Deep Virtual Markers for Articulated 3D Shapes
Supplementary Material

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

POSTECH

This supplement provides a detailed description of our network architecture and detailed information on the sparse marker annotation.

1. Network

Architecture

We adopted a U-shaped residual network to classify input 3D model points into dense labels [2]. The detailed description (e.g., the numbers of convolution and deconvolution layers, kernel and stride sizes, etc.) of the network is the same as the model named 'Mink16UNet34C'[1], except for the voxel size that is 1cm in our case.

Training

Our network uses the multi-class cross-entropy loss function (Eq. (2)). To minimize the loss function, we used the stochastic gradient descent (SGD) optimizer. At each iteration for training, we construct a batch of size B, where the batch consists of annotated 3D data selected from our augmented training dataset (Sec. 3.3). In the batch, the memory usage of each partial or full mesh is about 200MB or 500MB, respectively. NVIDIA Titan RTX and Quadro 8000 GPU's memory capacities are 24 and 48 GB, respectively, and we set a suitable batch size by considering the memory capacity. We perform the procedure maximally N times unless the training meets the condition for the early stopping.

(1) ***Ours-multiview.*** The network is trained using only annotated partial meshes from scratch. Batch size B and maximum iteration N are set to 60 and 60000, respectively.

(2) ***Ours-oneshot.*** For this case, we finetuned the network trained for the multiview approach. We use annotated full meshes as well as partial ones. The ratio of full meshes to partial ones in a single batch is 9:1. B and N are set to 50 and 10000, respectively.

*Joint corresponding authors.

[1] https://github.com/chrischoy/SpatioTemporalSegmentation

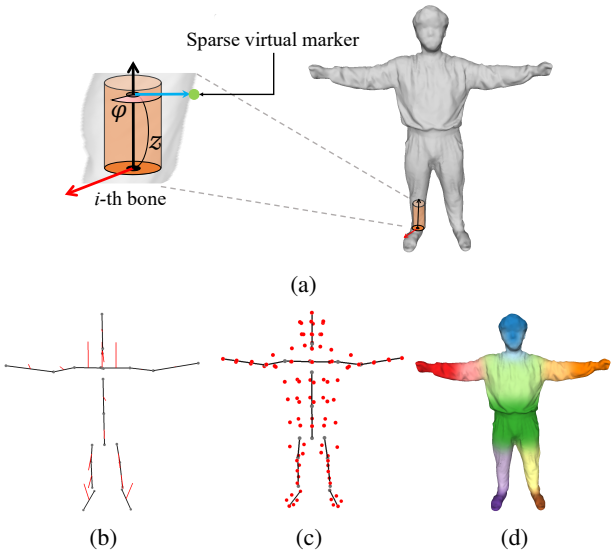

Figure 1. (a) A sparse virtual marker defined in the local cylindrical coordinates of a lower leg. The green arrow is the ray used to sample the marker. (b) Bones (black) and polar axes (red) of the local cylindrical systems. (c) Skeleton and sparse markers were sampled on the human model's surface. (d) Colored human models with dense virtual markers.

2. Sparse markers sampling

To obtain sparse markers, we firstly require experienced users to pick the 3D points, which construct the bone and polar axis of each bone's cylindrical coordinate system (ρ, φ, z), as shown in Figure 1 (b). Positions of sparse markers associated with the i-th bone are specified by only two variables (φ, z). The sampling number of sparse markers varies every bone because parts associated with bones have different circumferences. The detailed information of virtual marker coordinates is shown in Table 1.

We enumerate a total of 33 models and use them for training our network. Sparse markers for respective models are shown in Figure 2, where models are color-coded with dense virtual markers. In Figure 3, we show augmentation of 3D models with Mixamo motions [1] used for training.

Table 1. Coordinates of sparse markers associated with each bone. z^k denotes the coordinates normalized by a bone length, and the unit of φ^k is degree. Here, 'R' and 'L' stand for right and left, respectively.

Body part name	$\{\varphi^k\} \times \{z^k\}$	Body part name	$\{\varphi^k\} \times \{z^k\}$
Head	$\{0, 60, 120, 180, 240, 300\} \times \{0.3, 0.7\}$	Upper Body	$\{0, 45, 90, 135, 180, 225, 270, 315\} \times \{0.25, 0.8\}$
Head_end	$\{0\} \times \{0.0\}$	Lower Body	$\{0, 45, 90, 135, 180, 225, 270, 315\} \times \{0.5\}$
Neck	$\{0, 90, 180, 270\} \times \{0.6\}$	Upper R_Leg	$\{90, 210, 330\} \times \{0.5\}$
Thorax + Upper back	$\{0\} \times \{0., 1.\}$	Lower R_Leg	$\{0, 180\} \times \{0.0, 0.4, 0.8\}$
R_Shoulder	$\{0, 90, 270\} \times \{0.25\}$	Upper L_Leg	$\{-90, -210, -330\} \times \{0.5\}$
L_Shoulder	$\{0, 90, 270\} \times \{0.25\}$	Lower L_Leg	$\{0, 180\} \times \{0.0, 0.4, 0.8\}$
Upper R_Arm	$\{90, 210, 330\} \times \{0.5\}$	R_Foot	$\{0, 90, 180, 270\} \times \{0.65\}$
Lower R_Arm	$\{0, 180\} \times \{0.0, 0.35, 0.7\}$	R_Foot_start	$\{0\} \times \{0.0\}$
Lower R_Arm_end	$\{0\} \times \{0.0\}$	R_Foot_end	$\{0\} \times \{0.0\}$
Upper L_Arm	$\{-90, -210, -330\} \times \{0.5\}$	L_Foot	$\{0, 90, 180, 270\} \times \{0.65\}$
Lower L_Arm	$\{0, 180\} \times \{0.0, 0.35, 0.7\}$	L_Foot_start	$\{0\} \times \{0.0\}$
Lower L_Arm_end	$\{0\} \times \{0.0\}$	L_Foot_end	$\{0\} \times \{0.0\}$

References

[1] Mixamo-3d animation online services, 3d characters, and character rigging. https://www.mixamo.com/. 1, 4

[2] 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

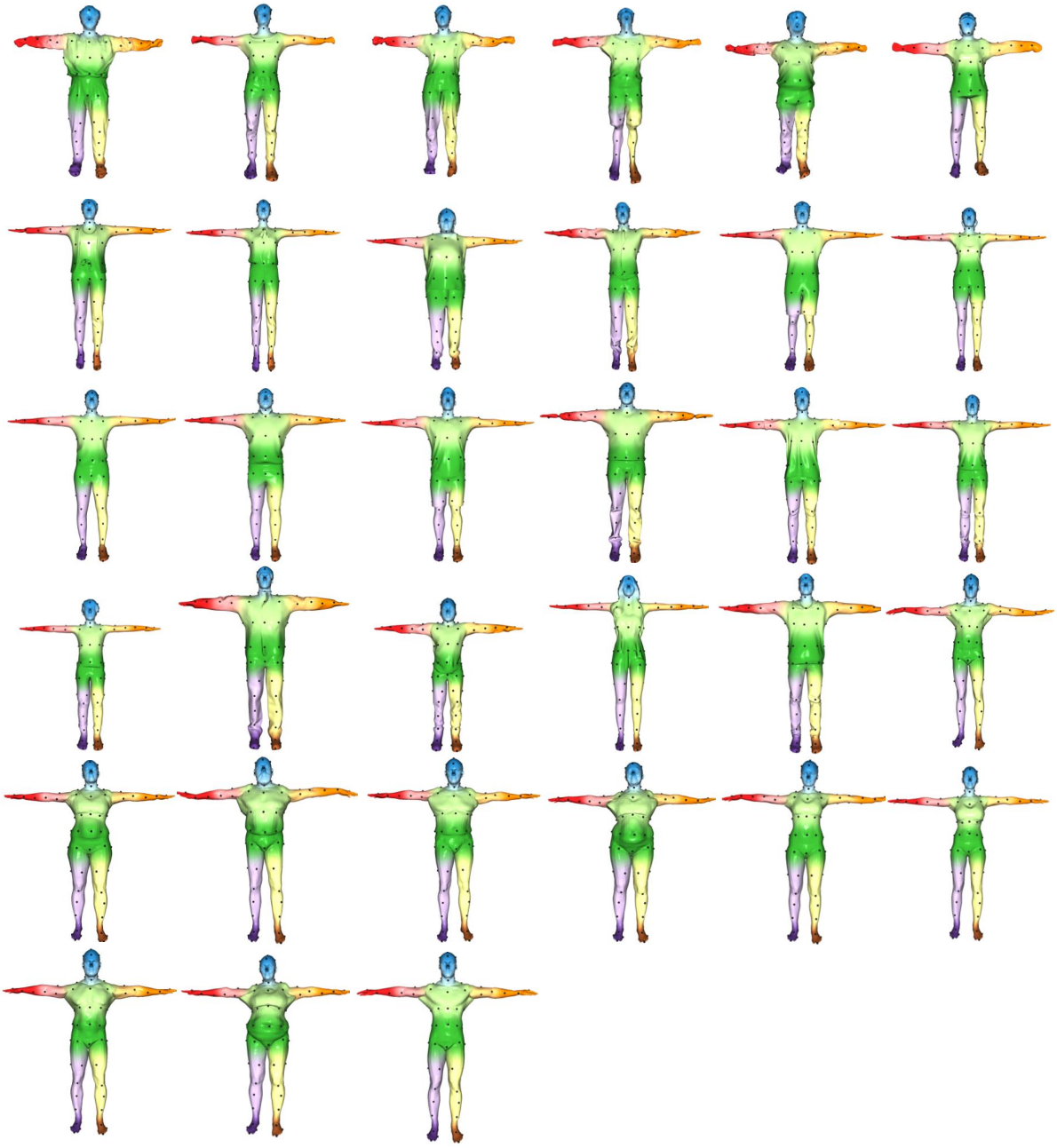

Figure 2. Total 33 human models (T-pose) in our training dataset. Respective models are color-coded using the ground truth soft labels, and sparse virtual markers (black dots) are placed on the models.

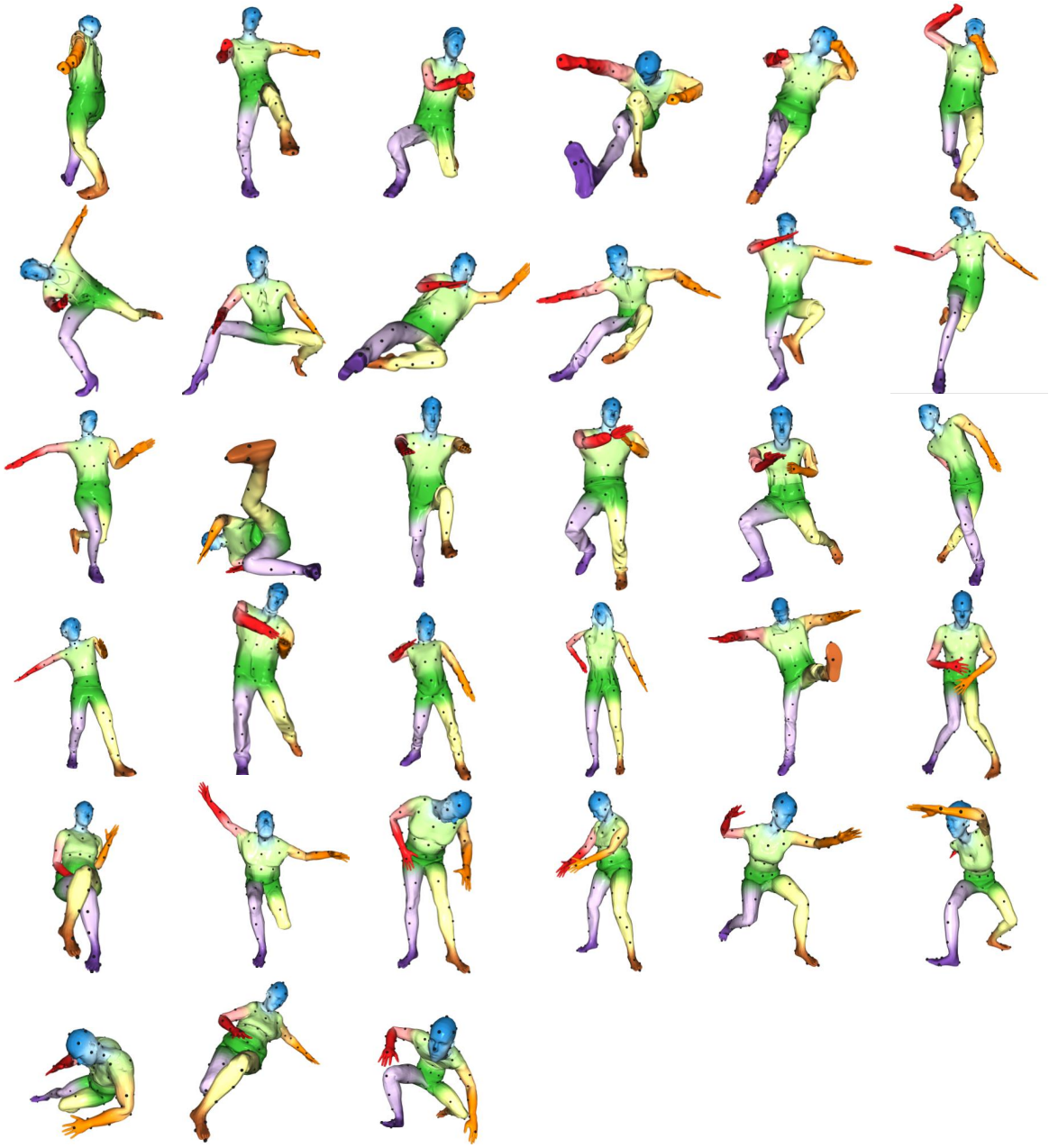

Figure 3. Various poses of 3D mesh models in the training set. Each model takes a pose captured when the model is animated with a Mixamo motion [1].