HeadSplat: Generalizable 3D Head-Reconstruction

Fatma Ben Ayed*
Antonio Oroz†
Hakan Alp‡

Abstract

The creation of photorealistic 3D head avatars has become increasingly important for applications in gaming, social media, and online shopping, where convincing user representations are crucial. However, achieving high-quality 3D head reconstructions with minimal input data remains a significant challenge for existing techniques, which struggle with sparse views and lack geometric precision. This paper presents HeadSplat, a novel approach for generating photo-realistic 3D avatars using a generalizable Gaussian-Splatting [4] based approach, leveraging geometrical priors and projected features from image encoders. Our method allows us to reduce the required inputs to as little as two or even a single view, while maintaining visually realistic and geometrically accurate results. Our experiments demonstrate that HeadSplat can outperform current state-of-the-art methods and our ablations demonstrate its applicability to real-world applications.

1. Introduction

Digital representations of humans have become increasingly relevant in a variety of real-world applications. Especially realistic 3D head avatars have the potential to revolutionize social media interactions and gaming, by providing convincing and personalized representations of users, and even transform personalized online shopping by providing more engaging customer experiences. The growing demand for high-quality 3D head models in these domains underscores the importance of developing efficient and effective methods for their creation.

However, producing these high-fidelity 3D head models is not a trivial task. Photo-realistic novel-view synthesis methods, such as Neural Radiance Fields [7] and 3D Gaussian Splatting [4], typically require a large number of precisely captured images to generate accurate results. While these approaches have shown promising and photo-realistic results, they are not ideal for scenarios where only a few images are available. For the use cases mentioned earlier, reducing the number of required input images significantly enhances the method's usability and appeal.

Generalized Few-shot approaches like PixelNeRF [10], PixelSplat [1], or DUST3R [8] have advanced the state-of-the-art in novel view synthesis from limited input data. However, they cannot always achieve photo-realistic quality for 3D head avatars. This is largely because these models are designed to cover a broad range of objects, which prevents them from applying strong geometrical priors specific to human faces.

In our work, since we operate within a narrow domain: human head reconstruction, we can impose useful prior knowledge about geometry and visual appearances. Specifically, we leverage depth priors and FLAME meshes [6] to provide a coarse initial geometry of the head from sparse input views. The FLAME mesh serves as a first step to provide an initial template for facial landmarks and structures, which significantly reduces the number of input images required.

We introduce HeadSplat, a Gaussian-based, generalizable head reconstruction model that can produce high-quality 3D head avatars from as few as one or two input images. By conditioning our model on these images and their associated camera parameters, our method infers a 3D Gaussian representation of the head avatar, which can be rendered from novel viewpoints. This enables the network to be trained across multiple subjects, allowing it to perform novel view synthesis in a feed-forward manner from a sparse set of views.

Our approach addresses key limitations of existing methods and also expands the usability and practicality of 3D head modeling. By requiring as few as one image, HeadSplat makes photorealistic 3D head reconstruction more accessible and scalable for real-world use.

We conduct extensive experiments to evaluate the efficacy of our framework. Our results demonstrate that HeadSplat is capable of generating photo-realistic novel views from a single image and using a generic FLAME mesh. Furthermore, we qualitatively and quantitatively compare

*TUM, email: Fatma.ben-ayed@tum.de
†TUM, email: antonio.oroz@tum.de
‡TUM, email: hakan.alp@tum.de

our model against 3D Gaussian Splatting and DUST3R, where HeadSplat consistently produces higher-quality 3D head avatars than these prior approaches.

2. Related Work

2.1. 3D Gaussian Splatting

3D Gaussian Splatting [4] is a method to represent 3D objects using Gaussian distributions, enabling smooth and flexible renderings of complex shapes. Surfaces are represented using Gaussian ellipsoids, which are refined and created through the backpropagation of loss gradients against ground-truth images. In contrast to Gaussian Splatting, our method is designed to work in sparse-view settings where as little as one view is available of a person's head.

2.2. FLAME

FLAME [6] (Faces Learned with an Articulated Model and Expressions) is a statistical model for creating realistic 3D face meshes. A general mesh template is formed into a personalized one using fitted parameters which control identity, expression and pose. This model provides a structured way to represent facial geometry, making it suitable for tasks that require personalized facial reconstructions. In our project, FLAME is used to initialize a general 3D head. This mesh serves as the baseline structure that our model refines by predicting the necessary displacements to match the true facial geometry.

2.3. PixelNerf

PixelNeRF (Pixel-Based Neural Radiance Fields) reconstructs scenes by extracting feature maps from images with an image encoder, then constraining the neural radiance field (NeRF [7]) on these features. For each 3D point found along the camera ray, it uses the corresponding pixel features to predict color and density. Color and density found are then accumulated through volumetric rendering to reconstruct the final 3D scene, allowing the model to generalize from 2D views to novel 3D viewpoints. Our work also uses image encoders to help the model generalize faster, but unlike PixelNeRF, we use Gaussian Splatting as a novel and more efficient NeRF replacement. Additionally, since we operate on the more specialized domain of human heads, we're able to rely on strong geoemetric priors.

2.4. PixelSplat

PixelSplat [1] projects 3D points onto a 2D plane. Each point is represented by a Gaussian splat. It is similar to 3D Gaussian Splatting but in 2D space. It focuses on 2D projection and compositing using a depth buffer, blending the splats, and handling depth and visibility directly in the image space. While PixelSplat is for a general 3D reconstruction, our work is focused on a more specialized domain, the head reconstruction, helping us leverage both geometric head priors and specific image encoders that benefit the constraints of the head geometry.

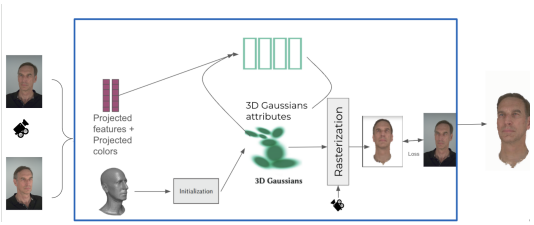

Figure 1. **Proposed architecture in the two-view case**. Project FLAME mesh into each of the given input images. For each valid projected FLAME vertex, extract precomputed image features and image colors. Then, aggregate these features and colors. these are decoded along with Gaussian positions into Gaussian attributes and rendered into novel views

2.5. DUSt3R

DUSt3R introduces a novel approach to dense and unconstrained stereo 3D reconstruction. It detects point maps from arbitrary images and does not require prior camera calibration or pose information. It leverages Transformer-based architectures. It enables consistent 3D modeling, depth estimation, and camera pose recovery. Like the other methods discussed, DUSt3R is also a general reconstruction pipeline. Since our work focuses specifically on face reconstruction, we can leverage more constraints. Additionally, using Gaussians to represent a head geometry instead of fusing a point cloud is more effective in handling artifacts and sparse regions of the object.

3. Method Overview

We now describe our Gaussian-based approach to render novel views of human heads from given input images. Inspired by the PixelNeRF [10] framework, our method is designed to generate high-quality 3D head avatars from as few as one or two input images. By adapting the PixelNeRF [10] pipeline to 3D Gaussian Splatting [4], we leverage 2D image encoders and 3D geometric priors to create photo-realistic head models with minimal input data. We show our pipeline schematically in Fig. 1 .

To provide a clear overview of our approach, we want to present an abstract formulation of the HeadSplat model before going into detail: Given input images I_{in}, their camera parameters π_{in} and a set of FLAME vertices v, we first project the 2D input images and their extracted features (using FaRL) onto the vertices. This gives us the projected colors c and features f.

$$c = \text{project}(v, I_{\text{in}}, \pi_{\text{in}}) \tag{1}$$

$$f = \text{project}(v, \text{FaRL}(I_{\text{in}}), \pi_{\text{in}}) \tag{2}$$

We then apply our MLP model to predict the gaussians and their attributes G.

$$G = \text{MLP}(v, c, f) \tag{3}$$

In the last step we use a rasterizer to render our gaussians from all 16 viewpoints π_{all}, in order to compare it against the ground truth images I_{all}, giving us our training loss L.

$$L = \text{loss}(\text{rasterize}(G, \pi_{\text{all}}), I_{\text{all}}) \tag{4}$$

3.1. Data and Preprocessing

For our model training and experiments we rely on the NeRSemble [5] dataset. We use a subset of 270 faces in a neutral position, split into 260 datapoints for training and 10 for validation. Each face has 16 input images, split equally into a top and bottom row, as visualized in figure 2. Additionally we also have access to corresponding FLAME [6] fittings, face segmentation masks, COLMAP [2] point clouds, camera parameters and color correction matrices.

During preprocessing we mask out the background and the torso for each image. We further apply the color correction in order to achieve inter-view color consistency between the images. The FLAME mesh is also upscaled from a standard resolution of 5k vertices to 80k vertices using midpoint-subdivision, in order to provide higher number of initial gaussian positions for our model.

3.2. 3D Gaussian Initialization

The first step in our approach involves initializing the 3D Gaussians based on the FLAME mesh. FLAME provides a coarse but accurate representation of facial geometry, which we use as the basis for positioning the 3D Gaussians. Specifically, points on the FLAME mesh are utilized as the initial positions for the Gaussians in 3D space.

3.3. Coordinate System

We standardize the coordinate system by defining the canonical space of the FLAME mesh as a reference. Instead of transforming the FLAME mesh into the world space of the cameras, we apply the inverse transformation to align the camera poses across persons such that 3D landmarks are always roughly at the same world coordinates. This helps our MLP to generalize over world space properties as shown in Fig. 2.

3.4. Feature Extraction and Aggregation

We use a pre-trained image encoder, FaRL [12] to encode the input images into a pixel-aligned feature grid.

Farl is trained specifically for feature extraction from facial images. It produces a set of feature maps that are aligned with the pixels of the input image, capturing detailed spatial information. These feature maps consist of multiple channels, the first channels represent different aspects of the image, such as edges, textures, and other high-level features, while the last feature channels contain segmentation masks of facial regions such as the eyes, nose, mouth, and other key features, as shown in Fig 3. By projecting our initialized 3D gaussian positions, we are able to access the corresponding feature and color information. In our aggregation procedure we compute the distance of each gaussian to the input views available, and correspondingly use the information from the nearest image.

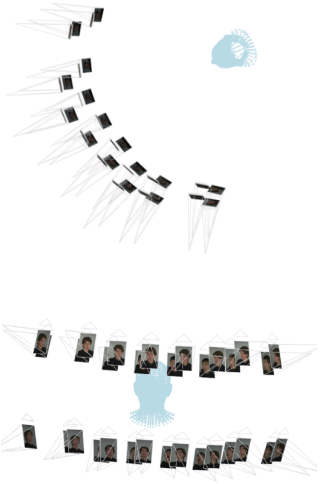

Figure 2. Aligning the camera poses to the FLAME canonical space

Figure 3. FaRL [12] features, 768 feature channels + 11 classes

3.5. Gaussian Parameter Prediction

This aggregated information along with gaussian positions serves as the input to a multi-layer perceptron (MLP), which is responsible for decoding the remaining attributes necessary for 3D Gaussian Splatting. Specifically, the MLP predicts the 3D Rotation, 3D Scale, Color, and Opacity for each Gaussian. Our MLP consists of linear layers and residual blocks and we employ standard techniques like improved activation functions (LeakyReLU), layer normalization and dropout. The specific structure is available in our codebase.

The final step in our pipeline is rendering the 3D Gaussians into novel views using 3D Gaussian Splatting's [4] differentiable tile rasterizer. The rendered images are then compared to ground truth images, and the difference is computed as an image-space loss. This loss is used to supervise the learning process, guiding the model towards producing photorealistic human head avatars.

3.6. Multi-View Supervision

The total loss L_{total} is a weighted sum of several loss components, each serving a specific purpose in refining the 3D head model:

$$L_{\text{total}} = 0.8L_{\text{L1}} + 0.2L_{\text{SSIM}} + 1.0L_{\text{LPIPS}} + 0.5L_{\text{depth}} + 0.1L_{\text{normalization}}$$

- **L1 Loss (L_{L1})**: This is the standard comparison loss, measuring the pixel-wise difference between the rendered images and the target images. It ensures that the overall appearance of the rendered images closely matches the target images.

- **SSIM Loss (L_{SSIM}) & LPIPS Loss (L_{LPIPS})**: These losses focus on perceptually important features, ensuring that the rendered images not only match in terms of pixel values but also in terms of perceived image quality. SSIM [9] measures structural similarity, while LPIPS [11] focuses on perceptual differences, capturing higher-level image features.

- **Depth Loss (L_{depth})**: To ensure geometric accuracy, this loss penalizes discrepancies in depth between the rendered and target images. It ensures that the reconstructed geometry is correct and not just visually appealing from certain viewpoints. Our ground-truth 3D geometry is extracted using a depth map, which we obtain by projecting a provided point cloud (generated by COLMAP [2]) into the 16 views. For our depth loss, we used a relaxed relative loss from literature, the Pearson correlation, on our GT and rendered depth maps from the 3DGS renderer. It measures the distribution difference between 2D depth maps without being hindered by the inconsistencies in absolute depth values, following the below function:

$$\text{Corr}(\hat{D}_{\text{ras}}, \hat{D}_{\text{est}}) = \frac{\text{Cov}(\hat{D}_{\text{ras}}, \hat{D}_{\text{est}})}{\sqrt{\text{Var}(\hat{D}_{\text{ras}})\,\text{Var}(\hat{D}_{\text{est}})}}. \quad (5)$$

- **Normalization Loss ($L_{\text{normalization}}$)**:

  This loss aims to prevent large displacements and ensures that the 3D Gaussians maintain reasonable positions. It also helps in stabilizing the optimization.

4. Experiments and results

4.1. Experimental Setup

For our experiments, we use our standard model as described in our method section. Unless otherwise specified (e.g., removal of encoder, etc.). We project the images and features from two views, one from the top-left side and one from the top-right side. The specific training configurations can be found in our codebase. Our model was trained for 105 epochs, taking approximately 4h on a single NVIDIA RTX 3080.

4.2. Baseline Comparison

To evaluate the effectiveness of our 3D head reconstruction model, we compare its performance against three established baselines: 3D Gaussian Splatting with FLAME initialization, DUSt3R, and FLAME initialization with color projection. Each of these baselines represents a different approach to 3D head reconstruction, providing a comprehensive framework for comparison. Since our work is focused on reconstructing on sparse sets of images, all comparisons are made using the same two images.

4.2.1 Quantitive Analysis

Table 1 compares our work and 3D Gaussian Splatting across various people from the validation using the LPIPS, SSIM, and PSNR metrics. Our work consistently achieves lower LPIPS scores (average of 0.1650) and higher SSIM scores (0.8438) compared to 3D Gaussian Splatting (0.3961 LPIPS and 0.7673 SSIM), indicating better perceptual and structural similarity. However, 3D Gaussian Splatting performs better in regards toPSNR, with an average of 18.01 compared to 17.64, suggesting a marginally better signal-to-noise ratio at the possible expense of perceptual and structural quality.

4.2.2 3D Gaussian Splatting with FLAME as initial mesh

In figure 4, we compare our method against 3D Gaussian Splatting [4] with FLAME initialization, giving it the same geometric prior as our method to deal with the sparse set of images. Nonetheless, our model offers a significantly clearer visual appearance with a clear defined surface, while 3D Gaussian Splatting has difficulties dealing with artifacts and transparency issues. This leads us to the conclusion that our model is notably better suited to deal with the sparse inputs, being able rely on learned knowledge about face structures during its training process.

ID	Our Work			3D Gaussian Splatting		
	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑
059	**0.1033**	**0.8750**	**21.34**	0.4048	0.7630	18.42
070	**0.2558**	**0.7576**	13.35	0.4046	0.7528	**17.43**
368	**0.1198**	**0.8512**	17.86	0.3595	0.7838	**17.97**
369	**0.1717**	**0.8525**	16.23	0.4070	0.7730	**16.84**
370	**0.2363**	**0.8210**	14.60	0.4075	0.7609	**17.57**
371	**0.1593**	**0.8682**	**19.35**	0.3654	0.7560	18.43
372	**0.1396**	**0.8505**	18.68	0.4183	0.7824	**18.64**
373	**0.1475**	**0.8396**	**19.28**	0.4633	0.6943	18.32
374	**0.2038**	**0.8486**	15.49	0.3611	0.8043	**17.93**
375	**0.1132**	**0.8740**	**20.18**	0.3691	0.8024	18.53
All	**0.1650**	**0.8438**	17.64	0.3961	0.7673	**18.01**

Table 1. Comparison of our work and 3D Gaussian Splatting across different people from the validation set using LPIPS, SSIM, and PSNR metrics.

Figure 4. Comparison of different models using 2 images: (a) Our model, (b) Gaussian splatting, and (c) DUSt3R

4.2.3 DUSt3R

DUSt3R offers a straightforward and efficient reconstruction process when given more images, but it falls short in capturing the finer details of facial geometry when executed with two images. Comparing DUSt3R against our method in figure 4, we can see that DUSt3R's outputs can also achieve good quality. The hair, for one, looks significantly better with DUSt3R, showing our shortcomings when dealing with geometry not covered by the FLAME initialization. On the other hand, our model is able to achieve better results in the face area. Firstly, we can see that in some cases, DUSt3R has difficulties to predict the face proportions accurately, as seen in the third person. It also shows significantly more artifacts for all three persons and some empty regions leading to transparency issues. The nature of DUSt3R's point cloud prediction also plays into the advantage of our method, as we also predict other attributes of the Gaussians, leading to a complete surface reconstruction. DUSt3R's online application is also able to create smooth surfaces, but these lead to even more artifacts in the face regions. In conclusion, we argue that while the results from DUSt3R offer good reconstructions, our model has the ability to leverage its comparatively narrow focus on human heads to be able to generate more realistic looking 3D avatars.

4.2.4 FLAME Initialization with Color Projection

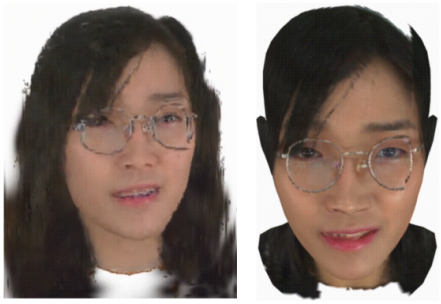

Figure 5. Comparison of different models using 2 images: (a) Our model, (b) FLAME Initialization with Color Projection

Separately, we also want to compare our method against a rather simple initialization of a FLAME mesh with projected color in 5. Due to the direct application of color, we can see a highly-detailed reconstruction of the projected pictures, outperforming the visual quality of our method in some regions like the eye area. This projection works really well, if the FLAME mesh resembles the geometry of

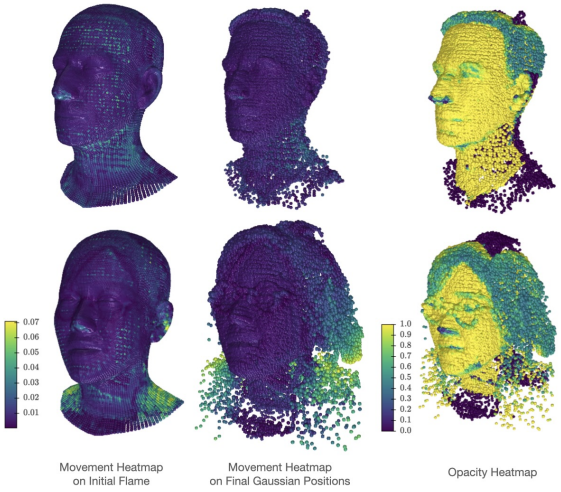

Figure 6. Visualizations of the predicted xyz delta (left) and opacity (right) of the Gaussians

the real head. But in areas where this is not the case, i.e., the hair region or the glasses, the FLAME method seems unnatural, only providing flat surfaces. This highlights the importance of our generalized Gaussian Splatting pipeline, as it is able to adjust the Gaussians and their positions, in order to create realistic representations. Applied to the glasses and the hair, our reconstructions create a more life-like 3D avatar.

5. Ablations

Method	LPIPS ↓	SSIM ↑	PSNR ↑
Standard Model	0.1650	0.8438	17.64
No Depth	0.1617	0.8462	18.03
No Encoder	0.1775	0.8437	17.11
ResNet Encoder	**0.1583**	0.8493	17.83
Single Image	0.1684	0.8468	17.80
Template Mesh	0.1597	**0.8507**	**18.13**

Table 2. Comparison of our ablated models against our standard model. We compute LPIPS, SSIM and PSNR for all 16 rendered views against the ground truth images.

5.1. How good are our initializations?

In figure 6 we show visualizations of xyz delta and opacity predictions for two persons, one with short hair and one with long hair. With this visualization we want to determine how much our FLAME initialization helps our model to produce the final output.

With a nearly perfect initialization, the movement of the gaussians would be close to zero and all gaussians would be reused, i.e., have opacities greater than zero. Let's first discuss the simpler task of generating a 3D avatar of a person with short hair. In this case we can clearly see that the movement of gaussians is minimal. The face area nearly has no movement at all, while for the hair, the positions need to be slightly adjusted, which is to be expected as FLAMEs do not include hair geometry. Regarding the opacity, we can see that most gaussians are reused, except for the torso region, which we mask out during our preprocessing.

Now we want to discuss the difficult scenario in which a person has long hair, i.e., our FLAME might be far from the true 3D geometry. Again, as expected, the face area sees minimal movement, except for the glasses. On the other hand, to generate the hair geometry, the gaussians need to move significantly more. Interestingly, a lot of gaussians from the shoulder region are used for this, as shown by the movements heatmap and opacity heatmap. This difference in movement between the short-haired and long-haired person shows us that our initializations still have room for improvement. Furthermore, our model also doesn't aggregate any information based on the local neighborhood, forcing it to predict the gaussians independently. These two points explain the worse geometry and strong artifacting for longer hair, which we observed in our baseline comparisons.

5.2. How much does depth help?

For this ablation we have removed the depth from our loss function and compare it against the same model with depth loss enabled. The qualitative comparison between both experiments is shown in figure 7. For this comparison, we will split the discussion into two parts, first focusing on the inner region of the avatars, i.e., the face, and then we'll go through the outer parts, focusing especially on the hair.

In the face region we can see less artifacts for the eyes with depth loss enabled, this is especially visible for the upper avatar. We credit this change to the depth loss as a regularizing factor for the eyes, forcing the highly fluctuating region (visually) to keep it's geometry truthful to the comparatively flat nature of an eyeball. Otherwise we don't observe significant differences for the face.

For the outer regions of the face we can see more significant differences. For person depicted on the bottom we observe that there's significantly less noise under the ear, while for the top person the artifacts on the top of the head are reduced as well. We attribute the worse hair without depth loss to the noisy nature of the masking step during preprocessing, where we try to remove the background region, but cannot do so perfectly. The depth loss on the other hand, is significantly cleaner for this region, giving a good learning signal to avoid these artifacts.

While the visual perception of our 3D avatars is improved through depth loss, the quantitative comparison in table 2 shows that, on paper, the no-depth model performs slightly better. The depth loss being an additional loss

Figure 7. Qualitative Comparison of Results with and without depth loss

solely focusing on geometry, could be a factor for this result.

In conclusion, we would nonetheless describe our depth loss as a method to boost visual quality, which can further improve already good results.

5.3. How much do the image encodings help?

In our standard model we use projected image features generated with an image encoder specifically trained for the task of face segmentation, namely FaRL. The idea behind using image encodings and classes provided by FaRL is that inter-person similarities, i.e., hair, have similar features and therefore enable faster generalizability. In this ablation we want to compare this to a model which relies on color information only and to a model relying on additional image features from a general image encoder, namely ResNet. The visual comparison is depicted in figure 8.

First, let's evaluate the color-only model to the two models with image features. For the color-only model we can first spot flat hair geometry for both individuals depicted, while both ResNet and FaRL models show a more truthful hair geometry, which is farther distributed from the head. This confirms our expectation that image features would help with better generalizability across people. The models with image encoders are able to adapt the learnt behaviour of hair to our validation set, while the no-encoder model struggles, as it has, apart from the gaussian positions, only

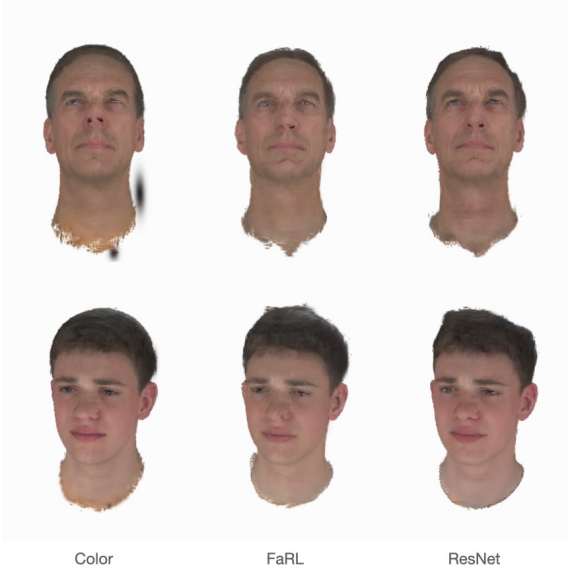

Figure 8. Qualitative Comparison of Results with different projected inputs into our model, specifically color information only, Color and FaRL features (standard model), and Color and ResNet features

noisy color-information of hair available. We can also see a more pronounced color difference between the two different image projections used in the color-only model, this line is visible to the right of the nose from the viewers perspective. These color-differences are a result from slight changes in color provided by the input pictures. The image-encoder models do not have such pronounced differences, leading us to believe that the input color in these models is not reproduced one-to-one and that the projected image features also play a role in producing the final color of each gaussian. We can also confirm our results quantitatively through table 2. Both the standard FaRL model and the ResNet model are able to outperform the no-encoder model in all three metrics.

Comparing the two image-encoder models, differences become less pronounced, indicating that both models are able to do a similarly good job in regards to generalizability. Some notable differences are that the throat area looks slightly better with the FaRL model, while nose geometry seems to work better with the ResNet model. Overall, the ResNet model is even able to outperform the FaRL model in regards to our metrics (table 2).

5.4. Can we do single-image reconstruction?

In this experiment we want to test a scenario which improves the real-world usability of our method. In our standard model we use two images for color and features projection. These images are taken from the far-left and far-right side of the face, ensuring good coverage of the face.

Figure 9. Visual Comparison of the 3D avatars using a single front image and our default method with two images.

Figure 10. Visual Comparison of the 3D avatars using a template mesh and a personalized mesh (standard method).

We now want to compare a model which works only with a single image as input in figure 9. The input image is taken from the upper-front of the face.

In our comparison using metrics shown in table 2, the single-image model is able to outperform our standard model on two of our three metrics (namely SSIM and PSNR). This would indicate a better performance with less information given. Our qualitative analysis on the other hand, will show some disadvantages. Starting with the bottom person in the single-image scenario, we can see that the chin area has a different color compared to it's surrounding region, while the the two image model has no difficulties in this area. This change can be solely attributed to the fact of missing information and wrongly projected color. This also demonstrates a weakness of our model, which is the missing aggregation functionality of neighborhood information. For the top person we can also see that in the single-image model the hair seems more flat, indicating that from the single-image viewpoint we don't learn enough information about the hair geometry to resemble it well, even though loss is performed on all 16 images.

Nonetheless, apart from these smaller imperfections, even with a single image, our approach is able to produce realistic and usable results. Especially in the face area, both models are able to perform well, as both models have good face coverage using the projected features. Therefore we can conclude that real-world usability with a single image should not pose any significant constraints on our method.

5.5. Can we rely on a template mesh?

In this last ablation we want to investigate the effects of using a general template FLAME mesh as an initialization to our model. For our standard model we used personalized FLAME meshes, which resemble the head geometry closely. While this should produce better results, it might be hard to rely on the availability of them in all usecases. It is noteworthy nonetheless, that there are existing methods that can produce personalized FLAME meshes from images (MICA [13]). Our comparisons with the template mesh can be found in figure 10.

For most of the individuals in our validation set, the personalized mesh is thinner than the template mesh. This leads to results with wider head geometries than which would have been truthful to the source image. This is especially observable when comparing the person depicted on the top, as the template mesh model creates a noticeably wider 3D avatar. However, for both persons, apart from the slightly imperfect geometry compared to the real person, the visual quality is still good. Taking the quantitative comparisons from table 2 into account, we can even argue that the results might be visually better than the standard method, as in all three metrics the model with the template mesh is able to outperform our standard approach. Consequently, we can confidently say that our method does not

strictly require personalized FLAME meshes, making our model significantly more usable. This usage of a single template mesh reduces the required inputs for our method to images and the camera parameters aligned with regards to the FLAME head.

6. Conclusions, Limitations & Future Work

6.1. Conclusions

In this research projected we have shown that even with a greatly reduced amount of images than required by other methods like Gaussian Splatting, we can achieve photo-realistic 3D head avatars.

Our first contribution was a strong initialization, giving the model strong 3D information, which would have been otherwise hard to extract from only two or less images. This initialization using FLAME is indeed helpful, as the positions do not need to move a lot in the face area, making it easier for the model to capture other important properties like color.

With our ablations we have demonstrated that real-world adaptability is also realistically possible. Instead of using the personalized initialization, it is possible to rely on a general template mesh, requiring less input or preprocessing steps. This change does result in some loss of 3D truthfulness to the real head shape, but still provides great visual performance. Futhermore, we also do not need to rely on multiple images, as used in our main comparisons, but can instead use one image with good coverage of the face. This further improves usability in real-world environments, where multiple images might require more effort on the user side.

6.2. Limitations

Albeit achieving realistic results, our method still has room for improvement. The first issue we have covered in our results section is the visual degradation of areas which are geometrically not covered by a FLAME mesh. Such regions might be accessories like glasses, but especially one significant area, which is long hair. In these cases, the 3D geometry is often unrealistic and covered with artifacts. Another issue are unobserved regions, where no or wrong color information is available. This results in unrealistic parts in the 3D avatars.

6.3. Future Work

In this section we want to discuss some potential further paths of exploration to improve the performance of our proposed method.

Firstly, an expansion of the model to include an aggregation method for the local neighborhood could improve performance greatly for small unobserved regions. It would also benefit regions in which the initial FLAME geometry needs to be modified significantly, as our method currently needs to predict the placement of gaussians independently. The local aggregations could also be further by aggregation features based on face symmetries, which could help in scenarions where only one side of the face is captured.

Another enhancement could focus on even better geometry initialitations. As discussed above, the FLAME initialization work great for the face, but do not include any 3D accessories or even hair. One such potential enhancement is a switch to NPHM [3] head models as initializations, which are highly detailed and cover additional geometry including hair.

To utlize our improved reconstruction technique for the area of animated 3D avatars, the FLAME initializations might prove useful, as FLAME models have the ability to be transformed into different expressions. In this scenario, we could transfer the transformations applied to the initial FLAME model onto the Gaussians. It is noteworthy, that this approach would lead to potentially significant new challenges, including new unseen parts and Gaussians which end up in wrong places, as they weren't close to the initial position anymore.

Lastly, we also want to mention potential ethical implications of our work. In this research project we set out to demonstrate the possibility of 3D head avatars from few images, but human faces and therefore also their digital 3D counterparts are a personal and sensitive topic. In a real world scenario it would therefore be important to mitigate biases in regards to underrepresented groups.

References

[1] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction, 2024. 1, 2

[2] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting, 2024. 3, 4

[3] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Learning neural parametric head models, 2023. 9

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 1, 2, 4

[5] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics*, 42(4):1–14, July 2023. 3

[6] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 1, 2, 3

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1, 2

[8] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2023. 1

[9] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 4

[10] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021. 1, 2

[11] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 4

[12] Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu Huang, Lu Yuan, Dong Chen, Ming Zeng, and Fang Wen. General facial representation learning in a visual-linguistic manner, 2022. 3

[13] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces, 2022. 8