent and accurate even near the boundaries of the data.

Smoothness priors can be enforced through first- or second-order difference operators. Both operators focus on the immediate neighborhood of each data point, providing local analysis of temporal smoothness. The boundary conditions ensure consistent and accurate smoothness analysis even at the edges of the data sequence. First-order operators constrain velocity changes and are effective for motions with relatively constant speeds or gradual transitions. Second-order operators constrain accelerations and are particularly suitable for repetitive or cyclical activities, where biomechanical principles support smooth acceleration patterns. In contrast, abrupt or impact-heavy movements (e.g., sudden direction changes, collisions, or contact interactions) often violate these assumptions. In such cases, overly strict smoothness enforcement may distort the motion by artificially damping rapid variations or suppressing natural discontinuities. Therefore, the choice between first- and second-order smoothness should be guided by the motion characteristics, with adaptive formulations offering a potential solution for handling mixed or unpredictable dynamics.

B. wavelets transform. Wavelets in the context of MoCap data analysis serve to analyze and characterize temporal and spatial features of human motion. The Discrete Wavelet Transform (DWT) [92–94] decomposes the motion data into different scales, offering a multiresolution view. By decomposing the motion data matrix $\mathbf{X}$ using a wavelet basis matrix $\mathbf{W}$, the transform enables a comprehensive analysis of motion smoothness through the smoothness assumption defined by the wavelet transformation:

$$\Theta(\mathbf{X}) = \|\mathbf{W}\mathbf{X}\|_q. \tag{14}$$

Here, $\|\cdot\|_q$ represents the l_q-norm (typically l_1 or l_2), which allows for flexible feature extraction and noise reduction. The wavelet basis matrix $\mathbf{W}$ is defined as:

$$\mathbf{W} = \begin{bmatrix} \psi_{s(1),k(1)}(1) & \psi_{s(1),k(1)}(2) & \cdots & \psi_{s(1),k(1)}(n) \\ \psi_{s(2),k(2)}(1) & \psi_{s(2),k(2)}(2) & \cdots & \psi_{s(2),k(2)}(n) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_{s(n),k(n)}(1) & \psi_{s(n),k(n)}(2) & \cdots & \psi_{s(n),k(n)}(n) \end{bmatrix}.$$

The elements of matrix $\mathbf{W}$ are derived from wavelet basis function $\psi_{s,k}(j)$ is defined as:

$$\psi_{s,k}(j) = \frac{1}{\sqrt{2^s}} \psi\left(\frac{j - k \cdot 2^s}{2^s}\right),$$

where $\psi(t)$ is the mother wavelet function, $s \in \mathbb{Z}$ represents the scale parameter controlling dilation, and $k \in \mathbb{Z}$ represents the translation parameter controlling position.

The wavelet coefficients resulting from the transform $\mathbf{WX}$ provide a time–frequency representation where the larger coefficients correspond to significant motion features at specific scales and temporal locations. This multiresolution property makes wavelets particularly effective for analyzing hierarchical motion structures.

C.Fast Fourier transform (FFT). The FFT is a popular tool for analyzing the frequency of signals and understanding how a subject evolves. FFT decomposes signals into their frequency components [94]. In cases of periodic signals, the FFT decomposes the signal into its constituent low-frequency components. By decomposing signals into their fundamental frequency components, the FFT provides insights into the periodic nature of movement. The mathematical representation of the FFT-based smoothness assumption is expressed as:

$$\Theta(\mathbf{X}) = \|\mathbf{F}\mathbf{X}\|_q, \tag{15}$$

where $\mathbf{F}$ is the FFT applied on $\mathbf{X}$, with $\mathbf{F}$ as a transformation matrix. Let $\mathbf{F}_{k,n}$ be the element in the k-th row and n-th column of the $N \times N$ matrix $\mathbf{F}$, with $k, n \in \{0, 1, 2, \ldots, N-1\}$.

$$\mathbf{F}_{k,n} = e^{-j2\pi kn/N}.$$

The matrix form is given in the form:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi\cdot 1/N} & \ldots & e^{-j2\pi\cdot(N-1)/N} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-j2\pi\cdot(N-1)/N} & \ldots & e^{-j2\pi\cdot(N-1)^2/N} \end{bmatrix}.$$

The FFT decomposes complex motion signals into low-frequency components, revealing the fundamental periodic patterns inherent in human movement while filtering out high-frequency noise.

5. Low rank priors

The problem of minimizing the rank of a matrix or tensor, as in Eqs. (4) and (7), is ill-posed and generally considered NP-hard. Convex relaxations of the rank function make the problem computationally feasible and provide theoretical guarantees for data recovery under certain conditions. Tensors, as higher-order generalizations of matrices, introduce additional complexity in rank definition and optimization strategies. Unlike in the matrix case, the definition of the tensor rank is not well established. Thus, various definitions of the tensor rank and their convex relaxations have been studied to characterize the low-rankness of tensors. The tensor rank definitions have been proposed based on different decomposition methods. These decompositions include the Canonical Polyadic (CP) decomposition [95], which breaks tensors into a sum of rank-one tensors; Tucker decomposition [96], which provides a hierarchical factorization; and the tensor train decomposition [97,98], which offers an efficient representation for high-dimensional tensors. These approaches decompose high-dimensional tensors into a sequence of lower-dimensional tensors, effectively simplifying the completion process by revealing underlying low-dimensional structures and patterns. This strategy addresses the computational challenges and provides a more interpretable approach to understanding multidimensional data.

5.1. Nuclear norm-based regularization

The use of low-rank matrix completion theory for human motion data processing was first introduced in [69]. The problem (4) was addressed by replacing the matrix rank with its nuclear norm, which is a convex relaxation as demonstrated in Theorem 1 from [99].

Theorem 1. *The convex envelope of the function $\phi(\mathbf{X}) = \mathrm{Rank}(\mathbf{X})$, on $C = \{\mathbf{X} \in \mathbb{R}^{m\times n} \mid \|\mathbf{X}\| \leq 1\}$, is $\phi_{env}(\mathbf{X}) = \|\mathbf{X}\|_*$, where $\|\mathbf{X}\|_* = \sum_i \sigma_i(\mathbf{X})$ is the nuclear norm and $\|\mathbf{X}\| = \sigma_1(\mathbf{X})$ denotes the spectral norm, which is the largest singular value of matrix* $\mathbf{X}$.

The theorem shows that a lower bound for the optimal solution of the rank minimization problem can be derived by solving the nuclear norm relaxation. The nuclear norm is the convex hull of the set of rank-one matrices with a spectral norm bounded by one. Furthermore, when the number of observed entries satisfies

$$m \geq Cn^{6/5} r \log n,$$

where C is a constant, r refers to the matrix rank, and n is the matrix dimension, exact matrix recovery can be achieved with high probability for matrices satisfying appropriate incoherence conditions; see Theorem 1.1 in [88]. However, this is not always achievable in practical scenarios. Moreover, the nuclear norm treats all singular values equally, ignoring the practical significance of singular values for obtaining the matrix information. Additionally, the nuclear norm commonly involves the overshrinking problem, which is likely to result in diminished performance when there is noise in the measurements.

5.1.1. Truncated nuclear norm

To overcome the limitations of the nuclear norm, the truncated nuclear norm (TNN) approach provides a solution to the problem of overshrinking by minimizing only the sum of smaller singular values while truncating the larger ones [100]. For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the truncated nuclear norm is defined as the sum of the $\min(m,n) - r$ smallest singular values, i.e.

$$\|\mathbf{X}\|_r = \sum_{i=r+1}^{\min(m,n)} \sigma_i(\mathbf{X}). \tag{16}$$

Since the values of the largest r nonzero singular values will not affect the rank of the matrix, the truncated nuclear norm leaves them free and focuses on minimizing the sum of the smallest $\min(m,n) - r$ singular values.

5.2. Schatten p-norm based regularization

To better bridge the gap between nuclear norm and real rank, the Schatten p-norm involves computing the sum of the p-th $(0 \leq p \leq 1)$ power of all singular values. The Schatten p-norm is defined as

$$\|\mathbf{X}\|_{S_p}^p = \sum_{i=1}^{\min(m,n)} \left(\sigma_i(\mathbf{X})\right)^p.$$

However, similar to the nuclear norm, the Schatten p-norm still deals with all singular values equally, while different singular values have different importance. Therefore, truncated Schatten p-norm and weighted Schatten p-norm have been introduced [101,102].

5.2.1. Truncated schatten-p norm

For a given matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the truncated Schatten p-norm, which is defined as the sum of the pth $(0 \leq p \leq 1)$ power of $\min(m,n) - r$ minimum singular values, can be given as:

$$\|\mathbf{X}\|_{r,S_p}^p = \sum_{i=r+1}^{\min(m,n)} \left(\sigma_i(\mathbf{X})\right)^p.$$

Compared with the nuclear norm, the truncated Schatten p-norm considers the physical interpretations of singular values and truncates the large singular values from shrinking so that the main component of a matrix is preserved [102]. Moreover, compared with the truncated nuclear norm, the adjustable parameter p, the truncated Schatten p-norm more flexible in dealing with different applications.

5.2.2. Weighted Schatten p-norm

The weighted Schatten p-norm tackles the overshrinking problem by assigning varying weights to singular values. Larger singular values receive smaller weights, whereas smaller singular values are assigned larger weights. The weighted Schatten p-norm of matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ is defined as

$$\|\mathbf{X}\|_{w,S_p}^p = \sum_{i=1}^{\min\{n,m\}} w_i \left(\sigma_i(\mathbf{X})\right)^p,$$

where $\mathbf{w} = \left[w_1, \ldots, w_{\min\{n,m\}}\right]$ is a non-negative vector, and σ_i is the i-th singular value of $\mathbf{X}$.

5.2.3. Powered Schatten p-norm

To further improve the approximation of the rank function, the t-th power of Schatten p-norm has been introduced in [40]:

$$\left(\|\mathbf{X}\|_{S_p}^p\right)^t = \left(\sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{X})^p\right)^t, \tag{17}$$

where $0 < p < 1$ and $t \geq 1$. The parameter t provides additional flexibility in controlling the rank approximation, with empirical studies suggesting optimal performance at $t = 2$ and $p = 0.5$ for motion capture applications [40].

5.3. Nonlinear low-rank prior

In situations where the data is subject to complex nonlinear changes, such as significant pose variations, diverse expressions, or varying illumination conditions, low-rank matrix completion may fail to recover the details of data with complex and diverse structures. To effectively cope with the nonlinear data, kernel-based methods have been introduced [103–105]. These models learn a combined low-rank kernel via multiple-kernel learning. The geodesic exponential kernel is used for the multiple kernel learning process. The geodesic exponential kernel is built upon the geodesic distance, which is particularly suitable for measuring distances between points on a Riemannian manifold. In this context, we denote the coordinates of two points on a 2-dimensional sphere as $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{y} \in \mathbb{R}^3$. The geodesic exponential kernel is formulated as:

$$k_\gamma(\mathbf{x},\mathbf{y}) = e^{\frac{-d_g(\mathbf{x},\mathbf{y})}{2\gamma^2}},$$

where γ is a width parameter, and d_g represents the geodesic distance between the points $\mathbf{x}$ and $\mathbf{y}$ and is defined as

$$d_g(\mathbf{x},\mathbf{y}) = \arccos\left(\mathbf{x}^T\mathbf{y}\right),$$

where arccos: $[-1,1] \to [0,\pi]$ is the usual inverse cosine function. The function $\Phi : \mathbf{f} \to \Phi(\mathbf{f})$ represents an implicit mapping defined on the kernel function $k_{\mathbf{A}}\left(\mathbf{f}_a, \mathbf{f}_b\right)$ and the inner product of two frames $\mathbf{f}_a$ and $\mathbf{f}_b$ is defined as

$$\left\langle \Phi\left(\mathbf{f}_a\right), \Phi\left(\mathbf{f}_b\right)\right\rangle = k_{\mathbf{A}}\left(\mathbf{f}_a, \mathbf{f}_b\right) = \sum_{j=1}^{d}\sum_{i=1}^{n_w} \alpha_{j,i}^2 k_{\gamma_i}\left(\mathbf{c}_{a,j}^n, \mathbf{c}_{b,j}^n\right),$$

$\mathbf{c}_{a,j}^n$ and $\mathbf{c}_{b,j}^n$ are respectively normalized positions of $\mathbf{c}_{a,j}$ and $\mathbf{c}_{b,j}$ defined as:

$$\mathbf{c}_{a,j}^n = \frac{\mathbf{c}_{a,j} - \mathbf{c}_{pa,j}}{\|\mathbf{c}_{a,j} - \mathbf{c}_{pi,j}\|_2}, \qquad \mathbf{c}_{b,j}^n = \frac{\mathbf{c}_{b,j} - \mathbf{c}_{pi,j}}{\|\mathbf{c}_{b,j} - \mathbf{c}_{pb,j}\|_2},$$

and $\mathbf{c}_{a,j}, \mathbf{c}_{b,j}$ are respectively the j-th joint in $\mathbf{f}_a$ and $\mathbf{f}_b$, n_w is the number of the width parameters γ_i which correspond to different geodesic

exponential kernels $k_{\gamma_i}, \alpha_{j,i}$ is the weight of k_{γ_i}. The weights $\alpha_{j,i}$ can be arranged in the following weight matrix $\mathbf{A} \in \mathbb{R}^{d \times n_a}$

$$\mathbf{A} = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1n_a} \\ \vdots & \ddots & \vdots \\ \alpha_{d1} & \cdots & \alpha_{dn_a} \end{pmatrix}. \tag{18}$$

Let $\mathbf{Q} = \left[\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_d}\right]$ be a training set composed of a large number of frames of multiple types of motion sequences, where n_q is the number of frames in $\mathbf{Q}$. $\mathbf{Q}_j \in \mathbb{R}^{3 \times n_q}$ corresponds to all the j-th joints in $\mathbf{Q}$. Multiple kernel learning aims to minimize the rank of $\Phi(\mathbf{Q})$. Thus, using Schatten p-norm instead of the rank of a matrix, the $\operatorname{rank}(\Phi(\mathbf{Q}))$ can be formulated as

$$\begin{aligned} \operatorname{rank}(\Phi(\mathbf{Q})) &\approx \left(\operatorname{Tr}\left(\left((\Phi(\mathbf{Q}))^T \Phi(\mathbf{Q})\right)^{\frac{p}{2}}\right)\right)^{\frac{1}{p}}, \\ &= \left(\operatorname{Tr}\left((K_r(\mathbf{Q}))^{\frac{p}{2}}\right)\right)^{\frac{1}{p}}, \end{aligned} \tag{19}$$

where $K_r(\mathbf{Q})$ is the kernel matrix of $\mathbf{Q}$ defined on $k_{\mathbf{A}}\left(\mathbf{f}_a, \mathbf{f}_b\right)$.

5.4. Non-negative matrix factorization

Algorithms based on singular value decomposition, such as the nuclear norm, are computationally costly, especially when the underlying matrices are high-dimensional. The low-rank matrix factorization derives optimization algorithms of lower computational complexity as compared to relevant convex approaches, specifically algorithms derived from nuclear norm-based methods [106]. It is based on the fact that data exhibit latent structures; by uncovering them, we can obtain a compressed representation of the current data. For any matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$ of a rank up to r, NNMF finds an approximate factorization $\mathbf{M} \approx \mathbf{XY}$ into non-negative factors $\mathbf{X} \in \mathbb{R}^{n \times r}$ and $\mathbf{Y} \in \mathbb{R}^{r \times m}$, which can be attained through the minimization of the following equation:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{Y}} \ \frac{1}{2} \|\mathbf{XY} - \mathbf{M}\|_F^2, \\ &\text{s.t} \quad \mathbf{X} \geq 0, \quad \mathbf{Y} \geq 0. \end{aligned} \tag{20}$$

NNMF has been effectively used to exploit the low-rank property of the data. It's particularly well-suited for data with non-negative values and a low-rank structure.

5.5. Tensor decompositions

Tensor decompositions break down a high-dimensional tensor into a series of lower-dimensional tensors, facilitating the extraction of meaningful patterns and simplifying tensor completion.

CP decomposition. The CP decomposition factorizes multidimensional data, or tensors, into a sum of rank-one tensors. For an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we have

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} = [[\mathbf{\Lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]],$$

The symbol "$\circ$" represents the vector's outer product. This means that each element of the tensor is the product of the corresponding vector elements

$$\mathcal{X}_{i_1, i_2, \ldots, i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \cdots a_{i_N}^{(N)}, \quad \text{for all } 1 \leq i_n \leq I_n.$$

For any $n \in \{1, \ldots, N\}$, $\mathbf{a}_r^{(n)} \in \mathbb{R}^{I_n}$ stands for the factor (or loading) vector and λ_r represents weight parameters. The matrix $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ is called the n-th factor (or loading) matrix and refers to the combination of the vectors from the rank-one components, i.e., $\mathbf{A}^{(n)} = \left[\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_R^{(n)}\right]$, $\mathbf{\Lambda} = \operatorname{diag}\left[\lambda_1, \ldots, \lambda_R\right] \in \mathbb{R}^{R \times \cdots \times R}$ is a super-diagonal tensor, and R is the tensor rank also known as CP rank.

Tucker decomposition. Tucker decomposition factorizes multidimensional tensors into a core tensor multiplied by matrices along each mode. For a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ of order N, Tucker decomposition is given as

$$\begin{aligned} \mathcal{X} &= \sum_{s_1=1}^{R_1} \sum_{s_2=1}^{R_2} \cdots \sum_{s_N=1}^{R_N} \mathcal{G}_{s_1, \ldots, s_N} \mathbf{a}_{s_1}^{(1)} \circ \mathbf{a}_{s_2}^{(2)} \circ \ldots \circ \mathbf{a}_{s_N}^{(N)}, \\ &= \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} =: [[\mathcal{G}; \{\mathbf{A}\}]], \end{aligned}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ $(R_n \leq I_n)$ is the core tensor, $\{\mathbf{A}\}$ stands for the set of N matrices, and $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$ is factor matrix. The Tucker rank; also called the multilinear rank; is the vector $R = [R_1, R_2, \ldots, R_N]$, thus we say $\mathcal{X}$ is rank-$(R_1, R_2, \ldots, R_N)$ tensor. The matricized version of Tucker decomposition is given as:

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{G}_{(n)} (\mathbf{A}^{(N)} \otimes \ldots \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \ldots \otimes \mathbf{A}^{(1)})^T,$$

where $\mathbf{G}_{(n)}$ denotes mode-n matricization of $\mathcal{G}$ and $\mathbf{X}^T$ denotes the transpose of matrix $\mathbf{X}$. Tucker decomposition is generally non-unique, so practical applications often impose constraints on the core tensor and factor matrices to ensure a meaningful factorization. These constraints, such as orthogonality, non-negativity, or sparsity, are selected based on the specific requirements of the problem. When the factor matrices are orthogonal, they capture the most significant patterns or components in the data along each tensor dimension (or mode). Meanwhile, the elements of the core tensor $\mathcal{G}$ represent the level of interaction between the different modes.

6. Matrix-based motion completion approaches

6.1. Nuclear norm based kinematic constraints

The use of low-rank matrix completion theory for human motion data processing was first presented in [69]. In this work, the authors addressed both the completion and the denoising of the walking sequence data. Their proposed method relies on a low-rank prior without any kinematic information. The recovered MoCap matrix is obtained using the singular value thresholding algorithm (SVT) [89]. The SVT Algorithm 2 approximates the nuclear norm by minimizing the following problem:

$$\begin{aligned} &\min_{\mathbf{X}} \ \tau \|\mathbf{X}\|_* + \frac{1}{2} \|\mathbf{X}\|_F^2, \\ &\text{s.t} \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}). \end{aligned} \tag{21}$$

Nuclear norm with bone length constraint. The main limitation of the SVT-based MoCap completion method is that in different scenarios, the distance between any two adjacent joints of a motion sequence recovered by the SVT algorithm is not preserved. To address this problem, the bone length constraint (22) has been used within the SVT approach [75]. The objective of this approach is to maintain the relative distances between joints throughout the data recovery process. Therefore, the skeleton-constrained SVT model is given as:

$$\begin{aligned} &\min_{\mathbf{X}} \ \tau \|\mathbf{X}\|_* + \frac{1}{2} \|\mathbf{X}\|_F^2, \\ &\text{s.t} \ \ \mathbf{x} = Vec(\mathbf{X}), \\ &\mathbf{Ax} = \mathbf{b}, \\ &\|\mathbf{C}_i \mathbf{x}\|_2 \leq d_i, \quad \text{for} \ \ i = 1, \ldots, p, \end{aligned} \tag{22}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ extracts the observed entries of $\mathbf{x}$, and $\mathbf{b}$ is the vector containing the m observed entries of the matrix $\mathbf{M}$ to be recovered, i.e., $\mathbf{b} = \mathcal{P}_\Omega(\mathbf{M})$.

Nuclear norm with smoothness priors. The MoCap data recovery problem (21) has been converted into a robust matrix completion problem

[72,73,107] and using the low-rank structure, the smoothness prior (13) of the MoCap data, and the noise effect are considered. Hence, given the typical presence of noise in motion capture data, a realistic representation of the imperfect motion data matrix $\mathbf{M}$ involves decomposing it into two components expressed as follows:

$$\mathbf{M} = \mathbf{X} + \mathbf{E},$$

with $\mathbf{X}$ represents the complete and clean motion data, and $\mathbf{E}$, accounts for noise or outliers. Skeleton corruption often happens in challenging scenarios such as occlusion, so the error matrix $\mathbf{E}$ is sparse but unknown. The robust matrix completion model is given as:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{E}} \ \|\mathbf{X}\|_* + \alpha\|\mathcal{H}\circ\mathbf{E}\|_1 + \frac{\beta}{2}\left\|\mathbf{X}\mathbf{D}^T\right\|_F^2, \\ &\text{s.t.} \ \ \mathbf{M} = \mathbf{X} + \mathbf{E}, \end{aligned} \tag{23}$$

here, $\|\cdot\|_1$ is the l_1-norm, α and β are two regularization parameters, and $\mathcal{H} \in \{0,1\}^{d\times n}$ is the mask matrix (6). A similar optimization model has been used in [72], but with the truncated nuclear norm instead of the standard nuclear norm to better capture the low-rank structure of human MoCap data.

A multi-level fine-grained fusion method [108] introduces a fine-grained partitioning operator $\mathbf{L}_c^d$ that automatically segments motion data into anatomically-based body parts, where $d \in \{2,3,5\}$ represents different partitioning levels.

$$\mathbf{X} = \frac{1}{3}\sum_{d\in B}\sum_{c=1}^{d}\mathbf{X}\mathbf{L}_c^d, \quad B = \{2,3,5\}. \tag{24}$$

The final reconstruction combines results across multiple partitioning levels:

$$\begin{aligned} &\min_{\mathbf{X}} \ \ \frac{1}{3}\sum_{d\in B}\sum_{c=1}^{d}\|\mathbf{X}\mathbf{L}_c^d\|_* + \frac{\beta}{2}\left\|\mathbf{X}\mathbf{L}^T\right\|_F^2, \\ &\text{s.t.} \ \ \mathbf{M} = \mathbf{X}\circ\mathcal{H}, \quad B = \{2,3,5\}, \end{aligned} \tag{25}$$

where $\mathbf{X}\mathbf{L}_c^d$ denotes the c-th module resulting from the fine-grained partitioning at level d. The method provides alternative formulations using nuclear norm and Schatten p-norm approaches, allowing comparative evaluation of different low-rank priors on the same segmented data.

Nuclear norm with skeleton and smoothness prior. A hybrid prior within the low-rank completion problem is introduced [109]. This model leverages skeleton constraints and motion smoothness by combining spatial correlation from the skeleton structure and temporal information. Specifically, the spatiotemporal prior involves the isometry constraint, encouraging consistent bone lengths and exploring spatial correlations among skeleton sequences to overcome large joint errors in challenging cases. Therefore, let $\mathbf{M}$ be the corrupted matrix, $\mathbf{X}$ be the clean skeleton matrix, and $\mathbf{E}$ be the sparse noise matrix (sparse error). The spatiotemporal motion recovery problem is given as:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{E}} \ \|\mathbf{X}\|_* + \lambda\|\mathbf{E}\|_1 + \gamma\Theta_{\text{iso}}(\mathbf{X}) + \mu\|\mathbf{W}\mathbf{X}\|_1, \\ &\text{s.t.} \ \ \mathbf{M} = \mathbf{X} + \mathbf{E}, \end{aligned} \tag{26}$$

where $\Theta_{\text{iso}}(\mathbf{X})$ is the isometry term (11), the last term is the wavelet transform (14), and λ, γ, μ are regularization parameters to balance these terms. The isometry term exploits the spatial correlation of a skeleton sequence by suppressing positional errors of skeletal joints, while the smoothness term with sparse prior exploits the temporal correlation of a skeleton sequence by ensuring the piece-wise smoothness of the recovered motion. In this way, the proposed model can fully exploit the characteristics of skeleton motions.

QR decomposition-based approximate SVD. While nuclear norm minimization provides theoretical guarantees for low-rank matrix completion, the computational overhead associated with repeated singular value decomposition operations limits its practical applicability, particularly for real-time motion capture applications. To address this limitation, an approximate SVD approach based on QR decomposition has been introduced [91]. This method exploits the relationship between the nuclear norm and the $\ell_{2,1}$ norm to achieve significant computational savings while preserving recovery accuracy.

The approach decomposes the motion matrix $\mathbf{X}$ into three matrices $\mathbf{M} \in \mathbb{R}^{m\times r}$, $\mathbf{Q} \in \mathbb{R}^{r\times r}$, and $\mathbf{N} \in \mathbb{R}^{r\times n}$ such that,

$$\mathbf{X} = \mathbf{M}\mathbf{Q}\mathbf{N}, \tag{27}$$

where $\mathbf{M}^T\mathbf{M} = \mathbf{I}$ and $\mathbf{N}\mathbf{N}^T = \mathbf{I}$, and $r \ll \min(m,n)$ represents the effective rank. The key insight is that $\|\mathbf{X}\|_* = \|\mathbf{Q}\|_*$, allowing nuclear norm minimization of the much smaller matrix $\mathbf{Q}$ instead of the full matrix $\mathbf{X}$. Furthermore, by establishing the relationship $\|\mathbf{Q}\|_* \le \|\mathbf{Q}\|_{2,1}$, the nuclear norm is replaced with the $\ell_{2,1}$ norm, removing the need for iterative singular value decompositions and substantially reducing computational cost.

6.2. Schatten p-norm with kinematic constraints

The truncated Schatten p-norm has been proposed in [70] for MoCap completion. Moreover, the smoothness and the bone-length constraints are used to preserve the spatial-temporal smoothness and structural characteristics of human motion

$$\begin{aligned} &\min_{\mathbf{X}} \ \|\mathbf{X}\|_{r,S_p} + \frac{\beta}{2}\|\mathbf{X}\mathbf{D}\|_F^2, \\ &\text{s.t.} \quad \mathcal{H}\circ\mathbf{X} = \mathbf{M}, \ \text{and} \ g(\mathbf{X}) = 0, \end{aligned} \tag{28}$$

where $g(\mathbf{X})$ is the function to compute the bone length error in Eq. (10).

In human motion, it is noticeable that similar actions are frequently replicated in multiple instances, see Fig. 6. Such an assumption implies that human motion has strong nonlocal self-similarity (NSS). Motivated by this, the nonlocal low-rank regularization technique has been adopted for human MoCap recovery in [74]. To model the NSS technique of human motion, patch grouping for human motion sequences is defined to find the most similar poses among all the frames in the motion sequences. First, let $\mathbf{P}_i \in \mathbb{R}^{3d\times c}$ be a motion patch in $\mathbf{M}$, and we search for its similar patches along the entire motion sequence by the weighted l_2 distance. After patch grouping, we obtain a formed dataset $\mathbf{G}_i = \left[\mathbf{P}_{i_1}, \mathbf{P}_{i_2}, \ldots, \mathbf{P}_i, \ldots, \mathbf{P}_{i_s}\right] \in \mathbb{R}^{m\times h}$ for each exemplar patch $\mathbf{P}_i$. The completion problem is based on the weighted Schatten p-norm and formulated as

$$\begin{aligned} &\min_{\mathbf{G}_i} \ \left\|\mathbf{G}_i\right\|_{w,S_p}^p, \\ &\text{s.t.} \quad \mathcal{H}_i\circ\mathbf{G}_i = \mathbf{M}_i, \\ &g\left(\mathbf{G}_i\right) = 0. \end{aligned} \tag{29}$$

6.3. Nonlinear low-rank approach

In [110], the kernel-based low-rank matrix handles the completion problem after obtaining the learned kernel k_A by minimizing (19) with $p = 1$. Therefore, the nonlinear low-rank completion problem is given as:

$$\begin{aligned} &\min_{\mathbf{X}} \ \text{Tr}\left((\mathbf{K}(\mathbf{Q}))^{\frac{1}{2}}\right) + \frac{\beta}{2}\|\mathbf{X}\mathbf{D}\|_F^2, \\ &\text{s.t.} \quad \mathcal{H}\circ\mathbf{X} = \mathbf{M}, \\ &g(\mathbf{X}) = 0. \end{aligned} \tag{30}$$

The LRMC seeks the minimal rank of $\mathbf{X}$. However, considering the nonlinear structure of some motion sequences, $\mathbf{X}$ may not be of low

rank for MoCap data. Thus, the nonlinear low-rank minimizes the rank in the feature space, which is more suitable in theory. Although the non-linearization of LRMC improves accuracy, it comes at the cost of increased computational time.

6.4. Multi-subspace low-rank approaches

The low-rank matrix completion methods assume that motion data lies within a single low-dimensional subspace. However, this assumption breaks down for complex motion sequences containing multiple distinct activities, where the concatenated motion exhibits significantly higher rank than individual activity segments [111]. To address this limitation, multi-subspace approaches model complex motions as combinations of multiple low-dimensional subspaces. The discrete subspace structure constraint addresses this challenge by jointly optimizing subspace clustering and matrix completion within a unified framework. Given a motion sequence $\mathbf{X} \in \mathbb{R}^{3d\times n}$ containing c different activities, the DSS formulation is:

$$\begin{aligned}&\min_{\mathbf{X},\mathbf{E},\mathbf{u}_i,\mathbf{F}_i} \sum_{i=1}^{c}\left(\left\|(\mathbf{X}-\mathbf{u}_i\mathbf{1}^T)\mathbf{F}_i\right\|_{S_p}^{p}\right)^{t}+\lambda_1\|\mathcal{P}_\Omega(\mathbf{E})\|_1+\frac{\lambda_2}{2}\Theta(\mathbf{X}),\\ &\text{s.t.}\quad \mathbf{X}+\mathbf{E}=\mathcal{P}_\Omega(\mathbf{M}),\quad \mathbf{F}_i\subseteq\{0,1\}^{n\times n},\quad \sum_{i=1}^{c}\mathbf{F}_i=\mathbf{I}_n,\end{aligned} \tag{31}$$

where $\mathbf{F}_i$ are diagonal indicator matrices specifying which frames belong to the i-th subspace, $\mathbf{u}_i$ are translation vectors to center each subspace, and the t-th power of Schatten p-norm provides better rank approximation than the nuclear norm. Unlike previous two-stage approaches that perform subspace clustering and matrix completion sequentially, this formulation allows the two processes to mutually benefit each other through joint optimization. The algorithm alternates between solving for the recovered data $\mathbf{X}$ given the segmentation matrices $\mathbf{F}_i$ and solving for the segmentation given $\mathbf{X}$, enabling improved accuracy for both tasks.

6.5. Hierarchical block-based NNMF

The completion process proposed in [76] has taken advantage of adaptive NNMF combined with hierarchical block-based human motion data recovery. The motion sequence is processed in terms of block-based subchain motion clips. The proposed method consists of two main layers:

- *Interior layer:* In this layer, the motion is segmented into block-based subchain motion clips by decomposing the skeleton data into five correlated blocks. Second, the adaptive NNMF is then used to recover the moving trajectories of each sub-chain motion individually by solving the following problem:

$$\min_{\mathbf{X},\mathbf{Y}} \left\|\mathcal{P}_\Omega(\mathbf{XY}-\mathbf{M})\right\|_F^2 \quad \text{s.t. } \mathbf{X}\geq 0, \mathbf{Y}\geq 0. \tag{32}$$

- *Exterior layer:* It exploits the recovered sub-chain motion clips from the interior layer and the known entries in the MoCap data to refine the corresponding restored data of the same position. Therefore, natural-looking human motions can be obtained from the two layers.

7. Tensor decomposition for MoCap data completion

Tensor decomposition approaches address the completion problem by factorizing the multi-dimensional data structure into core components. These methods capitalize on the inherent temporal and spatial correlations in motion data, providing advantages over matrix-based techniques. The decomposition into lower-dimensional factors enables the preservation of underlying motion patterns while accurately reconstructing missing values.

7.1. CP decomposition algorithms

The goal of the CP decomposition is to find a set of R normalized rank-one tensors $\{\mathbf{a}_r^{(1)}\circ\mathbf{a}_r^{(2)}\circ\cdots\circ\mathbf{a}_r^{(N)}\}_{r=1}^{R}$ that best approximates $\mathcal{X}$. Thus, in [77], MoCap tensor completion using CP decomposition is formulated as:

$$\begin{aligned}&\min_{\mathcal{X},\{\mathcal{A}_r\}_{r=1}^{R}} H(\mathcal{X},\mathcal{A}_1,\ldots,\mathcal{A}_R)=\min_{\mathcal{X},\{\mathcal{A}_r\}_{r=1}^{R}}\frac{1}{2}\|\mathcal{X}-\sum_{r=1}^{R}\lambda_r\mathbf{a}_r^{(1)}\circ\mathbf{a}_r^{(2)}\circ\cdots\circ\mathbf{a}_r^{(N)}\|_F^2,\\ &\text{s.t}\quad \mathcal{P}_\Omega(\mathcal{X})=\mathcal{P}_\Omega(\mathcal{M}),\quad \|\mathbf{a}_r^{(n)}\|_2=1,\ n\in[N], r\in[R],\end{aligned} \tag{33}$$

where $\mathcal{A}_r=\lambda_r\mathbf{a}_r^{(1)}\circ\mathbf{a}_r^{(2)}\circ\cdots\circ\mathbf{a}_r^{(N)}$. Normalizing the loading factors in CP decomposition offers certain advantages, such as improving the robustness to noise and preventing the minimization process from assigning excessively large values to some components.

Given the inherent complexity of MoCap data, two modified variations of CP decomposition are introduced: SmoothCP and SparseCP. These modified decompositions incorporate constraints on the factor vectors to enhance the extraction of meaningful patterns by leveraging smoothness and sparsity properties, respectively. Smoothness is an essential characteristic of MoCap data, which can be effectively utilized to predict missing elements more accurately. In the SmoothCP decomposition, 1D total variation constraint on the factor vectors is incorporated.

$$\begin{aligned}&\min_{\mathcal{X},\{\mathcal{A}_r\}_{r=1}^{R}} H(\mathcal{X},\mathcal{A}_1,\ldots,\mathcal{A}_R)+\sum_{r=1}^{R}\sum_{n=1}^{N}\alpha_n\|\mathbf{D}_{tv}\mathbf{a}_r^{(n)}\|^2,\\ &\text{s.t}\quad \mathcal{P}_\Omega(\mathcal{X})=\mathcal{P}_\Omega(\mathcal{M}),\quad \|\mathbf{a}_r^{(n)}\|_2=1,\ n\in[N], r\in[R],\end{aligned} \tag{34}$$

where $\{\alpha_n\}_{n=1}^{N}$ are parameters controlling the level of the constraint on each factor vector. This approach ensures that the decomposition respects the smooth transitions inherent in human motion. Conversely, sparsity focuses on identifying and utilizing the most informative parts of the data, which is crucial in scenarios where only a few significant elements contribute to the overall structure. In the SparseCP decomposition, the sparsity constraint is introduced by applying the l_1-norm to the factor vectors.

$$\begin{aligned}&\min_{\mathcal{X},\{\mathcal{A}_r\}_{r=1}^{R}} H(\mathcal{X},\mathcal{A}_1,\ldots,\mathcal{A}_R)+\sum_{r=1}^{R}\sum_{n=1}^{N}\alpha_n\|\mathbf{a}_r^{(n)}\|_1,\\ &\text{s.t}\quad \mathcal{P}_\Omega(\mathcal{X})=\mathcal{P}_\Omega(\mathcal{M}),\quad \|\mathbf{a}_r^{(n)}\|_2=1,\ n\in[N], r\in[R].\end{aligned} \tag{35}$$

The SparseCP approach guides the decomposition to produce sparse factor vectors, highlighting the most informative components and reducing redundancy.

7.2. Tucker decomposition algorithms

Tucker decomposition factorizes the MoCap tensor into a core tensor multiplied by matrices along each mode [112]. The orthogonality constraint ensures that components in each mode are independent, which improves numerical stability by representing each mode through a set of independent features. The tensor completion problem via Tucker decomposition thus exploits this structured, low-rank representation to reconstruct missing tensor entries:

$$\begin{aligned}&\min_{\mathbf{A},\mathcal{G},\mathcal{X}}\frac{1}{2}\|\mathcal{X}-\mathcal{G}\times_1\mathbf{A}^{(1)}\times_2\mathbf{A}^{(2)}\ldots\times_N\mathbf{A}^{(N)}\|_F^2,\\ &\text{s.t}\qquad \mathcal{P}_\Omega(\mathcal{X})=\mathcal{P}_\Omega(\mathcal{M}),\quad \mathbf{A}^{(n)}\mathbf{A}^{(n),T}=\mathbf{I}_n,\ \ n\in[N].\end{aligned} \tag{36}$$

MoCap data exhibits inherent temporal continuity, with motion evolving smoothly across successive frames. Given the high-dimensional nature of such data, explicit gradient-based smoothness regularization becomes computationally prohibitive. The nuclear norm preserves temporal smoothness by promoting low-rank structure, which inherently favors gradual patterns over abrupt changes. By applying the nuclear norm to the temporal mode of the MoCap tensor, abrupt variations are

Algorithm 1: Augmented Lagrange multiplier (ALM) algorithm.

Initialization;
$\mathbf{x}_0$, $\mathbf{y}_0$, $k = 1$
while *not convergent and* $k \leq Kmax$ **do**
  $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} L\left(\mathbf{x}, \mathbf{y}_k, \mu_k\right)$,
  $\mathbf{y}_{k+1} = \mathbf{y}_k + \mu_k A\left(\mathbf{x}_{k+1}\right)$,
  Update μ_k to μ_{k+1}.
end
return $\mathbf{x}_k$

mitigated, ensuring reconstructed motion maintains coherent temporal patterns. The Tucker-based temporal nuclear norm approach is given as:

$$\begin{aligned} &\min_{\mathbf{A},\mathcal{G},\mathcal{X}} \ \frac{1}{2}\|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \ldots \times_N \mathbf{A}^{(N)}\|_F^2 + \lambda\|\mathbf{X}_{(1)}\|_*, \\ &\text{s.t} \quad \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M}), \quad \mathbf{A}^{(n)}\mathbf{A}^{(n),T} = \mathbf{I}_n, \quad n \in [N]. \end{aligned} \tag{37}$$

where $\mathbf{X}_{(1)}$ is the mode-1 unfolding (matricization) of $\mathcal{X}$. The proposed approach leverages this global smoothness constraint within the Tucker decomposition framework, providing a computationally efficient method to preserve motion continuity.

8. Minimization algorithms

The completion problem for motion capture data is presented as a minimization problem, where the goal is to recover missing or corrupted motion sequences by minimizing an objective function. The following sections review the key minimization methods used to address the MoCap completion approaches.

8.1. Augmented lagrange multipliers (ALM) method

The ALM is an optimization technique for solving constrained optimization problems [113–115]. As an extension of the Lagrange multipliers, ALM addresses the challenges of handling equality and inequality constraints [116]. The standard constrained optimization problem is formulated as:

$$\begin{aligned} &\min_{\mathbf{x}} \quad f(\mathbf{x}), \\ &\text{s.t} \quad A(\mathbf{x}) = 0, \end{aligned} \tag{38}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is real-valued function and $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator. ALM defines the augmented Lagrangian function by introducing Lagrange multipliers for the constraints. The augmented Lagrangian combines the original objective function with penalty terms related to the constraints:

$$L(\mathbf{x}, \mathbf{y}, \mu) = f(\mathbf{x}) + \langle \mathbf{y}, A(\mathbf{x})\rangle + \frac{\mu}{2}\|A(\mathbf{x})\|_F^2,$$

where μ is a regularization parameter. In each iteration, ALM performs alternating minimization of the augmented Lagrangian over $\mathbf{X}$. At iteration k, ALM carries out the following steps

$$\begin{aligned} \mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\mu_k}{2}\|A(\mathbf{x}) + \mathbf{y}_k\|_F^2, \\ \mathbf{y}_{k+1} &= \mathbf{y}_k + \mu_k A(\mathbf{x}_{k+1}). \end{aligned}$$

The complete algorithmic procedure is detailed in Algorithm 1.

8.2. Singular value thresholding (SVT) algorithm

The SVT algorithm is an iterative optimization technique primarily employed for solving matrix recovery and completion problems [89]. It can be viewed as a special case of the Augmented Lagrangian Method (ALM). The SVT algorithm is designed to recover a low-rank matrix from a given set of noisy or incomplete observations. Practically, the SVT algorithm solves the minimization problem (21) iteratively. The key building block of the SVT algorithm is the singular value shrinkage operator. Consider the SVD of a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ of rank $r \leq \min(m, n)$

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \quad \text{where} \quad \mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r), \tag{39}$$

where $\mathbf{U}$ and $\mathbf{V}$ are respectively $m\times r$ and $n\times r$ matrices with orthonormal columns, and the singular values σ_i are positive. For each $\tau \geq 0$, the soft-thresholding operator D_τ is defined as follows:

$$D_\tau(\mathbf{X}) = \mathbf{U}D_\tau(\mathbf{\Sigma})\mathbf{V}^*, \quad D_\tau(\mathbf{\Sigma}) = \operatorname{diag}((\sigma_i - \tau)_+), \tag{40}$$

where $(\sigma_i - \tau)_+ = \max(0, \sigma_i - \tau)$. Fixing $\tau > 0$ and a sequence δ_k of scalar step sizes, the SVT aims to find a low-rank solution by iteratively updating matrix pairs $(\mathbf{X}_k, \mathbf{Y}_k)_{k\in\mathbb{N}}$ as follows:

$$\begin{cases} \mathbf{X}_k = \mathcal{D}_\tau(\mathbf{Y}_{k-1}), \\ \mathbf{Y}_k = \mathbf{Y}_{k-1} + \delta_k \mathcal{P}_\Omega(\mathbf{M} - \mathbf{X}_k). \end{cases}$$

The complete algorithmic procedure is detailed in Algorithm 2.

Algorithm 2: Singular value thresholding (SVT) algorithm.

Initialization;
$\mathbf{X}_0$, $\mathbf{Y}_0$, $k = 1$
while *not convergent and* $k \leq Kmax$ **do**
  $\mathbf{X}_{k+1} = \mathcal{D}_\tau(\mathbf{Y}_k)$, (soft-thresholding: Eq. (40))
  $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \delta_k \mathcal{P}_\Omega(\mathbf{M} - \mathbf{X}_{k+1})$,
  Update δ_k to δ_{k+1}.
return $\mathbf{X}_k$

8.3. Alternating direction method of multipliers (ADMM) algorithm

The ADMM is an optimization algorithm used to solve problems that can be formulated as convex optimization problems with linear constraints [117,118]. The ADMM algorithm solves problems of the form:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}), \\ &\text{s.t} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \tag{41}$$

where f and g are convex functions, not necessarily differentiable, $\mathbf{A}, \mathbf{B}$ are given matrices and $\mathbf{c}$ a given vector. The optimal value of problem (41) will be obtained by minimizing the augmented Lagrangian function

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \langle \mathbf{y}, \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\rangle + \frac{\gamma}{2}\|A\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2.$$

In each iteration, ADMM performs alternating minimization of the augmented Lagrangian over $\mathbf{x}$ and $\mathbf{z}$. At iteration k, ADMM carries out the following steps:

$$\begin{aligned} \mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\gamma}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_k - \mathbf{c} + \mathbf{y}_k\|_F^2, \\ \mathbf{z}_{k+1} &= \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\gamma}{2}\|\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{y}_k\|_F^2, \\ \mathbf{y}_{k+1} &= \mathbf{y}_k + \gamma\left(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{c}\right), \end{aligned}$$

where γ is a positive parameter. The complete algorithmic procedure is detailed in Algorithm 3.

8.4. Fixed-point proximity algorithms

Fixed-point proximity algorithms provide an alternative to ADMM for solving multi-term convex optimization problems with guaranteed

Algorithm 3: Alternating direction method of multipliers (ADMM) algorithm.

Initialization;
$\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0, k = 1$
while *not convergent and* $k \leq Kmax$ **do**
  $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\gamma}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_k - \mathbf{c} + \mathbf{y}_k\|_F^2$,
  $\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\gamma}{2}\|\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{y}_k\|_F^2$,
  $\mathbf{y}_{k+1} = \mathbf{y}_k + \gamma\left(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{c}\right)$.
end
return $\mathbf{x}_k$ *and* $\mathbf{y}_{k+1}$

Algorithm 4: Fixed-point proximity algorithm (FPPA).

Initialization;
$\mathbf{U}^{(0)} = \mathbf{0}$, $\mathbf{X}^{(0)} = \mathbf{X}_0$
while *not convergent and* $k \leq Kmax$ **do**
  $\mathbf{U}^{(k+1)} = \text{prox}_{g^*,\mathbf{P}}(\mathbf{U}^{(k)} - \mathbf{P}^{-1}\mathbf{X}^{(k)} + \mathbf{P}^{-1}\mathbf{X}_0)$,
  $\overline{\mathbf{U}}^{(k+1)} = 2\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)}$, $\overline{\mathbf{X}}^{(k+1)} = \mathbf{X}^{(k)} - \frac{1}{\lambda}\mathbf{Q}^{-1}\mathbf{L}^T\nabla h(\mathbf{L}\mathbf{X}^{(k)})$,
  $\mathbf{X}^{(k+1)} = \text{prox}_{f,\mathbf{Q}}(\overline{\mathbf{X}}^{(k+1)} + \mathbf{Q}^{-1}\overline{\mathbf{U}}^{(k+1)})$.
end
return $\mathbf{X}^{(k+1)}$

convergence properties. The Fixed-Point Proximity Algorithm (FPPA) addresses the general form [107]:

$$\min_{\mathbf{X}} f(\mathbf{X}) + g(\mathbf{X}) + h(\mathbf{L}\mathbf{X}), \tag{42}$$

where f and g are proper convex functions with computable proximity operators, h is differentiable and convex, and $\mathbf{L}$ is a linear operator. The algorithm characterizes solutions through fixed-point equations. For positive parameters λ, $\mathbf{P}$, $\mathbf{Q} \in \mathbb{S}_+^m$, the solution satisfies:

$$\mathbf{U} = \text{prox}_{g^*,\mathbf{P}}(\mathbf{U} - \mathbf{P}^{-1}\mathbf{X} + \mathbf{P}^{-1}\mathbf{X}_0), \tag{43}$$

$$\mathbf{X} = \text{prox}_{f,\mathbf{Q}}(\mathbf{X} + \mathbf{Q}^{-1}\mathbf{U} - \frac{1}{\lambda}\mathbf{Q}^{-1}\mathbf{L}^T\nabla h(\mathbf{L}\mathbf{X})). \tag{44}$$

FPPA avoids matrix inversions required in ADMM through eigen-decomposition techniques and provides guaranteed convergence for three-term problems where ADMM convergence may fail.

8.5. The block coordinate descent (BCD) algorithm

The BCD algorithm is a popular method for solving large-scale optimization problems. It iteratively optimizes a function by updating one block of variables at a time while keeping the others fixed. This approach is particularly effective when the objective function can be decomposed into components that can be easily optimized over individual blocks. The BCD minimizes problems of the form:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_s} f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s), \tag{45}$$

where variable $\mathbf{x}$ is decomposed into s blocks $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s$, f is a real-valued function, each $\mathbf{x}_i$ belongs to $\mathbb{R}^{n_i}$. For each iteration k, BCD updates each variable block $\mathbf{x}_i^{k+1}$ by minimizing the objective function over the current block:

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} g(\mathbf{x}_i) = \arg\min_{\mathbf{x}_i} f(\mathbf{x}_1^{k+1}, \ldots, \mathbf{x}_{i-1}^{k+1}, \mathbf{x}_i, \mathbf{x}_{i+1}^{k}, \ldots, \mathbf{x}_s^{k}).$$

This process is repeated cyclically for all blocks. BCD is widely used because it is easy to implement and efficient for large-scale problems, especially when each block update can be computed independently. However, standard BCD can face slow convergence and instability when dealing with nonsmooth or constrained objectives. To address this, the

Algorithm 5: Block coordinate descent (BCD) algorithms.

Initialization: $\mathbf{x}^0 = (\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_s^0)$, $k = 1$
while *not convergent and* $k \leq Kmax$ **do**
  for $i = 1, 2, \ldots, s$ **do**
    If BCD, update: $\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} g(\mathbf{x}_i)$,
    If Prox-BCD, update:
    $\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} g(\mathbf{x}_i) + \frac{\rho_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_F^2$.
  end
end
return $\mathbf{x}^{k+1}$

Proximal Regularized BCD (Prox-BCD) method incorporates a proximal term into the objective function, which helps stabilize the updates and improve convergence [119–121]. The Proximal-BCD algorithm first uses a proximal point modification of (45) as follows

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_s} f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s) + \sum_{i=1}^{s} \frac{\rho_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_F^2,$$

where $\frac{\rho_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_F^2$ is the proximal term added to the objective to regularize the update, preventing large deviations from the current estimate $\mathbf{x}_i^k$ and ρ_i is a regularization parameter. Given $\mathbf{x}^0 = (\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_s^0)$ as an initial estimate, the algorithm updates the estimates of $\mathbf{x}^{k+1} = (\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \ldots, \mathbf{x}_s^{k+1})$ alternately in the $(k+1)^{\text{th}}$ iteration as follows:

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} f(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \ldots, \mathbf{x}_{i-1}^{k+1}, \mathbf{x}_i, \mathbf{x}_{i+1}^{k}, \ldots, \mathbf{x}_s^{k}) + \frac{\rho_i}{2}\|\mathbf{x} - \mathbf{x}_i^k\|_F^2,$$

The complete algorithmic procedure for the two BCD algorithms is detailed in Algorithm 5.

9. Data and tools

9.1. Motion capture datasets

9.1.1. CMU motion capture database

The CMU[3] contains motion capture sequences recorded using an optical motion capture system available in both C3D and ASF/AMC data formats. Data were captured using a Vicon motion capture system with 12 MX-40 infrared cameras, recording at 120 Hz with a 4-megapixel resolution in a working volume of 3m × 8m. Subjects were equipped with 41 markers and dressed in stylish black garments. The dataset encompasses recordings from several subjects performing various actions across 6 categories: Human Interaction, Interaction with Environment, Locomotion, Physical Activities and Sports, Situations and Scenarios, and Test Motions. This collection includes a range of movements, including simple walking motions and complex actions such as martial arts, dances, and sports.

9.1.2. HDM05 dataset

The HDM05[4] contains more than three hours of systematically recorded and well-documented motion capture data, available in both C3D and ASF/AMC data formats. The dataset encompasses over 70 motion classes, each with 10 to 50 realizations performed by different actors. Five actors performed several repetitions of each motion sequence, with additional freestyle sequences containing miscellaneous motions recorded for some performers. Data were captured using a Vicon MX system comprising twelve high-resolution cameras, with six operating in the visible red spectral range and six in the infrared spectral range. All recordings were conducted at a sampling rate of 120 Hz. The cameras were arranged to provide a viewing volume diameter of approximately five meters.

[3] http://Mocap.cs.cmu.edu/.
[4] https://resources.mpi-inf.mpg.de/HDM05/.

9.2. MATLAB motion capture toolbox

The MoCap Toolbox[5] provides comprehensive functionality for analyzing and visualizing motion capture data within the MATLAB environment. Primarily designed for music-related movement analysis, this versatile toolbox offers applications across various research domains. It supports multiple industry-standard file formats, including.c3d, Qualisys system's.tsv and.mat formats,.bvh, and.wii format from WiiDataCapture software. Researchers can leverage this toolbox's specialized functions to process, analyze, and generate visual representations of motion capture data, making it an essential resource for movement analysis studies [122].

10. Discussion

10.1. Algorithmic complexity, robustness, and practical considerations

Different low-rank formulations offer distinct advantages and limitations for motion capture data completion. Nuclear norm-based methods provide a convex relaxation of the rank minimization problem and offer theoretical guarantees under certain incoherence conditions. However, they require repeated singular value decompositions, with a per-iteration complexity of $O(mn\min(m,n))$ for an $m\times n$ matrix. This limits their scalability to long motion sequences. Moreover, nuclear norm minimization tends to over-shrink singular values, which may distort important motion dynamics. While they are relatively insensitive to initialization, their performance depends on the choice of regularization parameters, and they can be sensitive to noise, often requiring robust variants with explicit sparse error modeling.

Schatten-p norm methods generalize the nuclear norm and provide more flexible rank approximations. Their computational complexity remains comparable to nuclear norm methods ($O(mnr)$ per iteration), but the inclusion of the parameter p introduces additional sensitivity: smaller p promotes sparsity but may slow convergence, while larger p behaves more like the nuclear norm. These methods can better preserve dominant motion modes and reduce overshrinkage, but they demand careful parameter tuning to balance accuracy and stability. In terms of robustness, they handle moderate noise better than the nuclear norm, yet may still fail under severe occlusion or highly corrupted sequences.

Matrix factorization approaches avoid repeated full SVD computations and reduce the per-iteration complexity to $O(mnr)$. However, they are inherently non-convex, and performance is sensitive to the choice of rank r and initialization. Poor parameter selection may lead to convergence to local minima. While factorization approaches are more scalable, they are less robust to high levels of noise and outliers, since the learned factors may overfit corrupted entries without additional regularization.

Nonlinear kernel-based approaches project motion data into a higher-dimensional feature space where low-rank assumptions become more effective for capturing nonlinear dependencies among markers. While they improve accuracy for complex or highly articulated movements, this comes at substantial computational cost: kernel matrix construction and decomposition typically scale as $O(n^2)$–$O(n^3)$ with sequence length n, making them less suitable for long sequences. These methods are highly sensitive to kernel choice (e.g., Gaussian, polynomial, geodesic) and associated hyperparameters, such as kernel width, which strongly influence recovery quality. Although they can be robust to mild nonlinear distortions, their performance degrades with noisy or mismatched kernel parameters, and their scalability remains a major drawback in practice.

Tensor decomposition methods extend low-rank recovery to multidimensional data, preserving correlations across time, markers, and spatial coordinates. CP decomposition incurs $O(R(IJ+IK+JK))$ per iteration for an $I\times J\times K$ tensor, while Tucker decomposition scales with $O(IJR_3+IKR_2+JKR_1)$. Although they capture complex multimodal dependencies and achieve high reconstruction accuracy, they suffer from high computational demands and memory costs, limiting their use for large-scale motion capture data. Parameter sensitivity is also pronounced: choosing multilinear ranks (R_1,R_2,R_3) requires empirical tuning, and overestimation can increase computation without clear accuracy gains. They are relatively robust to structured noise due to multi-mode modeling, but remain sensitive to long-duration gaps and to errors in marker placement.

10.2. Motion prior integration and trade-offs

Incorporating biomechanical and smoothness constraints into low-rank formulations enhances the accuracy and realism of motion completion methods, but also introduces nontrivial trade-offs. Smoothness constraints are particularly common, as they are computationally tractable. However, they implicitly assume continuity in motion trajectories that may not hold across all movement types. The choice between different orders of smoothness reflects prior assumptions about the underlying motion dynamics. For instance, first-order constraints emphasize constant velocity patterns and work well for cyclical activities such as walking or running, whereas second-order constraints impose acceleration smoothness, making them better suited for gestural or reaching motions. The application of these constraints becomes problematic for movements involving rapid changes or contact interactions. Jerky actions, collisions, or object contacts introduce discontinuities that directly contradict smoothness priors. In such cases, constraints may lead to reconstructions that are overly damped, lacking natural dynamics, or stripped of distinctive features. To address this, a promising direction is to develop adaptive or context-aware constraints that automatically adjust their strength according to the local motion dynamics. This would ensure biomechanical plausibility in smooth segments while avoiding the oversmoothing or distortion of highly dynamic movements.

Kinematic constraints such as bone-length preservation embed anatomical knowledge into the recovery process. These constraints enforce fixed skeletal proportions and joint connectivity, ensuring that reconstructed poses remain biomechanically feasible. They also enhance robustness to missing markers by ruling out implausible limb configurations. However, their effectiveness is highly dependent on accurate skeletal calibration and reliable marker placement. In noisy or uncontrolled capture environments, rigid enforcement of bone-length constraints may conflict with corrupted data, introducing artifacts such as unnatural joint angles or body distortions.

Biomechanical and smoothness priors provide strong guidance that improves plausibility and recovery quality, particularly when data is heavily corrupted or incomplete. At the same time, they narrow the solution space in ways that can exclude valid but unexpected motions. These trade-offs highlight the need for future work on scalable, noise-robust formulations that balance accuracy, efficiency, and physical realism to enable reliable deployment of motion capture recovery in practical applications.

10.3. Applicability across motion types

The effectiveness of low-rank methods fundamentally depends on the underlying assumption that human motion exhibits low-dimensional structure. This assumption proves robust for many common activities but faces significant challenges across the full spectrum of human movement. Periodic and semi-periodic motions such as walking, running, and basic gestures hold strong low-rank characteristics due to their repetitive nature and biomechanical constraints. The singular value decomposition of such motions typically shows rapid decay, confirming that relatively few principal components capture the majority of motion variance.

However, this fundamental assumption weakens considerably for aperiodic or highly variable movements. Complex dance sequences and freestyle sports activities exhibit substantially higher intrinsic

[5] www.jyu.fi/music/coe/materials/mocaptoolbox.

dimensionality due to their unpredictable trajectories and creative variations. The effective rank of such motions can increase significantly compared to repetitive activities, leading to proportional degradation in completion accuracy. However, the inherent correlations between joints and markers, enforced by the skeletal structure and biomechanical constraints (e.g., bone length preservation, joint coupling), partially mitigate this effect. Since bones impose fixed-length relationships, the motion data still exhibit redundancies that reduce the effective rank relative to treating marker trajectories as fully independent. This structural coupling explains why even complex, aperiodic motions remain partially amenable to low-rank recovery, though with lower accuracy than highly regular activities.

11. Conclusion and future directions

This paper provides a comprehensive review of motion capture data recovery methods based on the low-rank structure. We have discussed approaches that rely on motion properties and skeleton structure, as well as methods centered on introducing low-rank priors. The integration of kinematic constraints directly into the completion process, referred to as MoCap data priors, has been used to refine the low-rank matrix completion problem. Moreover, the analysis of various low-rank priors provides valuable insights into the diverse strategies employed in MoCap data recovery. Matrix-based approaches such as nuclear norm minimization and its Schatten-p norm extensions provide strong theoretical guarantees but remain computationally demanding and prone to overshrinkage of singular values. Matrix factorization reduces computational cost and improves scalability but introduces non-convexity and sensitivity to initialization and rank selection. Kernel-based methods capture nonlinear dependencies effectively but are computationally expensive and highly sensitive to kernel choice and hyperparameters. Tensor decompositions extend recovery to multidimensional data yet they remain limited by high computational requirements. Finally, biomechanical and temporal priors improve physical plausibility and robustness but may oversimplify dynamics or introduce artifacts under noisy capture conditions. From this comparative review, several future research directions emerge:

- *Scalable algorithms:* Current low-rank optimization methods often involve repeated singular value decompositions or high-dimensional tensor factorizations, which become computationally expensive for long sequences or large-scale motion capture datasets. Future research could explore randomized, or streaming (online) algorithms that incrementally update low-rank factors, enabling real-time or large-scale applications without sacrificing accuracy.
- *Integration of physical priors:* Current constraint formulations (e.g., bone-length preservation, smoothness) are typically applied in a fixed manner across all motions. However, their strict enforcement may conflict with noisy or high-variability data. A promising future direction is the design of context-aware or adaptive priors that adjust constraint strength dynamically based on motion type, capture quality, or confidence in skeletal calibration. For example, bone-length constraints could be relaxed when marker errors are detected, while smoothness priors could adaptively scale depending on whether the motion is cyclical or abrupt. Such adaptive integration could balance physical plausibility with flexibility in diverse motion scenarios.
- *Advanced tensor decompositions:* Current tensor-based methods are restricted to CP and Tucker decompositions, which face challenges such as high computational cost and sensitivity to rank selection. Future research could explore higher-order or alternative tensor formats, including Tensor Train, Tensor Ring, and Hierarchical Tucker decompositions. These models address some of the core limitations of current approaches by providing more compact representations, improving scalability to long sequences, and enabling multi-scale analysis. Furthermore, higher-order decompositions allow the integration of additional modes beyond the time–marker–coordinate structure, such as multi-subject interactions or multimodal signals, thereby capturing richer spatiotemporal dependencies and improving robustness in complex scenarios.
- *Multi-subject motion capture:* Most existing algorithms assume single-subject data, yet real-world applications often involve interactions among multiple individuals (e.g., couples dancing, group sports, collaborative tasks). Extending low-rank formulations to multi-subject scenarios requires capturing inter-subject correlations, handling overlapping trajectories, and ensuring consistent skeletal calibration across individuals. This represents an important but underexplored direction that could expand the applicability of motion capture recovery to richer social and collaborative contexts.

CRediT authorship contribution statement

S. Mohaoui: Methodology, Conceptualization, Validation, Visualization, Writing—original draft. **A. Dmytryshyn:** Methodology, Conceptualization, Supervision, Writing—review, Validation.

Declaration of generative AI in scientific writing

Generative AI is only used in the writing process to enhance the readability and language of the manuscript.

Funding

Carl Tryggers Stiftelse supports the first author's work under grant CTS 22:2196. The Swedish Research Council (VR) supports the second author's work under grant 2021-05393.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Y. Hou, Z. Li, P. Wang, W. Li, Skeleton optical spectra-based action recognition using convolutional neural networks, IEEE Trans. Circuits Syst. Video Technol. 28 (3) (2016) 807–811.
[2] M.E. Hussein, M. Torki, M.A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations, in: Ijcai, vol. 13, 2013, pp. 2466–2472.
[3] C. Zhao, M. Chen, J. Zhao, Q. Wang, Y. Shen, 3d behavior recognition based on multi-modal deep space-time learning, Appl. Sci. 9 (4) (2019) 716.
[4] N.E.D. Elmadany, Y. He, L. Guan, Multimodal learning for human action recognition via bimodal/multimodal hybrid centroid canonical correlation analysis, IEEE Trans. Multimed. 21 (5) (2018) 1317–1331.
[5] M. Li, H. Leung, Graph-based approach for 3D human skeletal action recognition, Pattern Recognit. Lett. 87 (2017) 195–202.
[6] I. Kapsouras, N. Nikolaidis, Action recognition by fusing depth video and skeletal data information, Multimed. Tools Appl. 78 (2) (2019) 1971–1998.
[7] A. Daffertshofer, C.J.C. Lamoth, O.G. Meijer, P.J. Beek, PCA in studying coordination and variability: a tutorial, Clin. Biomech. 19 (4) (2004) 415–428.
[8] N.F. Troje, Decomposing biological motion: a framework for analysis and synthesis of human gait patterns, J. Vis. 2 (5) (2002) 2.
[9] D. Rojas-Valverde, J. Pino-Ortega, C.D. Gómez-Carmona, M. Rico-González, A systematic review of methods and criteria standard proposal for the use of principal component analysis in team's sports science, Int. J. Environ. Res. Public Health 17 (23) (2020) 8712.
[10] E. Archibeck, R. Halvorson, P. Silvestros, A. Torres-Espin, G. O'Connell, J. Bailey, 3D motion capture data into a kinematic composite score for assessing musculoskeletal impairments, J. Biomech. (2025) 112725.
[11] Q. Wei, et al., Structured low-rank matrix factorization for hyperspectral image restoration, IEEE Trans. Image Process. 30 (2021) 1723–1737.
[12] R. Hu, J. Tong, J. Xi, Q. Guo, Y. Yu, Matrix completion-based channel estimation for mmwave communication systems with array-inherent impairments, IEEE Access 6 (2018) 62915–62931.
[13] K.F. Masood, J. Tong, J. Xi, J. Yuan, Q. Guo, Y. Yu, Low-rank matrix sensing-based channel estimation for mmwave and thz hybrid MIMO systems, IEEE J. Sel. Top. Signal Process. 17 (4) (2023) 777–793.
[14] S. Majidian, M.M. Mohades, M.H. Kahaei, Matrix completion with weighted constraint for haplotype estimation, Digit. Signal Process. 108 (2021) 102880.

[15] N. Rao, H.-F. Yu, et al., Collaborative filtering with graph information: consistency and scalable methods, Adv. Neural Inf. Process. Syst. 28 (2015).

[16] J. Xu, et al., Inductive matrix completion based on graph neural networks, in: International Conference on Learning Representations (ICLR), 2020.

[17] D. Chawin, I. Haviv, New hardness results for low-rank matrix completion, arXiv preprint arXiv:2506.18440, 2025.

[18] Z. He, et al., Accelerated dynamic mr imaging with joint balanced low-rank tensor and sparsity constraints, Med. Phys. 50 (6) (2023) 3791–3805.

[19] P. Li, Y. Hu, Learned tensor low-cp-rank and Bloch response manifold priors for non-cartesian MRF reconstruction, IEEE Trans. Med. Imaging 42 (12) (2023) 3702–3714.

[20] T. Cao, Z. Hu, X. Mao, Z. Chen, A.C. Kwan, Y. Xie, D.S. Berman, D. Li, A.G. Christodoulou, Alternating low-rank tensor reconstruction for improved multiparametric mapping with cardiovascular mr multitasking, Magn. Reson. Med. 92 (4) (2024) 1421–1439.

[21] B.-Z. Li, X.-L. Zhao, J.-L. Wang, Y. Chen, T.-X. Jiang, J. Liu, Tensor completion via collaborative sparse and low-rank transforms, IEEE Trans. Comput. Imaging 7 (2021) 1289–1303.

[22] T. Yokota, A. Cichocki, Tensor completion via functional smooth component deflation, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, pp. 2514–2518.

[23] M. Imaizumi, K. Hayashi, Tensor decomposition with smoothness, in: International Conference on Machine Learning, PMLR, 2017, pp. 1597–1606.

[24] Y.-B. Zheng, T.-Z. Huang, T.-Y. Ji, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, Low-rank tensor completion via smooth matrix factorization, Appl. Math. Model. 70 (2019) 677–695.

[25] S. Mohaoui, A. Hakim, S. Raghay, Smooth tensor robust principal component analysis with application to color image recovery, Digit. Signal Process. 123 (2022) 103390.

[26] S. Mohaoui, A. Hakim, S. Raghay, Parallel matrix factorization-based collaborative sparsity and smooth prior for estimating missing values in multidimensional data, Pattern Anal. Appl. 25 (4) (2022) 963–980.

[27] T. Wu, J. Fan, Smooth tensor product for tensor completion, IEEE Trans. Image Process. (2024).

[28] Y.-Y. Liu, X.-L. Zhao, Y.-B. Zheng, T.-H. Ma, H. Zhang, Hyperspectral image restoration by tensor fibered rank constrained optimization and plug-and-play regularization, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–17.

[29] Y. Chen, T.-Z. Huang, W. He, X.-L. Zhao, H. Zhang, J. Zeng, Hyperspectral image denoising using factor group sparsity-regularized nonconvex low-rank approximation, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–16.

[30] T. Xie, L. Liu, L. Zhuang, Plug-and-play priors for multi-shot compressive hyperspectral imaging, IEEE Trans. Image Process. 32 (2023) 5326–5339.

[31] X. Wan, D. Li, Y. Lv, F. Kong, Q. Wang, Hyperspectral image reconstruction based on low-rank coefficient tensor and global prior, Int. J. Remote Sens. 44 (13) (2023) 4058–4085.

[32] J. Liu, M. Feng, X. Xiu, X. Zeng, J. Zhang, Tensor low-rank approximation via plug-and-play priors for anomaly detection in remote sensing images, IEEE Trans. Instrum. Meas. 74 (2025) 5503014.

[33] L. Guo, K. Gao, Z.-H. Huang, Low rank tensor recovery by Schatten capped p norm and plug-and-play regularization, Neurocomputing 534 (2023) 171–186.

[34] L. Sun, Q. Cao, Y. Chen, Y. Zheng, Z. Wu, Mixed noise removal for hyperspectral images based on global tensor low-rankness and nonlocal svd-aided group sparsity, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–17.

[35] B. Shen, R. Wang, A.C.C. Law, R. Kamath, H. Choo, Z. Kong, Super resolution for multi-sources image stream data using smooth and sparse tensor completion and its applications in data acquisition of additive manufacturing, Technometrics 64 (1) (2022) 2–17.

[36] J. Li, J. Shang, Y. Chen, Fourier low-rank and sparse tensor for efficient tensor completion, arXiv preprint arXiv:2505.11261, 2025.

[37] H. Luo, X. Liu, W. Liu, Y. Zhang, Low-rank reduced biquaternion tensor ring decomposition and tensor completion, Appl. Math. Comput. 504 (2025) 129504.

[38] R. Bian, C. Cai, L. Zhang, X. Zhang, Fast and provable tensor-train format tensor completion via preconditioned riemannian gradient descent, arXiv preprint arXiv:2501.13385, 2025.

[39] J. Xue, Y. Zhao, S. Huang, W. Liao, J.C.-W. Chan, S.G. Kong, Multilayer sparsity-based tensor decomposition for low-rank tensor completion, IEEE Trans. Neural Netw. Learn. Syst. 33 (11) (2021) 6916–6930.

[40] H. Kong, X. Xie, Z. Lin, T-schatten-p norm for low-rank tensor recovery, IEEE J. Sel. Top. Signal Process. 12 (6) (2018) 1405–1419.

[41] J. Xue, Y. Zhao, W. Liao, J.C.-W. Chan, S.G. Kong, Enhanced sparsity prior model for low-rank tensor completion, IEEE Trans. Neural Netw. Learn. Syst. 31 (11) (2019) 4567–4581.

[42] E. Martini, A. Calanca, N. Bombieri, Denoising and completion filters for human motion software: a survey with code, Comput. Sci. Rev. 58 (2) (2025) 100780.

[43] F.N. Fritsch, R.E. Carlson, Monotone piecewise cubic interpolation, SIAM J. Numer. Anal. 17 (2) (1980) 238–246.

[44] S.J. Howarth, J.P. Callaghan, Quantitative assessment of the accuracy for three interpolation techniques in kinematic analysis of human movement, Comput. Methods Biomech. Biomed. Eng. 13 (6) (2010) 847–855.

[45] H.E.A. Reda, I. Benaoumeur, B. Kamel, A.F. Zoubir, Mocap systems and hand movement reconstruction using cubic spline, in: 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE, 2018, pp. 1–5.

[46] J. Smołka, M. Skublewska-Paszkowska, Comparison of interpolation methods based on real human motion data, Prz. Elektrotech. 90 (10) (2014) 226–229.

[47] K. Dorfmüller-Ulhaas, Robust Optical User Motion Tracking Using a Kalman Filter, Ins. Für Informatik, Saarbrücken, Germany, Tech. Rep., 2003.

[48] Q.M. Wu, P. Boulanger, Real-time estimation of missing markers for reconstruction of human motion, in: 2011 IEEE Virtual Reality Conference (VR), IEEE, 2011, pp. 161–168.

[49] G. Liu, L. McMillan, Estimation of missing markers in human motion capture, Vis. Comput. 22 (2006) 721–728.

[50] M. Tits, J. Tilmanne, T. Dutoit, Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging, PLoS One 13 (7) (2018) e0199744.

[51] W. Li, S. Peng, Y. Liu, W. Wang, Predicting missing marker trajectories in human motion data using marker intercorrelations and Bayesian inference, PLoS One 11 (4) (2016) e0152616.

[52] W. Li, S. Peng, Y. Liu, W. Wang, A novel approach to solve the “missing marker problem” in marker-based motion capture, PLoS One 8 (11) (2013) e78689.

[53] X. Yang, A. Somasekharan, J.J. Zhang, Curve skeleton skinning for human and creature characters, Comput. Animat. Virtual Worlds 17 (3–4) (2006) 281–292.

[54] W. Cheng, R. Cheng, L. Xiaoyong, D. Shuling, Automatic skeleton generation and character skinning, in: 2011 IEEE International Symposium on VR Innovation, IEEE, 2011, pp. 299–304.

[55] L. Li, J. McCann, N. Pollard, C. Faloutsos, Bolero: a principled technique for including bone length constraints in motion capture occlusion filling, in: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2010, pp. 179–188.

[56] T. Kucherenko, J. Beskow, H. Kjellström, A neural network approach to missing marker reconstruction in human motion capture, arXiv preprint arXiv:1803.02665, 2018.

[57] O. Yuhai, A. Choi, Y. Cho, H. Kim, J.H. Mun, Deep-learning-based recovery of missing optical marker trajectories in 3D motion capture systems, Bioengineering 11 (6) (2024) 560.

[58] Q. Cui, H. Sun, Y. Li, Y. Kong, A deep bi-directional attention network for human motion recovery, in: IJCAI, 2019, pp. 701–707.

[59] I. Vernikos, E. Spyrou, Skeleton reconstruction using generative adversarial networks for human activity recognition under occlusion, Sensors 25 (5) (2025) 1567.

[60] M.D. Nguyen, T.L. Hoang, V.C. Ta, Reconstructing missing joints in 3D human motion with temporal-structural awareness graph neural network, in: 2023 15th International Conference on Knowledge and Systems Engineering (KSE), IEEE, 2023, pp. 1–6.

[61] W. Lee, S. Park, T. Kim, Denoising graph autoencoder for missing human joints reconstruction, IEEE Access 12 (2024) 57381–57389.

[62] J. Hou, L.-P. Chau, Y. He, J. Chen, N. Magnenat-Thalmann, Human motion capture data recovery via trajectory-based sparse representation, in: 2013 IEEE International Conference on Image Processing, IEEE, 2013, pp. 709–713.

[63] J. Xiao, Y. Feng, W. Hu, Predicting missing markers in human motion capture using l1-sparse representation, Comput. Animat. Virtual Worlds 22 (2–3) (2011) 221–228.

[64] G. Xia, H. Sun, G. Zhang, L. Feng, Human motion recovery jointly utilizing statistical and kinematic information, Inf. Sci. 339 (2016) 189–205.

[65] Gløersen and, P. Federolf, Predicting missing marker trajectories in human motion data using marker intercorrelations, PLoS One 11 (3) (2016) e0152616.

[66] Z. Li, H. Yu, H.D. Kieu, T.L. Vuong, J.J. Zhang, Pca-based robust motion data recovery, IEEE Access 8 (2020) 76980–76990.

[67] M.A.O. Vasilescu, Human motion signatures: analysis, synthesis, recognition, in: 2002 International Conference on Pattern Recognition, vol. 3, IEEE, 2002, pp. 456–460.

[68] P.A. Federolf, 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 8 (10) (2013) e78689.

[69] R.Y.Q. Lai, P.C. Yuen, K.K.W. Lee, Motion capture data completion and denoising by singular value thresholding, in: Eurographics (Short Papers), 2011, pp. 45–48.

[70] B. Chen, H. Sun, G. Xia, L. Feng, B. Li, Human motion recovery utilizing truncated Schatten p-norm and kinematic constraints, Inf. Sci. 450 (2018) 89–108.

[71] K. Kamali, A.A. Akbari, C. Desrosiers, A. Akbarzadeh, M.J.-D. Otis, J.C. Ayena, Low-rank and sparse recovery of human gait data, Sensors 20 (16) (2020) 4525.

[72] W. Hu, Z. Wang, S. Liu, X. Yang, G. Yu, J.J. Zhang, Motion capture data completion via truncated nuclear norm regularization, IEEE Signal Process. Lett. 25 (2) (2017) 258–262.

[73] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J.J. Zhang, R. Song, Exploiting temporal stability and low-rank structure for motion capture data refinement, Inf. Sci. 277 (2014) 777–793.

[74] Q. Cui, B. Chen, H. Sun, Nonlocal low-rank regularization for human motion recovery based on similarity analysis, Inf. Sci. 493 (2019) 57–74.

[75] C.-H. Tan, J. Hou, L.-P. Chau, Motion capture data recovery using skeleton constrained singular value thresholding, Vis. Comput. 31 (2015) 1521–1532.

[76] S.-J. Peng, G.-F. He, X. Liu, H.-Z. Wang, Hierarchical block-based incomplete human mocap data recovery using adaptive nonnegative matrix factorization, Comput. & Graph. 49 (2015) 10–23.

[77] S. Mohaoui, A. Dmytryshyn, CP decomposition-based algorithms for completion problem of motion capture data, Pattern Anal. Appl. 27 (4) (2024) 133.

[78] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.

[79] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.

[80] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, IEEE, 2013, pp. 171–184, no. 1.

[81] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.
[82] L. Zhang, M. Yang, X. Feng, A novel low-rank model for face recognition: robust matrix regression with sparse noise (rmr-sn), Inf. Sci. 281 (2014) 621–638.
[83] Z. Liu, Z. Wang, Q. Liu, S. Zhang, X. Zhang, Y. Wang, Y. Liang, Large-scale affine matrix rank minimization with a novel nonconvex regularizer: theory and algorithms, IEEE Trans. Neural Netw. Learn. Syst. 32 (3) (2021) 1167–1182.
[84] X. Zhou, C. Yang, H. Zhao, W. Yu, Low-rank modeling and its applications in image analysis, ACM Comput. Surv. 47 (2) (2014) 1–33.
[85] Z. Zha, B. Wen, X. Yuan, S. Ravishankar, J. Zhou, C. Zhu, Learning nonlocal sparse and low-rank models for image compressive sensing, IEEE Trans. Image Process. 31 (2022) 1.
[86] S. Ravishankar, Y. Bresler, Image recovery via transform learning and low-rank modeling: the power of compressibility priors, IEEE Signal Process. Mag. 35 (1) (2018) 22–35.
[87] Z. Liu, Z. Wang, Q. Liu, S. Zhang, Sparse inverse synthetic aperture radar imaging using structured low-rank matrix completion, IEEE Trans. Geosci. Remote Sens. (2020) 1–16.
[88] E. Candes, B. Recht, Exact matrix completion via convex optimization, Commun. ACM 55 (6) (2012) 111–119.
[89] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.
[90] C.-H. Tan, J. Hou, L.-P. Chau, Human motion capture data recovery using trajectory-based matrix completion, Electron. Lett. 49 (12) (2013) 752–754.
[91] M.S.S. Raj, S.N. George, A fast and efficient approach for human action recovery from corrupted 3-D motion capture data using QR decomposition-based approximate SVD, IEEE Trans. Hum.-Mach. Syst. 54 (4) (2024) 395–405.
[92] S. Mallat, A Wavelet Tour of Signal Processing, Elsevier, 1999.
[93] C. Torrence, G.P. Compo, A practical guide to wavelet analysis, Bull. Am. Meteorol. Soc. 79 (1) (1998) 61–78.
[94] R.J. Marks, Handbook of Fourier Analysis & Its Applications, Oxford University Press, 2009.
[95] J.D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition, Psychometrika 35 (3) (1970) 283–319.
[96] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.
[97] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317.
[98] Q. Zhao, G. Zhou, S. Xie, L. Zhang, A. Cichocki, Tensor ring decomposition, arXiv preprint arXiv:1606.05535, 2016.
[99] M. Fazel, H. Hindi, S.P. Boyd, A rank minimization heuristic with application to minimum order system approximation, in: Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148), vol. 6, IEEE, 2001, pp. 4734–4739.
[100] Y. Hu, D. Zhang, J. Ye, X. Li, X. He, Fast and accurate matrix completion via truncated nuclear norm regularization, IEEE Trans. Pattern Anal. Mach. Intell. 35 (9) (2012) 2117–2130.
[101] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, L. Zhang, Weighted schatten p-norm minimization for image denoising and background subtraction, IEEE Trans. Image Process. 25 (10) (2016) 4842–4857.
[102] L. Feng, H. Sun, Q. Sun, G. Xia, Image compressive sensing via truncated schatten-p norm regularization, Signal Process. Image Commun. 47 (2016) 28–41.
[103] M. Yin, J. Gao, Y. Guo, Nonlinear low-rank representation on Stiefel manifolds, Electron. Lett. 51 (10) (2015) 749–751.
[104] H. Nguyen, W. Yang, F. Shen, C. Sun, Kernel low-rank representation for face recognition, Neurocomputing 155 (2015) 32–42.
[105] S. Xiao, M. Tan, D. Xu, Z.Y. Dong, Robust kernel low-rank representation, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2015) 2268–2281.
[106] Z. Wen, W. Yin, Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, Math. Program. Comput. 4 (4) (2012) 333–361.
[107] W. Hu, Y. Lu, J. Ren, A fixed-point proximity algorithm for recovering low-rank components from incomplete observation data with application to motion capture data refinement, J. Comput. Appl. Math. 410 (2022) 114224.
[108] H. Huang, Q. Li, Y. Gao, W. Hu, L. Xu, J. Li, Multi-level fine-grained fusion based robust low-rank approximation for human motion data restoration, Neurocomputing (2025) 131587.
[109] J. Yang, X. Guo, K. Li, M. Wang, Y.-K. Lai, F. Wu, Spatio-temporal reconstruction for 3d motion recovery, IEEE Trans. Circuits Syst. Video Technol. 30 (6) (2019) 1583–1596.
[110] G. Xia, H. Sun, B. Chen, Q. Liu, L. Feng, G. Zhang, R. Hang, Nonlinear low-rank matrix completion for human motion recovery, IEEE Trans. Image Process. 27 (6) (2018) 3011–3024.
[111] W. Hu, X. Zhu, T. Wang, Y. Yi, G. Yu, Discrete subspace structure constrained human motion capture data recovery, Appl. Soft Comput. 129 (2022) 109617.
[112] S. Mohaoui, A. Dmytryshyn, Tucker Decomposition with a Temporal Regularization for Gap Recovery in 3D Motion Capture Data, 2025.
[113] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv preprint arXiv:1009.5055, 2010.
[114] D.P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 2014.
[115] E.G. Birgin, J.M. Martínez, Practical Augmented Lagrangian Methods for Constrained Optimization, SIAM, 2014.
[116] M.R. Hestenes, Multiplier and gradient methods, J. Optim. Theory Appl. 4 (5) (1969) 303–320.
[117] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.
[118] S. Ma, Alternating direction method of multipliers for sparse principal component analysis, J. Oper. Res. Soc. China 1 (2013) 253–274.
[119] Y. Xu, W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, SIAM J. Imaging Sci. 6 (3) (2013) 1758–1789.
[120] L. Grippo, M. Sciandrone, On the convergence of the block nonlinear gauss–seidel method under convex constraints, Oper. Res. Lett. 26 (3) (2000) 127–136.
[121] H. Attouch, J. Bolte, B.F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods, Math. Program. 137 (1) (2013) 91–129.
[122] B. Burger, P. Toiviainen, Mocap toolbox–a Matlab toolbox for computational analysis of movement data, in: Proceedings of the Sound and Music Computing Conference, Logos Verlag Berlin Stockholm, Sweden, 2013, pp. 172–178.