Skeleton Extraction for Articulated Objects with the Spherical Unwrapping Profiles

Zimeng Zhao, Wei Xie, Binghui Zuo and Yangang Wang, *Member, IEEE*

Abstract—Embedding unified skeletons into unregistered scans is fundamental to finding correspondences, depicting motions, and capturing underlying structures among the articulated objects in the same category. Some existing approaches rely on laborious registration to adapt a predefined LBS model to each input, while others require the input to be set to a canonical pose, *e.g.*T-pose or A-pose. However, their effectiveness is always influenced by the water-tightness, face topology, and vertex density of the input mesh. At the core of our approach lies a novel unwrapping method, named SUPPLE (Spherical UnwraPping ProfiLEs), which maps a surface into image planes independent of mesh topologies. Based on this lower-dimensional representation, a learning-based framework is further designed to localize and connect skeletal joints with fully convolutional architectures. Experiments demonstrate that our framework yields reliable skeleton extractions across a broad range of articulated categories, from raw scans to online CADs.

Index Terms—Skeleton Embedding, Spherical Unwrapping, Surface-to-image Representation

1 Introduction

Advances in learning-based vision and graphics are boosting the acquisition of customized scans for humans and animals. Bringing these scans to life has the potential to enable numerous additional downstream AR/MR/VR applications, *e.g.*, telepresence, remote interaction, and *etc.*. To achieve this goal, the embedded skeleton is the most feasible method to describe pose [1], [2], perform animations [3], [4], [5] and retarget motions [6], [7], [8]. Therefore, extracting skeletons from those scans quickly and directly is urgently needed for the community.

To achieve this goal, some methods [4], [5], [9] attempt to extract the skeleton from the full body. They usually assume that articulated objects are symmetrical or pre-aligned to a unified state (T-pose or A-pose). Unfortunately, this poses a constraint when scanning is difficult to define or execute for other categories such as animals or hands. Other methods [10], [11], [12], [13] consider this task in broader categories. However, they rely on additional mesh properties, *e.g.*water-tightness or sequential consistency, which may not be easily satisfied by those raw scans.

The heatmap regression paradigm effectively estimates the articulated skeleton from 2D image [1], [14], [15]. Instead of regressing the value of target 2D coordinates, it tames a DNN to predict a confidence distribution map whose ground truth is Gaussian centered at target 2D coordinates. When adopting it to 3D space, however, obstacles arise from defects in general 3D representations. As for point cloud [16], [17], [18] and mesh [19], [20], the order and the adjacency of their data are subject-specific or topology-specific. This hinders the learning efficiency of a DNN which needs to be trained with *feature-aligned* data. On the other hand, although voxel-grid stores 3D features in a fixed order independent of object attributes, DNNs [21], [22], [23] with 3D distributions as output have expensive inference costs.

In this paper, we propose SUPPLE , a novel surface-to-image representation that **makes it possible to recast the 3D skeleton extraction task to a 2D heatmap regression task**. Here SUPPLE is the abbreviation of the **S**pherical **U**nwra**P**ping **P**rofi**LE**s, which projects a 3D surface to three 2D image plane with the spatial order and connectivity maintenance. Based on this formulation, the skeleton extraction can disengage the constraints on mesh topology, and utilize more 2D learning paradigms for 3D skeleton extraction.

Our insight to use SUPPLE for skeleton extraction is as follows: The possible 3D position of a skeletal joint is spherically distributed. Generally, each distribution is conditional on the distance from this joint to the object surface and the outer surface features around this joint (See **Sup. Mat** Sec. C for details). However, most general 3D representations (*e.g.*voxel grid, point cloud, mesh, *etc.*) are defined in the Cartesian coordinate system. They either record sparse outer surface features or rearrange these features in a non-intuitive adjacency relationship. By contrast, SUPPLE is defined in the spherical coordinate system and projects surface features with their adjacencies into three complementary 2D subspaces. These subspaces are associated with the spherical distribution because they compactly and orderly store the intersections between a given surface and spherical curves. Among them, the first subspace emphasizes more on outer surface features, and the other two subspaces compensate for the first one. Compared to the orthogonal projections defined in the Cartesian coordinate system, SUPPLE records more details and causes fewer surface overlaps.

Most existing methods predict all joint features in parallel [14], [24], [25]. This design leads to a flat network structure and ignores the hierarchical features of a skeleton.

- *This work was supported in part by the National Natural Science Foundation of China (No. 62076061), in part by the Natural Science Foundation of Jiangsu Province (No. BK20220127), by the "Young Elite Scientists Sponsorship Program by CAST"(No. YES20200025) and by the "Zhishan Young Scholar" Program of Southeast University (No. 2242021R41083).*
- *Zimeng Zhao, Wei Xie, Binghui Zuo and Yangang Wang are with the School of Automation, Southeast University, Nanjing, China, 210096.*
- *Corresponding author: Yangang Wang.*

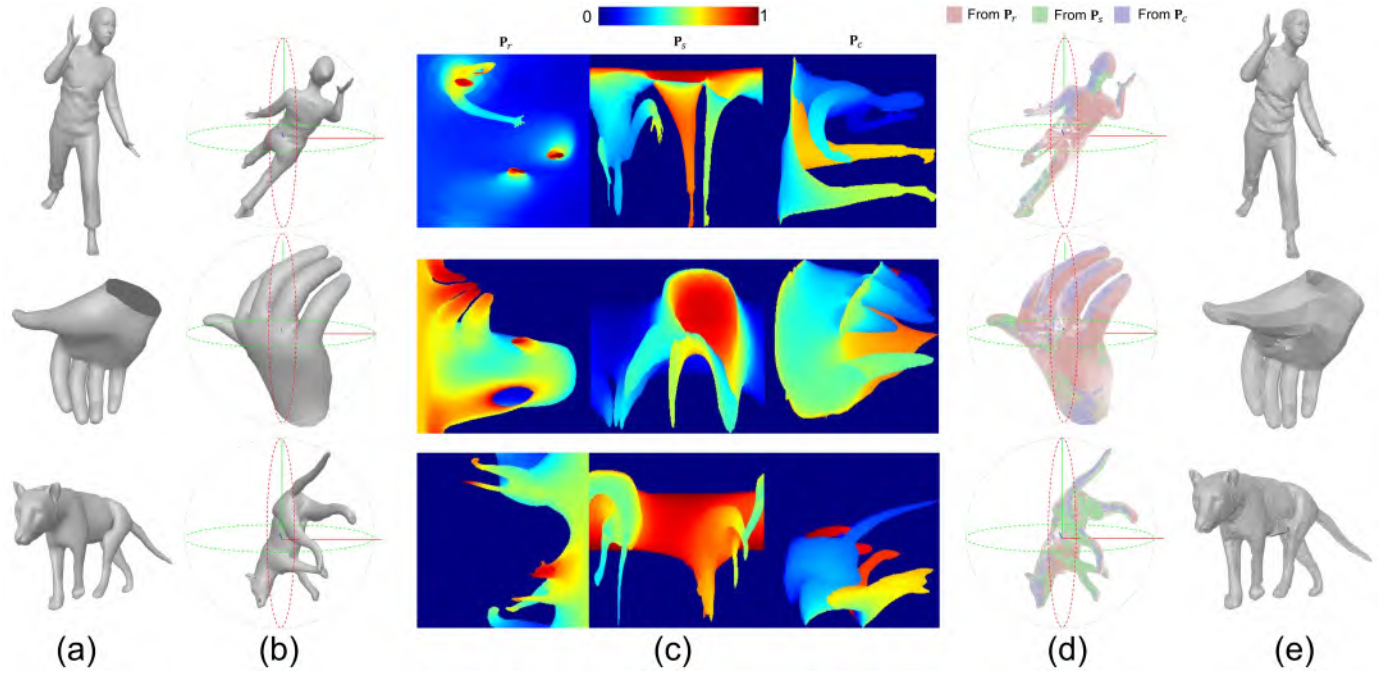


Fig. 1. **Examples of Mesh-SUPPLE conversion (better viewed in color).** (a)-(c) describe the conversion from meshes to SUPPLE images, and (c)-(e) describe the inverse conversion from SUPPLE images to meshes. During the conversion, SUPPLE retains sufficient surface features related to the skeleton. Each row corresponds to a typical instance from an articulated object category. (a) Input mesh; (b) Canonicalized mesh; (c) SUPPLE ; (d) Queried points from SUPPLE ; (e) Reconstructed surface from in-map points. For better visualization, the scalar value of each channel in SUPPLE is mapped into pseudo color with the JET format.

Relatedly, Georgakis*et al.* [26] introduced additional pose parameter refinement to each limb based on the body torso, and Chen *et al.* [25] explored more semantics by mixing features of different joints. For the first time, we regress heatmaps according to the skeletal hierarchy in a coarse-to-fine manner. Specifically, our learning framework first predicts the multi-joint heatmap of each skeletal branch, *i.e.*a heatmap containing multiple Gaussians. After that, the framework recurrently predicts the single joint heatmap according to a proximal-to-distal order and parses the corresponding 3D position according to the heatmap. All these improvements slim down the network weight and further increase its generalization capability.

The main contributions of this work are summarized as follows.

- A novel representation called SUPPLE to unwrap a 3D surface into image space effectively;
- A CNN-based framework to extract skeletons from unrestricted mesh in various categories;
- A recurrent paradigm to improve network architecture and generalization by considering skeletal hierarchy.

A preliminary version of this paper has appeared in [27], which introduces an effective method for skeleton extraction from a hand mesh without shape, pose, and topology constraints. The present work makes the following additional contributions. (i) Without the dependence of a pre-defined joint order or template, this framework predicts a skeleton in a tree structure and becomes more applicable to diverse articulated objects; (ii) By aggregating the joints in the same branch, a more lightweight LocNet localizes branch-wise joint distribution; (iii) By identifying the sequential joints within a branch recurrently, AssembleNet assembles branches according to the hierarchy; (iv) By minimizing the overlap among the three SUPPLE channels, a canonical alignment further enhances the extraction performance; (v) With more exhaustive data augmentation strategies, our framework is further enabled to extract skeletons from partial point clouds; (vi) By optimizing the calculation, the conversion process from scan mesh to SUPPLE is further accelerated, which facilities its data augmentation process and reduces the training time; (vii) By including more detailed analysis and comparisons, the advantages of our SUPPLE representation has further reveled.

2 Related Work

2.1 Skeleton Embedding

In different applications, a skeleton varies in concept from a series of curves [12], [28], [29] to a joint hierarchy [3], [4]. Here we concern more about the latter one, which is widely used in motion capture and automatic rigging to effectively describe the pose and motion of articulated objects.

Skeleton for Motion Capture. The complex motion of an articulated object is always regarded as a composite of transformation of different rigid parts [30], [31], [32]. When describing its pose at a specific time, the keypoints between adjacent parts are regarded as joints (sometimes including endpoints), and the connection between points follows the kinematic characteristics. Traditional methods utilized visible markers [33], [34] to track joint trajectories and capture accurate motion, which may hinder the natural motion of the objects. In recent years, with the popularity of data-driven methods, markerless capturing approaches show promising prospects because they can reduce the capturing costs and eliminate hardware calibration. Most of them consider color image [1], [15], [19], [35] or depth

image [22], [36], [37], [38], [39] as input to extract skeleton features. Most of them take monocular images as input to regress joint coordinates [35], mesh vertex coordinates [19], joint rotation axis-angles(pose parameters) [40], [41], [42], [43], [44], [45], or joint distribution (heatmap) [1], [15]. Among them, the heatmap paradigm is borrowed by most subsequent works since it allows for accurately localizing the joints in the image via per-pixel predictions. To further obtain accurate 3D surfaces or skeletons, these 2D schemes require images from multiple perspectives [46], [47] as well as camera parameters, which greatly increases the cost of data storage. Therefore, recent work [18], [48], [49], [50], [51], [52] on skeleton extraction directly from 3D data has been explored. However, most of them take point cloud data as input, because it can be quickly obtained from depth images. With the development of 3D reconstruction methods [53], [54], [55] and commercial scanning technology in recent years, it is also becoming easier to obtain plausible but unregistered meshes. Therefore, the algorithm proposed in this paper considers the skeleton extraction problem with unregistered mesh as input. Compared to the point cloud, it has more geometric features but may still have surface holes, noises, or even multiple surface layers.

Skeleton for Automatic Rigging. Most automatic rigging methods [4], [5], [56] aimed to embed (or extract) a skeleton of character for motion control. These character meshes are mostly designed as symmetrical structures by artists and placed under simple scan poses like T-pose and A-pose. Other work [10], [13], [57], [58] performed this task with the dependence of multiple pose references under the same mesh topology, which could not be guaranteed by some scanning results from a single object. In addition, their final joint number and layout could not be adjusted flexibly and are greatly affected by the number and pose variation of input examples.

Skeleton for Mesh Registration. The purpose of registering different original scans to a unified parametric skin model is to obtain the time-invariant mesh topology of the instance surface and further enable the deformation analysis. Most of the work takes the skeleton as the initial configuration, either optimization-based [59], [60] or learning-based [61], [62] methods. The former ones require the creation of non-linear or non-convex loss functions, as well as the optimization of that loss with the joint rotation of the articulated body. Using the inverse kinematic solution between the two skeletons as the initial value can effectively avoid the optimization process from falling into a local optimum. The latter approach finds matches between points through a pre-trained implicit function. However, most existing implicit methods [63], [64], [65] for articulated objects rely on bone transformations of the skeleton as the input.

2.2 Surface Representation

In the realm of 2D learning, there exist dominant representations and paradigms [66], [67], [68]. These topics, however, are still in their infancy in 3D learning.

Explicit Data. Explicitly representing a surface means approximating a continuous surface with discrete primitives, *e.g.*points, or patches. It includes multi-view stereo, voxel grids, point clouds, and mesh. Since a 3D object can be represented as a series of images taken by an array of cameras around it, the method of understanding 3D objects directly based on multi-view RGB or depth images has been continuously concerned [69], [70], [71]. However, the number and placement of cameras vary greatly for different subjects and problems. Voxelization of a surface is a natural extension of the 2D image space, and numerous learning methods [21], [72], [73], [74] based on it have been performed for 3D learning. Anyhow, the surface details would often be sacrificed when limiting its cubic memory requirements. Although this can be ameliorated with the introduction of octrees [75], [76], existing methods have still limited the resolution to small orders of magnitude. Point cloud data is usually directly captured by depth sensors. Corresponding surface information can be reconstructed through multiple post-processing [77], [78], [79]. The thorny problem that data-driven methods created based on it have to face is the disorder within a point cloud. Sampling and neighborhood aggregation [18], [80], [81], [82] are the two most commonly used operations, and their effectiveness depends on repeated distance matrix calculation. Therefore, it is inefficient to extract features from point cloud data. Mesh can effectively describe the deformation of a specific object with pre-defined vertices and faces. Therefore, it is popular in the animation field. For articulated object animation, skinning approaches [3], [83], [84] represent the transformation of the mesh vertices as a combination of the individual skeleton joint transformations. Some improved methods [30], [31], [85], [86], [87] further combined with learning weights to improve the artifacts in the deformation process. However, it is difficult to compare meshes with different topologies, and additional registration operations are needed to align the features. Some data-driven methods use graph convolutional network(GCN) [88] to extract features from mesh. Unfortunately, it seems that the ability of GCN could not be significantly enhanced [89], [90], [91] with the increase of network layers as CNN.

Implicit Function. Implicitly representing a surface means describing it analytically using an equation. Because the neural network can fit any function theoretically and does not suffer from discretization artifacts, its equivalent hyperplane can be regarded as a surface. This topic has attracted extensive attention in the community in recent years, from rendering [92], [93], [94], modeling [95], [96], [97] to animation [61]. Several attempts have been made to implicitly represent articulated object surfaces by some MLPs, which are regarded as a mapping from the 2D atlas to 3D space [97], [98], [99] or an occupancy classifier in 3D space [63], [64], [65], [95], [100], [101], [102], [103]. These approaches could continuously query the target attribute of a spatial point, but most of them are category-specific or even subject-specific, hard to capture surface details, and ineffective for self-intersection [64].

Unwrapping Image. UV map [33], [104], [105], [106] is initially created to flat the surface to easily wrap textures. Some studies [33], [107], [108], [109], [110], [111] utilize UV map to store the vertex positions. Unfortunately, the seams to wrap a surface destroy its continuity under this representation, and the impact of seam layouts on the learning results has not been quantified. In addition, the UV map still relies on the topology constraints of the original mesh. This work

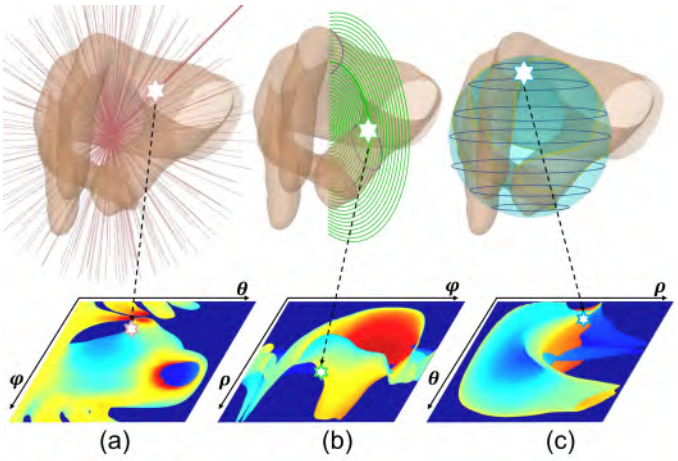


Fig. 2. **SUPPLE projection process.** The three subspaces in SUPPLE are defined based on interaction calculations: (a)The intersection of the surface and rays; (b)The intersection of the surface and longitudes; (c)The intersection of surface and latitudes.

employs another surface unwrapping method based on spherical projections. Some attempts utilized similar techniques for object classification [112], [113] and mesh morphing [114], [115]. Compared with traditional UVs, spherical unwrapping is truly free from the dependence on the input mesh topology. The mapping relationship it establishes from 3D points to 2D plane points is directly determined by the azimuthal angle of the point localizations relative to the center of the sphere. This allows the surface points to be arranged on the 2D image in the order of azimuthal changes.

3 SUPPLE Formulation

Our framework in Sec. 4 will perform 3D skeleton extraction in a projecting space defined by the Spherical Unfolding ProfiLEs (SUPPLE). In this section, we start by introducing the process that projects a continuous 3D surface $\partial\mathbf{\Omega}$ into a 3-channel image space $\mathbf{P}_{rsc}$ in Sec. 3.1. Then the techniques to prepare the input of Sec. 4, *i.e.*an SUPPLE surface $\mathbf{P}_{rsc}(\mathcal{M})$ of a mesh $\mathcal{M}$, is described in Sec. 3.2. And the techniques to prepare the output of Sec. 4, *i.e.*SUPPLE heatmap $\mathbf{H}_{rsc}(\boldsymbol{\jmath})$ of a 3D Gaussian centered at $\boldsymbol{\jmath}$, is described in Sec. 3.3.

3.1 SUPPLE Definition

SUPPLE Composition. As shown in Fig. 2, SUPPLE involves three complementary projections in the spherical coordinate system (ρ, θ, φ). The first channel of SUPPLE , denoted as $\mathbf{P}_r$, projects $\partial\mathbf{\Omega}$ to the spherical subspace θ-φ. The second channel $\mathbf{P}_s$ projects $\partial\mathbf{\Omega}$ to φ-ρ, and the third channel $\mathbf{P}_c$ projects $\partial\mathbf{\Omega}$ to ρ-θ. When $\partial\mathbf{\Omega}$ is bounded in the unit sphere, $\boldsymbol{p}(\rho, \theta, \varphi)$ sampled on the $\partial\mathbf{\Omega}$ has the following range: the length of the ray segment from the origin to the point $\rho \in [0, 1]$; The angle between the positive z-axis and the above-mentioned ray $\theta \in [0, \pi]$; The angle between the positive x-axis and the above-mentioned ray $\varphi \in [0, 2\pi]$. Because atan2 ranges from $-\pi$ to π, 2π are added to the negative φ results. If not explicitly stated, all subsequent references to 3D coordinates are to spherical coordinates.

Ray Profile. Each value on $\mathbf{P}_r$ records the intersections between $\partial\mathbf{\Omega}$ and all points with identical (θ, φ):

$$\mathbf{P}_r(\partial\mathbf{\Omega})[\frac{\theta}{\pi}W_a, \frac{\varphi}{2\pi}H_a] \triangleq \arg\max_{\rho} \left\{\boldsymbol{p}|\boldsymbol{p} \in (\overrightarrow{OR} \cap \partial\mathbf{\Omega})\right\} \quad (1)$$

For each ray $\overrightarrow{OR} = (\forall\rho, \theta, \varphi)$, the ρ value of the outermost intersected point on $\partial\mathbf{\Omega}$ is reserved. If the intersections between the surface and the ray occur, the farthest intersections are recorded; Otherwise, the value is set to zero.

Longitude Profile. Each value on $\mathbf{P}_s$ records the intersections between $\partial\mathbf{\Omega}$ and all points with identical (φ, ρ):

$$\mathbf{P}_s(\partial\mathbf{\Omega})[\frac{\varphi}{2\pi}W_a, \rho H_a] \triangleq \frac{1}{\pi}\underset{\|\theta-0.5\pi\|}{\arg\min} \left\{\boldsymbol{p}|\boldsymbol{p} \in (\widehat{A} \cap \partial\mathbf{\Omega})\right\} \quad (2)$$

For each longitude (semicircle) $\widehat{A} = (\rho, \forall\theta, \varphi)$, the θ value of the intersected point closest to the XY plane is reserved.

Latitude Profile. Each value on $\mathbf{P}_c$ record the intersections between $\partial\mathbf{\Omega}$ and all points with identical (ρ, θ):

$$\mathbf{P}_c(\partial\mathbf{\Omega})[\rho W_a, \frac{\theta}{\pi}H_a] \triangleq \frac{1}{2\pi}\underset{\|\varphi-\pi\|}{\arg\min} \{\boldsymbol{p}|\boldsymbol{p} \in (\odot_C \cap \partial\mathbf{\Omega})\} \quad (3)$$

For each latitude (circle) $\odot_C = (\rho, \theta, \forall\varphi)$, the normalized φ value of the point closest to the $\varphi = \pi$ half-plane on $\partial\mathbf{\Omega}$ is reserved. Finally, the three profiles are concatenated together with size $(3, H_a, W_a)$:

$$\mathbf{P}_{rsc}(\partial\mathbf{\Omega}) \triangleq \mathbf{P}_r(\partial\mathbf{\Omega}) \otimes \mathbf{P}_s(\partial\mathbf{\Omega}) \otimes \mathbf{P}_c(\partial\mathbf{\Omega}) \quad (4)$$

As a result, $\mathbf{P}_{rsc}(\partial\mathbf{\Omega})$ realizes the attributes possessed by voxel grid with the spatial complexity $\mathcal{O}(n^2)$ for $\forall\partial\mathbf{\Omega}$: (i) Each value records an intersection from a parametric curve at a fixed spatial location. (ii) Adjacent values in one profile record intersections from adjacent parametric curves.

Overlap Minimization. An alignment performed on $\partial\mathbf{\Omega}$ could further reduce the overlapping ratio of the projections, allowing SUPPLE to record more surface features. To reduce the overlap in the ρ dimension, the center of $\partial\mathbf{\Omega}$ is normalized to the origin. For the other two dimensions, minimizing overlap is regarded as maximizing the variance (S) of the points $\mathbf{V}$ uniformly sampled from the surface: $\arg\max_{\mathbf{R}} \; S[\Pi_{rsc}(\mathbf{RV})]$. Where $\mathbf{R}$ is the rotation matrix that aligns $\mathbf{V}$ to specific principal axes. Π_{rsc} is a composite function that project the 3D point into the three SUPPLE profiles. Interestingly, when optimizing different categories of the articulated object surface, $\mathbf{R}$ always align their principal axes close to:

$$\mathbf{\Lambda} = \left[\begin{array}{c:c:c} \boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 & \boldsymbol{\lambda}_3 \end{array}\right] \triangleq \left[\begin{array}{c:c:c} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{1+\sqrt{3}}{2\sqrt{3}} & \frac{1-\sqrt{3}}{2\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{1-\sqrt{3}}{2\sqrt{3}} & \frac{1+\sqrt{3}}{2\sqrt{3}} \end{array}\right] \quad (5)$$

This canonical alignment is adopted for the application requiring fewer overlaps.

3.2 Mesh-SUPPLE Conversion

Mesh to SUPPLE. As shown in Fig. 1(a)-(c), a surface represented by mesh $\mathcal{M} = (\mathbf{V}, \mathbf{F})$ is converted to SUPPLE in three steps. First, N surface points sampled uniformly are used to estimate the object center $\boldsymbol{c} \in \mathbb{R}^3$, principal axes

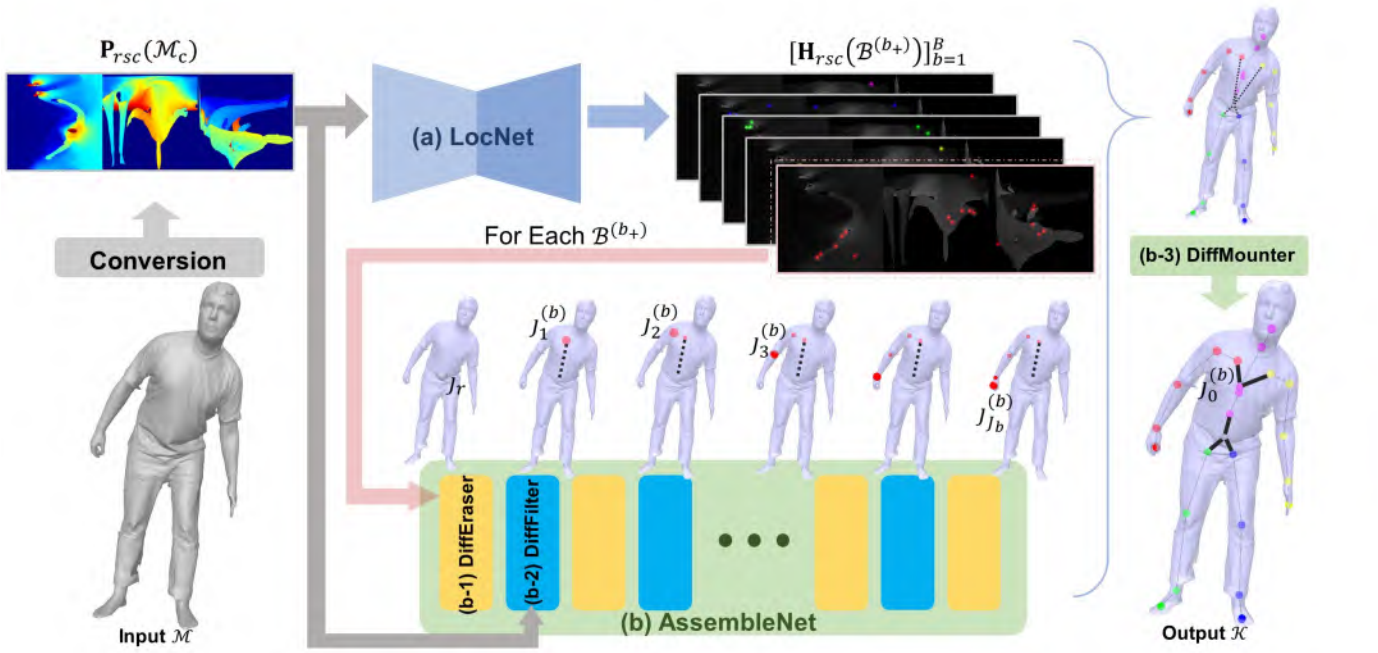


Fig. 3. **Skeleton extraction with SUPPLE .** An input scan is first canonicalized and converted to a SUPPLE surface image (Sec. 3.2). Then, (a) branch-root localization (Sec. 4.1) and (b) in-branch assembly(Sec. 4.2) are performed in SUPPLE subspaces. Finally, the reconstructed skeleton is converted back to 3D space. For better visualization, each $\mathbf{H}_{rsc}(\mathcal{B}^{(b_+)})$ channel takes the corresponding $\mathbf{P}_{rsc}(\mathcal{M})$ channel as the gray background, and the 2D Gaussian kernels are painted in different colors according to the branch index b.

$\mathbf{A} \in \mathcal{SO}(3)$ and maximum radius s from the center. They canonicalize $\mathcal{M}$ by $\mathbf{V}_c = \frac{1}{(1+\varepsilon)s}\mathbf{\Lambda}\mathbf{A}^T(\mathbf{V}-\boldsymbol{c})$. Where $\varepsilon = 0.1$ is used to ensure the whole surface is squeezed into the unit sphere. $\mathbf{\Lambda}$ is defined in Eqn. 5. After that, the intersections of the canonical mesh $\mathcal{M}_c = (\mathbf{V}_c, \mathbf{F})$ with rays, semicircles and spheres [116] are accelerated by building a BVH [117] of $\mathcal{M}$. We further boost the calculations of $\mathbf{P}_s(\mathcal{M})$ according to the fact that the parametric longitudes with the same ϕ can be obtained at one time by calculating all intersections $\mathcal{D}_\theta$ between $\mathcal{M}$ and the half-plane these longitudes are located. Similarly, the calculations of $\mathbf{P}_c(\mathcal{M})$ are boosted according to the fact that the parametric latitudes with the same ρ can be obtained at one time by calculating intersections $\mathcal{D}_\rho$ between $\mathcal{M}$ and the sphere these latitudes are located. (See **Sup. Mat** Sec. B for the details about these two algorithms.) While larger image sizes lead to more detailed surface records, they also increase time costs.

SUPPLE to Mesh. The inverse process is shown in Fig. 1(c)-(e). In a SUPPLE image, each non-zero pixel corresponds to a point on the surface. Therefore, a surface mesh with details can be obtained by querying those pixel coordinates and transforming them into Cartesian space. Since these points are dense enough, the marching cube algorithm [77] can be introduced to reconstruct a plausible mesh topology.

3.3 Joint SUPPLE Heatmap

Ground-truth preparation. The network described later will localize joints in the three subspaces of SUPPLE (θ-φ, φ-ρ and ρ-θ) through a heatmap regression paradigm [1], [14]. Projecting a 3D Gaussian $\mathcal{N}(\boldsymbol{\jmath}, \sigma)$ centered at $\boldsymbol{\jmath}(\rho_\mathrm{j}, \theta_\mathrm{j}, \varphi_\mathrm{j})$ into 2D subspace can be realized by removing the irrelevant dimension. To further convert this projection into image space with size $H_b = W_b$, the heatmaps in the three image spaces are uniformly defined as:

$$\mathbf{H}(\boldsymbol{\jmath})[u, v] = \exp\left(-\frac{(u-u_\mathrm{j})^2 + (v-v_\mathrm{j})^2}{2\sigma^2}\right) \tag{6}$$

where $[u_\mathrm{j}, v_\mathrm{j}]$ refers to $[\frac{\theta_\mathrm{j}}{\pi}W_b, \frac{\varphi_\mathrm{j}}{2\pi}H_b]$ in $\mathbf{H}_r$, $[\frac{\varphi_\mathrm{j}}{2\pi}W_b, \rho_k H_b]$ in $\mathbf{H}_s$, and $[\rho_k W_b, \frac{\theta_\mathrm{j}}{2\pi}H_b]$ in $\mathbf{H}_c$. We further define a multi-joint heatmap $\{\boldsymbol{\jmath}\}_{j=1}^{J}$ as a Gaussian mixture model [118]:

$$\mathbf{H}(\{\boldsymbol{\jmath}\}_{j=1}^{J})[u, v] \triangleq \max_{j \in J}\{\mathbf{H}(\boldsymbol{\jmath})[u, v]\} \tag{7}$$

we use the maximum instead of the sum to ensure that the pixel does not exceed the value range. Finally, a joint SUPPLE heatmap is also defined as a three-channel image $\mathbf{H}_{rsc} \triangleq \mathbf{H}_r \otimes \mathbf{H}_s \otimes \mathbf{H}_c$.

Redundant parsing. This algorithm is used to parse the corresponding joint 3D coordinates $(\rho_\mathrm{j}, \theta_\mathrm{j}, \varphi_\mathrm{j})$ according to the 3-channel projections of its Gaussian distribution $\mathbf{H}_{rsc}(\boldsymbol{\jmath})$:

$$\begin{cases} (\frac{W_b}{\pi}\theta'_\mathrm{j}, \frac{H_b}{2\pi}\varphi'_\mathrm{j}) = \arg\max\limits_{[u,v]} \ \mathbf{H}_r(\boldsymbol{\jmath}) \\ (\frac{W_b}{2\pi}\varphi''_\mathrm{j}, H_b\rho'_\mathrm{j}) = \arg\max\limits_{[u,v]} \ \mathbf{H}_s(\boldsymbol{\jmath}) \\ (W_b\rho''_\mathrm{j}, \frac{H_b}{\pi}\theta''_\mathrm{j}) = \arg\max\limits_{[u,v]} \ \mathbf{H}_c(\boldsymbol{\jmath}) \end{cases} \tag{8}$$

Ideally, $\rho'_\mathrm{j} = \rho''_\mathrm{j} = \rho_\mathrm{j}, \theta'_\mathrm{j} = \theta''_\mathrm{j} = \theta_\mathrm{j}, \varphi'_\mathrm{j} = \varphi''_\mathrm{j} = \varphi_\mathrm{j}$. Considering that the three channel of $\mathbf{H}_{rsc}(\boldsymbol{\jmath})$ predicted from a neural network may be inconsistent, the coordinate with a larger maximum in $\mathbf{H}(\boldsymbol{\jmath})$ is selected as the parsing result for each dimension.

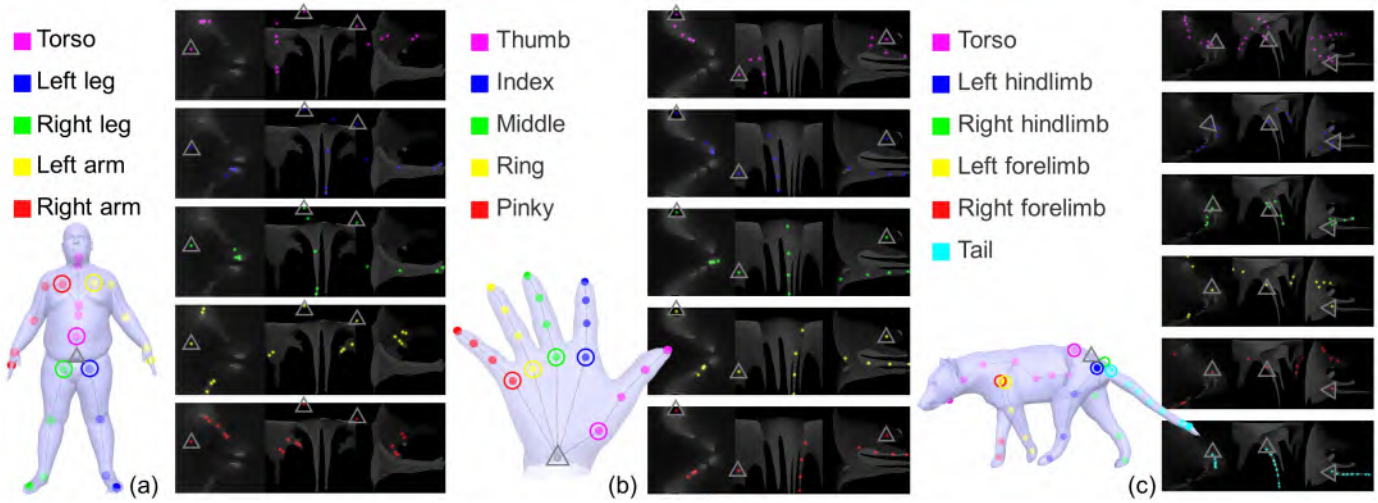


Fig. 4. **Skeleton branch partitions of** (a) Human body, (b) Human hand, and (c) Quadruped. In each example, the root $\mathcal{J}_r$ is gray and emphasized by $\triangle$. Joints in the same color are from the same non-overlapping branch elements $\mathcal{B}^{(b)}$. $\mathcal{J}_1^{(b)}$ (not junction $\mathcal{J}_0^{(b)}$) in each $\mathcal{B}^{(b)}$ is emphasized by $\odot$. For better visualization, each $\mathbf{H}_{rsc}(\mathcal{B}^{(b+)})$ channel takes the corresponding $\mathbf{P}_{rsc}(\mathcal{M})$ channel as the gray background, and the 2D Gaussian kernels are painted in different colors according to the branch index b.

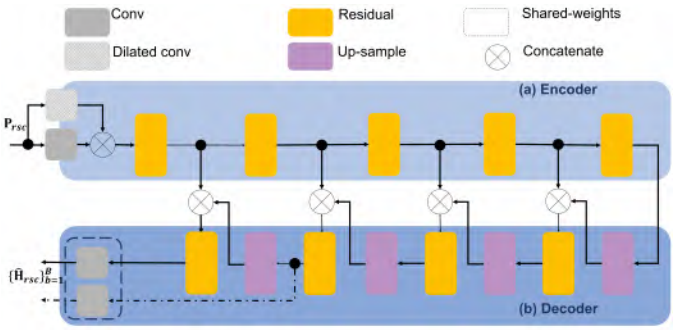


Fig. 5. **LocNet architecture**. The dashed arrow means that the process is only used during training. For a skeleton partitioned with B branches, the input channel number is 3 and the output channel number is $B \times 3$.

4 Skeleton Extraction

A skeletal hierarchy can always be decomposed as multiple branches $\mathcal{K} = \{(\mathcal{J}_0^{(b)}, \mathcal{B}^{(b)})\}_{b=1}^{B}$. Each branch starts from its proximal junction $\mathcal{J}_0^{(b)}$ to its distal leaf $\mathcal{J}_{J_b}^{(b)}$. When $\mathcal{J}_0^{(b)}$ is excluded, other joints within a branch form an ordered sequence $\mathcal{B}^{(b)} = [\mathcal{J}_1^{(b)}, ..., \mathcal{J}_{J_b}^{(b)}]$ containing implicit edge definition. Different branches contain no-overlapping joints, *i.e.*$\mathcal{B}^{(b)} \cap \mathcal{B}^{(b')} = \varnothing$ for $b \neq b'$. One special junction is the skeletal root $\mathcal{J}_r \in \{\mathcal{J}_0^{(b)}\}_{b=1}^{B}$, which records the transformation of the entire skeleton in animation. We denote the joint sequence in a branch with the root as $\mathcal{B}^{(b+)} \triangleq [\mathcal{J}_r] + \mathcal{B}^{(b)}$. An overview of our framework is shown in Fig. 3. First, the LocNet (Sec. 4.1) predicts the branch-root SUPPLE heatmaps $\mathbf{H}_{rsc}(\mathcal{B}^{(b+)})$ from $\mathbf{P}_{rsc}(\mathcal{M}_c)$. Then the AssembleNet (Sec. 4.2) recurrently picks the single joint SUPPLE heatmap $\mathbf{H}_{rsc}(\mathcal{J}_j^{(b)}), j > 0$ according to the proximal-to-distal order. After that, the junction heatmap of each branch $\mathbf{H}_{rsc}(\mathcal{J}_0^{(b)})$ is re-localized. All the positions of skeletal joints are parsed to 3D space according to Eqn. 8. In the following, the hat superscripts $\hat{\mathbf{H}}$ represent the variables regressed from the network, and the star superscripts $\mathbf{H}^\star$ represent the ground truth.

4.1 Branch-Root Localization

Parallel strategy. Because of the significance of $\mathcal{J}_r$, it is assumed as the shared junction of all branches at the beginning. LocNet predicts the branch-root SUPPLE heatmaps from a given SUPPLE surface. For a skeleton containing B branches, LocNet can be formulated as:

$$[\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b+)})]_{b=1}^{B} = F_l[\mathbf{P}_{rsc}(\mathcal{M}_c)] \tag{9}$$

During the training, its ground truth is prepared by projecting the related joint annotations according to Eqn. 7. The mean squared error (MSE) is adopted to supervise their difference:

$$L_l = \sum_{b=1}^{B} \|\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b+)}) - \mathbf{H}^\star_{rsc}(\mathcal{B}^{(b+)})\|_F \tag{10}$$

The inference phase only requires LocNet to regress valid distributions rather than accurate values, and the following steps will parse the correct joint coordinates through Eqn. 8.
Parsing. Because the root SUPPLE heatmap $\mathbf{H}_{rsc}(\mathcal{J}_r)$ appears in all $\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b+)})$, we sum the B branch maps to parse $\hat{\jmath}_r$ $(\hat{\rho}_r, \hat{\theta}_r, \hat{\varphi}_r)$ according to Eqn. 8. The joint number of each branch is estimated by the maximum peak number among the three channels: $\hat{J}_b = \max(n_p[\hat{\mathbf{H}}_r(\mathcal{B}^{(b+)})], n_p[\hat{\mathbf{H}}_s(\mathcal{B}^{(b+)})], n_p[\hat{\mathbf{H}}_c(\mathcal{B}^{(b+)}) - 1$.
Architecture. The detailed structure of LocNet is shown in Fig. 5. It keeps an encoder-decoder architecture. It first uses two parallel convolutions to extract features from the input profiles. The two features are then concatenated and fed to the encoder module with 5 residual blocks that progressively encode profile features with the gradual enlargement of receptive fields. The decoder consists of 4 stacked residual blocks and up-sampling. Each block takes a smaller feature produced by the encoder as input. Except for the final output, leaky-ReLU [119] is adopted for activation. All the padding modes are circular to compensate for the connectivity of profile edges. The penultimate features are also output and supervised by MSE to boost the training convergence.

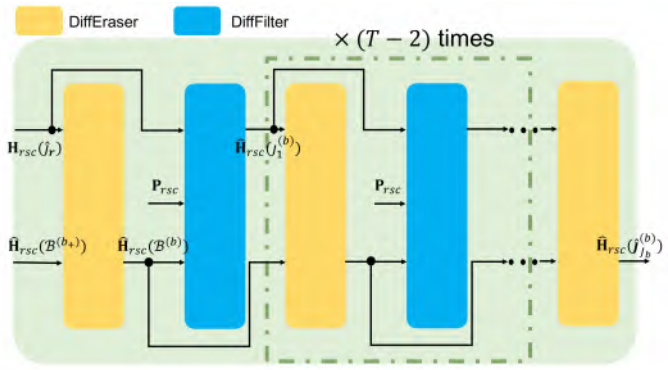


Fig. 6. **Network architecture for AssembleNet** . DiffEraser and DiffFilter forward alternately and recurrently to predict the single joint heatmap according to the proximal-to-distal order within each branch.

4.2 In-branch Assembly

Recurrent strategy. AssembleNet is designed to identify the in-branch joints in the proximal-to-distal order by taking the branch-root SUPPLE heatmap, the root coordinate, and the SUPPLE surface as the inputs:

$$\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b+)}), \mathbf{H}_{rsc}(\hat{\boldsymbol{\jmath}}_r), \mathbf{P}_{rsc}(\mathcal{M}_c) \rightarrow [\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_j^{(b)})]_{j=1}^{J_b} \quad (11)$$

As shown in Fig. 4, the in-branch joints are always distributed as a chain in each SUPPLE subspace. Based on this feature, Eqn. 11 is modeled as a Bayesian inference process. Specifically, the following two processes are conducted alternately and recurrently:

$$\begin{aligned}
\mathbf{H}_{rsc}\hat{\boldsymbol{\jmath}}_r), \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b+)}) &\xrightarrow{(1)} \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)}) \\
\mathbf{H}_{rsc}(\hat{\boldsymbol{\jmath}}_r), \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)}) &\xrightarrow{(2)} \hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_1^{(b)}) \\
\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_1^{(b)}), \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)}) &\xrightarrow{(1)} \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)} - \boldsymbol{\jmath}_1^{(b)}) \\
\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_1^{(b)}), \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)} - \boldsymbol{\jmath}_1^{(b)}) &\xrightarrow{(2)} \hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_2^{(b)}) \\
&\cdots \\
\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_{\hat{J}_b-2}^{(b)}), \hat{\mathbf{H}}_{rsc}([\boldsymbol{\jmath}_j^{(b)}]_{j=\hat{J}_b-2}^{\hat{J}_b}) &\xrightarrow{(1)} \hat{\mathbf{H}}_{rsc}([\boldsymbol{\jmath}_j^{(b)}]_{j=\hat{J}_b-1}^{\hat{J}_b}) \\
\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_{\hat{J}_b-2}^{(b)}), \hat{\mathbf{H}}_{rsc}([\boldsymbol{\jmath}_j^{(b)}]_{j=\hat{J}_b-1}^{\hat{J}_b}) &\xrightarrow{(2)} \hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_{\hat{J}_b-1}^{(b)}) \\
\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_{\hat{J}_b-1}^{(b)}), \hat{\mathbf{H}}_{rsc}([\boldsymbol{\jmath}_j^{(b)}]_{j=\hat{J}_b-1}^{\hat{J}_b}) &\xrightarrow{(1)} \hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_{\hat{J}_b}^{(b)})
\end{aligned} \quad (12)$$

The two processes are realized by CNN modules: the first one is called DiffEraser and the second one is called DiffFilter . For a branch with $\hat{J}_b$ joints, DiffEraser executes $\hat{J}_b$ times and DiffFilter executes $(\hat{J}_b - 1)$ times. Since the inferences among different branches are independent, they are placed parallelly in the same inference batch. Consequently, the parallel iteration amount is determined by the largest $\hat{J}_b$ among branches $T = \max\{\hat{J}_b\}_{b=1}^{B}$. For the branches with $\hat{J}_b \leq T$, both modules are required to output an all-zero result when the input becomes all-zero. Therefore, it is more efficient to partition the whole skeleton into branches with a similar joint number. We follow this principle to partition the human body, human hand, and quadruped skeleton shown in Fig. 4.

DiffEraser erases the single joint SUPPLE heatmap from the multi-joint SUPPLE heatmap in each iteration:

$$\begin{aligned}
&\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)} - [\boldsymbol{\jmath}_j^{(b)}]_{j=1}^{k}) = \\
&F_e[\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_k^{(b)}), \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)} - [\boldsymbol{\jmath}_j^{(b)}]_{j=1}^{k-1})]
\end{aligned} \quad (13)$$

where $\hat{\boldsymbol{\jmath}}_k^{(b)}$ is always the ancestor of remaining projected joints. According to our experiment, even if there exists no parent-child relationship, DiffEraser can still erase the corresponding single-joint heatmap from the given multi-joint heatmap. As a result, we use a random number of points sampled in the unit sphere for its training, and this universal module does not require retraining or fine-tuning on different object categories.

DiffFilter picks the successor SUPPLE heatmap $\hat{\boldsymbol{\jmath}}_{(i+1)}^{(b)}$ based on the parent joint SUPPLE heatmaps $\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_k^{(b)})$ and the remaining multi-joint distribution. Because the in-branch joint relationships can be affected by the surface, $\mathbf{P}_{rsc}(\mathcal{M}_c)$ is also a necessary input:

$$\begin{aligned}
&\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_{j+1}^{(b)}) = \\
&F_f\left(\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_j^{(b)}), \hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b)} - [\boldsymbol{\jmath}_j^{(b)}]_{j=1}^{k}) | \mathbf{P}_{rsc}(\mathcal{M}_c)\right)
\end{aligned} \quad (14)$$

The output is a single joint SUPPLE heatmap, and we use the algorithm in Eqn. 8 to parse its corresponding position. The skeletal edge within $\mathcal{B}^{(b)}$ is precisely the iterative order.

DiffMounter predicts the junction $\boldsymbol{\jmath}_0^{(b)}$ of each branch. Before this step, all $\mathcal{B}^{(b)}$ are assumed to be directly mounted on the root $\boldsymbol{\jmath}_r$. Because the actual junction may be in any other branch $d \neq b$, it is designed as follows:

$$\begin{aligned}
&\hat{\mathbf{H}}_{rsc}(\boldsymbol{\jmath}_0^{(b)}) \\
&= F_m(\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(b+)}) + \mathbf{P}_{rsc}(\mathcal{M}_c), \ [\hat{\mathbf{H}}_{rsc}(\mathcal{B}^{(d+)})]_{d\neq b}^{B})
\end{aligned} \quad (15)$$

where $\mathbf{P}_{rsc}(\mathcal{M}_c)$ is added on the selected branch b's SUPPLE heatmap, playing the role of position embedding [120], [121]. After that, the coordinate of the junction $\boldsymbol{\jmath}_0^{(b)}$ is parsed according to Eqn. 8. The positions of all junctions $[\boldsymbol{\jmath}_0^{(b)}]_{b=1}^{B}$ are parsed in parallel. Due to $\boldsymbol{\jmath}_0^{(b)}$ always duplicating, it is merged into the joint with the closest L2 distance in the existing joint set. In practice, the junction with a smaller merging distance is operated earlier.

4.3 Implementation Details

All AssembleNet modules $F_e(\cdot), F_f(\cdot), F_m(\cdot)$ are fully convolutional. They have the same structure with LocNet $F_l(\cdot)$ except for the number of input and output channels. With a heatmap regression paradigm [1], [14], their weights are optimized by MSE between the prediction and the ground truth (similar to Eqn. 10). Among them, $F_e(\cdot)$ is trained with random points sampled in the unit sphere. The others are trained for different categories. We use Adam optimizer [122] to train them on a single NVIDIA GeForce RTX 3090 GPU at a base learning rate of 1e-5 and a batch size of 64, respectively. 2D average pooling is adopted when the concatenation or addition between heatmaps with different sizes. The SUPPLE surfaces are projected into the image space with $H_a = W_a = 256$, and the joint SUPPLE heatmaps are projected into the image space with $H_b = W_b = 128$. The variance of Gaussian is set to $\sigma = 2$ in all heatmaps.

TABLE 1
Comparisons on the human body. The naked body testing data is from FAUST [33] and SHREC [123]. And the clothed body testing data is from CAPE [87] and BUFF [124]. All methods use the same data partition for training and testing.

Categories	Naked Body		Clothed Body	
Method	*CDJ*↓	*IoU*(%)↑	*CDJ*↓	*IoU*(%)↑
Pinocchio [4]	5.47	40.7	4.61	53.2
MV-CNN [125]	8.01	42.8	6.63	51.2
HandPointNet [16]	7.49	39.4	7.19	49.1
SkelVolNet [56]	5.36	43.9	4.55	56.3
RigNet [5]	5.31	63.7	4.49	67.2
SUPPLE-3DV21 [27]	4.76	65.3	4.51	69.2
Ours	**4.52**	**83.6**	**4.47**	**85.1**

TABLE 2
Comparisons on the human hand and the quadruped. The hand testing data is from DHM [86], Hand3D [46] and Panoptic [126], and the quadruped testing data is from real SMAL [87]. All methods use the same data partition for training and testing.

Categories	Hand		Quadruped	
Method	*CDJ*↓	*IoU*(%)↑	*CDJ*↓	*IoU*(%)↑
Pinocchio [4]	6.89	64.9	4.96	67.3
MV-CNN [125]	7.49	65.1	6.67	66.9
HandPointNet [16]	7.32	62.5	7.38	56.6
SkelVolNet [56]	5.86	71.2	4.89	74.6
RigNet [5]	5.46	78.9	4.83	77.1
SUPPLE-3DV21 [27]	4.85	81.6	4.89	76.3
Ours	**4.67**	**83.3**	**4.63**	**85.3**

5 EXPERIMENTS

In this section, the baselines to perform skeleton extraction are first enumerated in Sec. 5.1. Then the datasets and augmentation strategies used to train our framework and fine-tune those learning-based baselines are illustrated in Sec. 5.2. After that, the metrics for evaluations are introduced in Sec. 5.3. Different methods are compared in Sec. 5.4, and our key components are ablated in Sec. 5.5.

5.1 Baselines

Pinocchio [4] is an optimization-based method that fits a given mesh by searching a template skeleton according to minimum energy. Its original categories include humanoid and quadruped. We further increase the application scenario of this algorithm by adding a hand template [46].

RigNet [5] is a learning-based method that takes a mesh as well as its voxelized form as a composite input, and predicts joint positions and edge connectivity with three graph neural networks called JointNet, RootNet, and BoneNet. Because adaptive clustering is adopted in joint prediction, there exist adjustable hyperparameters in this method. During the inference time, we use their recommended settings, *i.e.*the bandwidth is set to 0.0429 and the threshold is set to 1×10^{-5}.

SkelVolNet [56] is a learning-based method that takes the SDF, vertex density, and several local geometric features of a voxelized mesh as a composite input, and predicts joint positions and edge connectivity with volumetric convolutional networks. Its pre-processing is performed on all testing data in an offline manner, including curvature calculating and SDF converting.

HandPointNet [16] is a learning-based method that takes object point clouds as input, and predicts the skeletal joints in a pre-defined order with PointNet [80] architecture. According to the parameters in the original paper, point clouds are prepared by uniformly sampling 6000 points from object mesh. Leaf joints, *e.g.*hand fingertips, are refined with 256 nearest neighboring points.

MV-CNN [125] is a learning-based method that takes 2D projections of a 3D object as input, and predicts the skeletal joints in a pre-defined order with CNN. Different from ours, it adopts the three orthogonal projections in the Cartesian coordinate system, *i.e.*x-y, y-z, and z-y. The input image size and output heatmap size are set to the same size as ours.

SUPPLE-3DV21 [27] is our preliminary version that takes SUPPLE surface of a 3D object as input, and predicts SUPPLE heatmaps of all joints in a pre-defined order with a single CNN.

Additional operations. All learning-based methods are fine-tuned with the same training datasets and data augmentations as ours. (i) Pinocchio [4] requires the input mesh to be watertight, and an additional hole-filling algorithm [127] is adopted before this optimization. (ii) HandPointNet [16], MV-CNN [125] and SUPPLE-3DV21 [27] require a pre-defined joint order and fixed joint number for each category. In practice, the 21-joint order of the hand model is set as MPII format [30], the 24-joint order of the human body is consistent with SMPL [85], and the 33-joint order of the quadruped is consistent with SMAL [87]. (iii) Pinocchio [4], RigNet [5] and SkelVolNet [56] normalize the input mesh to axis-aligned unit cube. The mesh fed to voxel-based methods [5], [56] is pre-aligned according to the requirements to different categories, *e.g.*, the human torso with Y+ axis and quadruped torso with Z+ axis. HandPointNet [16] and MV-CNN [125] normalize the object according to the oriented bounding box of input. For a fair comparison, SUPPLE-3DV21 [27] adopts the same canonical alignment introduced in Sec. 3.1 for input normalization.

5.2 Datasets

Data Partition. Our framework is mainly evaluated on three articulated categories, *i.e.*the human hand, human body, and quadruped. Different datasets are collected for the network training of each category. After that, all experiments are conducted on the testing set without duplicates.

• For body, SMPL [85] becomes the most commonly used skinning model in the community. The related models add cloth details [87] or other body parts [30], [32] to the original one. We use the same set of network weights to extract the skeleton from the body with and without cloth. In our training data processing, the skeleton format of those naked body datasets [30], [32], [128], [129], [130] and clothed body datasets [87], [131], [132] are uniformly adjusted to SMPL 24-joint layout. Caesar [133], [134], SHREC [123], PSB [135] and

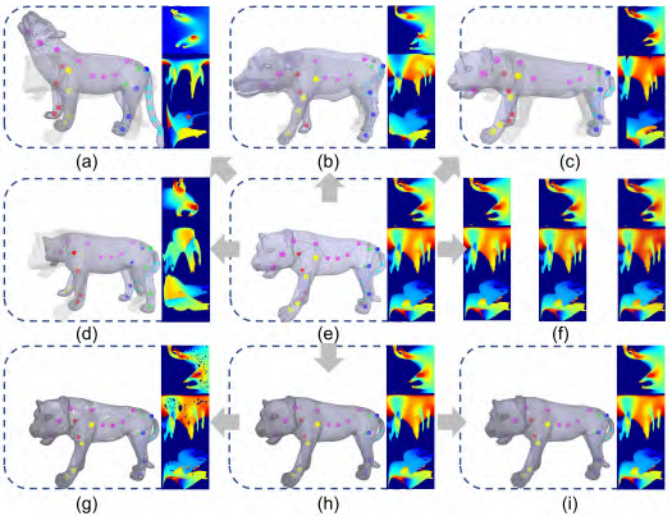


Fig. 7. **Data augmentation strategies**. Taking a quadruped mesh as an example, 8 typical augmentation strategies are adopted in training. (e) The original mesh is modeled by SMAL [31]. (a) Pose prior augmentation; (b) Shape prior augmentation; (c) Bone ratio augmentation; (d) Symmetry augmentation; (f) SUPPLE augmented with noises; (h) Mesh augmented with subdivision; (g) Mesh augmented with holes; (i) Mesh augmented with offsets.

BUFF scans [124] are used for qualitative evaluations. We further performed manual correspondence annotations and SMPL fitting to the 400 real captured meshes in SHREC [123] and BUFF [124]. This part of the data is used for quantitative evaluation together with FAUST [33] and CAPE [87].

• For hand, there are four skinning models [30], [46], [86], [136]. The most popular one is MANO [30]. The model datasets [137], [138], [139] registered to it are used as the majority for hand training. And those scans in [30] are used as a part of the testing set. The datasets [46], [140] registered to VCL hand model [46] are all adopted for quantitative evaluation. As for datasets [126] registered to MeshConv [136], we use the same training-testing partition as [136] provided. The scan model in DHM [86] is used as a part of the data for quantitative evaluation. Since the data contains long wrists, we use 1.2 to 1.5 times the radius of the joint bounding sphere as the reserved part of the original scan models. All the joints number in the training sets are unified to 21. In addition to the above dataset with annotations, we further collected 50 online meshes [141], [142], [143] and also used a handheld scanner to create 150 scanned models from 40 individuals containing diverse hand postures. They are used for qualitative evaluations.

• For quadruped, SMAL [31] is a skinning model that has a skeleton layout with 33 joints. However, the 3D mesh instances in [31], [44], [45] registered to this model are not adopted to train our network training because they contain less than 200 instances in total. Based on SMAL, we use the data augmentation method introduced later to dynamically and randomly generate animal meshes during training. Furthermore, animal data used in Deftransfer [144] and PSB [135] are also used for testing. In addition to the above dataset with annotations, we further collected 60 online meshes [141], [142], [143] and scanned 20 quadruped toys for qualitative evaluations.

Data Augmentation. Because the amount of scanning models in some categories is limited (especially quadrupeds), and most of the data used for training are high-quality and registered mesh models, both the robustness and generalization of our method may be compromised when regarding them naïvely as training sets. Therefore, a series of data augmentation strategies are adopted in the training process for each category of objects. Several augmentations taking quadrupeds as an example are shown in Fig. 7.

• In terms of model instances, multiple variational autoencoders [145], [146] to fit the shape and pose parameter distributions are trained based on the parameterized skinning models [30], [31], [85]. The well-trained decoders are used alone to generate interpolation models of those existing data during training. In addition, for each bone in the model skeleton, additional local bone length ratios are adopted to simulate local differences in each bone. Furthermore, we also consider the symmetry of different categories. The above four augmentations are shown in the transformation from Fig. 7(e) to (a), (b), (c), and (d);

• In terms of surface, loop subdivision [147] is performed on those skinning models with low vertex resolutions. To simulate cracks and holes, a certain proportion of vertices on the mesh are randomly shifted along the normal vector direction or randomly deleted with connected faces. Both operations are performed before and after subdivisions to make a difference in the size of the affected surface area. The above three augmentations are shown in the transformation from Fig. 7(e) to (g), (h), and (i);

• In terms of the image space of SUPPLE surface, salt-and-pepper, Poisson, and Gaussian noise are adopted in combination with each profile to simulate outliers and other distortion in scans. They are illustrated in Fig. 7(f).

5.3 Metrics

CD-joint error (*CDJ*) measures the joint localization accuracy. Compared with the mean per-point position error (*MPJPE*), Chamfer Distance does not require the corresponding number and order between predictions and ground truth. This is suitable for this application because joints generated from some methods [5], [56] are without a specific order, and some datasets [86] provide a non-uniform number of joint annotations. Considering the variations in scale, each instance with predicted skeletons is canonicalized together into the unit sphere before the *CDJ* metric calculation, which is almost identical to the normalization used by Xu *et al.* [5], [56].

Intersection over union (*IoU*) between two point sets should also be considered for those methods that predict an unfixed number of joints. This is because joint localization not only requires the position to be similar to the ground truth but also requires the number to be similar. First, the maximum match pattern between the two sets is obtained by the Hungarian algorithm [148]. Then, the point amount in the intersection of the two sets is defined as the number of points whose Euclidean distance between matching points is less than a given threshold. And the point amount in the union of the two sets equals the total number of points in the two sets minus the point number in the above intersection sets. We follow [5] to set the threshold at each joint as its local SDF value [11] *w.r.t.*the input mesh.

Tree edit distance (*TED*) measures the accuracy of predicted joint connectivity relationships. It is defined as the

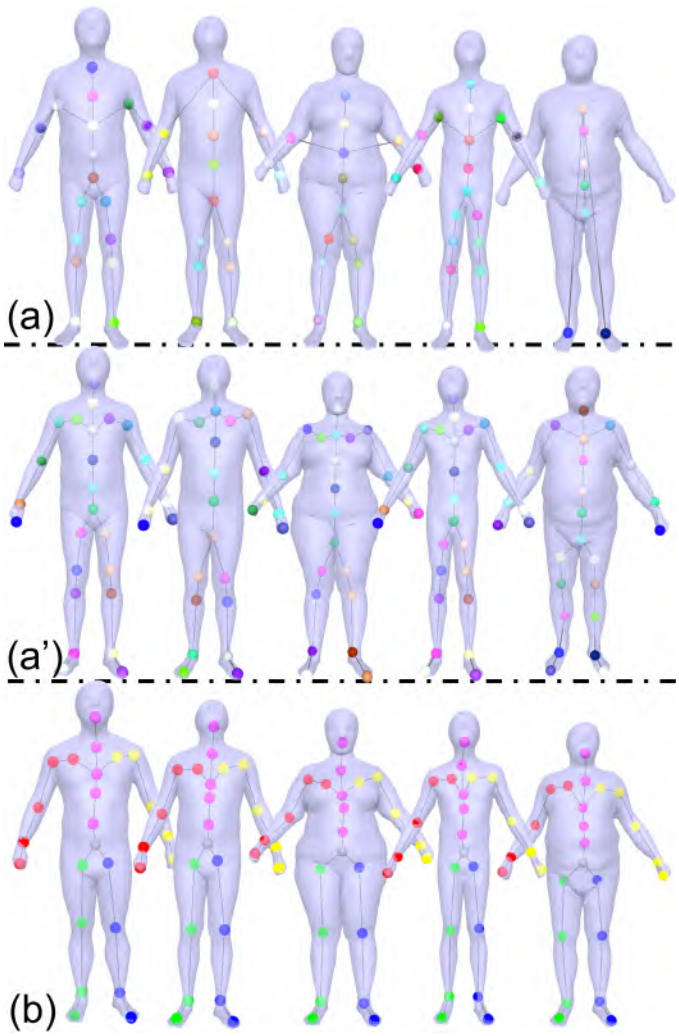


Fig. 8. **Qualitative comparisons on Caesar [134].** With the same CAD models as input, the skeletons extracted by (a) RigNet [5] with the recommended bandwidth. (b) RigNet [5] with the subject-specific bandwidth. (c) Ours.

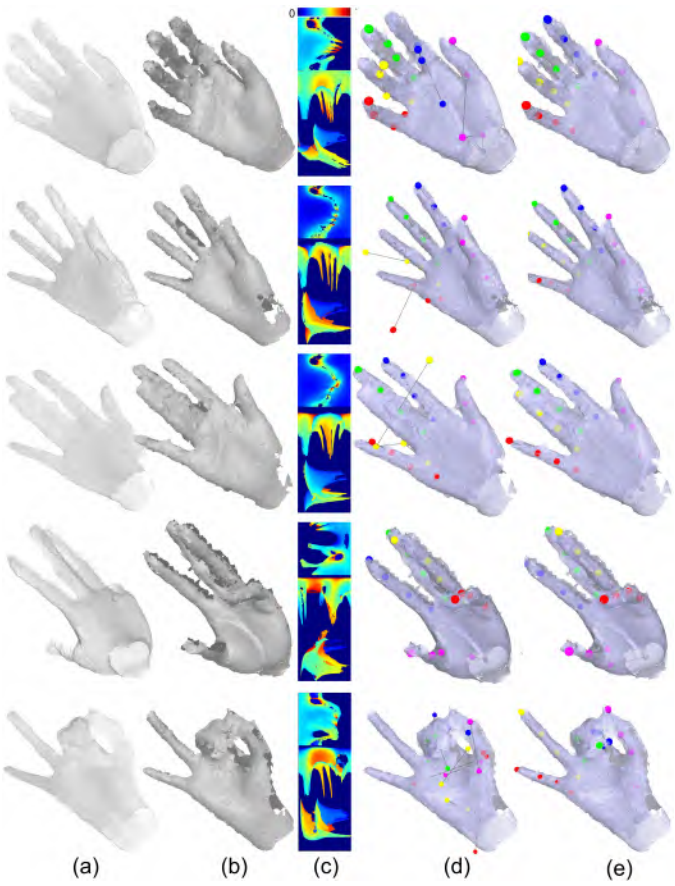


Fig. 9. **Skeleton extraction from point clouds.** The instance in each row shows an extraction from a hand partial point cloud. (a) Original point cloud; (b) Reconstructed surface; (c) Surface SUPPLE ; (d) Predicted skeleton without the data augmentation; (e) Predicted skeleton with the data augmentation.

minimum-cost series of node operations that convert one tree to another. It is proposed in [149] as a distance metric for hierarchical data.

Projection coverage ratio (*PCR*) measures the projection overlap degree of a surface-to-image projection process. It is defined as the coverage degree of the valid points recorded by a projection process to the original object surface points. In practice, we calculate this percentage by following the steps. First, all points recorded in projection images are inversely transformed into normalized 3D space and they form a point cloud $\mathbf{S}_A$. Another point cloud $\mathbf{S}_M$ is sampled by Poisson disk [150] from the original mesh with the same point number as $\mathbf{S}_A$. Then, Chamfer Distance is computed between the $\mathbf{S}_M$ and $\mathbf{S}_A$. For $\boldsymbol{p}_M \in \mathbf{S}_M$, if there exists $\boldsymbol{p}_A \in \mathbf{S}_A$ that satisfies $\|\boldsymbol{p}_A - \boldsymbol{p}_M\|_2 \leqslant 1 \times 10^{-3}$, $\boldsymbol{p}_M$ is considered as covered. Finally, the ratio of covered points to $\mathbf{S}_M$ is adopted as *PCR*.

Inference time is evaluated on the same machine configurations with Intel Core i7-9700K with 8 cores and 8 threads, and a single NVIDIA GeForce RTX 3090 GPU.

5.4 Comparisons

Localization accuracy. Tab. 1 and Tab. 2 extensively report the skeleton extraction performances of all baselines and our method. *CDJ* and *IoU* quantitatively evaluate the skeletal joint localization accuracy on the human body, human hand, and quadruped. The evaluation of the human category is performed separately for naked data and clothed data. The former subset has larger pose diversity because most of the instances are from dynamic MoCap. The latter has a greater variety of body shapes and more surface details as most instances are from static scans. These evaluations reveal the following characteristics: (i) As shown in Tab. 1, our method achieves similar performance to Tab. 1, Pinocchio [4], SkelVolNets [56] and RigNet [5] in the clothed subset, and exceeds them a lot in the naked subset. In addition, our connectivity predictions are generally superior to all of them. (ii) Our current method and preliminary version [27] significantly outperform MV-CNN [125] in all articulated categories. (iii) The method with point clouds [16] as input has a large variation in performance across categories. The overall performance is similar to that of MV-CNN [125].
Inference time. Tab. 3 summarizes the average time cost of each method. For a fair comparison, time records for all methods start from feeding a mesh, contain the representation conversion, and end by assembling a full skeleton. In all categories, our framework not only has the fastest repre-

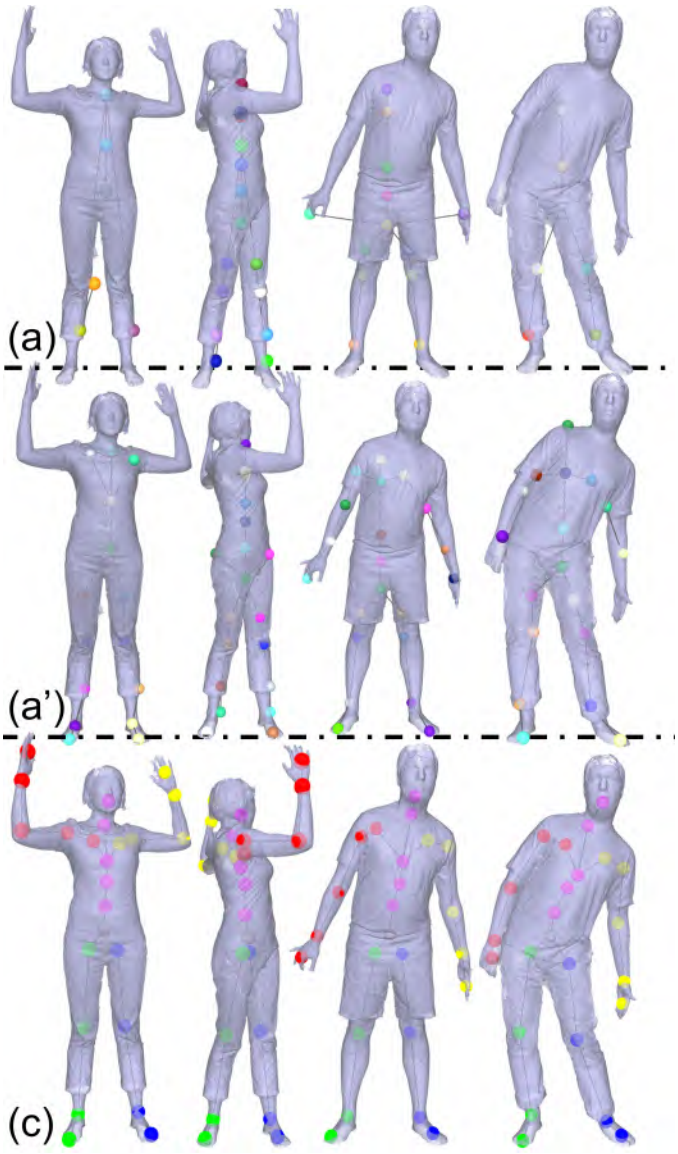


Fig. 10. **Qualitative comparisons on BUFF [124].** With the same scans as input, the skeletons extracted by (a) RigNet [5] with the recommended bandwidth. (b) RigNet [5] with the subject-specific bandwidth. (c) Ours.

sentation conversion speed but also the fastest embedding speed. By contrast, RigNet [5] needs to voxelize the input during its conversion, relies on clustering during embedding, and computes vertex-to-bone geodesic distance, all of which greatly increase its time cost. Pinocchio [4] in our evaluation is faster than its original version, which should be due to our more advanced machine. The embedding times our framework spent on the body and hand categories is relatively close because most of them have the same branch number and the same maximum number of joints within branches.

Extraction robustness. Some qualitative results of our methods on online CAD or scanning meshes for each category are illustrated in Fig. 11 and Fig. 12. See **Sup. Mat** Sec. E for more experimental results. According to these results, our framework is robust to the changes in shape and pose, mesh vertex density, and the integrity of patches on different object categories. In Fig. 11(c1)-(c3) and Fig. 12(c1)-(c3), Our framework adaptively extracts different numbers of

TABLE 3
Average time cost (in seconds) of each method on different categories. All methods are evaluated on the same machine configurations.

Categories	Body		Hand		Quadruped	
Method	Convert	Embed	Convert	Embed	Convert	Embed
Pinocchio [4]	-	6.49	-	5.31	-	5.02
RigNet [5]	9.02	279.84	7.96	274.56	8.44	253.04
Ours	**1.93**	**4.36**	**1.62**	**4.35**	**1.42**	**4.52**

joints depending on the tail length of different quadruped instances. These results have practical implications over the preliminary regression [27] with a fixed joint number.

As shown in Fig. 13, we further evaluate the *IoU* changes under the different thresholds, from $\times 0.1$ to $\times 1.0$ SDF value. A comparison shows that the three methods generally perform better on quadrupeds than on human bodies. This should be due to the fact that there is a greater variety of body poses than quadrupeds in the testing data. Therefore, a more detailed comparison is made with RigNet [5] on the human body datasets, and some results are shown in Fig. 8 and Fig. 10. As shown in Row-1 of both figures, RigNet performs poorly with the recommended hyperparameters. Although it is possible to adjust the bandwidth parameter for each subject to get more appealing results (Row-2), it is sensitive to shape variations and always fails under asymmetric poses. By contrast, our method is more robust among different shapes and has no symmetry pose constraints.

5.5 Ablation Study

Module Contributions. The two key innovations of this work are the SUPPLE representation and the hierarchical heatmap regression strategy. To clarify their respective contributions to the whole pipeline, we design the following alternatives: (i) SUPPLE + end-to-end. This is actually the pipeline of our preliminary version, which uses a single CNN to regress SUPPLE heatmaps of all skeleton joints in the pre-defined order. This pipeline is denoted as "$\mathbf{P}_{rsc}$-$e2e$". (ii) Voxel-grid + hierarchical strategy. This pipeline takes a voxelized mesh as input. Its architectures are 3D CNNs $F_l^3, F_e^3, F_f^3, F_m^3$ with the same block number as Sec. 4 and the same channel number as SkelVolNet [56]. It predicts the 3D joint distribution in an axis-aligned unit cube with 88^3 resolutions. It is denoted as "$\mathbf{V}^3 + \mathcal{B}$". (iii) Multi-view + hierarchical strategy. This pipeline takes the three orthogonal projections used in [125] as input. It uses the same network architecture, parsing algorithm, and map size as our frameworks. It is denoted as "$\mathbf{P}_{xyz} + \mathcal{B}$". (iv) SUPPLE + hierarchical strategy + GCN. This pipeline is the same as our full framework before F_l. Instead of using F_f, F_m, it adopts a GCN to predict the joint connectivity within each branch. It is denoted as "$\mathbf{P}_{xyz} + GCN$". These pipelines are trained with the same training data and augmentation strategies as our full framework. To analyze their localization performances on distal joints, we further introduce distal-weighted *CDJ* ("*CDJ*-d" in Tab. 4). For a branch $\mathcal{B}^{(b)}$ with J_b joints, the distance caused by $\hat{j}_j^{(b)}$ is weighted by j/J_b. In particular, its junction is weighted by $1/J_b$ and the root is weighted by $1/\max[J_b]_{b=1}^{B}$. Tab. 4 reports

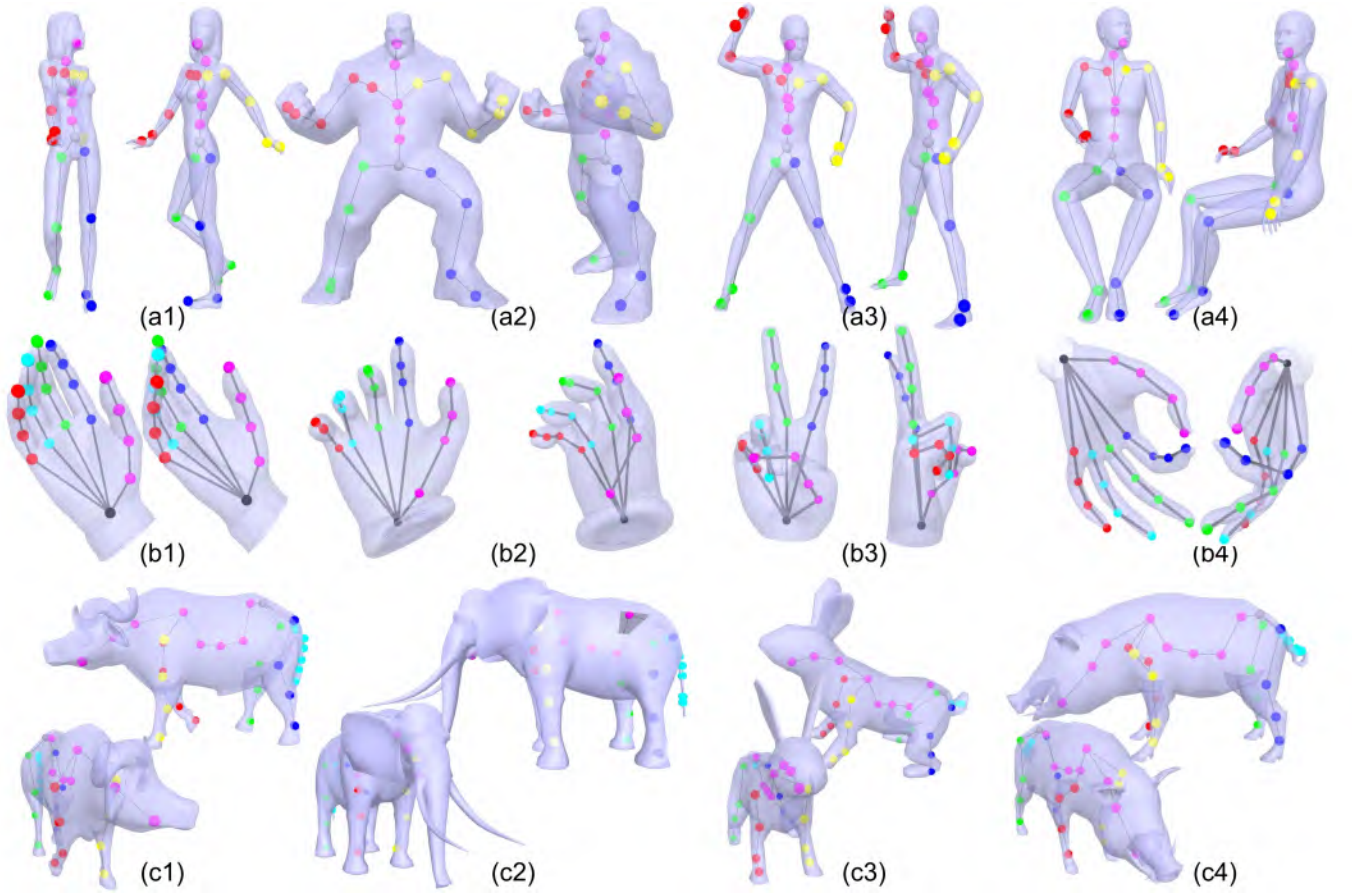


Fig. 11. **Qualitative results from online CAD.** The instances in each row are skeleton extraction results from different categories. Each mesh is viewed from two perspectives. All meshes are collected online.

their performances on the three articulated categories with 5 metrics. First, $\mathbf{P}_{rsc}$-$e2e$ is inferior to $\mathbf{V}^3+\mathcal{B}$ in terms of the whole skeleton metrics, but superior to $\mathbf{V}^3+\mathcal{B}$ in terms of the distal-weighted ones. This reveals that SUPPLE contributes more to modeling the distribution patterns of distal joints, while the hierarchical paradigm improves the joint localization in the whole skeleton scope. In addition, the two variants contain a similar number of parameters, the bloated 3D-CNN structure in $\mathbf{V}^3+\mathcal{B}$ makes it more computationally expensive. Second, both $\mathbf{P}_{rsc}$-$e2e$ and $\mathbf{V}^3+\mathcal{B}$ outperform $\mathbf{P}_{xyz}+\mathcal{B}$ in all metrics. This is probably because too many features that are helpful for joint localization overlapped during the projections happened in the Cartesian system. Nevertheless, $\mathbf{P}_{xyz}+\mathcal{B}$ still outperforms MV-CNN [125], which again demonstrates the effectiveness of our hierarchical paradigm. Compared with our recurrent strategy in Sec. 4.2, $\mathbf{P}_{xyz}+GCN$ is inefficient in connectivity prediction. Those inaccurate connections may also lead to the identity mismatching of joints in different channels, which further weakens its localization and seriously affect its "*CDJ*-d". The proposed pipeline, *i.e.*$\mathbf{P}_{rsc}+\mathcal{B}$, achieve the best performance under different metrics. This proves that the innovations in surface representation and learning paradigms are both valuable to a skeleton extraction task.

SUPPLE Formulation. Some choices on SUPPLE surface conversion and joint SUPPLE heatmap preparation are evaluated. According to Sec. 3.1, we record $\mathbf{P}_r$ with maximum ρ in each ray, $\mathbf{P}_s$ with maximum $|\theta\text{-}0.5\pi|$, and $\mathbf{P}_c$ with maximum $|\varphi\text{-}\pi|$. In ablations reported in Tab. 5 Row1-Row3, the variant ray profile $\mathbf{P}'_r$ is modified to record the point with maximum ρ (innermost point). The variant longitude profile $\mathbf{P}'_s$ is modified to record the average θ value on a single longitude arch. And the variant latitude profile $\mathbf{P}'_c$ is modified to record the average φ value on a single latitude circle. Among them, $\mathbf{P}'_r$ has the greatest impact on the whole method because it violates the original intention of designing SUPPLE to model distal joint patterns. The weakening effects from $\mathbf{P}'_s$ and $\mathbf{P}'_c$ are probably because the average operation used to construct them compromises the local features of the surface while making more features globally relevant. In Sec. 3.3, we prepare the ground-truth of joint SUPPLE heatmaps $\mathbf{H}_{rsc}$ with Gaussian variance $\sigma = 2.0$. In ablations illustrated in Tab. 5 Row4-Row5, the methods with different σ settings are evaluated. Its influence on network convergence, joint localizations, and connectivity predictions during training is shown in Fig. 14. During the training, CNN modules with larger σ converge faster from the heatmap perspective. However, during the inference, because the individual Gaussian distributions in a branch SUPPLE heatmap are too close to each other, it is difficult for DiffFilter to select individual joint features. A smaller σ may lead to a joint SUPPLE heatmap being too sparse, which is not conducive to all the learning modules.

Augment Effectiveness. To explore the improvement of our network brought by data augmentation, we introduced a more challenging task, *i.e.*, extracting skeletons from partial

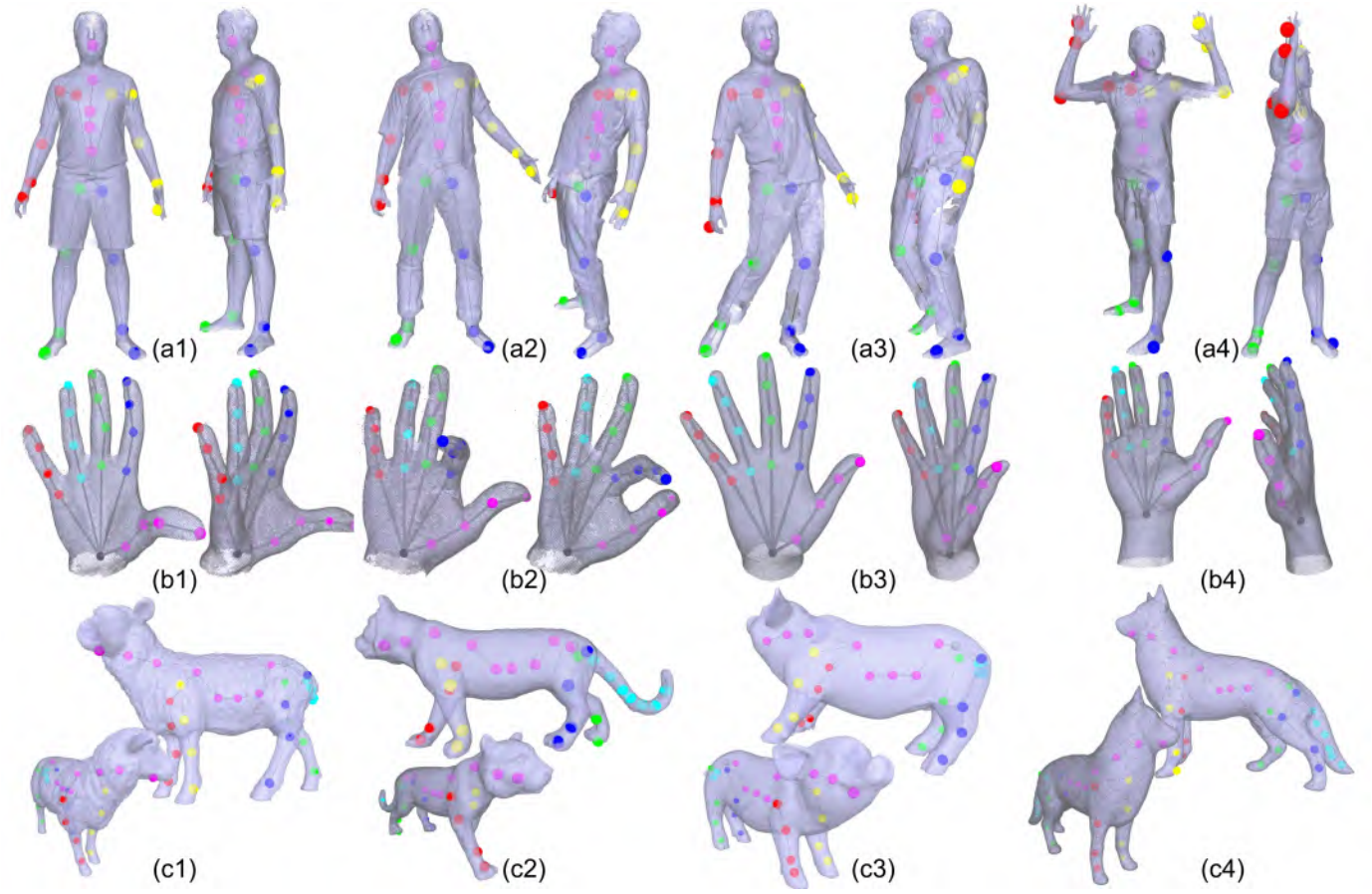


Fig. 12. **Qualitative results from scans.** The instances in each row come from different categories. Each mesh is viewed from two perspectives. All meshes come from the scanning of real objects.

TABLE 4
Ablations on Module Contributions. Each row corresponds to the performance of the variant pipeline. *TED*s of the variant "$\mathbf{P}_{rsc}$-$e2e$" are not reported due to the method's dependence on pre-defined joint orders.

	Categories		Body				Hand				Quadruped			
Variants	Params (M)	FLOPs (G)	*CDJ*↓	*IoU*(%)↑	*CDJ*-d↓	*TED*↓	*CDJ*↓	*IoU*(%)↑	*CDJ*-d↓	*TED*↓	*CDJ*↓	*IoU*(%)↑	*CDJ*-d↓	*TED*↓
$\mathbf{P}_{rsc}$-$e2e$	17.20	10.6	4.71	68.7	3.34	-	4.85	81.6	2.95	-	4.89	76.3	3.41	-
$\mathbf{V}^3 + \mathcal{B}$	17.68	538.12	4.63	79.4	3.42	2.64	4.80	82.1	2.99	2.53	4.72	83.9	3.45	2.47
$\mathbf{P}_{xyz} + \mathcal{B}$	49.24	82.05	6.02	63.9	4.39	3.16	6.11	71.4	3.26	2.97	6.07	72.6	4.21	2.83
$\mathbf{P}_{rsc} + GCN$	30.93	19.65	4.62	74.1	4.17	3.34	5.13	82.0	3.09	3.31	4.82	79.9	4.14	3.28
Full	49.24	82.05	**4.50**	**84.1**	**3.23**	**2.56**	**4.67**	**83.3**	**2.69**	**2.31**	**4.63**	**85.3**	**3.11**	**2.21**

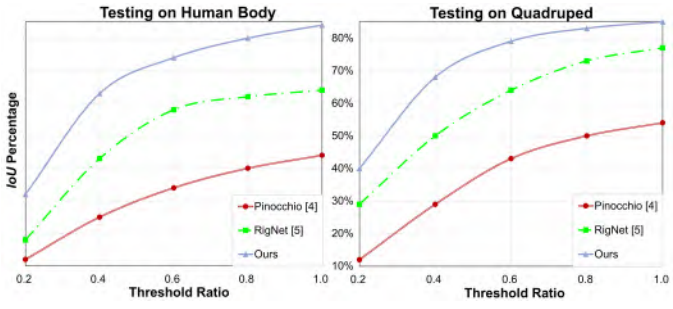


Fig. 13. **Localization evaluations with several *IoU* thresholds.** The line graphs depict the differences in localization performances among the three methods on the human body and quadruped testing data. Their horizontal and vertical axes share the same units and range.

TABLE 5
Ablations on SUPPLE Formulations. Each row corresponds to the performance of the variants on the three articulated categories.

Categories	Body			Hand			Quadruped		
Variants	*CDJ*↓	*IoU*(%)↑	*TED*↓	*CDJ*↓	*IoU*(%)↑	*TED*↓	*CDJ*↓	*IoU*(%)↑	*TED*↓
$+\mathbf{P}'_r$	4.72	71.2	2.76	5.13	72.7	2.67	4.82	71.5	3.16
$+\mathbf{P}'_s$	4.63	74.6	2.61	5.09	75.1	2.51	4.86	76.1	2.39
$+\mathbf{P}'_c$	4.69	75.9	2.63	5.14	74.8	2.48	4.90	76.3	2.29
$\sigma = 1.0$	4.63	82.2	2.60	4.69	81.6	2.38	4.66	82.9	2.24
$\sigma = 4.0$	4.59	83.0	2.62	4.72	82.8	2.43	4.70	83.7	2.35
Full	**4.50**	**84.1**	**2.56**	**4.67**	**83.3**	**2.31**	**4.63**	**85.3**	**2.21**

point clouds. As a preprocessing, the ball pivoting algorithm [151] is adopted to reconstruct the surface of the object according to its point cloud. As shown in Fig. 9(a)-(c), Because the original point clouds are incomplete and full of noise, numerous cracks and holes still exist in these meshes. However, this does not affect the reasoning process of our method. As shown in Fig. 9(d)-(e), after data enhancement,

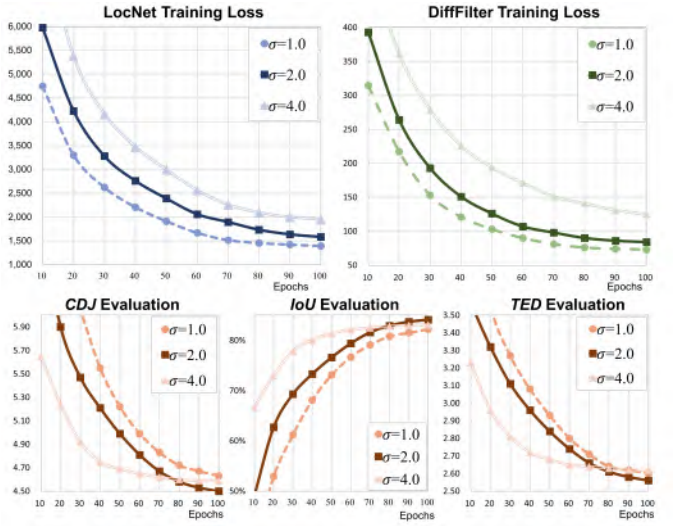


Fig. 14. **Ablations on different σ.** The line graphs depict the differences in training and testing performances among the three variants on the human body data.

TABLE 6
Coverage ratios of different profiles. The percentage indicates the coverage of the inverse transformed points from different types of projections relative to the original surface.

PCR(%)	1st channel	2nd channel	3rd channel	Total
$\mathbf{P}_{xyz}$	31.2	30.8	31.7	73.3
$\mathbf{P}_{rsc}$	42.6	36.8	38.9	84.4
$\mathbf{P}_{rsc}, \mathbf{\Lambda}$	61.2	37.9	39.3	**91.3**

our method can be promoted to this task more robustly without being affected by holes or outliers.

Profile Overlap. In comparison with the multi-view method, we argue that its drawback might be the high rate of projection overlaps and the low coverage of the original mesh. Therefore, we compare several relevant projection methods on their coverage ratio *w.r.t.*to the original surface. 10K mesh instances generated by those articulated skinning models with random pose and shape parameters are used to evaluate the overlap degree of different projection techniques. As illustrated in Tab. 6, the highest coverage of the original mesh surface is achieved by the SUPPLE with the axis alignment described in Eqn. 5. It should be admitted that SUPPLE is a surface descriptor in 3D space, just as the silhouette is an instance descriptor in a 2D image. Since it does not entirely record the surface, it may not be compared to those full-surface recorded representations in basic tasks, *e.g.*classification, and segmentation.

Accelerated Conversion. We compare the acceleration contributions of different techniques in SUPPLE conversion, including BVH and dictionary structure (denoted as $\mathcal{D}_\theta, \mathcal{D}_\rho$, see **Sup. Mat** Sec. B for details). Meshes in four configurations are selected as the tested objects, and we create $5K$ instances for each configuration. The average converting time from them to three profiles is shown in Tab. 7. In the four configurations of meshes, the first two correspond to the order of the number of vertices of skinning models, and the last corresponds to the order of the number of vertices of scanning data or high-precision CAD models. Through comparison, it can be found that the addition of BVH makes the transformation time of each map no longer proportional to the increase in the number of model vertices. The introduction of dictionary structure further increases the efficiency of computing intersection in $\mathbf{P}_s$ and $\mathbf{P}_c$, because the radius distribution of most mesh vertices is relatively concentrated. All three acceleration methods make it possible to dynamically perform data augmentation during our training.

TABLE 7
Time cost from Mesh to SUPPLE . Each row corresponds to the average time (in milliseconds) used in an approach to convert meshes with different vertex numbers.

Time (ms)	$\#V = 0.7K$	$\#V = 7K$	$\#V = 70K$
- BVH	641/672/619	3.1k/3.2k/3.0k	22.3k/21.9k/21.6k
- $(\mathcal{D}_\theta, \mathcal{D}_\rho)$	225/237/267	649/663/652	945/1.2k/1.2k
Full	225/96/45	649/205/100	945/411/298

6 Conclusion

This paper proposes a novel surface-to-image representation, named SUPPLE . The skeleton extraction developed with it avoids constraints on the shape, pose, and topology of the input 3D surface. With SUPPLE , the skeleton extraction is recast to be a series of 2D tasks that can be tackled by CNN architectures. SUPPLE compactly unwraps surface without topology dependency. Compared with other representations, the neural network with SUPPLE can go deeper and extract 3D features more efficiently. Compared with the previous version [27], the learning paradigm from a skeletal branch perspective makes full use of the skeletal hierarchy of articulated objects and further improves the universality and robustness of our method on multiple categories. The proposed method could be useful for finding correspondences and performing non-rigid registration among meshes with different topologies. It is also a brilliant idea to further regress the correspondences and skinning weights from the SUPPLE .

Limitation and Discussion. Although we managed to extend SUPPLE to applications on point clouds and to validate the effectiveness on multiple articulated categories, there are still some limitations in our framework. As an example shown in Fig. 15, although our method can extract the skeleton from the mesh in a variety of poses, the prediction results produce errors at the endpoints for some tight poses. This might be caused by two reasons. First, in a pose similar to fist-clenching, the surface regions corresponding to the joint are completely occluded in $\mathbf{P}_r$, and no complementary information is provided in other profiles. Secondly, in our training data, the mesh data under these tight poses is also lacking. A promising future direction is to design a coarse-to-fine pipeline based on SUPPLE to iteratively extract the skeleton from them.

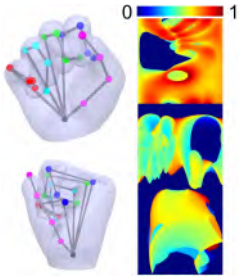


Fig. 15. **Failure cases.** The left side shows the two perspectives of the extracted skeleton, and the right side is the SUPPLE of the original surface.

REFERENCES

[1] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz, "Hand pose estimation via latent 2.5 d heatmap regression," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 118–134. 1, 2, 3, 5, 7

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019. 1

[3] N. Magnenat-Thalmann, R. Laperrire, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," in *In Proceedings on Graphics interface'88*. Citeseer, 1988. 1, 2, 3

[4] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," *ACM Transactions on graphics (TOG)*, vol. 26, no. 3, pp. 72–es, 2007. 1, 2, 3, 8, 10, 11

[5] Z. Xu, Y. Zhou, E. Kalogerakis, C. Landreth, and K. Singh, "Rignet: neural rigging for articulated characters," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 58–1, 2020. 1, 3, 8, 9, 10, 11

[6] D. Antotsiou, G. Garcia-Hernando, and T.-K. Kim, "Task-oriented hand motion retargeting for dexterous manipulation imitation," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0. 1

[7] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 62–1, 2020. 1

[8] Z. Yang, W. Zhu, W. Wu, C. Qian, Q. Zhou, B. Zhou, and C. C. Loy, "Transmomo: Invariance-driven unsupervised video motion retargeting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5306–5315. 1

[9] S. Ni, R. Luo, Y. Zhang, M. Budagavi, A. J. Dickerson, A. Nagar, and X. Guo, "Scan2avatar: Automatic rigging for 3d raw human scans," in *ACM SIGGRAPH 2020 Posters*, 2020, pp. 1–2. 1

[10] S. Schaefer and C. Yuksel, "Example-based skeleton extraction," in *Symposium on Geometry Processing*, 2007, pp. 153–162. 1, 3

[11] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," *The Visual Computer*, vol. 24, no. 4, pp. 249–259, 2008. 1, 9

[12] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, pp. 1–10, 2008. 1, 2

[13] B. H. Le and Z. Deng, "Robust and accurate skeletal rigging from mesh sequences," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–10, 2014. 1, 3

[14] Y. Wang, B. Zhang, and C. Peng, "Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization," *IEEE transactions on image processing*, vol. 29, pp. 2977–2986, 2019. 1, 5, 7

[15] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu, "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*, 2020, pp. 5346–5355. 1, 2, 3

[16] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8417–8426. 1, 8, 10

[17] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3d hand pose estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 475–491. 1

[18] H. Qin, S. Zhang, Q. Liu, L. Chen, and B. Chen, "Pointskelcnn: Deep learning-based 3d human skeleton extraction from point clouds," in *Computer Graphics Forum*, vol. 39, no. 7. Wiley Online Library, 2020, pp. 363–374. 1, 3

[19] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3d hand shape and pose estimation from a single rgb image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 10 833–10 842. 1, 2, 3

[20] H. Choi, G. Moon, and K. M. Lee, "Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose," in *European Conference on Computer Vision*. Springer, 2020, pp. 769–787. 1

[21] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920. 1, 3

[22] G. Moon, J. Y. Chang, and K. M. Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *Proceedings of the IEEE conference on computer vision and pattern Recognition*, 2018, pp. 5079–5088. 1, 3

[23] J. Malik, I. Abdelaziz, A. Elhayek, S. Shimada, S. A. Ali, V. Golyanik, C. Theobalt, and D. Stricker, "Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7113–7122. 1

[24] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang, "Interacting two-hand 3d pose and shape reconstruction from single color image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 354–11 363. 1

[25] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 274–13 283. 1, 2

[26] G. Georgakis, R. Li, S. Karanam, T. Chen, J. Košecká, and Z. Wu, "Hierarchical kinematic human mesh recovery," in *European Conference on Computer Vision*. Springer, 2020, pp. 768–784. 2

[27] Z. Zhao, R. Rao, and Y. Wang, "Supple: Extracting hand skeleton with spherical unwrapping profiles," in *2021 International Conference on 3D Vision*. IEEE, 2021, pp. 899–909. 2, 8, 10, 11, 14

[28] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via laplacian based contraction," in *2010 Shape Modeling International Conference*. IEEE, 2010, pp. 187–197. 2

[29] C. Lin, C. Li, Y. Liu, N. Chen, Y.-K. Choi, and W. Wang, "Point2skeleton: Learning skeletal representations from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4277–4286. 2

[30] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 6, pp. 1–17, 2017. 2, 3, 8, 9

[31] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3d menagerie: Modeling the 3d shape and pose of animals," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6365–6373. 2, 3, 9

[32] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 975–10 985. 2, 8

[33] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3794–3801. 2, 3, 8, 9

[34] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419. 2

[35] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4903–4911. 2, 3

[36] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4866–4874. 3

[37] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan, "A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 793–802. 3

[38] Z. Zhao, K. Zhang, and Y. Wang, "Pp-net: simultaneous pose and shape reconstruction from a single depth map," in *Optoelectronic Imaging and Multimedia Technology VII*, vol. 11550. International Society for Optics and Photonics, 2020, p. 115500N. 3

[39] N. Garau, N. Bisagno, P. Bródka, and N. Conci, "Deca: Deep viewpoint-equivariant human pose estimation using capsule autoencoders," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 677–11 686. 3

[40] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 2018, pp. 7122–7131. 3

[41] A. Boukhayma, R. d. Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 843–10 852. 3

[42] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2252–2261. 3

[43] B. Biggs, O. Boyne, J. Charles, A. Fitzgibbon, and R. Cipolla, "Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop," in *European Conference on Computer Vision*. Springer, 2020, pp. 195–211. 3

[44] S. Zuffi, A. Kanazawa, and M. J. Black, "Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3955–3963. 3, 9

[45] S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. J. Black, "Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild"," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5359–5368. 3, 9

[46] Z. Zhao, T. Wang, S. Xia, and Y. Wang, "Hand-3d-studio: A new multi-view system for 3d hand reconstruction," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2478–2482. 3, 8, 9

[47] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield *et al.*, "Dexycb: A benchmark for capturing hand grasping of objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9044–9053. 3

[48] Y. You, Y. Lou, C. Li, Z. Cheng, L. Li, L. Ma, C. Lu, and W. Wang, "Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 647–13 656. 3

[49] Z. Zhang, L. Hu, X. Deng, and S. Xia, "Weakly supervised adversarial learning for 3d human pose estimation from point clouds," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 5, pp. 1851–1859, 2020. 3

[50] R. Shi, Z. Xue, Y. You, and C. Lu, "Skeleton merger: an unsupervised aligned keypoint detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 43–52. 3

[51] Y. You, W. Liu, Y. Ze, Y.-L. Li, W. Wang, and C. Lu, "Ukpgan: A general self-supervised keypoint detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 042–17 051. 3

[52] C. Zhong, P. You, X. Chen, H. Zhao, F. Sun, G. Zhou, X. Mu, C. Gan, and W. Huang, "Snake: Shape-aware neural 3d keypoint field," *arXiv preprint arXiv:2206.01724*, 2022. 3

[53] Z. Dong, C. Guo, J. Song, X. Chen, A. Geiger, and O. Hilliges, "Pina: Learning a personalized implicit neural avatar from a single rgb-d video sequence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 470–20 480. 3

[54] B. Jiang, Y. Hong, H. Bao, and J. Zhang, "Selfrecon: Self reconstruction your digital avatar from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5605–5615. 3

[55] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, "Icon: Implicit clothed humans obtained from normals," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2022. 3

[56] Z. Xu, Y. Zhou, E. Kalogerakis, and K. Singh, "Predicting animation skeletons for 3d articulated models via volumetric nets," in *2019 International Conference on 3D Vision*. IEEE, 2019, pp. 298–307. 3, 8, 9, 10, 11

[57] E. De Aguiar, C. Theobalt, S. Thrun, and H.-P. Seidel, "Automatic conversion of mesh animations into skeleton-based animations," in *Computer Graphics Forum*, vol. 27, no. 2. Wiley Online Library, 2008, pp. 389–397. 3

[58] N. Hasler, T. Thormählen, B. Rosenhahn, and H.-P. Seidel, "Learning skeletons for shape and pose," in *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, 2010, pp. 23–30. 3

[59] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," *ACM transactions on graphics (TOG)*, vol. 22, no. 3, pp. 587–594, 2003. 3

[60] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," pp. 408–416, 2005. 3

[61] S. Saito, J. Yang, Q. Ma, and M. J. Black, "Scanimate: Weakly supervised learning of skinned clothed avatar networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2886–2897. 3

[62] S. Wang, A. Geiger, and S. Tang, "Locally aware piecewise transformation fields for 3d human mesh registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7639–7648. 3

[63] B. Deng, J. P. Lewis, T. Jeruzalski, G. Pons-Moll, G. Hinton, M. Norouzi, and A. Tagliasacchi, "Nasa neural articulated shape approximation," in *European Conference on Computer Vision*. Springer, 2020, pp. 612–628. 3

[64] M. Mihajlovic, Y. Zhang, M. J. Black, and S. Tang, "Leap: Learning articulated occupancy of people," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 461–10 471. 3

[65] K. Karunratanakul, A. Spurr, Z. Fan, O. Hilliges, and S. Tang, "A skeleton-driven neural occupancy representation for articulated hands," in *2021 International Conference on 3D Vision*. IEEE, 2021, pp. 11–21. 3

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012. 3

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 3

[68] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. 3

[69] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 767–783. 3

[70] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "Patchmatchnet: Learned multi-view patchmatch stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 194–14 203. 3

[71] Y. Ding, W. Yuan, Q. Zhu, H. Zhang, X. Liu, Y. Wang, and X. Liu, "Transmvsnet: Global context-aware multi-view stereo network with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8585–8594. 3

[72] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *ECCV (8)*, 2016. 3

[73] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3d object reconstruction," in *2017 International Conference on 3D Vision*. IEEE, 2017, pp. 412–420. 3

[74] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2088–2096. 3

[75] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3577–3586. 3

[76] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions On Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017. 3

[77] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," *Journal of graphics tools*, vol. 8, no. 2, pp. 1–15, 2003. 3, 5

[78] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006, p. 0. 3

[79] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, "Point2mesh: a self-prior for deformable meshes," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 126–1, 2020. 3

[80] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660. 3, 8

[81] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5105–5114, 2017. 3

[82] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019. 3

[83] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan, "Skinning with dual quaternions," in *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, 2007, pp. 39–46. 3

[84] B. H. Le and J. K. Hodgins, "Real-time skeletal skinning with optimized centers of rotation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–10, 2016. 3

[85] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015. 3, 8, 9

[86] G. Moon, T. Shiratori, and K. M. Lee, "Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling," in *European Conference on Computer Vision*, 2020, pp. 440–455. 3, 8, 9

[87] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black, "Learning to dress 3d people in generative clothing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6469–6478. 3, 8, 9

[88] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. 3

[89] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI conference on artificial intelligence*, 2018, pp. 3538–3545. 3

[90] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," *arXiv preprint arXiv:1905.10947*, 2019. 3

[91] H. Nt and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv preprint arXiv:1905.09550*, 2019. 3

[92] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421. 3

[93] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874. 3

[94] J. Zhang, X. Liu, X. Ye, F. Zhao, Y. Zhang, M. Wu, Y. Zhang, L. Xu, and J. Yu, "Editable free-viewpoint video using a layered neural representation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–18, 2021. 3

[95] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470. 3

[96] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174. 3

[97] L. Morreale, N. Aigerman, P. Guerrero, V. G. Kim, and N. J. Mitra, "Neural convolutional surfaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 333–19 342. 3

[98] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 216–224. 3

[99] Q. Ma, S. Saito, J. Yang, S. Tang, and M. J. Black, "Scale: Modeling clothed humans with a surface codec of articulated local elements," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 082–16 093. 3

[100] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2304–2314. 3

[101] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3504–3515. 3

[102] K. Karunratanakul, J. Yang, Y. Zhang, M. J. Black, K. Muandet, and S. Tang, "Grasping field: Learning implicit representations for human grasps," in *2020 International Conference on 3D Vision*. IEEE, 2020, pp. 333–344. 3

[103] D. Palmer, D. Smirnov, S. Wang, A. Chern, and J. Solomon, "Deepcurrents: Learning implicit representations of shapes with boundaries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 665–18 675. 3

[104] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung, "Scalable locally injective mappings," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 1, 2017. 3

[105] R. Poranne, M. Tarini, S. Huber, D. Panozzo, and O. Sorkine-Hornung, "Autocuts: simultaneous distortion and cut optimization for uv mapping," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–11, 2017. 3

[106] R. Sawhney and K. Crane, "Boundary first flattening," *ACM Transactions on Graphics (ToG)*, vol. 37, no. 1, pp. 1–14, 2017. 3

[107] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7297–7306. 3

[108] P. Yao, Z. Fang, F. Wu, Y. Feng, and J. Li, "Densebody: Directly regressing dense 3d human pose and shape from a single color image," *arXiv preprint arXiv:1903.10153*, 2019. 3

[109] T. Zhang, B. Huang, and Y. Wang, "Object-occluded human shape and pose estimation from a single color image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7376–7385. 3

[110] P. Chen, Y. Chen, D. Yang, F. Wu, Q. Li, Q. Xia, and Y. Tan, "I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 929–12 938. 3

[111] T. Z. Buzhen Huang and Y. Wang, "Pose2uv: Single-shot multi-person mesh recovery with deep uv prior," *IEEE Transactions on Image Processing*, 2022. 3

[112] Z. Cao, Q. Huang, and R. Karthik, "3d object classification via spherical projections," in *2017 international conference on 3D Vision*. IEEE, 2017, pp. 566–574. 4

[113] Y. Rao, J. Lu, and J. Zhou, "Spherical fractal convolutional neural networks for point cloud recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 452–460. 4

[114] C. Peng and S. Timalsena, "Fast mapping and morphing for genus-zero meshes with cross spherical parameterization," *Computers & Graphics*, vol. 59, pp. 107–118, 2016. 4

[115] C. Chen, K. Su, and X. Zhu, "Topological disk mesh morphing based on area-preserving parameterization," *Wuhan University Journal of Natural Sciences*, vol. 23, no. 3, pp. 201–209, 2018. 4

[116] T. Akenine-Mller, E. Haines, and N. Hoffman, *Real-Time Rendering, Fourth Edition*, 4th ed. USA: A. K. Peters, Ltd., 2018. 5

[117] J. Goldsmith and J. Salmon, "Automatic creation of object hierarchies for ray tracing," *IEEE Computer Graphics and Applications*, vol. 7, no. 5, pp. 14–20, 1987. 5

[118] D. A. Reynolds, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009. 5

[119] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3. 6

[120] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 6000–6010, 2017. 7

[121] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021. 7

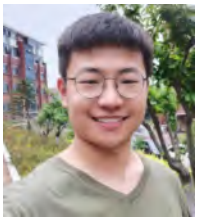
[122] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 7

[123] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. B. Hamza, A. Bronstein, M. Bronstein *et al.*, "Shape

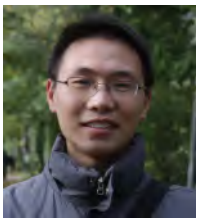
retrieval of non-rigid 3d human models," *International Journal of Computer Vision*, vol. 120, no. 2, pp. 169–193, 2016. 8, 9
[124] C. Zhang, S. Pujades, M. J. Black, and G. Pons-Moll, "Detailed, accurate, human shape estimation from clothed 3d scan sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4191–4200. 8, 9, 11
[125] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 3593–3601. 8, 10, 11, 12
[126] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153. 8, 9
[127] W. Zhao, S. Gao, and H. Lin, "A robust hole-filling algorithm for triangular mesh," *The Visual Computer*, vol. 23, no. 12, pp. 987–997, 2007. 8
[128] Z. Yu, J. S. Yoon, I. K. Lee, P. Venkatesh, J. Park, J. Yu, and H. S. Park, "Humbi: A large multiview dataset of human body expressions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2990–3000. 8
[129] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black, "Monocular expressive body regression through body-driven attention," in *European Conference on Computer Vision*. Springer, 2020, pp. 20–40. 8
[130] P. Patel, C.-H. P. Huang, J. Tesch, D. T. Hoffmann, S. Tripathi, and M. J. Black, "Agora: Avatars in geography optimized for regression analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 468–13 478. 8
[131] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7564–7573. 8
[132] G. Tiwari, B. L. Bhatnagar, T. Tung, and G. Pons-Moll, "Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing," in *European Conference on Computer Vision*. Springer, 2020, pp. 1–18. 8
[133] Y. Yang, Y. Yu, Y. Zhou, S. Du, J. Davis, and R. Yang, "Semantic parametric reshaping of human body models," in *2014 2nd International Conference on 3D Vision*, vol. 2. IEEE, 2014, pp. 41–48. 8
[134] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele, "Building statistical shape spaces for 3d human modeling," *Pattern Recognition*, vol. 67, pp. 276–286, 2017. 8, 10
[135] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3d mesh segmentation," *Acm transactions on graphics (tog)*, vol. 28, no. 3, pp. 1–12, 2009. 8, 9
[136] D. Kulon, H. Wang, R. A. Güler, M. M. Bronstein, and S. Zafeiriou, "Single image 3d hand reconstruction with mesh convolutions," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2019, p. 45. 9
[137] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "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*, 2019, pp. 813–822. 9
[138] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4990–5000. 9
[139] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 807–11 816. 9
[140] Z. Zhao, B. Zuo, W. Xie, and Y. Wang, "Stability-driven contact reconstruction from monocular color images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1643–1653. 9
[141] "The utah 3d animation repository." http://www.sci.utah.edu/~wald/animrep/, 2017. 9
[142] "Thingiverse." https://www.thingiverse.com/, 2022. 9
[143] "Turbosquid." https://www.turbosquid.com/, 2022. 9
[144] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Transactions on graphics (TOG)*, vol. 23, no. 3, pp. 399–405, 2004. 9
[145] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 9
[146] B. Huang, Y. Shu, T. Zhang, and Y. Wang, "Dynamic multi-person mesh recovery from uncalibrated multi-view cameras," in *2021 International Conference on 3D Vision*. IEEE, 2021, pp. 710–720. 9
[147] C. Loop, "Smooth subdivision surfaces based on triangles," *Master's thesis, University of Utah, Department of Mathematics*, 1987. 9
[148] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. 9
[149] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," *SIAM journal on computing*, vol. 18, no. 6, pp. 1245–1262, 1989. 10
[150] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions." *SIGGRAPH sketches*, vol. 10, no. 1, pp. 22–es, 2007. 10
[151] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 1999. 13

Zimeng Zhao received the bachelor's degree in Automation from Southeast University, Nanjing, China, in 2019. He is currently a Ph.D. student advised by Yangang Wang at Southeast University. His research interests span the fields of geometry computing, physical-based animation, and dynamic reconstruction.

Wei Xie received the bachelor's degree in Automation from Chongqing University of Posts and telecommunications, Chongqing, China. She is currently a M.S. Student advised by Yangang Wang at Southeast University. Her research interests include 3D reconstruction, geometry computing, and physics-based animation.



Binghui Zuo received the bachelor's degree in Automation from Qingdao University of Technology, Qingdao, China. He is currently a M.S. Student advised by Yangang Wang at Southeast University. His research interests include implicit reconstruction and physics-based animation.



Yangang Wang received his B.E. degree from Southeast University, Nanjing, China, in 2009 and his Ph.D. degree in control theory and technology from Tsinghua University, Beijing, China, in 2014. He was an associate researcher at Microsoft Research Asia from 2014 to 2017. He is currently an associate professor at Southeast University. His research interests include image processing, computer vision, computer graphics, motion capture, and animation.