Appendix

Mingjia Chen[1], Changbo Wang[1], and Ligang Liu[2]

[1]East China Normal University
[2]University of Science and Technology of China

A Feature Embedding

We use t-SNE technique to embed representations of different human models into a 2D space (perplexity = 30 by default). Figure 1 shows a two-dimensional visualization of features on several models (three different humans, each person has twenty different poses). Each dot represents a different pose of a human model, with colour indicating human identity. We can see that shapes with different appearances are clearly separated, and shapes with the same appearances but in different poses are also separated as far as possible. Note that these representations are generated under five observed views.

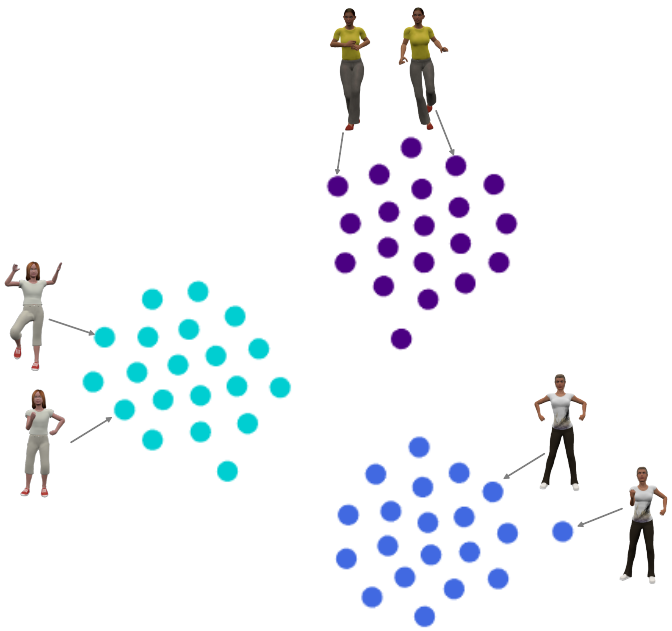

Figure 1: 2D visualization of the feature space using t-SNE. Each dot corresponds to a different pose of a human model, with colour indicating human identity.

B Comparison to DeepVoxels

Figure 2: We show a qualitative comparison of ground truth (first raw), DeepVoxels [2](center column) and our network (last raw) on DeepVoxels's testing dataset. We recommend to zoom in the figure to see the detailed performance.

C Input Configuration

Another concern that may raise is the placing of input viewpoints. We conduct an additional experiment to show the effectiveness of our method with randomly placed cameras.

As discussed in Section 3, all cameras are distributed at the circle of radius d centered at the centroid of shapes ($d = 0.375$ in experiments). And this circle is perpendicular to the vertical axis of shapes. In this experiment, we configure N randomly distributed cameras on this circle, and N is randomly chosen between 2 and 5. We use one actor (C8, which is shown in the supplementary video), 3 performances per actor and 200 consecutive 3D shapes (frames) per performance to construct the dataset. We create 400 training scenes and 200 testing scenes using this method. The training and testing datasets are disjoint. Other related settings of this experiment are same as discussed in the paper. As shown in Figure 3, our method is robust and effective for randomly distributed cameras.

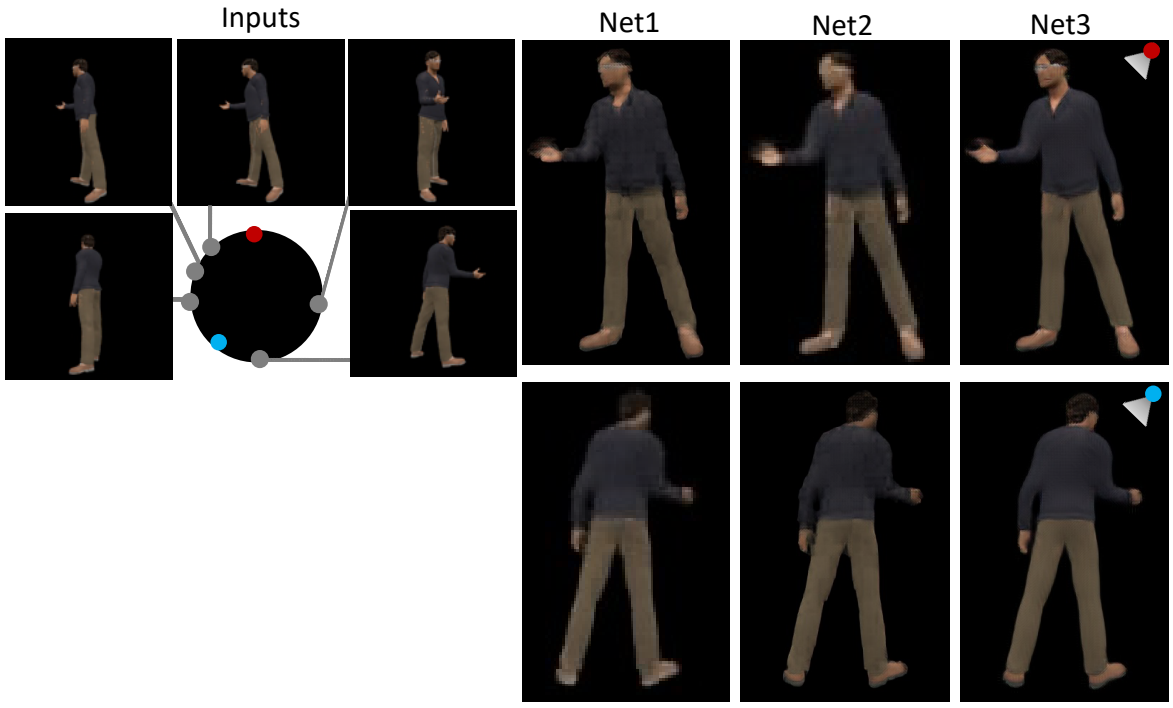

Figure 3: Synthesized results with randomly placed (input) cameras. The input viewpoints are shown in gray. Net1: the generative query network. Net2: the generative query network and the generative adversarial network, without fine-tuning. Net3: full architecture with fine-tuning. Net1, Net2 and Net3 have also been detailed in Ablation Study of Section 6 (in the formal paper). We show two generated viewpoints, whose viewing directions are marked with corresponding colors.

D Comparison to Zeng et al. [1]

We also compare our method with Zeng et al. [1] on their datasets. For a fair comparison, we retrain our network on this dataset. Figure 4 shows the qualitative and quantitative results. All synthesized results are rendered with 512×512 resolution. It can be seen that our method generates images of higher visual quality.

Figure 4: Comparisons with [Zeng et al. 2018] using their dataset. We show the results with corresponding MSEs and SSIMs (bottom). For each criteria, ↑ means the larger the better and ↓ means the smaller the better. We can see that our network shows better performance than [Zeng et al. 2018].

Reference

[1] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 336–354, 2018.

[2] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2437–2446, 2019.