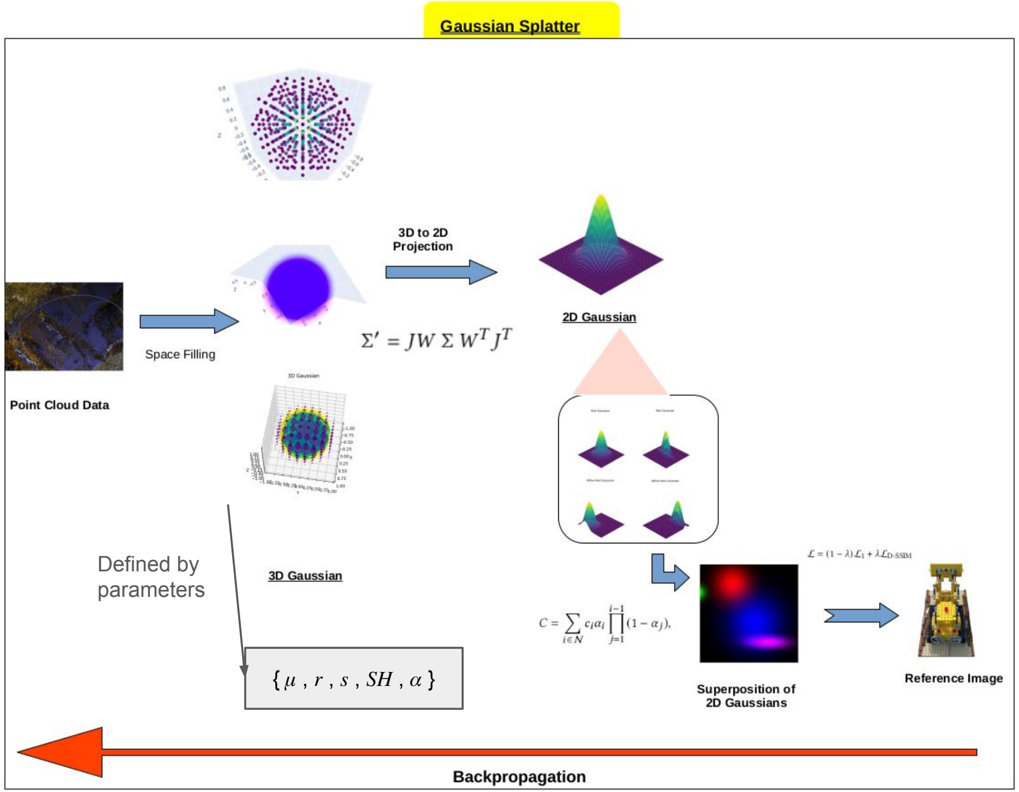
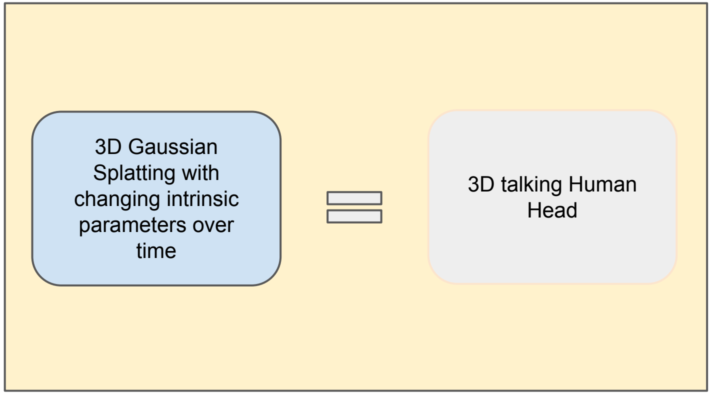
GaussianTalker: Real-Time High-Fidelity Talking Head Synthesis with Audio-Driven 3D Gaussian Splatting

Junaid Iqbal Khan


Gaussian Splatter
Point Cloud Data
Space Filling
3D to 2D Projection
$\Sigma' = JW\,\Sigma\,W^T J^T$
2D Gaussian
3D Gaussian
Defined by parameters
$\{\mu, r, s, SH, \alpha\}$
$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1-\alpha_j),$
Superposition of 2D Gaussians
$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$
Reference Image
Backpropagation

Inspiration

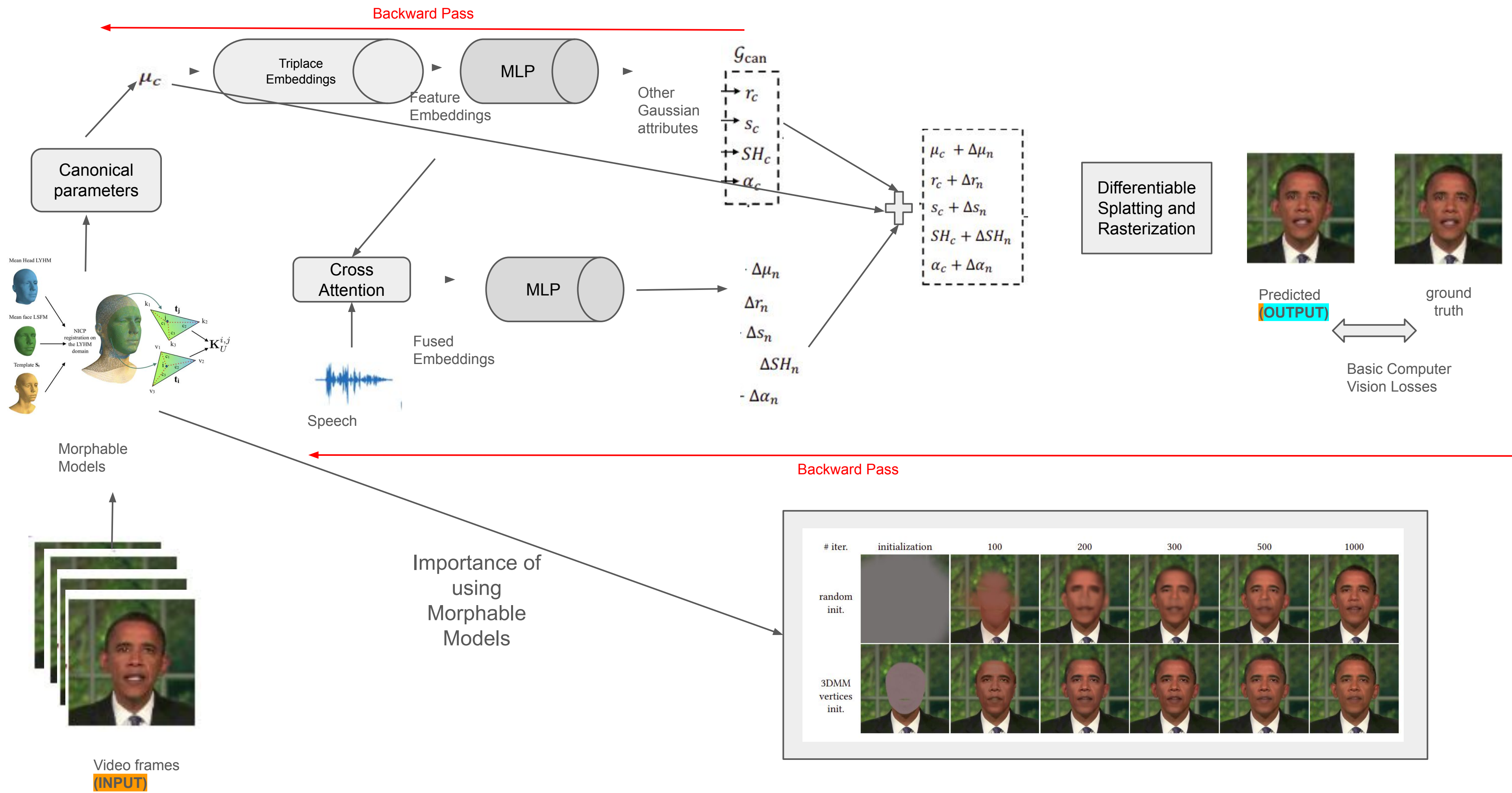


3D Gaussian Splatting with changing intrinsic parameters over time

=

3D talking Human Head

Main Idea



Backward Pass

μ_c

Triplace Embeddings

MLP

Feature Embeddings

Other Gaussian attributes

$\mathcal{G}_{\text{can}}$

r_c

s_c

SH_c

α_c

Canonical parameters

Cross Attention

MLP

Fused Embeddings

Speech

$\Delta\mu_n$

Δr_n

Δs_n

ΔSH_n

$\Delta\alpha_n$

$\mu_c + \Delta\mu_n$

$r_c + \Delta r_n$

$s_c + \Delta s_n$

$SH_c + \Delta SH_n$

$\alpha_c + \Delta\alpha_n$

Differentiable Splatting and Rasterization

Predicted **(OUTPUT)**

ground truth

Basic Computer Vision Losses

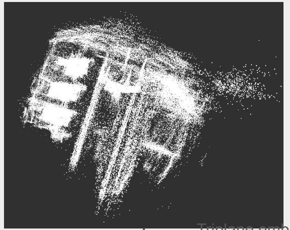
Backward Pass

Morphable Models

Importance of using Morphable Models

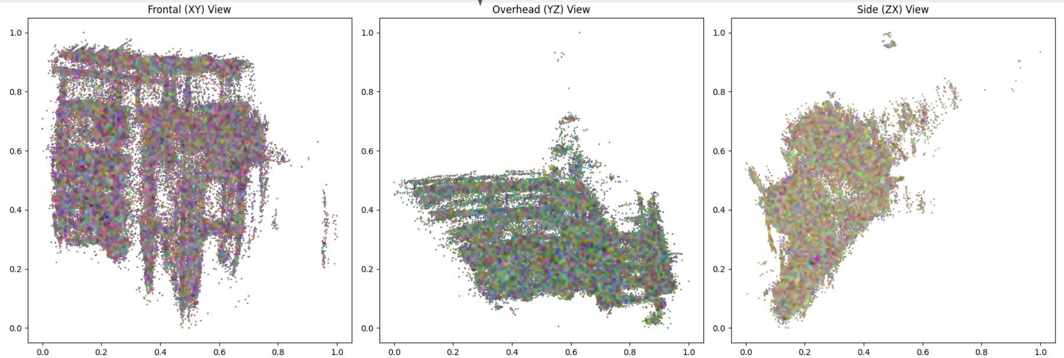
Video frames **(INPUT)**

Detailed Scheme (that just implements main idea)



My recent Gaussian splatting point cloud data (~23000 points)

Triplane embedding features, compressed to 3 representing RGB values (for purpose of visualization)



Triplanar Embeddings of Gaussian splatting points [2]

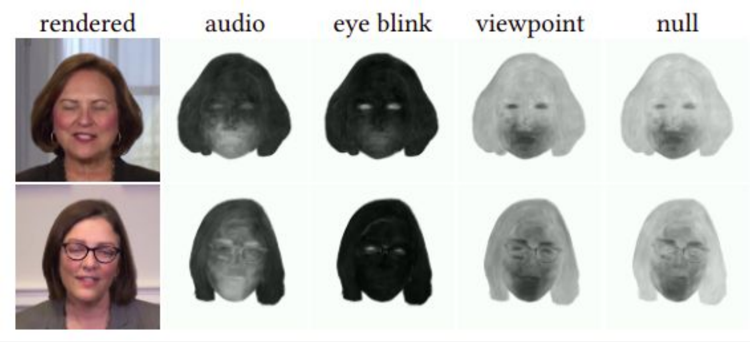
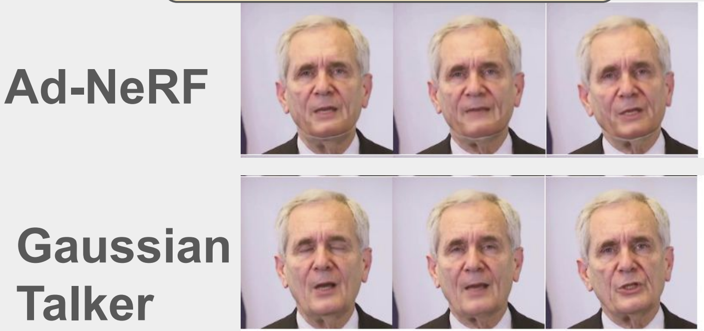


Illustration of Attention map of necessary (audio) and irrelevant (rest) features with respect to fused features (that essentially updates canonical Gaussian parameters)

Fast and with better global feature consistency



Criticism

1. In the phase where Gaussian splatting model's parameters are updated on each frame can be computationally intensive, given that a standard video is 3 minute long with 30 frames per second, as well as a moderate resolution video generates 25000 Gaussian points. The computations are O(10e9) in addition to original training of Gaussian splatting. It is excessive in computation, which fails to take advantage of intermediate encodings of Gaussian parameters being low dimensional.

2. The author's argument of disentangling is questionable because they aim to make rendering of lips independent of irrelevant movements, so they propose following cross attention action.

$$z_n'^{\,l} = \mathcal{T}_{CA}(z_n^{l-1}, \{a_n, e_n, v_n, \emptyset\}) + z_n^{l-1}, \quad l = 1...L.$$

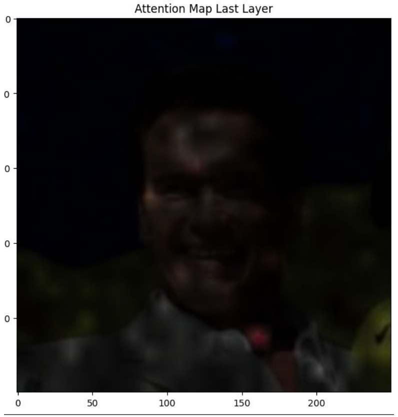
But intuitively speaking, it enforces the use of irrelevant movements (via associated auxiliary embeddings) into mapping towards Gaussian parameter update (ultimately). Fundamentally, their claim of independency gets immediately violated by using chain rule as follows.

$$\delta z' \propto \frac{\partial z'}{\partial e}\delta e = O(||\frac{\partial z'}{\partial e}||_F)\delta e$$

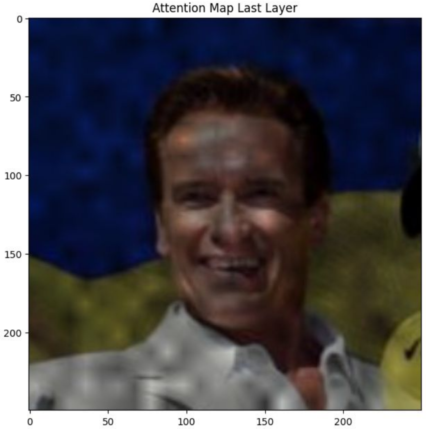
The argument of big-O notation is not guaranteed to approach zero, and hence the cross-attention outputs are not really independent of irrelevant video motions.

.

3. Another critique of author’s disentangling approach is that it may even deteriorate the performance. In following comparison, it can be seen that using vision transformer with noise embeddings deteriorate the attention map, in comparison with standard vision transformer.



Without Nose Embedding [2]



With Nose Embedding [2]

4. Using COLMAP-free gaussian splatting could be a great choice in contrast to using morphable models, which are just an large processing step just for initialization of point cloud (https://oasisyang.github.io/colmap-free-3dgs/). I think it is better alternative to not using SFM, instead of author’s choice of using deformable models.

References

[1] Cho, Kyusun, Joungbin Lee, Heeji Yoon, Yeobin Hong, Jaehoon Ko, Sangjun Ahn, and Seungryong Kim. "GaussianTalker: Real-Time High-Fidelity Talking Head Synthesis with Audio-Driven 3D Gaussian Splatting." *arXiv preprint arXiv:2404.16012* (2024).

[2] https://github.com/superdianuj/gaussian_talker_ablation