Dynamic Neural Face Morphing for Visual Effects

Lucio Moser
lmoser@d2.com
Digital Domain
Canada

Jason Selfe
jselfe@d2.com
Digital Domain
USA

Darren Hendler
darren@d2.com
Digital Domain
USA

Doug Roble
doug@d2.com
Digital Domain
USA

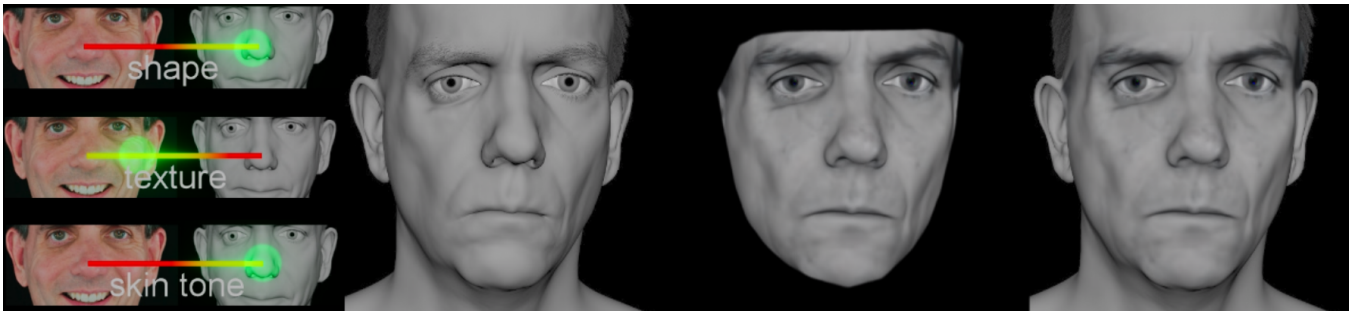

Figure 1: Interface example and results of morphing between CG character and real face. Left to right: (1) Visual interface with blending parameters for shape (100% CG), texture (72% real) and skin tone (100% CG); (2) Input image; (3) output masked image; (4) output image over the input image.

ABSTRACT

In this work we present a machine learning approach for face morphing in videos, between two or more identities. We devise an autoencoder architecture with distinct decoders for each identity, but with an underlying learnable linear basis for their weights. Each decoder has a learnable parameter that defines the interpolating weights (or *ID weights*) for the basis which can successfully decode its identity. During inference, the *ID weights* can be interpolated to produce a range of morphing identities. Our method produces temporally consistent results and allows blending different aspects of the identities by exposing the blending weights for each layer of the decoder network. We deploy our trained models to image compositors as 2D nodes with independent controls for the blending weights. Our approach has been successfully used in production, for the aging of David Beckham in the *Malaria Must Die* campaign.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; **Image manipulation**.

SA '21 Technical Communications, December 14–17, 2021, Tokyo, Japan

ACM ISBN 978-1-4503-9073-6/21/12…$15.00
https://doi.org/10.1145/3478512.3488596

KEYWORDS

face editing, face swapping, unsupervised learning, neural networks

ACM Reference Format:
Lucio Moser, Jason Selfe, Darren Hendler, and Doug Roble. 2021. Dynamic Neural Face Morphing for Visual Effects. In *SIGGRAPH Asia 2021 Technical Communications (SA '21 Technical Communications), December 14–17, 2021, Tokyo, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3478512.3488596

1 INTRODUCTION

Face morphing is used in Visual Effects when the identity of the actor has to gradually transition into another identity, be it from a real person or from a digital character. Another common use for face morphing, is when the actor's face is replaced by a non-existent identity that carries the traces of the actor, so that he/she is recognizable, but with one or more aspects changed, for example, gender, age, or changes in ethnicity.

Face morphing is equivalent to the problem of face swapping, and to the best of our knowledge, it hasn't been addressed before by a machine learning approach. Its requirements are the same: to have temporally consistent results and retain the head orientation, facial expressions, eye gaze and illumination from the actor's footage, independently of which identity he/she is changing to.

The core ideas of our work have been developed during the project *Malaria Must Die* [mal 2020] where both use cases for face morphing already described were explored. In that project, David Beckham morphs from his old self in the midst of his speech, while the moving camera rotates around him. Both David and a casted

older actor were recorded doing the same performance, over the same camera angles in a green screen recording set. The standard neural face swapping method could line up the actor to David's face, but the actor's physiognomy was not close enough to David's likeness, and the smooth transition between the two would still need to be fixed using traditional compositing methods.

After some experimentation with the autoencoders architecture used in face swapping [Perov et al. 2019], we obtained a successful ad hoc approach for morphing the two identities. We first train a regular neural face swapping model for both identities (A and B) until they reach desirable levels of quality. Then we train a second model with the opposite goal (B and A) refining the decoder weights learned in the first model while keeping the shared encoder weights constant. The second model produces decoders similar enough to the first model such that their learned weights can be interpolated to generate new decoders that render images of blended identities. The interpolation ratio dictates how closely the results look to either one of the identities. Further to that, we found that it is possible to apply different blending weights on each layer of the decoder and that the initial layers tend to determine the shape of the face and the remaining layers tend to determine the texture and skin tone.

The specific 2-identity solution just described inspired the search for a scalable N-identity solution. The face swapping architecture used can be extended for the case of N-identities by simply instantiating and training more autoencoders, but the original training procedure devised does not scale easily. We hypothesized that the solution space could be defined by a linear basis that spans the space of the weights for all decoders in a many-to-many face swapping architecture. In this scenario, each decoder has to only learn a small set of weights (the *ID weights*), which define all their layers through the shared basis. The underlying basis would force the identities of each decoder to share common representations and be close in the weight manifold, such that their *ID weights* can be interpolated to derive sensible intermediate morphing results. Our hypothesis was proved to be true with a simple regularization term on the shared basis parameters and the restriction on the number of components in the basis to be smaller than the number of identities.

The trained models are deployed to image compositors as 2D nodes, with sliders defining the blending ratios between the trained identities as well as the ability to group the decoder layers into logical layers (shape, texture and skin tone) for independent blending ratios and more artistic control.

We show results of the morphing for the two use cases mentioned: dynamic morphing between two known identities, and generation of new identities, by blending different properties of known identities.

In summary our contributions are:

- To our knowledge this is the first time an unsupervised neural face morphing approach with temporally consistent results is proposed.
- To our knowledge this is the first time that a linear weight basis is used to scale the typical face swapping architecture for a large number of identities.
- We show how the technique can be successfully used in production use cases to simplify the traditional approach of face morphing.

2 RELATED WORK

Neural face morphing has been explored in the context of *semantic face editing* primarily by manipulating the latent encodings of high-resolution pre-trained models such as StyleGan2 [Karras et al. 2019] from single images [Gabbay and Hoshen 2021; Shen et al. 2020; Zhu et al. 2020]. There are a number of limitations on those approaches: firstly, the projection to the latent representations, be it by optimization or by encoder-based inversion methods [Alaluf et al. 2021] does not guarantee precise reconstruction of the identities. Secondly, disentangling identity from all the other factors such as head pose, eye gaze, expression and lighting is still an active area of research. Thirdly, the projection is not a fast process and methods for optimizing the projection find a trade off between speed and quality [Lin et al. 2021] which also is not desirable. Lastly, to the best of our knowledge, there has been no demonstration of these approaches showing temporally consistent results.

In Pinkney and Adler [2020], two StyleGAN models are trained in different domains (cartoons and real faces) and blended to generate new models by using different sets of blending weights per layer to art-direct the resulting identities, very similarly to what we obtained in our 2-identity experiment, although with a different architecture. They use that technique in single images only and they haven't explored blending more than two models.

In Yao et al. [2020], a model is specifically trained to morph faces by only changing the age of the input face. It uses an age classifier to train a single fully connected network that modulates the latent features in the image autoencoder. Extending the strategy to other types of morphing would require different kinds of specific classifiers. The work is also based on single images and does not address temporal consistency.

At the architecture level, our weight linear basis approach can be seen as an example of multiplicative interactions [Jayakumar et al. 2020], where the learned *ID weights* give the contextual information for the conditional computation of each decoder. More specifically, our shared basis network could be considered a type of hypernetwork [Ha et al. 2017] that determines the weights of the main network (the decoders). In the original work, they use the same hypernetwork with different learned layer embeddings to generate the weights of each convolutional layer of a single network. In our approach, we use the same hypernetwork to generate the weights for the same layer on different decoder networks.

3 METHOD

Our generalized N-identity face morphing uses two training steps, in order to minimize the introduction of identity specific information in the learned shared representation.

The first step involves training a many-to-many neural face swapping model, following exactly the autoencoders with shared encoder architecture and losses as defined in Perov et al. [2019] for the "SAEHD-DF" model, with the difference that we instantiate as many autoencoders as identities. To support multiple identities, we separate the data into batches of a single identity and evaluate the loss only for the respective autoencoder. The gradients are aggregated over consecutive batches until all identities have been evaluated.

The training ends once the first model shows sufficient quality in the decoded images and facial expression correspondences between all identities.

For the second step, we replace all identity specific decoders by a single decoder with the same architecture but with each convolutional layer using weights dynamically defined by a specific hypernetwork as depicted in Figure 2. Each hypernetwork is a single fully connected linear layer network that learns the mapping from layer-specific identity weights to the parameters of the respective decoder's layer. We refer to the union of all layer-specific identity weights as *ID weights*. The hypernetworks are effectively defining a linear basis for all decoders, and the *ID weights* are the coordinates learned for each identity during training that we can interpolate at inference time. The number of components on the basis (or, equivalently, the number of input features for the hypernetworks) is defined by the user and can vary per-layer. The trainable parameters for this second model are the *ID weights* for each decoder and the parameters in the shared hypernetworks. All the shared parameters from the encoder and the first upsampling layer of the decoder from the model trained in the first step are kept constant to guarantee stable representations.

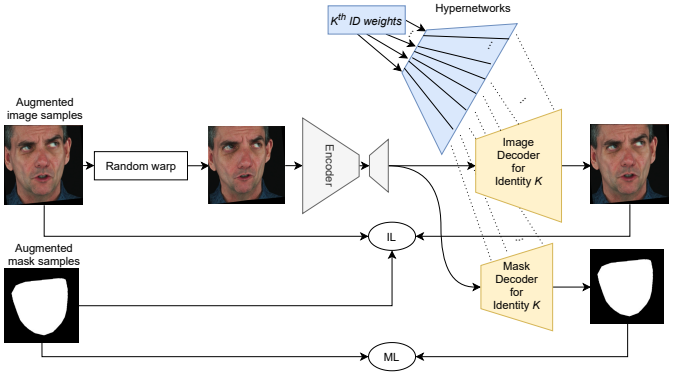

Figure 2: Structure to train neural face morphing model (fixed parameters in gray from the original shared layers of the many-to-many face swapping model, dynamically defined weights in yellow and learnable weights in blue). Showing computation for a single training batch, where only one autoencoder for identity *K* is updated. Augmented aligned face images from identity *K* are randomly warped and fed to the encoder. The *ID weights* for the *K*th identity is used as input to the hypernetworks to infer the weights for all decoder layers in the batch. Standard image reconstruction loss (IL) and mask reconstruction loss (ML) terms train the shared hypernetworks and the *ID weights* for identity *K*.

The model can be trained with standard image reconstruction losses, such as L1, structural similarity [Wang et al. 2004] and perceptual losses using pre-trained models, as well as trained with discriminators to improve image quality. Additionally we apply a regularization L1 loss over the hypernetwork parameters to encourage sparsity in the weight basis.

Once the model has been trained, the inference is done with two inputs: the aligned face image from one of the identities used in the training, and a set of *ID weights* that can either match one of the learned coordinates for any of the known identities or it can also be produced by linearly interpolating them, using layer-specific blending ratios. The *ID weights* are inputs to the hyperlayers to infer all decoder layer weights, prior to the computation of the output image.

Figure 3: Temporally consistent results in different frames (shown in each row) for different blending ratios between CG and real face (shown in each column).

In order to give control to artists, we provide a custom Nuke node [nuk 2021] where the user specifies the model, and for each trained identity, the blending weights for three high-level groupings of its decoder layers, which correspond roughly to shape, texture and skin tone. The specific layers for each grouping have been found experimentally and can also be defined by the user. Our node is embedded in simple user setups, having expressions constraining these weights such that the sum of blending weights for a given layer adds up to one across all trained identities. We show examples of simple visual interfaces which apply such techniques next.

4 RESULTS

For all experiments shown, we used an architecture specific for 512x512 input/output images. We trained the initial models (multiple decoders) using simple image reconstruction losses (L1 and structural similarity) and in the second models (single decoder) we additionally used PatchGAN discriminators [Isola et al. 2017]. No special treatment was given to areas such as the mouth or eyes.

Of note, our goal was to produce morphing results to be used by an artist for further integration to the footage, so our focus was not to match skin tones to the driving face. Techniques such as in Naruniec et al. [2020] can be applied for automatic blending.

Our first experiment is a model trained for two identities: a real person and a CG character based on that same person, which is a common use case in production, where characters do have a varying degree of similarity to the actor. Figure 1 shows the visual user interface with the slider controls for the blending on three conceptual layers roughly determining shape, texture and skin tone. The blending weights from these controllers are values between 0 and 1.

We show in Figure 3 the results of gradually morphing from the character (which is also the driving face in the input video)

Figure 4: Interface example and results on each row, blending different identities for changes in age, ethnicity and gender. Left to right: (1) User control for the shape, (2) texture and (3) skin tone attributes, followed by (4) Input image and (5) Output image composited on top of input image.

Figure 5: Temporally stable results under light variations for an application of aging a person by blending *ID weights* with older identities. Different frames displayed in the columns. From top to bottom: (1) input image, (2) output masked image, (3) output image composited on top of input image.

to the person, using a single blending ratio for all decoder layers. The Figure shows how the results hold in different expressions and therefore can be applied on a continuous video.

Our second experiment explores the case of multiple identities, for change in age, ethnicity and gender. The model was trained with footage from 13 people and their shape, texture and skin tones can be combined by the artist using a similar interface. This time, the user interface shown in Figure 4 allows to select and blend 3 out of the 13 identities based on the relative distances from the controller to each one the faces on the edges of a triangle, using the barycentric coordinates as blending weights.

Since the training material for the 13 identities included variations in lighting, the trained model can also morph faces with variable lighting and produce temporally consistent results, as shown in Figure 5.

5 LIMITATIONS AND CONCLUSIONS

In this work we present the workflow for training a neural face morphing model for multiple identities and its integration to the artist tool set, with control over shape, texture and skin tone. Guaranteeing precise local control and better separation over these attributes is left as future work.

Our architecture follows the design used in common neural face swapping applications and it shares same limitations, namely, (1) having matching data distribution between the identities is important for the quality of results, (2) reconstruction of hair and fine skin details is challenging on widely varying head poses, (3) high resolution models do require significantly more training time and computational resources. Any new research results improving neural face swapping should be directly applicable to our approach as well.

Our method, irrespective of the current limitations, is being successfully used in production to simplify and augment the standard approaches used in facial morphing tasks for visual effects.

REFERENCES

2020. Malaria must die. https://malariamustdie.com/campaign
2021. *Nuke, NukeX & Nuke Studio: VFX Software.* https://www.foundry.com/products/nuke
Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. arXiv:2104.02699 [cs.CV]
Aviv Gabbay and Yedid Hoshen. 2021. Scaling-up Disentanglement for Image Translation. *arXiv preprint arXiv:2103.14017* (2021).
David Ha, Andrew Dai, and Quoc V. Le. 2017. HyperNetworks. https://openreview.net/pdf?id=rkpACe1lx
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976. https://doi.org/10.1109/CVPR.2017.632
Siddhant M. Jayakumar, Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. 2020. Multiplicative Interactions and Where to Find Them. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rylnK6VtDH
Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. Analyzing and Improving the Image Quality of StyleGAN. *CoRR* abs/1912.04958 (2019).
Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. 2021. Anycost GANs for Interactive Image Synthesis and Editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
J. Naruniec, L. Helminger, C. Schroers, and R.M. Weber. 2020. High-Resolution Neural Face Swapping for Visual Effects. *Computer Graphics Forum* 39, 4 (2020), 173–184. https://doi.org/10.1111/cgf.14062
Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. 2019. *DeepFaceLab*. https://github.com/iperov/DeepFaceLab
Justin N. M. Pinkney and Doron Adler. 2020. Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains. *CoRR* abs/2010.05334 (2020). arXiv:2010.05334 https://arxiv.org/abs/2010.05334
Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the Latent Space of GANs for Semantic Face Editing. In *CVPR*.
Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861
Xu Yao, Gilles Puy, Alasdair Newson, Yann Gousseau, and Pierre Hellier. 2020. High Resolution Face Age Editing. *CoRR* abs/2005.04410 (2020).
Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-Domain GAN Inversion for Real Image Editing. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 592–608.