Neural Parametric Human Hand Modeling with Point Cloud Representation

Jian Yang
jianyang0227@gmail.com
[1] Institute of Automation, Chinese Academy of Sciences, [2]School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China

Weize Quan
qweizework@gmail.com
[1] Institute of Automation, Chinese Academy of Sciences, [2]School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China

Zhen Shen
zhen.shen@ia.ac.cn
[1] Institute of Automation, Chinese Academy of Sciences, [2]School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China

Dong-Ming Yan
yandongming@gmail.com
[1] Institute of Automation, Chinese Academy of Sciences, [2]School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China

Huai-Yu Wu*
huaiyu.wu@ia.ac.cn
[1] Institute of Automation, Chinese Academy of Sciences, [2]School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China

ABSTRACT

Recent multi-layer perceptron(MLP)-based implicit representations have achieved remarkable successes in hand modeling. Compared with previous explicit mesh-based representation methods, implicit methods are more compact shape representations. However, it is expensive to obtain explicit geometry surfaces from implicit functions with Marching Cubes, which limits the real-time performance in surface reconstruction applications with an implicit hand representation. To explore a more effective and efficient hand representation, we present a skeleton-driven method to represent a human hand with a point cloud. To achieve this goal, we propose a Tri-Axis Modeling method to modeling the motion pattern of the *xyz* coordinate of a patch of point cloud, and an Order Encoding strategy to construct a parameter-sharing and geometry-disentangled network. These two effective strategy make our method run in real-time and has super-high fidelity close to implicit methods. Qualitative and quantitative experiments on public datasets demonstrate the efficiency, effectiveness, and robustness of our method against state-of-the-art approaches. Our code would be soon released once our paper is accepted.

CCS CONCEPTS

• **Computing methodologies → Mixed / augmented reality**.

KEYWORDS

Hand Modeling, Point Cloud Generation, Disentangled Representation

ACM Reference Format:
Jian Yang, Weize Quan, Zhen Shen, Dong-Ming Yan, and Huai-Yu Wu. 2024. Neural Parametric Human Hand Modeling with Point Cloud Representation. In *Proceedings of the 2024 International Conference on Multimedia Retrieval (ICMR '24), June 10–14, 2024, Phuket, Thailand.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652583.3658012

*corresponding author

ICMR '24, June 10–14, 2024, Phuket, Thailand.

ACM ISBN 979-8-4007-0619-6/24/06
https://doi.org/10.1145/3652583.3658012

1 INTRODUCTION

Humans interact with the physical world mainly with their hands and bodies in daily life. Thus, in applications like virtual reality (VR) and augmented reality (AR) which aim to create a realistic digital world, the quality of the reconstructed hand is quite important to improve the user's immersive experience. In this work, we aim to propose an effective and efficient hand representation to satisfy the need of AR applications.

In all, existing hand representations can be roughly divided into three categories: skeleton representations [55, 21, 1, 12], mesh-based representations [5, 32, 8, 24], and implicit representations [17, 9]. Different hand representations have their own advantages in various applications. Hand skeleton representation is a set of 3D Euclidean points and a kinematic tree. While it lacks surface information, it meets the need of the hand gesture recognition task [14, 10]. The mesh-based representation consists of Euclidean vertices and triangular faces and can directly interact with a virtual object since it offers surface information. However, compared with 3D keypoint coordinates, the pose and shape parameters of the most-used hand mesh representation—MANO [42] are more difficult for a neural network to learn. This observation is consistent with the previous finding [58], where they demonstrate that the discontinuity of axis-angle affects the performance of pose estimation.

On the other hand, implicit representation has obtained increasing attention in human body [11, 35], hand [17, 9, 18] and heads [3,

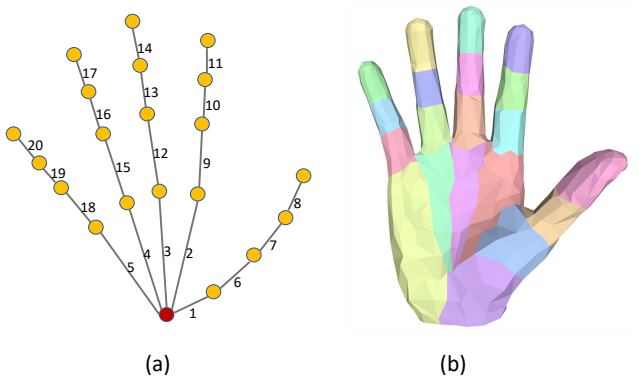

Figure 1: (a) display the order defined in hand kinematic tree. Red point is the wrist point. (b) illustrates of the pose-aligned mesh decomposition.

52, 56] modeling, due to its impressive capability in continuously modeling 3D shapes. While implicit representation methods cannot be easily employed in real-time application scenarios like AR, due to the huge computation demand of rendering explicit surfaces from implicit function networks. Concretely, the discrete 3D volume data, $\mathbb{R}^{N\times N\times N}$, is indispensable when using the Marching Cubes technique [27] to obtain explicit surfaces from implicit functions. However, for gaining the volume data, the neural implicit function would have to forward-propagate N^3 times, which means the compute cost is several orders of magnitude greater than the implicit network.

In the AR industry, the real-time interaction of virtual objects with hands plays a key role. The trustworthy interaction relies on an accurate perception of the spatial location of hands. This means the primary attribute of the hand representation in AR is representing the spatial location of a hand surface. This observation inspires us to introduce point clouds into hand representation methods since it is a natural and the simplest shape representation. On the other hand, previous studies have demonstrated that the point cloud representation supports real-time collision detection [44, 20, 36], and collision detection is substantial enough to make virtual intervention plausible. Hence, the point cloud representation can support the realistic experience in scenarios of real-time interaction with virtual objects.

Technically, following [29, 11, 17, 9], we segment the non-convex hand surface into 20 convex components, as shown in Fig. 1(b), and regress each local point cloud respectively. Each local surface corresponds to a bone in the hand kinematic tree and has an assigned order defined in Fig. 1(a). Our model parameterizes the hand surface's point cloud by a 3D hand skeleton in this geometry-disentangled way. Furthermore, for each local point cloud, we propose a Tri-Axis Modeling strategy to efficient represent local point cloud surface. This strategy allow us to precisely modeling local surface with only several layers MLP network. Experimental results demonstrate that our method has real-time inference speed and high accuracy which is comparable with implicit methods.

The main contributions are summarized as follows:

- We propose a novel neural hand model that parameterizes an explicit hand surface by the 3D coordinate space of a skeleton, and it is disentangled and parameter-sharing by our distinctive design, like *global spatial descriptor* (GSD) and Tri-Axis Modeling strategy.
- We are the first to involve a point cloud to represent the hand surface in the hand modeling task, and experiment results manifest that our method's accuracy is close to the implicit method while our inference speed outdistances it.
- We systematically research existing pose representations and discover that the coordinate-based representation of hand skeleton is more suitable for a neural network to process.

2 RELATED WORK

2.1 Auto-Decoder-Based Shape Learning

As an encoder-free framework, Auto-Decoder (AD) was first proposed by Tan *et.al.* [47], which simultaneously optimizes the latent vectors assigned to each data point and the decoder weight through back-propagation. Recently, AD has been involved in solving the 3D vision problems [37, 28, 7]. These methods approximate the Signed Distance Function [2, 51, 33, 23] or Occupancy Function [17, 11, 30, 29] of objects using AD. Their network architectures can be summarized in one general paradigm as shown at the beginning of Fig. 3. Hand and body modeling problems [17, 11] can be seen as a decoding process from a pose to a surface. They treat a hand or human pose as a decodable space in the implicit neural representation, and successfully model an articulated hand and body from a pose. Inspired by these, we introduce two kinds of decodable vectors: bones' order encoding for parameter-sharing and bones' position encoding for bone-wise reconstruction.

2.2 Implicit Human Representation

Neural implicit fields [37, 28, 7, 31, 54, 49, 17, 11, 9, 6, 48, 38, 4] have been widely researched in the 3D vision and graphics community: LISA [9] learns an implicit hand shape function and a color field from multi-view RGB video sequences; NASA [11] learns a neural occupancy field of the human body using model parameters of SMPL [26]; HALO [17] learns a hand occupancy function from the 3D hand skeleton; AlignSDF [6] proposes to jointly learn SDFs for hands and objects, with the leverage of priors provided by parametric mesh representations; Pose-NDF [48] advocates to model the unsigned distance to the manifolds of plausible human body poses in the pose space from non-Euclidean space of $SO(3)^K$.Although these implicit methods display amazing visual performance on fidelity, none of them achieve real-time performance due to the defect mentioned above. In our work, we bridge the framework of the implicit field and point cloud representation through three MLP-based Coordinate-Project Networks (CPNet). While the network architecture is simple, wide-ranging experiments validate the effectiveness and efficiency of this design.

2.3 Disentangled Representation

Disentangling parameters of certain properties, *e.g.*, the pose, shape, and color, allows the neural network to treat these properties independently. In 2D image synthesis, [22, 41, 34, 60, 45] have shown that disentangled representations are essential for learning a meaningful

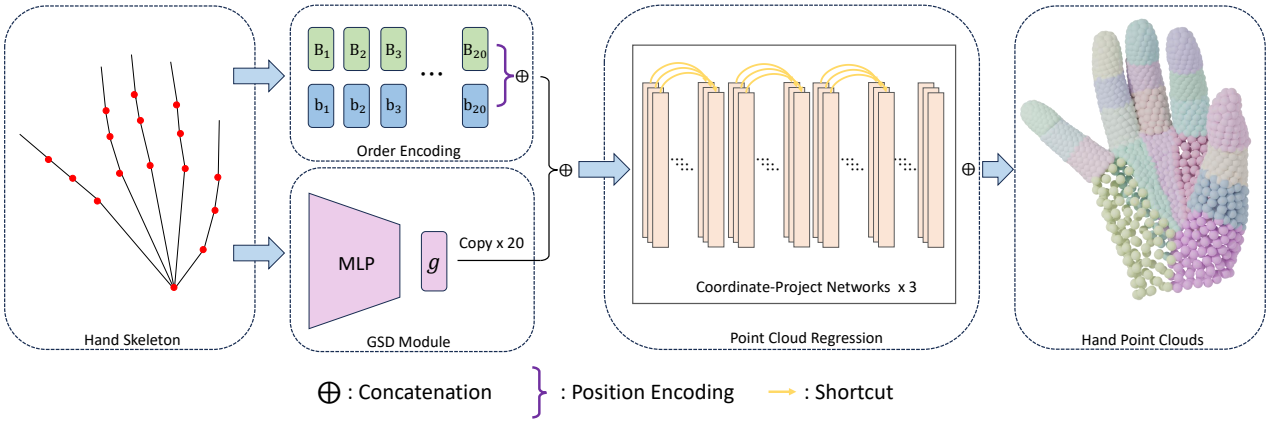

Figure 2: The overview of our method which is a simple but effective pure-MLP architecture. Firstly, a hand skeleton is separated into 20 bone vectors, $\{\mathrm{B}_i\}$, and they will be encoded through concatenating with $\{\mathrm{b}_i\}$ and position encoding. Secondly, the encoded bone vectors will be concatenated with Global Spatial Descriptor, g, as final inputs. Finally, these final inputs will be fed to the local point clouds regression network to generate 20 point cloud patches, where the local point cloud regression network consists of 3 parallel CPNets.

latent space. In the 3D community, disentanglement methods [25, 50, 56, 35, 33, 9, 29, 16] also have been receiving growing attention. NeuMesh [50] encodes the neural implicit field with disentangled geometry and texture codes on mesh vertices, which allows the network to perform mesh-guided geometry editing, as well as texture editing. A-SDF [33] factorizes shape embeddings and joint angles to model the articulated objects. LISA [9] proposes a generative hand representation with disentangled shape, pose, and appearance parameters, respectively. These methods are similar in one aspect, in that they train different networks to deal with different properties. Inspired by COAP [29] which encodes local point clouds with one-hot encoding to learn the occupancy function of the body, we involve the bones' order coding to encode different parts, making our network become parameter-sharing.

3 METHODOLOGY

In this work, we propose a novel point cloud representation of a hand. Our key insight comes from the success of the implicit representations which have validated the 3D representation capability of an MLP. Inspired by this, we propose a coordinate-wise generation strategy that regresses the xyz coordinates of the point cloud respectively by three CPNet which is also a full MLP design like implicit methods. We adopt a ***Divide-and-Conquer*** rule to reconstruct the non-convex surface of hands from skeletons as we explained in Sec. 1. Fig. 2 depicts the overall framework of our method: 1) order encoding encodes each bone vector to achieve parameter-sharing; 2) The GSD module offers supplemental spatial information, $\mathbf{g}$, to alleviate the local ambiguity problem; 3) encoded bone vectors and the $\mathbf{g}$ are concatenated and fed to the point cloud regression module to produce hand point clouds. Our experiments validate this simple but effective framework.

3.1 Order Encoding

Our neural parametric model can be described as a mapping from a root-relative hand pose space $\mathcal{P}$ to a space of point clouds on the hand surface. $\{\mathbf{B}_i\} \subset \mathbb{R}^6$ and $\{\mathbf{b}_i\} \subset \{0,1\}^{20}$ are respectively bone vectors and one-hot code of bones' order defined in Fig. 1(a). $\{\mathbf{B}_i\}$ are defined as follows:

$$\mathcal{P} := \left\{\mathbf{J}_i - \mathbf{J}_{wrist} \mid \mathbf{J}_i \in \mathbb{R}^3,\ i = 1, 2, \cdots, 21\right\}, \tag{1}$$

$$\{\mathbf{B}_i\} := \left\{(\mathbf{J}_i^0, \mathbf{J}_i^1) \mid \mathbf{J}_i^0, \mathbf{J}_i^1 \in \mathcal{P};\ i = 1, 2, \cdots, 20\right\}, \tag{2}$$

where $\mathbf{J}_{wrist} \in \mathcal{P}$ is the absolute 3D coordinates of the wrist shown as the red point in Fig. 1(a), $\mathcal{P}$ is the set of root-relative key-points of the skeleton, $\mathbf{J}_i^0$ is the father node of $\mathbf{J}_i^1$ in a hand kinematic tree shown in Fig. 1(a). In the order encoding phase, the bone vectors will be concatenated with $\{\mathbf{b}_i\}$, which helps the network recognize each bone vector of hands and hence achieves geometry-disentangled property.

3.2 Positional Encoding

In practice, we find that having the network directly operate on $\mathcal{P}$ input coordinates results in poor performance. This is consistent with a recent discovery [39] that deep networks are biased towards learning lower frequency functions. In [31], Mildenhall *et.al.* resolve this problem via the **Positional Encoding** technique, $\kappa(\cdot)$, which maps from R into a higher dimensional space $\mathbb{R}^{2L}$. We also incorporate $\kappa(\cdot)$ into our model, which maps input to a higher dimensional space before passing it to the network. The formulation of $\kappa(\cdot)$ is follow:

$$\begin{aligned}\kappa(x) =& (\sin(2^0\pi x), \cos(2^0\pi x), \cdots, \sin(2^{L-1}\pi x),\\ & \cos(2^{L-1}\pi x)),\end{aligned} \tag{3}$$

where $x \in [-1, 1]$, and the $\mathcal{P}$ will be scaled into an appropriate region to satisfy Eq. (3). We demonstrate the usefulness of positional encoding in our ablation study.

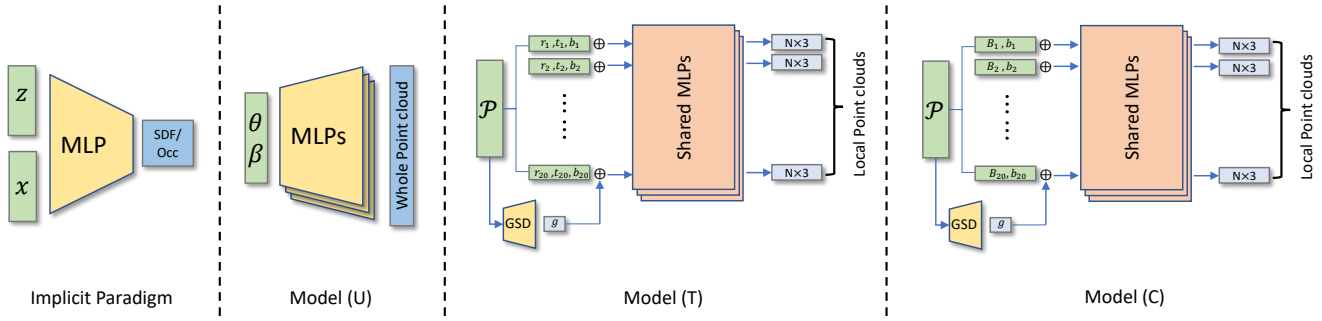

Figure 3: An illustration of the implicit paradigm, baselines (Model (U) and (T)), and our model (Model (C)). The implicit paradigm usually has a single MLP and outputs a single value. Our explicit paradigm has three parallel CPNets and outputs explicit geometry representation — Point Clouds.

3.3 Global Spatial Descriptor g

One basic problem in the distributed generation strategy is the uniqueness of the local pose of each bone's representation. In non-palm bones, two vertices and a root keypoint commonly define a plane that simultaneously defines one local coordinate system as depicted in Fig. 4.

The 6D pose of a bone can be confirmed by them. However, in the case of palm bones, no additional information can determine the local system, and this results in the network preferring to learn the mean shape of palm samples. We call this phenomenon the problem of local ambiguity in distributed generation and demonstrate it in Table 2. To address this problem, we introduce the Global Spatial Descriptor g, which takes $\mathcal{P}$ as input and outputs additional information to anchor the palm bones. The formulation of $\mathbf{g}$ is:

$$\mathbf{g} = \mathcal{G}_\eta(\mathcal{P}), \tag{4}$$

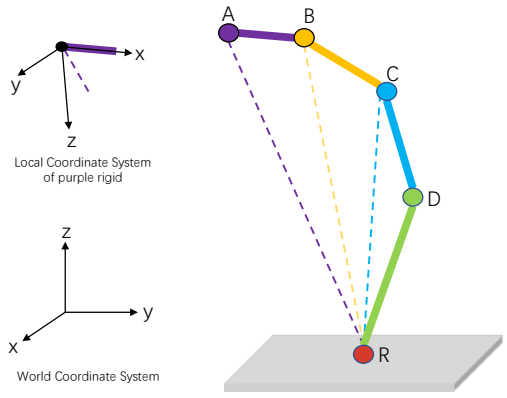

Figure 4: An illustration of the local ambiguity problem in the palm. The 6D pose of the purple rigid body can be computed by the pre-defined world coordinate system and local coordinate system. In the local system of the purple part, we set vector AB as the x-axis, the cross-product between vectors AR and AB as the y-axis, and the z-axis as the cross-product of the x-axis and y-axis.

where η is the learnable parameter of network $\mathcal{G}_\eta(\cdot)$.

3.4 Point Clouds Regression

For regressing the surface point cloud from each bone, we propose a Tri-Axis modeling strategy using three parallel CPNets to respectively predict the coordinate: x, y, z of N points and concatenate them as the final output. For every local point cloud, we have:

$$\begin{aligned} F_\omega(\kappa(\mathbf{B}_i, \mathbf{b}_i), \mathbf{g}) &= f^1_{\omega_1}(\kappa(\mathbf{B}_i, \mathbf{b}_i), \mathbf{g}) \oplus f^2_{\omega_2}(\kappa(\mathbf{B}_i, \mathbf{b}_i), \mathbf{g}) \\ &\oplus f^3_{\omega_3}(\kappa(\mathbf{B}_i, \mathbf{b}_i), \mathbf{g}) \\ &= \mathbf{v}^x \oplus \mathbf{v}^y \oplus \mathbf{v}^z \\ &= \bigcup_{i=1}^{N} \mathbf{v}_i \end{aligned} \tag{5}$$

where $\oplus$ denotes the concatenation, f^i_ω and $\omega_i, i = 1, 2, 3$ represent three CPNets and their network parameters, $\mathbf{b}_i$ is one-hot coding to encode the order assigned in Fig. 1(a), the $\mathbf{g}$ is the Global Spatial Descriptor, $\{\mathbf{v}^x, \mathbf{v}^y, \mathbf{v}^z\} \in \mathbb{R}^N$ are respectively the x, y, z coordinate vector of the local point cloud with the size of N, $\mathbf{v}_i \in \mathbb{R}^3$ is the i-th point in local point cloud, $\kappa(\cdot)$ is applied separately to each of the real values in $\mathbf{B}_i$ and $\mathbf{b}_i$, and $F_\omega(\cdot)$ is the local point cloud regression network.

The specific architecture of CPNet consists of a 17-layer Fully-Connected neural network with three empirical shortcut connections [15] which is shown in Fig. 2. Each CPNet takes bones' spatial information and $\mathbf{g}$ as input and outputs one coordinate of the corresponding point cloud. The bone-corresponding information consists of one-hot encoding, $\mathbf{b}_i$, which achieves parameter-sharing, and bone's spatial representation. We discuss the effects of different hand pose representations by our baselines.

3.5 Baselines and our Model

Existing pose representations are the axis-angle representation in MANO, transformation matrix in [17, 30, 11, 29], and 3D keypoint coordinates. Previous work [58] finds the defect when taking the axis-angle representation as output in the regression task, and we find a similar defect when taking it as input. For demonstration, we construct two baselines from two different pose representations and compare their performances with our model: 1) Unstructured

Model (U) takes as input MANO-type axis-angle representation of pose and shape parameter; 2) Transformation-based Model (T) takes as input the axis-angle of the rotation matrix and translation vector. These two models are shown in the middle of Fig. 3. In our proposal, we advocate parameterizing the hand surface into 3D keypoint coordinates to avoid the representation issues of the 3D rotation parameters. Consequently, we refer to our approach as a coordinate-based Model (C).

Unstructured Model (U). Similar to NASA [11], Model (U) does not explicitly encode the knowledge of an articulated hand. Its input consists of shape parameter $\boldsymbol{\beta}$ and pose parameter $\boldsymbol{\theta}$ in MANO [42], where $\boldsymbol{\theta} \in \mathbb{R}^{3\times 15+3}$ is composed of the axis-angle representation of 15 non-palm hand bones' relative rotation with respect to its parent in the kinematic tree and the global rotation. And Model (U) outputs a global point cloud of the hand. It is formulated as:

$$\mathcal{F}_\omega(\boldsymbol{\theta}, \boldsymbol{\beta}) = F_\omega(\boldsymbol{\theta}, \boldsymbol{\beta}) = \bigcup_{i=1}^{N} \mathbf{v}_i, \tag{6}$$

where ω is the network parameters, N is the size of generated points in our experiments, $\mathbf{v}_i$ represents the i-th point in the final point cloud and $F_\omega(\cdot)$ is similar to Eq. 5, which also consist of three CPNets, but has difference in input. **Note** there is no $\mathbf{g}$ in Eq. 6 because the $\boldsymbol{\theta}$ already contains a series of rotations which transform the resting skeleton to the target skeleton and Model (U) does not take the distributed generation strategy.

Transformation-based Model (T) explicitly utilizes the knowledge of an articulated hand, which decomposes the highly non-convex hand into a set of convex components as shown in Fig. 1(b) and generates each part individually. [17, 9, 11] have already demonstrated that the transformation matrices for each bone can well encode the local geometry information. To provide pose information in our distributed generation network, we compute the axis-angle, $\mathbf{r}_i$, of the rotation from each bone B_i in the current pose $\mathcal{P}$ to the corresponding bone $\mathbf{B}_i^*$ in a standard pose $\mathcal{P}^*$ and the translation vector $\mathbf{t}_i$. Although this representation is different from previous works [17, 9, 11], the performance in the experiment demonstrates its feasibility. Specifically, Model (T) can be formulated as:

$$\mathcal{F}_\omega(\mathcal{P}) = \bigcup_{i=1}^{20} F_\omega(\kappa(\mathbf{r}_i, \mathbf{t}_i, \mathbf{b}_i), \mathbf{g}) = \bigcup_{i=1}^{20}\bigcup_{j=1}^{N} \mathbf{v}_{ij}, \tag{7}$$

where N is an adjustable number of generated points in our experiments, ω is the network parameter, $\mathbf{g}$ is the Global Spatial Descriptor, $\mathbf{v}_{ij}$ is the j-th point of the i-th point cloud, $\{\mathbf{b}_i\}$ are one-hot coding to encode the order assigned in Fig. 1(a) and $F_\omega(\cdot)$ is similar to Eq. 5, which also consist of three CPNets, but has a difference in input.

Coordinate-based Model (C) does not take the rotational representation as input, but directly takes two endpoints' coordinates of each bone, $\{\mathbf{B}_i\}$, as input. Hence, Model (C) is formulated as:

$$\mathcal{F}_\omega(\mathcal{P}) = \bigcup_{i=1}^{20} F_\omega(\kappa(\mathbf{B}_i, \mathbf{b}_i), \mathbf{g}) = \bigcup_{i=1}^{20}\bigcup_{j=1}^{N} \mathbf{v}_{ij}, \tag{8}$$

where N is an adjustable number of generated points in our experiments, $\mathbf{g}$ is the Global Spatial Descriptor, $\mathbf{v}_{ij}$ is the j-th point of the i-th point cloud, ω is the parameter of our network and $F_\omega(\cdot)$ follows to Eq. 5. A comprehensive comparison of Model (C) with these two baselines enables us to identify the most suitable pose representation for neural network learning.

3.6 Loss Functions

Our model is trained to produce the point clouds of each bone with the corresponding ground truth as supervision. For training objectives, we think about three loss items: Chamfer distance (CD) [13], Earth Mover's distance [43] (EMD), and surface constraint.

Chamfer Distance. CD measures the distance between two point clouds by summing the squared distances of the nearest neighbor correspondences. Given two point sets: $S_1, S_2 \subseteq \mathbb{R}^3$, which denote predicted points and ground truth, respectively. The formulation of Chamfer Distance can be described as:

$$\begin{aligned} d_{CD}(S_1, S_2) = &\frac{1}{|S_1|}\sum_{\mathbf{x}\in S_1} \min_{\mathbf{y}\in S_2} ||\mathbf{x}-\mathbf{y}||_2^2 + \\ &\frac{1}{|S_2|}\sum_{\mathbf{y}\in S_2} \min_{\mathbf{x}\in S_1} ||\mathbf{y}-\mathbf{x}||_2^2. \end{aligned} \tag{9}$$

Strictly speaking, d_{CD} is not a distance function, because the triangle inequality does not hold. However, CD produces reasonable and high-quality results in practice.

Earth Mover's Distance. Different from the Chamfer distance, Earth Mover's distance requires two sets of points of the same size. The formulation of Earth Mover's distance can be described as:

$$d_{EMD}(S_1, S_2) = \min_{\theta: S_1 \to S_2} \frac{1}{N}\sum_{x\in S_1} ||\mathbf{x} - \theta(\mathbf{x})||_2^2, \tag{10}$$

where $\theta : S_1 \to S_2$ is a bijection, and S_1, S_2: $|S_1| = |S_2| = N$.

Please note that EMD requires equal size for two point sets. In most cases, we do not take it as the loss item unless the number of sampled points is equal to the number of our generating points.

Surface constraint. In this work, we apply a surface constraint [40] to speed up the convergence of the training process. It is formulated as follows:

$$\mathcal{L}_{surf} = \frac{1}{20\cdot N}\sum_{i=1}^{20}\sum_{j=1}^{N} \min_{\mathbf{c}_k\in\mathbf{M}} d(\mathbf{x}_{ij}, \mathbf{c}_k), \tag{11}$$

where $\mathbf{x}_{ij}$ means the predicted j-th point of the i-th bone, $\mathbf{c}_k$ is a triangular patch of the ground-truth mesh $\mathbf{M}$, and $d(\mathbf{x}_{ij}, \mathbf{c}_k) = \min_{\mathbf{y}\in\mathbf{c}_k} d(\mathbf{x}_{ij}, \mathbf{y})$ where $d(\cdot)$ means the L2 distance.

To this end, the final loss function is:

$$\mathcal{L} = d_{CD} + \lambda \cdot \mathcal{L}_{surf}, \tag{12}$$

where λ is the balance weight and we set it as 10 in our experiments. When satisfying the number requirement for d_{EMD}, we replace d_{CD} with d_{EMD} in Eq. (12).

4 DATASETS

Complement Dataset [5] is a synthetic dataset to resolve long-tailed distributions in popular benchmarks. In this dataset, each finger has two states: total bending and extending. This dataset contains 32 base poses according to the combination of five-finger states. For 496 pairs of base pose pairs, [5] produced three new poses through uniformly interpolating between poses in Maya software. All these 1,520 poses are the same kind of uniform samples

in the pose space without wrist rotations. To sample wrist rotation, [5] uniformly placed 216 cameras arranged in a hemisphere. The entire dataset contains 328,320 samples and all hands are the same in identity. To further study an optimal pose representation, we experiment with our model and other baselines on a small training dataset. In this experiment, our training set only contains 32 base poses (32 × 216 meshes) and our test set consists of 1,520 interpolated poses (1520 × 216 meshes).

FreiHAND [61] contains 130,240 training images and 32,560 training meshes from 32 subjects of different genders and ethnic backgrounds and we only use mesh data in our experiment. We demonstrate the high fidelity of our method and the ability to represent different hand shapes.

Data Preparation. We use ground-truth point clouds to supervise the training stage with the reason that the uniformity of generated point clouds cannot be guaranteed if we only use meshes for supervision. To satisfy the requirement of distributed reconstruction, we segment each mesh according to the corresponding pose, then we implement Poisson disk sampling [53] to obtain bone-corresponding point clouds to achieve distributed supervision. The disintegration way is shown in Fig. 1.

5 EXPERIMENTS

5.1 Implementation Details

We use the Adam optimizer [19] to train the network with a mini-batch size of 256. Models in experiments on the Complement Dataset are trained for 4,000 epochs and models in experiments on FreiHAND are trained for 800 epochs. The initial learning rate is 5×10^{-4}, multiplied by 0.5 for every 300 epochs on FreiHAND, and for every 800 epochs on the Complement dataset. In our experiments, every generated point cloud of bones has 100 points, and the supervised point cloud of bones has 500 points. When using EMD as a loss item, the point clouds of bones for supervised learning are 100 points to match the number of generated points. Unless otherwise specified, the number of neurons of hidden layers in CPNet and GSD is set to 256. For positional encoding, we set $L = 2$ for bones' one-hot encoding on Model (C) and Model (T) and $L = 5$ for all non-bones' order elements. We set $N = 100$ on Model (T) and (C) and set $N = 2000$ on Model (U) for a fair comparison.

5.2 Evaluation Criterions

We evaluate the quality of the generated point cloud from two aspects: *accuracy* and *uniformity*.

MP2S measures the mean per point position error with Euclidean distance (in millimeters) between the estimated point cloud and the ground-truth mesh. This metric evaluates the accuracy of the estimated point cloud and its calculation method follows Eq. (11).

We report the mean of each part's CD, **MCD**, as a uniformity metric (in mm^2). Furthermore, we report **MEMD** (in mm^2) and **MHD** (in mm) for comprehensive measuring the uniformity, where the CD and EMD respectively follow Eq. (9) and Eq. (10). HD means the Hausdorff Distance and is formulated as:

$$d_{HD}(S_1, S_2) = \max\{\sup_{\mathbf{x}\in S_1}\inf_{\mathbf{y}\in S_2} d(\mathbf{x},\mathbf{y}), \sup_{\mathbf{y}\in S_2}\inf_{\mathbf{x}\in S_1} d(\mathbf{x},\mathbf{y})\}, \quad (13)$$

Table 1: Quantitative results on FreiHAND dataset. † indicates the model without GSD module. ‡ indicates the reproduced Pose2Mesh method.

Method	MP2S↓	MCD↓	MEMD↓	MHD↓
MobRecon [5]	5.53	361.44	166.35	16.74
MeshGraphormer [24]	5.18	247.51	125.77	15.29
Model (C)+N(0,50)	4.49	200.25	129.21	19.26
Model (C)+N(0,20)	3.12	81.37	56.41	13.11
Pose2Mesh‡ [8]	2.32	12.75	7.88	5.89
HALO [17]	**0.39**	**5.11**	–	5.43
Model(T)†	1.85	35.68	59.97	9.54
Model(C)†	1.79	41.84	57.33	9.63
Model(U)	1.80	49.23	–	17.70
Model(T)	0.60	6.07	6.38	<u>4.47</u>
Model(C)	<u>0.45</u>	<u>5.36</u>	**5.96**	**4.08**

Best; <u>Second best</u>.

where $d(\cdot)$ is L2 distance, S_1 and S_2 are two point sets. HD can well measure the noise of prediction, because it is sensitive to outliers.

Infer time indicates the inference time of the model, and the unit is a millisecond.

Mult-Adds counts multiply-add operations.

5.3 Model Evaluation

5.3.1 Quantitative Evaluation. On the **FreiHAND** dataset, we report the performance of our model when training on rich data, and compare it with contemporary methods, as shown in Table 1. Pose2Mesh [8] reconstructs the hand surface from 2D pose to 3D pose and then to the final MANO-style mesh. For a fair comparison, we reproduce their method which reconstructs mesh directly from the 3D pose. In addition, we compare our model with MobRecon [5] and MeshGraphomer [24] which are the state-of-the-art monocular hand reconstruction methods, and we introduce *Gaussian noise* to input for fairness. For these mesh reconstruction methods, we first obtain the root-aligned mesh predictions, and then we segment meshes following the way in Fig. 1 and finally implement Possion disk sampling [53] to obtain point clouds for each part. All metrics are computed in this way. HALO [17] is an implicit method, we report its P2S as the accuracy reference. For the metric of MCD and MHD of HALO, the number of vertices of the mesh obtained by Marching Cubes [27] is usually greater than 80k. For fairness, we downsample the size of vertices to ~2k points through [53], then compute the MCD and MHD. The MEMD of HALO and Model (U) cannot be computed as the EMD needs two point clouds of have same size which HALO and Model (U) are both not satisfied in local areas.

In Table 1, Model (C) achieves state-of-the-art accuracy in the explicit representation and has compelling performance compared with the implicit representation. From the metric results of three baselines, we can conclude that: Similar to NASA [11], the structured model (T) and (C) which learn the articulated hand model via decomposition have striking advantages compared with the unstructured model (U), as quantified by the fact that the MP2S

Table 2: Statistics of CD of palm bones on FreiHAND test set. † indicates that the model does not have the GSD module.

Method	1st	2nd	3rd	4th	5th
Model(C)†	174.22	107.34	144.90	133.51	130.49
Model(T)†	161.24	95.89	110.31	101.93	98.85
Model (T)	9.84	10.28	7.76	6.97	16.05
Model (C)	**9.76**	**9.56**	**7.03**	**6.48**	**14.69**

Table 3: Quantitative results on Complement dataset.

Method	MP2S↓	MCD↓	MHD↓
Model(T)+HALO [17]	6.10	361.16	23.35
Model(T)+6D [58]	4.38	187.16	17.17
Model(T)	3.69	122.57	14.84
Model(C)	**1.60**	**29.31**	**9.64**

Table 4: Statistics of inference cost about different methods on the FreiHAND test set.

Method	Infer time↓	Forward Times↓	Total Mult-Adds (M)↓	MP2S↓
HALO [17]	4283.47	65^3	65910	**0.39**
Pose2Mesh‡ [8]	18.94	**1**	579.34	2.32
Model (C)	**5.64**	20	**76.05**	0.45

of (T) and (C) are respectively lower by 67% and 75% than the (U); Compared with the axis-angle representation of the pose, the coordinates of the pose is more suitable for the neural network to learn; our GSD module significantly improves the quality of generated point clouds. Indeed, the advantages of Model (C) are more remarkable when the training data is deficient as shown in Table 3.

Analysis of GSD module. As discussed in [17, 46], the local coordinate system can be defined by one bone and its father in the kinematic tree. Similarly, the 6D pose of i-th non-palm bone can be obtained by $\mathbf{B}_i$ and the wrist point. Without loss of generality, we explain this on an articulated object which has the same structure as a human hand. As depicted in Fig. 4, the purple rigid part corresponds to a non-palm bone, and its local coordinates system can be computed by the purple dotted line, because these two lines are not collinear. The orange and blue parts' local coordinate systems can also be obtained in this way. However, no other fixed points can help to determine the local coordinate system of the light green part. This is what we call the local ambiguity problem in distributed generation. To address this problem, we introduce a GSD module to provide additional spatial information. In Table 2, we quantitatively report CD of palm bones to better demonstrate the effect of the GSD module, and the order of i-th follows the defined order of the bones in Fig. 1(a). Note that similar improvement also occurs in Model (T), because $\{\mathbf{r}_i, \mathbf{t}_i\}$ contain information that is equivalent to $\{\mathbf{B}_i\}$.

On the **Complement dataset [5]**, all test poses are obtained by interpolation of training poses in our data split. In this experiment, we demonstrate that the coordinate-based pose has better generalization ability than the axis-angle-based method. For a comprehensive comparison, we conduct modifications on Model (T) to additionally compare our coordinate representation with other pose representations. We report the experimental results in Table 3 where "+6D" means that we replace axis-angles in Model (T), $\{\mathbf{r}_i\}$, to a continuous 6D rotation representation proposed by Zhou *et.al.* [58], and "+HALO" means we set the transformation matrices proposed in HALO [17] as input. It turns out that the coordinate-based method is better than all rotation-based methods. And this experiment also proclaims that our MLP-based point cloud representation has excellent interpolation ability like the implicit method, and draws a conclusion that in hand modeling, the coordinate representation of pose is better than the usual axis-angle representation.

Inference time. In industry, the Multi-Adds is closely related to the inference time of the neural network. The Multi-Adds of Model (C) is only 3.84M for per bone's inference and is 76.05M for the whole hand. On an NVIDIA TESLA V100 GPU, the inference from a hand pose to a whole point cloud requires just 5.64 milliseconds. In Table 4, we compare the cost for obtaining the explicit output from three hand representation methods, where ‡ indicates the reproduced Pose2Mesh method and Forward Times is the number of forward inferences required for each model to obtain the explicit surface. All tests are conducted on one NVIDIA TESLA V100 GPU. As we mentioned before, the implicit method (HALO) requires forward many times to obtain volume data for Marching Cubes [27], and this makes its final multiply-add operations overwhelm other explicit methods. Our point cloud representation has less compute cost and better accuracy than the Pose2Mesh[8].

5.3.2 Qualitative Evaluation. In this section, we mainly present the geometry accuracy of three different learning-based representations in Fig. 5. As displayed in Fig. 5, our method can reconstruct precisely explicit geometry and outperforms the mesh-based method [8]. In some cases, our method even outperforms the implicit model [17] while the MP2S of HALO is less than the MP2S of Model (C). It is reasonable to consider Model (C) since its standard deviation of P2S and the mean of max P2S of HALO are $(0.425mm, 3.57mm)$ and $(0.413mm, 3.26mm)$, respectively. This means that the geometry reconstruction quality of our Model (C) is better. Besides, Model (C) also has better MHD than HALO, which also indicates that our shape predictions are stabler.

5.4 Application

Mapping Skeleton to Surface. Effectively and efficiently mapping the hand skeleton to the surface for downstream tasks, for example collision detection, has great application value. And our method can directly achieve this goal effectively. To demonstrate the advantage of our method, we compare our model with an iterative-optimation-based IK solver[1] and a neural-network-based IK solver, IKNet [59], in Table 5. To discuss the influence of large global rotations, we report the experiment results of the iterative baseline with and without the consideration of global rotations. Experiments show the iterative IK method has better performance when the estimated skeleton is similar to the rest pose (like no global rotation). However, our method overwhelms the iterative method and also defeats IKNet which directly regresses the MANO parameters from a 3D skeleton.

[1] https://github.com/CalciferZh/Minimal-IK

Figure 5: The visual comparison of different methods on FreiHAND test set. For every instance, we display two different views. For HALO, we downsample its vertices from $\sim 80k$ to $\sim 2k$ using [57].

This demonstrates the difficulty of learning MANO parameters for a neural network, and our method is a better proposal to map a sparse 3D skeleton to a dense hand surface.

Table 5: Skeleton to Surface Mapping.† indicates the results regardless of the global rotation.

Methods	MP2S↓	MCD↓	MHD↓
Iterative Baseline	6.64	2539.52	34.44
Iterative Baseline†	2.57	49.02	9.37
IKNet	5.41	203.91	16.49
Model (C)	**0.45**	**5.36**	**4.08**

6 ABLATION STUDIES

Effect of the number of supervised point clouds and loss items. The number of supervised points is a significant consideration when using CD to supervise the training. We report experiments on the different numbers of supervised points in Table 6. On the FreiHAND dataset, the results show that the use of more points leads to better uniformity and accuracy. It is reasonable that the latter part of CD (in Eq. (9)) is a kind of uniformity constraint when the $|S_2| > |S_1|$. While the supervision of 300 points achieves the best results on the Complement dataset, the result with 500 supervision points is still compelling. We suppose this difference is caused by the limited training data. Furthermore, we report the quantitative results of different loss items. Similar to CD, EMD is also a frequently used loss function in point cloud generation. The results claim when the supervision points have equal size with generation points, taking EMD as a loss item increases the uniformity and the smoothness but decreases the accuracy of the generated point cloud. The EMD is an alternative loss in our method.

Table 6: Ablation study of supervision point cloud on Complement dataset.

Supervision	Loss	Complement			FreiHAND		
		MP2S↓	MCD↓	MHD↓	MP2S↓	MCD↓	MHD↓
100 points	EMD	1.67	<u>28.73</u>	**9.14**	0.53	5.73	**3.91**
100 points	CD	1.66	30.55	9.88	0.57	6.07	4.29
200 points	CD	<u>1.60</u>	28.89	9.66	0.53	5.87	4.26
300 points	CD	**1.59**	**28.00**	<u>9.53</u>	<u>0.51</u>	<u>5.66</u>	4.21
400 points	CD	1.62	28.74	9.84	0.54	5.79	4.31
500 points	CD	<u>1.60</u>	29.31	9.64	**0.45**	**5.37**	<u>4.08</u>

Best; <u>Second best</u>.

Table 7: The ablation study of Position Encoding on the FreiHAND dataset.

Position Encoding	MP2S↓	MCD↓	MEMD↓	MHD↓
×	0.57	5.89	**5.65**	4.49
✓	**0.45**	**5.36**	5.96	**4.08**

Effect of the Position Encoding. In Table 7, we report the experiment results with and without the position encoding module. It turns out that position encoding significantly improves the accuracy and quality of generated point clouds.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we first involve the point cloud in the hand representation and develop a disentangled and parameter-sharing model based on the flexible structure of the point cloud, the articulated property of hands, and strategy of Tri-Axis modeling. Our MLP-based model has the advantage of accurately modeling geometry shapes like implicit methods and of little compute cost. Extensive experiments on public datasets validate these merits. By this model, given hand skeleton information, the surface with high fidelity can be obtained in real-time which shows application potential in AR. One main hypothesis in our work is that the identity of the hand subject and the hand bones' length are highly correlated. However, it does not hold all the time. For instance, the thickness of different persons' hand bones can be different even when their lengths are the same. Hence, exploring a controllable shape parameterization, like MANO [42], for our model is one of our future goals.

ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China under Grants 92360307, 92267103, 62172415, 62102418.

REFERENCES

[1] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. 2019. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2272–2281.

[2] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. 2020. Deep local shapes: learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 608–625.

[3] Chuhan Chen, Matthew O'Toole, Gaurav Bharaj, and Pablo Garrido. 2023. Implicit neural head synthesis via controllable local deformation fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2023), 416–426.

[4] Mingfei Chen, Jianfeng Zhang, Xiangyu Xu, Lijuan Liu, Yujun Cai, Jiashi Feng, and Shuicheng Yan. 2022. Geometry-guided progressive nerf for generalizable and efficient neural human rendering. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*. Springer, 222–239.

[5] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. 2022. Mobrecon: mobile-friendly hand mesh reconstruction from monocular image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20544–20554.

[6] Zerui Chen, Yana Hasson, Cordelia Schmid, and Ivan Laptev. 2022. Alignsdf: pose-aligned signed distance fields for hand-object reconstruction. In *European Conference on Computer Vision*. Springer, 231–248.

[7] Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5939–5948.

[8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. 2020. Pose2mesh: graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*. Springer, 769–787.

[9] Enric Corona, Tomas Hodan, Minh Vo, Francesc Moreno-Noguer, Chris Sweeney, Richard Newcombe, and Lingni Ma. 2022. Lisa: learning implicit shape and appearance of hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20533–20543.

[10] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. 2019. Heterogeneous hand gesture recognition using 3d dynamic skeletal data. *Computer Vision and Image Understanding*, 181, 60–72. DOI: https://doi.org/10.1016/j.cviu.2019.01.008.

[11] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. 2020. Nasa neural articulated shape approximation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 612–628.

[12] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. 2020. Hope-net: a graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6608–6617.

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 605–613.

[14] Qing Gao, Yongquan Chen, Zhaojie Ju, and Yi Liang. 2022. Dynamic hand gesture recognition based on 3d hand pose estimation for human–robot interaction. *IEEE Sensors Journal*, 22, 18, 17421–17430. DOI: 10.1109/JSEN.2021.3059685.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[16] Boyi Jiang, Juyong Zhang, Jianfei Cai, and Jianmin Zheng. 2020. Disentangled human body embedding based on deep hierarchical neural network. *IEEE Transactions on Visualization and Computer Graphics*, 26, 8, 2560–2575. DOI: 10.1109/TVCG.2020.2988476.

[17] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. 2021. A skeleton-driven neural occupancy representation for articulated hands. In *International Conference on 3D Vision (3DV)*.

[18] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. 2020. Grasping field: learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 333–344.

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[20] Jan Klein and Gabriel Zachmann. 2004. Point cloud collision detection. In *Computer Graphics Forum* number 3. Vol. 23. Wiley Online Library, 567–576.

[21] Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. 2022. Interacting attention graph for single image two-hand reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2761–2770.

[22] Zhiheng Li, Martin Renqiang Min, Kai Li, and Chenliang Xu. 2022. Stylet2i: toward compositional and high-fidelity text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2022), 18197–18207.

[23] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. 2020. Sdf-srn: learning signed distance 3d object reconstruction from static images. *Advances in Neural Information Processing Systems*, 33, 11453–11464.

[24] Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12939–12948.

[25] Jingwang Ling, Zhibo Wang, Ming Lu, Quan Wang, Chen Qian, and Feng Xu. 2022. Semantically disentangled variational autoencoder for modeling 3d facial details. *IEEE Transactions on Visualization and Computer Graphics*, 1–1. DOI: 10.1109/TVCG.2022.3166666.

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. Smpl: a skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34, 6, 1–16.

[27] William E Lorensen and Harvey E Cline. 1987. Marching cubes: a high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21, 4, 163–169.

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4460–4470.

[29] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. 2022. Coap: compositional articulated occupancy of people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13201–13210.

[30] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. 2021. Leap: learning articulated occupancy of people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10461–10471.

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65, 1, 99–106.

[32] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. 2020. Deephandmesh: a weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *European Conference on Computer Vision*. Springer, 440–455.

[33] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. 2021. A-sdf: learning disentangled signed distance functions for articulated shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13001–13011.

[34] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. Hologan: unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7588–7597.

[35] Pablo Palafox, Nikolaos Sarafianos, Tony Tung, and Angela Dai. 2022. Spams: structured implicit parametric models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12851–12860.

[36] Jia Pan, Ioan A Şucan, Sachin Chitta, and Dinesh Manocha. 2013. Real-time collision detection and distance computation on point cloud sensor data. In *2013 IEEE International Conference on Robotics and Automation*. IEEE, 3593–3599.

[37] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174.

[38] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021. Neural body: implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9054–9063.

[39] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the spectral bias of neural networks. In *International Conference on Machine Learning*. PMLR, 5301–5310.

[40] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. 2020. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*.

[41] Yurui Ren, Xiaoqing Fan, Ge Li, Shan Liu, and Thomas H. Li. 2022. Neural texture extraction and distribution for controllable person image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2022), 13535–13544.

[42] Javier Romero, Dimitrios Tzionas, and Michael J Black. 2022. Embodied hands: modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*.

[43] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40, 2, 99–121.

[44] Johannes Schauer and Andreas Nüchter. 2015. Collision detection between point clouds using an efficient kd tree implementation. *Advanced Engineering Informatics*, 29, 3, 440–458.

[45] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9243–9252.

[46] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. 2020. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 211–228.

[47] Shufeng Tan and Michael L Mayrovouniotis. 1995. Reducing data dimensionality through optimizing neural network inputs. *AIChE Journal*, 41, 6, 1471–1480.

[48] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. 2022. Pose-ndf: modeling human pose manifolds with neural distance fields. In *European Conference on Computer Vision*. Springer, 572–589.

[49] Kangkan Wang, Sida Peng, Xiaowei Zhou, Jian Yang, and Guofeng Zhang. 2022. Nerfcap: human performance capture with dynamic neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 1–13. DOI: 10.1109/TVCG.2022.3202503.

[50] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. 2022. Neumesh: learning disentangled neural mesh-based implicit field for geometry and texture editing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*. Springer, 597–614.

[51] Shun Yao, Fei Yang, Yongmei Cheng, and Mikhail G. Mozerov. 2021. 3d shapes local geometry codes learning with sdf. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. (Oct. 2021), 2110–2117.

[52] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. 2021. I3dmm: deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2021), 12803–12813.

[53] Cem Yuksel. 2015. Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum* number 2. Vol. 34. Wiley Online Library, 25–32.

[54] He Zhang, Fan Li, Jianhui Zhao, Chao Tan, Dongming Shen, Yebin Liu, and Tao Yu. 2022. Controllable free viewpoint video reconstruction based on neural radiance fields and motion graphs. *IEEE Transactions on Visualization and Computer Graphics*, 1–16. DOI: 10.1109/TVCG.2022.3192713.

[55] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. 2022. Mixste: seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13232–13242.

[56] Mingwu Zheng, Hongyu Yang, Di Huang, and Liming Chen. 2022. Imface: a nonlinear 3d morphable face model with implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2022), 20343–20352.

[57] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. 2018. Open3d: a modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*.

[58] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5745–5753.

[59] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. 2020. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5346–5355.

[60] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. 2018. Visual object networks: image generation with disentangled 3d representations. *Advances in neural information processing systems*, 31.

[61] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. 2019. Freihand: a dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 813–822.