


Deep Learning-Based Unsupervised Human Facial Retargeting

Seonghyeon Kim¹ , Sunjin Jung¹ , Kwanggyoon Seo¹ , Roger Blanco i Ribera² , Junyong Noh¹ 

¹KAIST, Visual Media Lab ²C-JeS Gulliver Studios



Figure 1: Our method can successfully transfer the blendshape weights of a source face model on the far left to target models from the second to the last of different proportions, topology, and blendshape configurations.

Abstract

Traditional approaches to retarget existing facial blendshape animations to other characters rely heavily on manually paired data including corresponding anchors, expressions, or semantic parametrizations to preserve the characteristics of the original performance. In this paper, inspired by recent developments in face swapping and reenactment, we propose a novel unsupervised learning method that reformulates the retargeting of 3D facial blendshape-based animations in the image domain. The expressions of a source model is transferred to a target model via the rendered images of the source animation. For this purpose, a reenactment network is trained with the rendered images of various expressions created by the source and target models in a shared latent space. The use of shared latent space enable an automatic cross-mapping obviating the need for manual pairing. Next, a blendshape prediction network is used to extract the blendshape weights from the translated image to complete the retargeting of the animation onto a 3D target model. Our method allows for fully unsupervised retargeting of facial expressions between models of different configurations, and once trained, is suitable for automatic real-time applications.

CCS Concepts

• **Computing methodologies** → **Animation**; **Computer vision**;

1. Introduction

With the growth of the movie and game industries, creating high-quality facial animation remains important. A standard approach to creating realistic 3D facial animation is to use blendshape-

based models driven by motion-captured data or laboriously crafted keyframes by skilled artists. One reason for the popularity of the blendshape approach is the possibility to create semantically equivalent blendshape configurations for diverse characters of varying

facial proportions. This parallel parametrization allows users to easily drive or transfer the animation of various characters. However, even when following a standardized guide such as the Facial Action Coding System (FACS) [FE78], obtaining fully semantically matching blendshapes remains challenging [SL14], and thus a supervised tuning process is mandatory when aiming for high-quality facial animation [SML16]. Moreover, it is also common that different models will have different blendshape configurations defining different expressions spaces.

Recent facial expression retargeting techniques aim to address these issues and focus on preserving the semantic meaning of the original source expressions. A common approach to retargeting is to define a cross-mapping between expression spaces using a set of semantically matching expressions of the source and target models [SCSN11, DCFN06]. Other approaches aim at directly building semantically equivalent sets of target models [NN01, SLS*12, RZL*17, SP04]. In this way, the facial animation of a source model can be directly used on the generated target blendshapes. One drawback of these methods is that they require manual selection of correspondence data or paired parameters between the source and target models. To obviate the need for manual specification of correspondences, recent studies have proposed the use of autoencoders for deformation transfer [GYQ*18, ZCZ20]. In this case, however, the retargeting depends on the coarse-level features, which are unable to capture the subtle details of a facial expression [GYQ*18] or requires manually measured scores to be trained [ZCZ20].

Face manipulation and expression prediction are related to our work. In the field of face manipulation, applications such as face reenactment and swapping have gained much attention these days. There have been several studies focused on face reenactment [TZN*15, TZS*16, KGT*18, SLT*19, SWR*21] or face swapping [TVRF*20, TVRF*20, PGC*20, NHSW20, BCW*18, NYM18a, NYM18b, NKH19]. These methods achieve a quality of synthesized facial images that is almost indistinguishable from real images, even to human eyes. Meanwhile, several studies have been conducted on the topic of expression prediction [CWLZ13, CHZ14, LKA*17, TZB*18, TBG*19, TLL19, GZY*20]. These methods successfully predict the expression parameters of the face images.

Inspired by the recent success of these face manipulation and expression prediction approaches, we formulate blendshape retargeting as an image-based face reenactment problem by rendering the 3D source and target models to images in an unsupervised manner. In our retargeting framework, a reenactment network transfers the expression of the source model image to a target model image. Then, given the reenacted image, an expression prediction network predicts the blendshape weights of the target model. Our approach enables automatic retargeting of facial animations without the tedious process of pairing facial expression data. In addition, the retargeting pipeline runs in real-time on a consumer level GPU. Our retargeting framework is not limited to realistic human characters but can also handle stylized human characters, as shown in Figure 1. We compare our method to previous retargeting methods [SCSN11, RZL*17] and show that the results from our method are in quality visually similar or even superior to those from previous methods.

2. Related Work

2.1. Facial Retargeting

Facial retargeting is a process to transfer facial animation from a source model to a target model while preserving the semantic meaning of the facial expressions. Deng et al. [DCFN06] introduced a semi-automatic technique to animate a face by mapping the parameters for the motion captured data to blendshape weights based on the Radial Basis Function (RBF). Song et al. [SCSN11] suggested a retargeting method that preserves the style of the animation using RBF and kernel canonical correlation analysis based regression. Seol et al. [SLS*12] improved the smoothness and naturalness of a retargeted animation by considering the velocity of the points on the source and target face. Ribera et al. [RZL*17] further enhanced the quality of the retargeting by learning the manifold of source and target expression spaces to create actor-specific blendshapes and thereby accurately retarget the performance of an actor to a target model. These methods require either a training set of paired blendshape expressions [DCFN06, SCSN11] or manually selected corresponding vertices between source and target models [SLS*12, RZL*17]. Different from these previous methods, we solve the retargeting problem in an unpaired manner, in which the user does not need to manually match blendshape expressions or construct matching points between models.

Facial animation can be easily transferred with a parallel parametrization of the facial rig sets. Therefore, many studies have focused on building two semantically equivalent sets of facial rigs that can be used for retargeting. Noh and Neumann [NN01] suggested a method that clones per-vertex displacements of a source mesh to the corresponding points on a target mesh. The animation of a source mesh can be directly conveyed to a target mesh to enable semantically equivalent facial expressions. Sumner and Popović [SP04] proposed a method that can deform a target model using the deformation gradients of a source model. These methods also require manually constructed pairs of semantically equal shapes. To solve this problem, recent studies have proposed unpaired deformation transfer based on an autoencoder structure [GYQ*18, ZCZ20]. Unfortunately, they mostly rely on coarse-level features [GYQ*18], which makes it hard to capture the deformation details of 3D shapes or requires manual evaluation from the user [ZCZ20]. In contrast, our automatic method can generate comparable results preserving subtle details without relying on human evaluation.

Similar to our approach, Aneja et al. [ACF*16] trained two convolutional neural networks to learn the shared latent variables of human and character expressions. Using the trained network, the image of the facial expression of a 2D character can be found using a geometry and perceptual model mapping. This approach has been further improved to adapt it to 3D stylized character expression [ACC*18]. Using an expression recognition network, this method can generate the rig parameters that best match the human facial expression with an input facial image. These approaches also require a manually labeled dataset of facial expressions and rig parameters.

2.2. Face Reenactment and Swapping

Similar to facial retargeting, face reenactment refers to the task of transferring facial expressions to a target face; albeit, this is generally in the image-domain. Thies et al. [TZN*15, Tzs*16] used a parametric model to reenact a target actor from a source actor. Kim et al. [KGT*18] enhanced the visual quality and the range of head motion of a target actor by using convolutional neural networks as a photo-realistic rendering function. More recently, use of a neural texture further improved the visual quality of results from the previous method [TZN19]. Without any annotation, Siarohin et al. [SLT*19, SWR*21] decoupled the appearance and motion information so that each could be used for the face reenactment task.

The goal of face swapping is to replace the target face with a source face. Recently with the introduction of DeepFake [TVRF*20], face swapping has gained much attention because of the high quality results it produces. Several face swapping approaches [TVRF*20, PGC*20, NHSW20] used a shared encoder and target specific decoders. The encoder maps the source and target identities to the same latent space and the decoders translate the source latent code to target face identities. By jointly combining the encoder and decoder, transferring source facial expressions to the target face becomes possible. Other approaches use generative adversarial networks (GAN) to generate high quality images [BCW*18, NYM18a, NYM18b, NKH19]. We take advantage of these recent developments of high quality facial reenactment and swapping methods for 3D facial animation retargeting.

2.3. Expression Prediction

Expression prediction is the task of estimating facial expression parameters from a human face image. Cao et al. [CWLZ13, CHZ14] suggested facial performance capture by training a regressor that predicts blendshape parameters from a video stream with a sequence of facial images and facial landmarks. Laine et al. [LKA*17] used a convolutional neural network to predict the vertex positions of a facial mesh from the image of an actor.

Many studies have focused on using a parametric face model [BV99] to predict facial expressions. Tewari et al. [TZB*18, TBG*19] utilized a differentiable renderer that enabled unsupervised end-to-end learning of semantic facial parameters including expression and appearance. A method suggested by Tran et al. [TLL19] achieved a high level of detail of reconstructed face images by using a dual-pathway network architecture that consists of one global pathway and a local pathway with multiple sub-networks. To achieve speed and accuracy improvement, Guo et al. [GZY*20] suggested a meta-joint optimization strategy for a network that predicts a small set of 3D morphable model parameters from an image of a real human face. We train a neural network that can predict blendshape weights given a rendered face image of a virtual character.

3. Retargeting Method

We propose a retargeting approach that enables transfer of a source 3D blendshape-based animation to a target model without paired data. Our key insight is to exploit 2D information by rendering

the 3D facial animation and perform the expression transfer in the image-domain. The rendered images are input to a reenactment network, *ReenactNet* (Sec. 3.4), which reenacts the target images from the rendered source images. Then, a blendshape prediction network, *BPNet* (Sec. 3.5), predicts the blendshape weights using the generated target facial images. In the following, we will explain the blendshape formulation used by the proposed method (Sec. 3.1), the retargeting pipeline (Sec. 3.2), the training datasets for both *ReenactNet* and *BPNet* (Sec. 3.3), and the training schemes (Sec. 3.4, 3.5).

3.1. Delta Blendshape Formulation

We follow the delta blendshape formulation proposed by Lewis et al. [LAR*14], in which neutral facial expression b_0 is subtracted from blendshapes b_k to yield displacements. A new expression $V(w)$ is then obtained by applying a weighted sum of vertex displacements to the neutral expression b_0 :

$$V(w) = b_0 + \sum_{k=1}^n w_k (b_k - b_0). \quad (1)$$

$B = \{b_0, \dots, b_n\}$ is a set of blendshapes, and $w = \{w_0, \dots, w_n\}$ are the blendshape weights. In the proposed retargeting pipeline, we refer to $V_s(w_s)$ and $V_t(w_t)$ as a source and target model, respectively. Here and from this point on, subscripts s and t represent source and target, respectively.

3.2. Retargeting Pipeline

As shown in Figure 2, we render the posed expression V_s corresponding to the blendshape weights w_s with rendering parameter p_s and source texture M_s . Figure 3 shows that the encoder maps every source image I_s into the shared latent space Z . Then, the target decoder D_t receives the source latent code z as input to reenact the source facial expression into a target model image \tilde{I}_t . Given the reenacted image, *BPNet* predicts the blendshapes weights \bar{w}_t of the target model. Finally, the predicted weights are applied to the target model. This procedure is performed for every frame of the source animation.

3.3. Training Dataset

We construct a facial image dataset F consisting of images with posed expressions from the source and target models in order to train the networks. The image I_s and I_t form the source and target image spaces $S \subset F$ and $T \subset F$ are rendered using a differentiable renderer $R(\cdot)$ [RRN*20] to acquire the corresponding images. Because our purpose is to produce an intermediate image for the retargeting, instead of a realistic image, we chose to be time efficient and therefore utilized the Phong reflection model [Pho75]. The rendering process of the two models can be expressed as follows:

$$\begin{aligned} I_s &= R(V_s(w_s), M_s, p_s) \in S, \\ I_t &= R(V_t(w_t), M_t, p_t) \in T, \end{aligned}$$

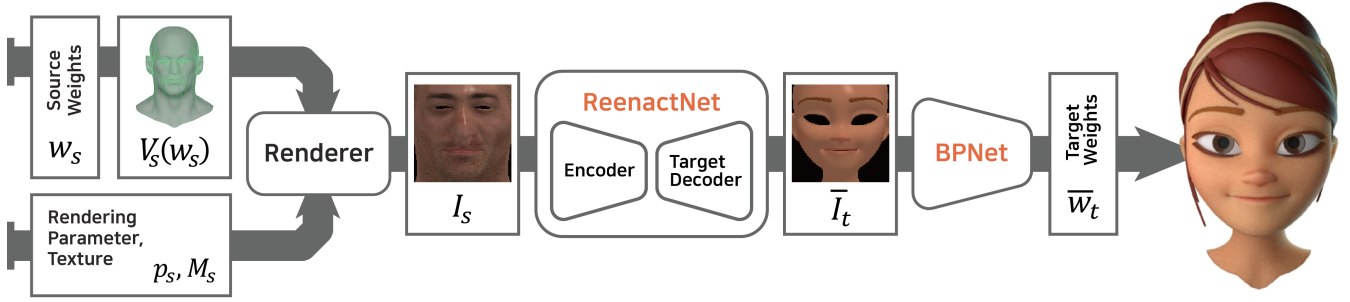


Figure 2: Overview of the proposed retargeting pipeline. Our retargeting pipeline receives model $V_S(w_S)$, a source rendering parameter p_S , and a source texture image M_S as input and renders a source facial image I_S . Target image \bar{I}_t is reenacted by ReenactNet $D_t(E(I_S))$ given I_S as input. Then, \bar{I}_t is fed to BpNet to predict blendshape weights \bar{w}_t . Through this pipeline, the blendshape weights w_S of the source model can be translated into the blendshape weights \bar{w}_t of the target model.

where $p \in \mathbb{R}^{19}$ are the rendering parameters consisting of camera transformation matrix, model position, model scale, and point light position. Note that the camera and light positions are shared between the two models. Using existing blendshape animation sequences that cover a wide range of expressions of the source and target models, we construct a paired dataset that associates the rendered image with the blendshape weights. Detailed information about the face models used in this study and dataset is described in Sec. 4.1.

3.4. Facial Reenactment

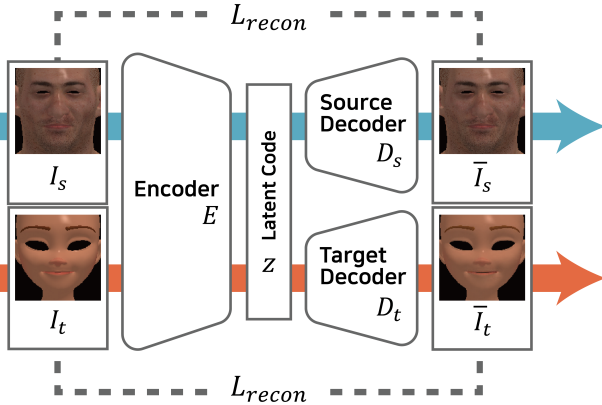


Figure 3: Training of the autoencoder. The autoencoder consists of a shared encoder E , which encodes an image into a latent code z , and two separate decoders. The decoders D_S and D_t are trained to reconstruct a source image I_S and a target image I_t , respectively, using L_{recon} . ReenactNet consists of E and D_t in the autoencoder.

ReenactNet predicts an image \bar{I}_t on the target image space T from a given image of I_S on the source image space S . To achieve this, we employ an autoencoder consisting of a shared encoder and two separate decoders. The shared encoder $E : F \rightarrow Z$ encodes input images $I_S, I_t \in F$ into a common latent space Z . A source de-

coder $D_S : Z \rightarrow S$ decodes a latent code $z \in Z$ into a predicted image $\bar{I}_S \in S$, and a target decoder $D_t : Z \rightarrow T$ decodes z into \bar{I}_t . While S and T reside in F , they are disjoint sets. Encoding the input images with a common encoder E enables the network to learn shared features such as facial expressions in Z . This property enables us to use a different decoder at inference time for facial reenactment.

As shown in Figure 3, the autoencoder is trained to reconstruct the original input image for both source and target in an unsupervised manner. The reconstruction loss L_{recon} is defined as follows:

$$L_{recon} = \|I_S - \bar{I}_S\|_1 + \|I_t - \bar{I}_t\|_1,$$

where predicted source and target images are denoted as $\bar{I}_S = D_S(E(I_S))$ and $\bar{I}_t = D_t(E(I_t))$, respectively. After the training, we combine E and D_t to construct $ReenactNet = D_t(E(\cdot))$. A reenacted target image can be acquired by $\bar{I}_t = ReenactNet(I_S)$.

3.5. Blendshape Prediction

BpNet is trained to predict w_t given I_t . To train BpNet, we use existing blendshape animations with weights w_t for the target model and the rendered images I_t , as explained in Sec. 3.3. For the loss term L_w , we use an L_1 loss on the error between the predicted \bar{w}_t and the ground truth weights w_t . L_w is defined as follows:

$$L_w = \|w_t - \bar{w}_t\|_1.$$

The prediction can be further improved by introducing a rendering loss L_r that accounts for the difference between input ground truth image I_t and \hat{I}_t , which is the image rendered with the predicted weights \bar{w}_t . L_r is defined as follows:

$$L_r = \|I_t - \hat{I}_t\|_1.$$

The image \hat{I}_t is rendered using a differentiable renderer [RRN*20] with a model $V_t(\bar{w}_t)$ and the same rendering parameters used to render the target image dataset explained in Sec. 3.3. Figure 4 illustrates an overview of the training process of BpNet. The total

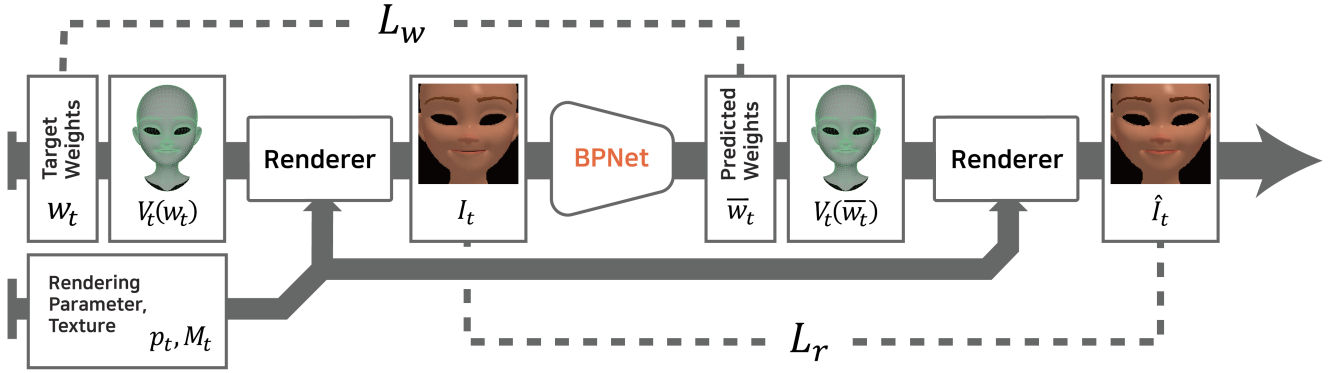


Figure 4: Training of BpNet. BpNet predicts blendshape weights \bar{w}_t from an input image I_t . The predicted weights \bar{w}_t are used to construct a target model. The model is rendered using a differentiable renderer resulting in \hat{I}_t . To train BpNet, we use two loss terms. L_w compares the weight difference using w_t and \bar{w}_t . L_r compares the pixel-level difference using I_t and \hat{I}_t .

loss of BpNet is defined as follows:

$$L_b = \lambda_w L_w + \lambda_r L_r,$$

where λ_w and λ_r are the weights for L_w and L_r , respectively.

After the training is completed, BpNet can predict blendshape weights from a target model image. Because reenacted image \bar{I}_t and I_t lie in the same target model image space T , BpNet can predict \bar{w}_t from \bar{I}_t . Therefore, using ReenactNet and BpNet, we can acquire the corresponding \bar{w}_t given w_s . Finally, applying \bar{w}_t to the target model produces the retargeted facial expression.

4. Experiments







In this section, we first describe the implementation details of our training settings. Then, we compare the visual results from our method to those from existing methods. The results can also be found in the accompanying video. Next, we perform an ablation study to validate the effectiveness of our training scheme. Finally, we evaluate the capacity of ReenactNet.

4.1. Implementation Details

For the experiments, we used six different 3D face models: Mery (©meryproject.com), Victor (©Faceware Technologies, Inc.), Polywink (©Polywink), Man A, Man B, and Man C. Numbers of blendshapes, training frames, and vertices are summarized in Table 1. We also prepared 3300 frames of blendshape weights for the source model as a validation set. These weights were never shown to the networks in the training process. We rendered facial images of the source and target models with the corresponding blendshape weights. The image resolution was $128 \times 128 \times 3$. The average rendering time per image was approximately 10ms. These images were used to train ReenactNet and BpNet.

We used the Adam optimizer [KB14] with a learning rate of 0.0003 to train ReenactNet for 16 epochs with a batch size of 16. BpNet was trained with pairs of target facial images and their corresponding blendshape weights. We used the Adam optimizer [KB14] with a learning rate 0.0003 to train BpNet for 8 epochs with

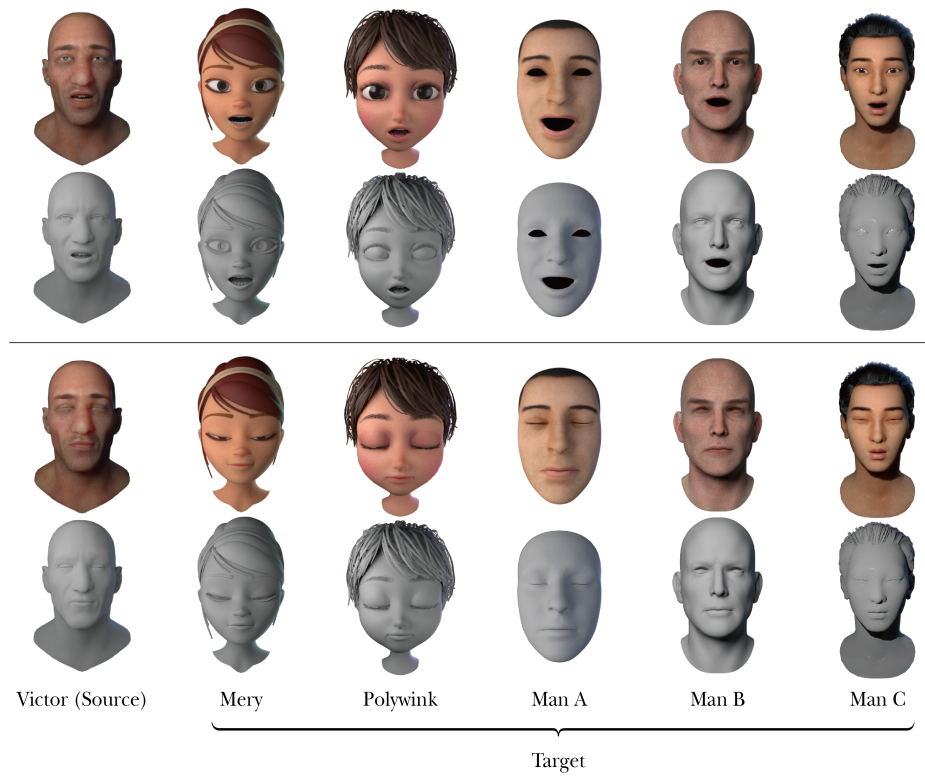
Table 1: The number of blendshapes, training frames, and vertices for each model. Mery, Polywink, Man A, and Man B share semantically identical blendshapes. We train each model with a large set of facial poses that cover a wide range of expressions such as squinting, smiling, grimacing, speech, etc.

		Blendshapes	Training Frames	Vertices
Victor		45	18611	20104
Mery		51	15502	5065
Polywink		51	15502	4689
Man A		51	15502	1220
Man B		51	18067	10892
Man C		29	15118	1934

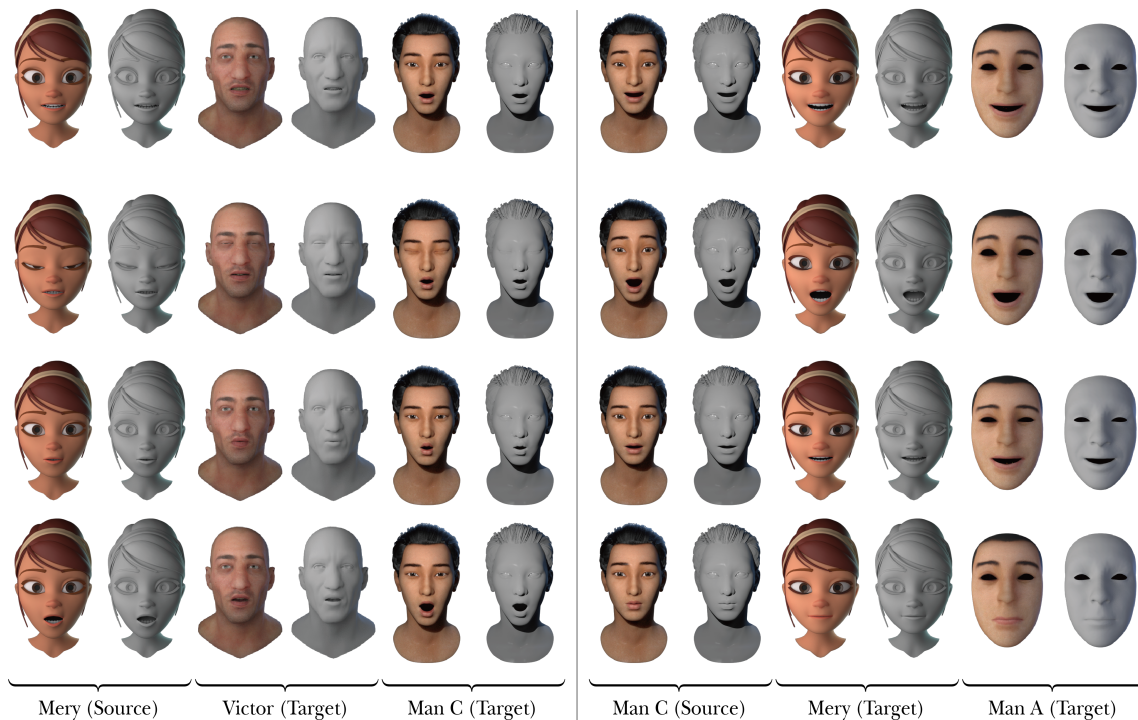
a batch size of 5. We set both λ_w and λ_r equal to one. Both networks were trained and evaluated on a consumer-grade GPU (NVIDIA RTX 2080 Ti). The training of ReenactNet and BpNet required approximately 40 minutes and 26 minutes, respectively. The inference times of ReenactNet and BpNet were approximately 14 ms and 0.6 ms, respectively. For architecture details of ReenactNet and BpNet, please refer to the supplementary material.

4.2. Results of Retargeting Pipeline

To show the capability of our retargeting pipeline, we set Victor as the source model and the remaining five models as target models. As shown in Figure 5a, our method successfully transferred the expression of the source model to target models that have differing number of vertices, style, and gender. Figure 5b shows that our



(a) Results of retargeting to various models with different shape, gender, and style.



(b) Results of our method using different models as source and target.

Figure 5: Retargeting results of our method. The expressions of the source model are reproduced well on the target models as shown in both (a) and (b). For better evaluation, we provide the results with and without texture.

method can handle retargeting with different models as source and target. More results can be found in the supplementary material.

We evaluate the robustness of our method using a cyclic consistency metric. For this, we try to recover the source model from a retargeted target model. We first retargeted the 3976 frames of animation of the source model to the rest of the models. Then, using the retargeted results as input, we recovered the original expressions and evaluated the error. In this experiment, we used Victor as the source model and measured the distance between the rendered images of the original animation and the rendered images of the recovered animation. Figure 6 shows the close visual similarity between the source and recovered images, regardless of the target models used for the retargeting. Small pixel errors are observed only in limited areas especially near openings of mouth and eyes. Table 2 reports quantitative errors from the cyclic retargeting using two metrics, Mean Absolute Error (MAE) and Structural Similarity Index Measure (SSIM). The computed values are similar to the ideal values indicating that the original animation and the recovered animation have few differences.

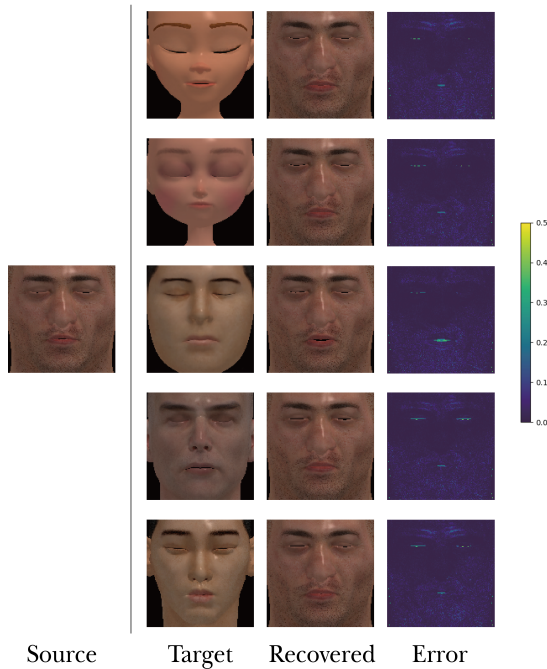


Figure 6: Cyclic retargeting of source animation. A source expression is retargeted to different models. We then retarget the resulting expression back to the source model and measure the per-pixel Euclidean distance in RGB (color channels in $[0, 1]$) between the source and the recovered expressions.

4.3. Comparison

4.3.1. Retargeting Methods

In this section, we compare our method with two existing retargeting methods: cross-mapping (CM) [SCSN11] and manifold alignment (MA) [RZL*17]. We set Victor as the source model for this

Table 2: Evaluation results from cyclic retargeting. The values represent average errors between 3976 rendered images of source animation and corresponding rendered images of recovered animation. When images are equal, MAE is 0.000 and SSIM is 1.000.

Model	MAE↓	SSIM↑
Victor → Mery → Victor	0.016	0.895
Victor → Polywink → Victor	0.016	0.892
Victor → Man A → Victor	0.016	0.893
Victor → Man B → Victor	0.016	0.894
Victor → Man C → Victor	0.017	0.886
Ideal	0.000	1.000

experiment. Mery and Man C were used as target models. For CM, we trained an RBF-based regressor that maps the source blendshape weights to the target blendshape weights using a manually paired training dataset. The size of the paired datasets was set to 31 and 22 for Mery and Man C, respectively. For MA, we built the blendshapes of the target faces using RBF with manually annotated corresponding points between the source and target neutral models. As shown in Figure 7, our method achieves comparable results as can be verified by Man C model. In case of the stylized character Mery, the results from MA tend to show exaggerated mouth expressions while our method tries to preserve the meaning of the original expression. In addition, as shown in the second and third rows, MA occasionally fails to transfer the expressions associated with the eyes. Also, our method outperforms CM in the mouth region as can be clearly seen from Mery and Man C. Due to the utilization of visual information in the form of a rendered image, our method can transfer the expression of the source model to each target with subtle details, as can be observed near the mouth or eye area. More comparison results can be found in the accompanying video.

4.3.2. ReenactNet

We compared *ReenactNet* to UNIT [LBK17], an unsupervised image-to-image translation method. We trained UNIT and *ReenactNet* with 14,532 facial images of Victor as the source and 16,050 facial images of Man B as the target. The image resolution was $128 \times 128 \times 3$. We set all hyperparameters according to the original setting of UNIT. The facial images of Victor translated to the images of Man B are shown in Figure 8. Unlike UNIT which focuses on generating realistic target facial images, our network focuses on precisely reproducing the expression of source images on target images, as shown in the red boxes.

4.4. Ablation Study

We conducted an ablation study to analyze the effectiveness of the loss design for *BPNet*. For the study, *BPNet* was trained with three different settings: only L_w , only L_r , and $L_w + L_r$. To evaluate the quality of the predicted blendshape weights, we rendered images of the Man B model using the predicted weights and measured the similarity of the rendered images to the ground truth images using four image quality metrics: Peak Signal-to-Noise Ratio (PSNR), SSIM, Learned Perceptual Image Patch Similarity (LPIPS) [ZIE*18] with AlexNet [KSH17] and VGG [SZ14]

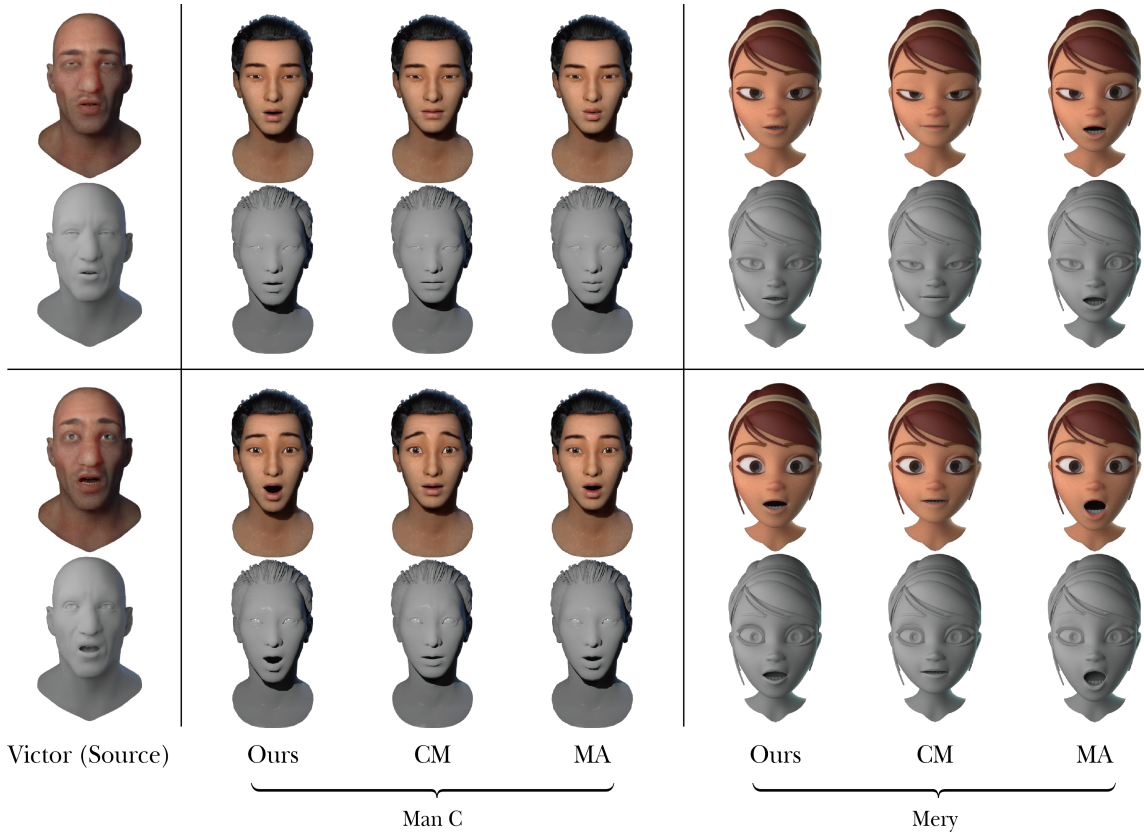


Figure 7: Comparison of retargeting results produced by our method (Ours), cross-mapping (CM), and manifold alignment (MA). In all cases, our method generates superior or comparable results to those of the other methods.

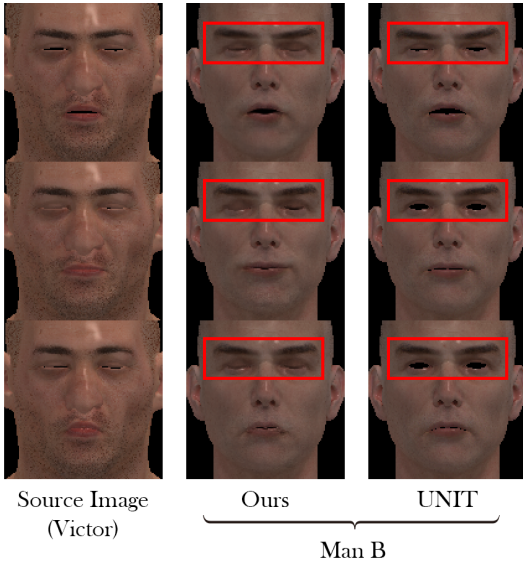


Figure 8: Comparison of results from ReenactNet with those from UNIT.

Table 3 shows the results of the quantitative evaluation of the three different settings. When only using L_r , *BPNet* was trained in an unsupervised manner. Without explicit supervision of the blendshape weights, the predicted results were not as good as the others. Using L_w , *BPNet* was trained with supervision of the blendshape weights, resulting in better metric values than when using L_r only. Using both loss terms L_w and L_r leads to better quantitative results than using the other settings do. Figure 9 shows that we obtained the best result when both terms were used.

Table 3: Quantitative results from loss ablation test. The best result in each metric is in bold.

Model	PSNR \uparrow	SSIM \uparrow	LPIPS (Alex) \downarrow	LPIPS (VGG) \downarrow
only L_r	34.409	0.966	0.03200	0.0294
only L_w	37.661	0.973	0.00964	0.0163
$L_w + L_r$	38.650	0.977	0.00795	0.0144

5. Discussion

Although the proposed method can successfully retarget a source expression to a target model, the method has some limitations. The method mainly focuses on human characters. While we demonstrate the flexibility of our method by experimenting with varying

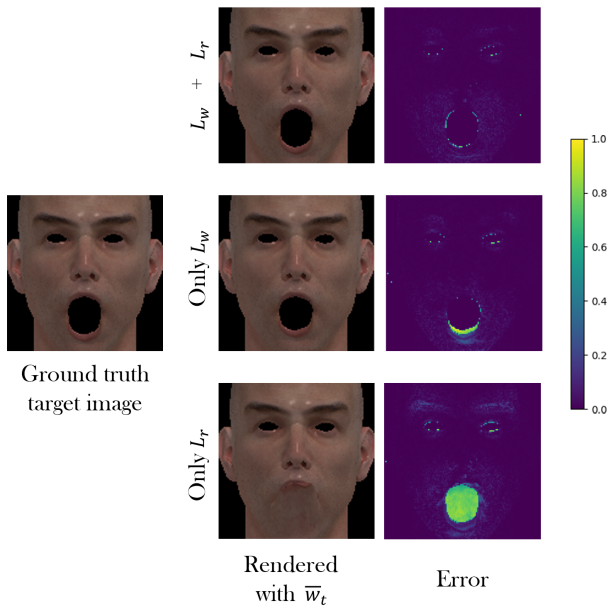


Figure 9: Influence of the adaptation of the rendering loss L_r to the rendered image. The left image is the ground truth target image I_t . The center column is the result rendered with the predicted weights \bar{w}_t using *BPNet*. Three images from the top of the center column are rendered with the result of *BPNet*, which was trained with L_w and L_r , L_w , and L_r , respectively. The right column shows the errors between the ground truth I_t and the rendered result \hat{I}_t .

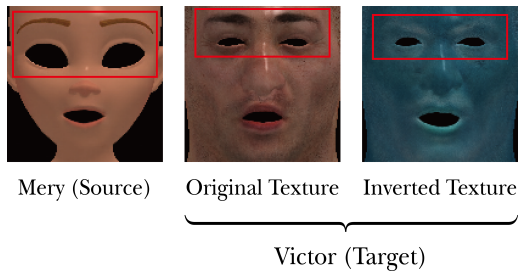


Figure 10: Reenactment results between source model and two target models with significantly different textures. Observe that the eyebrows of the character retargeted using the inverted texture went down, failing to reproduce the source expression correctly. While our method can handle texture variation to a certain degree, significant difference in texture between source and target models can result in semantically different expressions.

degrees of stylization and differing facial proportions, *ReenactNet* may fail to generate facial images correctly in the extreme cases where there is a significant difference in shape or texture between the source and target models as shown in Figure 10.

A key element in the proposed method is to perform the expression translation in the 2D image domain. However, certain similar expressions such as lip rolling, kissing motion or puck motions have subtle differences in the way the lips roll outwards or inwards.

While our tests indicate the solidity of our image domain approach we plan to study reincorporating additional 2D information such as normal maps or vector displacement maps, to partially reincorporate 3D information in order to improve the retargeting of challenging subtle expressions.

It should be noted that our method requires existing animation data both for the source and target models. However, our method does not require high quality animations and any animation should serve the purpose as long as it can cover a wide expression space of the models. Because the blendshape model itself is the generative basis of the model's expression space, one could consider preparing for a training dataset by randomly sampling from the expression space. In this case, as not all weights combinations produce valid faces, certain care is needed to ensure that a valid face is sampled. For instance, we can test local smoothness of the sampled expressions [RZL*17].

The proposed method does not consider temporal smoothness explicitly because we did not observe noticeable visual artifacts without it in the current training setting. One way to incorporate temporal smoothness would be to consider the approach proposed in Seol et al. [SLS*12].

6. Conclusion

We propose a retargeting method that transfers the blendshape weights of a source model to a target model without paired training data or specification of corresponding vertices. Our retargeting method consists of *ReenactNet* and *BPNet*. In the training stage, *ReenactNet* is trained using rendered facial images of the source and target models in an unsupervised manner. *BPNet* is trained with images of the target model and paired weights. In the retargeting stage, *ReenactNet* generates reenacted images of the target model from the rendered images of the source model using input blendshape weights. *BPNet* receives the generated target images as input and predicts the blendshape weights of the target images. We showed that the proposed retargeting method can handle stylized characters as well as human characters. The quality of produced results is comparable to or better than the results of previous retargeting methods [SCSN11, RZL*17]. For future work, we aim to expand our method to handle a wider range of models, including non-human characters with largely different facial features; we also aim to generalize the method to other types of facial rig parametrizations.

Acknowledgement

We thank the anonymous reviewers for their invaluable comments; Haemin Kim for providing the voice-over. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00450, A Deep Learning Based Immersive AR Content Creation Platform for Generating Interactive, Context and Geometry Aware Movement from a Single Image).

References

- [ACC*18] ANEJA D., CHAUDHURI B., COLBURN A., FAIGIN G., SHAPIRO L., MONES B.: Learning to generate 3d stylized character

- expressions from humans. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018), IEEE, pp. 160–169.
- [ACF*16] ANEJA D., COLBURN A., FAIGIN G., SHAPIRO L., MONES B.: Modeling stylized character expressions via deep learning. In *Asian conference on computer vision* (2016), Springer, pp. 136–153.
- [BCW*18] BAO J., CHEN D., WEN F., LI H., HUA G.: Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 6713–6722.
- [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* (1999), pp. 187–194.
- [CHZ14] CAO C., HOU Q., ZHOU K.: Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–10.
- [CWLZ13] CAO C., WENG Y., LIN S., ZHOU K.: 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.
- [DCFN06] DENG Z., CHIANG P.-Y., FOX P., NEUMANN U.: Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games* (2006), pp. 43–48.
- [FE78] FRIESEN E., EKMAN P.: Facial action coding system: a technique for the measurement of facial movement. *Palo Alto* 3, 2 (1978), 5.
- [GYQ*18] GAO L., YANG J., QIAO Y.-L., LAI Y.-K., ROSIN P. L., XU W., XIA S.: Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.
- [GZY*20] GUO J., ZHU X., YANG Y., YANG F., LEI Z., LI S. Z.: Towards fast, accurate and stable 3d dense face alignment. *arXiv preprint arXiv:2009.09960* (2020).
- [KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [KGT*18] KIM H., GARRIDO 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), 1–14.
- [KSH17] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60, 6 (2017), 84–90.
- [LAR*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F., DENG Z.: Practice and theory of blendshape facial models. In *Eurographics* (2014).
- [LBK17] LIU M.-Y., BREUEL T., KAUTZ J.: Unsupervised image-to-image translation networks. In *Advances in neural information processing systems* (2017), pp. 700–708.
- [LKA*17] LAINE S., KARRAS T., AILA T., HERVA A., SAITO S., YU R., LI H., LEHTINEN J.: Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation* (2017), pp. 1–10.
- [NHSW20] NARUNIEC J., HELMINGER L., SCHROERS C., WEBER R.: High-resolution neural face swapping for visual effects. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 173–184.
- [NKH19] NIRKIN Y., KELLER Y., HASSNER T.: Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 7184–7193.
- [NN01] NOH J.-Y., NEUMANN U.: Expression cloning. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 277–288.
- [NYM18a] NATSUME R., YATAGAWA T., MORISHIMA S.: Fsnnet: An identity-aware generative model for image-based face swapping. In *Asian Conference on Computer Vision* (2018), Springer, pp. 117–132.
- [NYM18b] NATSUME R., YATAGAWA T., MORISHIMA S.: Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447* (2018).
- [PGC*20] PETROV I., GAO D., CHERVONIY N., LIU K., MARANGONDA S., UMÉ C., JIANG J., RP L., ZHANG S., WU P., ET AL.: Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535* (2020).
- [Pho75] PHONG B. T.: Illumination for computer generated pictures. *Communications of the ACM* 18, 6 (1975), 311–317.
- [RRN*20] RAVI N., REIZENSTEIN J., NOVOTNY D., GORDON T., LO W.-Y., JOHNSON J., GKIOXARI G.: Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501* (2020).
- [RZL*17] RIBERA R. B. I., ZELL E., LEWIS J. P., NOH J., BOTSCH M.: Facial retargeting with automatic range of motion alignment. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.
- [SCSN11] SONG J., CHOI B., SEOL Y., NOH J.: Characteristic facial retargeting. *Computer Animation and Virtual Worlds* 22, 2–3 (2011), 187–194.
- [SL14] SEOL Y., LEWIS J. P.: Tuning facial animation in a mocap pipeline. In *ACM SIGGRAPH 2014 Talks*. 2014, pp. 1–1.
- [SLS*12] SEOL Y., LEWIS J. P., SEO J., CHOI B., ANJYO K., NOH J.: Spacetime expression cloning for blendshapes. *ACM Transactions on Graphics (TOG)* 31, 2 (2012), 1–12.
- [SLT*19] SIAROHIN A., LATHUILIÈRE S., TULYAKOV S., RICCI E., SEBE N.: First order motion model for image animation. *Advances in Neural Information Processing Systems* 32 (2019), 7137–7147.
- [SML16] SEOL Y., MA W.-C., LEWIS J.: Creating an actor-specific facial rig from performance capture. In *Proceedings of the 2016 Symposium on Digital Production* (2016), pp. 13–17.
- [SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 399–405.
- [SWR*21] SIAROHIN A., WOODFORD O., REN J., CHAI M., TULYAKOV S.: Motion representations for articulated animation. In *CVPR* (2021).
- [SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [TBG*19] TEWARI A., BERNARD F., GARRIDO P., BHARAJ G., ELGHARIB M., SEIDEL H.-P., PÉREZ P., ZOLLHOFER M., THEOBALT C.: Fml: Face model learning from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10812–10822.
- [TLL19] TRAN L., LIU F., LIU X.: Towards high-fidelity nonlinear 3d face morphable model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 1126–1135.
- [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.
- [TZB*18] TEWARI A., ZOLLHOFER M., BERNARD F., GARRIDO P., KIM H., PEREZ P., THEOBALT C.: High-fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder. *IEEE transactions on pattern analysis and machine intelligence* 42, 2 (2018), 357–370.
- [TZN*15] THIES J., ZOLLHÖFER M., NIESSNER M., VALGAERTS L., STAMMINGER M., THEOBALT C.: Real-time expression transfer for facial reenactment. *ACM Trans. Graph.* 34, 6 (2015), 183–1.
- [TZN19] THIES J., ZOLLHÖFER M., NIESSNER M.: Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.
- [TZS*16] THIES J., ZOLLHOFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: Face2face: Real-time face capture and reenactment

of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2387–2395.

[ZCZ20] ZHANG J., CHEN K., ZHENG J.: Facial expression retargeting from human to avatar made easy. *IEEE Transactions on Visualization and Computer Graphics* (2020).

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595.