SUPPLE : Extracting Hand Skeleton with Spherical Unwrapping Profiles

Zimeng Zhao Ruting Rao Yangang Wang*

Southeast University, China

Abstract

Embedding a unified skeleton into diverse hand meshes is a prominent task both for animation and pose estimation. Most existing methods extracted skeletons from humanoid characters under simple poses, e.g. *T-pose or A-pose. Applying them directly to hand meshes may yield inaccurate or implausible results because hands have higher dexterity and similar endpoints. Furthermore, these methods did not attempt to extract skeleton directly from a scan model which may be not watertight and has much more vertices. Our key idea is to unwrap meshes with different topologies in the same image-based representation, named SUPPLE (Spherical UnwraPping ProfiLEs), and then train a convolutional encoder-decoder to extract skeleton under this representation. Experiments demonstrate that our framework produces reliable and accurate skeleton estimation results across a broad range of datasets, from raw scans to artist-designed models.*

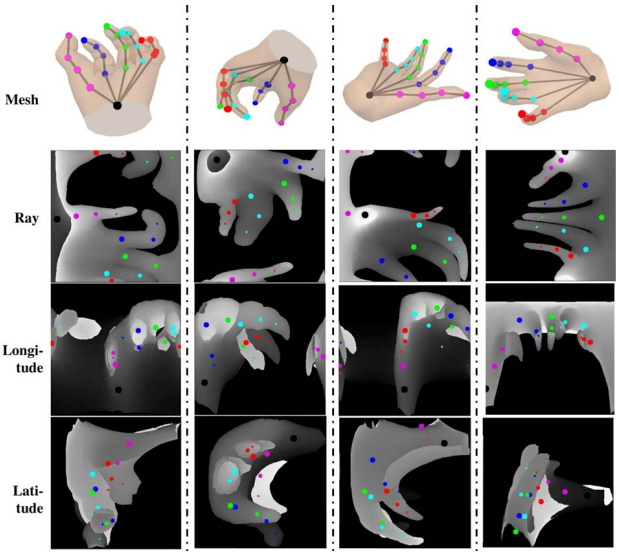


Figure 1. **Hand mesh and SUPPLE with joint annotations.** Each column corresponds to a mesh and its associated SUPPLE profiles. The same color is used to paint the joints from the same finger. In each profile, the circle size of the joint is smaller than the size of its parent.

1. Introduction

Advances in learning-based 3D vision are boosting the acquisitions of personalized hand scans from a set of images. 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 hand pose [27, 9], perform hand animations [43, 5, 69] and retarget hand motions [3, 71]. Therefore, extracting skeletons from those data quickly and directly is critical.

Most existing methods extract skeleton from humanoid characters [5, 69] or full-body scans [51]. In those tasks, the distinction of the shape is more considered, while the pose would be pre-aligned to a unified state (T-pose or A-pose), and the topology should be water-tight. However, the pose may be diverse for a hand scan, and the captured topology may contain noise and holes. Furthermore, the fingers on a hand have more geometric similarities than the limbs of a body, making chirality more difficult to discern.

Numerous learning-based approaches effectively estimate hand pose (skeleton) from RGB [78, 27, 18, 77] or depth [73, 47] with the explosion of image datasets. They are basically attributed to the successful collaboration between image data and CNN structures. When applying to typical 3D data, however, imperfections in the representation occur: neither triangle mesh with GCN [18, 12], voxel grid with 3D-CNN [68, 47, 44], nor point cloud with PointNet [16, 19, 57] could save computation while retaining neighborhood connectivity. This hinders the efficiency of skeleton extraction from 3D hand models.

To this end, we first propose SUPPLE , a novel surface-to-image representation that can be fast converted from mesh. Our key idea is to recast the 3D hand skeleton extraction as a 2D key-points localization task defined on SUP-

*Corresponding author. E-mail: yangangwang@seu.edu.cn. All the authors from Southeast University are affiliated with the Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, China.

PLE . SUPPLE is a combination of three complementary projections designed to unwrap a 3D model under spherical coordinates. Compared with the projections in Cartesian space, the spherical projection significantly reduces the mutual coverage between the surface parts, making the surfaces better fill the whole image. Compared with learning directly from mesh, SUPPLE is grid-based and independent to any fixed surface topologies; Compared with the point cloud, it is ordered and easy to obtain neighborhood features; Compared to voxel-grid, it records surface more densely and allows for a deeper network because of the minor computational complexity.

When extracting 3D skeleton based on SUPPLE , the advantages of CNN are given full play: through the convolution at the image level, the features of 3D surface are efficiently extracted; The architecture based on residual blocks and the overall layout as encoder-decoder overcome the defect of the in-homogeneity of spherical projections at the equator and poles. Furthermore, the data augmentation in the training process also enables our method to extract consistent skeletons from hand scans containing noise and the cartoon hand models designed by artists.

The main contributions of this work are summarized as follows.

- A novel representation is proposed to unwrap a 3D surface into image-like data effectively;
- A deep CNN architecture consuming our representation is designed to extract skeleton from hand scans and artist-designed models;
- A series of mesh data augmentation strategies are provided to significantly increase the size of the available training dataset.

2. Related Work

2.1. Skeleton Embedding

In different applications, skeletons vary in concept from curve skeleton [4, 8, 39, 52] to animation skeleton [43, 5]. Here we concern more about the latter one, *i.e*. a tree hierarchy with a series of joint locations.

Automatic Rigging. Most automatic rigging methods [5, 69, 70] aimed to embed (or extract) a skeleton of humanoid character. These character meshes are mostly designed by artists and placed under simple scan poses like T-pose and A-pose. Some other work [61, 14, 37, 24] performed this task with multiple example poses. These poses must be defined based on the same mesh topology. Applying them directly to hand meshes may produce erroneous or implausible results due to the increased dexterity and similar endpoints of hands. Furthermore, these methods did not attempt to extract skeleton directly from a scan model that may be not watertight and has much more vertices.

Hand Pose Estimation. Skeleton has become an efficient entity to describe hand pose in both 2D [62, 9, 67] and 3D [78, 27, 18, 77] pose estimation. There is a tendency that more works prefer to obey the same skeleton embedding style [63, 62, 73, 65, 79, 9, 76, 48, 49], which facilitates the comparison and inheritance of former ones. This work aims to estimate the joint position of the skeleton under this definition from a given mesh.

2.2. Surface Representation

In the realm of 2D learning, there exist dominant representation and paradigms [33, 25, 59]. These topics, however, are still in their infancy in 3D learning.

Explicit Data. The surface of an articulated object is represented as the polygonal mesh in most skinning algorithms [43, 30] and parametric models [41, 48]. Both rendering and skeletal animation are beneficial from it. However, it is not straightforward to find correspondence between two meshes with different topologies, which impairs the efficiency of the learning-based method. To extract features of the 3D surface, voxelization of surface data is perhaps the most natural extension of the well-known learning paradigms that have excelled in the 2D image domain. However, due to the cubically growing memory requirements, the work using voxel grid [68, 13] or its variants [22, 64] can not use a higher voxel resolution on the whole space to preserve fine surface details. On the other hand, point clouds with sufficient detail can recover surface information through multiple schemes [38, 31, 23]. However, feature extraction directly from point clouds usually requires extra sampling and neighborhood aggregation [55, 56, 66, 57], which is caused by the disorder of point cloud data.

Implicit Function. Several attempts [15, 29, 45, 28] have been made to implicitly represent the body surface using the neural occupancy function determined by the joint locations and rotations. This representation is useful for detecting collisions between objects, however it is ineffective for self-intersection [45]. To convert back to mesh, it relies on the explicit model [41] with the same joint configuration to determine a spatial range of the query point sets.

Unwrapping. UV map [6, 54, 60] is initially created to flat the surface of a 3D model to easily wrap textures. Some studies [6, 21, 72, 74, 11] utilized UV map to store the 3D position vertices. Under this representation, the seam destroys the continuity of the surface, and the impact of different seam designs on the estimation results has not been quantified. In addition, the UV map still relies on the topology constraints of the original mesh. This work employs another surface unwrapping method based on spherical projection. Some pioneer work [36, 10, 53] utilized this unwrapping method for surface correspondence mapping, registration, and object classification. We extend the vanilla

spherical projection to three kinds of profiles. Under our representation, the information between them compensates for each other, which is no longer limited to recording the shape information of aligned geometric objects and the surface variety of the hand under different poses.

3. Method

3.1. SUPPLE Formulation

SUPPLE is a surface-to-image representation independent of mesh topology and vertex number. It unwraps the 3D surface to an image using spherical projection, which preserves the connectivity on the original surface. Due to a single projection only keeps a monotonous profile of the 3D surface, three mutually compensated projections are utilized to record the surface from different perspectives.

Normalization. As shown in Fig. 2 (a), a mesh is first normalized into the unit sphere, so that the spherical coordinates of arbitrary point $P(\rho, \theta, \varphi)$ on the surface $\partial\mathbf{\Omega}$ (not limited to vertices) have the following range: the distance from the origin to the point $\rho \in [0, 1]$; The angle between the positive z-axis and the ray from the origin to the point $\theta \in [0, \pi]$; The angle between the positive x-axis and the same ray above $\varphi \in [0, 2\pi]$. To do that, $P(x, y, z)$ is converted to the spherical coordinate by:

$$\begin{cases} \rho = \sqrt{x^2 + y^2 + z^2} \\ \theta = \text{acos}(\frac{z}{\rho}) \\ \varphi = \text{atan2}(y, x) \end{cases} \tag{1}$$

Because atan2 ranges from $-\pi$ to π, 2π are added to the negative φ results. If not explicitly stated, the 3D coordinates used later in this paper are all spherical coordinates.

Ray Profile. The recorded surface point in this profile $\mathbf{P}_r$ can be considered to be a scan from a lidar fixed at the origin. In each direction determined by (θ, φ), the ρ value of the outermost point on $\partial\mathbf{\Omega}$ is reserved at:

$$\mathbf{P}_r(\frac{\theta}{\pi}W_a, \frac{\varphi}{2\pi}H_a) = \arg\max_{\rho} \left\{ P | P \in (\overrightarrow{OR} \cap \partial\mathbf{\Omega}) \right\} \tag{2}$$

where $\overrightarrow{OR}$ is a ray from the origin with direction $(1, \theta, \varphi)$. Specifically, ray-mesh intersections [46] are repeatedly tested between $\partial\mathbf{\Omega}$ and the ray. If the intersections between the surface and the ray occur, the farthest intersections are recorded; Otherwise, the value is set to be zero.

Longitude Profile. The recorded surface point in this profile $\mathbf{P}_s$ can be considered to be a combined section parallel to the equatorial plane. In each longitude determined by (φ, ρ), the normalized θ value of the point closest to the XoY plane on $\partial\mathbf{\Omega}$ is recorded at:

$$\mathbf{P}_s(\frac{\varphi}{2\pi}W_a, \rho H_a) = \frac{1}{\pi} \arg\min_{\|\theta - 0.5\pi\|} \left\{ P | P \in (\widehat{A_1 A_2 A_3} \cap \partial\mathbf{\Omega}) \right\} \tag{3}$$

where the arc $\widehat{A_1 A_2 A_3}$ is a semicircle with radius ρ and circumscribed by $A_1(\rho, 0, \varphi), A_2(\rho, 0.5\pi, \varphi), A_3(\rho, \pi, \varphi)$. It is generated by using a half-plane that passes through the z axis and A_2 to clip the mesh to find the intersection curve at first, and then change the ρ of the semicircle from 0 to 1.0 with interval $\frac{1}{H_a}$ to find the intersection points to the above curve.

Latitude Profile. In each latitude (circle) determined by (ρ, θ), the normalized φ value of the point closest to the $\varphi = \pi$ half-plane on $\partial\mathbf{\Omega}$ is unwrapped at:

$$\mathbf{P}_c(\rho W_a, \frac{\theta}{\pi}H_a) = \frac{1}{2\pi} \arg\min_{\|\varphi - \pi\|} \left\{ P | P \in (\odot_C \cap \partial\mathbf{\Omega}) \right\} \tag{4}$$

where the circle $\odot_C$ is a latitude pass through all $C(\rho, \theta, \forall\varphi)$. In practice, it is generated by using a sphere centered at the origin and radius ρ to clip the mesh to find the intersection curve at first, and then change the θ of the circle from the northernmost pole to find the intersection points to the above curve.

Concatenation. The three profiles are created separately and stored in three channels of a color image $(3, H_a, W_a)$:

$$\text{SUPPLE} = \mathbf{P}_r \otimes \mathbf{P}_s \otimes \mathbf{P}_c \tag{5}$$

where $\otimes$ indicates the concatenation operation. Although larger image size selection means that more detailed surface information will be recorded in SUPPLE , we choose the image resolution as $H_a = W_a = 128$ in practice. BVH [20] is adopted to accelerate all the mesh-primitives intersection calculations mentioned above.

Inverse Conversion. In a generated SUPPLE , each non-zero pixel corresponds to a point on the surface. Fig. 3 shows the overall point distribution after converting all pixels back to Cartesian space, which is dense enough and complement each other to cover the whole surface under a variety of hand poses. These dense point clouds can be reconstructed as a mesh through the marching cube [38], which is similar to the implicit methods [15, 45].

3.2. Skeleton Extraction

When representing a hand mesh as a SUPPLE image, the skeleton extraction can be regarded as a 2D key-points location task. The pipeline of our method is shown in Fig. 4. It has the following key components.

Profile Generation. The given hand mesh is transformed into a SUPPLE image following 3.1. The input hand mesh can vary significantly in terms of topology and poses. Fig. 4 shows an artist-designed hand mesh example collected online. In addition, as pointed out in work [16], if the central

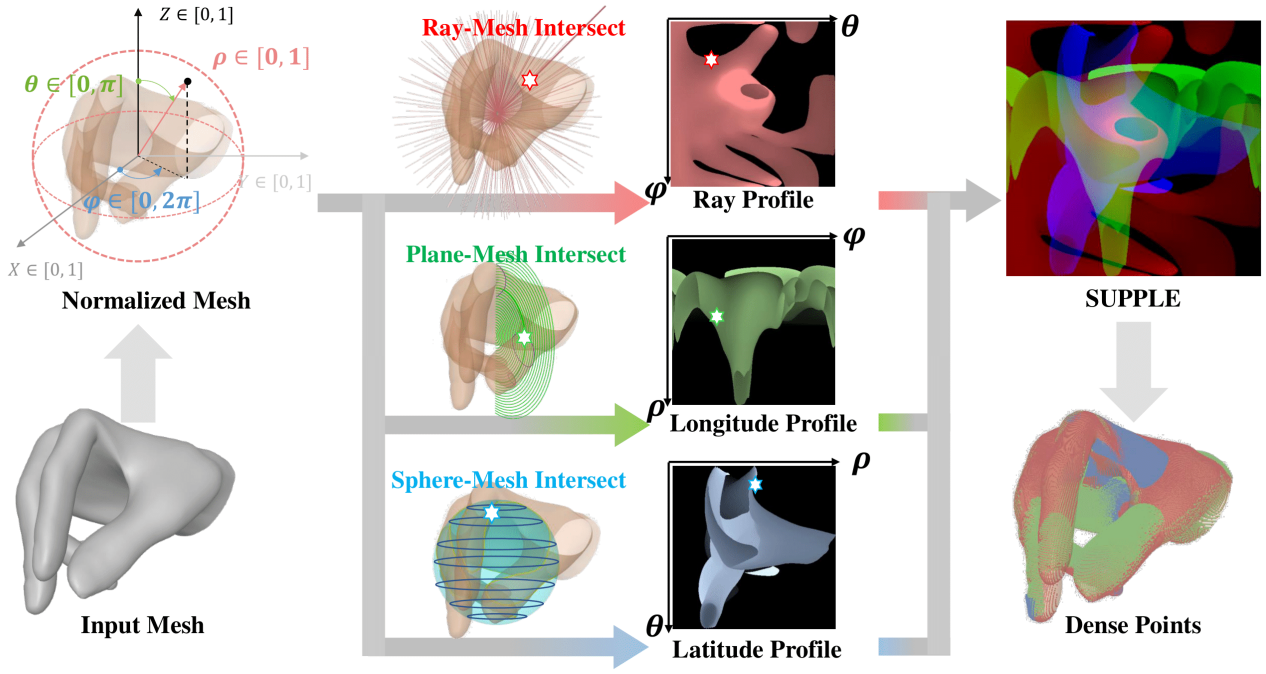


Figure 2. **Mesh-SUPPLE Conversion**. (a) The input mesh is first normalized into the unit sphere, which makes it easier to use spherical coordinates later; (b) Three different profiles were generated through intersecting tests. The colors in the figure are only for illustration, and they are actually grayscale. (c) SUPPLE can be converted back to a dense point cloud by querying pixels.

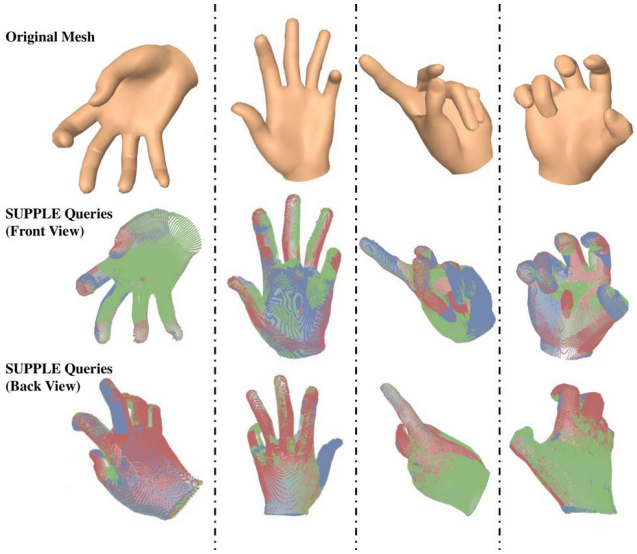


Figure 3. **Mesh and queried point clouds.** Each column corresponds to a mesh and its associated point cloud queried from SUPPLE . The first row is the original mesh. The other two rows are the point clouds from different views. The points marked as red are from the ray profile, green from the longitude profile, and blue from the latitude profile.

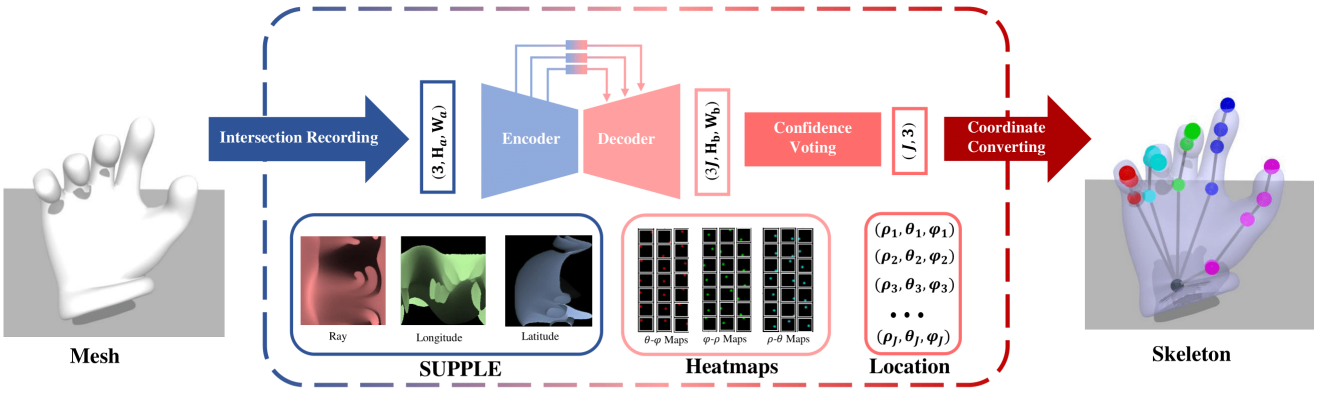
axis of the input model is aligned using PCA in advance, the difficulty of learning can be reduced. This step is optional in our method: even if the input mesh contains global rotation, our method can still extract proper skeletons.

Heatmap Regression. Fig. 1 shows several SUPPLE examples with the projection of skeleton joints. Our encoder-decoder network learns to estimate $J = 21$ joint heatmaps from each profile. Consequently, the output shape of this network is $(63, H_b, W_b)$. For $0 \leq k < J$, we assume that the first 21 channels $\mathbf{H}_r^{(k)}$ are regressed from $\mathbf{P}_r$, the middle 21 channels $\mathbf{H}_s^{(k)}$ from $\mathbf{P}_s$, and the last 21 channels $\mathbf{H}_c^{(k)}$ from $\mathbf{P}_c$. It should be noted that this division is purely for supervisory convenience and that these channels are not separated during the forward propagation. After balancing the amount of network parameters and the accuracy of estimation, we set $H_b = W_b = 32$. Consistent with other work on image-based heatmap regression [27, 67, 77], the ground-truth heatmap is defined by:

$$\mathbf{H}^{(k)}(u, v) = \exp\left(-\frac{(u - u_k)^2 + (v - v_k)^2}{2\sigma^2}\right) \quad (6)$$

For the joint k with the coordinate $(\rho_k, \theta_k, \varphi_k)$, image coordinate (u_k, v_k) refers to $(\frac{\theta_k}{2\pi}W_b, \frac{\varphi_k}{2\pi}H_b)$ in $\mathbf{H}_r^{(k)}$, $(\frac{\varphi_k}{2\pi}W_b, \rho_k H_b)$ in $\mathbf{H}_s^{(k)}$, and $(\rho_k W_b, \frac{\theta_k}{2\pi}H_b)$ in $\mathbf{H}_c^{(k)}$. σ is set as 2.0 in our experiments, which means that the sum of pixel values for a single channel should be less than 8π (the integral constraint of the Gaussian).

Network Architecture. The detailed structure of our joint extraction network is shown in Fig. 5. It first uses two parallel convolutions to extract features from the input profiles. The two branches are then concatenated and fed into an encoder module with 5 residual blocks that progressively



(a) Profile Generation (b) Heatmap Regression (c) Joint Localization

Figure 4. **Pipeline of the skeleton extraction process.** (a) An input mesh is firstly converted as a SUPPLE image; (b) This profile combination is then processed through an encoder-decoder network, whose task is to perform heatmap regression of the joint in each profile; (c) The three heatmaps corresponding to each joint are transformed into the joint coordinates by voting.

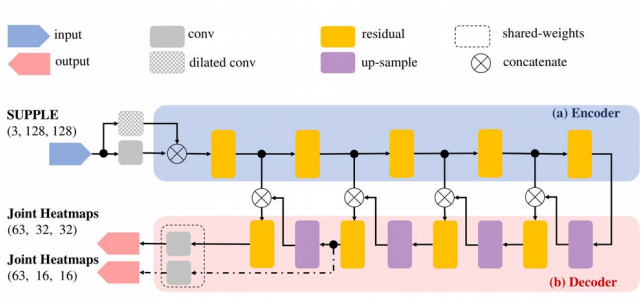


Figure 5. **Network architecture for skeleton extraction.** (a) The encoder module with 5 residual blocks; (b) The decoder module with 4 groups of residual blocks and up-sampling. The dashed arrow means that the process is only used during training.

encode profile feature maps with the gradual enlargement of receptive fields. The decoder consists of 4 layers of stacked residual block and up-sampling. Each layer takes a smaller feature map produced later by the encoder as input, scaled up by up-sampling to the same size as the feature map produced earlier by the encoder and concatenated together. Except for the final output, leaky-ReLU [42] is placed between layers for activation. The final output with size $(63, 32, 32)$ is passed through the channel-wise softmax and multiplied by 8π. All the padding modes in the network are circular to account for the connectivity between the profile's edges. To speed up the convergence of the training process, the second to last feature maps are also exported by the shared convolution of the final output. Both predicted joint heatmaps are supervised using mean squared error with the ground-truth generated using Eqn. 6.

Joint Localization. When using the predicted peak values $\hat{c}_r^{(k)}, \hat{c}_s^{(k)}, \hat{c}_c^{(k)}$ respectively in $\hat{\mathbf{H}}_r^{(k)}$ at (u_r, v_r), $\hat{\mathbf{H}}_s^{(k)}$ at (u_s, v_s), and $\hat{\mathbf{H}}_c^{(k)}$ at (u_c, v_c) as the confidences of the joint coordinate $(\hat{\rho}_k, \hat{\theta}_k, \hat{\varphi}_k)$, the information is redundant. We take advantage of this redundancy to create a mechanism for voting on each joint coordinate. As for $\hat{\rho}_k$, it is determined by:

$$\hat{\rho}_k = \begin{cases} \dfrac{v_s}{H_b}, & \hat{c}_s^{(k)} > \hat{c}_c^{(k)} \\ \dfrac{u_c}{W_b}, & \text{otherwise} \end{cases} \tag{7}$$

Similarly, $\hat{\varphi}_k$ is determined by comparing $\hat{c}_r^{(k)}$ and $\hat{c}_s^{(k)}$, $\hat{\theta}_k$ is determined by comparing $\hat{c}_c^{(k)}$ and $\hat{c}_r^{(k)}$. The channel-wise normalization in the previous step ensures that these comparisons are on the same order of magnitude. Finally, the joint coordinate is converted from spherical to Cartesian $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$ by the inverse of Eqn. 1.

Data Augmentation. Our training dataset contains the following components to ensure that the methods are robust to shape, pose, chirality, water-tightness, and wrist-length:

- $100K$ instances generated randomly from MANO. The shape parameters are sampled from the distribution $\mathcal{N}(0, 5)$, the global pose from $\mathcal{N}(0, 2\pi)$, and the local pose from $\mathcal{N}(0, 0.6\pi)$. The chirality is randomly selected with equal probability;
- $100K$ instances generated from a linear blend skinning (LBS) model with denser vertices. The pose parameters are randomly sampled in the same manner as the previous. In addition, the data is augmented in wrist length variation and surface wrinkles shown in Fig. 6. The length of the wrist region is evenly sampled between 0.1 and 0.8 times the length of the palm. Wrinkles are simulated by adding a random length offset from the distribution $\mathcal{U}(-0.06, 0.06)$ on each vertex in the direction of its normal;
- 5K instances randomly cropped from DeepHandMesh (DHM) scans [48], whose origin hand scan models include forearm components. Among them, $2K$ instances are used for subsequent quantitative testing;

- $40K$ instances from Panoptic Studio [62] multi-view reconstruction results fitted by Kulon *et al*. [35]. This is the part of the training set provided in [35]. The other parts are used for testing.

To further improve the robustness, random noise is added independently on each profile with an existence probability 0.5. The distribution is similar to the salt-and-pepper noise, and the value is evenly sampled from 0 to 1.0.

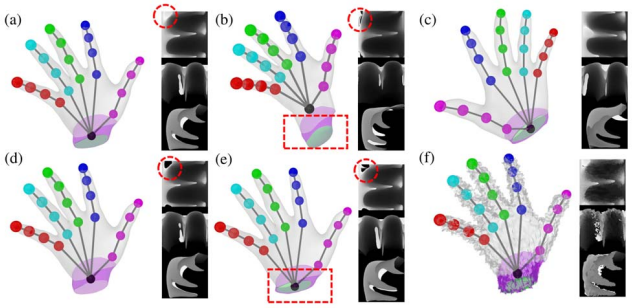


Figure 6. **Mesh data augmentation strategies.** All instances are set to the same pose for convenience of observation. The wrist regions are painted purple, and the base regions are painted green; the other regions are gray. (a) The water-tight mesh; (b) The mesh with a longer wrist; (c) The mesh with mirror transformation; (d) The mesh without base (has a hole on the wrist); (e) The mesh with shorter wrist; (f) The mesh with random surface wrinkles in normal directions.

Implementation Details. We use Adam optimizer [32] to train our network. Our networks are trained on a single NVIDIA GeForce RTX 3090 GPU at a base learning rate of 1e-4 and a batch size of 128, respectively. PyTorch initializes all weights with the default normal distribution. The variations of 21-joint annotations across datasets mentioned in Sec. 3.2 were not explicitly considered.

4. Experiments

4.1. Comparisons

Accuracy. On the test split of our dataset, our method and alternatives are evaluated quantitatively and qualitatively. As shown in Tab. 1, we compared the accuracy of skeleton extraction on MANO registered scans [58], DeepHandMesh (abbreviated as DHM) [48] testing part, Panoptic fitting [35] testing part, Hand3Dstudio [76], Freihand [79], and Youtube3D [34]. Since a strictly watertight mesh is required by the Pinocchio [5], a traditional mesh hole filling algorithm [75] is adopted only for it in advance. The other four competitors correspond to the state-of-the-art methods using multi-view [17], point cloud [16], voxel grid [70], and direct mesh [69] to extract skeletons in a data-driven manner. In the comparison, model size and estimation accuracy are used as evaluation criteria. Due to the joints generated from [70, 69] are without a specific order, CD-joint error [70] based on Chamfer distance are adopted. All the learning-based methods are trained and evaluated on the same dataset.

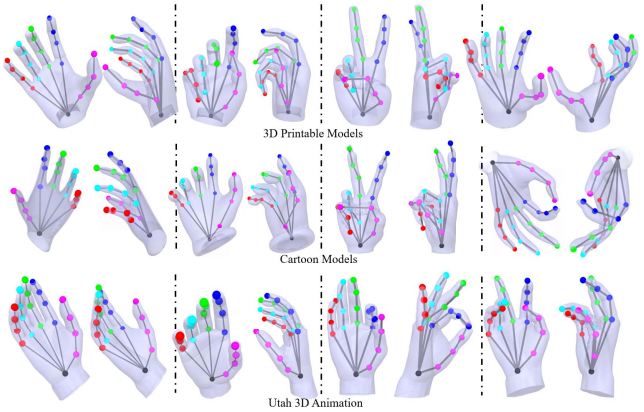


Figure 7. **Qualitative results for skeleton extraction from online meshes.** The samples in each row come from different datasets. Each mesh is viewed from two perspectives. The meshes in the first row are some printable CAD models; The meshes in the second rows are some cartoon models; The meshes in the last row are from [2].

Robustness. With the same model, the robustness is evaluated on both artist-designed hand meshes and hand scans. The artist-designed hand meshes are mainly downloaded from Thingiverse [1] and Utah repository [2]. They differ in shape, pose, topologies and water-tightness. The hand scans are collected from MANO [58] and DHM [48] testing part, as well as some dense hand meshes captured by a hand-held 3D scanner. As shown in Fig. 8, our scans have more diverse shapes; The raw scans in MANO contain many isolated points and holes; The decimated scans in DHM [48] contain different lengths of wrists and incomplete fingers. As illustrated in Fig. 7, some online models contain base or multilayer surface, others include exaggerated fingers or complex topologies. The proposed method has always been able to extract the consistent skeletons from those heterogeneous data because of its superiority in both representation and data augmentation.

4.2. Ablation Study

Conversion Time. The consumption time of converting mesh with different configurations to SUPPLE is shown in Tab. 2. 10K instances are used to test the average time consumption under each configuration. The first mesh configuration is the MANO model; The second one corresponds to the LBS hand model with higher resolution; The last one is the average configuration of hand scans. In each item, the time consumption of the three profiles is counted separately. All the time is tested on Intel Core i7-9700K with 8 cores and 8 threads. With the parallel acceleration of OpenMP and spatial binary search with BVH [20], it is efficient to convert a mesh to SUPPLE .

Methods	Average CD-joint Error (mm)					
	DHM [48]	MANO [58]	Hand3D [76]	Panoptic [62]	Freihand [79]	Youtube3D [34]
Multi-View CNNs [17]	15.101	18.153	21.029	15.006	12.998	28.387
Hand Pointnet [16]	14.279	17.925	20.371	13.074	12.704	27.261
Pinocchio [5]	12.091	17.934	18.803	13.720	12.116	26.043
Volumetric [70]	11.371	16.499	17.312	12.615	11.232	25.310
RigNet [69]	7.893	13.672	14.545	7.926	10.117	23.391
Ours with a variant of $\mathbf{P}_r$	6.570	12.903	11.104	7.131	7.983	15.174
Ours with a variant of $\mathbf{P}_s$	7.102	12.541	11.636	11.619	8.106	16.325
Ours with a variant of $\mathbf{P}_c$	4.327	8.748	7.427	4.302	6.749	11.387
Ours w/o channel softmax	4.361	8.892	7.904	4.387	6.745	13.903
Ours w/o confidence voting	4.347	8.876	7.831	4.411	6.877	14.719
Ours w/o circular padding	4.350	8.874	7.822	4.356	6.752	12.690
Ours with dense block	4.331	8.857	7.491	4.405	6.717	11.903
Ours with 16×16 output size	4.388	9.130	8.048	4.411	7.872	13.996
Ours with 64×64 output size	**4.322**	8.738	7.351	4.302	**6.657**	**11.381**
Ours	**4.323**	**8.737**	**7.349**	**4.291**	**6.657**	**11.381**

Table 1. **Accuracy for skeleton extraction.** CD-joint refers to Chamfer distance between the predicted skeleton joints and the ground-truth. The result of the related competitors are in the first 4 rows; The variants of our methods are in the last 7 rows.

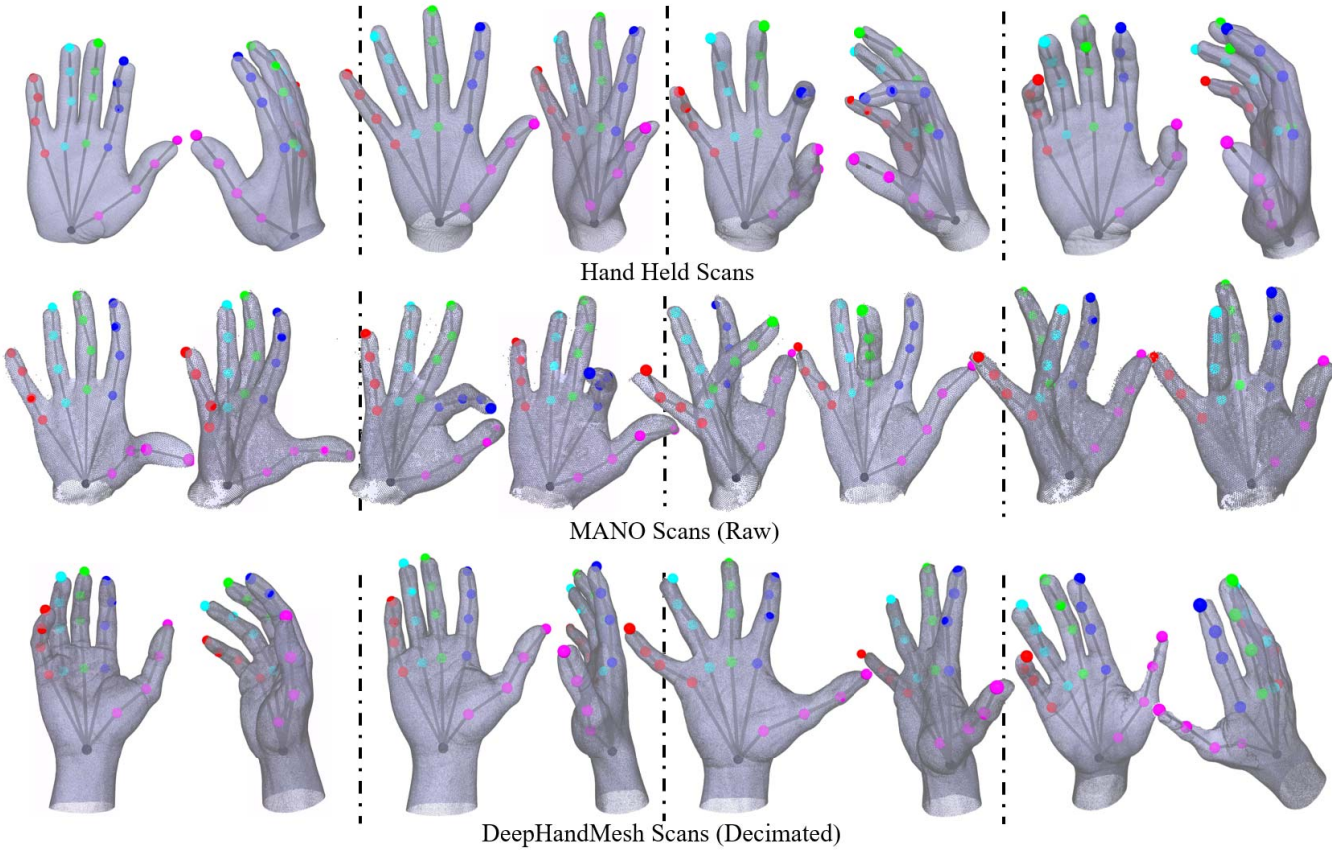


Figure 8. **Qualitative results for skeleton extraction from scans.** The samples in each row come from different datasets. Each mesh is viewed from two perspectives. The dense hand scans in the first row are captured by a handheld scanner. The sparse scans in the second row with isolated points are from MANO. The scans with longer wrists in the last row are cropped scans from DHM testing part.

Coverage Ratio. Since each profile records partial surface, we analyze the coverage of SUPPLE to the original surface. It is compared with the method of projecting directly along the positive direction of x-axis, y-axis, and z-axis in Cartesian space. The following evaluation criterion is adopted to the coverage metric. First, the point cloud

Mesh	Average Time (msec)	
	(128, 128)	(256, 256)
$\#V$=0.7K,$\#F$= 1.5K	57/30/16	225/96/45
$\#V$=7K,$\#F$= 18K	143/96/55	740/200/135
$\#V$=176K,$\#F$= 300K	80/150/443	303/280/1000

Table 2. **Converting time from mesh to SUPPLE .** The time recorded in each row corresponds to the converting time of 3 profiles under the same mesh configuration. Each column corresponds to SUPPLE of the same size.

$\mathbf{S}_A$ is queried from SUPPLE using the inverse conversion described in Sec. 3.1, or from direct projection maps using inverse Cartesian projection. Another point cloud $\mathbf{S}_M$ is sampled by Poisson disk [7] from the original mesh with the same point number as $\mathbf{S}_A$. Then, for each point in $\mathbf{S}_M$, if there exists a point in $\mathbf{S}_A$ is close enough (distance less than 1mm), this point in $\mathbf{S}_M$ is considered covered. As illustrated in Tab. 3, the ratio of covered point number to total point number is used to determine the extent to which the surface is covered. 10K meshes generated by MANO random pose and shape parameters are used for the evaluation.

Profile	Coverage Ratio		
	(256, 256)	**(128, 128)**	(64, 64)
Ray $\mathbf{P}_r$	42%	39%	33%
Longitude $\mathbf{P}_s$	37%	35%	31%
Latitude $\mathbf{P}_c$	39%	36%	33%
SUPPLE	91%	89%	83%
$\mathbf{P}_X \otimes \mathbf{P}_Y \otimes \mathbf{P}_Z$	73%	72%	69%

Table 3. **Coverage ratios of SUPPLE to its original mesh.** The ratio in each row are the single channel, the whole SUPPLE , and the projection map generated in Cartesian space. Each column corresponds to the SUPPLE of the same map size.

Alternative Choices. Evaluations of choices for our method are shown in the last 7 rows of Tab. 1. All the variants are trained in the same split and tuned in the same hold-out validation set.

In terms of the representation, the definitions of each profile were modified slightly. For the ray profile, the variant $\mathbf{P}'_r$ was modified to record the innermost point of each direction (θ, φ); For the longitude profile, its variant $\mathbf{P}'_s$ was modified to record the average θ value on a single longitude arch; For the latitude profile, the variant $\mathbf{P}'_c$ was modified to record the average φ value on a single latitude circle. None of the above variants has exceeded the original version in performance.

In terms of the network modules, we first examined the effect of removing channel soft-max and on the model. The reason for its performance degradation is that during the training process, the weight of some channels may be much greater than that of other channels, thus failing to activate the neurons in the whole network completely. We also tested the effect of switching the confidence voting to the confidence weighting. Although incorrect coordinate estimation has lower confidence, mixing it with higher confidence coordinate estimation will also damage the final result. In addition, we tried to use dense blocks [26] to replace the residual blocks and stack them into a similar network structure. Finally, we evaluated the performance of switching the circular paddings to the zero paddings. This also reduces the reasoning ability of the network because the latter can not make the features distributed on the four edges of the image combine well.

In terms of the size of the output heat map, we tried larger and smaller sizes compared with the final version of 32×32. When using smaller sizes as output, it is found that such accuracy is insufficient in joint localization. On the other hand, the ability to extract skeletons is not considerably improved by using a network with a bigger output size, yet the number of parameters is significantly increased.

5. Conclusion

This paper proposes a surface-to-image representation and corresponding methods for efficiently extracting skeletons from hand meshes without shape, pose, and topology constraints. To the best of our knowledge, they are the first step toward establishing a general, multi-modal framework for hand skeleton embedding. The representation, named SUPPLE , compactly unwraps surface without topology dependency. Compared with other traditional representations, the neural network with SUPPLE can go deeper and extract 3D features more efficiently. An encoder-decoder network is then designed, making the origin task a key-point localization task on SUPPLE . The proposed method shows high accuracy and robustness in extracting skeletons from both noisy hand scans and diverse artist-designed hand models in the experiments. Despite this, There are still several limitations. First, only the mesh-SUPPLE conversion has been thoroughly examined in this work. In the future, it will be obtained directly from hand point clouds or depth maps. Additionally, SUPPLE can be used to represent the surface of objects in a broader context, such as clothed humans or rigid bodies. It would also be interesting to investigate learning methods that jointly regress the correspondences [50] and skinning weights [40] from SUPPLE .

Acknowledgements. This work was supported in part by National Natural Science Foundation of China (No. 62076061, 61806054), Young Elite Scientist Sponsorship Program by the China Association for Science and Technology and "Zhishan Young Scholar" Program of Southeast University.

References

[1] Thingiverse. https://www.thingiverse.com/. 6

[2] The utah 3d animation repository. http://www.sci.utah.edu/~wald/animrep/. 6

[3] Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[4] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM transactions on graphics (TOG)*, 27(3):1–10, 2008. 2

[5] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007. 1, 2, 6, 7

[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 2

[7] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches*, 10(1), 2007. 8

[8] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhinxun Su. Point cloud skeletons via laplacian based contraction. In *2010 Shape Modeling International Conference*, pages 187–197. IEEE, 2010. 2

[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 1, 2

[10] Zhangjie Cao, Qixing Huang, and Ramani Karthik. 3d object classification via spherical projections. In *2017 international conference on 3D Vision (3DV)*, pages 566–574. IEEE, 2017. 2

[11] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong 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*, pages 12929–12938, 2021. 2

[12] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pages 769–787. Springer, 2020. 1

[13] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV (8)*, 2016. 2

[14] Edilson De Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. In *Computer Graphics Forum*, volume 27, pages 389–397. Wiley Online Library, 2008. 2

[15] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *European Conference on Computer Vision*, pages 612–628. Springer, 2020. 2, 3

[16] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018. 1, 3, 6, 7

[17] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel 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*, pages 3593–3601, 2016. 6, 7

[18] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10833–10842, 2019. 1, 2

[19] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 475–491, 2018. 1

[20] Jeffrey Goldsmith and John Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications*, 7(5):14–20, 1987. 3, 6

[21] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018. 2

[22] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 412–420. IEEE, 2017. 2

[23] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: a self-prior for deformable meshes. *ACM Transactions on Graphics (TOG)*, 39(4):126–1, 2020. 2

[24] Nils Hasler, Thorsten Thormählen, Bodo Rosenhahn, and Hans-Peter Seidel. Learning skeletons for shape and pose. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 23–30, 2010. 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 8

[27] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018. 1, 2, 4

[28] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. *arXiv preprint arXiv:2109.11399*, 2021. 2

[29] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020. 2

[30] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007. 2

[31] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[34] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020. 6, 7

[35] Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael M. Bronstein, and Stefanos Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. 6

[36] Ki-Hoon Kwon, Seung-Hyun Lee, and Min Young Kim. A three-dimensional surface registration method using a spherical unwrapping method and hk curvature descriptors for patient-to-ct registration of image guided surgery. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pages 89–92. IEEE, 2016. 2

[37] Binh Huy Le and Zhigang Deng. Robust and accurate skeletal rigging from mesh sequences. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014. 2

[38] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of graphics tools*, 8(2):1–15, 2003. 2, 3

[39] Cheng Lin, Changjian Li, Yuan Liu, Nenglun Chen, Yi-King Choi, and Wenping Wang. Point2skeleton: Learning skeletal representations from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4277–4286, 2021. 2

[40] Lijuan Liu, Youyi Zheng, Di Tang, Yi Yuan, Changjie Fan, and Kun Zhou. Neuroskinning: Automatic skin binding for production characters with deep graph networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 8

[41] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2

[42] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 5

[43] Nadia Magnenat-Thalmann, Richard Laperrire, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface'88*. Citeseer, 1988. 1, 2

[44] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier 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*, pages 7113–7122, 2020. 1

[45] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. Leap: Learning articulated occupancy of people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10461–10471, 2021. 2, 3

[46] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools*, 2(1):21–28, 1997. 3

[47] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu 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*, pages 5079–5088, 2018. 1

[48] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *European Conference on Computer Vision*, pages 440–455, 2020. 2, 5, 6, 7

[49] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 548–564. Springer, 2020. 2

[50] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019. 8

[51] Saifeng Ni, Ran Luo, Yue Zhang, Madhukar Budagavi, Andrew Joseph Dickerson, Abhishek Nagar, and Xiaohu Guo. Scan2avatar: Automatic rigging for 3d raw human scans. In *ACM SIGGRAPH 2020 Posters*, pages 1–2. 2020. 1

[52] JunJun Pan, Xiaosong Yang, Xin Xie, Philip Willis, and Jian J Zhang. Automatic rigging for animation characters with 3d silhouette. *Computer Animation and Virtual Worlds*, 20(2-3):121–131, 2009. 2

[53] Chao Peng and Sabin Timalsena. Fast mapping and morphing for genus-zero meshes with cross spherical parameterization. *Computers & Graphics*, 59:107–118, 2016. 2

[54] Roi Poranne, Marco Tarini, Sandro Huber, Daniele Panozzo, and Olga Sorkine-Hornung. Autocuts: simultaneous distor-

tion and cut optimization for uv mapping. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017. 2

[55] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2

[56] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[57] Hongxing Qin, Songshan Zhang, Qihuang Liu, Li Chen, and Baoquan Chen. Pointskelcnn: Deep learning-based 3d human skeleton extraction from point clouds. In *Computer Graphics Forum*, volume 39, pages 363–374. Wiley Online Library, 2020. 1, 2

[58] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):1–17, 2017. 6, 7

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[60] Rohan Sawhney and Keenan Crane. Boundary first flattening. *ACM Transactions on Graphics (ToG)*, 37(1):1–14, 2017. 2

[61] Scott Schaefer and Can Yuksel. Example-based skeleton extraction. In *Symposium on Geometry Processing*, pages 153–162, 2007. 2

[62] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017. 2, 6, 7

[63] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 824–832, 2015. 2

[64] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 2

[65] Yangang Wang, Cong Peng, and Yebin Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3258–3268, 2018. 2

[66] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[67] Yangang Wang, Baowen Zhang, and Cong Peng. Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization. *IEEE transactions on image processing*, 29:2977–2986, 2019. 2, 4

[68] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1, 2

[69] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: neural rigging for articulated characters. *ACM Transactions on Graphics (TOG)*, 39(4):58–1, 2020. 1, 2, 6, 7

[70] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, and Karan Singh. Predicting animation skeletons for 3d articulated models via volumetric nets. In *2019 International Conference on 3D Vision (3DV)*, pages 298–307. IEEE, 2019. 2, 6, 7

[71] Zhuoqian Yang, Wentao Zhu, Wayne Wu, Chen Qian, Qiang Zhou, Bolei Zhou, and Chen Change Loy. Transmomo: Invariance-driven unsupervised video motion retargeting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5306–5315, 2020. 1

[72] Pengfei Yao, Zheng Fang, Fan Wu, Yao Feng, and Jiwei Li. Densebody: Directly regressing dense 3d human pose and shape from a single color image. *arXiv preprint arXiv:1903.10153*, 2019. 2

[73] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun 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*, pages 4866–4874, 2017. 1, 2

[74] Tianshu Zhang, Buzhen Huang, and Yangang 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*, pages 7376–7385, 2020. 2

[75] Wei Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007. 6

[76] Zhengyi Zhao, Tianyao Wang, Siyu Xia, and Yangang 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)*, pages 2478–2482. IEEE, 2020. 2, 6, 7

[77] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng 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*, pages 5346–5355, 2020. 1, 2, 4

[78] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017. 1, 2

[79] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas 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*, pages 813–822, 2019. 2, 6, 7