Neural Head Avatars from Monocular RGB Videos
–Supplemental Document–

Philip-William Grassal[1*] Malte Prinzler[1*] Titus Leistner[1] Carsten Rother[1]
Matthias Nießner[2] Justus Thies[3]

[1]Heidelberg University [2]Technical University of Munich [3]Max Planck Institute for Intelligent Systems

Figure 1. Given a monocular portrait video of a person, we reconstruct a *Neural Head Avatar*. The resulting 4D avatars feature explicit geometry reconstruction and photo-realistic appearance.

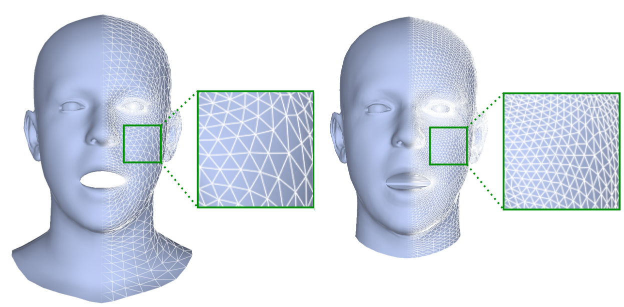
In this supplemental document, we describe the implementation details of our *Neural Head Avatars* method, provide quantitative evaluations for the novel viewpoint synthesis task, and show further qualitative results. In addition, we demonstrate facial reenactment.

1. Implementation Details

1.1. Template Model

We utilize the FLAME head model [15] in its updated version from 2020 as the geometric backbone of our method. As mentioned in the main paper, we perform minor adjustments to the FLAME topology, namely, we uniformly subdivide the faces (four-way subdivision), remove the faces belonging to the lower neck region, and add faces to close the mouth cavity, see Figure 2. This increases the original number of vertices from 5023 to 16227. Inspired by [4], we use $\tanh$ activation functions to limit the joint rotations of FLAME to physiologically plausible ranges.

* Both authors contributed equally to the paper



(a) Original FLAME topology (b) Our FLAME topology

Figure 2. As our template, we uniformly subdivide the FLAME [15] mesh and simplify the mouth cavity.

1.2. Network Architectures

Our method relies on two multi-layer perceptrons (MLPs). The Geometry MLP $\mathcal{G}$ refines the mesh resulting from the linear FLAME head model and adds facial detail and hair structure. The Texture MLP $\mathcal{T}$ synthesizes a dy-

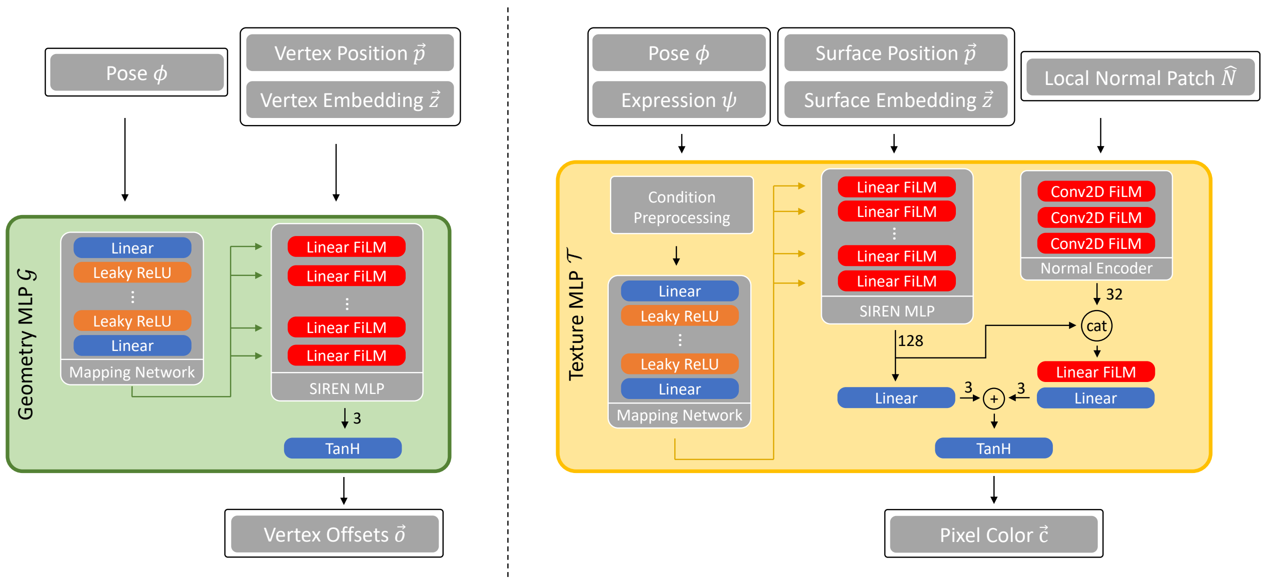


Figure 3. Overview of our model architectures. Our *Neural Head Avatar* relies on SIREN-based MLPs [21] with fully connected linear layers, periodic activation functions and FiLM conditionings [7, 19]. Inspired by [5], surface coordinates and spatial embeddings (either vertex-wise for $\mathcal{G}$, or as an interpolatable grid in uv-space for $\mathcal{T}$) are used as an input to the SIREN MLP. The dynamic frequencies and phase shifts of the Linear FiLM layers are predicted by fully connected mapping networks which are conditioned on the FLAME parameters. For the Texture MLP $\mathcal{T}$, a fully convolutional normal encoder generates additional conditions from a local patch of the predicted normal map.

namic, photo-realistic texture that is able to reproduce view- and expression dependent effects, e.g., wrinkles and reflections. This section details the model architecture of both networks. An overview is presented in Figure 3.

1.2.1 Geometry MLP $\mathcal{G}$

The Geometry MLP $\mathcal{G}$ adds facial detail and hair geometry to the mesh resulting from the FLAME head model. The inputs to the network are the 3D coordinates of the vertices on the template mesh (see Figure 2), normalized to the range $[-1, 1]$, as well as vertex-specific embedding vectors that are optimized during training. Dynamic geometry effects are enabled by also passing the FLAME pose parameters to $\mathcal{G}$. In practice, to avoid overfitting only the 3 pose parameters of the neck joint are used.

The core of $\mathcal{G}$ is a SIREN-based fully connected MLP [21] which takes the vertex positions and embeddings as input, processes them by a sequence of FiLM-conditioned linear layers [7, 19], and predicts the three-dimensional vertex offsets. The linear, FiLM-conditioned layers first perform an affine transformation on the input signal, followed by a sinusoidal activation function. The phase shifts and frequencies of the sinusoidal activation functions are generated by a mapping network which is a concatenation of linear layers with leaky ReLU activations. This architecture was greatly inspired by [5]. The input to the mapping network is the three-dimensional pose of the FLAME neck joint which enables the synthesis of dynamically changing geometry refinements. We ensure spatial geometry consistency by only allowing dynamic geometry refinements for the neck region. To this end, we compute the offsets twice: once conditioned on the orignial neck pose values and once with all conditions set to zero. We smoothly blend both offset predictions according to a fading body region mask.

Configuration details. The vertex embedding vectors have a feature dimension of 32 (for each vertex of template mesh). The mapping network comprises 3 hidden layers with 256 neurons each, followed by leaky ReLU activation functions with slope 0.2. The SIREN MLP consists of 6 consecutive linear FiLM layers with 128 neurons each. The final layer is a linear layer with 3 output neurons, followed by a $\tanh$ activation function.

1.2.2 Texture MLP $\mathcal{T}$

The architecture of our texture representation $\mathcal{T}$ is similar to $\mathcal{G}$. For obtaining the color value of a point on the mesh surface, its 3D coordinates on the FLAME template mesh as well as a surface embedding vector are fed into a SIREN MLP. The surface embedding vector is sampled from a discrete 2D grid in uv space. A mapping network takes features extracted from the FLAME pose and expression parameters and predicts the phase shifts and frequencies of the sinusoidal activation functions in the linear FiLM layers of the SIREN MLP to synthesize a dynamic texture. The dynamic texture is needed in the regions where the reconstructed geometry does not align well with the target surface. This is especially the case for the mouth cavity. Consequently, we process the FLAME pose and expression parameters into features that are highly correlated with the mouth articulation. More specifically, we determine the effective rotation that applies to the mouth region, i.e., combine global rotation and neck rotation, and use its axis-angle representation together with 10 pair-wise distances between vertices on the inner side of the lips as inputs to the mapping network. These conditions only cover mouth-related information and, therefore, are only used as input to the mapping network for the respective surface regions. For all other regions, we fill the conditioning vector with zeros. Effectively, this approach greatly limits the dynamic capacities of the $\mathcal{T}$ to ensure a close bound to the underlying geometry which in turn enables extrapolation to unseen poses and expressions. The outputs of the mapping network are used to dynamically adjust the phase shifts and frequencies of the linear FiLM layers in the SIREN MLP. Given these conditions and the surface position and surface embedding, the SIREN MLP produces a latent space vector. This vector serves as input to two network heads. In the first head, the latent vector is fed through a linear layer with 3 output neurons. In the second head, the latent vector is concatenated with the outputs of a normal encoder network. This network takes a local patch of the rendered normal map as input, passes it through a sequence of 2D-convolutional FiLM layers and outputs a feature vector in latent space. 2D-convolutional FiLM layers have the same structure as linear FiLM layers, but instead of a linear layer, a 2D-convolutional layer is used. The latent vector produced by the normal encoder contains information on the local geometry configuration and enables the synthesis of expression- and view-dependent effects (e.g., ambient occlusions and specular highlights). The output vectors of the SIREN MLP and the normal encoder are concatenated and fed through a sequence of linear layers to produce 3 output activations. The activations of both heads are summed up and a $\tanh$ activation is applied to achieve the final RGB predictions in a range of (-1, 1).

Configuration details. The surface embedding is sampled in uv space from a discrete feature grid with 256×256 feature vectors with 64 channels each via bilinear interpolation. We use a separate uv map for the inner mouth region with 64×64 vectors. The mapping network has the same architecture as for the Geometry MLP $\mathcal{G}$. The SIREN MLP consists of 8 consecutive linear FiLM layers with 256 neurons each, except for the last which has 128 neurons. The normal encoder is designed as a fully convolutional network with 3 consecutive 2D-convolutional layers with stride 1 and kernel size 3 and with periodic activations. All layers have 128 feature channels except for the last one which has 32. The first model head contains one fully connected layer with 3 output neurons. The second model head contains one linear FiLM layer with 128 neurons and one linear layer with 3 neurons.

1.3. Optimization from Monocular RGB Data

Detailed Geometry Objective E_{geom} (Eq. 2) As defined in Eq. 2 in the main paper, the geometry energy term is:

$$
\begin{aligned}
E_{\text{geom}} &= w_{\text{lmk}} \cdot E_{\text{lmk}} + w_{\text{normal}} \cdot E_{\text{normal}} \\
&\quad + w_{\text{semantic}} \cdot E_{\text{semantic}} + w_{\text{reg,geom}} \cdot E_{\text{reg, geom}}.
\end{aligned} \tag{1}
$$

E_{normal} and E_{semantic} are already defined in the main paper, for the other terms we provide additional explanation.

The landmark energy E_{lmk} measures the distance of detected 2D facial landmarks $\mathbf{l_i} \in \mathbb{R}^2$ and the projected counterparts on the mesh surface $\hat{\mathbf{l}}_\mathbf{i} \in \mathbb{R}^2$ and is given by:

$$
E_{\text{lmk}} = \sum_{i=1}^{70} ||\mathbf{l_i} - \hat{\mathbf{l}}_\mathbf{i}||_1 + w_{\text{lid}} \cdot \sum_{i \in \{\text{left,right}\}} ||\mathbf{d_i} - \hat{\mathbf{d}}_\mathbf{i}||_1.
$$

Besides the absolute positions of the landmarks $\hat{\mathbf{l}}_\mathbf{i}$, we measure relative distances $\hat{\mathbf{d}}_\mathbf{i}$ of the eye landmarks at the upper and lower lid to improve the reconstruction of eye lid closure [8]. However, we found that target lid distances $\mathbf{d_i}$ are less noisy when being computed on the facial segmentation (computed by [23]) rather than the detected landmarks. The 2D facial landmarks $\mathbf{l_i}$ are detected using [3, 16] and also contain two iris landmarks.

To avoid convergence to local minima, we employ several geometry regularization strategies summing up to the term $E_{\text{reg, geom}}$ which is given by:

$$
\begin{aligned}
E_{\text{reg,geom}} &= w_{\text{reg,flame}} \cdot E_{\text{reg,flame}} + w_{\text{reg,lapl}} \cdot E_{\text{reg,lapl}} \\
&\quad + w_{\text{reg,surface}} \cdot E_{\text{reg,surface}} + w_{\text{reg,edge}} \cdot E_{\text{reg,edge}}.
\end{aligned} \tag{2}
$$

Following [1, 22], $E_{\text{reg,flame}}$ uses the statistical properties of the linear shape model, and regularizes the prediction towards the average face:

$$
E_{\text{reg,flame}} = w_\beta \cdot |\boldsymbol{\beta}|_2^2 + w_\theta \cdot |\boldsymbol{\theta}|_2^2 + w_\psi \cdot |\boldsymbol{\psi}|_2^2.
$$

The Laplacian regularizer $E_{\text{reg,lapl}}$ computes the relative loss of the Laplace values $\lambda^*(V)$ of the predicted mesh vertices V w.r.t. the surface of the FLAME model V_{Flame}, and controls the smoothness of the predicted offsets:

$$
E_{\text{reg,lapl}} = |\mathcal{W}_{\text{reg,lapl}} \circ (\lambda^*(V) - \lambda^*(V_{\text{Flame}}))|_1.
$$

Note that $\lambda^*(.)$ denotes the discretized Laplace-Beltrami operator on the 1-ring neighborhood. $\mathcal{W}_{\text{reg,lapl}} \in \mathbb{R}_+^V$ defines

	$w_{\text{reg,flame}}$ Geometry Optim.	$w_{\text{reg,flame}}$ Joint Optim.	$w_{\text{reg,surface}}$ Geometry Optim.	$w_{\text{reg,surface}}$ Joint Optim.	$w_{\text{reg,edge}}$ Geometry Optim.	$w_{\text{reg,edge}}$ Joint Optim.	$w_{\text{reg,lapl}}$ Geometry Optim.	$w_{\text{reg,lapl}}$ Joint Optim.
Eyeballs	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	0	0
Eye Surrounding	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	5	10
Forehead	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	0.05	0.1
Face	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	0.05	0.1
Ears	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	25	50
Scalp	1.0E-03	1.0E-03	1.0E-04	1.0E-04	10	10	0.05	0.1
Neck	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	0.1	0.2
Lower Neck	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	0.25	0.5
Nose	1.0E-03	1.0E-03	1.0E-04	1.0E-04	0	0	2.50E-02	5.00E-02

Table 1. Geometry regularization weights. The regularization weights differ for the individual head regions. However, they do not change among target subjects, i.e., they can be used to reconstruct a wide variety of geometries such as different hair styles.

vertex specific weights which allow to control the smoothness in specific regions (e.g., the neck region has a higher regularization). $\circ$ denotes the component-wise Hadamard product. As in [4], we regularize pose-dependent offset variations by adding a pair-wise surface consistency term. For two randomly picked frames i, j, the ℓ_1-norm is computed over the difference between $\mathcal{G}(\phi_i)$ and $\mathcal{G}(\phi_j)$:

$$E_{\text{reg,surface}} = |\mathcal{G}(\phi_i) - \mathcal{G}(\phi_j)|_1.$$

For large geometry corrections (e.g., long hair), we noticed an irregular vertex distribution over the surface which resulted in areas with large triangles and coarse shape modelling. To mitigate this issue, we regularize the length of edges e_i in the scalp region if they deviate too much from the average edge length $\bar{e}$.

$$E_{\text{reg,edge}} = \sum_{e_i} \begin{cases} e_i - \bar{e} & \text{if } e_i > 1.5 \cdot \bar{e} \\ 0 & \text{otherwise} \end{cases}$$

As the individual facial regions are subject to different requirements, the applied regularization weights differ. Table 1 presents the weights for the individual regions. Please note that while the weights differ for the individual face regions, they are the same for all avatars. No subject-specific fine-tuning of regularization parameters is required while still being able to reconstruct a wide set of structures (e.g. long and short hair).

Detailed Appearance Objective E_{app} (Eq. 3) The appearance energy term is defined in Eq. 3 in the main paper as:

$$E_{\text{app}} = w_{\text{phot}} \cdot E_{\text{phot}} + w_{\text{perc}} \cdot E_{\text{perc}}. \tag{3}$$

The photo-metric energy term E_{phot} is defined on the intersection $\mathcal{V} = \mathcal{S} \cap \hat{\mathcal{S}}$ of the foreground segmentation of the input I and the region that is generated by our head model $\hat{I}$:

$$E_{\text{phot}} = |\mathcal{V} \cdot (\hat{I} - I)|_1.$$

To generate sharp textures [11, 14], we employ the style-based perceptual energy term E_{perc} proposed by [10].

Initialization and Optimization. As discussed in Section 3.2 of the main paper, texture and geometry have to be initialized before starting the joint optimization against RGB sequences to avoid converging to bad local minima. To this end, the FLAME head model is aligned with the training sequence to obtain a coarse geometry initialization. Based on this coarse geometry, the weights of $\mathcal{G}$ and the frame-specific parameters of the FLAME model, i.e., pose and expressions are optimized w.r.t. E_{geom} (Eq. (2) in main paper). As a result of this optimization, we obtain a geometry estimate that aligns well with the target silhouette. Using this refined geometry, $\mathcal{T}$ is trained to minimize E_{app} (Eq. (3) in main paper). In a final optimization step, all components are optimized jointly to minimize E_{joint} (Eq. (1) in main paper).

Given the 750 input frames of our real sequences, we initialize $\mathcal{G}$ for 150 epochs, $\mathcal{T}$ is initialized for another 100 epochs, and we jointly optimize both components as long as the perceptual loss on the holdout validation set decreases. This takes approximately 50 epochs. For longer sequences, we reduce the epoch count such that the number of iterations per optimization stage remains the same.

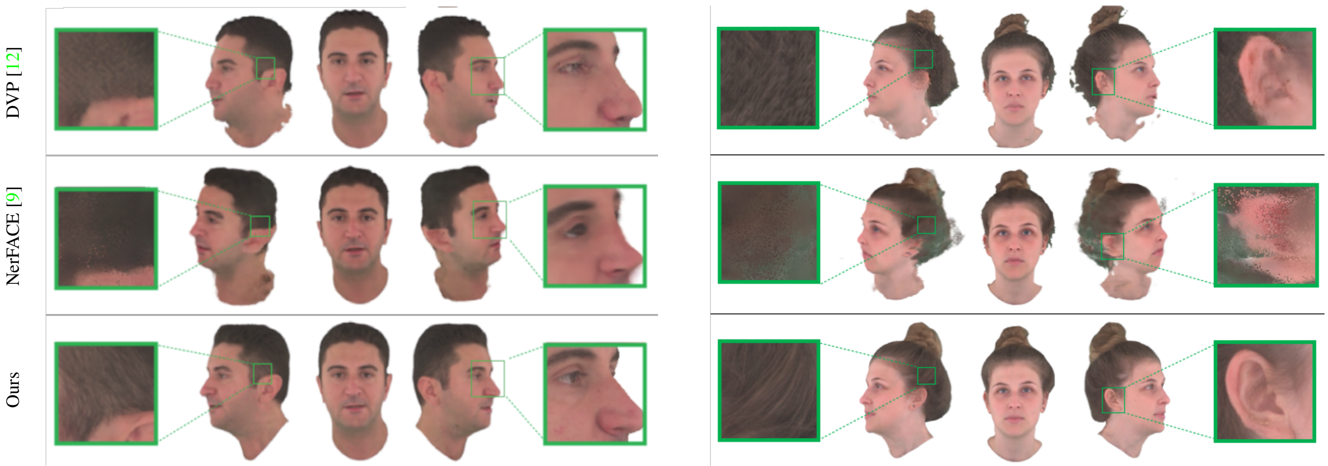


Figure 4. Additional qualitative novel view synthesis comparisons. We observe that the advantages of our method, namely spatial consistency and high texture detail even under extreme head rotations, apply to a variety of identities. Also see Figure 1 for additional subjects.

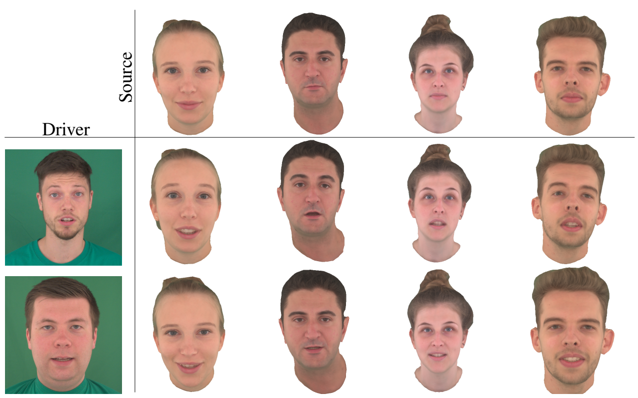


Figure 5. Expressions ψ and poses ϕ from the driving frames on the left are added to the neutral poses of optimized source avatars at the top. The reenacted avatars are displayed below.

We deploy a standard Adam optimizer [13] for all frame-agnostic parameters and a standard SGD optimizer for frame-specific components (e.g. expression and pose parameters). Weight decay is applied to all texture-related components. The entire pipeline is implemented with Pytorch [18] and Pytorch3D [20].

2. Additional Results

2.1. Avatar Reenactment

Neural Head Avatars are controllable by the disentangled pose and expression spaces of the FLAME head model. This naturally enables the reenactment of an optimized avatar via a driving sequence. We demonstrate this capability in Figure 5. In this experiment, we utilize expressions ψ and poses ϕ from a driver's video and add them to the neutral expression of optimized avatars. ψ and ϕ are extracted using our tracking algorithm discussed in the main paper.

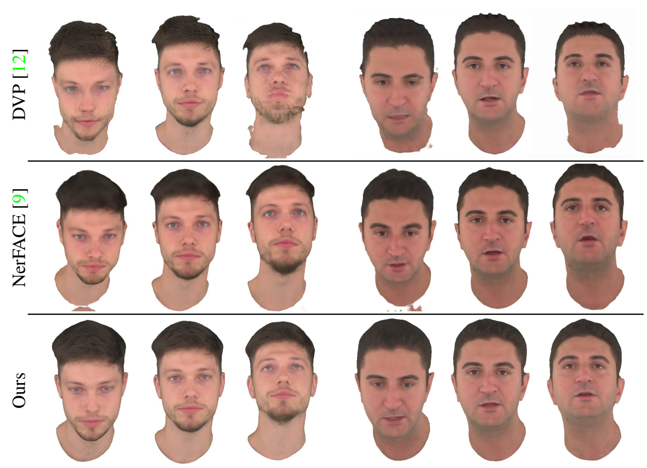


Figure 6. Qualitative novel view synthesis comparison for pitch rotations. We report the frontal view as well as synthesis results under a pitch angle of $\pm 15°$.

We observe that our method is able to faithfully transfer pose and expression between various subjects.

2.2. Novel View Synthesis

In Section 4.4 of the main paper, we qualitatively compare the synthesis from novel viewpoints against related methods. Figure 6 and Figure 4 provide further qualitative comparisons on different subjects and demonstrate that the advantages of our method apply to a variety of identities.

We also provide a quantitative evaluation of the novel view synthesis results in Figure 7. For a quantitative analysis, given that no ground truth is available for novel views, we evaluate the cosine similarity (CSIM) of the latent feature vectors predicted by a pretrained face recognition network [3] between front-facing ground truth and novel view prediction. We report the CSIM scores averaged over all

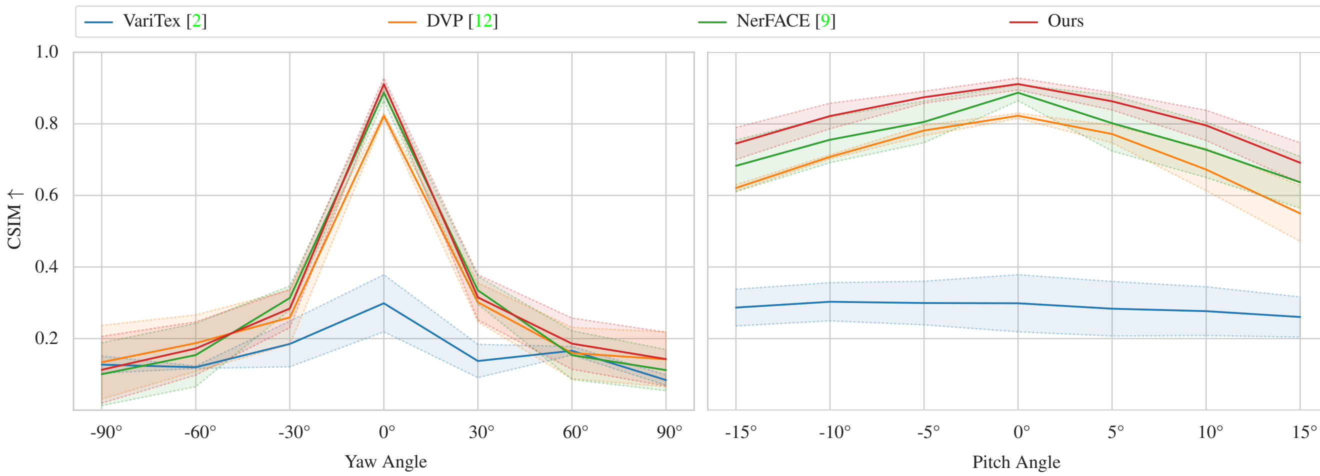


Figure 7. Quantitative novel view synthesis comparison. We report the cosine similarity score between latent features predicted by a pretrained face recognition network [6]. The feature vectors are compared between the front facing ground truth and the predicted image under different rotation angles. We report the average scores over all validation frames from two sequences in our real dataset together with the respective 1σ regions. Our method consistently outperforms related approaches for pitch angles between -15° and +15°. For extreme yaw rotations, we observe significantly reduced CSIM scores even though qualitative comparisons demonstrate high identity preservation under these conditions (see Figure 7 in main paper).

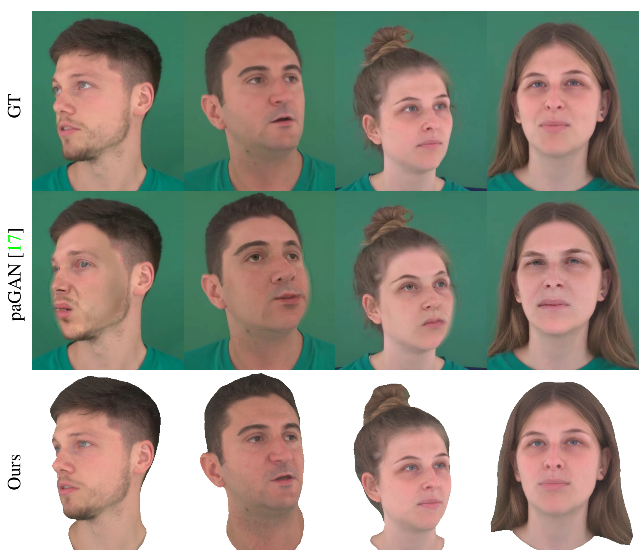


Figure 8. Qualitative comparison with paGAN [17] on the validation sequences. Note that only the facial region is synthesized by paGAN.

validation frames of two sequences in our real dataset. We find that our method outperforms related approaches consistently for pitch angles in a range of $\pm 15°$. However, we observe that for large yaw angles ($\geq \pm 30°$), the scores for all considered models decrease rapidly. Still, qualitative comparisons demonstrate that our method exhibits high identity preservation even under viewpoint changes in that range.

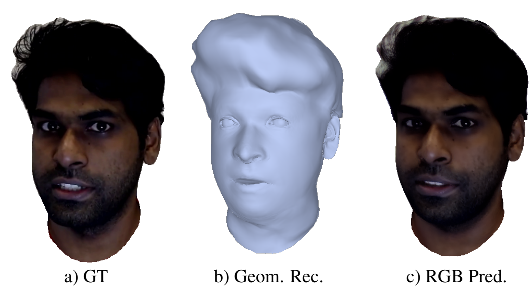


Figure 9. Synthesis results for a non-caucasian subject. The high quality of the synthesis results of our method is consistent also for people of color.

2.3. Comparison with paGAN [17]

In Figure 8, we compare our generated images of the real identities with the outputs of paGAN [17] provided by the authors. As their results are overlays on top of the original video, only the synthesized parts (facial region) should be considered when comparing to it.

2.4. Synthesis Results for Non-Caucasian Subjects

To validate that our method also performs well for non-caucasian subjects, we include synthesis results for a person of color in Figure 9. Also in this case, our method reconstructs the head geometry faithfully and renders visually plausible images.

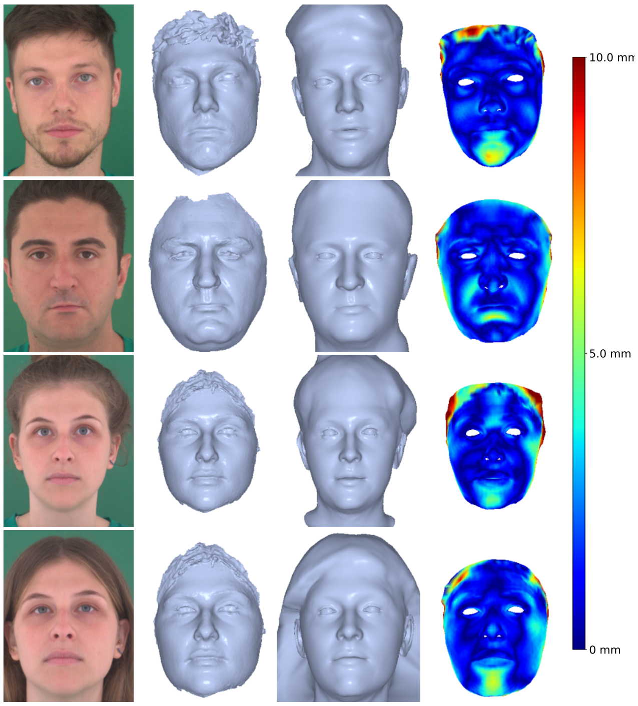


Figure 10. Geometry evaluation on the real identities. The right most column visualizes the Hausdorff distance from our predicted face mesh to the recorded GT. From top to bottom, the total alignment errors of the identities are 1.5, 1.6, 1.6, and 1.6 mm.

2.5. Geometry Evaluation

We compare the geometry predicted by our approach on the real dataset with multi-view stereo (MVS) recordings of the respective identities, see Figure 10. The MVS data was captured separately with a handheld DSLR camera. As hair styles and face dynamics can not be reproduced reliably in separate recordings, only the neutral pose of the face region is compared.

2.6. Energy Ablation

Figure 11 demonstrates the effect of further energy terms on the synthesis results. We observe that E_{phot} prevents color shifts, E_{semantic} improves alignment of overlapping semantic regions within the avatar's silhouette and E_{perc} results in textures with more detail.

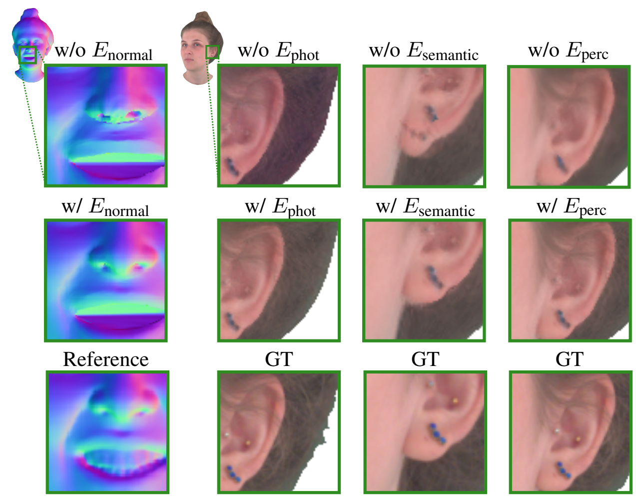


Figure 11. Energy Term Ablation. Reference normals are the inputs we use for optimization.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 3

[2] Marcel C Bühler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. Varitex: Variational neural face textures. *arXiv preprint arXiv:2104.05988*, 2021. 6

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 3, 5

[4] Andrei Burov, Matthias Nießner, and Justus Thies. Dynamic surface function networks for clothed human bodies. *arXiv preprint arXiv:2104.03978*, 2021. 1, 4

[5] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proc. CVPR*, 2021. 2

[6] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Proceedings of the IEEE Conference on European Conference on Computer Vision*, 2020. 6

[7] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 3(7):e11, 2018. 2

[8] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. volume 40, 2021. 3

[9] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. 2021. 5, 6

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 4

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4

[12] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 5, 6

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[14] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016. 4

[15] 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

[16] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 3

[17] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. pagan: real-time avatars using dynamic textures. pages 1–12, 12 2018. 6

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[19] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[20] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 5

[21] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 2

[22] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 3

[23] zllrunning. Face parsing pytorch. `https://github.com/zllrunning/face-parsing.PyTorch`, 2021. 3