Real-Time Neural Cloth Deformation using a Compact Latent Space and a Latent Vector Predictor

- Supplementary Material -

Chanhaeng Lee[1,3], Maksym Perepichka[1,3], Saeed Ghorbani[3], Sudhir Mudur[1], Eric Paquette[2], and Tiberiu Popa[1]

[1] Concordia University, Canada
{l_chanha,m_perepi}@live.concordia.ca
{sudhir.mudur,tiberiu.popa}@concordia.ca
[2] Ecole de Technologie Supérieure, Canada
eric.paquette@etsmtl.ca
[3] Ubisoft La Forge, Canada
saeed.ghorbani@ubisoft.com

1 Implementations

The network Φ in our compact latent learning stage consists of the encoder, the processor, the mesh convolution network, and the decoder with the learnable blend shapes.

The encoder and processor are constructed similarly to those in HOOD [1]. The encoder, which has six MLPs, encodes input vertex and edge features into latent features. These input features include garment and body vertex feature vectors, garment edge feature vectors, edge feature vectors from three coarsened graphs of the garment, and features for edges connecting body and garment vertices by their proximity. The processor consists of 15 message-passing steps, each of which includes an MLP for latent vertex features and MLPs for latent edge features. The processor includes two down sampling blocks and two up sampling blocks for its hierarchical message passing. Each MLP in the encoder and processor comprises three linear layers and one normalization layer at the end. The linear layers convert the size of input features into 128, with ReLU activation applied to the outputs of the first two linear layers.

Our mesh convolution network consists of four blocks. Each block contains a convolution layer and a residual layer, denoted as a combination of vcDownConv and vdDownRes, according to Zhou et al. [4]. These blocks require graph sampling information to perform down-samplings with the convolution and residual methods. We selected remaining vertices for the graph sampling information of each garment in the training dataset, using a stride of two and a vertex ring size of two for each down-sampling operation. The number of remaining vertices after the four blocks for each garment normally ranges from 80 to 300. The first three blocks convert input features of 128 dimension into features of 256, 512, and 1024 dimensions, respectively. The last block converts the feature dimension

of 1024 into a final dimension size where the product of the number of remaining vertices and the final feature dimension is around 2048. Therefore, the dimension L of a latent vector $\mathbf{Z}$ is around 2048, depending on the number of the remaining vertices. We used 32 weight bases for each block. Unlike the ELU activation function adopted by Zhou et al. [4], we use the ReLU activation function in our mesh convolution network.

The decoder has an MLP with 3 linear layers, converting the dimension of an input latent vector into 1024, 512, and 512, with ReLU activation applied after each layer. The number D of elements in the array of the learnable blend shapes matrices $\mathbf{D}$ is accordingly 512.

Our latent predictor has an MLP with three linear layers, converting the input features into dimensions of 1024, 1024, and the latent vector size L, respectively. We use ReLU activation for the outputs of the first two layers.

2 Sequence List for Training

We take pose sequences from the AMASS dataset [2] and adopt the sequence list from the VTO dataset [3]. However, there are unavailable sequences for us in the sequence list from VTO dataset. We replace the unavailable sequences (`104_17`, `104_04`, `104_53`, `104_54`, `144_30`, `26_11`, `104_11`) with the other sequences in similar pose categories (`105_36`, `111_23`, `127_04`, `127_20`, `144_32`, `26_10`, `105_11`).

References

1. Grigorev, A., Thomaszewski, B., Black, M.J., Hilliges, O.: HOOD: Hierarchical graphs for generalized modelling of clothing dynamics (2023)
2. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: International Conference on Computer Vision. pp. 5442–5451 (Oct 2019)
3. Santesteban, I., Thuerey, N., Otaduy, M.A., Casas, D.: Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
4. Zhou, Y., Wu, C., Li, Z., Cao, C., Ye, Y., Saragih, J., Li, H., Sheikh, Y.: Fully convolutional mesh autoencoder using efficient spatially varying kernels. Advances in neural information processing systems **33**, 9251–9262 (2020)