

3-D MOTION RECOVERY VIA LOW RANK MATRIX RESTORATION ON ARTICULATION GRAPHS

Kun Li¹, Meiyuan Wang¹, Yu-Kun Lai², Jingyu Yang^{1*}, Feng Wu³

¹ Tianjin University, Tianjin, China

² Cardiff University, Wales, UK

³ University of Science and Technology of China, Hefei, China

ABSTRACT

This paper addresses the challenge of 3-D skeleton recovery by exploiting the spatio-temporal correlations of corrupted 3-D skeleton sequences. A skeleton sequence is represented as a matrix. We propose a novel low-rank solution that effectively integrates both a low-rank model for robust skeleton recovery based on temporal coherence, and an articulation-graph-based isometric constraint for spatial coherence, namely consistency of bone lengths. The proposed model is formulated as a constrained optimization problem, which is efficiently solved by the Augmented Lagrangian Method with a Gauss-Newton solver for the subproblem of isometric optimization. Experimental results on the CMU motion capture dataset and a Kinect dataset show that the proposed approach achieves better recovery accuracy over a state-of-the-art method. The proposed method has wide applicability for skeleton tracking devices, such as the Kinect, because these devices cannot provide accurate reconstructions of complex motions, especially in the presence of occlusion.

Index Terms— skeleton, motion recovery, low rank, graph, occlusion

1. INTRODUCTION

This paper addresses the challenge of recovering accurate and smooth human motions from corrupted 3-D skeleton sequences, which is a fundamental problem in human motion reconstruction. Traditional motion capture systems are difficult to set up and involve substantial manual effort, which restricts their use. With the advent of Microsoft Kinect and similar devices, significant effort and advances [1, 2] have been made in recent years for low-cost, accessible human motion tracking systems. However, skeletons captured by low-cost devices such as Kinect often suffer from severe joint drifting and motion jitter, especially in the presence of self-occlusion or object occlusion [3].

Motion recovery is a critical problem in computer graphics and computer vision, and thus has attracted more and more interests. However, most work focuses on two main aspects, namely pose estimation from RGB images and depth images, and skeleton recovery from 2-D images. In the first aspect, there have been numerous classical works. Menier *et al.* [4] make use of foreground silhouette information to estimate skeletal poses. With the rise of neural networks and deep learning, increasing works use deep models to solve the human motion reconstruction problem. Ouyang *et al.* [5] fuse multiple information sources, including appearance score, mixture type and deformation into a deep model to estimate human poses. Toshev *et al.* [6] estimate human poses from RGB images by formulating it as a joint regression problem and solving it with a Deep Neural Network. Ever since the advent of Kinect-type depth cameras, the pose estimation problem can be better addressed by incorporating depth information. Wei *et al.* [7] develop an automatic motion capture system by integrating depth data, full-body geometry, *etc.* together using a single depth camera. Shotton *et al.* [8] introduce *body part classification* (BPC) and *offset joint regression* (OJR) algorithms to estimate human poses from a single depth image with robustness and efficiency. As for the second aspect, Bundle Adjustment (BA) has been a classical and popular algorithm. Leonardos *et al.* [9] introduce spherical tangent bundles and a Riemannian Extended Kalman Filter (REKF) model into the human motion reconstruction problem, achieving accurate reconstruction from image sequences with corrupted skeletons. Park *et al.* [10] reconstruct a 3-D smooth articulated trajectory from a 2-D trajectory using a monocular image sequence.

Little work focuses on straightforward recovery from corrupted 3-D skeleton sequences that can be obtained from Kinect. Wang *et al.* [11] achieve smooth 3-D skeleton recovery via low-rank matrix recovery. However, their model does not take *spatial* coherence into account, which can result in large joint errors in challenging cases. Saito *et al.* [12] recover corrupted skeletons by finding a subspace of valid motions, projecting corrupted skeletons onto the motion manifold and finally rebuilding valid motions through inverse pro-

*Corresponding author: yjy@tju.edu.cn. This work was supported in part by the National Natural Science Foundation of China (Grant 61571322 and 61372084).

jection. However, this method needs large amount of training data and a time-consuming training procedure.

In this paper, we propose a new 3-D motion recovery method from corrupted 3-D skeleton sequences. By exploiting both *spatial* and *temporal* correlations, our method is able to accurately recover clean and smooth skeleton motions. We introduce a new articulation-graph-based constraint to the low rank model, promoting isometry of bone lengths to ensure accurate recovery of joint positions. We also develop an efficient algorithm to solve the overall constrained optimization problem, combining a Gauss-Newton Solver and the Augmented Lagrangian Method. Through this method, the corrupted skeleton matrix with high proportion of noise and errors can be corrected. Experimental results on the CMU motion capture dataset [13] and a Kinect dataset show that the proposed approach achieves better recovery accuracy over a state-of-the-art method. Our code will be publicly available on the project website.

The remainder of this paper is organized as follows. Section 2 describes the proposed 3-D motion recovery method. Experimental results on both the CMU dataset and Kinect dataset are presented in Section 3, and finally the paper is concluded in Section 4.

2. THE PROPOSED METHOD

2.1. Background and Motivations

Given a sequence of corrupted 3-D skeletons as input, the aim of skeleton recovery is to produce a clean skeleton sequence without error. Let \mathbf{n}_i be the i -th joint of the skeleton. $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz})^T$, where n_{ix} , n_{iy} and n_{iz} represent the joint's x , y and z coordinates, respectively. $i \in \{1, 2, \dots, S\}$, where S is the number of skeleton nodes. Denote as \mathbf{n}_i^t the coordinates of the i -th joint in the t -th frame, and the total number of skeletons in the input sequence is T . The corrupted skeleton matrix $\mathbf{D} \in \mathbb{R}^{T \times 3S}$ is written as:

$$\mathbf{D} = \begin{pmatrix} \mathbf{n}_1^1 & \cdots & \mathbf{n}_S^1 \\ \vdots & \ddots & \vdots \\ \mathbf{n}_1^T & \cdots & \mathbf{n}_S^T \end{pmatrix}, \quad (1)$$

where each row contains all the coordinates of total S nodes in one skeleton. We assume an additive observation model:

$$\mathbf{D} = \mathbf{A} + \mathbf{E}, \quad (2)$$

where \mathbf{A} is the clean skeleton matrix, and \mathbf{E} represents error matrix. Human motions, as existing work suggests, are smooth and reasonable, and therefore should lie in a low-dimensional subspace [11]. By exploiting the temporal coherence of skeleton sequences, we can conclude that the rank of the motion matrix should be low. Skeleton corruption often happens in challenging scenarios such as occlusion, so the

error matrix should be sparse. Similar to [11], the 3-D skeleton recovery problem can be formulated as minimizing the following:

$$\begin{aligned} \min \text{rank}(\mathbf{A}) + \lambda \|\mathbf{E}\|_0 \\ \text{s.t. } \mathbf{D} = \mathbf{A} + \mathbf{E}, \end{aligned} \quad (3)$$

where $\text{rank}(\mathbf{A})$ is the rank of matrix \mathbf{A} , $\|\mathbf{E}\|_0$ is the ℓ_0 norm of matrix \mathbf{E} , representing the number of non-zero entries in the matrix and $\lambda > 0$ is a weighting parameter balancing the two terms.

The problem in Eq. (3) is NP-hard. This is made tractable by replacing $\text{rank}(\mathbf{A})$ with its convex substitute known as the nuclear norm $\|\mathbf{A}\|_* = \text{tr}((\mathbf{A}\mathbf{A}^T)^{1/2}) = \sum_i \sigma_i$, where σ_i is a singular value of matrix \mathbf{A} , and replacing ℓ_0 norm of matrix \mathbf{E} with its ℓ_1 norm $\|\mathbf{E}\|_1 = \sum_{ij} |\mathbf{E}_{ij}|$. We obtain a semidefinite programming problem [14]:

$$\begin{aligned} \min \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 \\ \text{s.t. } \mathbf{D} = \mathbf{A} + \mathbf{E}. \end{aligned} \quad (4)$$

Proper selection of the parameter λ is crucial to recovery accuracy [15]. Under our circumstance, λ should be small enough to remove noise (by keeping the variance low to obtain high stability), and large enough not to overshrink the original matrix (by keeping the bias low for flexible motion).

The above low rank term implies linear dependence of rows/columns of the matrix. However, the dependence is unstructured, due to the intrinsic ambiguity of corrupted joints. Articulated skeleton is usually described by a tree-structure, where each node represents a single skeleton joint and each edge between nodes represents a bone. This structure can be described as an articulation graph. On the other hand, bones have constant length for a specific person at all time instances. Therefore, we consider such a constant distance constraint for spatial coherence of skeleton sequences, and propose a low rank matrix recovery model on articulation graphs.

2.2. Skeleton Recovery on Articulation Graphs

We use articulation graphs to exploit spatial coherence of skeletons by promoting isometry (*i.e.* length preservation) of bones. More formally, let $G = (\mathcal{V}, \mathcal{E})$ be an undirected articulation graph, where \mathcal{V} is the set of graph vertices representing the skeleton joints, and \mathcal{E} is the set of graph edges corresponding to the bones. e_k represents the k -th bone of the skeleton, $k \in \{1, 2, \dots, H\}$, and H is the number of skeleton bones. Since for a specific person, the length of every bone remains the same over time, we introduce an energy term that penalizes non-isometric deformation:

$$E_{iso}(\mathbf{N}) = \sum_{t=1}^T \sum_{e_k=(i,j) \in \mathcal{E}} [d^2(\mathbf{n}_i^t, \mathbf{n}_j^t) - l_{ij}^2]^2, \quad (5)$$

where e_k connects the i -th and the j -th joints, and l_{ij} is the bone length between these two joints.

$d(\mathbf{n}_i^t, \mathbf{n}_j^t)$ denotes the distance between joints \mathbf{n}_i^t and \mathbf{n}_j^t in the skeleton matrix, and $d(\mathbf{n}_i^t, \mathbf{n}_j^t) = \sqrt{(n_{ix}^t - n_{jx}^t)^2 + (n_{iy}^t - n_{jy}^t)^2 + (n_{iz}^t - n_{jz}^t)^2}$. Combining the low-rank term with the isometry term, 3-D motion recovery is formulated as:

$$\begin{aligned} \min \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 + \frac{\gamma}{2} E_{iso}(\mathbf{N}) \\ \text{s.t. } \mathbf{D} = \mathbf{A} + \mathbf{E}, \quad \mathbf{N} = \mathbf{A}, \end{aligned} \quad (6)$$

where \mathbf{N} is a substitute of matrix \mathbf{A} for the convenience during computation procedure. $\gamma > 0$ is a weighting parameter balancing the importance of the isometry term.

Low rank promoted by the nuclear norm implies sparsity in the space of outer products of the singular vectors, which means the skeleton matrix \mathbf{A} can be expressed as linear combinations of some key poses. Similarly, the isometry term as a constraint of skeleton bone lengths, can be seen as the guarantee of recovered positions of skeleton joints. Overall, the whole model promotes simultaneous sparsity of the skeleton matrix both in the singular vectors' outer product space and isometry space.

2.3. Augmented Lagrangian Algorithm Combined with a Gauss-Newton Solver

The proposed model Eq. (6) is a minimization function with equality constraints. Many practical algorithms can solve this kind of minimization problems, and we choose the augmented Lagrangian method [16] to handle the problem under an iterative framework due to its high efficiency and accuracy. However, the isometry term Eq. (5) is a non-linear least squares problem. In this paper, we use Gauss-Newton solver to alternatively optimize the isometry term with other terms fixed at each ALM iteration. The augmented Lagrangian of Eq. (6) is

$$\begin{aligned} L(\mathbf{A}, \mathbf{E}, \mathbf{N}, \mathbf{Z}_1, \mathbf{Z}_2, \rho_1, \rho_2) = \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 + \frac{\gamma}{2} E_{iso}(\mathbf{N}) \\ + \langle \mathbf{Z}_1, \mathbf{E} - \mathbf{D} + \mathbf{A} \rangle + \frac{\rho_1}{2} \|\mathbf{E} - \mathbf{D} + \mathbf{A}\|_F^2 \\ + \langle \mathbf{Z}_2, \mathbf{N} - \mathbf{A} \rangle + \frac{\rho_2}{2} \|\mathbf{N} - \mathbf{A}\|_F^2, \end{aligned} \quad (7)$$

where $\|\mathbf{M}\|_F$ represents the Frobenius norm of a matrix \mathbf{M} , \mathbf{Z}_1 and \mathbf{Z}_2 are Lagrangian multipliers, and $\langle \cdot, \cdot \rangle$ denotes the inner product of two matrices considered as long vectors. Then, we iteratively solve three subproblems: $\min_{\mathbf{A}} L(\mathbf{A}, \mathbf{E}, \mathbf{N}, \mathbf{Z}_1, \mathbf{Z}_2)$, $\min_{\mathbf{E}} L(\mathbf{A}, \mathbf{E}, \mathbf{N}, \mathbf{Z}_1, \mathbf{Z}_2)$ and $\min_{\mathbf{N}} L(\mathbf{A}, \mathbf{E}, \mathbf{N}, \mathbf{Z}_1, \mathbf{Z}_2)$, respectively. The former two problems can be solved by pre-demonstrated methods [14]:

$$\begin{aligned} \arg \min_{\mathbf{A}} L(\mathbf{A}, \mathbf{E}, \mathbf{N}, \mathbf{Z}_1, \mathbf{Z}_2) &= S_{\frac{\lambda}{\rho_1}} \left(\mathbf{D} - \mathbf{A} - \frac{1}{\rho_1} \mathbf{Z}_1 \right), \\ \arg \min_{\mathbf{E}} L(\mathbf{A}, \mathbf{E}, \mathbf{N}, \mathbf{Z}_1, \mathbf{Z}_2) &= \\ M_{\frac{1}{\rho_1 + \rho_2}} \left(\frac{\rho_1 \mathbf{D} - \rho_1 \mathbf{E} - \mathbf{Z}_1 + \rho_2 \mathbf{N} + \mathbf{Z}_2}{\rho_1 + \rho_2} \right), \end{aligned} \quad (8)$$

where $S_{\delta}(x) = \text{sgn}(x) \max(|x| - \delta, 0)$ is a shrinkage operator and $M_{\delta}(\mathbf{X}) = \mathbf{U} S_{\delta}(\mathbf{A}) \mathbf{V}$ is a singular value thresholding operator.

The third problem is nonlinear and cannot be solved by the same strategy. Assuming δ^k is the update of \mathbf{N} at step k :

$$\mathbf{N}^{k+1} = \mathbf{N}^k + \delta^k. \quad (9)$$

Minimizing $E_{iso}(\mathbf{N})$ w.r.t. the unknown parameters is a non-linear least squares problem that can be rewritten as:

$$E_{iso}(\mathbf{N}) = \sum_{t=1}^T \sum_{h=1}^H r_{th}(\mathbf{N})^2, \quad (10)$$

where $r_{th}(\cdot)$ is the energy term related to the h -th bone of the t -th frame. We reformulate our objective $E_{iso}(\mathbf{N})$ in terms of its residual vector \mathbf{F} to obtain the classical Gauss-Newton form:

$$E_{iso}(\mathbf{N}) = \|\mathbf{F}(\mathbf{N})\|^2, \quad \mathbf{F}(\mathbf{N}) = [r_{11}(\mathbf{N}), \dots, r_{TH}(\mathbf{N})]^T.$$

Linearizing $\mathbf{F}(\mathbf{N})$ using Taylor expansion yields:

$$\mathbf{F}(\mathbf{N}^{k+1}) \approx \mathbf{F}(\mathbf{N}^k) + \mathbf{J}(\mathbf{N}^k) \delta, \quad \delta = \mathbf{N}^{k+1} - \mathbf{N}^k, \quad (11)$$

where $\mathbf{J}(\mathbf{N}^k)$ is the Jacobian of \mathbf{F} evaluated at the solution after k iterations, and δ is the iteration step. The resulting optimization problem is a linear least squares problem:

$$\delta^* = \arg \min_{\delta} \|\mathbf{F}(\mathbf{N}^k) + \mathbf{J}(\mathbf{N}^k) \delta\|^2. \quad (12)$$

We compute the optimal update step δ^* as the solution of the corresponding normal equation:

$$\mathbf{J}(\mathbf{N}^k)^T \mathbf{J}(\mathbf{N}^k) \delta = -\mathbf{J}(\mathbf{N}^k)^T \mathbf{F}(\mathbf{N}^k), \quad (13)$$

which can be solved jointly on the complete domain using iterative solution techniques like preconditioned conjugate gradient (PCG). Previous works [17, 18] demonstrate the feasibility of this strategy in a GPU optimization framework for dynamics simulation and non-rigid registration, respectively. For our problem, we can use the ideology of Gauss-Newton solver as an iteration step in the augmented Lagrangian method.

Combining Eq. (7) with the Gauss-Newton solver, the unknown update δ^k can be solved by:

$$\delta^k = (\gamma \mathbf{J}^T \mathbf{J} + \rho_2^k \mathbf{I})^{-1} [-\mathbf{Z}_2^k - \rho_2^k (\mathbf{N}^k - \mathbf{A}^k) - \gamma \mathbf{J}^T r(\mathbf{N}^k)], \quad (14)$$

where \mathbf{I} is the identity matrix. The overall ALM algorithm is summarized in Algorithm 1.

Algorithm 1: ALM Combined with Gauss-Newton Solver

- 1: **Input:** observed skeleton matrix $\mathbf{D} \in \mathbf{R}^{m \times n}$
 - 2: **Initialize:** $\mathbf{A}^0 = \mathbf{0}, \mathbf{E}^0 = \mathbf{0}, \mathbf{N}^0 = \mathbf{D}, \mathbf{Z}_1^0 = \mathbf{0}, \mathbf{Z}_2^0 = \mathbf{0}$
 $\rho_1, \rho_2 > 0, \eta_1, \eta_2 > 1, \text{maxIter} = 1000$
 - 3: **while not converged do**
 - 4: $\delta^k = (\gamma \mathbf{J}^T \mathbf{J} + \rho_2^k \mathbf{I})^{-1}$
 $\times [-\mathbf{Z}_2^k - \rho_2^k (\mathbf{N}^k - \mathbf{A}^k) - \gamma \mathbf{J}^T r(\mathbf{N}^k)]$
 - 5: $\mathbf{N}^{k+1} = \mathbf{N}^k + \delta^k$
 - 6: $\mathbf{E}^{k+1} = S \frac{\lambda}{\rho_1^k} \left(\mathbf{D} - \mathbf{A}^k - \frac{1}{\rho_1^k} \mathbf{Z}_1^k \right)$
 - 7: $\mathbf{A}^{k+1} = M \frac{1}{\rho_1^k + \rho_2^k} \left(\frac{\rho_1^k \mathbf{D} - \rho_1^k \mathbf{E}^{k+1} - \mathbf{Z}_1^k + \rho_2^k \mathbf{N}^{k+1} + \mathbf{Z}_2^k}{\rho_1^k + \rho_2^k} \right)$
 - 8: $\mathbf{Z}_1^{k+1} = \mathbf{Z}_1^k + \rho_1^k (\mathbf{E}^{k+1} - \mathbf{D} + \mathbf{A}^{k+1})$
 - 9: $\mathbf{Z}_2^{k+1} = \mathbf{Z}_2^k + \rho_2^k (\mathbf{N}^{k+1} - \mathbf{A}^{k+1})$
 - 10: $\rho_1^{k+1} = \eta_1 \rho_1^k, \eta_1 > 1$
 - 11: $\rho_2^{k+1} = \eta_2 \rho_2^k, \eta_2 > 1$
 - 12: **End while**
 - 13: **Output:** \mathbf{A}, \mathbf{E}
-

3. EXPERIMENTAL RESULTS

In this section, our method is evaluated on both the CMU mocap dataset [13] and a real captured Kinect v2.0 dataset. Each skeleton in the CMU dataset contains 25 skeleton joints (with finger joints removed) and 24 bones, while the Kinect dataset has 21 joints (also without finger joints) and 20 bones. We set $\lambda = 0.04$ and $\gamma = 10$ in the experiments. Both quantitative and qualitative results are presented. We apply our method to the corrupted skeleton sequence and compare the recovery result with the method in [11] which also does not require prior data for training. For quantitative evaluation, the metric of recovery error is formulated as: $\omega = \frac{1}{ST} \sum_{t=1}^T \sum_{p=1}^S d(\tilde{\mathbf{n}}_p^t, \mathbf{n}_p^t)$, where $\tilde{\mathbf{n}}_p^t$ and \mathbf{n}_p^t are the ground truth and reconstructed joint positions. ω represents the average absolute difference over joints in all the frames, known as Average Joint Error (AJE).

3.1. CMU Dataset

We present quantitative results on the CMU dataset as the skeleton data can be used as ground truth for evaluation. To this end, we simulate corruptions into the skeleton data. Specifically, random noise is added to a fraction of entities in the ground truth skeleton matrix $\tilde{\mathbf{A}}$, obtaining the observed skeleton matrix \mathbf{D} . Five different percentages of polluted entities in \mathbf{D} , i.e., 5%, 10%, 15%, 20%, 80%, are tested. The last set is tested to validate our efficiency in large corrupted percentage situation, and the former four sets are tested for situations similar to real captured data. The noise in the polluted joints is uniformly distributed in the range of [-25 25] cm in

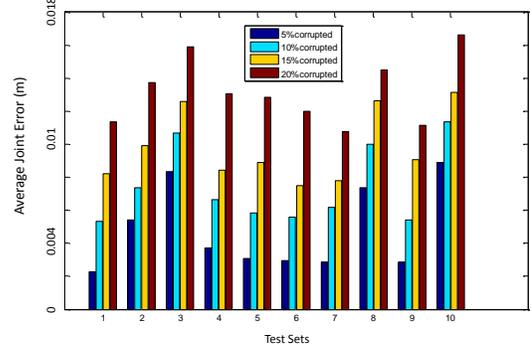


Fig. 1. Average joint error in meters of different sequences of subject 09.

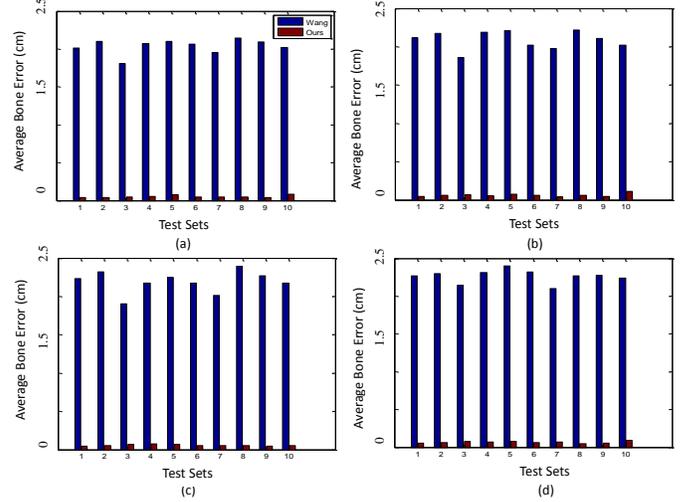


Fig. 2. Average bone length error in centimeters of subject 09 motion sequences with corruption percentages of (a) 5%, (b) 10%, (c) 15%, and (d) 20%.

each spatial dimension. This range is selected according to the average length of arms and legs as noisy joints are unlikely to go beyond this range. Recovery errors are presented in Fig. 1. We observe that, for different motion sequences of the same subject, our method can achieve consistent recovery errors with the same proportion of corruption. When 5% of the skeleton matrix is corrupted, our method can recover the motion sequence within 0.01 m AJE.

We also test recovery performance of several skeleton sequences from different subjects with various motions and frame lengths. Specifically, we use sequences from subject 05 to subject 140, including a variety of actions such as running, bending, dancing, *etc.* Table 1 presents comparative results and the overall AJE for all the sequences tested. Our method can reconstruct much more accurate motions compared with the method in [11] for the entire range of corruption levels. Even when 80% of the skeleton matrix is corrupted, our method can still reconstruct reasonable motion within 0.08 m AJE. Apart from the AJE, we also measure the average bone length error in centimeters for both methods. As Fig. 2 shows,

Table 1. Average joint error (m) comparison for skeleton recovery on CMU dataset with different corruption percentages.

Sub.	Method	5%	10%	15%	20%	80%
05	[11]	0.066	0.070	0.073	0.075	0.093
	Ours	0.002	0.005	0.008	0.011	0.074
09	[11]	0.076	0.079	0.083	0.085	0.122
	Ours	0.005	0.007	0.010	0.014	0.073
13	[11]	0.073	0.076	0.078	0.085	0.105
	Ours	0.008	0.011	0.012	0.016	0.072
24	[11]	0.069	0.075	0.078	0.082	0.114
	Ours	0.004	0.007	0.008	0.013	0.075
56	[11]	0.068	0.073	0.077	0.080	0.118
	Ours	0.003	0.006	0.009	0.013	0.074
86	[11]	0.072	0.075	0.077	0.083	0.121
	Ours	0.003	0.006	0.008	0.012	0.076
93	[11]	0.068	0.071	0.075	0.081	0.098
	Ours	0.003	0.008	0.010	0.011	0.081
115	[11]	0.072	0.076	0.078	0.085	0.130
	Ours	0.007	0.005	0.012	0.015	0.062
140	[11]	0.061	0.063	0.070	0.080	0.108
	Ours	0.003	0.005	0.010	0.012	0.053
Total	[11]	0.068	0.073	0.079	0.081	0.104
	Ours	0.005	0.007	0.009	0.012	0.073

our method generates the recovered skeletons with more accurate bone lengths due to the isometry constraint.

For qualitative evaluation, the visual performance of our method against [11] is tested with 5%, 10%, 15%, 20% and 80% of the skeleton matrix being corrupted. As shown in Fig. 3, the method in [11] recovers reasonable skeleton motions by sacrificing certain motion details. The recovered motions are more rigid compared with ground truth due to the low-rank constraint. On the contrary, our method can achieve more accurate and smooth reconstruction of corrupted skeleton motions with substantial details thanks to the isometry term. See also the video demo for dynamic results.

The running times for the CMU dataset are given in Table 2. All the experiments are performed on a desktop computer with Intel i5-4690K 3.5GHz CPU and 8GB RAM.

Table 2. The running times on the CMU dataset

Skeleton Sequence	Number of Frames	Running Time (s)
Sub.21 Seq.03	274	22.7150
Sub.115 Seq.05	585	47.5670
Sub.140 Seq.04	1103	96.0281
Sub.56 Seq.06	6784	478.3740

3.2. Kinect Dataset

In realistic circumstances, human motions captured by most low-cost mocap devices such as Kinect can suffer from severe joint drifting and motion jitter. To validate the applicability of our method, we also conduct experiments on a real captured Kinect dataset. In this case, no ground-truth data is available, so we estimate the median of the bone lengths using the first several frames as the credible bone length. Because the actor starts from the ‘‘T’’ pose, and the motions in these initial

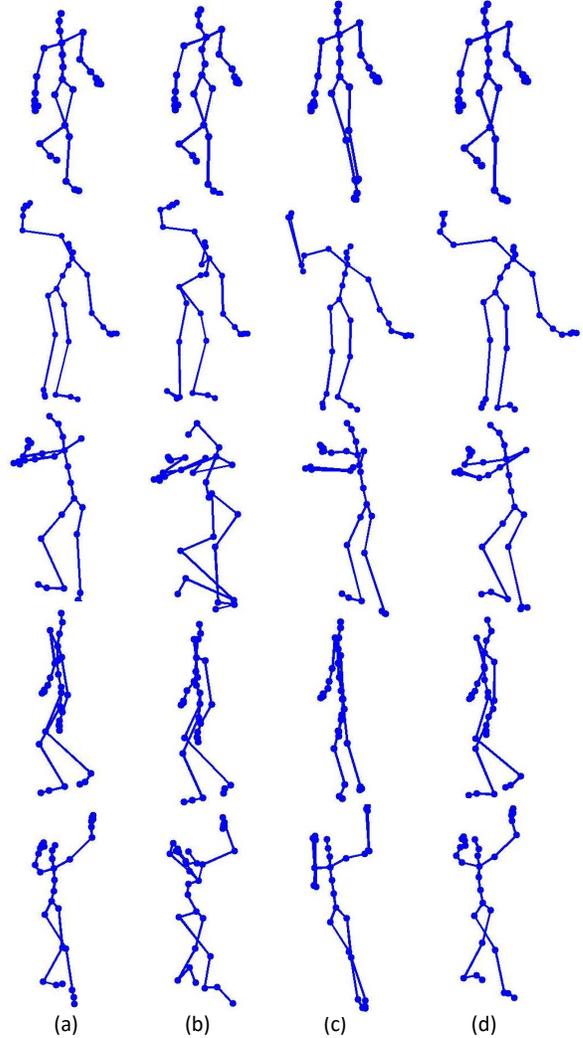


Fig. 3. Comparison results for frame 54, 648, 2046, 3071, and 4290 of subject 05 (from top to bottom): (a) ground-truth skeleton, (b) damaged skeleton, (c) recovered skeleton by [11], and (d) recovered skeleton by our method.

frames are very small. As shown in Fig. 4, the method in [11] reconstructs reasonable motions from the corrupted skeletons but the results are too rigid and lose some details, similar to Fig. 3. Our method recovers reasonable and detailed motions. Dynamic results are presented in our demo video.

4. CONCLUSION

This paper proposes a novel skeleton recovery method from corrupted 3-D skeleton sequences by utilizing both temporal and spatial constraints, which achieves accurate and smooth recovery of damaged motions. We introduce an isometry constraint based on articulation graphs to ensure consistent bone length, and a Gauss-Newton solver for the nonlinear least squares subproblem. Experimental results demonstrate the effectiveness of our method compared with a state-of-the-art

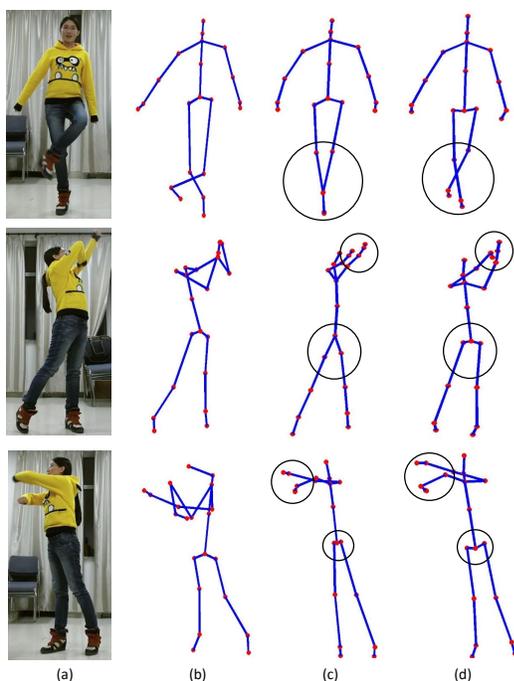


Fig. 4. Comparison results for frame 803 of sequence 1, frame 155 of sequence 3 and frame 458 of sequence 3 (from top to bottom): (a) captured color image, (b) captured Kinect skeleton, (c) recovered skeleton by [11], and (d) recovered skeleton by our method.

method, and also show its usefulness for improving skeletons obtained from Kinect-type motion capture devices. We will improve the slight motion jitter existing in some cases in our future work.

5. REFERENCES

- [1] B. Wang, Z. Chen, and J. Chen, "Gesture recognition by using kinect skeleton tracking system," in *Intl. Conf. Intel. Human-Machine Sys. and Cyb.*
- [2] D. S. Alexiadis and P. Daras, "Quaternionic signal processing techniques for automatic evaluation of dance performances from MoCap data," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1391–1406, 2014.
- [3] S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population," in *IEEE Engineering in Medicine and Biology Society*, 2012, pp. 1188–1193.
- [4] C. Menier, E. Boyer, and B. Raffin, "3D skeleton-based body pose recovery," in *Intl. Symp. 3D Data Processing Visualization and Transmission*, 2006, pp. 389–396.
- [5] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *CVPR*, 2014, pp. 2337–2344.
- [6] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *CVPR*, 2013, pp. 1653–1660.
- [7] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 439–445, 2012.
- [8] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, and A. Kipman, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Analys. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, 2013.
- [9] S. Leonardos, X. Zhou, and K. Daniilidis, "Articulated motion estimation from a monocular image sequence using spherical tangent bundles," in *IEEE Intl. Conf. Robotics and Automation*, 2016.
- [10] H. S. Park and Y. Sheikh, "3D reconstruction of a smooth articulated trajectory from a monocular image sequence," in *ICCV*, 2011, pp. 201–208.
- [11] M. Wang, K. Li, F. Wu, Y.-K. Lai, and J. Yang, "3-D motion recovery via low rank matrix analysis," in *IEEE Vis. Comm. Image Processing*, 2016, pp. 1–4.
- [12] J. Saito, D. Holden, and T. Komura, "Learning motion manifolds with convolutional autoencoders," in *SIG-GRAPH Asia Technical Briefs*, 2015.
- [13] "Carnegie-Mellon Mocap database," <http://mocap.cs.cmu.edu/>.
- [14] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.
- [15] J. Wright, A. Ganesh, S. Rao, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *NIPS*, 2009, pp. 2080–2088.
- [16] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv: :1009.5055*, 2010.
- [17] M. Zollhöfer, J. Thies, M. Colaianni, M. Stamminger, and G. Greiner, "Interactive model-based reconstruction of the human head using an RGB-D sensor," *Comp. Anim. Virt. Worlds*, vol. 25, no. 3-4, pp. 213–222, 2014.
- [18] W. Daniel, B. Jan, S. Markus, S. André, and F. Dieter, "Efficient GPU data structures and methods to solve sparse linear systems in dynamics applications," *Computer Graphics Forum*, vol. 32, no. 1, pp. 16–26, 2013.