3D Gaussian Parametric Head Model
Supplementary Material

Yuelang Xu[1], Lizhen Wang[1], Zerong Zheng[2], Zhaoqi Su[1], and Yebin Liu[1]

[1] Tsinghua University, Beijing, China
[2] NNKosmos, Hangzhou, China

1 Training Details

Before training starts, we first initialize the identity codes and the expression codes following NPHM [1]. For each different identity, we set a different identity code with a dimension of 128. For each different expression, we also set a different 128-dimension expression code. In addition, we enforce a constraint to ensure that the norm of these codes remains below 1.

Besides, we experimentally found that although it is not necessary, jointly optimizing the head pose during the training process can eliminate some minor errors generated during the BFM calibration process, promoting the consistency of the model as the codes change. Consequently, we optimize all the head poses with a very small learning rate 1×10^{-5} throughout the entire training stage of the model.

Next, we give a brief description of the network structure. For the upsampling network $\boldsymbol{\Psi}(\cdot)$, we adopt a very simple U-net [3] structure, with 1 convolutional layer for downsampling and 2 convolutional layers for upsampling. For the MLPs $\boldsymbol{f_{mean}}(\cdot)$, $\boldsymbol{f_{id}}(\cdot)$, $\boldsymbol{f_{exp}}(\cdot)$, $\boldsymbol{f_{col}}(\cdot)$, $\boldsymbol{f_{att}}(\cdot)$, we set the width to 512 with 4 hidden layers. And for the injection MLP $\boldsymbol{f_{inj}}(\cdot)$, we set the width to 512 with only 3 hidden layers. The mesh and the Gaussians are both rendered as 256-resolution images and turned into 512-resolution images through the upsampling network.

With the chosen parameter settings, our model is capable of generating an image in just 31ms (32fps). During training the guiding geometry model, we use 256-resolution tetrahedral grid for extracting the mesh via DMTet. For the optimization, we use an Adam [2] optimizer, and set the learning rate to 1×10^{-4} for the identity codes $\boldsymbol{z^{id}}$, 1×10^{-3} for the expression codes $\boldsymbol{z^{exp}}$, 1×10^{-4} for all the networks $\boldsymbol{f_{inj}}(\cdot)$, $\boldsymbol{f_{mean}}(\cdot)$, $\boldsymbol{f_{id}}(\cdot)$, $\boldsymbol{f_{exp}}(\cdot)$, $\boldsymbol{f_{col}}(\cdot)$, $\boldsymbol{\Psi}(\cdot)$, and 1×10^{-4} for the 3D landmarks $\boldsymbol{P_0}$. We use a batch size of 2, with each batch containing 6 images of a specific expression from a given identity. Training the guiding geometry model requires 2 RTX4090 graphics cards and approximately 3 days.

While training the Gaussian model, we also use an Adam optimizer and set the learning rate: 1×10^{-5} for the identity codes $\boldsymbol{z^{id}}$, 5×10^{-5} for the expression codes $\boldsymbol{z^{exp}}$, 1×10^{-4} for all the networks $\boldsymbol{f_{id}}(\cdot)$, $\boldsymbol{f_{exp}}(\cdot)$, $\boldsymbol{f_{col}}(\cdot)$, $\boldsymbol{f_{att}}(\cdot)$, $\boldsymbol{\Psi}(\cdot)$ and 1×10^{-4} for the 3D landmarks $\boldsymbol{P_0}$. For the mean Gaussian attributes, we set the learning rates as: 1×10^{-5} for the positions $\boldsymbol{X_0}$, 1×10^{-5} for the per-vertex feature

$\boldsymbol{\Gamma_0}$, 3×10^{-5} for the scale $\boldsymbol{S_0}$, 1×10^{-5} for the rotation $\boldsymbol{Q_0}$ and 1×10^{-4} for the opacity $\boldsymbol{A_0}$. We use a batch size of 2, with each batch containing a single image. Following the training of the guiding geometry model, we transfer its parameters to the Gaussian model and continue training on 2 RTX4090 graphics cards for 10 days until convergence is achieved.

References

1. Giebenhain, S., Kirschstein, T., Georgopoulos, M., Rünz, M., Agapito, L., Nießner, M.: Learning neural parametric head models. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2023) 1
2. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017) 1
3. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015) 1