**I M Avatar: Implicit Morphable Head Avatars from Videos**

Yufeng Zheng\(^1,3\) Victoria Fernández Abrevaya\(^2\) Marcel C. Bühler\(^1\) Xu Chen\(^1,3\)
Michael J. Black\(^2\) Otmar Hilliges\(^1\)
\(^1\)ETH Zürich \(^2\)Max Planck Institute for Intelligent Systems, Tübingen
\(^3\)Max Planck ETH Center for Learning Systems

**Abstract**

Traditional 3D morphable face models (3DMMs) provide fine-grained control over expression but cannot easily capture geometric and appearance details. Neural volumetric representations approach photorealism but are hard to animate and do not generalize well to unseen expressions. To tackle this problem, we propose IMavatar (Implicit Morphable avatar), a novel method for learning implicit head avatars from monocular videos. Inspired by the fine-grained control mechanisms afforded by conventional 3DMMs, we represent the expression- and pose-related deformations via learned blendshapes and skinning fields. These attributes are pose-independent and can be used to morph the canonical geometry and texture fields given novel expression and pose parameters. We employ ray marching and iterative root-finding to locate the canonical surface intersection for each pixel. A key contribution is our novel analytical gradient formulation that enables end-to-end training of IMavatars from videos. We show quantitatively and qualitatively that our method improves geometry and covers a more complete expression space compared to state-of-the-art methods. Code and data can be found at https://ait.ethz.ch/projects/2022/IMavatar/.

1. **Introduction**

Methods to automatically create animatable personal avatars in unobtrusive and readily available settings (i.e., from monocular videos) have many applications in VR/AR games and telepresence. Such applications require faithful renderings of the deforming facial geometry and expressions, detailed facial appearance and accurate reconstruction of the entire head and hair region. Conventional methods [14, 16, 18, 19, 43, 45, 47, 49] based on morphable mesh models [2, 28, 38] can fit 3DMM shape and texture parameters to images for a given subject. However, such mesh-based approaches suffer from an inherent resolution-memory trade-off, and cannot handle topological changes caused by hair, glasses, and other accessories. Recent methods build on neural radiance fields [32] to learn personalized avatars [17, 36, 37] and yield high-quality images, especially if the generated expressions are close to the training data. A key challenge for building animatable facial avatars with implicit fields is the modeling of deformations. Previous work either achieves this by conditioning the implicit representation on expressions [17] or via a separate displacement-based warping field [36, 37]. Such under-constrained formulations limit the generalization ability, requiring large numbers of training poses.

In this paper, we propose Implicit Morphable avatar (IMavatar), a novel approach for learning personalized,
learned via FLAME \[28\], we represent the deformation fields on texture, and pose- and expression-related deformations. In multilayer perceptrons (MLPs), that represent the geometry, the fine-grained expression control provided by 3DMMs with the high-fidelity geometry and texture details offered by resolution-independent implicit surfaces, taking advantage of the strengths of both methods. IMavatars are modeled via three continuous implicit fields, parametrized by neural networks that control the underlying expression blendshapes as well. Yenamandra \[23\] and Ramon \[41\] propose an implicit morphable model that decouples shape, expression, appearance, and hair style. The seminal work of Blanz and Vetter \[2\] used principal component analysis (PCA) to model facial appearance and geometry on a low-dimensional linear subspace, known as the 3D Morphable Model (3DMM). Extensions include multilinear models for shape and expression \[5, 56\], full-head PCA models \[12, 28, 39\], deep non-linear models \[42, 54\] and fully articulated head models with linear blend skinning (LBS) and corrective blendshapes \[28\]. 3DMM and its variants have been widely used within optimization-based \[19, 43, 46, 52\] and deep learning-based approaches \[14–16, 27, 29, 45, 47, 49–51\]. These methods can obtain overall accurate estimates of the facial area but typically lack details, do not model the entire head, and cannot properly represent eyes, teeth, hair, and accessories.

A related line of research estimates personalized rigs from monocular input, i.e. 3D representations of the head along with a set of controls that can be used for animation. This has been traditionally addressed by recovering a personalized set of blendshape bases, obtained through deformation transfer \[6, 18, 21, 22, 48\] or deep neural networks \[1, 8, 59\]. To represent detailed geometry, a few methods additionally reconstruct a layer of mid- or fine-level correctives that can be deformed along with the underlying coarse mesh \[16, 18, 22, 59\]. Here, we present a new approach that can recover a higher-fidelity facial rig than prior work, and is controlled by expression blendshapes as well as jaw, neck, and eye pose parameters.

Neural Face Models. The recent success of neural implicit shape representations \[10, 26, 30, 33, 35, 60\] has led to several methods that build 3D facial models within this paradigm. Yenamandra \textit{et al.} \[61\] propose an implicit morphable model that decouples shape, expression, appearance, and hair style. These representations enable the encoding of thin structures such as hair, and they can model complex surface/light

In summary, we contribute:

- a 3D morphing-based implicit head avatar model with detailed geometry and appearance that generalizes across diverse expressions and poses,
- a differentiable rendering approach that enables end-to-end learning from videos, and
- a synthetic video dataset for evaluation.

2. Related work

3D Face Models and Avatar Reconstruction. Estimating 3D shape from monocular input is an ill-posed problem, traditionally addressed by using data-based statistical priors. The seminal work of Blanz and Vetter \[2\] used principal component analysis (PCA) to model facial appearance and geometry on a low-dimensional linear subspace, known as the 3D Morphable Model (3DMM). Extensions include multilinear models for shape and expression \[5, 56\], full-head PCA models \[12, 28, 39\], deep non-linear models \[42, 54\] and fully articulated head models with linear blend skinning (LBS) and corrective blendshapes \[28\]. 3DMM and its variants have been widely used within optimization-based \[19, 43, 46, 52\] and deep learning-based approaches \[14–16, 27, 29, 45, 47, 49–51\]. These methods can obtain overall accurate estimates of the facial area but typically lack details, do not model the entire head, and cannot properly represent eyes, teeth, hair, and accessories.
Given a pixel location, our method performs ray marching in the deformed space. For each deformed point $x_d^i$, we conduct correspondence search to find the corresponding canonical point $x_c^i$. Our novel implicit morphing leverages the canonical blendshape and skinning-weight fields $E$, $W$ and $P$ to morph the canonical point $x_c^i$ to its deformed location $x_d^i$ given expression and pose conditions. After finding the nearest canonical surface intersection $x_c$, our novel analytical gradient formula allows efficient computation of gradients for the geometry and deformation fields. Finally, we predict the RGB values by querying the canonical texture network. We use an image reconstruction loss and a mask loss, and optionally supervise the predicted blendshapes and skinning-weights.

**Implicit deformation fields.** Modeling dynamic objects with implicit neural fields is an active research topic, with currently three main approaches. The first is to condition each frame on a latent code, e.g., a time stamp [58], a learned latent vector [36,57], or a vector from a pre-computed parametric model [13, 17, 31, 44]. A second, possibly complementary approach, is to use a “backward” deformation field. This is an additional neural network that maps observations in deformed space to observations in canonical space, where the implicit function is then evaluated [34, 36, 36, 40, 44, 53, 55]. The deformation fields are modeled as velocity fields [34], translation fields [40, 53], rigid transformations [36], or skinning-weight fields [23, 31, 44]. While they have demonstrated an impressive capacity for learning correspondences even in the absence of ground-truth supervision, the backwards formulation makes them pose-dependent and hence requires a large dataset for learning, showing reduced generalization when deformations are too far away from the training set. To tackle this, forward deformation fields have been recently proposed [9]. These learn a continuous forward skinning weight field, and corresponding canonical points are found using iterative root finding. To improve generalization outside the scope of train-time expressions, here we extend the idea of forward skinning to the problem of facial deformations and propose a new analytical gradient formula that allows the deformation fields to be directly learned from videos.

**3. Method**

We propose IMavatar, an implicit morphable head avatar that equips implicit surfaces with fine-grained expression control by leveraging morphing-based deformation fields. In this section, we first recap the deformation formulation of the FLAME face model [28], followed by the representations for the canonical geometry, deformation, and texture fields. Then, we introduce correspondence search to find canonical points for image pixels and derive the analytical gradients for end-to-end training.

**3.1. Recap: FLAME Face Morphable Model**

The FLAME face model [28] parameterizes facial geometry with shape, pose, and expression components. Since we focus on personal facial avatars, we specifically represent the pose- and expression-dependent shape variations.
The simplified FLAME mesh model is denoted by:

\[ M(\theta, \psi) = LBS(T_P(\theta, \psi), J(\psi), \theta, W), \]  

(1)

where \( \theta \) and \( \psi \) denote the pose and expression parameters, and \( LBS(\cdot) \) and \( J(\cdot) \) define the standard skinning function and the joint regressor, respectively. \( W \) represents the per-vertex skinning weights for smooth blending, and \( T_P \) denotes the canonical vertices after adding expression and pose correctives, represented as:

\[ T_P(\theta, \psi) = \bar{T} + B_E(\psi; \mathcal{E}) + B_P(\theta; \mathcal{P}), \]  

(2)

where \( \bar{T} \) is the personalized canonical template. \( B_P(\cdot) \) and \( B_E(\cdot) \) calculate the additive pose and expression offsets using the corrective blendshape bases \( \mathcal{P} \) and \( \mathcal{E} \) given the animation conditions \( \theta \) and \( \psi \). Our method extends the discrete \( \mathcal{W}, \mathcal{E} \), and \( \mathcal{P} \) defined on vertices to be continuous fields represented by MLPs, making it possible to morph continuous canonical representations.

3.2. IMavatar

IMavatar is represented by three neural implicit fields, defining the canonical geometry, deformation bases, and texture of the person, as shown in Fig. 2. Details of the network architecture can be found in the Sup. Mat.

**Geometry.** We represent the canonical geometry using an MLP that predicts the occupancy values for each canonical 3D point. We additionally condition the geometry network \( f_{\sigma_j} \) on a per-frame learnable latent code \( l \in \mathbb{R}^{n_l} \), similar to NerFace [17], and leverage positional encoding [33] to encourage high frequency details in the canonical geometry. We represent the canonical geometry using an MLP that predicts the occupancy values for each canonical point, as shown in Fig. 2. Details of the method extending the discrete \( \mathcal{W}, \mathcal{E} \), and \( \mathcal{P} \) defined on vertices to be continuous fields represented by MLPs, making it possible to morph continuous canonical representations.

\[ f_{\sigma_j}(x, l) : \mathbb{R}^3 \times \mathbb{R}^{n_l} \rightarrow \text{occ}. \]  

(3)

**Deformation.** Following FLAME [28], our deformation network \( d_{\sigma_d} \) predicts the additive expression blendshape vectors \( \mathcal{E} \in \mathbb{R}^{n_e \times 3} \), the pose correctives \( \mathcal{P} \in \mathbb{R}^{n_p \times 9 \times 3} \), and the linear blend skinning weights \( \mathcal{W} \in \mathbb{R}^{n_v} \) for each point in the canonical space, where \( n_e \) and \( n_j \) denote the number of expression parameters and bone transformations.

\[ d_{\sigma_d}(x) : \mathbb{R}^3 \rightarrow \mathcal{E}, \mathcal{P}, \mathcal{W}. \]  

(4)

In a slight abuse of notation we reuse \( \mathcal{E}, \mathcal{P}, \text{and } \mathcal{W} \) from FLAME – please note that these denote continuous implicit fields from here on. For each canonical point \( x_c \), the transformed location \( x_d := w_{\sigma_d}(x_c) \) is:

\[ x_d = LBS(x_c + B_P(\theta; \mathcal{P}) + B_E(\psi; \mathcal{E}), J(\psi), \theta, W), \]  

(5)

where \( J \) is the joint regressor from FLAME. This defines the forward mapping from canonical points \( x_c \) to deformed locations \( x_d \). We detail the computation of the inverse mapping from deformed to canonical space in Sec. 3.3.

**Normal-conditioned texture.** We leverage a texture MLP \( c_{\sigma} \) to map each location in the canonical space to an RGB color value. To account for non-uniform lighting effects, we additionally condition the texture network on the normal direction of the deformed shape. For implicit surfaces, the normal direction can be calculated as the normalized gradient of the occupancy field w.r.t. the 3D location. In our case, the gradient of the deformed shape is given by:

\[ \frac{\partial f_{\sigma_j}(x_c)}{\partial x_d} = \frac{\partial f_{\sigma_j}(x_c)}{\partial x_c} \frac{\partial x_c}{\partial x_d} = \frac{\partial f_{\sigma_j}(x_c)}{\partial x_c} \left( \frac{\partial w_{\sigma_d}(x_c)}{\partial x_c} \right)^{-1}. \]  

(6)

Since the appearance in the mouth region cannot be modeled purely by warping due to dis-occlusions [36], our final predicted color \( c \) is calculated from the canonical location \( x_c \), the normal direction of the deformed shape \( n_d \), and the jaw pose and expression parameters \( \theta \) and \( \psi \).

\[ c_{\sigma}(x_c, n_d, \theta, \psi) : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{50} \rightarrow c. \]  

(7)

3.3. Differentiable Rendering

To optimize the canonical networks from videos with expressions and poses, we first introduce non-rigid ray marching to find the canonical surface point for each ray, and introduce analytical gradients that enable end-to-end training of the geometry and deformation networks.

**Non-rigid ray marching.** Given a camera location \( r_o \) and a ray direction \( r_d \) in the deformed space, we follow IDR [60] and perform ray marching in the deformed space. To determine the occupancy values for the sampled points \( x_d \), we follow SNARF [9] and leverage iterative root finding to locate the canonical correspondences \( x_c \) and query their occupancy values. Thus, we can locate the nearest canonical surface intersection for each ray iteratively.

**Gradient.** To avoid back-propagation through the iterative process, we derive analytical gradients for the location of the canonical surface point \( x_c \), leveraging that \( x_c \) must satisfy the surface and ray constraints:

\[ f_{\sigma_j}(x_c) = 0.5, \quad (w_{\sigma_d}(x_c) - r_o) \times r_d = 0, \]  

(8)

(9)

where 0.5 is defined as the level set for the surface. For convenience, we rewrite this equality constraint as \( F_{\sigma_j}(x_c) = 0 \), which implicitly defines the canonical surface intersection \( x_c \). The learnable parameters of the geometry and deformation networks are \( \sigma_F = \sigma_P \cup \sigma_d \). We leverage implicit differentiation to attain the gradient of \( x_c \) w.r.t. the param-
eners of the geometry and deformation networks:

\[
\frac{\partial F_{\sigma_p}(x_c)}{\partial \sigma_F} = 0 \\
\Leftrightarrow \frac{\partial F_{\sigma_p}(x_c)}{\partial \sigma_F} + \frac{\partial F_{\sigma_p}(x_c)}{\partial x_c} \frac{\partial x_c}{\partial \sigma_F} = 0 \quad (10)
\]

We also supervise non-surface rays with a mask loss (Eq. 12), in which case the equality constraint is defined as \( w_{\sigma_d}(x_c) = x^*_d \), where \( x^*_d \) is the point on the ray with the smallest occupancy value, located via ray sampling.

3.4. Training Objectives

**RGB loss** supervises the rendered pixel color:

\[
L_{\text{RGB}} = \frac{1}{|P|} \sum_{p \in P^{\text{in}}} \| C_p - c_{\sigma_c}(x_c) \|_1, \quad (11)
\]

where \( P \) denotes the set of training pixels, and \( P^{\text{in}} \subset P \) denotes the foreground pixels where a ray-intersection has been found. \( C_p \) and \( c_{\sigma_c}(x_c) \) represent the ground-truth and predicted RGB values of pixel \( p \). The analytical gradient formula enables \( L_{\text{RGB}} \) to not only optimize texture, but also the geometry and deformation networks.

**Mask loss** supervises the occupancy values for non-surface rays \( p \in P \setminus P^{\text{in}} \):

\[
L_M = \frac{1}{|P|} \sum_{p \in P \setminus P^{\text{in}}} CE(O_p, f_{\sigma_f}(x^*_c)), \quad (12)
\]

where \( CE(\cdot) \) is the cross-entropy loss calculated between the ground-truth \( O_p \) and predicted occupancy values \( f_{\sigma_f}(x^*_c) \). The mask loss also optimizes the deformation network thanks to the analytical gradient.

An **optional FLAME loss** leverages prior knowledge about expression and pose deformations from FLAME [28] by supervising the deformation network with the corresponding values of the nearest FLAME vertices:

\[
L_{FL} = \frac{1}{|P|} \sum_{p \in P^{\text{in}}} \left( \lambda_e \| \mathcal{E}_p^{\text{GT}} - \mathcal{E}_p \|_2 + \lambda_p \| P_p^{\text{GT}} - P_p \|_2 + \lambda_w \| W_p^{\text{GT}} - W_p \|_2 \right), \quad (13)
\]

where \( \mathcal{E}_p, P_p, \) and \( W_p \) denote the predicted values of the deformation network, and \( \mathcal{E}_p^{\text{GT}}, P_p^{\text{GT}}, \) and \( W_p^{\text{GT}} \) denote the pseudo ground truth defined by the nearest FLAME vertices. We set \( \lambda_e = \lambda_p = 1000, \) and \( \lambda_w = 0.1 \) for our experiments. Our final training loss is:

\[
L = L_{\text{RGB}} + \lambda_M L_M + \lambda_{FL} L_{FL}, \quad (14)
\]

where \( \lambda_M = 2 \) and \( \lambda_{FL} = 1 \).

Figure 3. **Qualitative results on synthetic data.** As the expression strength increases from left to right, baseline methods either collapse to a neutral expression (D-Net, B-Morph) or produce invalid geometry (C-Net, Fwd-Skin). In contrast, our method successfully handles even the most extreme expressions.

4. Experiments

This section empirically evaluates the benefits of the proposed approach in terms of geometry accuracy and expression generalization. We conduct experiments on both synthetic data with known geometry and real video sequences.

4.1. Datasets

**Synthetic Dataset.** We conduct controlled experiments on a synthetic dataset by rendering posed and textured FLAME meshes. For the training set, we render a video that is representative of a speech sequence. We take FLAME expression parameters from the VOCA speech dataset [11] and head poses fitted from real videos. We build the test set with unseen, stronger expressions extracted from COMA [42]. Our synthetic dataset consists of 10 subjects with varied facial shapes and appearance, with an average of 5,368 frames for training and 1876 frames for testing per subject. For testing, we subsample every 10th frame. We release the synthetic dataset for research purposes.

**Real Video Dataset.** We evaluate on real videos from a single stationary camera. We calculate foreground masks with
MODNet [25] and estimate the initial FLAME parameters using DECA [16], which are refined by fitting to 2D facial keypoints [4]. Please see Sup. Mat. for more details. The real video dataset consists of 4 subjects, with roughly 4,000 frames for training and 1,000 frames for testing per subject. The training videos cover mostly neutral expressions in a speech video, while the test videos include unseen, difficult expressions such as jaw opening, big smiles, and more. We subsample every 10th frame for testing.

4.2. Ablation Baselines

This paper tackles the key difficulty in building animatable avatars: capturing the per-frame deformations with respect to the canonical shape. We compare our method with the commonly used previous approaches by replacing our deformation module with the following alternatives:

**Pose- and expression-conditioned network (C-Net).** C-Net is inspired by NerFACE [17] but is designed for implicit surfaces. It first applies a rigid transformation to the deformed shape, which brings the full upper body to the canonical space with the inverse head pose transformation; it then models other deformations by conditioning on the pose and expression parameters.

**Displacement warping (D-Net).** D-Net uses a deformation network with pose and expression parameters as input, and predicts displacement vectors for deformed points, warping them to the canonical space. The predicted displacements are supervised with FLAME, similar to Eq. 13.

**Backward morphing (B-Morph).** B-Morph leverages the morphing formulation of FLAME and predicts expression blendshapes, pose corrective vectors, and LBS weights. However, the deformation network is conditioned on the deformed location as well as pose and expression parameters and performs backward morphing. In contrast, our deformation network takes as input only the canonical point, which is pose- and expression-independent, enabling better generalization [9]. The learned blendshapes and weights of this baseline are supervised by FLAME pseudo GT.

**Forward skinning + expression-conditioning (Fwd-Skin).** This baseline is adapted from SNARF [9], which was originally proposed for human body avatars. Here, the deformation network only models the LBS weights, while the expression and pose-related correctives are handled by conditioning on the geometry and texture networks.

**IMAvatar unsupervised (Ours-).** This baseline eliminates the FLAME pseudo GT supervision, learning solely from images and masks (only used for exp. with real data).

4.3. Metrics

The goal of this work is to obtain an animatable 3D head from a video, and hence we evaluate the geometric accuracy

<table>
<thead>
<tr>
<th>Method</th>
<th>Expression ↓</th>
<th>Normals ↓</th>
<th>$L_1$ ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-Net</td>
<td>3.248</td>
<td>9.108</td>
<td>0.02245</td>
<td>26.67</td>
<td>0.9829</td>
<td>0.02812</td>
</tr>
<tr>
<td>D-Net</td>
<td>7.452</td>
<td>26.174</td>
<td>0.07881</td>
<td>19.62</td>
<td>0.9832</td>
<td>0.03053</td>
</tr>
<tr>
<td>B-Morph</td>
<td>4.941</td>
<td>12.150</td>
<td>0.03293</td>
<td>24.95</td>
<td>0.9726</td>
<td>0.03340</td>
</tr>
<tr>
<td>Fwd-Skin</td>
<td>2.825</td>
<td>8.130</td>
<td>0.01920</td>
<td>27.30</td>
<td>0.9852</td>
<td>0.02812</td>
</tr>
<tr>
<td>Ours</td>
<td>2.558</td>
<td>5.901</td>
<td>0.01807</td>
<td>28.75</td>
<td>0.9900</td>
<td>0.01581</td>
</tr>
</tbody>
</table>

Table 1. Quantitative results for synthetic experiment. Compared to the baselines, our method achieves more consistent surface normals, better image quality, and more accurate expressions.

![Image](309x550 to 545x617)

Figure 4. Expression extrapolation. Performance of baseline methods worsen drastically as expressions become more extreme (higher norm). *Geom. error* denotes the angular error of the surface normals (lower is better, see Sec. 4.3).

(only available for the synthetic dataset), image quality, and expression fidelity. Image quality is measured via the Manhattan distance ($L_1$), SSIM, PSNR and LPIPS [63] metrics, following the practice in NerFACE [17]. To measure geometric consistency for synthetic data, we report the average angular normal error between the generated normal map and ground truth, denoted as the *Normals* metric in Tab. 1. Since we focus on modeling deformation-related geometry and texture, both normal consistency and image similarity metrics are measured in the face interior region. For both synthetic and real data, we measure the expression fidelity by calculating the distance between generated and (pseudo) GT facial keypoints. We estimate the facial keypoints of predicted images with [4], and (pseudo) GT keypoints are obtained from posed FLAME meshes.

4.4. Results on Synthetic Dataset

We train IMAvatar and baseline methods for the 10 synthetic identities and measure geometry, expression and image reconstruction errors on 12 sequences with renderings from the COMA dataset. We outperform all baselines by a large margin on all metrics (Tab. 1).

**Extrapolation.** While other methods are limited to interpolation, our method is capable of extrapolating beyond seen expressions and poses. In Fig. 4, we plot the geometric error for different strength of expressions. Most methods perform well for mild expressions (small expression norm). For stronger expressions, however, their errors increase significantly. In contrast, our method only incurs a slight increase even for strong expressions (large norm). See Sup. Mat. for an analogous plot for the jaw pose. Figure 3 shows visual examples for neutral, medium, and strong expressions.
This finding is also supported by Tab. 2. We show in Sup. Mat. that the FLAME loss can be replaced via more training data without losing accuracy.

Figure 5, demonstrates control over expressions and poses by interpolating and extrapolating an example expression (first expression component in FLAME [28]), plus jaw (pitch), and neck (yaw) poses separately. For each parameter, we show generated images and the training data distribution with 5 vertical lines corresponding to the 5 samples. This shows that our method generalizes to expressions and poses far beyond the training distribution. More interpolation and extrapolation examples can be found in Sup. Mat.

5. Conclusion

We propose IMAvatar, an implicit morphable head avatar, controlled via expression and pose parameters in a similar way to 3DMMs, but with the ability to model diverse and detailed hairstyles and facial appearance. Our method—learned end-to-end from RGB videos—demonstrates accurate deforming geometry and extrapolates to strong expressions beyond the training distribution.

While our method contributes towards building controllable implicit facial avatars, some challenges remain. First, surface representations achieve detailed facial geometry, but they cannot model the fine occlusions produced by hair. Future work could address this by combining volumetric representations [33] with animatable surfaces. Second, the iterative non-rigid ray marching makes IMAvatar slow to train (∼ 2 GPU days). Initializing with mesh-ray intersections could speed up the process, as done in concurrent work [24]. Third, our method relies on accurate face tracking and our performance degenerates with noisy 3DMM parameters (See Sup. Mat.). Refining the poses and expressions during training is a promising future direction. Finally, the appearance in the mouth interior region can be

<table>
<thead>
<tr>
<th>Method</th>
<th>Expression ↓</th>
<th>$L_1$ ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-Net</td>
<td>3.615</td>
<td>0.05824</td>
<td>22.23</td>
<td>0.9524</td>
<td>0.03421</td>
</tr>
<tr>
<td>D-Net</td>
<td>3.769</td>
<td>0.06130</td>
<td>21.77</td>
<td>0.9474</td>
<td>0.03227</td>
</tr>
<tr>
<td>B-Morph</td>
<td>2.786</td>
<td>0.04980</td>
<td>23.50</td>
<td>0.9599</td>
<td>0.02231</td>
</tr>
<tr>
<td>Fwd-Skin</td>
<td>3.088</td>
<td>0.05456</td>
<td>22.92</td>
<td>0.9586</td>
<td>0.02784</td>
</tr>
<tr>
<td>NerFACE</td>
<td>2.994</td>
<td>0.04564</td>
<td>23.58</td>
<td>0.9586</td>
<td>0.02156</td>
</tr>
<tr>
<td>Ours-</td>
<td>2.843</td>
<td>0.04918</td>
<td>23.68</td>
<td>0.9615</td>
<td>0.02155</td>
</tr>
<tr>
<td>Ours</td>
<td>2.548</td>
<td>0.04878</td>
<td>23.91</td>
<td>0.9655</td>
<td>0.02085</td>
</tr>
</tbody>
</table>

Table 2. Quantitative results on real videos. We compare our method with the SOTA and baselines on test sequences with unseen expressions and poses. Our method reconstructs the expressions more accurately while being on par in terms of image quality.
unrealistic (last two examples in Fig. 6). We propose an improvement in the Sup. Mat. We discuss potential negative societal impact in light of disinformation and deep-fakes in the Sup. Mat.

Acknowledgements: We thank the authors of [20] for sharing their evaluation dataset. Yufeng Zheng and Xu Chen were supported by the Max Planck ETH Center for Learning Systems. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program grant agreement No 717054. MJB has received research gift funds from Adobe, Intel, Nvidia, Meta/Facebook, and Amazon. MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH. MJB’s research was performed solely at, and funded solely by, the Max Planck.
References


