GaussianAvatars: Photorealistic Head Avatars with Rigged 3D Gaussians

Supplementary Material

A. FLAME Tracking

For FLAME [4] tracking, we optimize for per-frame parameters (translation $\boldsymbol{t}_i$, joint poses $\boldsymbol{\theta}_i$, expression $\boldsymbol{\psi}_i$) and shared parameters (shape $\boldsymbol{\beta}$, vertex offset $\Delta\boldsymbol{v}$, and an albedo map A). Our optimization combines a landmark loss, a color loss, and regularization terms.

We use a state-of-the-art facial landmark detector [10] to obtain 68 facial landmarks in 300-W [7] format. Among them, we exclude 17 facial contour landmarks to avoid inconsistency caused by occlusion. We use NVDiffRast [3] to render FLAME meshes and obtain gradients of vertex positions regarding the color loss by texel interpolation for the interior and anti-aliasing on the boundary. For regularization, we apply a Laplacian smoothness term on the vertex offset and temporal smoothness terms on the per-frame parameters.

We optimize all the parameters on the first time step of the video sequence until convergence, then optimize per-frame parameters for 50 iterations for each following time step with the previous one as initialization. Afterward, we conduct global optimization for 30 epochs by randomly sampling time steps to fine-tune all parameters.

We use the 2023 version of FLAME [4] for the revised eye regions. Furthermore, we manually add 168 triangles for teeth to the template mesh of FLAME and make the upper and lower teeth triangles rigid to the neck and jaw joints, respectively. This improves the fidelity of our avatar as shown in Fig. 1.

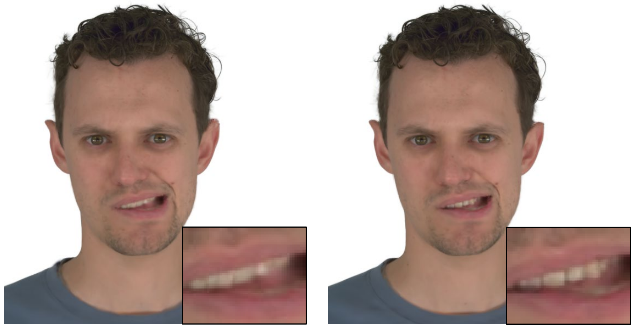


Figure 1. Adding triangles that move rigidly with the head and the jaw helps Gaussian splats to capture teeth details.

B. Dataset Division

We use 11 sequences for each subject from the NeRSemble [2] dataset. Tab. 1 lists concrete sequence types and IDs for different settings. Among the emotion (EMO) and expression (EXP) sequences, we randomly hold out one for self-reenactment evaluation (Tab. 2) and use the rest nine for training.

Setting	Novel View Synthesis & Self-reenactment										Cross-identity Reenactment
Sequence Type	EMO				EXP						FREE
Sequence ID	1	2	3	4	2	3	4	5	8	9	-

Table 1. The types and IDs of sequences for different settings.

Subject ID	074	104	218	253	264	302	304	306	460
Test Sequence	EMO-4	EXP-2	EXP-9	EMO-4	EXP-9	EMO-2	EXP-2	EXP-2	EMO-3

Table 2. The held-out sequence of each subject for self-reenactment evaluation.

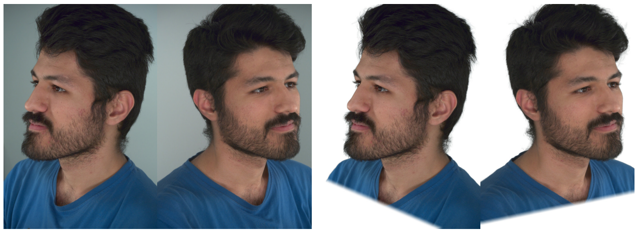


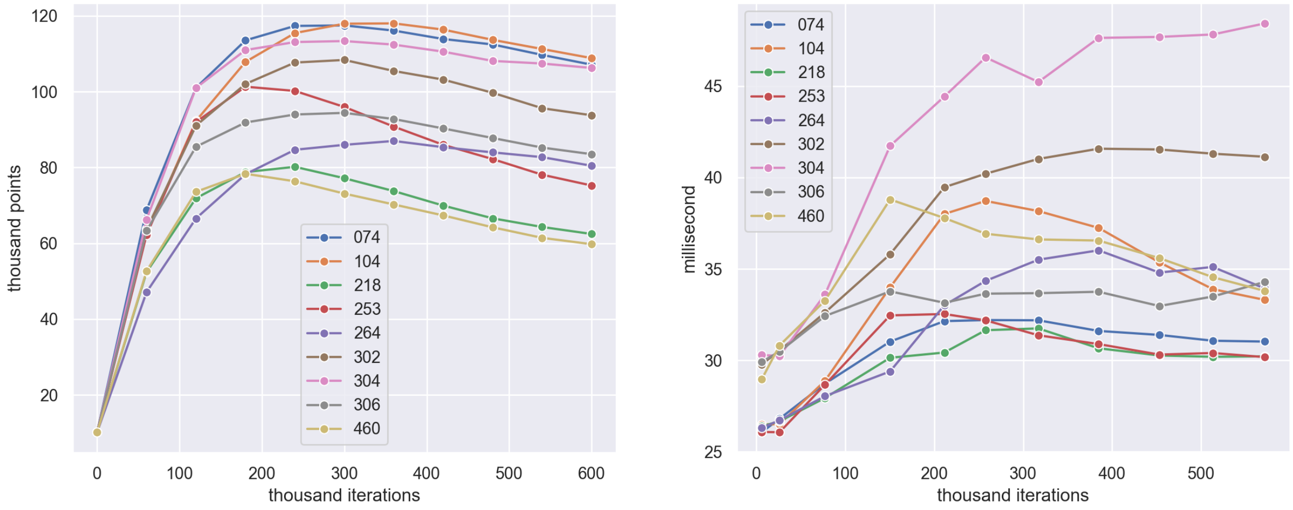
Figure 2. We remove the background and pixels below the shoulder to focus on the head region.

To simplify the pipeline for Gaussian splat optimization, we remove the background of raw images with Background Matting V2 [5]. Additionally, we fit a line across the bottom vertices of each tracked FLAME mesh and project the line to each viewpoint to remove the pixels below. We show an example of pre-processing results in Fig. 2.

C. Computation Efficiency

Our method binds 3D Gaussians to triangles in an efficient way, maintaining high rendering and optimization speed. Given that 3D Gaussians are actively added and pruned during optimization, the running speed of the program also changes.

Efficiency during optimization. We show the evolution of the number of Gaussians and the run-time of an iteration in Fig. 3. According to Fig. 3a, the number of Gaussians grows from 10,144 (that is, the number of triangles in our modified FLAME mesh) to around 100,000 (on average). After this point is reached however, the number of Gaus-



(a) The number of 3D Gaussians throughout the optimization process.

(b) The run-time of an optimization iteration.

Figure 3. The number of 3D Gaussians increases by a factor of around 10 from its starting point for all subjects. After this, the number of 3D Gaussians stops growing. Despite this growth in Gaussians, the run time of each training iteration at most only doubles. Each curve corresponds to a different subject.

sians no longer increases. Thanks to this, longer training times do not mean ever-increasing memory requirements. In fact, our model can fit and be trained on an NVIDIA RTX 2080 Ti Graphics card with 12 gigabytes of VRAM. Moreover, while the number of Gaussians grows by as much as 1000% during training, the run-time of each optimization iteration increases by less than 100% (Fig. 3b) at this peak. This validates the efficiency of the differentiable tile rasterizer [1], which sorts splats before blending and terminates ray marching once zero transmittance is reached. The threshold of our scaling loss (see Section 3 of the main paper) is crucial to this efficiency. Without it, rendering time would increase substantially, as the rasterizer would need to blend many more Gaussians before reaching zero transmittance.

Efficiency during inference. Although our data are processed into a fixed resolution, the optimized model can be rendered in arbitrary resolutions. We show the average rendering FPS in variant resolutions to validate the efficiency of our method to suffice real-time applications.

Resolution	401×225	802×550	1604×1100	3208×2200	6416×4400
FPS	187	187	156	95	36

Table 3. Rendering speed tested on subject #306.

D. Baselines

We compare our method with three state-of-the-art methods for head avatar creation.

INSTA [11] directly warps points according the nearest FLAME [4] mesh triangle. It adds triangles to the mouth, and conditions radiance field queries in the mouth region on the expression code of FLAME to improve the quality of the mouth interior. The loss weight for the mouth region is 40× higher than other regions. It also applies a depth loss on the facial region.

PointAvatar [9] uses a point-based representation, which is closely related to 3D Gaussians. It does not directly rely on the FLAME surface but uses its pose and expression parameters to condition a deformation field. During optimization, it applies a coarse-to-fine strategy to progressively increase the size of the point cloud and decrease the radius of each point. It also uses a post-processing operation to fill holes by applying erosion and dilation to rendered images.

AvatarMAV [8] uses voxel grids for both a canonical radiance field and a set of bases of a motion field. It models deformation by blending the motion bases with the tracked expression vectors of a 3D morphable model [6]. We adapt this method to use our tracked FLAME poses and expressions to ensure fairness.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 2

[2] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view ra-

diance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), 2023. 1

[3] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 1

[4] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 1, 2

[5] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8762–8771, 2021. 1

[6] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009. 2

[7] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016. 1

[8] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2

[9] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21057–21067, 2023. 2

[10] Zhenglin Zhou, Huaxia Li, Hong Liu, Nanyang Wang, Gang Yu, and Rongrong Ji. Star loss: Reducing semantic ambiguity in facial landmark detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15475–15484, 2023. 1

[11] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4574–4584, 2023. 2