Real-Time Neural Character Rendering with Pose-Guided Multiplane Images (Supplementary Material)

Hao Ouyang[1], Bo Zhang[2], Pan Zhang[2], Hao Yang[2], Jiaolong Yang[2], Dong Chen[2], Qifeng Chen[1], and Fang Wen[2]

[1] Hong Kong University of Science and Technology
[2] Microsoft Research Asia

1 Implementation Details

1.1 Network structure

Our network structure adopts the encoder-decoder structure following U-Net [5], which has been proved very effective for conditional image generation. The encoder consists of a series of residual blocks, gradually reducing the feature spatial size and doubling the number of channels. From the latent feature, the decoder recovers the feature spatial size with skip connection from the features of the encoder. The number of features of the first block is 64 and we adopt 5 blocks for the encoder or decoder respectively. The final layer of the decoder is a 1×1 convolutional layer that outputs the multiplane images (MPIs). We adopt a texture sharing strategy following Nex [7] to achieve a more compact MPI, where every 4 plane shares the same RGB textures. Specifically, the network outputs 240 channels, where the first 192 channels represent the alpha channels and the remaining 48 channels are the RGB textures.

Also, we introduce a small U-Net structure where we make the following modifications: 1) Reduce the number of feature channels to 1/3 of the baseline. 2) Remove one residual block for encoder and decoder respectively. As verified in the quantitative study, this smaller network only introduces minor quality degradation but enables real-time rendering.

1.2 Data preprocess

After we capture the multi-camera video sequences, we perform the following data processing for training

Synchronization Our pose-guided MPI synthesis requires temporally synchronized driving frame and multi-views. We use audio to synchronize the videos from multiple cameras. We utilize a commercial software, Adobe Premiere, to synchronize the videos according to their audio tracks, which offers synchronization accuracy within a few tens milliseconds to five milliseconds. Because there inevitably exists misalignment, we ask the character to move a little bit slower than usual to reduce the misalignment between frames. Instead of processing the whole video sequence, we divide the videos into overlapping clips and synchronize the video clips, which we find leads to more accurate alignment.

Video clip and frame extraction We capture the character for $1 \sim 3$ minutes and encourage the character to perform diverse poses during the video capture. We subsample the video frames by $3 \sim 10$ fps depending on the clip length so that we obtain roughly 1000 frames (or ~ 200 different human poses) for training. It is noteworthy that our network has the potential to fit more frames and we expect this will lead to improved character modeling. However, leveraging more training frames from the moving capture rig is computationally prohibitive in the SfM stage and we intend to address this in our future work.

Camera pose estimation from SfM We use COLMAP [6] to estimate the camera poses. For all the sequences, we first parse the foreground person using [1] along with mask dilation to exclude the moving regions during the feature matching. For video sequences with less than 1000 frames, we use the exhaustive matcher with the guided feature match and run the standard mapper for the scene construction. When dealing with more than 1000 frames, we use vocabulary tree matching for feature matching. Then we adopt a hierarchical mapper followed by a few iterations of triangulation and bundle adjustment to reconstruct the scene. Since we use smartphone cameras of the same type, we enforce shared camera intrinsics during the COLMAP computation. After the sparse reconstruction, we apply the undistortion operation on the original image based on the estimated distortion parameter. These undistorted images are utilized as the ground truth.

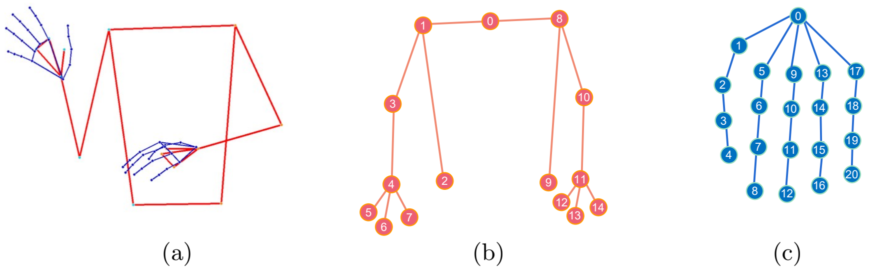
Keypoints Extraction We use MediaPipe [2] to extract the holistic human keypoints for the driving frames. Specifically, we obtain 468, 33 and 42 landmarks for face, body and hands respectively. After the detection, we draw the extracted keypoints with a pre-defined color scheme. As MediaPipe does not support frames with multiple persons, for such cases we crop the input image, detect the keypoints for each person individually and finally stitch the results altogether.

1.3 Motion transfer

When trying to transfer the body pose of the driving character to the source character, we need to keep the direction of driving limbs and the limb length of the source. To achieve this goal, we treat the body landmarks as a tree structure, as shown in Fig. 1(b). The tree root is the midpoint of the left shoulder and right shoulder. When the location of the parent node is known, we can calculate its children nodes. We use t_b^c to denote the children node of the driving body and t_b^p for the parent node of the driving body. The transferred children node is

$$\frac{t_b^c - t_b^p}{l_b^t} * l_b^s, \tag{1}$$

where l_b^t is the length of the driving body limb and l_b^t is the length of the corresponding source limb. And the limb length is determined as the maximum



(a) (b) (c)

Fig. 1. The illustration of the body tree structure and hand tree structure. (a) Captured body and hand pose. (b) The body tree structure where node 0 is the tree root. (c) The hand tree structure where node 0 is the tree root.

limb length of all the driving/source frames. The gesture transfer is similar to body pose transfer, except that the tree root is changed to the wrist. Fig. 1(c) depicts the tree structure for fingers.

2 Generalization Ability of NeRF-based Approaches

We also explore the generalization ability of NeRF-based approaches. To achieve this, we modify the state-of-the-art method Nerfies [3] and HyperNerf [4] with the keypoints as input. Specifically, for Nerfies, instead of learning a latent code for the deformation of each pose, we directly use the body keypoints as the input to learn the deformation. For HyperNeRF, we modify it to a hybrid way, where we expect the input body keypoints to learn the large deformation — leg movement or head pose — relative to the canonical template while the hyper code modulates the canonical space and accounts for small deformation such as expression changes.

We show the results for novel pose combination in Fig. 2. In this example, we want to combine the right hand of the first input with the body and the face with the second input. We can see that HyperNeRF does not fully utilize the information of the body keypoints and fails in the combination. Nerfies is able to generate the coarse new pose but suffers from large distortion and obvious artifacts in the resulting image. The generated pose is also not accurately aligned with the desired combination. As the NeRF-based method is based on the implicit field, we show that the generalization ability is not as good as the proposed method which allows explicit control and shows better generalization ability.

3 More Results

In the next, we provide additional results including the depth visualization and generalization ability experiments. We highly recommend the readers refer to the accompanied videos for more visual results.

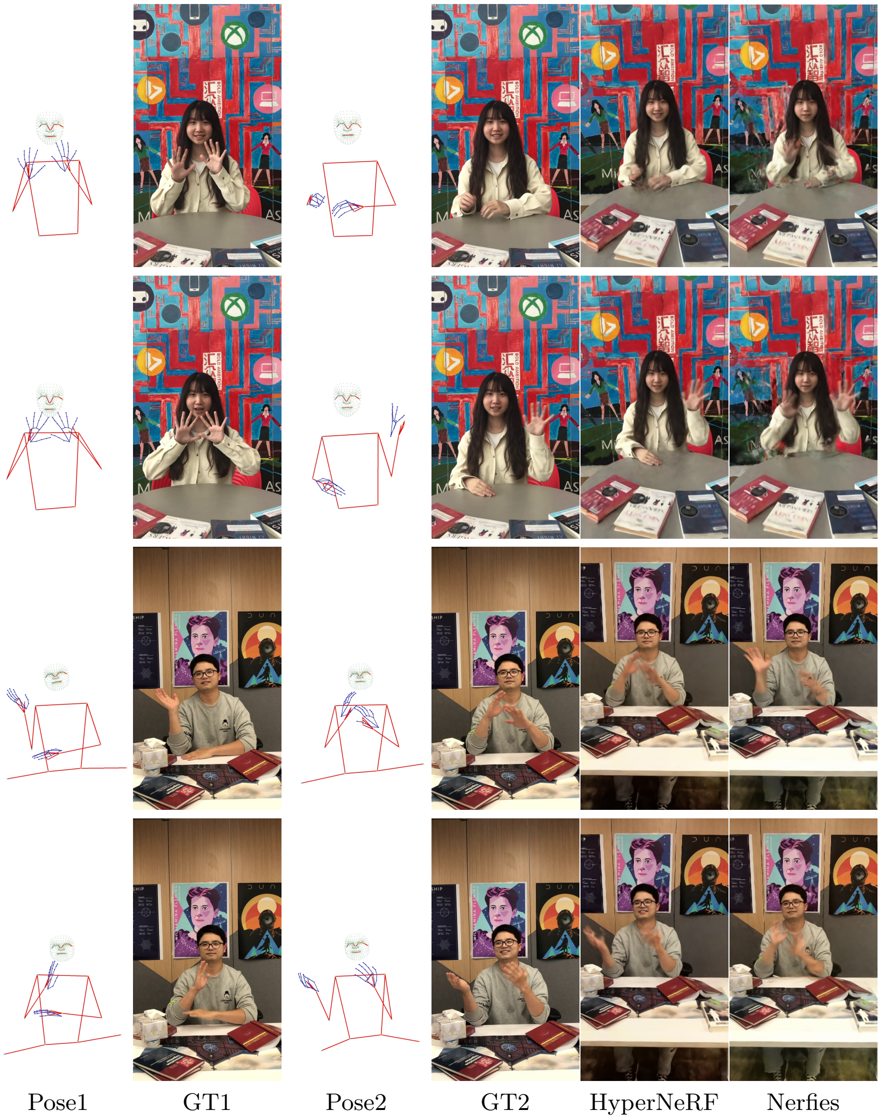


Fig. 2. Novel pose combination using NeRF-based approaches. Due to the implicit modeling of deformation, these methods are good at memorizing the scene but are not friendly to explicit control, thus failing to generalize to unseen poses that are slightly different from the training samples.

Depth visualization By compositing the depth instead of the color texture, we can generate the depth map for each frame as visualized in Fig. 3.

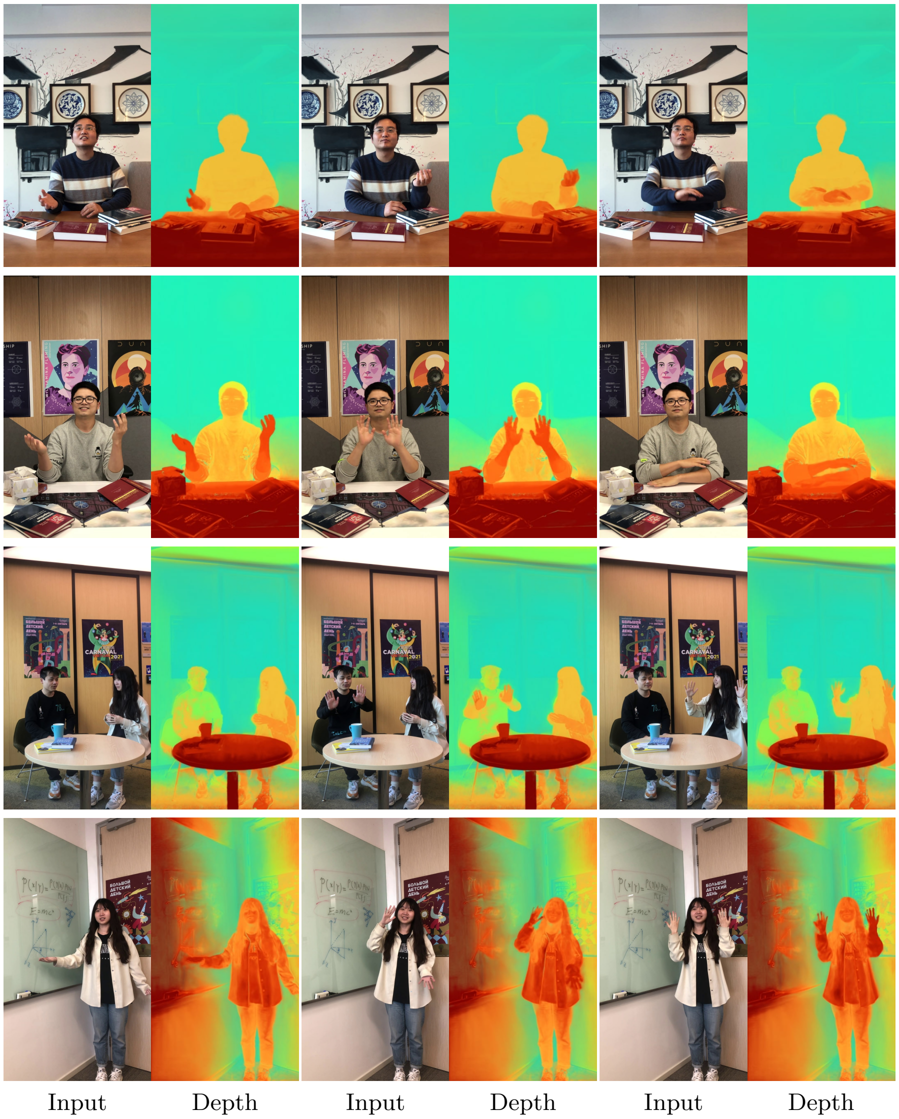


Fig. 3. Depth visualization. Our method can generate a smooth, boundary crisp depth map, showing that the learned MPI does well model the 3D geometry of the scene.

Generalization ability We also include more examples on pose interpolation and extrapolation as in Fig. 4.

More results of motion transfer In Fig. 5 we showcase additional results on motion transfer. Our method is able to animate the character following the driving subject while yielding photo-realistic quality.

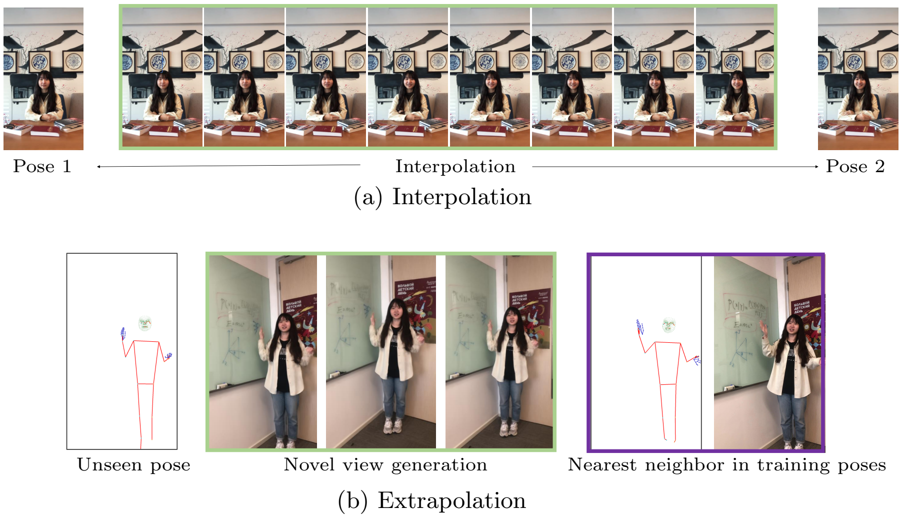


Fig. 4. Generalization study. Comparing to implicit approaches, our method can generalize to unseen poses due to the generative ability of CNNs. Here we show examples on motion interpolation and extrapolation.

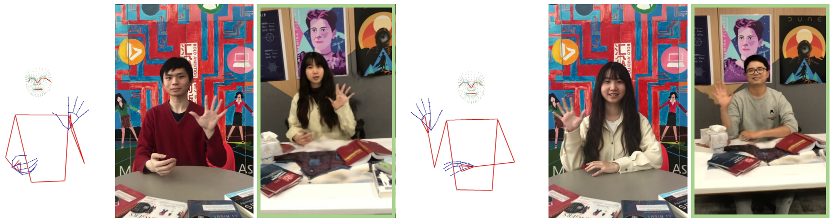


Fig. 5. Additional results on motion transfer.

References

1. Li, P., Xu, Y., Wei, Y., Yang, Y.: Self-correction for human parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
2. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J., et al.: Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172 (2019)
3. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5865–5874 (2021)
4. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. arXiv preprint arXiv:2106.13228 (2021)
5. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

6. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
7. Wizadwongsa, S., Phongthawee, P., Yenphraphai, J., Suwajanakorn, S.: Nex: Real-time view synthesis with neural basis expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8534–8543 (2021)