

# Deep Virtual Markers for Articulated 3D Shapes

Hyomin Kim   Jungeon Kim   Jaewon Kam   Jaesik Park\*   Seungyong Lee\*  
 POSTECH

## Abstract

We propose deep virtual markers, a framework for estimating dense and accurate positional information for various types of 3D data. We design a concept and construct a framework that maps 3D points of 3D articulated models, like humans, into virtual marker labels. To realize the framework, we adopt a sparse convolutional neural network and classify 3D points of an articulated model into virtual marker labels. We propose to use soft labels for the classifier to learn rich and dense interclass relationships based on geodesic distance. To measure the localization accuracy of the virtual markers, we test FAUST challenge, and our result outperforms the state-of-the-art. We also observe outstanding performance on the generalizability test, unseen data evaluation, and different 3D data types (meshes and depth maps). We show additional applications using the estimated virtual markers, such as non-rigid registration, texture transfer, and realtime dense marker prediction from depth maps.

## 1. Introduction

Estimating positional information of surface points on articulated objects (such as humans and animals) is crucial for various applications such as virtual/augmented reality, movie industries, and entertainments [47, 11, 48, 13]. One common way to capture such information is that actors wear special suits with beacons that are tracked by a special optical system equipped with multiple cameras. Markerless approaches [29, 62, 30] have also been developed, but they are not as robust as using markers and more error-prone in difficult situations like fast motions.

We introduce *deep virtual markers*, a framework that can instantly produce dense and accurate positional annotations for a 3D observation of an articulated object, such as a depth map of the object captured by a depth camera or a 3D point set sampled from the object. In contrast to prior works that predict 3D skeletons [51, 71, 66] or pose parameters of parametric models [8, 34], our approach di-

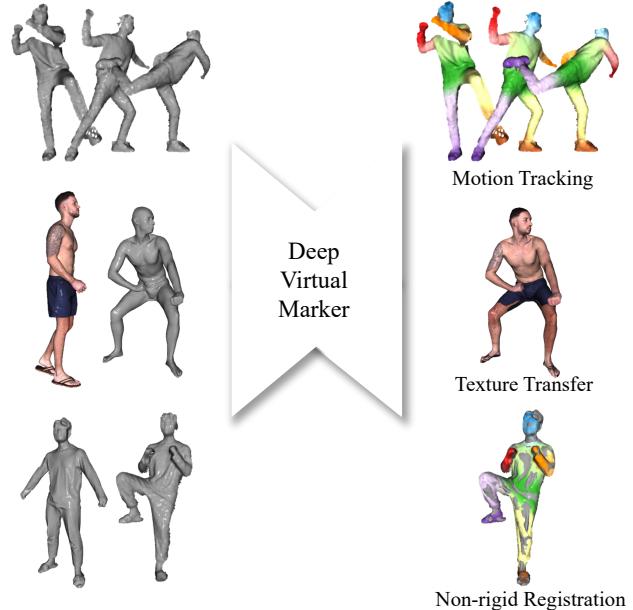


Figure 1. Deep virtual markers instantly produce dense and accurate positional annotations from a mesh or depth map. Applications of deep virtual markers include motion analysis/tracking, texture transfer, and non-rigid registration.

rectly maps any observed 3D points to the canonical locations, providing much richer information. As a result, our approach predicts dense and reliable makers that can be a useful tool for various applications, such as 3D motion analysis, texture transfer, and non-rigid registration of human scans. Examples of the applications are shown in Figure 1.

Our approach for dense marker prediction is learning-based, and it is built upon 3D sparse convolution [15]. Due to the fully convolutional nature of the neural network, the shape can be understood well in the cluttered or occluded cases. The proposed framework accurately determines dense marker positions via single feed-forward pass of the neural network. Therefore, our approach runs in real-time and it does not involve heuristic modules.

To train the network, we propose a new dataset that consists of realistic depth renderings of template models. To

\*Joint corresponding authors.

annotate dense supervisory labels, we begin with sparse marker annotations on the template models, and solve for heat equilibrium to obtain dense annotations, where the annotation of a surface point is represented using the relative influences from sparsely annotated points. The annotated models are augmented with realistic motions from Mixamo [2] and random poses that follows physical ranges of body joints.

The positional accuracy of the predicted virtual markers are evaluated with several benchmarks that provides precise correspondences, such as FAUST [9] and SCAPE [4]. Computing virtual markers of the two scans is equivalent to building dense correspondences, as virtual markers provide positional annotations on common canonical domain. In the evaluation, our approach achieves the state-of-the-art performance in the non-rigid human correspondence identification benchmark. We also perform cross-validation with different datasets.

In contrast to prior works [42, 28, 57, 31, 38, 17, 43, 26, 25], our deep virtual markers are applicable to various 3D data types, such as full 3D meshes, point clouds, and depth maps. Our approach is also intended to be applied for any articulated objects. Therefore, we prepare a new data set for animals as well, and we show that our approach can handle depth map of a real cat. Codes are publicly available.<sup>1</sup>

In summary, our key contributions are as follows:

- We propose a real-time approach to extract dense markers from 3D shapes regardless of data types, such as full 3D meshes and depth maps.
- We propose an effective approach to annotate dense markers for template models for preparing training sets. We also propose a new dataset of dense markers.
- Our approach achieves state-of-the-art performance on non-rigid correspondence identification task that requires accurate marker localization.
- Experiment results show the generalizability of our approach on various datasets and data types. We also demonstrate that our approach can handle various articulated objects such as humans and cats.

## 2. Related Work

The proposed deep virtual marker is closely related to the prior arts that are built for body part recognition, geometric feature descriptor, and non-rigid human body alignment. We review representative approaches here, and the summary is shown in Table 1.

**Body part recognition.** Several works have been proposed to segment body parts of articulated objects using color images or depth images. Many approaches [68, 53, 37, 71,

<sup>1</sup><https://github.com/T2Kim/DeepVirtualMarkers>.

Table 1. Comparison with related work. Our approach is a learning-based real-time method that assigns dense markers. The proposed method can handle a mesh, partial observation (such as depth map), and multiple instances. Our approach does not require heavy pre-processing. In the table, FC Res., GCNN, MLP, AE, and RF indicate fully connected residual net, graph convolutional neural network, multi-layer perceptron, autoencoder, and random forest, respectively. Please see Sec. 2 for the details.

| Method              | Network       | Learning-based | Dense prediction | Handle mesh | Handle partial obs. | No topology | Multiple objects | Real-time | No preprocessing | Publication |
|---------------------|---------------|----------------|------------------|-------------|---------------------|-------------|------------------|-----------|------------------|-------------|
| SmoothShells [21]   | -             |                | ✓                | ✓           |                     |             |                  |           |                  | '20         |
| FARM [43]           | -             |                | ✓                | ✓           |                     |             |                  |           |                  | '20         |
| UnsupFMNet [31]     | AE            | U              | ✓                | ✓           |                     |             |                  |           |                  | '19         |
| SurFMNet [59]       | FC Res.       | U              | ✓                | ✓           |                     |             |                  |           |                  | '19         |
| CyclicFM [26]       | FC Res.       | Self           | ✓                | ✓           |                     |             |                  |           |                  | '20         |
| Deep Shells [22]    | GCNN          | U              | ✓                | ✓           |                     |             |                  |           |                  | '20         |
| Shotton et al. [63] | RF            | S              |                  |             | ✓                   | ✓           | ✓                | ✓         | ✓                | '11         |
| GCNN [45]           | Geo. CNN      | S              | ✓                | ✓           |                     |             |                  |           |                  | '15         |
| DHBC [70]           | 2D CNN        | S              | ✓                | ✓           | ✓                   | ✓           |                  | ✓         | ✓                | '16         |
| Nishi et al. [52]   | 2D CNN        | S              |                  |             | ✓                   | ✓           |                  | ✓         | ✓                | '17         |
| FMNet [42]          | FC Res.       | S              | ✓                | ✓           |                     |             |                  |           |                  | '17         |
| Fey et al. [23]     | SplineCNN     | S              | ✓                | ✓           |                     |             |                  | ✓         | ✓                | '18         |
| FeaStNet [67]       | GCNN          | S              | ✓                | ✓           |                     |             |                  | ✓         | ✓                | '18         |
| Lim et al. [41]     | FC RNN        | S              | ✓                | ✓           |                     |             |                  |           |                  | '18         |
| 3D-CODED [28]       | AE            | S/U            | ✓                | ✓           |                     |             |                  | ✓         | ✓                | '18         |
| SpiralNet++ [27]    | SpiralCNN     | S              | ✓                | ✓           |                     |             |                  | ✓         | ✓                | '19         |
| MGCN [69]           | GCNN          | S              | ✓                | ✓           |                     |             |                  |           |                  | '20         |
| GeoFMNet [18]       | 3D CNN        | S              | ✓                | ✓           |                     |             |                  |           |                  | '20         |
| LSA-Conv [25]       | GCN           | S              | ✓                | ✓           |                     |             |                  | ✓         | ✓                | '20         |
| Marin et al. [44]   | MLP           | S              | ✓                | ✓           | ✓                   | ✓           |                  | ✓         | ✓                | '20         |
| Ours-oneshot        | Sparse 3D CNN | S              | ✓                | ✓           | ✓                   | ✓           | ✓                | ✓         | ✓                | -           |
| Ours-multiview      |               |                |                  |             |                     |             |                  |           |                  |             |

[66, 50, 63, 32, 52] focus on human part segmentation, whereas others target for animal part segmentation [68, 50]. As human part segmentation and human pose estimation are closely related to each other, some works try to tackle both problems simultaneously [71, 66, 50]. Since these works usually adopt data-driven approaches, they need annotated body parts for training.

The works by Shotton et al. [63] and Nishi et al. [52] are close to our approach. Their methods use depth images and learn a classifier to assign human body part labels to depth pixels. However, the part labels obtained by the methods are too sparse to precisely align two human models of different shapes. In contrast, our approach generates dense labels enough for human body alignment.

**Geometric features.** Traditional methods have been developed for hand-crafted geometric features [33, 24, 60, 61, 64, 5]. Recent methods use deep neural networks (DNNs), and GCNN [45], ACNN [10], and Monet [49] define convolutions on non-Euclidean manifolds. Other approaches adopt the graph convolutional network (GCN) to utilize the informative connectivity of input geometry [23, 67, 41, 27, 69, 25]. FCGF [16] adopts 3D sparse convolutional network [15] to compute dense geometric features.

Our approach is largely inspired by FCGF [16].

FCGF [16] shows high performance on partial observations, but it is crafted for rigid registration of point clouds. Our approach extends its applicability to non-rigid articulated objects, such as humans and cats.

**Non-rigid shape matching.** Understanding human body shape and coherent geometric description are actively researched for 3D human body matching. Stitched Puppet [75] devises a part-based human body model and optimizes the parameters of each part to fit the target mesh. 3D-CODED [28] uses autoencoder architecture to deform the template mesh. LoopReg [7] proposes an end-to-end framework that utilizes a parametric model. The parameters are implicitly diffused over 3D space, and the approach utilizes a parametric model to find shape correspondence. Therefore, the representation capability of the model can restrict the final accuracy.

After its introduction for shape matching, a functional map [54] has led many follow-up studies. FMNet [42] extracts SHOT descriptors [61] and transforms them through a fully connected network. CyclicFM [26] proposes a self-supervised learning method that reduces cyclic distortions. While most functional map based methods use the Laplace-Beltrami operator [55] to construct the basis, Marin et al. [44] uses a data-driven approach by learning the probe function. GeoFMNet [18] uses KPConv [65] to enable the network to directly extract features from an input 3D shape. FARM [43] exploits non-rigid ICP algorithm to refine a functional map. Smooth Shells [21] and Deep Shells [22] simplify the shapes by reducing spectral bases, and they align the generated multi-scale shapes progressively in a coarse-to-fine manner.

Functional map-based methods generally conduct pre-processing to construct the basis and need a mesh topology. In contrast, our method does not require a mesh topology at inference time and can handle various 3D data types.

### 3. Method

Our framework assigns *dense virtual markers* to an articulated model. A virtual marker indicates a position on a canonical 3D shape. Our approach takes a raw point cloud and produces a virtual marker for each input point. Our approach is learning-based, so we prepare a new dataset for the training. For this procedure, we annotate sparse markers on a canonical 3D model and densify them to cover the whole surface.

#### 3.1. Virtual markers

Virtual markers are positional labels that are densely distributed on the surface of a canonical 3D shape of an articulated model. A method that predicts such positional labels can help us understand the 3D geometry. For instance, it can be used for direct texture map transfer between two 3D

scans of human body without surface parameterization and finding correspondences, as shown in Figure 1.

Our idea for defining virtual markers is to utilize the intrinsic structure of the shape of an articulated object. For instance, humans have four limbs, one head, and identically working joints. In this manner, even if persons' appearances are not the same, we can think of coherent markers that are placed on the human body.

Similarly to ours, assigning markers on the surface of a human model and solving the label classification problem is the common procedure used in the previous works [63, 52]. However, the most noticeable difference is that the prior methods use hard labels of markers to indicate separate human parts, as shown in Figure 2 (c). On the other hand, our virtual markers can smoothly indicate every single point on the surface of the human model as a soft label based on geodesic distance (Figure 2 (d)).

#### 3.2. Annotation

To establish dense virtual markers for making a training dataset, we begin with annotating sparse markers on the 3D mesh model using the joint positions of a 3D human skeleton (Figure 2 (b)). Then, the sparse markers are densely propagated to cover the whole surface of the model by solving a heat equilibrium equation (Figure 2 (d)). Note that this procedure is to make a dataset for training. In the test time, only a single feed-forward network is used to directly predict dense virtual markers from any type of 3D input data (depth images, point sets, or meshes).

**Sparse virtual markers.** We annotate several points of interest on a mesh model with a rest pose (such as T-pose). Using the annotated points, we construct the skeleton of the template mesh model, and define sparse virtual markers.

Consider a local cylindrical coordinate system  $(\rho, \varphi, z)$  that lies on the  $i$ -th bone of the skeleton (Figure 2 (a)). Origin of the coordinate system is an endpoint of the bone and its longitudinal axis corresponds to the direction of the bone. Consider rays emitted perpendicularly from the longitudinal axis, which can be specified using two coordinates  $(\varphi_i, z_i)$ . We uniformly sample a set of rays using the coordinates, and then the intersections of the sampled rays with the template mesh model become the sparse virtual markers from the  $i$ -th bone. As the surface area of the part associated with each bone differs, we alter the number of samples for each bone. As a result, 99 markers are defined for our human model (Figure 2 (b)). In the case of a cat model, we define 57 sparse markers using the same approach. The detailed positions of the sparse markers are provided in the supplementary material.

For acquiring such markers, it could be possible to use automatic rigging methods to detect the skeleton of the model [6, 72]. However, we experimentally found that these automatic methods often fail for challenging shapes. In-

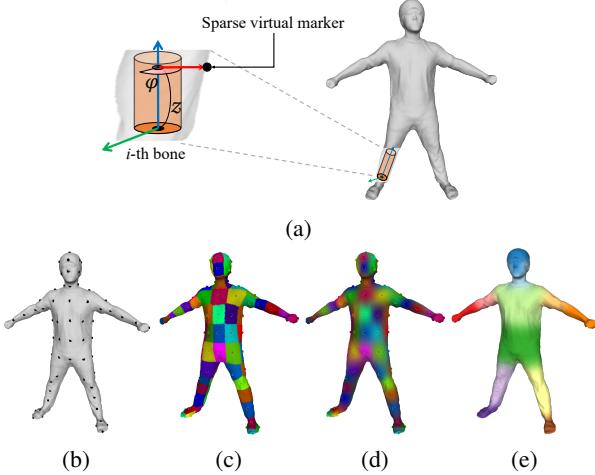


Figure 2. Virtual markers. (a) A sparse virtual marker in the local cylindrical coordinates of a skeleton. The red arrow is the ray used for sampling the marker. (b) Sparse markers on the surface of our human model. (c) and (d) Colored human models with hard and soft labels, respectively. (e) Our color-coding scheme to visualize soft markers.

stead, we hired experienced 3D annotators to assign a precise bone structure by user interaction. The user annotations include joint positions and bone orientations (polar axes of local cylindrical coordinate systems).

**Dense virtual markers.** Given sparse markers, we may classify the surface points by assigning one-hot vectors (or hard labels) to the surface points. This process would result in a conventional part segmentation [63, 52] dataset, as shown in Figure 2 (c). However, such regional labels would not consider any fine-level relationships with multiple sparse markers.

In this work, instead, we use a *soft* label scheme. Given  $S$  sparse markers, the label for an arbitrary surface point is represented as a  $S$ -dimensional vector, whose  $i$ -th element indicates an affinity with the  $i$ -th sparse marker. That is, a soft label can have multiple non-zero entries, differently from a hard label.

We acquire a soft label for every point on the surface by solving for heat equilibrium [6]. Let  $\mathcal{S} = \{s | 1 \leq s \leq S\}$  denote the set of indices for sparse markers, and  $\mathcal{D} = \{d | 1 \leq d \leq D\}$  be the set of surface point indices on the template mesh. We define  $\mathbf{L} \in \mathbb{R}^{S \times D}$ , where  $l_{s,d}$  indicates affinity between the  $d$ -th point on the mesh with the  $s$ -th sparse marker. A column vector of  $\mathbf{L}$  is normalized, so that  $\sum_s l_{s,d} = 1$ . If we define  $\mathbf{L}^\top = \mathbf{W}$  that satisfy  $l_{s,d} := w_{d,s}$ , the weight vector  $\mathbf{w}_s = [w_{1,s}, w_{2,s}, \dots, w_{D,s}]^\top$  of the sparse markers  $s$  for each surface point is calculated by solving the following sparse linear system:

$$-\Delta \mathbf{w}_s + \mathbf{H} \mathbf{w}_s = \mathbf{H} \mathbf{p}_s, \quad (1)$$

where  $-\Delta$  is the Laplacian operator of the mesh,  $\mathbf{p}_s$  is a binary indicator vector,  $\mathbf{H} \in \mathbb{R}^{D \times D}$  is the diagonal matrix having the heat contribution of the sparse marker  $s$  to vertex  $d$ <sup>2</sup>. If sparse marker  $s$  is nearest to vertex  $d$ ,  $p_{s,d}$  is set to 1, and the  $d$ -th diagonal component  $H_{d,d}$  of matrix  $\mathbf{H}$  is defined as  $\frac{c}{\mathcal{D}(d,s)^2}$ , where  $c$  is a constant (we use  $c = 1$ ) and  $\mathcal{D}(d,s)$  is the geodesic distance from vertex  $d$  to sparse marker  $s$ . If  $k$  virtual markers are equidistant from vertex  $d$ ,  $p_{s,d} = \frac{1}{k}$  and  $H_{d,d} = \frac{kc}{\mathcal{D}(d,s)^2}$ . Otherwise,  $p_{s,d}$  becomes 0.

We solve the heat equilibrium condition in Eq. (1) for every sparse marker. As a result, we have the weight matrix  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_S]$ . The  $d$ -th row of matrix  $\mathbf{W}$  becomes the weights of all sparse markers for vertex  $d$ . We set the  $d$ -th row of matrix  $\mathbf{W}$  to the soft label  $\mathbf{l}_{:,d}$  for vertex  $d$ . The computed soft labels are visualized in Figure 2 (e).

This approach is related to the work of Baran *et al.* [6]. They compute weights between vertices and skeleton for mesh skinning. We extend the idea for computing soft labels for vertices on the template mesh. Besides, instead of Euclidean distance used in [6], we use the *geodesic distance* between a vertex  $d$  and a soft marker  $s$  to acquire  $l_{s,d}$ . Computing soft labels takes about 150 seconds per model, but needs to be done only for training data preparation.

### 3.3. Dataset

With the proposed approaches to define sparse and dense markers, we build two datasets to train a neural network.

**Human dataset.** The dataset is made with 33 full mesh models of humans. They are captured with Doublefusion [74] (6 models) and collected from Renderpeople [3] (17 models) and FAUST [9] (10 models). We subsample meshes of the human models to have 60k vertices for the annotation. The manual annotation is conducted by experienced annotators to provide 33 consistent joint positions.

To deal with various non-rigid deformations, we augment each template mesh model with diverse motions. We obtained 15,000 motions from Mixamo [2]. Note that we *deform the annotated meshes*, so we do not need to annotate the deformed models again. We use linear blend skinning (LBS) [35] to animate the template models. In addition to this pose augmentation, we deploy random movements to the joints of the body. The random motions are bounded by physical ranges of the joints. In this manner, we can obtain a diverse dataset with different people and poses.

In addition to the pose augmentation, we extend the dataset for partial geometric observations, such as depth maps. We produce synthetic depth maps by rendering densely annotated mesh models from arbitrary viewpoints. Since a rendered depth pixel knows its original position in the annotated mesh, each depth pixel obtains a proper soft

<sup>2</sup>We denote the  $s$ -th sparse marker as  $s$ , and the  $d$ -th vertex as  $d$  for notational convenience.

label. We used 40 viewpoints for rendering.

Note that we augment the densely annotated mesh model using the Mixamo and random poses *on-the-fly* in the training phase. For the case of depth map generation, the dataset is augmented even more by using random viewpoints. This scheme introduces great diversity in our training dataset.

**Cat dataset.** To demonstrate that our approach is applicable to other classes of articulated objects, we prepare another dataset for an animal shape. We select two cat models from SMAL [76], and use the same procedure for dense annotation of the models.

### 3.4. Training

We employ the ResUNet architecture based on sparse tensors [15] to classify 3D points of an articulated model. Details of the network architecture is provided in the supplementary material. The classifier directly utilizes soft labels described in Section 3.2 for the supervision.

Our network employs a multi-class cross entropy loss defined as follows:

$$\mathcal{L} = - \sum_d^D \sum_s^S l_{s,d} \log \left( \frac{l'_{s,d}}{\sum_j^S e^{l'_{j,d}}} \right), \quad (2)$$

where  $l_{s,d}$  denotes the  $s$ -th element of a soft label  $\mathbf{l}_{:,d}$ , and  $l'_{s,d}$  is the  $s$ -th element of the inferred soft label  $\mathbf{l}'_{:,d}$  for a vertex  $d$ .  $D$  are the number of vertices, and  $S$  is the number of sparse markers.

As shown in Figure 2 (e), soft labels encode the smooth geometric relationships with adjacent surface points on the template mesh model. Consequently, without using any regularization on the smoothness of the prediction, our trained network gets to predict *smoothly varying* virtual markers.

**Two approaches.** We propose to train and test our network with two different configurations.

(1) ***Ours-oneshot*.** In the first setting, for training the network, we use annotations for full mesh models together with partial observations (depths). In this case, the network can infer dense virtual markers for any input data (partial or full) instantly without per-view prediction and merging. This approach saves substantial computation time when inferring virtual markers for full mesh models.

(2) ***Ours-multiview*.** In the second setting, we use only partial observations (3D point cloud of depth maps) to train the network. Then, to handle a full 3D model, we use the rendered depths of the model from 72 different viewpoints. We feed each rendered depth to our network and obtain dense virtual markers for 72 views. Finally, we aggregate them to obtain the dense virtual markers for the whole 3D model. For the aggregation, we convert and combine the 72 rendered depth maps to a single point set, and the virtual marker for a vertex of the 3D model is determined

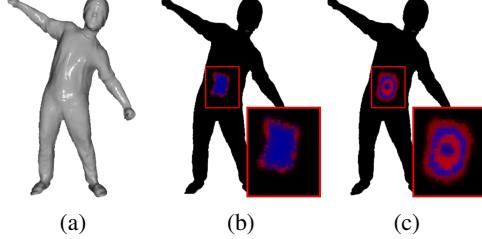


Figure 3. Effects of soft labels. (a) A reference 3D model. (b) and (c) Visualizations of the weights of a sparse marker obtained from networks trained with hard (one-hot) labels and soft labels as supervision, respectively. Red and blue stripes show alternating weight intervals with gap of 0.2 in the range of [0.2 to 1]. Soft labels induce smooth isotropic weight distribution.

by weighted averaging the virtual markers of the  $k$ -nearest neighbors in the point set. This approach is slower than *Ours-oneshot*, but it produces more reliable result due to the multiview concensus.

### 3.5. Timing complexity

We train and test our network with a workstation equipped with an Intel i7-7700K 4.2GHz CPU, 64GB RAM, and NVIDIA Titan RTX, Quadro 8000 GPU. The number of network parameters is about 38 million. Training takes about  $3 \sim 4$  and  $8 \sim 10$  seconds per iteration using partial observations (depths) and full mesh models, respectively. In test time, inference time is about  $0.05 \sim 0.07$  seconds per a depth map or 3D point set, nearly achieving 20 frames per second.

## 4. Results

### 4.1. Effectiveness of soft labels

To analyze the effects of soft labels, we trained two networks using hard labels and soft labels. Figure 3 visualizes the weights of a sparse marker obtained using the two networks. We can find that the network trained with soft labels outputs smoothly varying virtual markers.

### 4.2. Shape correspondence

To validate the accuracy of virtual markers, we extensively evaluate with shape correspondence challenges. The motivations for this experiment are as follows. (1) A large-scale public dataset to measure the accuracy of dense 3D markers is not present. (2) Dataset for finding shape correspondences is well-established instead, and such datasets are captured with high-end 3D capturing systems. (3) Good performance on finding correspondences indicates consistent and coherent marker positions for diverse poses.

**FAUST challenge benchmark** is the standard benchmark to measure the accuracy of finding correspondences from two high-quality human mesh models [9]. This challenge is

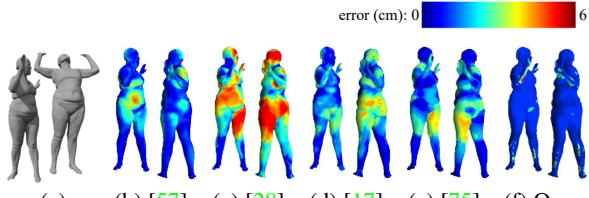


Figure 4. Example of FAUST correspondence finding challenge. We compare our results with BPS [57], 3D-CODED [28], AtlasNetV2 [17], SP [75]. The matching between two meshes on the leftmost is obtained, and its accuracy is visualized using a jet color map. This is an example of the intra-subject test set, where the two meshes are captured from the same person with different poses.

a good fit for validating the proposed *deep virtual markers*, since the accurate markers should be localized for precise matching between non-rigid models.

The total dataset consists of 10 persons, and each person exhibits arbitrary 30 poses. Its ground-truth correspondence is not publicly available, so we submitted our results and obtained the evaluation results. The benchmark test set provides pairs of two full 3D scans. Each pair shows extreme pose variations from the same person (called *intra-subject* challenge) or different persons (called *inter-subject* challenge). Since our approach is not restricted to the person’s identity or pose variations, we report our performance for both tasks. The FAUST experiment is also a good way to check the generalization performance on pose and identity variations.

For comparison, we refer to the numbers in the public FAUST benchmark leaderboard. The leaderboard includes the results of approaches that require manual annotations. Hence we compare only with the methods that do not require manual inputs. The results are shown in Table 2. Our approach shows the most accurate results compared with prior works on both the inter- and intra-subject challenges. Visualizations of errors are shown in Figure 4.

**Comparison with DHBC [70].** The original FAUST challenge only tackles the correspondence for two full meshes, which is called full-to-full correspondence identification. However, handling *partial* observations, such as depth maps, is also important for practical applications. For this reason, we additionally compare our approach with DHBC [70] that can handle full-to-full, full-to-partial, and partial-to-partial dense correspondences.

Since DHBC is not shown on the FAUST leaderboard [9], we conduct the comparison by following the evaluation protocol suggested by Chen et al. [14]. The protocol uses some pairs of 3D scans in the FAUST training set and performs inter- and intra-subject challenges. The protocol reports the average error on all pairs (AE) and the average error on the worst pair (WE). Table 3 shows the results of the experiment on the full-to-full case. We also added the re-

Table 2. FAUST challenge results (error in cm). We omit the results of approaches that require manual annotations in the test time. Approaches are classified into learning-based (first three rows) and non-learning-based methods (the remaining rows).

| Method             | Type of challenge |               |
|--------------------|-------------------|---------------|
|                    | Intra-subject     | Inter-subject |
| SP [75]            | 1.568             | 3.126         |
| RobustCovex [14]   | 4.860             | 8.304         |
| Smooth Shells [21] | -                 | 3.929         |
| FMNet [42]         | 2.436             | 4.826         |
| 3D-CODED [28]      | 1.985             | 2.769         |
| BPS [57]           | 2.010             | 3.020         |
| Unsup FMNet [31]   | 2.510             | -             |
| LBS-AE [38]        | 2.161             | 4.079         |
| AtlasNetV2 [17]    | 1.626             | 2.578         |
| FARM [43]          | 2.810             | 4.123         |
| Cyclic FM [26]     | 2.120             | 4.068         |
| LSA-Conv [25]      | -                 | 2.501         |
| Ours-oneshot       | 1.417             | 2.495         |
| Ours-multiview     | <b>1.185</b>      | <b>2.372</b>  |

Table 3. Additional comparison with DHBC [70] and RobustCovex [14] using the FAUST training dataset (error in cm).

|                  | Intra AE    | Intra WE    | inter AE    | inter WE    |
|------------------|-------------|-------------|-------------|-------------|
| RobustCovex [14] | 4.49        | 10.96       | 5.95        | 14.18       |
| DHBC [70]        | 2.00        | 9.98        | 2.35        | 10.12       |
| Ours-oneshot     | 2.26        | 3.93        | 2.44        | 4.56        |
| Ours-multiview   | <b>1.93</b> | <b>2.76</b> | <b>2.12</b> | <b>3.25</b> |

sults from RobuxtCovex [14]. Our approach favorably outperforms all the cases. In particular, our results show about three times smaller intra-subject and inter-subject WEs than other baselines. Besides, in our case, WEs are similar to AEs. This demonstrates that our approach is reliable and robust to different persons’ identities and various poses.

In addition to the full-to-full comparison, we also perform a quantitative comparison with DHBC [70] with *full-to-partial* and *partial-to-partial* 3D model pairs. The new pairs are obtained by rendering 3D scans from specific viewpoints, and we used FAUST [9] and SCAPE [4] datasets for the rendering. The results are shown in Table 4. Our method outperforms DHBC [70] in these cases too.

**Various training and test sets.** We verify the generalizability of our method by changing the training and test sets. For this experiment, we use the dataset provided by Ren et al. [58]. The dataset contains re-meshed versions of FAUST [9] and SCAPE [4] datasets.

Table 5 shows the results, where the evaluation is based on the protocol proposed by Kim et al. [36]. SCAPE dataset contains a single human model with various poses, and FAUST includes several models with various poses. Hence, for the case of testing on FAUST and training on SCAPE (F / S), the matching accuracies of some approaches [59, 42, 28] are much lower than other combinations since the variety of model shapes for training is insufficient.

Table 4. Finding correspondences using full-to-partial (F2P) and partial-to-partial (P2P) datasets. We compare our results with DHBC [70] using the FAUST [9] and SCAPE [4] datasets for training. For each three numbers, the first one is the error of DHBC, and the second and third are the errors of ours-multiview and ours-oneshot, respectively (error in cm).

| Data      | Type | Method                                    |              |             |               |
|-----------|------|---|--------------|-------------|---------------|
|           |      | (DHBC [70], Ours-multiview, Ours-oneshot) | Intra AE     | Intra WE    | Inter AE      |
| FAUST [9] | F2P  | 6.52                                      | 14.61        | 6.75        | 11.79         |
|           |      | <b>2.39</b>                               | <b>3.52</b>  | 2.63        | 4.62          |
|           | P2P  | 2.41                                      | 4.40         | <b>2.58</b> | <b>4.52</b>   |
|           |      | 9.51                                      | 22.46        | 9.81        | 32.27         |
| SCAPE [4] | F2P  | <b>3.40</b>                               | <b>11.06</b> | <b>3.45</b> | <b>11.166</b> |
|           |      | 3.96                                      | 20.24        | 3.74        | 16.06         |
|           | P2P  | 4.33                                      | 11.76        | -           | -             |
|           |      | <b>3.11</b>                               | <b>4.92</b>  | -           | -             |
|           | F2P  | 3.23                                      | 7.29         | -           | -             |
|           |      | 17.19                                     | 38.21        | -           | -             |
|           | P2P  | <b>8.32</b>                               | <b>27.76</b> | -           | -             |
|           |      | 8.99                                      | 36.94        | -           | -             |

Table 5. Normalized mean geodesic error (in %) using the dataset by Ren et al. [58]. This experiment validates the cases with various training and test sets. F and S denote FAUST and SCAPE, respectively. From top to bottom, the previous methods compared in this table are classified into three categories; axiomatic (no training), unsupervised learning, and supervised learning. The numbers except for ours are courtesy of [22].

| Method               | Test / Training dataset |            |            |            |
|----------------------|-------------------------|------------|------------|------------|
|                      | F / F                   | S / S      | F / S      | S / F      |
| BCICP [58]           | 15.0                    | 16.0       | -          | -          |
| ZoomOut [46]         | 6.1                     | 7.5        | -          | -          |
| Smooth Shells [21]   | 6.1                     | 7.5        | -          | -          |
| SurFMNet+ICP [59]    | 7.4                     | 6.1        | 23.0       | 19.0       |
| Unsup FMNet+pmf [31] | 5.7                     | 10.0       | 9.3        | 12.0       |
| Deep Shells [22]     | <b>1.7</b>              | <b>2.5</b> | <b>2.7</b> | 5.4        |
| FMNet+pmf [42]       | 5.9                     | 6.3        | 14.0       | 11.0       |
| 3D-CODED [28]        | 2.5                     | 31.0       | 33.0       | 31.0       |
| GeoFMNet+zo [18]     | 1.9                     | 3.0        | 4.3        | 9.2        |
| Ours-oneshot         | 2.1                     | 2.7        | 4.9        | <b>3.1</b> |
| Ours-multiview       | <b>1.5</b>              | <b>2.0</b> | 4.1        | <b>1.9</b> |

Note that some of the approaches [42, 28, 18] in Table 5 use the ground-truth correspondences for training. In contrast, we only use skinned synthetic shapes for training and augment the training set using Mixamo poses. It is clearly beneficial to utilize available human motions because animated models are readily obtainable.

### 4.3. Other results

**Unseen data.** We show visual results of our method on two different types of *unseen datasets* - our real depth sequences captured with an Azure Kinect DK [1] and SHREC14 [56] dataset. For this experiment, we use the training dataset described in Sec. 3.3. The upper part of Figure 5 shows the results on unseen depth data. The depth sequences capture single or multiple people with various motions. Although

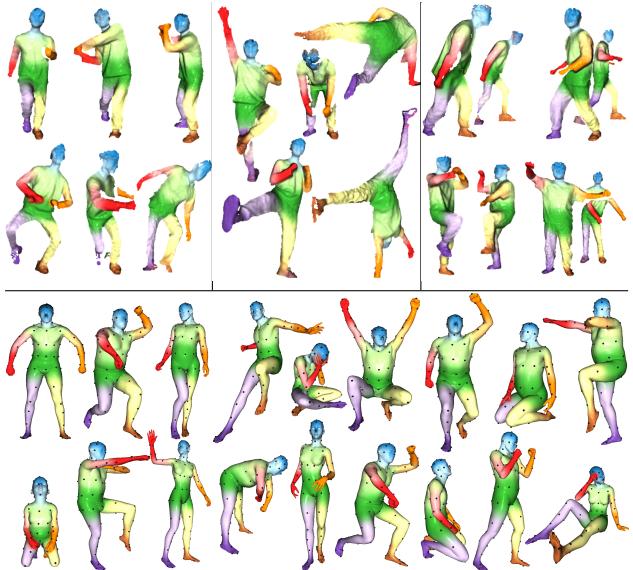


Figure 5. Estimated virtual markers in various unseen datasets. (upper part) From left to right, the subfigures show ordinary to challenging cases. The results show that our method is robust to occlusions and diverse poses, and multiple instances. (lower part) Estimated virtual markers on SHREC14 [56] dataset. The black dots on the surface indicate estimated sparse virtual markers. To decide the sparse virtual markers, we select the vertex with the highest confidence for each label. The results are consistent with various shapes and motions. Note that we used the training dataset described in Sec. 3.3, and these examples are not exposed for the training.

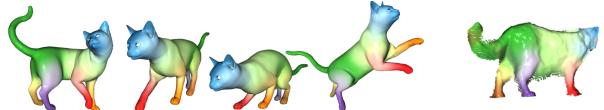


Figure 6. Example of a cat model. We train the network using 3D cat models obtained by SMAL [76] with proper random pose data. We then test the network on cat models in TOSCA [12] (the left four results) and a depth map of a real cat (on the right). The reliable results indicate the wide usage of our deep virtual marker.

real depth images are noisy, the results demonstrate that our method is reliable and robust. The lower part of Figure 5 showcases color-coded models with our inferred virtual markers. The results show that our method works well on unseen, various dynamic models.

**Cats.** Figure 6 shows an example for a cat. It shows reasonable predictions on models in TOSCA [12] and a real depth image.

### 4.4. Applications

**Non-rigid registration.** Our approach predicts dense markers, so the correspondence can help to solve non-rigid registration. One example is shown in Figure 7. For non-rigid surface registration, we compare ours with Li et al.'s approach [39] that is a popular baseline for performance cap-

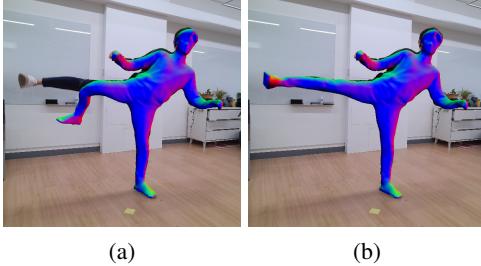


Figure 7. Non-rigid registration. (a) A large misalignment around a leg due to fast motion cannot be recovered by a local alignment based approach [39]. (b) Correspondences from our method can successfully remedy such a large misalignment.



Figure 8. Texture transfer from a textured mesh to another mesh (first and second row) or a depth map (third row). In the mesh-to-mesh case, we transfer texture coordinates and apply texture mapping to the mesh. In the mesh-to-depth map case, we directly transfer the texture colors. The results show that our deep virtual markers produce sufficiently adequate correspondences even with clothed humans, significant shape differences, or depth noise.

ture approaches [40, 20, 19, 40, 73]. Li et al.’s approach [39] progressively deforms a source mesh model to fit to the target one. If pose difference between two mesh models is large, it tends to get stuck at local minima since the method uses neighbor search for constructing vertex correspondences, as shown in Figure 7 (a). The misalignment issue can be alleviated by using dense vertex correspondences estimated by our approach, as shown in Figure 7 (b).

**Texture transfer.** As another application, we present texture transfer. By using dense correspondences obtained by our deep virtual markers, the texture in a source mesh model



Figure 9. Failure cases of our approach. The cases of loose clothes, body-to-object interactions, and intra-body close interactions are shown from left to right.

can be instantly mapped to the target mesh model. Similarly, texture transfer can be performed between a full mesh and a depth map, as our deep virtual markers can handle a variety of 3D inputs in a consistent way. Figure 8 shows examples.

#### 4.5. Limitations

Although our approach is validated with many standard 3D human models from FAUST [9], SCAPE [4], and SHREC14 [56], our virtual markers may not deal with humans wearing highly loose clothes, such as long skirts and coats. In addition, our method is not very accurate for the cases of body-to-object and intra-body close interactions, as shown in Figure 9. We may consider a data-driven solution, but our current pipeline for preparing a dataset requires human annotations, which is not directly applicable to people wearing challenging clothes. Human models interacting with objects are not tested as well. Future work would be to tackle these issues.

#### 5. Conclusions

In this paper, we proposed *deep virtual markers*, a methodology that infers dense virtual marker labels for articulated 3D objects. Our approach can handle various types of input, including meshes, point clouds, and depth images. We proposed an effective method to annotate dense virtual markers on 3D meshes to build a training dataset. Our framework utilizes a fully convolutional neural network as a soft label classifier, and our oneshot model applies single feed-forward operation to instantly assign dense markers ( $0.05 \sim 0.07$  seconds). Compared to state-of-the-art methods, we showed a favorable performance in finding correspondences. Lastly, we demonstrated practical applications such as texture transfer and non-rigid surface registration. Our future work includes dealing with humans wearing highly loose clothes, such as long skirts and coats.

**Acknowledgements** This work was supported by the IITP grants (SW Star Lab: 2015-0-00174 and Artificial Intelligence Graduate School Program (POSTECH): 2019-0-01906) and the NRF grant (NRF-2020R1C1C1015260) from the Ministry of Science and ICT (MSIT), and the KOCCA grant (R2021040136) from the Ministry of Culture, Sports, and Tourism (MCST).

## References

- [1] Azure kinect dk. <https://azure.microsoft.com/en-us/services/kinect-dk/>. 7
- [2] Mixamo-3d animation online services, 3d characters, and character rigging. <https://www.mixamo.com/>. 2, 4
- [3] Renderpeople-scanned 3d people models provider. <http://renderpeople.com>. 4
- [4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005. 2, 6, 7, 8
- [5] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011. 2
- [6] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007. 3, 4
- [7] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. *Advances in Neural Information Processing Systems*, 33, 2020. 3
- [8] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision*, pages 561–578. Springer, 2016. 1
- [9] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE. 2, 4, 5, 6, 7, 8
- [10] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3197–3205, 2016. 2
- [11] Chris Bregler. Motion capture technology for entertainment [in the spotlight]. *IEEE Signal Processing Magazine*, 24(6):160–158, 2007. 1
- [12] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 7
- [13] Jacky CP Chan, Howard Leung, Jeff KT Tang, and Taku Komura. A virtual reality dance training system using motion capture technology. *IEEE transactions on learning technologies*, 4(2):187–195, 2010. 1
- [14] Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2039–2047, 2015. 6
- [15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 5
- [16] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019. 2, 3
- [17] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *arXiv preprint arXiv:1908.04725*, 2019. 2, 6
- [18] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020. 2, 3, 7
- [19] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. 36(6), Nov. 2017. 8
- [20] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016. 8
- [21] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020. 2, 3, 6, 7
- [22] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *arXiv preprint arXiv:2010.15261*, 2020. 2, 3, 7
- [23] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2
- [24] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *European conference on computer vision*, pages 224–237. Springer, 2004. 2
- [25] Zhongpai Gao, Guangtao Zhai, Juyong Zhang, Junchi Yan, Yiyang Yang, and Xiaokang Yang. Learning local neighboring structure for robust 3d shape representation. *arXiv e-prints*, pages arXiv–2004, 2020. 2, 6
- [26] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *European Conference on Computer Vision*, pages 36–52. Springer, 2020. 2, 3, 6
- [27] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 2

- [28] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 2, 3, 6, 7
- [29] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2):1–17, 2019. 1
- [30] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063, 2020. 1
- [31] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019. 2, 6, 7
- [32] Antonio Hernández-Vela, Nadezhda Zlateva, Alexander Marinov, Miguel Reyes, Petia Radeva, Dimo Dimov, and Sergio Escalera. Graph cuts optimization for multi-limb human segmentation in depth maps. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 726–732. IEEE, 2012. 2
- [33] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999. 2
- [34] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1
- [35] 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. 4
- [36] Vladimir Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4), July 2011. 6
- [37] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6050–6059, 2017. 2
- [38] Chun-Liang Li, Tomas Simon, Jason Saragih, Barnabás Póczos, and Yaser Sheikh. Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11967–11976, 2019. 2, 6
- [39] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2009)*, 28(5), December 2009. 7, 8
- [40] Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T Barron, and Gleb Gusev. 3d self-portraits. *ACM Transactions on Graphics (TOG)*, 32(6):1–9, 2013. 8
- [41] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2
- [42] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017. 2, 3, 6, 7
- [43] Riccardo Marin, Simone Melzi, Emanuele Rodola, and Umberto Castellani. Farm: Functional automatic registration method for 3d human bodies. In *Computer Graphics Forum*, volume 39, pages 160–173. Wiley Online Library, 2020. 2, 3, 6
- [44] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3
- [45] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 2
- [46] Simone Melzi, Jing Ren, Emanuele Rodola, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019. 7
- [47] Alberto Menache. *Understanding motion capture for computer animation and video games*. Morgan kaufmann, 2000. 1
- [48] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001. 1
- [49] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2
- [50] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12395, 2020. 2
- [51] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 1
- [52] K. Nishi and J. Miura. Generation of human depth images with body part labels for complex human pose recognition. *Pattern Recognition*, 71:402–413, 2017. 2, 3, 4
- [53] Gabriel L Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. Deep learning for human part discovery in images. In *2016 IEEE International conference on robotics and automation (ICRA)*, pages 1634–1641. IEEE, 2016. 2

- [54] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 3
- [55] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, SIGGRAPH ’17, New York, NY, USA, 2017. Association for Computing Machinery. 3
- [56] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, S. Bu, U. Castellani, S. Cheng, V. Garro, A. Giachetti, A. Godil, J. Han, H. Johan, L. Lai, B. Li, C. Li, H. Li, R. Litman, X. Liu, Z. Liu, Y. Lu, A. Tatsuma, and J. Ye. SHREC’14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, EG 3DOR’14. Eurographics Association, 2014. 7, 8
- [57] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4332–4341, 2019. 2, 6
- [58] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)*, 37(6):1–16, 2018. 6, 7
- [59] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1617–1627, 2019. 2, 6, 7
- [60] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009. 2
- [61] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 2, 3
- [62] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. 39(6), 2020. 1
- [63] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011. 2, 3, 4
- [64] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009. 2
- [65] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3
- [66] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 1, 2
- [67] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2
- [68] Jianyu Wang and Alan L Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1788–1797, 2015. 2
- [69] Yiqun Wang, Jing Ren, Dong-Ming Yan, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. Mgcn: descriptor learning using multiscale gcns. *ACM Transactions on Graphics (TOG)*, 39(4):122–1, 2020. 2
- [70] Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016. 2, 6, 7
- [71] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6769–6778, 2017. 1, 2
- [72] 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. 3
- [73] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Body-fusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 910–919, 2017. 8
- [74] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7287–7296, 2018. 4
- [75] Silvia Zuffi and Michael J Black. The stitched puppet: A graphical model of 3d human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3546, 2015. 3, 6
- [76] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 5, 7