Eurographics Symposium on Rendering 2020
C. Dachsbacher and M. Pharr
(Guest Editors)

Volume 39 (2020), Number 4

High-Resolution Neural Face Swapping for Visual Effects

J. Naruniec [1], L. Helminger[2], C. Schroers [1] and R.M. Weber [1]

[1]Disney Research|Studios
[2]ETH Zürich

Figure 1: *The results of our face-swapping method. All images are generated in megapixel resolution as frames in temporally coherent video footage (best viewed in video; see supplementary material).*

Abstract

In this paper, we propose an algorithm for fully automatic neural face swapping in images and videos. To the best of our knowledge, this is the first method capable of rendering photo-realistic and temporally coherent results at megapixel resolution. To this end, we introduce a progressively trained multi-way comb network and a light- and contrast-preserving blending method. We also show that while progressive training enables generation of high-resolution images, extending the architecture and training data beyond two people allows us to achieve higher fidelity in generated expressions. When compositing the generated expression onto the target face, we show how to adapt the blending strategy to preserve contrast and low-frequency lighting. Finally, we incorporate a refinement strategy into the face landmark stabilization algorithm to achieve temporal stability, which is crucial for working with high-resolution videos. We conduct an extensive ablation study to show the influence of our design choices on the quality of the swap and compare our work with popular state-of-the-art methods.

CCS Concepts

• ***Computing methodologies*** → ***Image manipulation; Unsupervised learning; Neural networks;***

1. Introduction

The swapping of the appearance of a target actor and a source actor while maintaining the target actor's performance is a longstand-

ing and challenging problem in visual effects. The problem typically arises in cases in which a character needs to be portrayed at a younger age or when an actor is not available or is perhaps even long deceased. Other applications include stunt scenes that would be dangerous for an actor to perform but still require high-quality face images.

In the film and TV industry, a variety of approaches for face swapping have been explored over the years, and the ones in use today are typically elaborate and labor-intensive computer-graphics methods. They require great care on set as well as extensive frame-by-frame animation and post-processing by digital-effects professionals. The methods have only very recently matured to the point that filmmakers have become more willing to brave the "uncanny valley" and show detailed views of virtual actors. Examples include Paul Walker in *Furious 7* and Peter Cushing and Carrie Fisher in *Rogue One*.

While those results are impressive, they are expensive to produce and typically take many months of work to achieve mere seconds of footage. In contrast to these computer-graphics approaches, deep-learning methods for face swapping have attracted considerable attention in recent years. Those methods allow for an automatic, data-driven processing pipeline. Many approaches exist, typically employing either autoencoders [LBK17; KSDT17], GANs [DNWG17; NYM18b], or morphable models [DSJ*11; NMT*18]. However, several issues arise when these methods are used in high-resolution video face swapping. 3D model-based methods are capable of producing high-resolution images, but they currently lack temporal stability in the generated faces, resulting in unrealistic, rapidly changing appearances. GANs and autoencoders often have difficulty generating high-resolution images due to memory limitations, instability of the training procedure, and the choice of data samples.

In this work we present a method to generate high-resolution, photo-realistic, and temporally stable face swaps. We achieve this through the following core contributions:

1. We introduce a progressively trained, multi-way *comb network* that embeds input faces in a shared latent space and decodes them as any of the selected identities while maintaining the input face expression. This allows for richer, more realistic results than in the typical single-source, single-target setting.
2. We propose a full face-swapping pipeline including a contrast- and light-preserving compositing step and a landmark stabilization procedure that allows for generating temporally stable video sequences.
3. Finally, we report a comprehensive ablation study demonstrating the influence of particular design choices and procedures on swapping quality.

We demonstrate our method on challenging high-resolution video data gathered in a variety of settings and lighting conditions. We also compare our work with a number of state-of-the-art face-swapping methods, showing that our method is a major step toward photo-realistic face swapping that can successfully bridge the uncanny valley. As our system is also capable of multi-way swaps—allowing any pair of performances and appearances in our data to be swapped—the possible benefits to visual effects are extensive, all at a fraction of the time and expense required using more traditional methods.

2. Related work

A vast literature exists on the synthesis, editing, manipulation, and transfer of facial imagery in pictures and video. To survey existing work, we will use the following categories: *encoder-decoder (autoencoder) methods*, *GAN-based image-to-image translation*, and *geometry-based morphable models*. We will briefly review existing methods and also relate our work to recent *reenactment* and *puppeteering* methods.

2.1. Encoder-Decoder Methods

Liu et al. [LBK17] introduced a model with a strong influence on the present work. Although their model structure is quite different, featuring dual encoders and decoders based on the VAE-GAN framework, a key idea from their work is the concept of a *shared latent space* for encoded images, which is enforced via tied weights in several of the layers of the encoders and decoders closest to the encoded bottleneck.

Korshunova et al. [KSDT17] approach the problem of face swapping from the perspective of *style transfer*, in which the identity of a face is the *style* and the dynamic behavior is the *content*. They use a multiscale texture network with both content and style losses measured in a VGG-19 feature space.

Yan et al. [YHL*18] explore a Y-shaped, single-encoder, dual-decoder architecture that can be seen as a limiting case of our model structure. During training, they introduce warp distortions to the input images while tasking the decoders with reconstructing the undistorted images, akin to denoising autoencoders. Zhao et al. [ZTD*18] show impressive face-swapping performance using an encoder-decoder architecture with a multitask objective including face alignment and segmentation goals. However, their model requires extensive labeled training data and is, at its core, a supervised method, while our work is self-supervised. Natsume et al. [NYM18a] employ several encoder-decoder networks, each specializing in different features extracted from an input image (binary mask, isolated face, and facial landmarks) and use a separate generator to combine the target face with a source image.

2.2. GAN-Based Methods

Generative adversarial networks (GANs) [GPM*14] have become immensely popular for image synthesis and have recently entered the megapixel-and-beyond domain, most notably due to a progressive-training approach described by Karras et al. [KALL18; KLA18]. The general approach that has proved most successful for face swapping is image-to-image translation using *conditional* GANs [DNWG17; IZZE17; WLZ*18]. This approach, however, introduces a requirement for paired data, which can be difficult to produce. Subsequent methods have been developed to relax or altogether circumvent this paired-data requirement [ZPIE17].

In an application specific to faces, Natsume et al. [NYM18b] compose the output of two separator networks—one for the face

and one for hair, similar in form to the work described in Natsume et al. [NYM18a]—and use a GAN to "verify" and tune the result. Shu et al. [SYH*17] take the interesting approach of treating face representation as a rendering problem and use a GAN to create surface normal, albedo, lighting, and alpha matte information from input images to allow for more compelling image edits. Pumarola et al. [PAM*18] perform facial-expression synthesis and animation from single images by conditioning on action units from the Facial Action Coding System [ER97]. GAN-based facial animation has seen impressive subsequent development in recent work by [ZSBL19]. Recently Nirkin et al. [NKH19] presented a face-swapping and reenactment pipeline that can generalize to novel faces based on very few examples. However, due to the proposed face view interpolation, the results are slightly smoothed and inadequate for high resolution.

2.3. Geometry-Based Methods (Morphable Models)

Three-dimensional morphable models [BV*99] are explicit parametric representations of the geometry of the human face. In their classic form, 3D morphable models live in a vector space spanned by a basis of exemplars learned from images paired with 3D scans. Recent work has expanded the capabilities for creating such models, allowing them to be learned from sets of 2D images using deep encoder-decoder networks [TL18]. In the context reviewed here, morphable models are distinct from the detailed geometric models that can be made to capture *individual* faces in high fidelity [ARL*09; ZTG*18].

Blanz et al. [BSVS04] applied morphable models to face swapping, although with results falling short of photo-realistic. Dale et al. [DSJ*11] got impressive results by using 3D models to better align source and target images, which were then combined using an edit-based technique and additional post-processing. Yang et al. [YWS*11] use a geometric approach to perform transfers of individual facial *components*, while Shu et al. [SSSH17] achieve excellent results by specializing in manipulating the eyes in images to eliminate closed eyes and look-aways. Lin et al. [LWLT12] create a 3D model from a single frontal 2D image of a person's face, employing color transfer and a multi-resolution spline technique to achieve seamless blending. Nirkin et al. [NMT*18] present an approach for face swapping using semi-supervised data, with 3D models employed to register points for transferring image intensities from source to target.

2.4. Reenactment and Puppeteering

It is important to distinguish face *swapping* from the face *reenactment* problem [TZS*16; KCT*18; GSZ*18; SSK17; GVS*15; KEZ*19]. While at first sight the problems appear very similar, in the latter case, the behavioral performance is copied from the *source* to the *target* face appearance, while the identity remains intact. In face swapping, we have essentially the opposite situation: The behavioral performance is left intact, while the identity is copied from the *source* to the *target* appearance. Recent studies have shown that, while face reenactment manipulations are often difficult to detect by human observers, face swaps are typically easy to spot [RCV*19], which illustrates the challenges inherent in our present work.

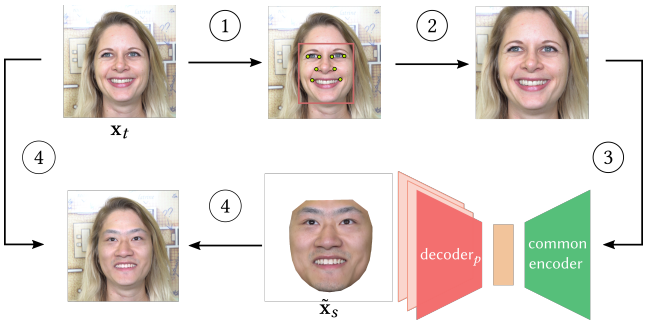

Figure 2: *A schematic of the full pipeline for swapping a source face of identity s onto a person t ≠ s. In steps (1) and (2) we pre-process the input by cropping and normalizing the face. In step (3) the pre-processed image is fed into the common encoder and decoded with corresponding decoder D_s. In (4) we use our multi-band blending to swap the target with the source face.*

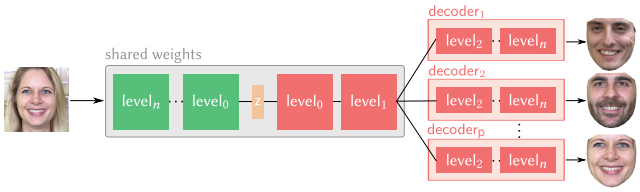

Figure 3: *Single-encoder, multi-decoder network architecture.*

3. High-Resolution Face-Swapping Pipeline

We now present our complete method for performing photo-realistic face swaps at megapixel resolution (see Fig. 2). The following outline summarizes the steps we take for exchanging identity *s* (the appearance *source*) and identity *t* (the behavior and background *target*):

1. For image $\mathbf{x}_t$, detect the face and localize the facial landmarks.
2. Normalize the face to 1024×1024 resolution, saving normalization parameters.
3. Feed the normalized face into the network and save the output $\tilde{\mathbf{x}}_s$ of the *s*-th decoder.
4. Reverse image normalization using saved parameters from step 2. on image $\tilde{\mathbf{x}}_s$ and blend the resulting image with image $\mathbf{x}_t$ using the introduced compositing method.

The most crucial component of the pipeline is the model itself, which we discuss next (Section 3.1). We then discuss our landmark alignment and stabilization method (Section 3.2), which allows for temporal consistency in the swapped images. This is followed by a description of our light- and contrast-preserving compositing procedure (Section 3.3).

3.1. Network Architecture and Training

Identity transformation is achieved through a domain-transfer approach. Images from all identities are embedded in a shared latent space using a common encoder, and these embeddings are then mapped back into pixel space using the decoder corresponding

to the desired source appearance. While domain-transfer and face-swapping approaches are typically used to switch between exactly two spaces, in this paper we generalize this idea to P arbitrary domains (see Fig. 3). As a result, the encoding part of the network is shared, while the decoding path is forked into P domains. We refer to this architecture as a *comb model*, with the various decoders representing the "teeth" of the comb. In addition to the benefit of being able to have multiple source-target pairs handled by a single model, our ablation study (Section 5) shows that training the model with multiple identities improves the fidelity of the expressions in comparison to a two-way model. Furthermore, since the multi-way decoder allows for generating different outputs, these can correspond to various identities or the same identity in different lighting conditions. As we are able to use a single network, this leads to a reduction of training time compared to generating all possible pairs of appearances with two-way networks.

The training of our network is executed using a progressive regime, which we have adapted to work in a non-adversarial setting. This process starts from coarse, low-resolution images formed by down-sampling high-resolution input data and then gradually expands the network's capacity as higher-resolution images are used for training. We provide a more detailed explanation of the progressive-training process in the appendix. Note that the ultimate output of the network is limited by the resolution of the training input. When high-resolution training data are lacking, super-resolution methods [WPM*18] are worth exploring as either a pre-processing step to create richer training data or as a post-processing step to adjust the model's output. It is worth noting, however, that super-resolution methods specific to face data, many of which employ task-specific priors [CTL*18], are likely to produce superior results.

We partition the data $\mathbf{X}$ into P subsets, where each subset corresponds to an individual identity. We normalize all available examples to 1024×1024 resolution. Note that in the progressive regime, these images will be downsized in the initial stages of training, while 1024×1024 is the final resolution (see the appendix for details). This process is performed by applying an affine transformation to the face image, which aligns the position of a set of defined localized landmarks to the average landmark locations at the desired resolution. In our implementation, we use outer eye corners, outer nose points, and outer mouth corners from the standard 68 landmark point set as our reference.

For each normalized face image we create a binary mask $\mathbf{m}_{\mathbf{x}_p}$ that is used during the training process. This mask is delimited by the convex hull of the set of standard 68 facial landmarks fit to $\mathbf{x}_p$. The mask is additionally upscaled by 10 percent to ensure that important features such as eyebrows are not missed due to slight misalignment of the landmarks. The values inside the convex hull are set to 1, while the values outside the hull are set to 0.

All P identities are encoded via a shared encoder, E. We create P decoders, D_p, $p \in \{1,\dots,P\}$, to produce the pixel-space basis representations of the input identities. Let $\mathbf{x}_p \in \mathbf{X}_p \subset \mathbf{X}$ be an image belonging to identity p. We then have $\tilde{\mathbf{x}}_p = D_p(E(\mathbf{x}_p)) \approx \mathbf{x}_p$, as in a standard autoencoder.

Training is performed by feeding the network images from all P subsets in random order. The data is augmented by random translation, rotation and scaling. We only minimize the reconstruction loss on the output assigned to the currently evaluated face identity. We also do not enforce any swap or cycle consistency. Because we are interested only in the interior of the face, we multiply the input and the image output by the mask $\mathbf{m}_{\mathbf{x}_p}$. Thus our level-dependent loss function we minimize becomes

$$\mathcal{L}_l = \sum_{p=1}^{P} \frac{1}{|\mathbf{X}_p|} \sum_{\mathbf{x}_p \in \mathbf{X}_p} f_l\left(\mathbf{x}_p \odot \mathbf{m}_{\mathbf{x}_p}, \tilde{\mathbf{x}}_p \odot \mathbf{m}_{\mathbf{x}_p}\right), \tag{1}$$

where $\mathbf{x}_p$ is the ground-truth image, $\mathbf{m}_{\mathbf{x}_p}$ is the mask of the face, $\tilde{\mathbf{x}}_p = D_p\left(E\left(\mathbf{x}_p\right)\right)$ is the reconstruction, and $\odot$ represents element-wise multiplication. For levels $0 \leq l \leq 2$, we set f_l to be SSIM, a *structural similarity index* introduced by Wang et al. [WBSS*04]. For implementation reasons (we use the TensorFlow implementation of SSIM), we upscale the input images to 16×16 during training the first two levels of the model. For the remaining levels, we set f_l to be MS-SSIM, the multi-scale version of this index [WSB03].

3.2. Face Alignment and Landmark Stability

Most face-alignment methods (e.g. [KNT17]) are designed to improve the accuracy of public benchmarks, which are typically made up of single images. Although some video benchmarks are available [SZC*15], the resulting alignment algorithms are usually not tested for temporal consistency. While this is not a problem in many use cases, in our task any temporal inconsistencies lead to significant degradation of the realism of the resulting swaps. Since face normalization is based on facial landmarks, small differences in network inputs result in slightly different network outputs. With most facial-alignment algorithms' being developed on relatively low-resolution data, minor inconsistencies from frame to frame are amplified when one moves to high resolution, leading to temporally unstable results that become particularly evident at resolutions of 512×512 pixels and higher. This can be observed as a slight trembling and deformation of various facial features.

One possible solution to this problem could be to train a facial-feature localization algorithm in higher resolutions. However, most existing data sets are insufficient for this task and this would not necessarily eliminate the problem of trembling.

We instead propose a method to stabilize existing landmark-localization algorithms to attenuate problems when they are applied to high-resolution sequential data. More specifically, we perform an initial detection and alignment and note the width w of the face bounding box. We then re-initialize the original bounding box n times by perturbing it by βw pixels in various directions of the image plane, where β is a small value to control the range of the perturbations. We repeat the face-alignment procedure for each translation and average the resulting sets of localized landmark points. This strategy effectively amounts to a variance-reduction technique to offset landmark-location uncertainty amplified by operating in high resolution. In this case, this amounts to creating an ensemble of n landmark estimates and averaging their results. We found that using $\beta = 0.05$ and $n = 9$ when working at 1024×1024 resolution removed all visible temporal artifacts (see supplemental video).

3.3. Contrast-Preserving, Multi-Band Compositing

Properly compositing a source face onto a target image is challenging even if the faces are already in perfect geometric alignment, with the pose and facial expressions exactly matching. This is due to photometric misalignment, which can result in clearly visible seams when simply pasting a source onto a target.

As a remedy, many existing approaches use Poisson blending [PGB03], which tries to achieve seamless cloning by operating in the gradient domain. This method often achieves passable results, but if the lighting of the source and the target faces is different, this may introduce visible artifacts in the face interior.

Multi-band blending [BA83], as recently used by Thies et al. [TZN*15] in the context of face-image compositing is a competing approach to Poisson blending. In this method a given mask defines the area to be cloned from the source to target image. A smooth transition between the two images is ensured by decomposing them into a Laplacian pyramid and then, at each level of the pyramid, smoothing the transition near the boundaries of the given mask. Such an approach does not ensure, however, that the cloned area will match the target lighting, which is desirable in our application. With this in mind, we copy the two coarsest (i.e. low-frequency) levels of the target's Laplacian pyramid and blend only the remaining, more detailed levels. The final image is then obtained by reconstructing from the blended Laplacian pyramid.

We also enforce that the boundary smoothing effect is propagated only into the interior of the face. This way we ensure that the outer face outline is not smoothed away by the blending procedure.

While our modified multi-band blending procedure is well suited to preserve low-frequency lighting, we observed that it can still lead to uncanny compositing results in cases in which source and target are captured in considerably different conditions. The contrast in source and target varies greatly in those cases, and this is not accounted for by multi-band blending alone.

Therefore, we additionally align the amount of contrast in the generated source face to match the contrast of the target. We estimate the contrast of an image using the Global Contrast Factor (GFC) [MNN*05]. GFC provides a scalar measure of contrast based on a weighted sum of local contrast values at multiple image scales. We calculate a contrast coefficient as the ratio of the GFC of the target image and the GFC of the network output. Finally, we multiply each pixel of our generated image by this coefficient.

This allows us to obtain high-quality compositing results with robustness to different capture conditions. A detailed comparison is presented in our ablation study (Section 5).

To ensure that the edges of the face generated by the network are not transferred to the cloned face, for instance due to different head sizes, the blending mask should be chosen carefully. We shrink the boundary of the mask defined by the convex hull of the outer face landmarks so that the resulting mask does not cover the outer edges of the face.

4. Experiments

4.1. Data Acquisition and Training Details

For testing purposes we sought a high-resolution video data set gathered under a variety of lighting and pose conditions, featuring different genders and facial hair styles. Publicly available high-resolution data sets usually consist of only still images of celebrities. For this reason we decided to create our own data set. We recorded nine volunteers—seven males and two females—in different lighting conditions, including controlled frontal and side light, non-controlled natural light, and outdoor footage. We recorded 4K (3840×2160 at 25 fps) videos using a Sony ILCE-7SM2 camera. During indoor, controlled-light recordings we asked the volunteers to read a short public-domain text from a page held by the camera operator. For the remaining sessions we asked participants only to describe the weather and their surroundings. Each sequence was approximately two to four minutes long.

From the data set we chose six people, with two of these people captured in two different lighting conditions, forming eight total sets that were used to create our eight-output model. Each level of the network was trained until 10^5 images of each person were presented. All experiments were performed at 1024×1024 final image resolution. We use the Adam optimizer [KB15] with a learning rate of 10^{-4}. Training the full 1024×1024 network for two identities takes about three days using one GeForce 1080Ti GPU.

4.2. Comparison with the State of the Art

We compare our progressive comb model with three open-source approaches that currently constitute the state of the art in facial appearance transfer. Specifically, we tested the work of Nirkin et al. [NMT*18], an open-source implementation of the original "DeepFakes" method (`https://github.com/deepfakes/faceswap`), and a model from the open-source repository DeepFaceLab (`https://github.com/iperov/DeepFaceLab`). The first method employs 3D morphable models, while the latter two implement Y-shaped autoencoder architectures. The "DeepFakes" method is known for producing convincing face swaps and has achieved broad media attention. For DeepFaceLab, we chose the "Stylized Autoencoder" (SAE), as we consider this to be that repository's best performing model. In this architecture, the style transfer relies on matching the mean and standard deviation of the target image with the original face, both for color balancing and to mitigate the effect of seams. In this approach the face and the background are modeled together. Both DeepFakes and DeepFaceLab use Poisson blending as implemented in OpenCV [PGB03] for blending the source image into the target.

We swapped faces for five pairs of people. For each person we used the same images we used to train our eight-way model. The algorithm of Nirkin et al. [NMT*18] relies on morphable models and does not require prior training. Also, it should be noted that this algorithm is intended specifically for image-to-image swapping. To achieve the best possible result with this method, we chose the neutral expression face as a source for the swapping procedure (second column in Fig. 4). DeepFaceLab, DeepFakes, and our algorithm were each trained to convergence in an unsupervised fashion on the same set of images.

Figure 4: *Comparison of the face swapping methods. From the left: target image, source identity, our model in* 1024×1024 *resolution, our model in* 256×256 *resolution, DeepFakes, DeepFaceLab, Nirkin et el.*

The comparison of face swapping for the chosen images is shown in Fig. 4. (See also the supplemental material for a video comparison among the methods.) Because of the GPU memory requirements and software limitations of the DeepFaceLab implementation, the highest possible resolution we were able to achieve on an 11GB GPU was 256×256 pixels. For DeepFakes we were able to produce 128×128 images. For the morphable models approach, we achieved a resolution of 500×500 pixels. Note that the source images were directly used for swapping only for Nirkin et al. [NMT*18], while the remaining methods performed swapping based on the network-generated images.

The experiments show that the morphable models are also able to produce faces at relatively high resolution (500×500), but they introduce artifacts that tend to make the face look unrealistic. Furthermore, we noticed that this approach does not preserve temporal consistency, and the model output can change rapidly even if very small transformations of the target image occur. The images produced by our implementation of this method were approximately half the resolution of the original image. We therefore upsampled them to match the resolution of the original using a Lanczos filter.

Both the DeepFaceLab and DeepFakes models behave similarly. In some situations, the seams of the cloned image are visible as an effect of using Poisson blending for non-matching boundaries. These effects and other artifacts other methods produce can be seen in Fig. 4. It is also worth noting that each of these models had to be trained separately for each pair of swaps, while our algorithm was trained for all people simultaneously.

5. Ablation Study

We performed several experiments to visualize the effect of different aspects of our network architecture and algorithms on the quality of the facial swaps:

1. the effect of progressive training versus training the full network all at once
2. the effect of using a multi-way comb-model instead of separate two-way models
3. a comparison of our contrast-preserving, multi-band compositing method with Poisson blending
4. the effect of our landmark stabilization method

A separate study of the number of shared decoder layers is presented in the appendix.

Progressive training. Although a model trained fully end-to-end at the highest resolution is capable of producing reasonable face images, it often does so without adequate regard to the target behavior to be captured. In Fig. 5 we show representative examples of this effect. The center image, produced via progressive training, closely matches the pose and expression of the input face on the left, while the rightmost image, produced via end-to-end training, effectively loses pose information and even introduces artifacts. (Interestingly, these artifacts worsened with additional exposure to the data, ultimately leading to a complete performance collapse after 80K iterations.)

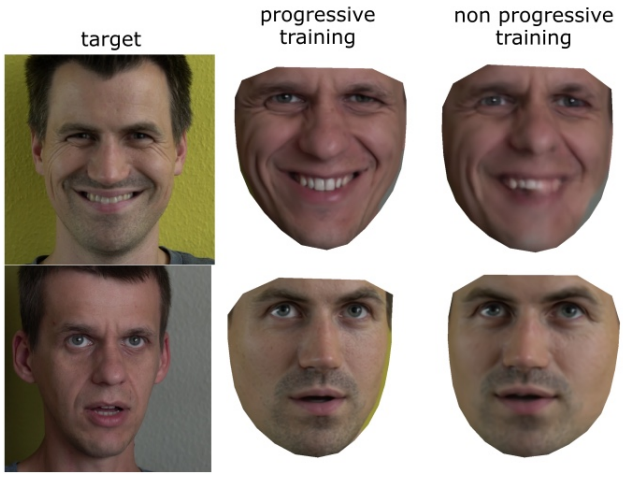

Figure 5: *Effect of training with and without progressive training. From the left (in columns): input image, output of the network trained with progressive training, output of the network without progressive training. Notice that the pose and expression do not match when the network is trained without progressive training.*

Comb model. In this experiment we trained an eight-output model, using data from six individuals, with two additional "identities" coming from data gathered from two of these six in radically different lighting conditions. As a comparison, we trained three separate two-output models for randomly chosen pairs from our eight-way data set.

In Fig. 7 we show the benefit of using a multi-way comb model compared with the two-way model. Although we controlled training across all models so that each model had the same number of iterations on the data, we noted that the multi-way model was better able to capture certain expressions in cases in which data for the source appearance was lacking. For example, the multi-way model was better able to reproduce closed eyes and protruding tongues when this was part of the target data but not part of the source data. To illustrate the lack of data for these expressions, in this figure we also show two nearest-neighbor results for our eight-way model's output, one based on facial landmark distance and the other based on distance in RGB space. Additional examples of the swapping between identities with the eight-way model are presented in Fig. 6.

Poisson blending versus our compositing method. In Section 3.3 we introduced our contrast- and light-preserving compositing algorithm. In Fig. 8 we compare the performance of our method with Poisson blending. Our method better preserves the global lighting characteristics of the target face, while the Poisson algorithm can cause a certain "bleaching" or washing-out effect.

In Fig. 9 we show a comparison of classical multi-band blending with our approach. Copying the two smallest Laplacian pyramid levels ensures that the global lighting characteristics of the target image are preserved. Copying the four smallest levels of the pyramid, on the other hand, introduces artifacts that manifest as a mixture of the target image and the network output. The figure also shows that contrast correction is an important factor in the realism of the generated images.

Facial landmark stability. For facial feature alignment we used a TensorFlow implementation of Deep Alignment Network (DAN) [KNT17]. To measure the stability of the aligned landmarks to random factors, we perturbed input images by simple, invertible image transformations to determine if the detected landmarks were assigned to the same semantic locations of the face. We chose a random set of 100 1024×1024 face images from our gallery and localized facial landmarks for each image. We then perturbed each picture using a random affine transformation performing rotation, scaling, and translation. The facial landmarks from the unperturbed images were treated as ground truth, and we compared these values with the landmarks fitted to the perturbed images after performing matching transformations on the ground-truth landmarks. We used L^2 distance as our error measure, and we cumulatively averaged the results over 10 random perturbations for each image. The results of our experiments are shown in Fig. 10.

We noted that the error plateaus at around 10 random perturbation initializations, as described in the methods section.

6. Limitations and Discussion

While we are able to produce compelling, photo-realistic transfers of facial appearance in high resolution, there are still a few limitations to our approach. Expressions and poses that are typically not well captured in the data, such as extreme profile views, can lead to imperfect results including blur and other artifacts. In those cases, a straightforward remedy is to capture more extensive data and ensure that certain expressions and side views are included. A more principled approach would be to incorporate more complete information about facial appearance and behavior into the model to facilitate the process of filling in missing information in the training data for a specific person.

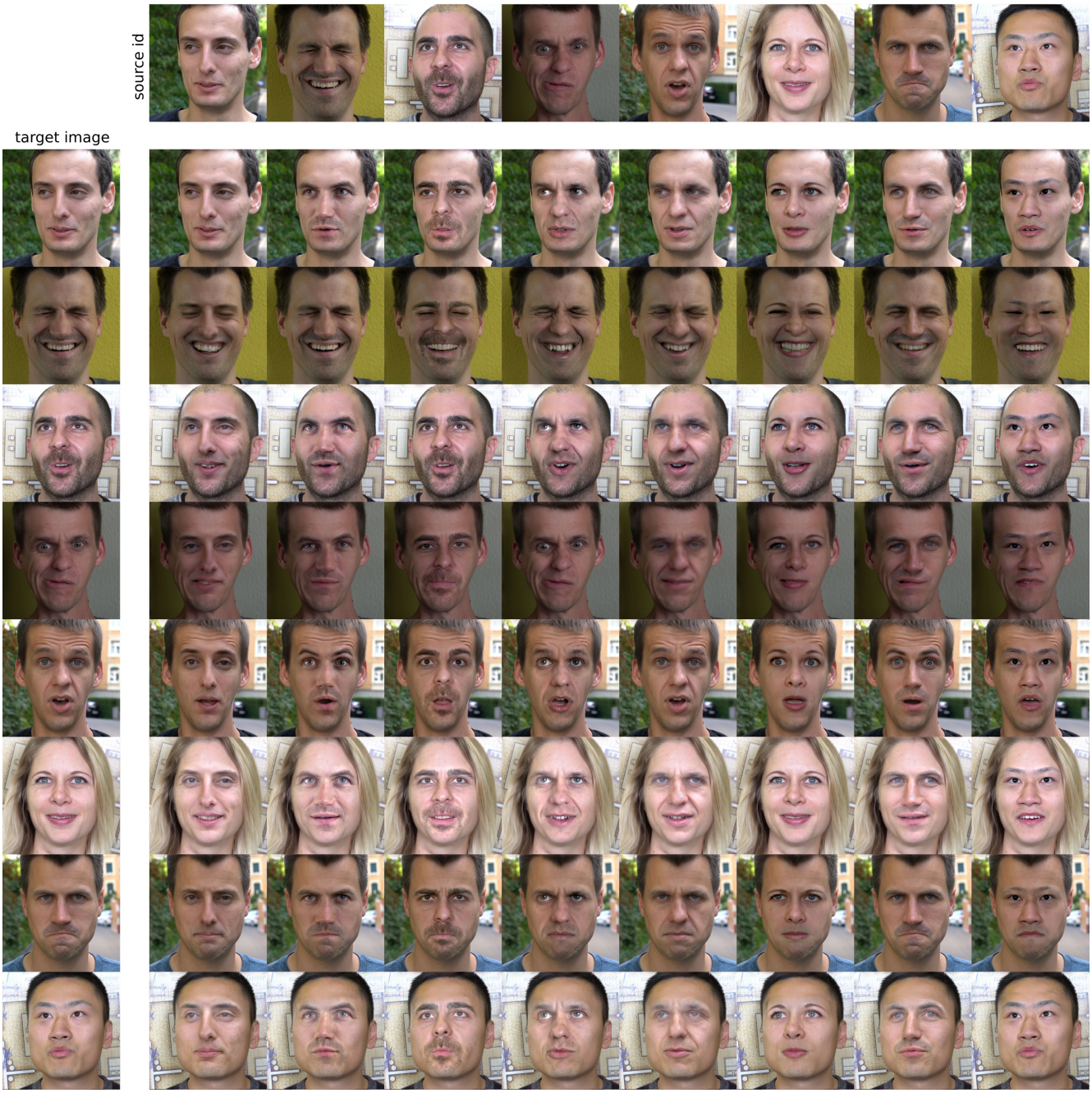

Figure 6: *Swapping results for the eight-way comb network model. Notice that two people have double outputs, corresponding to data collected in different lighting conditions.*

Example failure cases are presented in Fig. 11. Note also that despite the fourth and fifth swaps' (fifth and sixth column) corresponding to the same subject, the results are different, in particular with the level of eye opening present in the image. This is due to the the fact that for the fourth swap the person was captured in controlled, indoor-lighting conditions, while for the fifth swap the same person was captured in an outdoor settings, where the sunlight caused him to squint his eyes.

One possible issue with using multi-band blending is that because we copy only the low-resolution elements of the face appearance, the method is necessarily capable of transferring only the low-frequency characteristics of the lighting, which could prove inadequate in some cases. (This same limitation applies to Poisson

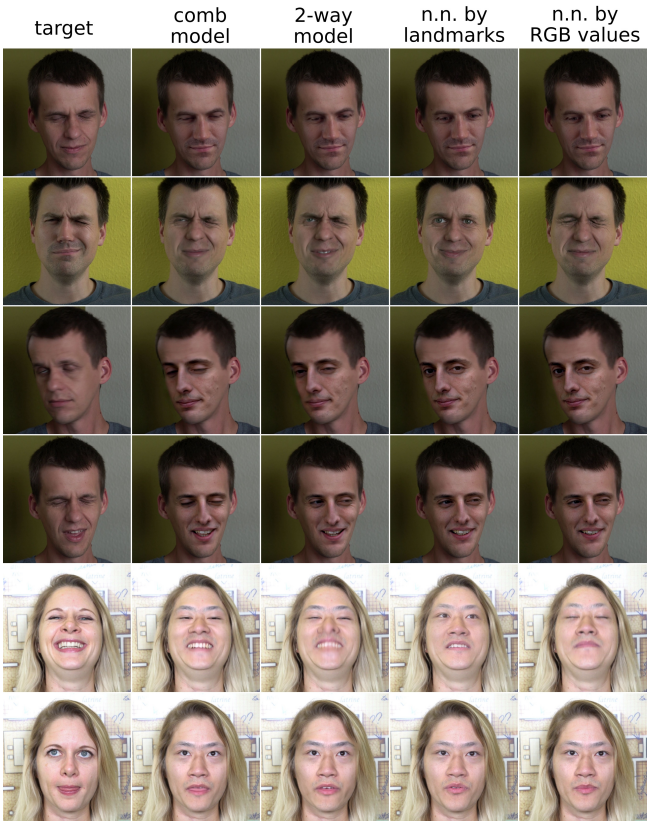

Figure 7: *Comparison of the eight-way versus the two-way model. Columns correspond to (respectively) target images, images swapped with the eight-way model, images swapped with the two-way model, nearest neighbour to the eight-way model result (landmark space), nearest neighbour to the eight-way model result (RGB space).*

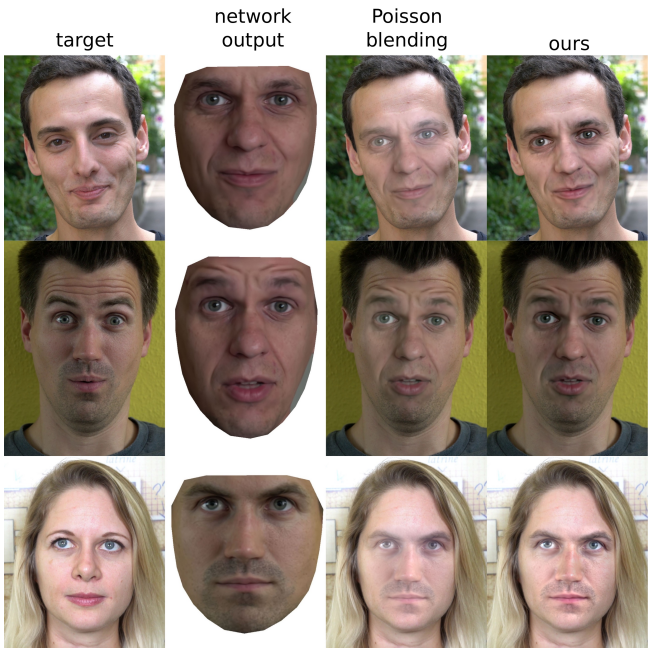

Figure 8: *Comparison of face swapping with our compositing and Poisson methods. From left to right: target image, network output, Poisson blending and our compositing.*

blending, however.) While our contrast-preservation step adjusts for this, an additional solution would be to allow for the transfer of high-frequency lighting elements by decoupling albedo from the other lighting characteristics through the learning of additional image channels, a question we will address in future work.

Another limitation, not only of our method but also of the other state-of-the-art approaches we examined, is that current facial-appearance transfer methods focus on replacing the face while it retains the original head shape. Transferring the head shape could be an interesting opportunity for future work, which would put a strong emphasis on performing correct background in-painting in cases in which the face is smaller.

It is also worth noting that the present method is incapable of performing convincing swaps of people wearing glasses. This is not a matter of being unable to *render* glasses using our method but rather one of how the face is blended with the surrounding image afterward. Although it is possible that the careful selection of source and target data featuring matching eyewear could produce passable results, it would not succeed in the general case and has not been a goal of the present work.

Finally, we mention that a multi-way model will require increased training time relative to a two-way model, roughly linear in the amount of data required for the represented identities. In applications in which multi-way swaps are the goal, this training time is simply part of the bargain and, as we mentioned earlier, is actually *less* than what would be required to train multiple two-way models to perform the same task. However, this additional training cost may be worth paying even in two-way swapping applications in cases where source data may be lacking but realism is at a premium. As we demonstrated in Section 5, multi-way training allows for some degree of improved synthesis of expressions and behavior even when those expressions are not part of the source-data observations.

7. Conclusions

In this work, we presented a novel approach for the unsupervised learning of multi-subject face swapping. Our method is, to our knowledge, the first to achieve convincing face-swapping results on high-resolution video in the megapixel-and-beyond domain.

We demonstrated the importance of progressive training in high-resolution face swapping. We showed that using our landmark stabilization procedure ameliorates unrealistic trembling effects and other temporal instability that can occur when operating in the high-resolution domain.

We further showed the benefits of a multi-way network beyond the convenience of allowing for multiple pair swaps with a single

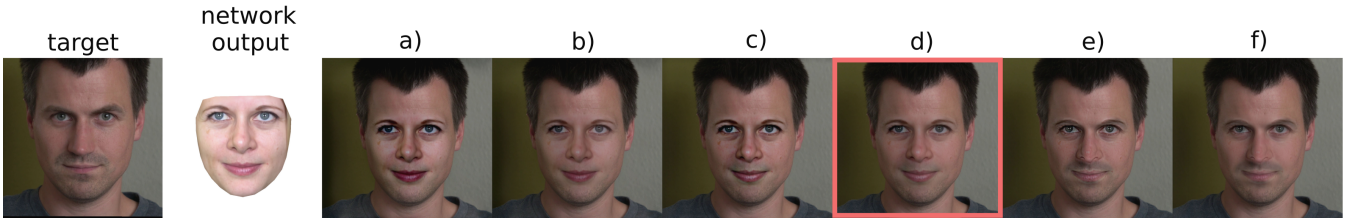

Figure 9: *Comparison of a standard multi-band blending method, with and without contrast correction and with various numbers of transferred Laplacian pyramid levels. Image a) corresponds to standard multi-band blending, while in image b) contrast correction is applied before multi-band blending. Images c) and e) illustrate, respectively, the effect of copying the two and four smallest levels of the Laplacian pyramid from the target image. Images d) and f) present the effect of applying contrast correction to images c) and e). In our work, we use the option represented by choice d).*

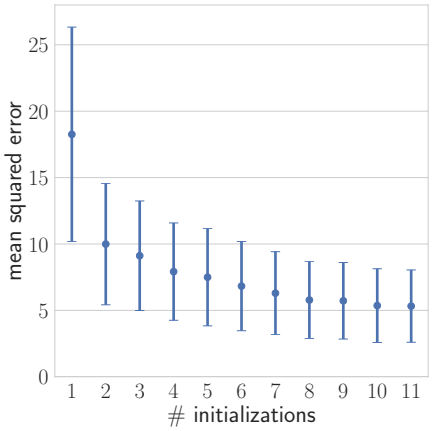

Figure 10: *Effect of averaging landmarks localized with different random perturbation initializations. The horizontal axis corresponds to the number of initializations, while the vertical axis corresponds to the mean squared error in pixels of the detected landmarks relative to the "stable" landmark positions in* 1024×1024 *resolution.*

model. By increasing the number of identities presented to the network, we can achieve higher fidelity of the swapped expressions relative to using only a pair of identities. We attribute this benefit to the learning of richer representations of faces that allow for generalization to occur in cases in which expression data for specific individuals is lacking.

Finally, we showed that our proposed compositing method, consisting of contrast normalization and a multi-band, light-preserving blending procedure, can be used to overcome many problems with different lighting conditions in the data. This leads to results that, in our judgment, represent a considerable advance in the pursuit of face-swapping visual effects using neural methods.

References

[ARL*09] ALEXANDER, O., ROGERS, M., LAMBETH, W., et al. "Creating a Photoreal Digital Actor: The Digital Emily Project". *2009 Conference for Visual Media Production*. Nov. 2009, 176–187 3.

[BA83] BURT, PETER J. and ADELSON, EDWARD H. "A Multiresolution Spline with Application to Image Mosaics". *ACM Trans. Graph.* 2.4 (Oct. 1983), 217–236. ISSN: 0730-0301 5, 14.

[BSVS04] BLANZ, VOLKER, SCHERBAUM, KRISTINA, VETTER, THOMAS, and SEIDEL, HANS-PETER. "Exchanging Faces in Images". *Computer Graphics Forum* 23.3 (2004), 669–676 3.

[BV*99] BLANZ, VOLKER, VETTER, THOMAS, et al. "A morphable model for the synthesis of 3D faces." *Siggraph*. Vol. 99. 1999. 1999, 187–194 3.

[CTL*18] CHEN, YU, TAI, YING, LIU, XIAOMING, et al. "Fsrnet: End-to-end learning face super-resolution with facial priors". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 2492–2501 4.

[DNWG17] DONG, HAO, NEEKHARA, PAARTH, WU, CHAO, and GUO, YIKE. "Unsupervised Image-to-Image Translation with Generative Adversarial Networks". *CoRR* abs/1701.02676 (2017) 2.

[DSJ*11] DALE, KEVIN, SUNKAVALLI, KALYAN, JOHNSON, MICAH K, et al. "Video face replacement". *ACM Transactions on Graphics (TOG)*. Vol. 30. 6. ACM. 2011, 130 2, 3.

[ER97] EKMAN, PAUL ED and ROSENBERG, ERIKA L. "What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)." (1997) 3.

[GPM*14] GOODFELLOW, IAN, POUGET-ABADIE, JEAN, MIRZA, MEHDI, et al. "Generative adversarial nets". *Advances in neural information processing systems*. 2014, 2672–2680 2.

[GSZ*18] GENG, JIAHAO, SHAO, TIANJIA, ZHENG, YOUYI, et al. "Warp-guided GANs for single-photo facial animation". Vol. 37. Dec. 2018, 1–12 3.

[GVS*15] GARRIDO, PABLO, VALGAERTS, LEVI, SARMADI, HAMID, et al. "Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track". *Computer graphics forum*. Vol. 34. 2. Wiley Online Library. 2015, 193–204 3.

[IZZE17] ISOLA, PHILLIP, ZHU, JUN-YAN, ZHOU, TINGHUI, and EFROS, ALEXEI A. "Image-to-image translation with conditional adversarial networks". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 1125–1134 2.

[KALL18] KARRAS, TERO, AILA, TIMO, LAINE, SAMULI, and LEHTINEN, JAAKKO. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". *International Conference on Learning Representations*. 2018 2.

Figure 11: *Failure cases. Left column shows the original image, following columns show the "comb" network swap results. First row corresponds to occlusion, where in some cases the hand is blended with the face region creating "transparency" effect. In the second row, in some cases the face region is blurred due to the lack of correspondences of extreme pose images in the training set. In the third row there are some artifacts at the border of the blended face region. Fourth row shows problematic cases causes by an expression that was not presented by all of the people.*

[KB15] KINGMA, DIEDERIK P. and BA, JIMMY. "Adam: A Method for Stochastic Optimization". *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015 5.

[KCT*18] KIM, HYEONGWOO, CARRIDO, PABLO, TEWARI, AYUSH, et al. "Deep video portraits". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 163 3.

[KEZ*19] KIM, HYEONGWOO, ELGHARIB, MOHAMED, ZOLLHÖFER, MICHAEL, et al. "Neural Style-Preserving Visual Dubbing". *arXiv preprint arXiv:1909.02518* (2019) 3.

[KLA18] KARRAS, TERO, LAINE, SAMULI, and AILA, TIMO. "A style-based generator architecture for generative adversarial networks". *arXiv preprint arXiv:1812.04948* (2018) 2.

[KNT17] KOWALSKI, MAREK, NARUNIEC, JACEK, and TRZCINSKI, TOMASZ. "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, 7.

[KSDT17] KORSHUNOVA, IRYNA, SHI, WENZHE, DAMBRE, JONI, and THEIS, LUCAS. "Fast Face-Swap Using Convolutional Neural Networks". *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 3697–3705 2.

[LBK17] LIU, MING-YU, BREUEL, THOMAS, and KAUTZ, JAN. "Unsupervised Image-to-Image Translation Networks". *Advances in Neural Information Processing Systems 30*. Ed. by GUYON, I., LUXBURG, U. V., BENGIO, S., et al. Curran Associates, Inc., 2017, 700–708 2.

[LWLT12] LIN, YUAN, WANG, SHENGJIN, LIN, QIAN, and TANG, FENG. "Face Swapping under Large Pose Variations: A 3D Model Based Approach". July 2012, 333–338. ISBN: 978-1-4673-1659-0 3.

[MNN*05] MATKOVIC, KRESIMIR, NEUMANN, LÁSZLÓ, NEUMANN, ATTILA, et al. "Global Contrast Factor - a New Approach to Image Contrast". *Computational Aesthetics*. 2005 5.

[NKH19] NIRKIN, YUVAL, KELLER, YOSI, and HASSNER, TAL. "Fsgan: Subject agnostic face swapping and reenactment". *Proceedings of the IEEE International Conference on Computer Vision*. 2019, 7184–7193 3.

[NMT*18] NIRKIN, YUVAL, MASI, IACOPO, TRAN TUAN, ANH, et al. "On Face Segmentation, Face Swapping, and Face Perception". May 2018, 98–105. DOI: 10.1109/FG.2018.00024 2, 3, 5, 6.

[NYM18a] NATSUME, RYOTA, YATAGAWA, TATSUYA, and MORISHIMA, SHIGEO. "FSNet: An Identity-Aware Generative Model for Image-based Face Swapping". *Proc. of Asian Conference on Computer Vision (ACCV)*. Springer, Dec. 2018 2, 3.

[NYM18b] NATSUME, RYOTA, YATAGAWA, TATSUYA, and MORISHIMA, SHIGEO. "RSGAN: face swapping and editing using face and hair representation in latent spaces". Aug. 2018, 1–2. DOI: 10.1145/3230744.3230818 2.

[PAM*18] PUMAROLA, ALBERT, AGUDO, ANTONIO, MARTINEZ, ALEIX M, et al. "Ganimation: Anatomically-aware facial animation from a single image". *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 818–833 3.

[PGB03] PÉREZ, PATRICK, GANGNET, MICHEL, and BLAKE, ANDREW. "Poisson image editing". *ACM Trans. Graph.* 22 (2003), 313–318 5.

[RCV*19] RÖSSLER, ANDREAS, COZZOLINO, DAVIDE, VERDOLIVA, LUISA, et al. "FaceForensics++: Learning to Detect Manipulated Facial Images". (Jan. 2019) 3.

[SSK17] SUWAJANAKORN, SUPASORN, SEITZ, STEVEN M, and KEMELMACHER-SHLIZERMAN, IRA. "Synthesizing obama: learning lip sync from audio". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 95 3.

[SSSH17] SHU, ZHIXIN, SHECHTMAN, ELI, SAMARAS, DIMITRIS, and HADAP, SUNIL. "Eyeopener: Editing eyes in the wild". *ACM Transactions on Graphics (TOG)* 36.1 (2017), 1 3.

[SYH*17] SHU, ZHIXIN, YUMER, ERSIN, HADAP, SUNIL, et al. "Neural face editing with intrinsic image disentangling". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, 5541–5550 3.

[SZC*15] SHEN, JIE, ZAFEIRIOU, STEFANOS, CHRYSOS, GRIGORIS G, et al. "The first facial landmark tracking in-the-wild challenge: Benchmark and results". *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015, 50–58 4.

[TL18] TRAN, LUAN and LIU, XIAOMING. "Nonlinear 3D face morphable model". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 7346–7355 3.

[TZN*15] THIES, JUSTUS, ZOLLHÖFER, MICHAEL, NIESSNER, MATTHIAS, et al. "Real-time Expression Transfer for Facial Reenactment". *ACM Trans. Graph.* 34.6 (Oct. 2015), 183:1–183:14. ISSN: 0730-0301 5.

[TZS*16] THIES, J., ZOLLHÖFER, M., STAMMINGER, M., et al. "Face2Face: Real-time Face Capture and Reenactment of RGB Videos". *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2016 3.

[WBSS*04] WANG, ZHOU, BOVIK, ALAN C, SHEIKH, HAMID R, SIMONCELLI, EERO P, et al. "Image quality assessment: from error visibility to structural similarity". *IEEE transactions on image processing* 13.4 (2004), 600–612 4.

[WLZ*18] WANG, TING-CHUN, LIU, MING-YU, ZHU, JUN-YAN, et al. "Video-to-Video Synthesis". *Advances in Neural Information Processing Systems (NeurIPS)*. 2018 2.

[WPM*18] WANG, YIFAN, PERAZZI, FEDERICO, MCWILLIAMS, BRIAN, et al. "A fully progressive approach to single-image super-resolution". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, 864–873 4.

[WSB03] WANG, Z., SIMONCELLI, E. P., and BOVIK, A. C. "Multiscale structural similarity for image quality assessment". *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. Nov. 2003, 1398–1402 Vol.2 4.

[YHL*18] YAN, SHUQI, HE, SHAORONG, LEI, XUE, et al. "Video Face Swap Based on Autoencoder Generation Network". *2018 International Conference on Audio, Language and Image Processing (ICALIP)*. IEEE. 2018, 103–108 2.

[YWS*11] YANG, FEI, WANG, JUE, SHECHTMAN, ELI, et al. "Expression flow for 3D-aware face component transfer". *ACM transactions on graphics (TOG)* 30.4 (2011), 60 3.

[ZPIE17] ZHU, JUN-YAN, PARK, TAESUNG, ISOLA, PHILLIP, and EFROS, ALEXEI A. "Unpaired image-to-image translation using cycle-consistent adversarial networks". *Proceedings of the IEEE international conference on computer vision*. 2017, 2223–2232 2.

[ZSBL19] ZAKHAROV, EGOR, SHYSHEYA, ALIAKSANDRA, BURKOV, EGOR, and LEMPITSKY, VICTOR. "Few-Shot Adversarial Learning of Realistic Neural Talking Head Models". *arXiv preprint arXiv:1905.08233* (2019) 3.

[ZTD*18] ZHAO, YUCHENG, TANG, FAN, DONG, WEIMING, et al. "Joint face alignment and segmentation via deep multi-task learning". *Multimedia Tools and Applications* (Jan. 2018). ISSN: 1573-7721. DOI: 10.1007/s11042-018-5609-1 2.

[ZTG*18] ZOLLHÖFER, MICHAEL, THIES, JUSTUS, GARRIDO, PABLO, et al. "State of the art on monocular 3D face reconstruction, tracking, and applications". *Computer Graphics Forum*. Vol. 37. 2. Wiley Online Library. 2018, 523–550 3.

Appendix A: Appendix

In this appendix we share additional details and insights about the design and functionality of our face-swapping pipeline. In the first section we report an experiment showing the effect of different choices in the number of shared decoder layers on the network output. Next we provide details about the progressive-training regime that played a key role in producing our high-resolution results along with information about used hardware. We also present the algorithm for our multi-band blending. Finally we provide more insights about the network capabilities in the context of interpolating between images in the latent space. The structure of our network architecture is presented in Table 1.

Number of common layers

In principle, the split to the person-specific decoders can be placed directly after the encoded latent vector (the *bottleneck*, which in our case is in $\mathbb{R}^{512}$) or later in the network following a series of weight-sharing layers. We observed that if we perform the split at level 0 (which corresponds to a $512 \times 4 \times 4$ feature map), directly after the bottleneck, and before level 3 (which corresponds to a $512 \times 32 \times 32$ feature map), the generated images in most cases look quite realistic. However, we also observed that the more levels that are shared, the more the generated face departs from the source appearance and the more it resembles the input image. If the split comes too early, such as right after the latent space, or too late, such as after the third level, then the network occasionally introduces undesirable artifacts. Results of training with different split points are presented in Fig.12. In our implementation we chose level 1 as our split point, as it seemed to provide the best trade-off between source and target fidelity.

Progressive training

Our model is trained in a progressive regime, starting from coarse, low-resolution 4×4 pixel images and then gradually expanding the network's capacity as higher-resolution images are used for training, up to 1024×1024 pixels. The base architecture, which focuses on the lowest-resolution data, corresponds to "level 0" in Figures 3 and 13 and Table 1. Each new "level" of the network doubles input and output resolution by adding a composition of two convolutional layers and a down- or up-scaling layer in the encoder and decoder, respectively. During training, additional convolutional "to-RGB" layers are added to the end of the decoder portion of the network to transform multi-channel output to three-channel RGB output. Analogously, the beginning of the encoder part of the network includes "from-RGB" layers to accept image data at the current level's resolution. These intermediate-resolution to- and from-RGB layers are discarded after their respective level's training, leaving only those for the final resolution trained. The "shock" of expanding the network by adding new, untrained network components is attenuated by a gain parameter, $\alpha \in [0,1]$, which acts as a fader switch that gradually blends the activations of the new network components with those of the trained, smaller network. This gain parameter is increased linearly within its range over the course of a new level's training. This process is presented schematically in Fig. 13.

Hardware specification

All the models were trained on a single NVIDIA 1080Ti GPU workstation (Intel® Core™ i7-6700K CPU @ 4.00GHz).

Data manifold

The results we presented in Section 5 showed that the comb model can successfully reproduce certain source expressions for which there is no exactly matching target data. We further illustrate this capability in Fig. 14.

We chose a set of short video sequences and generated source-target swaps using our model. We then searched our source data for nearest neighbors of the generated images, namely using L^2 distance in pixel space, limited to the face area, with all faces aligned and normalized to 1024×1024 resolution. We observed that for many expressions there were indeed no corresponding images in the training set, meaning that the network was able to "hallucinate" and fill in some missing details.

We further experimented by swapping the target face with the nearest-neighbor images instead of our network-generated faces. Since we use the same blending technique, it is not surprising that individual frames look quite good. However, when we performed this procedure frame by frame, the resulting video contained considerable "jumps" due to multiple frames' corresponding to the same images in the training set or to images that departed significantly from the target expression.

We are further interested in the overall coherence of the data manifold induced by the encoder. More precisely, while we know that training examples are properly encoded, we are also interested in the area *between* these points.

To investigate this, we conducted the following experiment: We selected two training examples from a randomly chosen subject, p. These images, $\mathbf{x}_p^{(1)}$ and $\mathbf{x}_p^{(2)}$, we will refer to as *anchor points*. We then computed the latent representations $\mathbf{z}^{(i)} = E\left(\mathbf{x}_p^{(i)}\right)$ (for $i \in \{1,2\}$) of each anchor point and interpolated the space between them by defining the parametric path

$$\mathbf{z}(\lambda) = (1-\lambda)\,\mathbf{z}^{(1)} + \lambda \mathbf{z}^{(2)} \tag{2}$$

for $\lambda \in [0,1]$. We then took nine equally spaced values of λ, evaluated $z(\lambda)$, and decoded the resulting images using decoders corresponding to five separate identities to examine their realizations in pixel space.

The results of this experiment are shown in Fig. 15. The left- and right-most images are the anchor points. In the first row we show the reconstruction using the decoder D_p, corresponding to the person present in the anchor images. In the subsequent rows we decode the latent vectors with decoders D_q, $q \neq p$. We can see that for all of the identitities the transition of the facial expression between the anchor points is smooth and encodes intermediate facial behavior consistent across identities.

Out-of-sample generalization

In a second experiment, we selected two anchor points $\mathbf{x}_{p'}^{(i)}$, $i \in \{1,2\}$ of identity p', a subject that the model did *not* see during

Lvl	**Encoder**	Activation	Output shape	Params
8	Input Image	-	$3 \times 1024 \times 1024$	-
	Conv 1×1	LeakyReLU	$16 \times 1024 \times 1024$	64
	Conv 3×3	LeakyReLU	$16 \times 1024 \times 1024$	2.3k
	Conv 3×3	LeakyReLU	$32 \times 1024 \times 1024$	4.6k
	Downsample	-	$32 \times 512 \times 512$	-
7	Conv 3×3	LeakyReLU	$32 \times 512 \times 512$	9.2k
	Conv 3×3	LeakyReLU	$64 \times 512 \times 512$	18k
	Downsample	-	$64 \times 256 \times 256$	-
6	Conv 3×3	LeakyReLU	$32 \times 256 \times 256$	37k
	Conv 3×3	LeakyReLU	$128 \times 256 \times 256$	74k
	Downsample	-	$128 \times 128 \times 128$	-
5	Conv 3×3	LeakyReLU	$128 \times 128 \times 128$	148k
	Conv 3×3	LeakyReLU	$256 \times 128 \times 128$	295k
	Downsample	-	$256 \times 64 \times 64$	-
4	Conv 3×3	LeakyReLU	$256 \times 64 \times 64$	590k
	Conv 3×3	LeakyReLU	$512 \times 64 \times 64$	1.2M
	Downsample	-	$512 \times 32 \times 32$	-
3	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
	Downsample	-	$512 \times 16 \times 16$	-
2	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
	Downsample	-	$512 \times 8 \times 8$	-
1	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
	Downsample	-	$512 \times 4 \times 4$	-
0	Conv 3×3	LeakyReLU	$512 \times 4 \times 4$	2.4M
	Conv 4×4	LeakyReLU	$512 \times 1 \times 1$	4.M
	Latent vector	-	$512 \times 1 \times 1$	513
				23.1M

Lvl	**Decoder**	Activation	Output shape	Params
0	Latent vector	-	$512 \times 1 \times 1$	-
	Conv 4×4	LeakyReLU	$512 \times 4 \times 4$	4.2M
	Conv 3×3	LeakyReLU	$512 \times 4 \times 4$	2.4M
1	Upsample	-	$512 \times 8 \times 8$	-
	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
2	Upsample	-	$512 \times 16 \times 16$	-
	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
3	Upsample	-	$512 \times 32 \times 32$	-
	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
4	Upsample	-	$512 \times 64 \times 64$	-
	Conv 3×3	LeakyReLU	$256 \times 64 \times 64$	1.2M
	Conv 3×3	LeakyReLU	$256 \times 64 \times 64$	590k
5	Upsample	-	$256 \times 128 \times 128$	-
	Conv 3×3	LeakyReLU	$128 \times 128 \times 128$	295k
	Conv 3×3	LeakyReLU	$128 \times 128 \times 128$	148k
6	Upsample	-	$128 \times 256 \times 256$	-
	Conv 3×3	LeakyReLU	$64 \times 256 \times 256$	74k
	Conv 3×3	LeakyReLU	$64 \times 256 \times 256$	37k
7	Upsample	-	$64 \times 512 \times 512$	-
	Conv 3×3	LeakyReLU	$32 \times 512 \times 512$	18k
	Conv 3×3	LeakyReLU	$32 \times 512 \times 512$	9.2k
8	Upsample	-	$32 \times 1024 \times 1024$	-
	Conv 3×3	LeakyReLU	$16 \times 1024 \times 1024$	4.6k
	Conv 3×3	LeakyReLU	$16 \times 1024 \times 1024$	2.3k
	Conv 1×1	sigmoid	$3 \times 1024 \times 1024$	51
				23.1M

Table 1: *Detailed description of our encoder (left) and decoder (right). For the Leaky rectified unit (LeakyReLU) we use* $\alpha = 0.2$.

training. We again computed the latent vectors of these anchor points, took equidistant points on the parametric paths between them and decoded the points with the same decoders D_q as in Fig. 15.

The result of this experiment is shown in Fig. 16. When comparing the anchor points $\mathbf{x}_{p'}^{(i)}$ with the decoded images, it is clear that the encoder is capable of representing the facial behavior of an out-of-sample identity p'. Further, the transition between the anchor points is smooth and contains only valid intermediate facial expressions. Our results suggest that our latent representations are both well structured and essentially identity-free.

Blending algorithm

Pseudocode for our multi-band blending approach, adapted and modified from Burt et al. [BA83], is presented in Algorithm 1.

Figure 12: *Output of the network with various number of shared decoder levels. The target image is presented on the left side. Images from left to right correspond to the output of the network with the split placed after: latent vector, level 0, level 1, level 2 (which is our choice) and level 3.*

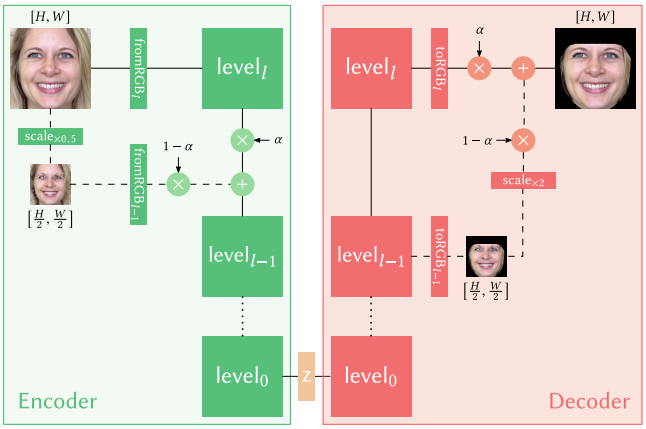

Figure 13: *Encoder-decoder network architecture illustrating the progressive approach. After adding a new level the gain parameter* $\alpha \in [0,1]$ *acts as a fader switch that gradually blends the activations of the new network with those of the trained, smaller network.*

Algorithm 1: Blending source image into target image

Input: *source* image S and *target* image T of equal sizes, set of outer facial landmarks $\mathbf{L}$ in image T, standard deviation σ, generated face image resolution r (in our case 1024)

Output: blended *output* image O

$n = \log_2 r$;

Decompose *source* image S and *target* image T into corresponding Laplacian pyramids $\mathbf{P}(S)_i$ and $\mathbf{P}(T)_i$, where i is a pyramid level, $i \in \langle 1, n \rangle$;

Initialize output pyramid $\mathbf{P}(O)$ for output image O of the same sizes as $\mathbf{P}(T)$ and fill its values with zeroes;

for $i = 1$ *to* n **do**

Compute background mask $\hat{M}_i$ defined as an image of the same size as $\mathbf{P}(T)_i$, where all pixels in the interior of the polygon formed by $\mathbf{L}$ are equal to 0 and 1 otherwise;

$\hat{M}_i = G(\hat{M}_i, \sigma)$, where $G(\hat{M}_i, \sigma)$ denotes gaussian smoothing of $\hat{M}_i$ with standard deviation σ;

Calculate face mask: $M_i = \mathbf{1} - \hat{M}_i$;

Copy background from the ***target*** image to the ***output*** image: $\mathbf{P}(O)_i = \mathbf{P}(O)_i + \hat{M}_i \mathbf{P}(T)_i$;

if $i \leqslant 2$ **then**

Copy face from the ***target*** image to the ***output*** image: $\mathbf{P}(O)_i = \mathbf{P}(O)_i + M_i \mathbf{P}(T)_i$;

else

Copy face from the ***source*** image to the ***output*** image: $\mathbf{P}(O)_i = \mathbf{P}(O)_i + M_i \mathbf{P}(S)_i$;

end

Reconstruct and return *output* image O from $\mathbf{P}(O)$;

end

Figure 14: *Visualization of swaps using the "comb" network output (ours) compared with nearest neighbors (n.n.) from the data set. Note that there are no exact correspondences between n.n. and network outputs, which suggests that the network is able to generate previously unseen intermediate states. Nearest neighbors are computed by L^2 similarity to the network output in in the pixel space of the face region.*

Figure 15: *Visualization of a segment of the data manifold learned by the common encoder. We show that the facial behavior of the anchor points ($\mathbf{x}_p^{(1)}$ and $\mathbf{x}_p^{(2)}$) can be encoded and transferred to different identities. Here λ corresponds to the mixing ratio between the anchor points' latent representations.*

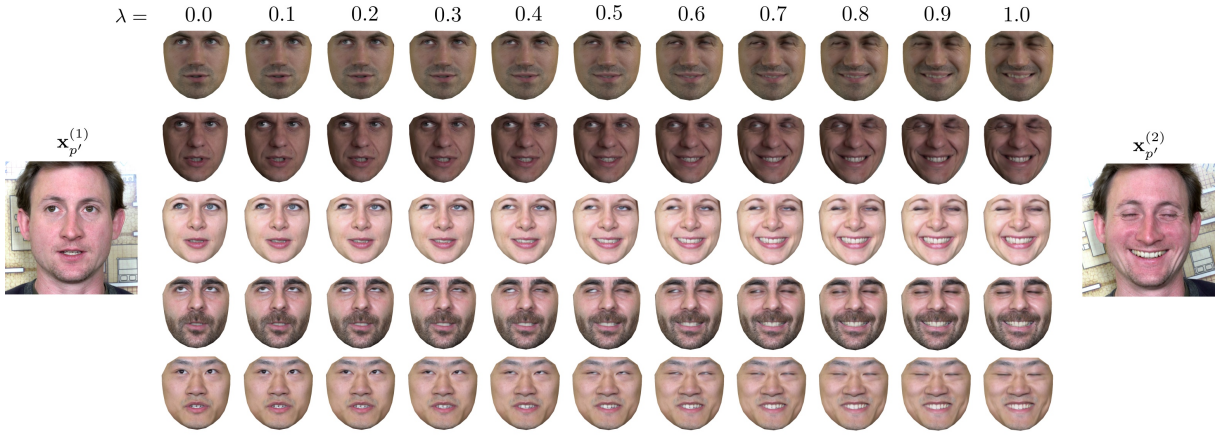

Figure 16: *Visualization of the manifold path traversed for an out-of-sample identity. We show that the facial behavior of the anchor points ($\mathbf{x}_{p'}^{(1)}$ and $\mathbf{x}_{p'}^{(2)}$) can be encoded and transferred to different identities. Note that the input face was not presented to the network during the training.*