SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting

Supplementary Material

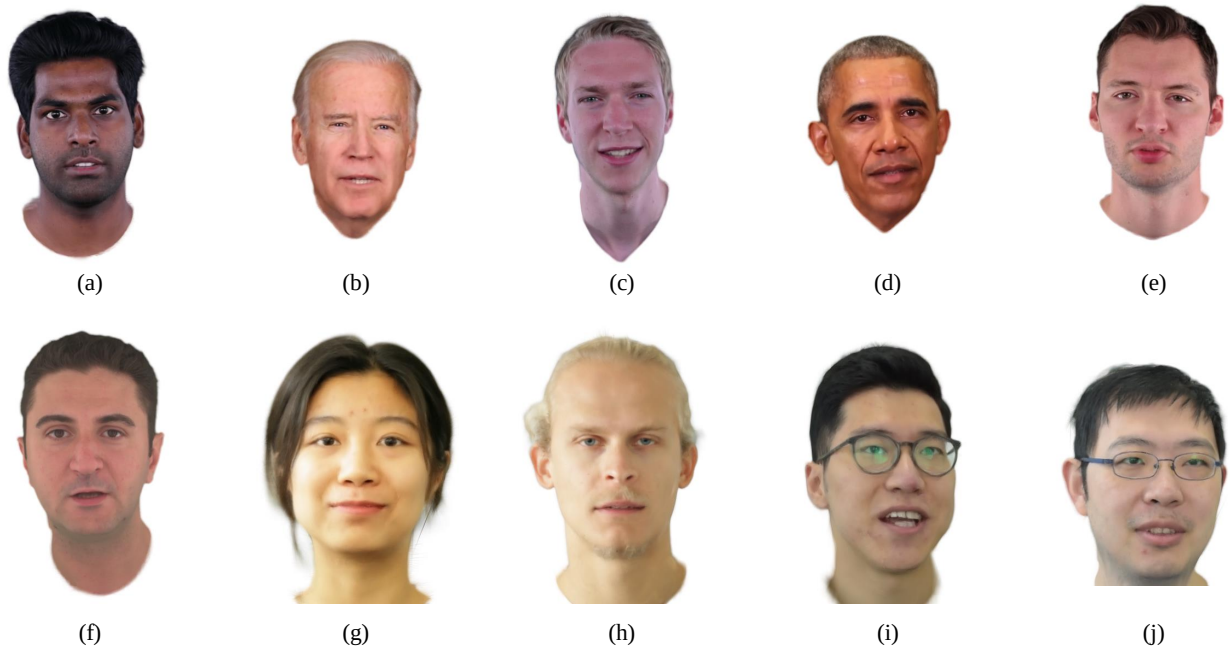


Figure A1. **Dataset for head avatar.** We collected 10 subjects from publicly available datasets for the evaluation of head avatar modeling, with (a–e) from INSTA [7], (f) from NHA [3], (g, h) from IMAvatar [5], and (i, j) from NerFace [2]. We show the rendering results on the testing samples. Our method captures high quality details, for example the light in the eyes, the texture of the hair, and off-surface geometry like the glasses.

In this supplemental document, we elaborate details about the dataset for head avatar in Sec. 1, implementation details in Sec. 2, and additional experimental comparisons in Sec. 3.

1. Dataset

In Figure A1, we show the 10 evaluated subjects that we collected from publicly available datasets, i.e., INSTA [7], NHA [3], IMAvatar [5], and NerFace [2]. The rendering results are from *Ours+FLAME*. Our method show high quality rendering capability with high fidelity details especially in the eyes, hair, and glasses.

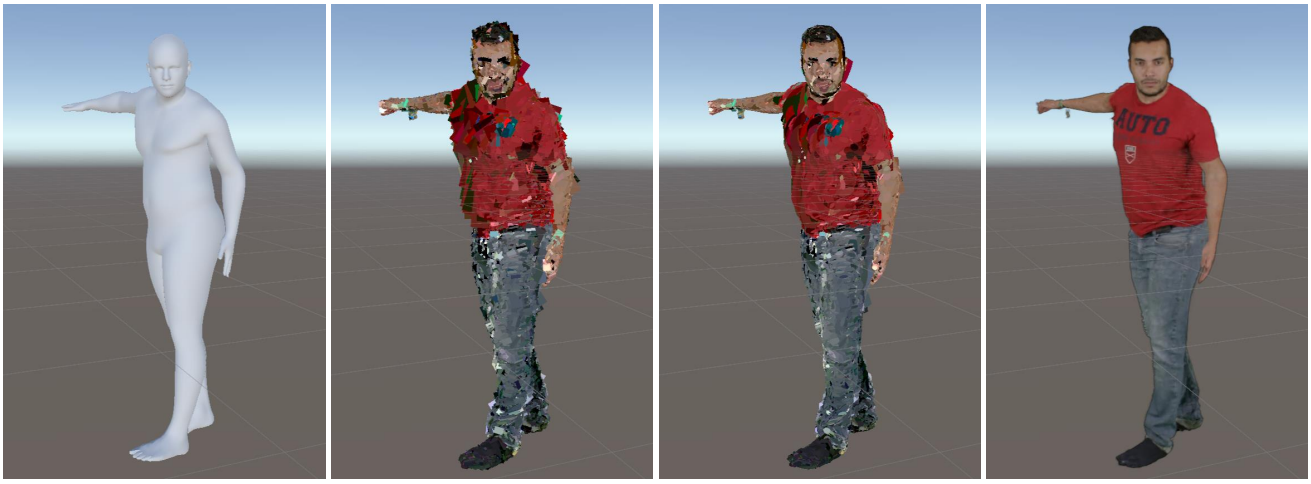
2. Implementation Details

Training. We chose $\lambda_{mse} = 10.0$, $\lambda_l = 0.01$, $\lambda_s = 1.0$, $T_s = 10.0$ and $T_r = 0.008$ all through the experiments. We followed the original implementation of 3D Gaussian Splatting [4] to set the total number of iterations to 30,000 for each subject. Starting from iteration 600, the densify and prune process were conducted every 100 iterations. Every 3000 iterations, the opacity of all the Gaussians were reset to zero. We find this opacity-reset step effective in removing redundant Gaussians. The densify, prune, and opacity-reset process stop at iteration 15,000.

Unity rendering. As described in the main paper, in our Unity implementation, we draw one quad primitive for each Gaussian. The quad primitives are illustrated in Figure A2. Benefiting from our trainable embedding scheme, the embeddings of the Gaussians were efficiently ported to compute shaders for the motion control of the Gaussians, leading to an animatable avatar running over 300 FPS on an NVIDIA RTX 3090 GPU.

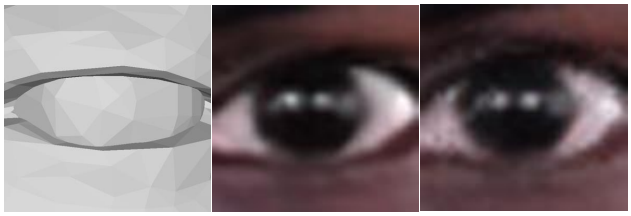
Running time. With our pybind11 implementation, the *walking on triangle* step takes around 3.5 ms. We conduct this step after densifying and pruning. For comparison, *densify-clone* takes 2.5 ms and *densify-split* takes 6 ms.

The whole optimization follows the conversion of the original Gaussian Splatting that the number of total iterations is 30000, and the *densify*, *prune*, and *walking on triangle* steps are performed every 100 iterations.



(a) (b) (c) (d)

Figure A2. **Gaussian Splatting rendering in Unity.** Our Unity implementation of Gaussian Splatting is conducted by drawing one quad primitive for each Gaussian. We show (a) the driving mesh for the current pose, (b) the quad primitive for each Gaussian, (c) the 2D covariance of the Gaussians illustrated by eclipses, and finally (d) the rendering result with α-*blending*.



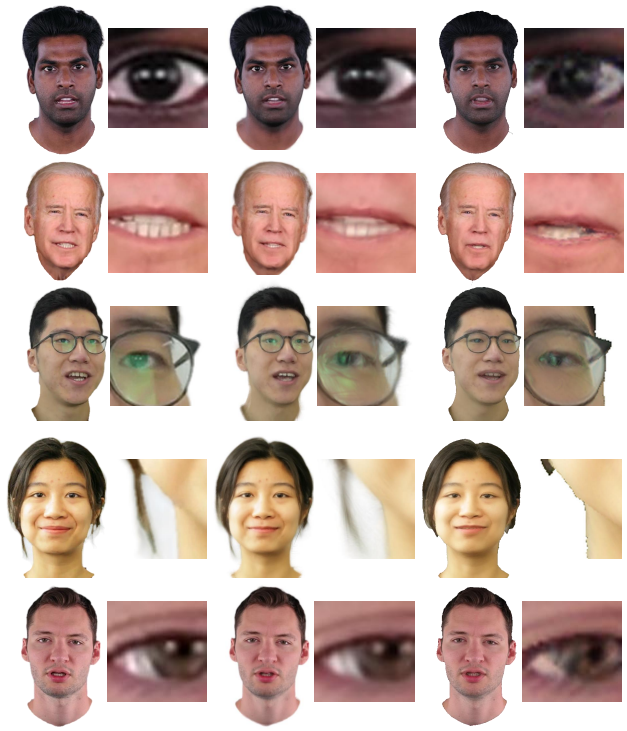
Driving Mesh Ours INSTA

Figure A3. **Comparison with INSTA in the eye region.** INSTA [7] propose to find the nearest triangle when deforming a point in the posed space to the canonical space, causing unstable sampling in the canonical space and strong noise when dealing with complex geometries like the eye. Our embeddings-based motion control of the Gaussians leads to smooth rendering results.

3. Additional Results

Comparison with FLARE. FLARE [1] is a mesh-based avatar modeling approach focusing on relightable avatar reconstructed from monocular videos, which is published very recently. In Table A1, we show comparison with FLARE on our head avatar dataset. FLARE [1] reconstruct accurate geometry and materials of the avatar that our method does not focus on, while the strength of our method is the significant improvement in photometric quality and efficiency in rendering. Qualitative comparison is shown in Figure A4.

Non-ambiguous motion control. One of the key benefits of our method is the non-ambiguous motion control comparing to the backward tracing process of NeRF-based avatar rendering. INSTA [7] propose to simplify this step by finding the nearest triangle for the deformation from the posed space to the canonical space. We show in Figure A3 that this simplification causes significantly more noise when dealing with complex geometries like in the eye region.



Ground Truth Ours+FLAME FLARE

Figure A4. **Comparison with FLARE.** We show qualitative comparison with FLARE [1].

Error map. Due to the limitation of segmentation and head tracking in the pre-processing pipeline. The metrics of pho-

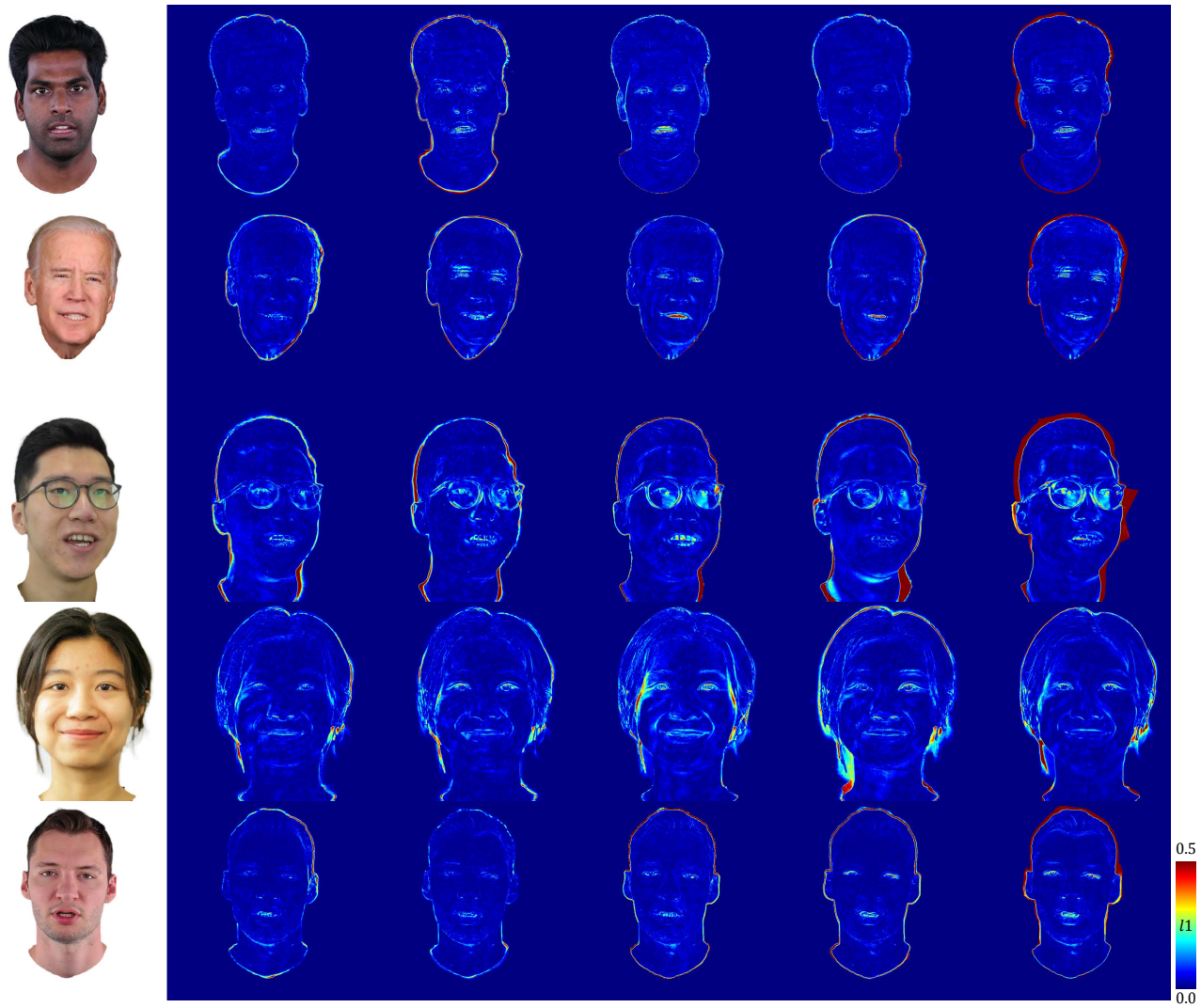


Figure A5. **Heatmaps of $l1$ error.** We show the heatmaps illustrating the $l1$ RGB distance of the rendered images. Our methods and INSTA [7] show overall better quality. The rendering quality of PointAvatar [6] and NHA [3] are limited by their point-based and mesh-based representations respectively.

Method	PSNR↑	SSIM↑	LPIPS↓
FLARE [1]	23.87	0.893	0.129
Ours+FLAME	<u>28.19</u>	**0.931**	<u>0.063</u>
Ours+NHA	**28.86**	**0.931**	**0.060**

Table A1. **Quantitative comparison with FLARE.** We show comparison with the recently published avatar modeling method FLARE [1] on our head avatar dataset.

	Method	PSNR↑	SSIM↑	LPIPS↓
bala	w/o walking	29.91	0.933	0.070
	w/ walking	**30.04**	**0.938**	**0.062**
male-3-casual	w/o walking	32.48	0.979	0.024
	w/ walking	**33.01**	**0.982**	**0.020**

Table A2. **Quantitative ablation on *walking on triangle*.**

tometric error in the main paper was affected by the error mostly in the neck area. We show in Figure A5 the error maps of the evaluated methods. Our methods and INSTA [7] show overall better quality. PointAvatar [6] and NHA [3] both focus on relightable modeling with explicit shape representations, which compromise their performance in terms of pixel-wise metrics.

(a) w/ *walking on triangle*. (b) w/o *walking on triangle*.

Figure A6. **Ablation on *walking on triangle*.** Disabling *walking on triangle* leads the Gaussians to stick and pile up on triangle boundaries, and cause artifacts when animated by novel poses.

Ablation on *walking on triangle*. We firstly conducted an ablation study on head avatar *bala* where we disabled the *walking on triangle* mechanism and clipped the UV values to prevent the Gaussians from moving beyond their corresponding triangles. In addition to the performance drop as listed in Table A2, the Gaussians tend to stick and pile up on the boundaries of the mesh triangles as shown in Figure A6. The performance drop was more significant in the second experiment on full-body avatar *male-3-casual*. Especially when animated by novel poses, turning off *walking-on-triangle* resulted in noticeable artifacts.

References

[1] Shrisha Bharadwaj, Yufeng Zheng, Otmar Hilliges, Michael J. Black, and Victoria Fernandez Abrevaya. FLARE: Fast learning of animatable and relightable mesh avatars. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, page 15, 2023. 2, 3

[2] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, 2021. 1

[3] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural Head Avatars From Monocular RGB Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18653–18664, 2022. 1, 3

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1

[5] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I M Avatar: Implicit Morphable Head Avatars From Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13545–13555, 2022. 1

[6] 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 (CVPR)*, pages 21057–21067, 2023. 3

[7] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant Volumetric Head Avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4574–4584, 2023. 1, 2, 3