The Visual Computer (2021) 37:2699–2711
https://doi.org/10.1007/s00371-021-02250-y

ORIGINAL ARTICLE

3D hand reconstruction from a single image based on biomechanical constraints

Guiqing Li[1] · Zihui Wu[1] · Yuxin Liu[1] · Huiqian Zhang[1] · Yongwei Nie[1] · Aihua Mao[1]

Accepted: 4 July 2021 / Published online: 27 July 2021

Abstract
This paper investigates the estimate of motion parameters from 3D hand joint positions. We formulate the issue as an inverse kinematics problem with biomechanical constraints and propose a fast and robust iterative approach to address the constrained optimization. It elaborately designs a coordinate descent algorithm to decompose the problem into a sequence of decisions on the transformation around each kinematic node (i.e., joint), while the decision for each node is equivalent to a point matching problem. Addressing the whole optimization then amounts to considering all nodes of the kinematic tree from its root to leaves one by one. This not only accelerates the process but also improves the accuracy of the solution of the inverse kinematic optimization. Experiments show that our approach is able to yield results comparable to and even better than those by the state-of-the-art methods.

Keywords 3D hand motion reconstruction · Biomechanical constraints · Block coordinate descent · MANO parameterization

1 Introduction

Hands play an indispensable role in human daily life interactions. Reconstruction of 3D hand geometries has key significance in a variety of computer graphics applications such as computer animation, 3D game, virtual reality (VR), augmented reality (AR) and human–computer interaction [2,10,49].

A great number of approaches have been explored for estimating 3D hand joint positions in the literature. Early work generally makes good use of optical markers [34,54] or glove techniques [12,26,47] which are capable of recovering joint positions with high fidelity. Recent development shows that deep neural networks (DNNs) are very promising to reconstruct 3D hand poses from RGB/D images taken by consumer-level cameras [40,42,45,52,56], although it is still a challenging task for these approaches to predict 3D joint positions with accuracy comparable to traditional methods due to complexity and occlusion.

Joint positions should further be converted to motion parameters for most computer graphics applications such as animation and virtual/mixed reality. Traditional approaches usually leverage the inverse kinematic technique to estimate, while recent works resort to deep neural networks to regress the parameters. However, the former are usually time-consuming due to the nonlinear optimization, while the latter are required to further improve in accuracy.

We propose a stable, accurate and fast enough method to estimate motion parameters from given 3D hand joint positions. The problem is formulated as an nonlinear optimization with finger movement range constraints. To force the estimate falling within the hand motion space, we exploit hand biomechanical constraints to restrict the rotation of hand joints within a specific range. We then attack the optimization with a block coordinate descent method by extremely decomposing it into a set of optimizations for a single joint rotation. Specifically, we alternatively optimize each coordinate on the kinematic chain from the root joint to leaf joints.

✉ Aihua Mao
ahmao@scut.edu.cn

Guiqing Li
ligq@scut.edu.cn

Zihui Wu
437898809@qq.com

Yuxin Liu
465367868@qq.com

Huiqian Zhang
791771249@qq.com

Yongwei Nie
nieyongwei@scut.edu.cn

[1] South China University of Technology, Guangzhou, China

When optimizing the motion parameters of a joint, we fix those of all other joints. This not only helps deal with motion constraints but also makes it possible to derive closed-form solutions. To sum up, our approach has the following contributions:

- The biomechanical constraints are introduced to reduce the solution space of hand poses. It not only makes it easier to obtain reasonable hand poses but also helps accelerate the solution of the optimization.
- A block coordinate descent algorithm is designed to solve the optimization, which cooperates with the hand motion constraints to fit the MANO model to 2D or 3D joint positions.
- A variety of experiments show that our method can obtain more accurate motion parameters and reasonable hand poses than the state-of-the-art approaches.

2 Related works

There are a variety of disciplines to acquire 3D hand shape and pose meshes in the literature. According to input data, 3D hand mesh models can be reconstructed from a single image, multiview images, videos, 3D information (by scanners or glove sensors). We mainly focus on image-based modeling approaches from two aspects according to whether involving parametric models or not. For a comprehensive review, one can refer to the survey by Ahmad et al. [1].

2.1 Nonparametric 3D hand reconstruction

2.1.1 Direct hand pose and shape reconstruction

Early methods acquire dense point clouds of hands either by scanning or via multiview image reconstruction. Less works are particularly contributed to hand reconstruction in this category [21,33] because of the difficulty of combining hand priors.

The introduction of machine learning approaches has changed this situation [6]. Given a hand image, Kulon et al. [22] employ an encoder to extract its latent code and then generate a 3D hand mesh. Ge et al. [11] build graph CNNs to recover 3D hand models from images. Peng et al. [21,33] leverage a three-stage and coarse-to-fine GCN to regress the vertex coordinates of the hand mesh. Nevertheless, these approaches are still in the stage of preliminary exploration and usually difficult to achieve high accuracy.

2.1.2 Hand pose reconstruction based on database retrieval

By building a database of different 3D hand shapes and poses as well as their different view images, one can retrieve an approximatory mesh model in the database for a given image [3,16,37]. Miyamoto et al. [28] propose a tree structure to speed up the database retrieval process. Besides, to improve the matching accuracy, Imai et al. [17] introduce a mismatched likelihood index. Wang et al. [47] build an image dataset of hands wearing customized color gloves. It is difficult to include all possible poses in a database due to the diversity of hand poses. This makes this kind of approaches hard to achieve high accuracy.

2.2 3D hand reconstruction based on parametric models

This category assumes a hand parametric model has been built. It only needs to estimate shape and motion parameters for reconstructing 3D hands. In the following, we first briefly recall some hand parametric models and then review the reconstruction methods.

2.2.1 Hand parametric models

Inspired by the idea of linear blend skinning (LBS) for human body [25], a number of parametric models are proposed to represent hand shape and pose [49]. For example, Bray et al. [5] create an LBS for hand with a mesh template of 9051 vertices and a skeleton of 30 degrees of freedom. Oikonomidis et al. [30] substitute 37 cylinders and spheres for the mesh template to approximate hand shape and pose for hand tracking. Melax et al. [27] further reduce the geometry to convex polyhedra for fast skeletal hand tracking.

Wheatland et al. [48] perform PCA on the American sign language database to extract pose PCA bases in order to reduce the dimensionality of the hand motion space. A more popular model is MANO by Romero et al. [38], which adds a set of shape parameters to the LBS model in order to capture the hand shape of different individual subjects. The PCA technique is combined to simplify the model. Qian et al. [36] consummate MANO by augmenting it with a parametric texture model.

Different skeletal structures include different number of joints. For example, NYU lab [45] builds a dataset based on 14 joints, the ICVL dataset [44] adopts 16 joints, and the MSRA dataset [35] involves 21 joints. Most parametric models adopt 21 joints to describe hand motion [38].

2.2.2 Hand pose reconstruction

Most of earlier methods aim at optical-marker-based motion capture, which are able to obtain joint positions of high accuracy. Fitting the LBS to these tracked joints is the so-called inverse kinematics (IK) problem. Numerical methods such as Jacobian inverse technique [20] are usually employed to address the issue. The core idea is to model the forward

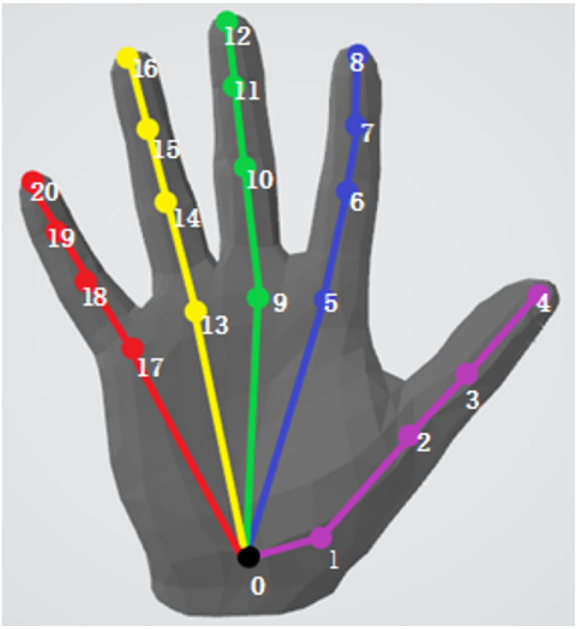

Fig. 1 Illustration of MANO hand model [38]: the hand mesh as well as its skeleton tree in which color segments and circles, respectively, illustrate the bones and joints of the hand skeleton

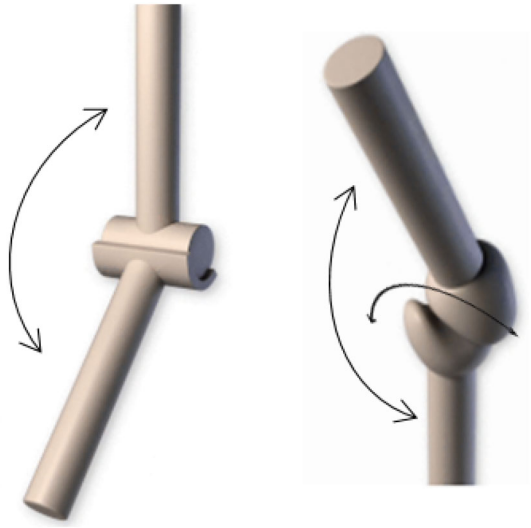

Fig. 2 Hinge joint (left) and saddle joint (right)

kinematics equation using Taylor expansion to simplify the solution.

To improve the accuracy of joint prediction, some works [15,23,43,54] introduce additional image features such as edges, silhouettes and textures to guide the pose fitting. IK becomes an energy term of a more general minimization problem. La Gorce et al. [23] employ a quasi-Newton method to tackle the optimization, while Zhao et al. [54] apply the particle swarm algorithm to address the issue.

Both Xiang et al. [51] and Pavlakos et al. [31] take advantage of the gradient descent algorithm to address the optimization mixing constraints of joint positions, image features as well as priors to recover whole body shape/pose from a single image. The algorithm may trap in local minima because of improper initialization and suffer low efficiency due to the complexity of the search space. Making use of PCA to reduce the space of motion parameters can accelerate the process but is hard to support joint constraints due to lack of semantic meanings [38,48]. Instead of resorting to image features, we introduce hand biomechanical constraints on the optimization and solve it with an iterative coordinate descent algorithm. We clearly define the motion space of hand joints by considering the degree of freedom (DOF) of every joint according to the hand biomechanical constraints. This not only makes our results approximate the valid pose as much as possible but also increases the efficiency and stability of the solution.

Deep neural networks are also employed to address parametrization-driven hand pose reconstruction. Zhou et al. [55] use existing motion capture data to train a six-layer perceptron (MLP) to regress 2D joint positions and 3D joint angles. Zhang et al. [53] and Boukhayma et al. [4], respectively, propose an end-to-end neural network to predict motion parameters with 2D image as input. Qian et al. [36] leverage the network in [4] to obtain the motion parameters of MANO and further to refine the mesh model by photometric loss. Other deep learning-based approaches either leverage depth information [29,32] or combine image and depth information together. As pose data usually distributes near the mean pose of the dataset, neural networks are prone to smooth the pose and make the result near the mean [41].

3 Preliminaries

For convenience of description, we first introduce the formulation of MANO and then present the notion of hand anatomical kinematics as preliminaries.

3.1 MANO

MANO (hand Model with Articulated and Non-rigid deformations) [38] is a hand parametric model built after SMPL [24], a human body parametric model. Like SMPL, MANO employs a group of PCA bases to capture the shape variation of specific hands and the skeleton skinning technique [25] to generate motion gestures of the specific hand. The skeleton used in MANO has 21 joints [35,44,45], as shown in Fig. 1 in which color line segments represent bones (edges of the tree) and color circles indicate joints. Vertices of the hand mesh in Fig. 1 are evaluated using MANO [38].

Denote the average hand mesh in rest (zero) pose by $\overline{T} =< \overline{V}, E, F >$, where $\overline{V} = \{\overline{v}_i, i = 1, \ldots, N\}$ is the set of vertices, and $E \subset [1 \cdots N]^2$, $F \subset [1 \cdots N]^3$ are, respectively, the set of edges and the set of triangles. The shape variation of a specific hand with respect to $\overline{T}$ is captured by the set of PCA bases $\mathcal{S} = \{ S_i \in R^{3N} : i = 1, \ldots, |\mathcal{S}|\}$, where $|\mathcal{S}|$ indicates the element number in $\mathcal{S}$. Without causing confusion, we also use $\overline{V}$ to denote the $3N$-dimensional vector concatenated by the coordinates of its vertices.

The variation of an arbitrary hand shape with respect to $\overline{T}$ can then be computed by blending the bases with coefficients $\beta = \{ \beta_i \in R : i = 1, \ldots, |\mathcal{S}|\}$:

$$B_S(\beta, \mathcal{S}) = \sum_{k=1}^{|\mathcal{S}|} \beta_i S_i$$

To improve the skinning accuracy, MANO makes a compensation to the rest pose in terms of motion parameters

$$B_P(\theta, \mathcal{P}) = \sum_{k=1}^{9K} (R_i(\theta) - R_i(\theta)^*) P_i$$

where K is the number of bones, $\mathcal{P} = \{ P_i, i = 1, \ldots, 9K\}$ is a set of pose bases for blending vertex offsets, $\boldsymbol{\theta} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_{20})$ is the motion parameters of an arbitrary pose and $R_i(\theta)$ indicates the ith entry of the rotational matrices (total K rotational matrices and each matrix has $3 \times 3 = 9$ elements). The full form of MANO can then be written as

$$\mathrm{M}(\beta, \theta) = \mathrm{W}(T_P(\beta, \theta), \mathrm{J}(\beta), \theta, \mathcal{W})$$

where W is the skinning function (linear blending skinning) $\mathrm{J}(\beta) = (\mathbf{j}_0, \mathbf{j}_1, \ldots, \mathbf{j}_{20}) \in R^{21 \times 3}$ is the set of joint position in the rest pose, $\mathcal{W}$ is the weight matrix, and

$$T_P(\beta, \theta) = \overline{V} + B_S(\beta, \mathcal{S}) + B_P(\theta, \mathcal{P}).$$

$\overline{T}$, $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{W}$ are known for our reconstruction task.

3.2 Hand anatomical kinematics

We follow the anatomical kinematics structure in a hand animation approach [9] which is also adopted by robotics arms [46]. Such structure employs the same skeleton as shown in Fig. 1 in which hand joints are classified into hinge joints including joints 2, 3, 6,7, 10, 11, 14, 15, 18 and 19, and saddle joints containing joints 1, 5, 9, 13 and 17.

Specifically, a hinge joint has 1 degree of freedom, namely bones shooting from this kind of joints can only conduct bending motion as shown in Fig. 2 (left). A saddle joint has two degrees of freedoms as depicted in Fig. 2 (right), which are respectively described by two different rotational angles

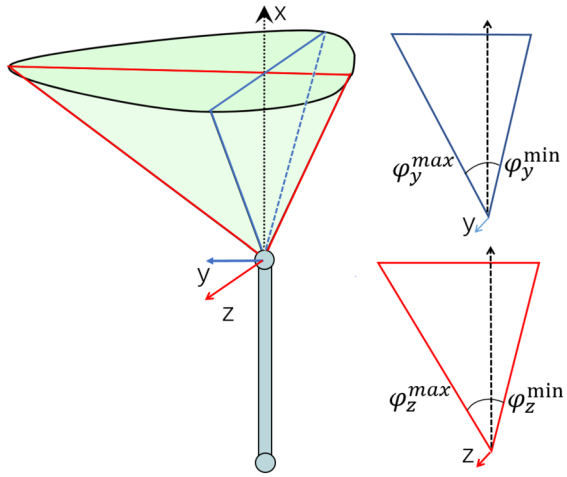

Fig. 3 Rotation range of saddle joint

(flexion/extension, abduction/adduction) as shown in Fig. 3. Biomechanical constraints [50] are exploited to constrain the motion range of these angles.

4 The proposed method

Now, we describe our framework which involves biomechanical constraints of hand motions, the mathematical model for reconstructing hand poses from joint positions, and the solution of the model.

4.1 Hand motion constraints

According to [9,50] (see Sect. 3.2), there are two kinds of joints in a hand, i.e., hinge joints and saddle joints. For the sake of formal description, we create a local coordinate system for each joint (say k for example) in order to describe the orientation of the bone starting from k. As shown in Fig. 3, its origin is placed at k, its x axis points from the parent joint of k to k itself, and the rotational axis of the bending motion around k is viewed as z axis such that the bending motion observes the right-hand rule as shown in Fig. 3.

In the local coordinate system of hinge joint k, the motion of the bone starting from k is actually a rotation around axis z. Let $\phi_{z,k}$ be the rotational angle. We can then express the rotation matrix as [50]

$$e^{\boldsymbol{\omega}_k} = e^{\mathbf{n}_z \phi_{z,k}} \tag{1}$$

with $\mathbf{n}_z = (0, 0, 1)$ and $\phi_{z,k} \in [\phi_{z,k}^{\min}, \phi_{z,k}^{\max}]$, where e^* is called Rodrigues function.

For saddle joint k, the motion of the bone shooting from it is a composition of the two rotations, respectively, around axis z and axis y. Similarly, denoting the two angles by $\phi_{z,k}$

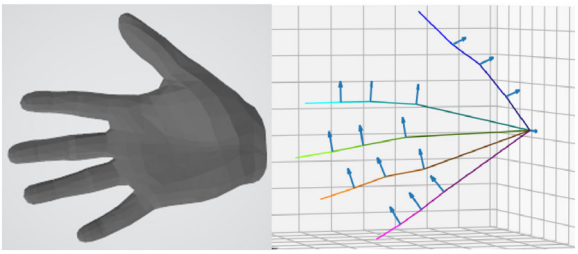

Fig. 4 Right-hand mesh template (left) and z axes of its joints (blue arrows in the right)

and $\phi_{y,k}$ separately, we then have [50]

$$e^{\boldsymbol{\omega}_k} = e^{(\mathbf{n}_z \phi_{z,k})} e^{(\boldsymbol{n}_y \phi_{y,k})}, \tag{2}$$

where $\mathbf{n}_y = (0, 1, 0)$. The above rotational angles satisfy the following ellipse constraint

$$\left(\frac{\phi_{y,k}}{\bar{\phi}^*_{y,k}}\right)^2 + \left(\frac{\phi_{z,k}}{\bar{\phi}^*_{z,k}}\right)^2 \leq 1, \tag{3}$$

where $* =' \min'$ or $'\max'$ as shown in Table 1 [2,17] depending on which quadrant the child joint of joint k locates in.

The above rotation transformations will be formulated into MANO [38]. Figure 4 depicts z axis of the local system of all joints. We determine the z axes using skin surface details and registration data. Notice that leaf joints (finger tips) have no additional information since there is no bone shooting from them. On the other hand, as the root of the kinematic tree, the wrist joint is free of constraints. In practical terms, the translation of the root joint can be viewed as the inverse translation of the camera. So, we can fix the root node and only estimate the global transformation.

4.2 Hand pose reconstruction from 3D joint positions

In our setting, we neglect the hand shape and only reconstruct the pose using MANO. Denote the position vector of all joints by $\mathbf{J}^{3D} = (\mathbf{j}_0^{3D}, \mathbf{j}_1^{3D}, \ldots, \mathbf{j}_{20}^{e}) \in R^{21\times 3}$. Particularly, denote their rest pose counterparts by $\mathbf{J}^* = (\mathbf{j}_0^*, \mathbf{j}_1^*, \ldots, \mathbf{j}_{20}^*) \in R^{21\times 3}$ (see Fig. 4).

Pose change is captured by rotating around joints. In our setting, only 15 joints are rotatable. The wrist joint is fixed and its orientation is described using the camera parameters instead. In addition, tips of five fingers have not parameters. Namely, $\boldsymbol{\omega}_0$ remains unchanged, while $\boldsymbol{\omega}_4$, $\boldsymbol{\omega}_8$, $\boldsymbol{\omega}_{12}$, $\boldsymbol{\omega}_{16}$ and $\boldsymbol{\omega}_{20}$) are known and have no impact on the pose. According to the forward kinematics, this yields the following global

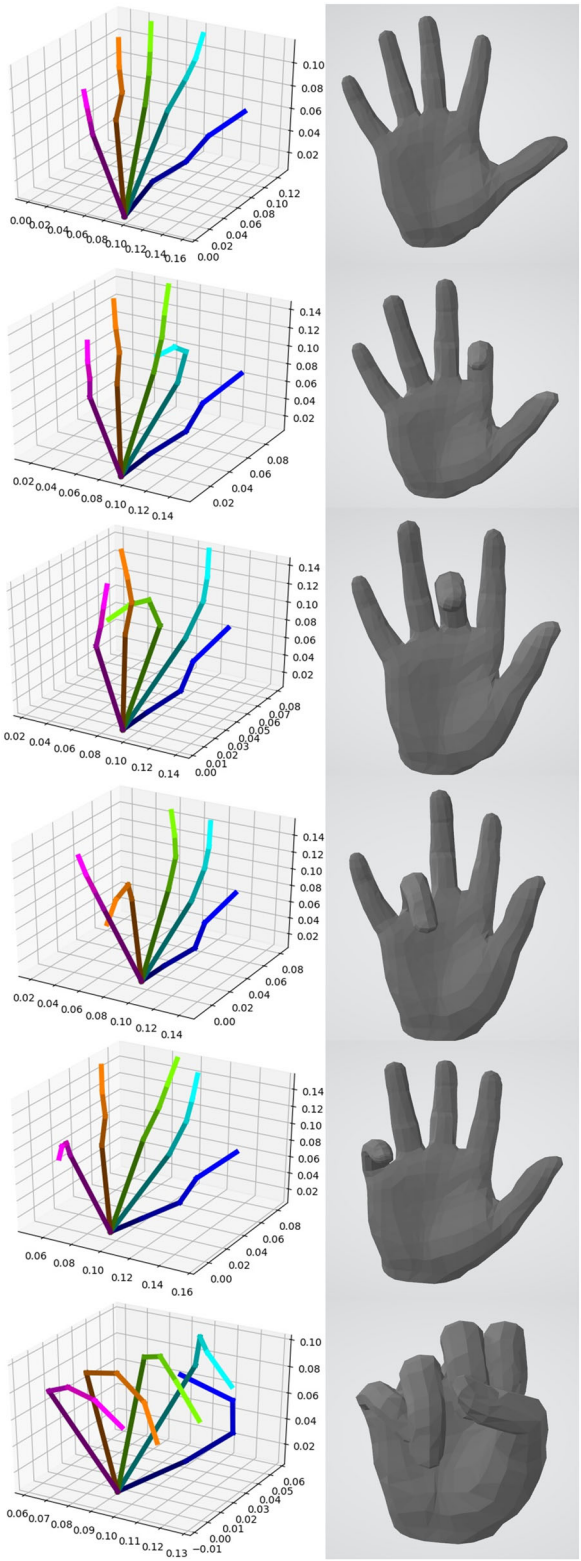

Fig. 5 Visual results of the proposed algorithm. Left is the input joint positions; right shows the corresponding results

Table 1 Rotation range of hand joints (unit:)

Joint	DOF	$\phi_{z,k}^{\min}$	$\phi_{z,k}^{\max}$	$\phi_{y,k}^{\min}$	$\phi_{y,k}^{\max}$
6, 10, 14, 18	1	0	100	N/A	N/A
3, 7, 11, 15, 19	1	0	90	N/A	N/A
2	1	0	80	N/A	N/A
1	2	−20	20	−30	30
5, 9, 13, 17	2	−10	85	−10	10

transformation matrix for joint k:

$$\mathbf{G}_k(\boldsymbol{\theta}, \mathbf{j}^*) = \prod_{i \in A(k)} \begin{bmatrix} e^{(\boldsymbol{\omega}_i)} & \mathbf{j}_i^* - \mathbf{j}_{p(i)}^* \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where $e^{(\boldsymbol{\omega}_i)}$ is the rotation matrix of joint i, $A(k)$ represents the node path from the second level ancestor, which is adjacent to node 0, to the parent node of joint k; $p(i)$ denotes the parent node of i; $\mathbf{G}_k(\boldsymbol{\theta}, \mathbf{J})$ describes the transformation of joint k related to the world system. Equation (4) can further be simplified as

$$\mathbf{G}_k(\boldsymbol{\theta}, \mathbf{J}^*) = \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ 0 & 1 \end{bmatrix} \tag{5}$$

where $\mathbf{R}_k$ is a 3×3 rotation matrix and $\mathbf{t}_k$ is the translation of joint k. Let $[\mathbf{R}^g|\mathbf{t}^g]$ be the global rigid transformation. If a set of joint positions in J^e are given, motion estimation can then be formulated as minimizing the mean square distance between the prediction joint positions and the given ones under hand biomechanical constraints:

$$\underset{\boldsymbol{\theta}, R^g, t^g}{\arg\min} \sum_k (\mathbf{R}^g \mathbf{t}_k + \mathbf{t}^g - \mathbf{j}_k^e)^2 \tag{6}$$

s.t.

(i) if DOF(k)=1,$\boldsymbol{\omega}_k = \mathbf{n}_z \phi_{z,k}$, $\phi_{z,k} \in [\phi_{z,k}^{min}, \phi_{z,k}^{max}]$;

(ii) if DOF(k)=2,$e^{(\boldsymbol{\omega}_k)} = e^{(\mathbf{n}_z \phi_{z,k})} e^{(\mathbf{n}_y \phi_{y,k})}$;

$$\text{and} \left(\frac{\phi_{y,k}}{\bar{\phi}_{y,k}^*}\right)^2 + \left(\frac{\phi_{z,k}}{\bar{\phi}_{z,k}^*}\right)^2 \leq 1.$$

The arguments to be optimized are pose parameters $\boldsymbol{\theta}$, global transformation $\mathbf{R}^g$ and $\mathbf{t}^g$. It is difficult to consider all the constraints simultaneously. Hence, inspired by the idea of ICP and based on the hierarchical structure of the hand kinematic tree, we devise a block coordinate descent scheme to address Eq. 6.

4.3 Numerical solution

We divide the variables into different blocks according to their joint number in order to solve Eq. 6. For each block, we minimize a subproblem by fixing all other blocks. The order of optimizing different blocks follows the join number order in a hierarchical manner: first $\mathbf{R}^g$ and $\mathbf{t}^g$, then from $\boldsymbol{\omega}_1$ to $\boldsymbol{\omega}_{19}$. In this subsection, we will separately discuss the subproblems according to their types including camera parameters, saddle joints and hinge joints.

4.3.1 Initialization

In the beginning, we first need to initialize the current poses. Specifically, we evaluate the mean pose of the MANO dataset as the initial pose and use it to estimate the camera external parameters. $\mathbf{G}_k$ in Eq. 5 can then be obtained from current pose parameters $\boldsymbol{\theta}$.

4.3.2 Update of global rigid transformation

Global transformation is in place of the root pose. Hence, all hand joint positions are transformed correspondingly. While only considering $\mathbf{R}^g$ and $\mathbf{t}^g$, the optimization has the form:

$$\begin{aligned} &\underset{\mathbf{R}^g, \mathbf{t}^g}{\arg\min} \sum_k (\mathbf{R}^g \mathbf{t}_k + \mathbf{t}^g - \mathbf{j}_k^e)^2, \\ &\text{s.t.} \quad (\mathbf{R}^g)^T \mathbf{R}^g = \mathbf{I} \end{aligned} \tag{7}$$

A method to address this problem uses the difference between barycenters of source points and target points in order to find the translation $\mathbf{t}^g$, while $\mathbf{R}^g$ can be solved with Kabsch algorithm [19].

4.3.3 Update of saddle joint parameters

Saddle joints locate in the second level in the kinematic chain, which are directly adjacent to the root. Updating the transformation of this kind of joints only influences the position of their descendent joints. Therefore, the optimization for saddle joint k reduces to

$$\begin{aligned} &\underset{\boldsymbol{\omega}_k}{\arg\min} \sum_{i \in D(k)} (\mathbf{R}^g (\mathbf{R}_k e^{\boldsymbol{\omega}_k} (\mathbf{j}_i^* - \mathbf{j}_k^*) + \mathbf{t}_k) + \mathbf{t}^g - \mathbf{j}_i^e)^2, \\ &\text{s.t.} \\ &\quad e^{\boldsymbol{\omega}_k} = e^{\mathbf{n}_k^* \phi_{z,k}} exp^{\boldsymbol{\omega}_{y,k}^* \phi_{y,k}}; \end{aligned} \tag{8}$$

Table 2 MPJPE and MPVPE on the MANO dataset

Method	MPJPE (mm)	Std of MPJPE (mm)	MPVPE (mm)	Std of MPVPE (mm)	Time (ms)
GDC [38]	7.10	3.01	7.32	3.05	1477
MLP [55]	9.19	3.89	12.0	5.75	**15**
Ours	**5.35**	**1.91**	**5.84**	**1.96**	34

Table 3 PA MPJPE and PA MPVPE on the MANO dataset

Method	PA MPJPE (mm)	Std of PA MPJPE (mm)	PA MPVPE (mm)	Std of PA MPVPE (mm)
GDC [38]	6.18	2.33	6.53	2.50
MLP [55]	7.55	3.13	8.02	3.11
Ours	**4.77**	**1.80**	**5.24**	**1.83**

Table 4 Hand reconstruction from a single image: comparison

Metrics	AUC of mesh PCK	Err of mesh (mm)	AUC of joint PCK	Err of joints (cm)	Time (s)
MANO CNN [57]	0.783	1.09	0.784	1.09	1.57
MANO fit [57]	0.729	1.37	0.730	1.35	5.83
Obman [14]	0.738	1.32	0.739	1.32	1.61
Hand only [4]	0.736	1.33	0.737	1.33	2.59
Minimal hand [55]	0.742	1.31	0.746	1.30	**0.012**
Ours	**0.792**	**1.04**	**0.799**	**1.01**	0.046

and

$$\left(\frac{\phi_{y,k}}{\bar{\phi}^*_{y,k}}\right)^2 + \left(\frac{\phi_{z,k}}{\bar{\phi}^*_{z,k}}\right)^2 \leq 1.$$

where $D(k)$ is the descendant set of joint k. Equation 8 solves the rotation of joint k by minimizing the error of the prediction to the ground truth of the descendents of joint k. Noting $(R^g)^{-1} = (R^g)^T$ and $R_k^{-1} = R_k^T$, we have the following equivalent form of Eq. 8

$$\arg\min_{\boldsymbol{\omega}_k} \sum_{i \in D(k)} (e^{\boldsymbol{\omega}_k}(\mathbf{j}_i^* - \mathbf{j}_k^*) + \mathbf{R}_k^T(\mathbf{t}_k + (\mathbf{R}^g)^T(\mathbf{t}^g - \mathbf{j}_i^e)))^2 \tag{9}$$

Equation 9 is actually the orthogonal Procrustes problem [13] if neglecting the constraints of Eq. 8. Henceforth, we tackle it via two steps. First, Kabsch algorithm is employed to address the unconstrained problem to yield $\mathbf{R}_{\text{temp}}$. Second, Euler angles around z and y axes are extracted from $\mathbf{R}_{\text{temp}}$ [8]. Viewing the pair of Euler angles as a point, we then find its nearest point within the ellipse by using Newton–Raphson algorithm. Let $\mathbf{R}_k$ be the rotation matrix constructed from the new Euler angles, $\phi_{y,k}$ and $\phi_{z,k}$, which are around axes y and z, respectively.

It should be noted that an additional internal rotation for thumb joints should be considered. We formulate it as $\mathbf{R}_{in} = e^{(\mathbf{j}_{c(k)} - \mathbf{j}_k)\alpha\phi_{y,k}}$, where α is a predefined heuristic parameter. The final transformation for joint k of the thumb is then computed as $R_{in}R_k$. The internal rotation of other fingers can be ignored because the rotation around the z axis of other fingers is generally perpendicular to the hand palm.

4.3.4 Update of hinge joint parameters

Update of hinge joints is similar to update of saddle joints but their constraints are simpler. Namely, only one rotational angle is constrained for each joint. Hence, the approach for saddle joints is also applicable here. Specifically, we first solve the unconstrained problem to obtain a rotation transformation, then extract Euler angle of the fixed axis, and finally apply the constraints to the angle for computing the final rotation.

4.3.5 Stop criteria and collision avoidance

In an iteration of updating all the parameters, our algorithm successively optimizes the parameters of each single joint as shown in Algorithm 1. It stops when there is no improvement or the iteration number exceeds the threshold. To avoid the possible finger intersection, we introduce an additional step

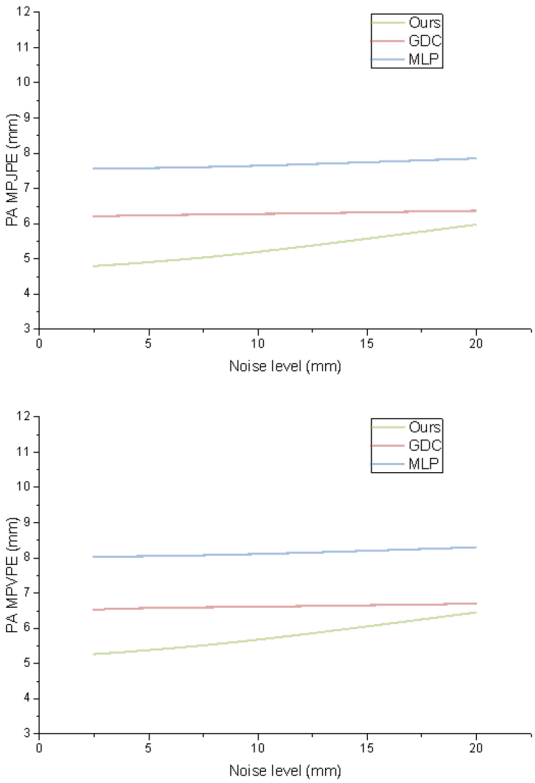

Fig. 6 PA MPJPE and PA MPVPE curves of the three algorithms (ours, GDC [38] and MLP [55]) for inputs with different levels of Gaussian noise. The errors are evaluated using the reconstructed mesh and the ground truth

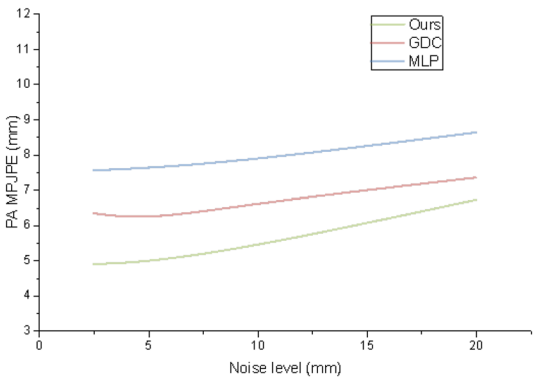

Fig. 7 PA MPJPE curves of the three algorithms (ours, GDC [38] and MLP [55]) for inputs with different levels of Gaussian noise. The errors are evaluated using the reconstructed joint positions and the groud truth

to detect collision. It treats each bone as a capsule. Therefore, once the distance between two line segments (of the bones) is less than a specified threshold determined by the thickness of the two bones, they are considered intersection. In this case, we move one of the bones along the opposite direction to ensure the distance threshold.

Algorithm 1 Evaluate joint motion and external parameters of the camera

Require:
The target positions of all joints, $\mathbf{J}^e$;
joint rotation constraints, stop criteria
Ensure:
The pose parameters $\boldsymbol{\theta} = (\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_{20})$;
and global transformation pose $\mathbf{R}^g$ and $\mathbf{t}^g$
1: Initialize $\boldsymbol{\theta}$ to a pre-compute mean pose;
2: **Repeat**
3: update $\mathbf{R}^g$ and $\mathbf{t}^g$;
4: update all $\boldsymbol{\omega}_i$ in $\boldsymbol{\theta}$ one by one
5: according to
its joint type;
6: **Until** The convergent criteria are satisfied.
7: END

4.4 Reconstruction from images

Our approach can be applied to reconstruct 3D poses from a single image. The process consists of two steps. Firstly, we employ existing methods to detect 2D joints on the image and estimate the 3D joint positions from 2D ones. In our experiments, we use PoseNet [7] to do this. After that, we can use Algorithm 1 to estimate motion data which is finally used to skin the MANO model to generate hand meshes.

5 Experiments

The proposed algorithm is implemented with Python on a PC with Intel(R) Core (TM) i7-4470 CPU @ 3.4GHz. This section presents a variety of experiments to show the performance of the proposed algorithm.

5.1 Visual results

We first take use of some examples with typical hand poses to show the effectiveness of our method which solves the motion of each finger independently and combine them together to express complicated poses. In each example, the 3D joint positions are given. Figure 5 illustrates that our approach is able to rotate and redirect the fingers to exactly register the joint positions.

5.2 Accuracy on the MANO registration dataset

To quantitatively evaluate our approach, we conduct an experiment on the MANO dataset [38] which is built by using the MANO hand parametric model to register the MoCap scans. The dataset includes 1554 poses of real human hands from 31 subjects. Each sample consists of a set of 3D hand joint positions as well as the corresponding pose parameters. We recover the pose parameters from 3D joint positions and then compared them with the ground truth.

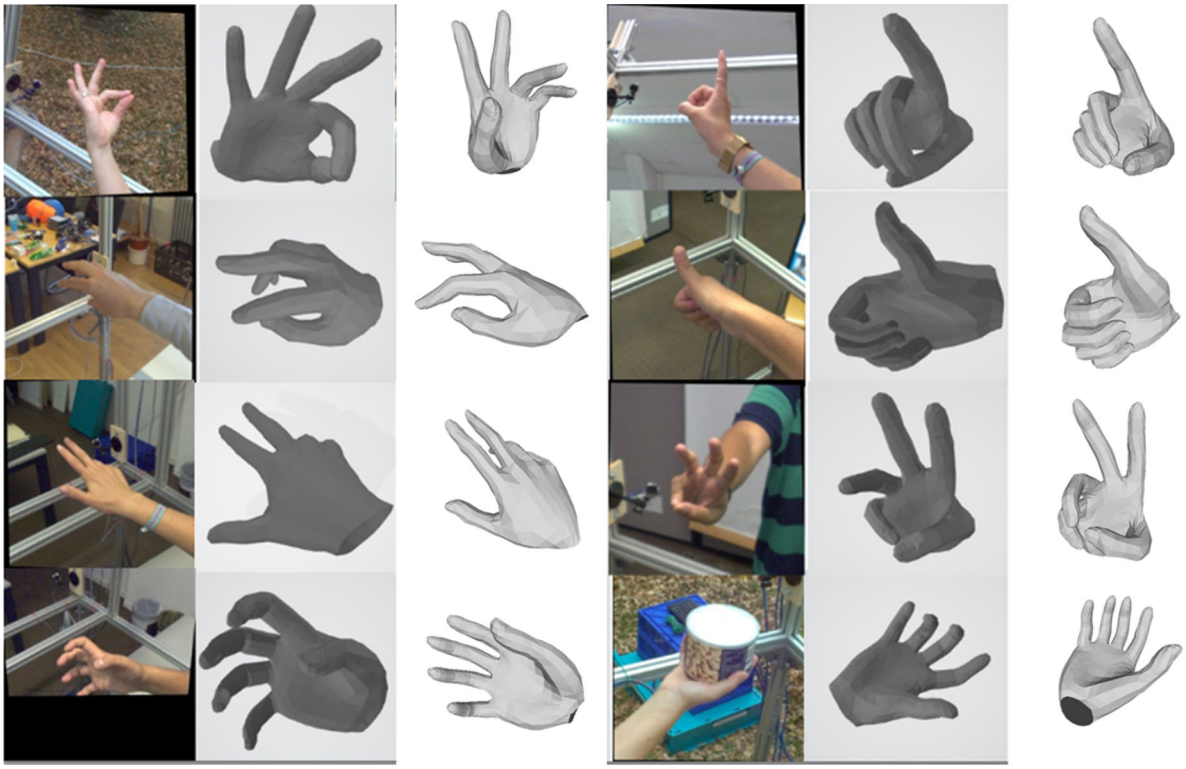

Fig. 8 Single image reconstruction: each row shows two examples and total 8 examples are depicted. Each example includes three images which are, respectively, the input image, the reconstructed hand model by our method, and the result by FrankMoCap [39]

Two error metrics are used: mean per-joint position error (MPJPE) and mean per-vertex position error (MPVPE). In addition, we also compute PA MPJPE and PA MPVPE, the rigid alignment values of MPJPE and MPVPE respectively. The gradient descent (GDC) method [38] and the MLP method [55] are selected to compare with our approach. Table 2 summarizes the mean errors of all samples in the dataset, their standard deviation (std) and timings. Results for PA MPJPE and PA MPVPE are listed in Table 3. Both tables illustrate that our method outperforms the other two in both accuracy and stability (with smaller std). In addition, GDC is the slowest one, while our method is comparable to the MLP method (Table 3).

5.2.1 Accuracy on the dataset with simulated noise

Considering that real data acquired from motion capture devices are usually contaminated, we evaluate our method using inputs with different levels of simulated noise. As the average length of hand bones is 33.4 mm in the template pose, we, respectively, add Gaussian noise with standard deviation of 2, 5, 10 and 20 (mm) to every joint position of the MANO registration dataset.

Figure 6 shows the curves of PA MPJPE and PA MPVPE between the reconstructed hand mesh and the ground truth, while Fig. 7 depicts the PA MPJPE curves of joint positions. Both demonstrates that our approach outperforms the other two state-of-the-art approaches [38,55] in reconstruction accuracy owing to introducing mechanical constraints. Nevertheless, our method is more sensitive to the noise level change. This is because some noisy inputs may happen to be a valid pose, while our algorithm still fits the pose parameters to such a deformed pose. An elaborate comparison with the state-of-the-art methods [4,14,55,57] is conducted as illustrated in Table 4. Our approach achieves the highest accuracy.

5.3 Pose reconstruction from a single image

To reconstruct hand pose from a single RGB image, we first apply PoseNet [7] to estimate 3D joint positions from the image and then compute motion parameters with our algorithm. The experiment is conducted on the FreiHAND dataset [57] which consists of 130K training images and 4K test images with MANO pose parameters. Figure 8 depicts some visual results (MANO models). We also illustrate the results by FrankMoCap [39] as comparison. Visually, the hand gestures by our method are better than those by FrankMoCap [39]. In the first example, the gesture by FrankMoCap [39] is even wrong (see column 3 of row 1 in the figure).

We also compare our algorithm with the state-of-the-art approaches described in [4,14,55,57]. PoseNet [7] is trained using the training images to estimate 3D joint positions as

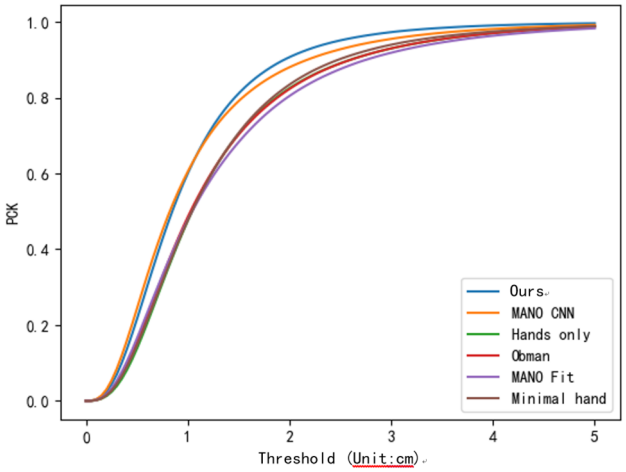

Fig. 9 3D Hand mesh reconstruction PCK curves for our method and approaches in [4,14,55,57]

input of our method. All involved reconstruction algorithms are directly used without additional training. Figure 9 depicts the PCK curves by these approaches and ours. It can be observed that our approach performs best.

Table 4 summarizes the reconstruction errors by the involved approaches measured by a variety of error metrics, which also shows that our method outperforms the state-of-the-art methods in almost all indices. Though minimal hand [55] is faster than our approach, all accuracy indices are worst.

To show the stability of our approach, we apply it to reconstruct a sequence frame by frame. The data come from sequence 171204_pose6 of the database in [18]. Totally, 100 frames are reconstructed (see the attached video), and the indices of these 100 frames are from 20,701 to 20,800. Here, we depict 5 of the 100 frames with indices of 20,720, 20,740, 20,760, 20,780 and 20,800 in Fig. 10. PA MPJPE by our method is 5.3302 with std = 0.4320, while PA-MPJPE by FrankMoCap is 5.3442 with std = 0.5151. Our approach is more stable than FrankMoCap [39].

5.4 Limitations

We accomplish the task of reconstructing hand poses from hand joint coordinates by proposing an iterative algorithm based on mechanical constraints. A variety of experiments demonstrate that it exhibits excellent performance compared to the state-of-the-art approaches. However, it does not distinguish the shape of different hands. In addition, it depends on PoseNet when recovering motion parameters from images. Therefore, once PoseNet fails to predict 3D hand joint positions correctly, our algorithm cannot rectify the case.

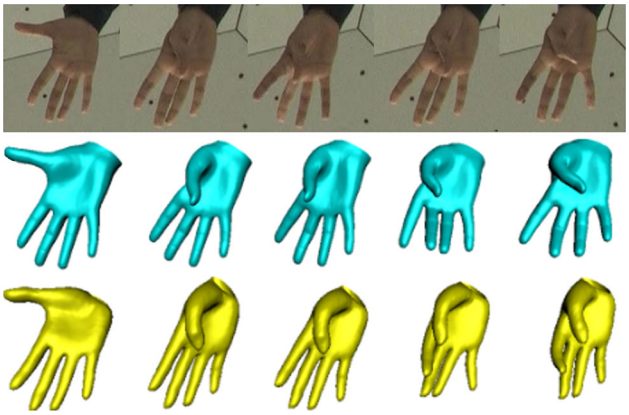

Fig. 10 Sequence reconstruction: the top row shows the corresponding images, the middle row shows results reconstructed by our approach, and the bottom row shows results by FrankMoCap [39]

6 Conclusions

We propose a coordinate descent algorithm for reconstructing hand motion parameters from joint positions, in which the joint rotation of each finger bone is solved successively by fixing other pose motion parameters. The natural structure of hands is used to constrain joint motions in order to reduce the search space. These two contributions make our algorithm exhibits advantages in accuracy, robustness and running time. As future work, it is interesting to extend the proposed framework to estimate hand motion sequences by utilizing the inter-frame coherence to sustain the stability of the sequence.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00371-021-02250-y.

Funding This work was partially supported by NSFC (61972160, 62072191), Guangdong Basic and Applied Basic Research Foundation (2021A1515012301, 2019A1515010833, 2019A1515010860), and the Fundamental Research Funds for the Central Universities (2020ZYGXZR089, D2190670).

Declaration

Conflicts of interest The authors declare that they have no conflicts of interest.

References

1. Ahmad, A., Migniot, C., Dipanda, A.: Hand pose estimation and tracking in real and virtual interaction: A review. Image Vis. Comput. **89**, 35–49 (2019)
2. Aristidou, A.: Hand tracking with physiological constraints. Vis. Comput. **34**(2), 213–228 (2018)
3. Athitsos, V., Sclaroff, S.: Estimating 3d hand pose from a cluttered image. In: 2003 IEEE Computer Society Conference on Computer

Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA, pp. 432–442 (2003)
4. Boukhayma, A., de Bem, R., Torr, P.H.S.: 3d hand shape and pose from images in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 10,843–10,852 (2019)
5. Bray, M., Koller-meier, E., Müller, P., Gool, L.V., Schraudolph, N.N.: 3d hand tracking by rapid stochastic gradient descent using a skinning model. In: 1st European Conference on Visual Media Production (CVMP, pp. 59–68 (2004)
6. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond euclidean data. IEEE Signal Process. Mag. **34**(4), 18–42 (2017)
7. Choi, H., Moon, G., Lee, K.M.: Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In: A. Vedaldi, H. Bischof, T. Brox, J. Frahm (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII, *Lecture Notes in Computer Science*, vol. 12352, pp. 769–787 (2020)
8. Diebel, J.: Representing attitude: Euler angles, unit quaternions, and rotation vectors. (2006). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.5134
9. ElKoura, G., Singh, K.: Handrix: animating the human hand. In: R. Parent, K. Singh, D.E. Breen, M.C. Lin (eds.) Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, July 26-27, 2003, pp. 110–119 (2003)
10. Feng, Z., Zhang, M., Pan, Z., Yang, B., Xu, T., Tang, H., Li, Y.: 3d-freehand-pose initialization based on operator's cognitive behavioral models. Vis. Comput. **26**(6–8), 607–617 (2010)
11. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3d hand shape and pose estimation from a single RGB image. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 10,833–10,842 (2019)
12. Glauser, O., Wu, S., Panozzo, D., Hilliges, O., Sorkine-Hornung, O.: Interactive hand pose estimation using a stretch-sensing soft glove. ACM Trans. Graph. **38**(4), 41:1–41:15 (2019)
13. Gower, J.C., Dijksterhuis, G.B.: Procrustes Problems. Oxford University Press, Oxford (2004)
14. Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C.: Learning joint reconstruction of hands and manipulated objects. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 11,807–11,816. Computer Vision Foundation / IEEE (2019)
15. Heap, T., Hogg, D.C.: Towards 3d hand tracking using a deformable model. In: 2nd International Conference on Automatic Face and Gesture Recognition (FG '96), October 14-16, 1996, Killington, Vermont, USA, pp. 140–145 (1996)
16. Imai, A., Shimada, N., Shirai, Y.: 3-d hand posture recognition by training contour variation. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea, pp. 895–900 (2004)
17. Imai, A., Shimada, N., Shirai, Y.: Hand posture estimation in complex backgrounds by considering mis-match of model. In: Y. Yagi, S.B. Kang, I. Kweon, H. Zha (eds.) Computer Vision - ACCV 2007, 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18-22, 2007, Proceedings, Part I, *Lecture Notes in Computer Science*, vol. 4843, pp. 596–607 (2007)
18. Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., Banerjee, S., Godisart, T.S., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social interaction capture. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
19. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A **32**, 922–922 (1976)
20. Kitagawa, M., Windsor, B.: MoCap for Artists: Workflow and Techniques for Motion Capture (2012). https://doi.org/10.4324/9780080877945
21. Kolluri, R.K., Shewchuk, J.R., O'Brien, J.F.: Spectral surface reconstruction from noisy point clouds. In: J. Boissonnat, P. Alliez (eds.) Second Eurographics Symposium on Geometry Processing, Nice, France, July 8-10, 2004, ACM International Conference Proceeding Series, vol. 71, pp. 11–21 (2004)
22. Kulon, D., Güler, R.A., Kokkinos, I., Bronstein, M.M., Zafeiriou, S.: Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pp. 4989–4999 (2020)
23. de La Gorce, M., Fleet, D.J., Paragios, N.: Model-based 3d hand pose estimation from monocular video. IEEE Trans. Pattern Anal. Mach. Intell. **33**(9), 1793–1805 (2011)
24. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM Trans. Graph. **34**(6), 248:1–248:16 (2015)
25. Magnenat-thalmann, N., Laperrire, R., Thalmann, D., Montréal, U.D.: Joint-dependent local deformations for hand animation and object grasping. In: In Proceedings on Graphics interface '88, pp. 26–33 (1988)
26. Marquardt, A., Maiero, J., Kruijff, E., Trepkowski, C., Schwandt, A., Hinkenjann, A., Schöning, J., Stuerzlinger, W.: Tactile hand motion and pose guidance for 3d interaction. In: S.N. Spencer, S. Morishima, Y. Itoh, T. Shiratori, Y. Yue, R. Lindeman (eds.) Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology, VRST 2018, Tokyo, Japan, November 28 - December 01, 2018, pp. 3:1–3:10. ACM (2018)
27. Melax, S., Keselman, L., Orsten, S.: Dynamics based 3d skeletal hand tracking. In: F.F. Samavati, K. Hawkey (eds.) Graphics Interface 2013, GI '13, Regina, SK, Canada, May 29-31, 2013, Proceedings, pp. 63–70 (2013)
28. Miyamoto, S., Matsuo, T., Shimada, N., Shirai, Y.: Real-time and precise 3-d hand posture estimation based on classification tree trained with variations of appearances. In: Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012, pp. 453–456 (2012)
29. Mueller, F., Davis, M., Bernard, F., Sotnychenko, O., Verschoor, M., Otaduy, M.A., Casas, D., Theobalt, C.: Real-time pose and shape reconstruction of two interacting hands with a single depth camera. ACM Trans. Graph. **38**(4), 82:1–82:12 (2019)
30. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: D.N. Metaxas, L. Quan, A. Sanfeliu, L.V. Gool (eds.) IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011, pp. 2088–2095 (2011)
31. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 10,975–10,985 (2019)
32. Pavllo, D., Porssut, T., Herbelin, B., Boulic, R.: Real-time marker-based finger tracking with neural networks. In: K. Kiyokawa, F. Steinicke, B.H. Thomas, G. Welch (eds.) 2018 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018, Tuebingen/Reutlingen, Germany, 18-22 March 2018, pp. 651–652 (2018)
33. Peng, H., Xian, C., Zhang, Y.: 3d hand mesh reconstruction from a monocular RGB image. Vis. Comput. **36**(10), 2227–2239 (2020)
34. Polygerinos, P., Galloway, K.C., Savage, E., Herman, M., O'Donnell, K., Walsh, C.J.: Soft robotic glove for hand rehabilitation and task specific training. In: IEEE International Conference

on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015, pp. 2913–2919 (2015)
35. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pp. 1106–1113 (2014)
36. Qian, N., Wang, J., Mueller, F., Bernard, F., Golyanik, V., Theobalt, C.: HTML: A parametric hand texture model for 3d hand reconstruction and personalization. In: A. Vedaldi, H. Bischof, T. Brox, J. Frahm (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI, *Lecture Notes in Computer Science*, vol. 12356, pp. 54–71. Springer (2020)
37. Romero, J., Kjellström, H., Kragic, D.: Monocular real-time 3d articulated hand pose estimation. In: 9th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2009, Paris, France, December 7-10, 2009, pp. 87–92. IEEE (2009)
38. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: modeling and capturing hands and bodies together. ACM Trans. Graph. **36**(6), 245:1–245:17 (2017)
39. Rong, Y., Shiratori, T., Joo, H.: Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. arXiv preprint arXiv:2008.08324 (2020)
40. Simon, T., Joo, H., Matthews, I.A., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pp. 4645–4653 (2017)
41. Spurr, A., Iqbal, U., Molchanov, P., Hilliges, O., Kautz, J.: Weakly supervised 3d hand pose estimation via biomechanical constraints. In: A. Vedaldi, H. Bischof, T. Brox, J. Frahm (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII, *Lecture Notes in Computer Science*, vol. 12362, pp. 211–228 (2020)
42. Spurr, A., Song, J., Park, S., Hilliges, O.: Cross-modal deep variational hand pose estimation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 89–98 (2018)
43. Stenger, B., Mendonça, P.R.S., Cipolla, R.: Model-based 3d tracking of an articulated hand. In: 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA, pp. 310–315 (2001)
44. Tang, D., Chang, H.J., Tejani, A., Kim, T.: Latent regression forest: Structured estimation of 3d articulated hand posture. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pp. 3786–3793 (2014)
45. Tompson, J., Stein, M., LeCun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. **33**(5), 169:1–169:10 (2014)
46. Tuffield, P., Elias, H.: The shadow robot mimics human actions. Ind. Robot **30**(1), 56–60 (2003)
47. Wang, R.Y., Popovic, J.: Real-time hand-tracking with a color glove. ACM Trans. Graph. **28**(3), 63 (2009)
48. Wheatland, N., Jörg, S., Zordan, V.B.: Automatic hand-over animation using principle component analysis. In: R. McDonnell, N.R. Sturtevant, V.B. Zordan (eds.) Motion in Games, MIG '13, Dublin, Ireland, November 6-8, 2013, pp. 197–202 (2013)
49. Wheatland, N., Wang, Y., Song, H., Neff, M., Zordan, V.B., Jörg, S.: State of the art in hand and finger modeling and animation. Comput. Graph. Forum **34**(2), 735–760 (2015)
50. Wilding, J., Corcos, D.M.: Basic biomechanics of the musculoskeletal system, ed 3. (reviews). (book review). Physical Therapy (December) (2001)
51. Xiang, D., Joo, H., Sheikh, Y.: Monocular total capture: Posing face, body, and hands in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 10,965–10,974 (2019)
52. Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., Yang, Q.: A hand pose tracking benchmark from stereo matching. In: 2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017, pp. 982–986 (2017)
53. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular RGB image. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 2354–2364 (2019)
54. Zhao, W., Chai, J., Xu, Y.: Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. In: J. Lee, P.G. Kry (eds.) Proceedings of the 2012 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2012, Lausanne, Switzerland, 2012, pp. 33–42 (2012)
55. Zhou, Y., Habermann, M., Xu, W., Habibie, I., Theobalt, C., Xu, F.: Monocular real-time hand shape and motion capture using multi-modal data. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pp. 5345–5354 (2020)
56. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single RGB images. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pp. 4913–4921 (2017)
57. Zimmermann, C., Ceylan, D., Yang, J., Russell, B.C., Argus, M.J., Brox, T.: Freihand: A dataset for markerless capture of hand pose and shape from single RGB images. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 813–822 (2019)

Guiqing Li is a professor in the School of Computer Science and Engineering, South China University of Technology. He has published 100+ papers and more than 50 papers published in international mainstream journals on computer graphics such as ACM TOG, IEEE TVCG, IEEE TIP, IEEE TVCST, CGF, CAGD, C&G, and TVC. He often serves as the PC member of international conferences, including Geometric Modeling and Processing, Shape Modeling International, and CAD/Graphics. He is now on the editorial board of the Journal of Computer-aided Design and Computer Graphics. His research interests are dynamic geometry processing, image and video editing, and digital geometry processing.

Zihui Wu is a software engineer in Alibaba incorporated company. He received his graduate degree in computer science engineering from South China University of Technology in 2020. His research interests are image and video processing, computer graphics, geometry modeling and processing and 3D reconstruction.

Yuxin Liu is a graduate student in the School of Computer Science and Engineering, South China University of Technology. He has published several papers in computer graphics journals. His research interests are computer graphics, geometry modeling and 3D reconstruction.

Huiqian Zhang is a graduate student in the School of Computer Science and Engineering, South China University of Technology. His research interests are image processing, computer graphics and 3D human reconstruction.

Yongwei Nie is an associate professor in the School of Computer Science and Engineering, South China University of Technology. He has published more than 30 papers, mainly in mainstream journals (IEEE TVCG, IEEE TIP, IEEE TCSVT, IEEE TMM etc.) and conferences (AAAI, PG, CAD/Graphics, CASA, Chinagraph, etc.) in Computer Graphics and Computer Vision. His current research interests are computer graphics, computer vision (artificial intelligence), digital image/video processing.

Aihua Mao is a professor in the School of Computer Science and Engineering, South China University of Technology. He has published more than 50 papers, mainly in mainstream journals (IEEE TVCG, ACM TOMM, Computer-Aided Design, C&G, TVCJ, etc.) and conferences (Siggraph Asia, CAD/Graphics, Chinagraph, etc.) in Computer Graphics and Computer Vision. His current research interests are computer graphics, 3D vision, intelligent design.