DOI: 10.1111/cgf.14705
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

Learning Dynamic 3D Geometry and Texture for Video Face Swapping

C. Otto[1,2], J. Naruniec[1], L. Helminger[1,2], T. Etterlin[2], G. Mignone[1], P. Chandran[1,2],
G. Zoss[1], C. Schroers[1], M. Gross[1,2], P. Gotardo[1], D. Bradley[1], R. Weber[1]

[1]DisneyResearch|Studios, Switzerland
[2]ETH Zürich, Switzerland

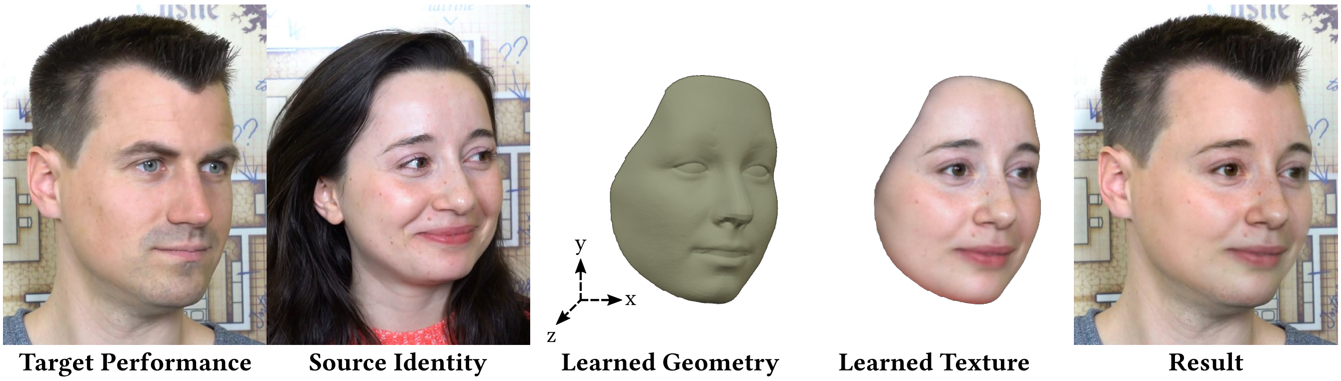

Figure 1: *We present a new method to swap the face of a target video performance with a new source identity. Our method learns dynamic 3D geometry and texture to obtain more realistic face swaps with better artistic control than common 2D approaches.*

Abstract

Face swapping is the process of applying a source actor's appearance to a target actor's performance in a video. This is a challenging visual effect that has seen increasing demand in film and television production. Recent work has shown that data-driven methods based on deep learning can produce compelling effects at production quality in a fraction of the time required for a traditional 3D pipeline. However, the dominant approach operates only on 2D imagery without reference to the underlying facial geometry or texture, resulting in poor generalization under novel viewpoints and little artistic control. Methods that do incorporate geometry rely on pre-learned facial priors that do not adapt well to particular geometric features of the source and target faces. We approach the problem of face swapping from the perspective of learning simultaneous convolutional facial autoencoders for the source and target identities, using a shared encoder network with identity-specific decoders. The key novelty in our approach is that each decoder first lifts the latent code into a 3D representation, comprising a dynamic face texture and a deformable 3D face shape, before projecting this 3D face back onto the input image using a differentiable renderer. The coupled autoencoders are trained only on videos of the source and target identities, without requiring 3D supervision. By leveraging the learned 3D geometry and texture, our method achieves face swapping with higher quality than when using off-the-shelf monocular 3D face reconstruction, and overall lower FID score than state-of-the-art 2D methods. Furthermore, our 3D representation allows for efficient artistic control over the result, which can be hard to achieve with existing 2D approaches.

CCS Concepts
• ***Computing methodologies*** → *Image manipulation; Rendering; Neural Networks;*

1. Introduction

The demand for high-quality visual effects in film and television is growing faster than ever before. Shots requiring face swapping—such as digital de-aging or virtually resurrecting a deceased performer—are becoming almost commonplace. This fact creates significant pressure on traditional visual-effects pipelines, where production time and costs are key limiting factors. While current pipelines can achieve results of stunning quality, they often require very careful (and expensive) on-set preparation, light-stage capturing, and detailed 3D modeling. On the other hand, recent progress using deep learning for face swapping [PGC*20, NHSW20] has shown remarkable results in certain scenarios, suggesting an opportunity for revolutionizing existing visual effects pipelines. This could lead to much more efficient workflows with substantial time and cost savings as well as quicker iterations during production.

While the recent developments in face swapping are very promising, there are still significant limitations. The lack of certain poses and expressions in the training data can considerably reduce the quality of the generated faces, as 2D approaches often struggle in generalizing due to their having no explicit understanding of underlying facial geometry. In addition, working in 2D also limits the level of control over the final output. Understanding and leveraging the underlying 3D geometry and facial texture for video face swapping has the potential to achieve superior generalization and better cope with scenarios in which it is not possible to find enough reference data of an actor to cover certain poses. Unfortunately, the few face swap methods that do incorporate 3D geometry (e.g. [NMT*18]) rely on fixed, non-adaptive priors. As a result, they often fail to capture the identity features that are unique to a particular face, such as characteristic nose shapes that become more noticeable in the challenging profile views.

In this paper, we address the aforementioned challenges and propose a new face-swapping approach that leverages explicit and adaptive 3D face geometry. Our new method simultaneously learns convolutional facial autoencoders for both the source and target identities, with distinct decoders and a shared encoder (*i.e.*, latent code). As a key novelty, each of our decoders first lifts the input latent code into a learned 3D representation, for better 3D modeling and generalization, before projecting the 3D face back onto the image plane using a differentiable renderer. In essence, each autoencoder reconstructs a dynamic 3D face solely by training on video images depicting the face under different viewpoints. We show that this results in more accurate 3D geometry relative to using off-the-shelf monocular 3D face reconstruction. The final result is a face swap that better reproduces actor-specific facial shapes compared to previous methods.

The main contributions of this paper are the following:

- We propose a full pipeline for performing high-quality 3D geometry-aware neural face swapping.
- We introduce a method for learning dynamic 3D geometry and texture based on a video sequence without requiring 3D supervision.
- Finally we show superiority of our approach over other state-of-the-art tools, both in terms of image quality and also the ability to artistically control the result.

2. Related Work

The vast majority of face-swapping methods operate in the 2D image domain. These methods can produce high-quality swaps, but they often struggle with faces in extreme poses, especially when the coverage of such images is insufficient in the training data. 3D-based swapping methods can utilize underlying geometry and pose information to increase performance under new, unseen views but are prone to errors in the reconstruction of the 3D model. Additionally, the access to texture and geometry that comes with 3D-based methods can allow for more natural artistic control of the swaps. In the following subsections we give an overview of both 2D and 3D face swapping methods.

2.1. 2D Face Swapping

Most successful deep-learning-based face-swapping methods rely on autoencoder architectures by mapping images from different identities into a shared latent space, similar to translation networks presented in [LBK17]. The state-of-the-art open-source face swapping method, DeepFaceLab [PGC*20], which is being used in most publicly available examples, produces realistic and detailed face swaps. Results of their autoencoder, in which two produced outputs correspond to two swapped identities, are further improved by training models with adversarial loss. An additional super-resolution post-processing step may be used to upscale low resolution outputs. Those additional techniques, despite increasing the resolution and details of the final image, often introduce characteristic, undesirable artifacts.

Naruniec et al. [NHSW20] create high-resolution 1024×1024 face swaps by using a progressive training regime [KALL17] and multiple decoders. They improve the temporal consistency of the results by introducing a landmark-stabilization procedure. Their approach still struggles with side poses, which can be noticed in the blurriness of the face edges in the network output. FSGAN [NKH19] utilizes an adversarial loss [GPAM*14] to swap arbitrary identities without requiring identity-specific training. That work relies on landmark displacements between faces of the source and target image to condition a recurrent network to produce the desired pose and expression. SimSwap [CCNG20] introduces a weak feature-matching loss to inject source identity information into the latent code while preserving other face attributes. MegaFS [ZLW*21] shows results in megapixel resolution but has some difficulties in preserving the full source identity due to the pre-trained StyleGAN2 prior [KLA*20]. In [XDW*22] the authors try to overcome the limitations caused by the pretrained GAN prior through disentangeling the latent semantics and deriving structure and appearance attributes from different decoder layers. [LWXS22] propose an end-to-end framework where attributes and identity are disentangled by dedicated encoders. In [XZH*22] a region-aware face swapping network based on GAN inversion is presented. It generates high-resolution and identity consistent swaps, although due to the StyleGAN2 prior, the method fails to handle difficult face poses.

We develop a new method that is based on the 2D method of [NHSW20]. However, we make significant alterations to their 2D method, incorporating 3D information via differentiable rendering and employing two separate decoders per identity. We use an identity specific approach as opposed to identity agnostic approaches (like [CCNG20, ZLW*21] or [NKH19]), since identity agnostic methods tend to make assumptions about the underlying appearance, like teeth or expressions, that we wish to reconstruct as faithfully as possible. For identity agnostic methods, the resulting images often have good subjective image quality but fail to preserve the characteristics of the original identity. In the case of [NHSW20], however, difficulties with extreme poses can be traced to a lack of correspondence in poses in the source and target training data or from inaccuracies in landmark alignment during preprocessing. In contrast, our novel 3D-based approach suffers less from this limitation, since its explicit 3D face model incorporates pose information which help to generalize better to novel viewpoints. Our decoders provides expression and texture updates complementary to the geometric information that is learned separately for each person. Furthermore, we will show that our proposed approach yields better swap quality (i.e. a lower FID score) than these 2D methods.

2.2. 3D-Based Face Reconstruction and Swapping

Traditionally, face swapping in the 3D domain is posed as a geometry retargeting problem, where 3D models of both the source and target identity are either scanned or hand-crafted. The target performance is reconstructed from the video, retargeted to the source geometry, and then re-rendered back into the scene [Sey19]. This process is very laborious and expensive, and even with that effort it remains challenging to achieve a thoroughly convincing final result over a full range of expressions and behavior.

Reconstruction of the 3D geometry of the face from a single image is usually performed using a parametric model, such as in 3D morphable face models [BV99, BV03], or by neural-network models such as PRNet [FWS*18]. Face swapping within parametric models can be achieved by exchanging the geometric parameters (e.g. translation, rotation and blendshape weights) and texture between 3D faces of different identities [BSVS04]. For example, Nirkin et. al. [NMT*18] use the Basel face model [PKA*09] and 3D dense face alignment (3DDFA) [ZLLL17] to fit a 3D shape to the source and target images. They then swap the texture of the source face onto the target face geometry before rendering the image. Newer 3D face reconstruction methods, such as DECA [FFBB20] and 3DDFA-V2 [GZY*20], improve upon these earlier results but still suffer from the uncertainty about the geometry when estimating it from a single image. HifiFace [WCZ*21] preserves the face shape of the source identity in the swap using a 3D face reconstruction method [DYX*19], but is constrained by the 3DMM model space. Our method predicts vertex position deltas instead of 3DMM parameters and is thus not constrained to the 3DMM model space. Similar to Dale et al. [DSJ*11] we take advantage of having a sequence of images that allow us to explicitly model the shape of the face. However, Dale et al. [DSJ*11] only replace the face without being able to keep the target performance in the process. We further estimate dynamic geometry and textures using differentiable rendering, which has been explored for different applications in the context of GANs [GPAM*14] in works such as GANFIT [GPKZ19], Olszewski et al. [OLY*17] and Nagano et al. [NSX*18]. An application that is related but distinct from face swapping is facial reenactment and puppeteering in the 3D domain [TZS*16, KCT*18, GSZ*18]. Face reenactment swaps facial expressions but does not swap the identity. As a result, it does not require relighting or blending the new face, which simplifies the task. Often, the original head pose is also preserved, making precise 3D reconstruction less important. In face swapping, the new identity must be rendered with a new expression, new lighting, and a new head pose, which makes it a more challenging task.

In contrast to current state-of-the-art face swapping methods [ZLW*21, PGC*20, NHSW20, CCNG20, NKH19, NMT*18], we learn dynamic subject-specific texture and geometry and use it to perform artistically controllable face swaps.

3. Video Face Swapping

We approach the problem of face swapping by training a convolutional autoencoder for each of the two swapped identities. The network's encoder is shared between identities, allowing for the joint representation of specific features like expression or lighting conditions, while the decoders are identity specific. Each decoder consists of two parts, one responsible for the geometry and one for the texture. We follow the terminology of [NHSW20] and refer to this architecture as a *comb model*. The generated identity is determined entirely by the choice of decoder at inference time. Note that face swap fundamentally requires decoders to generalize well, since their input code at test (swap) time comes from a different subject, not seen by the decoder during training.

A key novelty in our approach is that the decoders aim to lift the input latent code into a *3D representation*, comprising a dynamic face texture and a deformable 3D face shape. Similar to the texture, we encode the 3D shape in UV space as an XYZ (3-channel) image grid, where each pixel represents a 3D point location. This geometry image captures the entire 3D face mesh [FWS*18, GGH02] while allowing our autoencoder to process it with convolutions. The 3D face is projected back onto the input image plane under the same pose of the original face using a differentiable renderer. We leverage the fact that having a video consisting of many viewpoints can provide additional information useful in predicting the particular 3D geometry of a person, especially compared to off-the-shelf monocular 3D methods. Consequently, our face-swapping pipeline can provide detailed geometry and a high-quality texture.

Our network operates on preprocessed face images that are first cropped and normalized (Section 3.1). Our proposed network, its dynamic model of facial texture and 3D geometry, and the differentiable renderer are presented in Section 3.2. At inference time, the encoder/dual-decoder pair can be reassembled to perform face swapping between pairs of trained identities. A final step then blends the new face onto the original image (Section 3.4).

3.1. Preprocessing

To train a model that is capable of swapping the facial expression of a *target* performance to a *source* identity, a sequence of frames for each identity is necessary. We use the dataset captured by Naruniec et al. [NHSW20] that consists of sequences of eight different identities under three different lighting conditions and in a variety of poses and expressions. Each of the clips is captured at 4K and are between two to six minutes long.

Image Normalization and Pose Detection Our preprocessing step crops, rotates, and scales each training image to provide a smaller, normalized image with a centered face. This preprocessing is also required at inference (swapping) time, and all parameters of this normalization are saved so the process can be reversed when blending a new face back onto the original image. The normalization parameters are computed based on the positions of the eyes, nose, and mouth, which are provided by an off-the-shelf facial landmark detector [KNT17]. The detected landmarks are averaged from ten random face bounding box perturbations as proposed by [NHSW20] to achieve better temporal stability across frames. For pose detection we use 3DDFA [ZLLL17], since it provides a face mesh with 3D landmarks and an associated head pose (projection) matrix that our 3D decoders require during training and testing, as described below. Note, however, that our method will predict a more identity-specific facial shape than that of 3DDFA, which we illustrate in Section 4, and so the primary responsibility of 3DDFA in our work is for head pose computation.

Face Segmentation Mask To define the 2D target face region for training and swapping, we compute a face mask for each input image using the BiSeNet [YWP*18] face segmentation network implementation of [Zll19], which is pre-trained on the CelebAMask-HQ dataset.

3.2. Comb Network Architecture with 3D Decoders

Our autoencoder architecture is inspired by the comb model originally proposed in [NHSW20] for 2D face swapping. A general comb model consists of a single convolutional encoder $E(\cdot)$ and multiple convolutional decoders $D_p(\cdot)$, $p \in \{1,\dots P\}$, one for each of the P target identities considered at training time. In this work, we have two decoders per identity and we fix $P = 2$ to train dedicated swapping networks between a chosen source s and target identity t, although a more general training scheme is also possible.

The identity-specific decoders allow for disentangling the semantic details of the performance (e.g. head pose, eye gaze, facial expression, lighting) from the personal appearance of the identity. The performance is encoded on a per-frame basis within the shared latent code, and the personal appearance given those semantic details are decoded by the identity-specific decoders. Before face swapping at test time, the network is trained as two autoencoders with the task of reconstructing performances of the individual identities (with a single shared encoder).

A significant departure of our approach from the general architecture in 2D based comb [NHSW20] is in training two separate facial decoders per identity, one for texture and one for 3D geometry (i.e. D^T_p and D^G_p), which operate together with a differential renderer to reconstruct and project 3D faces onto the original 2D image plane (Fig. 2). This novel architecture allows us to perform training and swapping with explicit 3D geometry and texture, pooling data across the different viewpoints seen at training time. This helps to generalize to viewpoints at inference time that were not seen by the subject-specific decoder during training. Furthermore, we effectively embed and solve simultaneous 3D reconstruction problems during the training of each autoencoder. These model-free, data-driven reconstructions allow us to better adapt and capture identity features that are unique to particular faces (e.g. the shape of the nose and chin), leading to face shapes that are more accurate than those achieved with off-the-shelf 3D face reconstruction methods that use fixed geometry priors.

As shown in Fig. 2, the geometry decoder D^G_p produces delta shape images ΔG_p, encoded as the difference from a static 3D geometry $\overline{G}_p$. The static geometry can be considered the mean face shape of the identity. At the start of the training 3DDFA is fitted to a neutral expression image once per identity. We use this fit to initialize $\overline{G}_p$. Note that $\overline{G}_p$ is not an output of the decoder, but is rather learned together with the decoder weights during training over all frames. This is beneficial, for instance, when the input shows a profile or occluded face, the static term provides increased robustness for inferring the correct output. To obtain a per-frame geometry estimate, the dynamic geometry component ΔG_p is added to the static prior $\overline{G}_p$. In contrast to the comb model in [NHSW20], rather than outputting final 2D faces, the first decoder outputs a 3-channel image made up of a delta XYZ geometry map, and the second decoder outputs an RGB texture, both set in the UV texture plane of our 3D mesh topology (as in [GZY*20]). The decoded per-frame 3D geometry and texture are then fed into the differentiable renderer `redner` [LADL18] to produce the output 2D face. The rendering input also includes the face pose (rotation and translation) parameters extracted by fitting 3DDFA to the normalized input image. This rigid transformation is applied to the learned geometry before rendering. Here, we model most of the lighting implicitly, in a data-driven way, into the latent code learned by the shared image encoder. Combined specular and diffuse facial shading corresponding to the input lighting condition is decoded directly into the output texture. Thus, we set up `redner` with a constant, fully white environment map that in effect introduces only an additional (diffuse) ambient occlusion component to the rendered facial appearance. During training, the rendered face is compared to real images of the same identity (without swapping) and errors are backpropagated into the learned geometry and texture parameters. Our network design aims to better model the identity-specific features, such as particular nose shape and skin features, and therefore produce renderings with higher fidelity. Please refer to the supplementary material for more details on our network architecture.

At inference (face-swap) time, the rendered 2D face image is finally blended over the original image of the input scene. This blending step is summarized in Section 3.4. Optionally, before blending, the face geometry, texture, and the white environment map fed into `redner` can all be manually altered to incorporate artistic control over the rendered face, as demonstrated in Section 4.3.

3.3. Training

We train our network with loss functions that aim to reconstruct the images as closely as possible, considering structural similarity of face pixels and the face silhouette, with regularizers to control the geometry smoothness. Unlike [NHSW20], we do not follow a progressive training regime, which was necessary for gently increasing the resolution capabilities of the model while also learning similar representations for similar poses and expressions. In our case, the pose information is explicitly supplied and serves effectively as a conditioning signal to the network. As a result, we directly

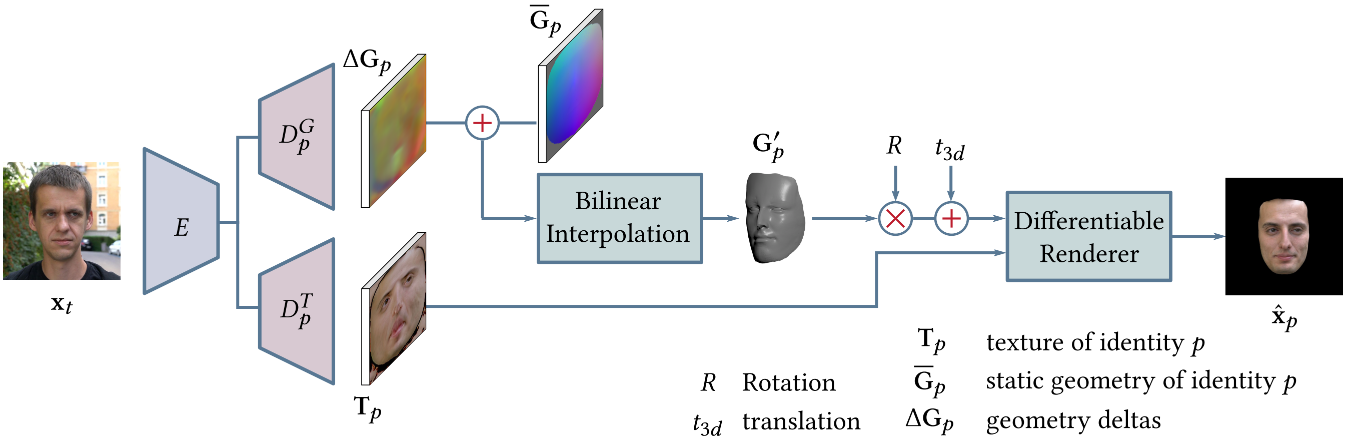

Figure 2: *The normalized target image $\mathbf{x}_t$ is fed to our autoencoder network with shared encoder E across identities. Two identity-specific decoders D_p^G and D_p^T are trained to generate, respectively, a 3D geometry delta image ΔG_p and a texture image T_p for each frame. During training, $p = t$, where t refers to the target identity. However, during swapping, $p = s$, where s is the second identity (source) that our network is trained on. The geometry delta image ΔG_p is added to the learned static geometry $\overline{G}_p$ and converted to a 3D face mesh G'_p, which is posed and rendered back to the input image plane using a differentiable renderer. The resulting image $\hat{x}_p$ can be used for training or swapping. See the supplementary material for network architecture details.*

start training on our final resolution (512 × 512). Images larger than 512 × 512 are downsized accordingly for training and swapping.

Reconstruction Loss: With only the face area being of interest, we mask the background region of all images. This is accomplished by predicting a binary face segmentation mask $\mathcal{M}(\mathbf{x}_p)$ for an image $\mathbf{x}_p$ belonging to identity p (Section 3.1). Our reconstruction loss seeks to minimize the discrepancy between the target image and the network output $\hat{\mathbf{x}}_p$:

$$\mathcal{L}_{\text{rec}} = d\left(\mathbf{x}_p \odot \mathcal{M}(\mathbf{x}_p),\, \hat{\mathbf{x}}_p \odot \mathcal{M}(\mathbf{x}_p)\right), \tag{1}$$

where $d(x,y) = \frac{1}{2}(1 - \text{SSIM}(x,y))$ is the *structural dissimilarity* between x and y and $\odot$ is the elementwise Hadamard product [WSB03].

Silhouette Loss: During training we observed that the model struggles to learn the correct subject-specific facial boundaries (i.e. nose shape - Figure 10, row 3, column 4). To overcome this problem, we constrain the learned geometry toward the correct shape boundary. We define the *silhouette loss* as the ℓ_1-loss between the face segmentation mask and the silhouette of the learned geometry [WZL*18], an additional output of the differentiable renderer:

$$\mathcal{L}_{\text{sil}} = \|\mathcal{M}(\mathbf{x}_p) - \mathcal{S}(\mathbf{G}'_p)\|_1, \tag{2}$$

where $\mathcal{S}(\mathbf{G}'_p)$ is the learned geometry silhouette.

Laplacian Smoothing Loss: While the silhouette loss forces the model to learn the correct mesh boundary, it does not prevent the generation of uneven vertices inside the silhouette (Figure 10, column 5). We can control for this problem by adding the *Laplacian regularizer* as proposed in [WZL*18]. This additional loss term serves as a local detail-preserving operator, which encourages a vertex v to move in the same direction as its neighbors $\mathcal{N}(v)$.

To compute the loss, we calculate the Laplacian coordinate δ_v for each vertex $v \in \mathcal{R}^3$:

$$\delta_v = v - \frac{1}{|\mathcal{N}(v)|} \sum_{v' \in \mathcal{N}(v)} v'. \tag{3}$$

We then constrain the movements of each vertex in the learned mesh to be smooth compared to the original 3DDFA mesh via a simple ℓ_2 penalty:

$$\mathcal{L}_{\text{sm}} = \sum_v \|\delta_{v_{\text{3DDFA}}} - \delta_{v_{\text{learned}}}\|_2^2. \tag{4}$$

Geometry Delta Regularizers: As mentioned in Section 3.2, we assume that the geometry can be separated into identity-specific static and dynamic components. While the static component is constant over all frames, the dynamic component should only contain frame-specific information, such as the expression. We refer to the dynamic component as *deltas* per frame.

The geometry deltas $\Delta\mathbf{G}_p$ represent the offsets of the model's estimated vertices for a given frame relative to the static geometry. To first learn an identity-specific representation of the static geometry, we constrain the geometry deltas to be close to zero in the beginning of the training:

$$\mathcal{L}_{\Delta\mathbf{G}_1} = \|\Delta\mathbf{G}_p\|_2^2. \tag{5}$$

Once we arrive at an identity-specific static geometry, regularizing the deltas towards 0 becomes limiting, and we begin regularizing instead using the estimated per-frame meshes from 3DDFA. Specifically, let $\mathbf{D}_p$ represent the frame-specific differences between 3DDFA's neutral geometry for identity p and the frame-specific 3DDFA expression in image $\mathbf{x}_p$, then our regularization term is given by

$$\mathcal{L}_{\Delta\mathbf{G}_2} = \|\Delta\mathbf{G}_p - \mathbf{D}_p\|_2^2. \tag{6}$$

This loss effectively constrains the model's sense of geometric displacement to resemble that of 3DDFA. Importantly, this loss is gradually phased out in order to allow the model greater freedom to represent geometry outside of 3DDFA's set of priors.

Overall Objective: During training, the differentiable rendering is guided by these losses and helps our model to learn identity-specific features and render the faces with higher fidelity. The combined objective for each frame is obtained by adding up the weighted loss terms:

$$\mathcal{L} = \beta_r \mathcal{L}_{\text{rec}} + \beta_s \mathcal{L}_{\text{sil}} + \beta_l \mathcal{L}_{\text{sm}} + \beta_{g_1} \mathcal{L}_{\Delta \mathbf{G_1}} + \beta_{g_2} \mathcal{L}_{\Delta \mathbf{G_2}}, \qquad (7)$$

where $\beta_r, \beta_s, \beta_l$, β_{g_1} and β_{g_2} are loss-weighting hyperparameters. At training time, we sequentially iterate over both identities and process one identity per batch. The final objective is given by

$$\mathcal{L}_{\text{final}}(\mathcal{X}) = \frac{1}{2} \sum_{p=1}^{2} \frac{1}{N_p} \sum_{i=1}^{N_p} \mathcal{L}\left(\mathbf{x}_p^{(i)}, \hat{\mathbf{x}}_p^{(i)}\right), \qquad (8)$$

where N_p is the number of frames corresponding to identity p.

Training Details: The proposed model is trained by minimizing equation (8) using the Adam optimizer [KB15]. The training itself consists of multiple stages. From the start, we train the full model with regularizer $\mathcal{L}_{\Delta G_1}$ with weight $\beta_{g_1} = 1$. The regularization term β_{g_1} is set to zero after 50K iterations, since at this stage static geometry contains most of the identity-specific information and only frame-specific differences are left to learn. We then use $\mathcal{L}_{\Delta G_2}$ with weight $\beta_{g_2} = 1$ for 30K iterations to regularize the geometry deltas towards the frame-specific expressions, before phasing it out to allow the model to have greater freedom to represent geometries and expressions outside of the 3DDFA prior. The remaining regularization weights are fixed throughout the training: $\beta_r = 1$, $\beta_s = 1$, and $\beta_l = 5$. In total we train our model for 200K iterations for each identity pair taking roughly 9 days on a 3090 GPU or 18.5 days on a TITAN X GPU. We expect more modern GPUs to further decrease the training time. On a TITAN X GPU, one full iteration with two IDs takes around 8 seconds, mainly consisting of the forward pass with rendering (0.6 secs/ID), computing the gradients (2.5 secs/ID), and computing the several losses (where by far the most expensive is the smoothing loss: 1.1 secs/ID). Due to limitations of our memory, we train with a batch size of one on the TITAN X GPU.

3.4. Post-Processing

At test (swap) time, the rendered face image $\hat{\mathbf{x}}_p$ is moved to the correct position on the target frame $\mathbf{y_t}$ by reversing the image normalization process. We create a blending mask by rendering a painted texture onto the learned geometry (Figure 3). For the final swap, multi-band blending [BA83, NHSW20] is applied to the mask and the output image.

4. Experiments

This section offers experimental results showing the quality of our method's (1) learned geometry, as well as (2) face swapping results in comparison to other state-of-the-art methods. Swaps are performed on images that display a variety of head poses, including challenging profile views that 2D methods struggle with. Finally, we also demonstrate (3) how artistic control can be easily incorporated into our method, (4) our model's generalization capacity, and (5) an ablation study of our training loss.

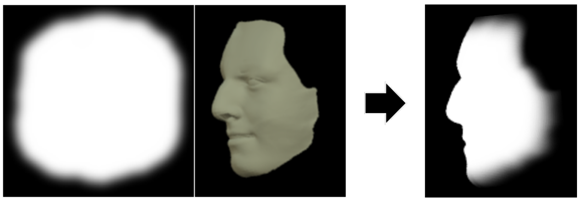

Figure 3: *We render the texture mask onto the learned geometry to generate our blending mask.*

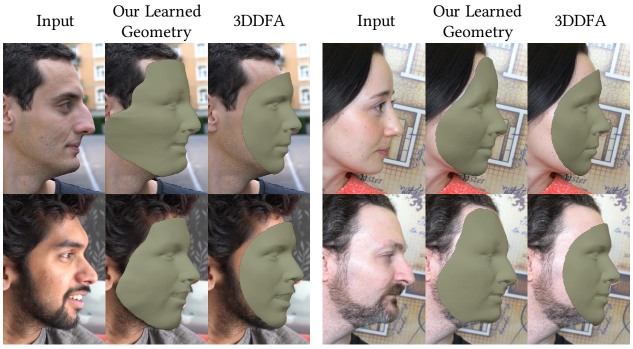

Figure 4: *Profile views illustrating the geometry learned by our method for four identities, showing the improvement over the initial result obtained using 3DDFA.*

4.1. Learned Geometry

Figure 4 illustrates the geometry learned by our method in comparison to the initial 3DDFA estimate on four subjects, in profile view. Our method successfully captures the characteristics of the individual faces, such as nose and chin shape, which are poorly modeled by 3DDFA. Although 3DDFA correctly estimates pose and expressions, 3DDFA's results across different subjects often show very similar facial geometry that miss most of the variability due to unique identity features of each subject. Other fitting errors also occur when parts of the estimated facial geometry incorrectly overlap with the background. For these reasons, our method takes the 3DDFA fit for each identity only as a first initialization of the subject's static geometry. During training, the static geometry learns to adapt more identity-specific features. The improvement is especially noticeable in profile views, where the generic nose shape turns into an identity-specific nose shape (Figure 4).

We expect these facts to also hold with respect to other pretrained 3D face reconstruction methods that process individual images as input [FFBB20, GZY*20], as opposed to multi-image methods like ours. Single-image methods are thus strongly biased by their training dataset, making them less than ideal for face swapping on video of novel faces with unique geometric features.

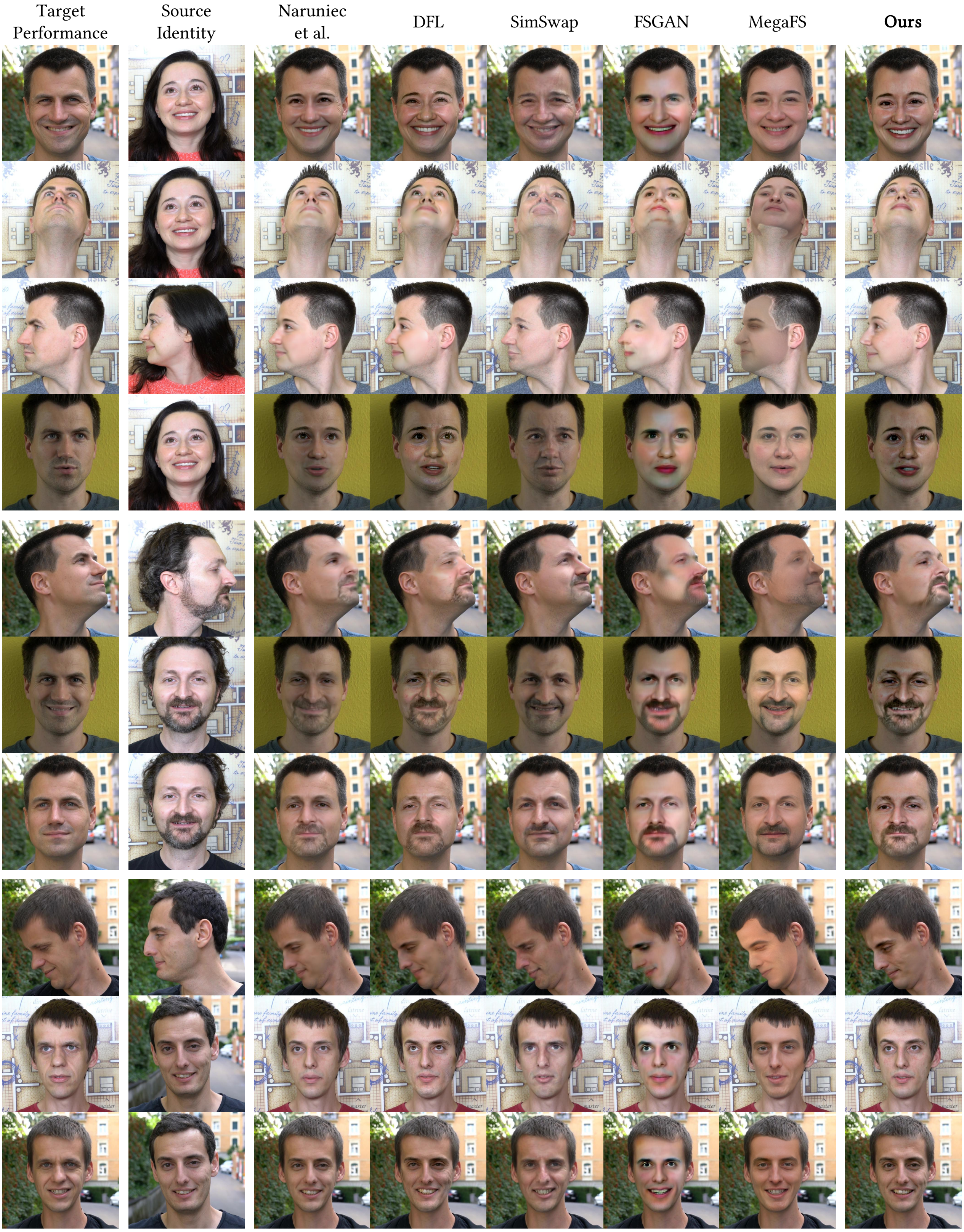

Figure 5: *Comparison of state-of-the-art face swapping methods.*

4.2. Comparison with State-of-the-Art Face Swapping

We compare our method to DeepFaceLab (DFL) [PGC*20], Naruniec et al. [NHSW20], SimSwap [CCNG20], MegaFS [ZLW*21] and FSGAN [NKH19], five state-of-the-art face-swapping methods that show convincing, high-quality results. For a better comparison, we retrain the two subject-specific methods (DFL and Naruniec et al.) on the dataset of [NHSW20]. For DFL, we train each identity pair for 300K iterations, applying an adversarial loss for the last 50k iterations. We use the highest resolution that fits into the memory of our hardware (352x352). We retrain the model of Naruniec *et al.* [NHSW20] in a progressive fashion for 100K presented images, at each resolution level, up to our common final resolution of 512x512. SimSwap, MegaFS and FSGAN all claim to be subject-agnostic and therefore do not require retraining [CCNG20, NKH19, ZLW*21].

Figure 5 shows the side-by-side comparison among the six methods. Swaps from DeepFaceLab (DFL) show a high level of detail and realism, especially for front poses, but fail to capture the target lighting in some settings (row 4 and 6). Similar, Naruniec *et al.* show realistic swaps in most of the poses but lack overall sharpness of the face (row 1 and row 9). While SimSwap, FSGAN and MegaFS allow for subject-agnostic face swapping, they struggle to maintain faithful source identity (row 4, 6 and 9). FSGAN and MegaFS show less realistic profile views (row 3 and 5) and have difficulties in keeping the target lighting (row 1, 4 and 7). Our method shows high details in front poses (teeth in row 1) and correctly captures the target performance (shows teeth in row 6). For profile poses our method benefits from the pose information and the learned geometric shape that captures the source identities nose shape. Row 2, 3, 5 and 8 show, how our subject-specific decoders generalize to these novel viewpoints at test (swap) time. Additional comparisons are shown in the supplementary video.

To assess which method performs face swapping better, we also conducted a perceptual user study with 59 adult participants, rating our approach alongside four of the five competing face-swapping methods. Specifically, we compare our method to Naruniec et al. [NHSW20], DeepFaceLab [PGC*20], FSGAN [NKH19] and SimSwap [CCNG20]. Four source-target pairs were randomly selected from the available data, and we used each competing method to perform a swap between source and target. For each source-target pair, the participants viewed a 10-second video clip featuring a gallery of the outputs from all five methods in random spatial order, with no limitation on the time allowed for evaluation. We followed a randomized ranked-choice design using questions similar to those asked in [LBY*19], focusing on realism, similarity to the intended identities, pose and expression quality, lighting, and profile quality, all adapted to a video context. The results of our study showed that no method was dominant in every category. Our method produced results ranked at the top by a number of study participants across all categories, although the 2D method by Naruniec et al. [NHSW20] received the most votes overall. See the supplemental material for more information on study design and detailed results.

Similar to [NKH22] and [XZH*22], we quantitatively compare our method to other face-swapping methods using the FID score, which correlates with human perception the quality of generated images [HRU*17]. We use the FID score to judge the visual quality of the swaps (generated images) when compared to a distribution of real images displaying the source identity. We evaluate on around 12000 real images and 12000 swaps per method using the dataset of Naruniec *et al.* [NHSW20] and the PyTorch FID implementation of [Sei20]. Our method achieves the best FID score among all evaluated face-swapping methods (see Table 1), indicating the best-matching swap relative to the source identity imagery.

Method	FID score ↓
FSGAN [NKH19]	35.06
MegaFS [ZLW*21]	32.19
SimSwap [CCNG20]	25.67
Naruniec et al. [NHSW20]	23.59
DFL [PGC*20]	20.54
Ours	**18.14**

Table 1: *Our swaps show the best image quality among the evaluated methods as judged by the FID score.*

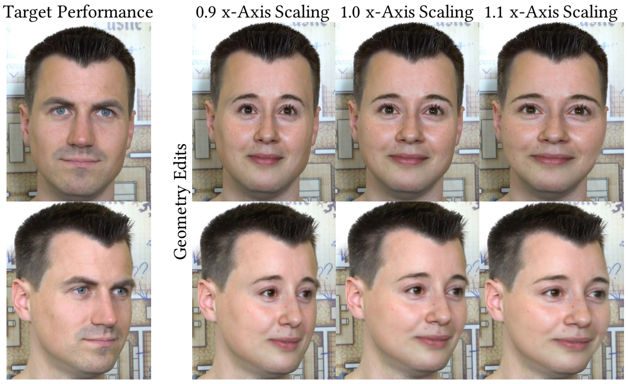

Figure 6: *Incorporating artistic control on the learned geometry: result of geometry edits that scale along the x-axis with a factor of 0.9, 1.0 and 1.1. The source identity is the same as in Figure 7*

.

4.3. Artistic Control

One of the benefits of our 3D approach for video-based face swapping is that it allows more natural artistic control of the result, as compared to traditional 2D methods. Artists are familiar with manipulating 3D geometry and textures and their edits can be seamlessly integrated into our pipeline at test time, after decoding the geometry and texture, just before rendering and compositing. We highlight this benefit by demonstrating the ease with which the swap result can be edited to modify the 3D face shape (in Figure 6) or the facial texture (adding a tattoo, in Figure 7). The ability to relight the face by adjusting the environment map before rendering is also demonstrated in Figure 8 and in the supplementary video.

4.4. Generalization

During inference (swapping), our subject-specific decoders might have to generate the subject under novel viewpoints since each

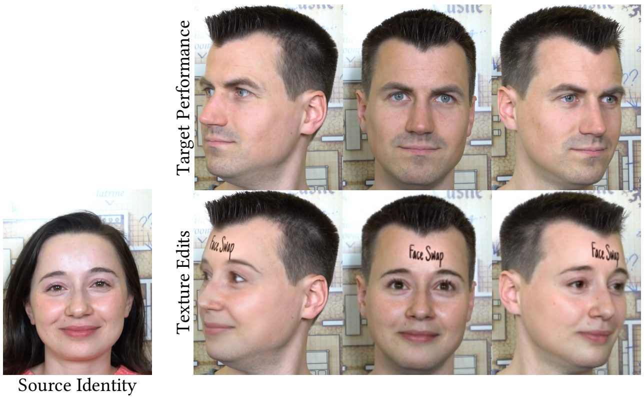

Figure 7: *Incorporating artistic control on the learned texture: result of adding a* tattoo *on the forehead of the source identity.*

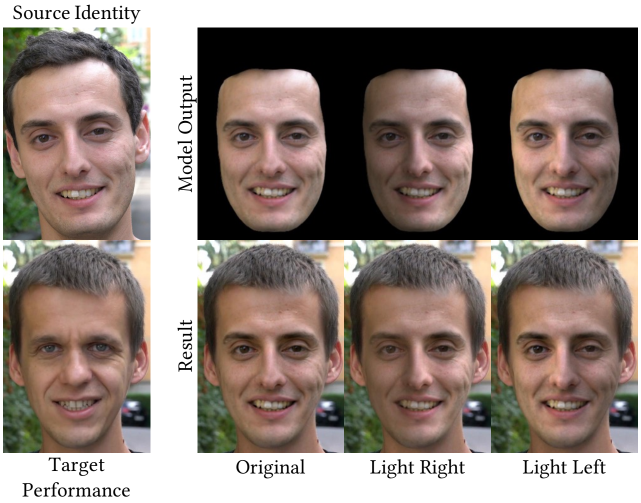

Figure 8: *Artistic control by relighting when rendering: example of taking the original result (column 2) and placing an additional light on the right (column 3) or on the left (column 4).*

of the decoders is evaluated on a code coming from another subject that was not seen during training. In contrast, the encoder is shared across both identities and must learn identity-agnostic features. While the network is primarily designed to be applied only to the source and target identity, it may further generalize to new identities or a known identity with new expression and lighting (Figure 9).

4.5. Ablation study

We show the effect of leaving out different parts of our loss function visually in Figure 10. For example, it can be seen how the silhouette loss helps to learn a nose shape that is closer to the source identity's nose shape (column 2, row 3 vs. column 4, row 3). This example indicates that performing face swapping with subject-specific face shapes leads to more realistic results than performing face swapping with more generic off-the-shelf 3D reconstructed face shapes.

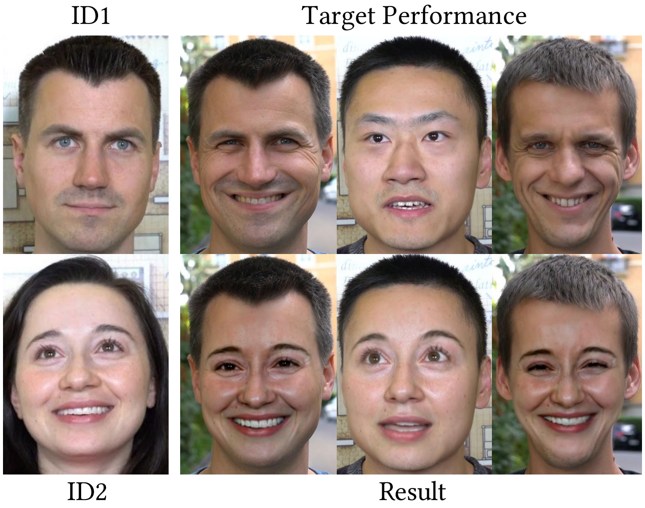

Figure 9: *Our model trained only on ID1 and ID2 (column 1) can create swaps that generalize: It can encode an unknown target ID and decode ID2 (column 3). Furthermore, when combined with multi-band blending [BA83, NHSW20] it can create swaps of ID2 in illumination conditions that are novel to the decoder (column 2), and it can encode an unknown target ID and create swaps of ID2 in novel illumination conditions (column 4).*

5. Limitations

We have identified four sources of limitations of the proposed method. First, imperfections of the face segmentation algorithm can affect the reconstruction quality. As the segmentation mask is used in the silhouette loss defined in Section 3.3, the segmentation significantly influences the boundary of the learned geometry. However, current segmentation algorithms are already quite accurate in many cases and advancements in facial segmentation will likely lead to improved reconstructions in the remaining cases.

Second, in some sequences the model is not perfectly stable in time and we can observe minor jittering of the results. This is caused by the inaccurate fit of the underlying 3DDFA model used for the face pose estimation. In future experiments, we intend to additionally regress the pose directly from the input image to mitigate this behavior.

Third, our approach occasionally has difficulties with faithfully swapping rare or extreme target expressions, e.g. very wide smiles or yawning. In these cases, the result may contain blurry teeth or mouth regions (Figure 11, rows 1 and 2). Additionally, incorrect eye gaze direction may occur if the training data lacks the target gaze direction for the source identity (Figure 11, row 3). We aim to investigate these issues further in future work.

Lastly, the differentiable rendering and geometry smoothing loss are two bottlenecks that lead to long training times (Section 3.3).

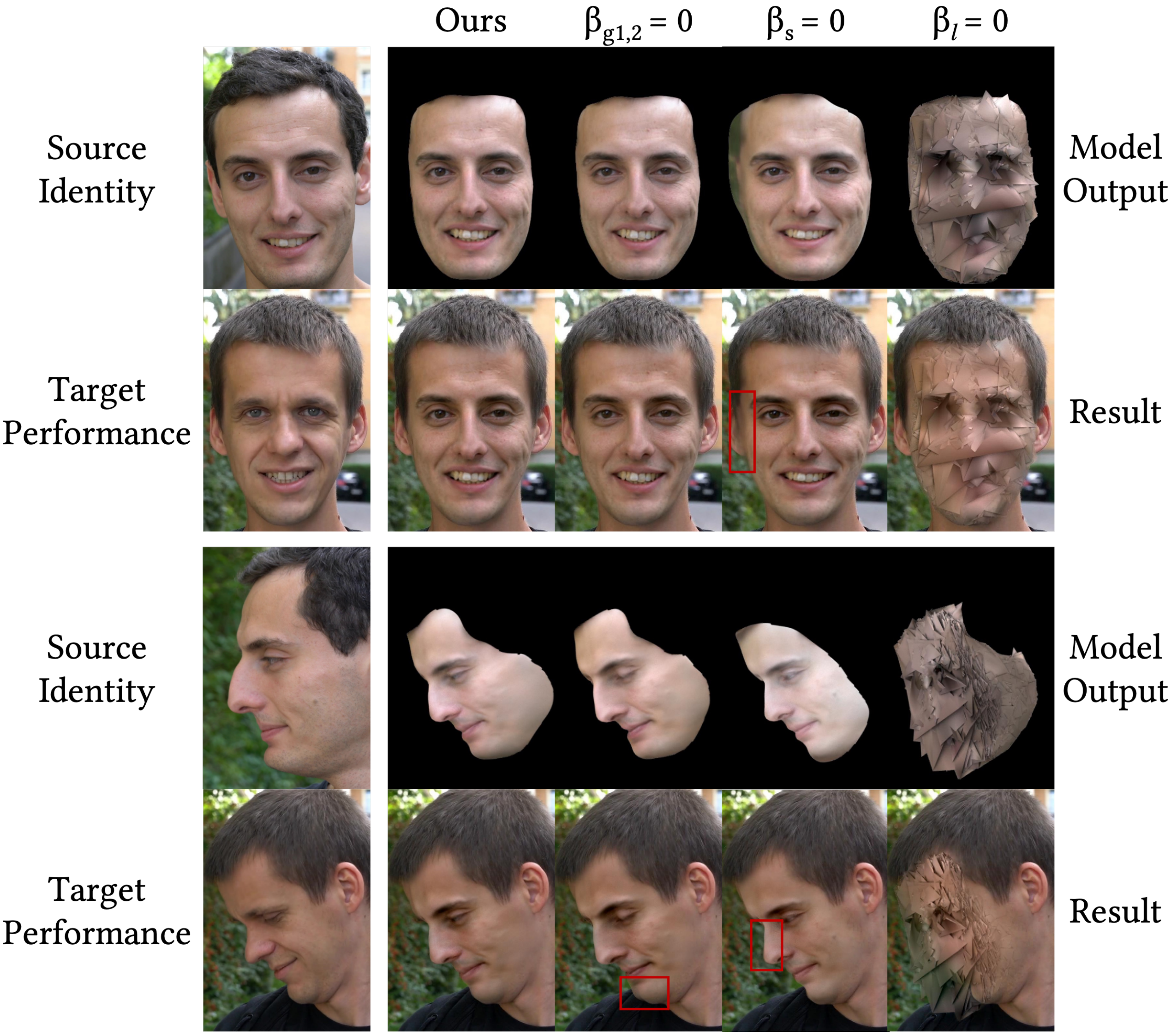

Figure 10: *Ablation study. Column 2 shows the results of our method when setting the loss weights as described in section 3.3. In column column 3, row 4 (setting the geometry regularizers to 0 for the full training), it can be seen that the source identity's chin is sticking out too much in front, leading to artifacts. Column 4, row 3 and 4 show that the source identities' nose shape is not learned by the geometry when the silhouette loss is set to 0. Similarly, blending artifacts occur for the front pose (column 4, row 2) when the silhouette is not constrained. Column 4 shows the effect of leaving out the geometry smoothing loss term (i.e. setting the weight to 0). The underlying face geometry collapses.*

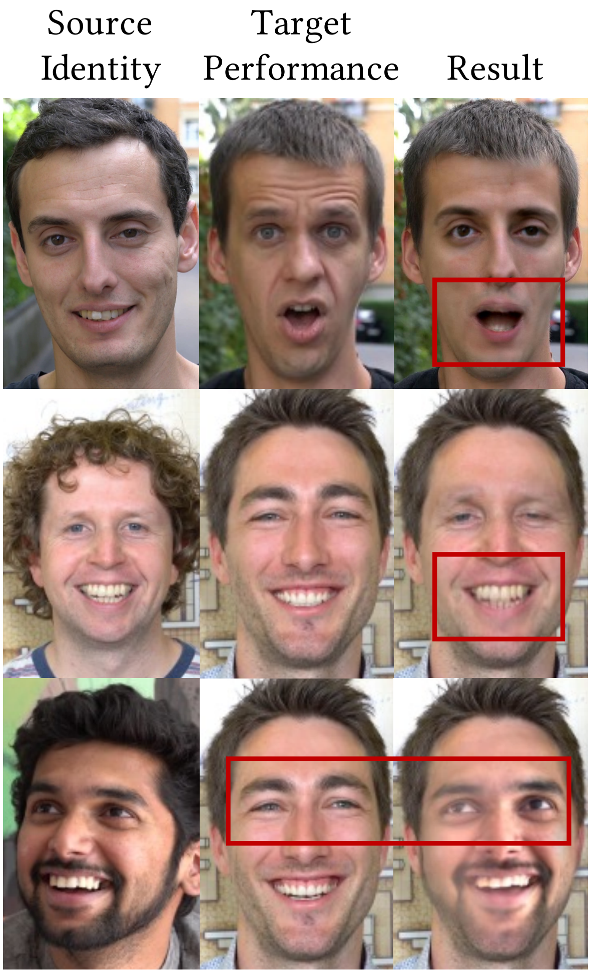

Figure 11: *Visual failure cases. The results in row 1 and 2, column 3 show blurry teeth for extreme expressions such as a mouth that is wide open or a big smile. Row 3, column 3 shows problems in displaying the correct eye gaze direction, likely due to the fact that the training data of the source identity mostly contains examples of looking to the upper left when displaying a wide smile.*

Making either of these processes more efficient would be one key to reducing training time in the future.

6. Ethical Concerns

Any method that generates photorealistic facial imagery carries with it the potential for misuse. We condemn such misuse and support the growing research effort into automatically detecting manipulated imagery [Wes19, TVRF*20]. We also stress the legitimate role that face-swapping technology plays in visual effects and intend for our work to be applied only in that direction.

7. Conclusion

In this paper we present a novel face-swapping pipeline that employs 3D information by simultaneously learning facial textures with person-specific face shapes and frame-dependent, expression-derived updates to these shapes. We showed that, compared to other methods which operate only in 2D, we can generate higher-quality swaps as measured by the popular FID score. Additionally, our 3D approach gives more control to the artist in compositing the result, where changes to the shape, texture and lighting are all possible. Finally, we also demonstrated that our approach leads to improved geometry estimates compared to traditional monocular face capture methods like 3DDFA.

References

[BA83] Burt P. J., Adelson E. H.: A multiresolution spline with application to image mosaics. *ACM Trans. Graph. 2*, 4 (1983), 217–236. 6, 9

[BSVS04] Blanz V., Scherbaum K., Vetter T., Seidel H.-P.: Exchanging faces in images. *Computer Graphics Forum 23*, 3 (2004), 669–676. 3

[BV99] Blanz V., Vetter T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., p. 187–194. 3

[BV03] Blanz V., Vetter T.: Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 9 (2003), 1063–1074. 3

[CCNG20] Chen R., Chen X., Ni B., Ge Y.: *SimSwap: An Efficient Framework For High Fidelity Face Swapping*. Association for Computing Machinery, New York, NY, USA, 2020, p. 2003–2011. 2, 3, 8

[DSJ*11] Dale K., Sunkavalli K., Johnson M. K., Vlasic D., Matusik W., Pfister H.: Video face replacement. *ACM Trans. Graph. 30*, 6 (2011), 1–10. 3

[DYX*19] Deng Y., Yang J., Xu S., Chen D., Jia Y., Tong X.: Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops* (2019). 3

[FFBB20] Feng Y., Feng H., Black M. J., Bolkart T.: Learning an animatable detailed 3D face model from in-the-wild images. In *arxiv* (2020). 3, 6

[FWS*18] Feng Y., Wu F., Shao X., Wang Y., Zhou X.: Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV* (2018). 3

[GGH02] Gu X., Gortler S. J., Hoppe H.: Geometry images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (2002), SIGGRAPH '02, p. 355–361. 3

[GPAM*14] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680. 2, 3

[GPKZ19] Gecer B., Ploumpis S., Kotsia I., Zafeiriou S.: Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 3

[GSZ*18] Geng J., Shao T., Zheng Y., Weng Y., Zhou K.: Warp-guided gans for single-photo facial animation. *ACM Transactions on Graphics 37* (12 2018), 1–12. 3

[GZY*20] Guo J., Zhu X., Yang Y., Yang F., Lei Z., Li S. Z.: Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2020). 3, 4, 6

[HRU*17] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS'17, Curran Associates Inc., p. 6629–6640. 8

[KALL17] Karras T., Aila T., Laine S., Lehtinen J.: Progressive growing of gans for improved quality, stability, and variation. *CoRR abs/1710.10196* (2017). arXiv:1710.10196. 2

[KB15] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015). 6

[KCT*18] Kim H., Carrido P., Tewari A., Xu W., Thies J., Niessner M., Pérez P., Richardt C., Zollhöfer M., Theobalt C.: Deep video portraits. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 163. 3

[KLA*20] Karras T., Laine S., Aittala M., Hellsten J., Lehtinen J., Aila T.: Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR* (2020). 2

[KNT17] Kowalski M., Naruniec J., Trzcinski T.: Deep alignment network: A convolutional neural network for robust face alignment. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017), 2034–2043. 4

[LADL18] Li T.-M., Aittala M., Durand F., Lehtinen J.: Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 37*, 6 (2018), 222:1–222:11. 4

[LBK17] Liu M.-Y., Breuel T., Kautz J.: Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems 30* (2017), Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., (Eds.), Curran Associates, Inc., pp. 700–708. 2

[LBY*19] Li L., Bao J., Yang H., Chen D., Wen F.: Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457* (2019). 8

[LWXS22] Li Q., Wang W., Xu C., Sun Z.: Learning disentangled representation for one-shot progressive face swapping, 2022. 2

[NHSW20] Naruniec J., Helminger L., Schroers C., Weber R.: High-resolution neural face swapping for visual effects. *Computer Graphics Forum 39*, 4 (2020), 173–184. 2, 3, 4, 6, 8, 9

[NKH19] Nirkin Y., Keller Y., Hassner T.: FSGAN: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7184–7193. 2, 3, 8

[NKH22] Nirkin Y., Keller Y., Hassner T.: FSGANv2: Improved subject agnostic face swapping and reenactment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022). 8

[NMT*18] Nirkin Y., Masi I., Tran A. T., Hassner T., Medioni G.: On face segmentation, face swapping, and face perception. In *IEEE Conference on Automatic Face and Gesture Recognition* (2018). 2, 3

[NSX*18] Nagano K., Seo J., Xing J., Wei L., Li Z., Saito S., Agarwal A., Fursund J., Li H.: Pagan: Real-time avatars using dynamic textures. *ACM Transactions on Graphics 37* (12 2018), 1–12. 3

[OLY*17] Olszewski K., Li Z., Yang C., Zhou Y., Yu R., Huang Z., Xiang S., Saito S., Kohli P., Li H.: Realistic dynamic facial textures from a single image using gans. In *IEEE International Conference on Computer Vision (ICCV)* (10 2017), pp. 5439–5448. 3

[PGC*20] Perov I., Gao D., Chervoniy N., Liu K., Marangonda S., Umé C., Dpfks M., Facenheim C. S., RP L., Jiang J., Zhang S., Wu P., Zhou B., Zhang W.: Deepfacelab: A simple, flexible and extensible face swapping framework. *CoRR abs/2005.05535* (2020). 2, 3, 8

[PKA*09] Paysan P., Knothe R., Amberg B., Romdhani S., Vetter T.: A 3d face model for pose and illumination invariant face recognition. In *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS) for Security, Safety and Monitoring in Smart Environments* (Genova, Italy, 2009), IEEE. 3

[Sei20] Seitzer M.: pytorch-fid: FID Score for PyTorch, 2020. Version 0.2.1. URL: https://github.com/mseitzer/pytorch-fid. 8

[Sey19] Seymour M.: De-aging the irishman, 2019. URL: https://www.fxguide.com/fxfeatured/de-aging-the-irishman/. 3

[TVRF*20] Tolosana R., Vera-Rodriguez R., Fierrez J., Morales A., Ortega-Garcia J.: Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion 64* (2020), 131–148. 11

[TZS*16] Thies J., Zollhöfer M., Stamminger M., Theobalt C., Niessner M.: Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* (2016). 3

[WCZ*21] Wang Y., Chen X., Zhu J., Chu W., Tai Y., Wang C., Li J., Wu Y., Huang F., Ji R.: Hififace: 3d shape and semantic prior guided high fidelity face swapping. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21* (8 2021), Zhou Z.-H., (Ed.), International Joint Conferences on Artificial Intelligence Organization, pp. 1136–1142. 3

[Wes19] Westerlund M.: The emergence of deepfake technology: A review. *Technology Innovation Management Review 9*, 11 (2019). 11

[WSB03] Wang Z., Simoncelli E. P., Bovik A. C.: Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003* (2003), vol. 2, pp. 1398–1402 Vol.2. 5

[WZL*18] Wang N., Zhang Y., Li Z., Fu Y., Liu W., Jiang Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV* (2018). 5

[XDW*22] Xu Y., Deng B., Wang J., Jing Y., Pan J., He S.: High-resolution face swapping via latent semantics disentanglement, 2022. 2

[XZH*22] Xu C., Zhang J., Hua M., He Q., Yi Z., Liu Y.: Region-aware face swapping, 2022. 2, 8

[YWP*18] Yu C., Wang J., Peng C., Gao C., Yu G., Sang N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018). 4

[Zll19] Zllrunning: Face-parsing pytorch, 2019. URL: https://github.com/zllrunning/face-parsing.PyTorch. 4

[ZLLL17] Zhu X., Liu X., Lei Z., Li S. Z.: Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence* (2017). 3, 4

[ZLW*21] Zhu Y., Li Q., Wang J., Xu C., Sun Z.: One shot face swapping on megapixels. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (2021), pp. 4834–4844. 2, 3, 8