Versatile Face Animator: Driving Arbitrary 3D Facial Avatar in RGBD Space

Haoyu Wang
why22@mails.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Haozhe Wu
wuhz19@mails.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Junliang Xing
jlxing@tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Jia Jia*
jjia@tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing National Research Center for Information Science and Technology
Beijing 100084, China

ABSTRACT

Creating realistic 3D facial animation is crucial for various applications in the movie production and gaming industry, especially with the burgeoning demand in the metaverse. However, prevalent methods such as blendshape-based approaches and facial rigging techniques are time-consuming, labor-intensive, and lack standardized configurations, making facial animation production challenging and costly. In this paper, we propose a novel self-supervised framework, Versatile Face Animator, which combines facial motion capture with motion retargeting in an end-to-end manner, eliminating the need for blendshapes or rigs. Our method has the following two main characteristics: 1) we propose an RGBD animation module to learn facial motion from raw RGBD videos by hierarchical motion dictionaries and animate RGBD images rendered from 3D facial mesh coarse-to-fine, enabling facial animation on arbitrary 3D characters regardless of their topology, textures, blendshapes, and rigs; and 2) we introduce a mesh retarget module to utilize RGBD animation to create 3D facial animation by manipulating facial mesh with controller transformations, which are estimated from dense optical flow fields and blended together with geodesic-distance-based weights. Comprehensive experiments demonstrate the effectiveness of our proposed framework in generating impressive 3D facial animation results, highlighting its potential as a promising solution for the cost-effective and efficient production of facial animation in the metaverse.

*Corresponding author

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612065

CCS CONCEPTS

• **Computing methodologies** → **Animation**.

KEYWORDS

3D facial animation; Motion capture; Motion retargeting.

ACM Reference Format:
Haoyu Wang, Haozhe Wu, Junliang Xing, Jia Jia. 2023. Versatile Face Animator: Driving Arbitrary 3D Facial Avatar in RGBD Space. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, Ottawa, Canada, 9 pages. https://doi.org/10.1145/3581783.3612065

1 INTRODUCTION

Creating 3D facial animation is a complex task with numerous applications in fields such as movie production and the gaming industry [4]. With the increasing popularity of the *metaverse*, the demand for 3D facial animation has significantly grown. In the metaverse, users expect to interact with diverse avatars, including themselves, celebrities, and fictional characters like the *Na'vi* in *Avatar*. These needs necessitate sophisticated facial animation techniques that accurately capture the nuances of human expressions and emotions. However, creating believable facial animation remains a challenge even for the most skilled animators, Hollywood filmmakers, or game developers due to the high level of expertise and the significant amount of time required.

One of the standard approaches for creating 3D facial animation in the industry is *performance retargeting*. This method involves capturing facial motion from a real actor and transferring that motion to a target 3D avatar [4]. The most common technique for achieving this is the blendshape-based methods [3, 19, 20]. Blendshapes consist of predefined deformations of the facial mesh, which can be combined to create a wide range of facial expressions using various input weights. This process allows for transferring facial motion from a source video to a target character, even if visually dissimilar. However, creating blendshapes with high flexibility and rich expressiveness is time-consuming and labor-intensive, as it may require hundreds of various expressions for a single character.

For example, in *The Curious Case of Benjamin Button*, filmmakers captured 170 blendshapes of Brad Pitt [9]. Additionally, the absence of a standard configuration for creating blendshapes complicates cross-mapping between different expression spaces and hinders motion transfer across distinct avatars. Another prevalent approach for generating 3D facial animation is facial rigging, which involves manipulating motion controls to create the desired animation [27]. However, facial rigging is typically an iterative and laborious process since no consistent rig can be used for all possible motions. As a result, the rigging process often becomes a bottleneck in 3D animation production. Furthermore, varying standards across different software packages make transferring facial motion across characters with distinct rigs extremely challenging.

Both blendshape-based methods and facial rigging, as discussed above, are facing similar challenges: (i) they are time-consuming and labor-intensive, which limits their accessibility to common users, and (ii) they lack standardization, making it challenging to transfer facial motion across different characters with varying rigs or blendshape configurations. These challenges hinder the development of the metaverse, where users expect to act on arbitrary characters in a short set-up time. Motivated by these issues, we aim to explore a new solution that directly drives the facial mesh with raw RGBD videos, eliminating reliance on blendshapes or rigs.

To this end, we propose a novel framework, Versatile Face Animator (VFA), that combines facial motion capture with motion retargeting to drive the facial mesh with captured RGBD videos end-to-end. We aim to model the facial motion in color and depth fields and generate RGBD animation to drive facial mesh. To achieve this goal, our framework consists of the RGBD animation module and the mesh retarget module. First, the RGBD animation module generates the animated RGBD frame with hierarchical motion dictionaries. It then estimates the correspondence between the source RGBD image and the animated frame with a distilled flow generator. More specifically, the RGBD animation module encodes arbitrary facial motion into a combination of basic transformations in the motion dictionary and generates the animated frame from coarse to fine. The flow generator is then trained to estimate a dense optical flow field for building correspondence between source images and animated frames. The flow generator is distilled from the RGBD generator under animated RGBD frames' supervision, eliminating the need for extra labels. The mesh retarget module then deforms the facial mesh with the dense optical flow. It first detects the controlling points of the mesh automatically and then calculates the geodesic-distance-based controlling weights of each vertex. Afterward, the mesh retarget module estimates controlling point transformations according to the dense optical flow. The transformations are then blended with calculated weights to deform the mesh.

To summarize, this work makes three main contributions:

- We propose VFA, a novel self-supervised framework that combines facial motion capture with facial motion retargeting in an end-to-end manner, providing a cost-effective solution for 3D facial animation production.
- We introduce a new method to learn facial motion in both the color field and the depth field with hierarchical motion dictionaries and generate RGBD animation coarse-to-fine.
- We present a new pipeline for transferring RGBD animation to create 3D animation by deforming the mesh with controller transformations, which are estimated from a dense optical flow field and blended with geodesic-distance-based controlling weights.

Our approach presents two main advantages: 1) It employs self-supervised training using raw facial RGBD data, eliminating the need for annotation or additional configuration; and 2) it can animate arbitrary 3D characters, regardless of their topology, blendshapes, or rigs. A comprehensive set of experiments, encompassing both qualitative and quantitative analyses, showcases the outstanding performance of our method in generating 3D facial animations at a relatively low cost. This positions our approach as a promising solution for 3D facial animation production.

2 RELATED WORK

Blendshapes and Facial Rigging. Blendshapes, an approximate semantic parameterization of facial expression, have become the standard approach to generating realistic facial animation in the industry [19]. With little computation, an extensive range of expressions can be expressed by linearly combining blendshape targets. However, hundreds of blendshape targets are required to build an expression space with enough expressiveness. To reduce this unbearable cost, researchers have proposed methods to reduce the demands of training expressions [3] or to fine-tune the blendshape model based on a generic prior [20]. To deal with transferring blendshape weights across different expression spaces, Kim *et al.* proposed a method that animated rendered images in the 2D domain and then estimated blendshape weights from the retargeted images [17], which is similar to our proposed framework but can only drive a particular set of avatars due to the reliance of blendshapes. Facial rigging is another widely used technique that seeks to build motion controls and animate the target character [27]. To some extent, blendshape weights can be seen as a kind of control rig. Several neural approaches have been proposed to estimate facial rigs from animation using neural networks [1, 32, 45], which enables motion transfer across characters. However, both blendshape-based methods and facial rigging techniques still suffer from their high configuration costs and lack of standard criteria in production. In our framework, we aim to model the facial motion of RGBD frames via 2D facial animation methods. This approach eliminates the need for blendshapes or facial rigs, reducing the configuration cost while maintaining satisfactory performance.

Data-driven 3D Facial Retargeting. Facial retargeting techniques have significantly advanced with the development of neural networks. The Variational Auto Encoder (VAE) [18] has been introduced to disentangle facial identity and expression in the latent space, allowing for the transfer of facial motion [29, 33]. Recently, Zhang *et al.* [43] proposed training character-specific VAE models to transfer characters' expressions across different domains. Most current studies on neural facial retargeting methods are based on the 3D morphable model (3DMM), which isolates identity and expressions [8, 21]. Moser *et al.* [24] inspired our work by proposing to treat 3D facial retargeting as 2D face swapping between the actor and the target character. They animated the rendered images using an unsupervised image-to-image translation model and

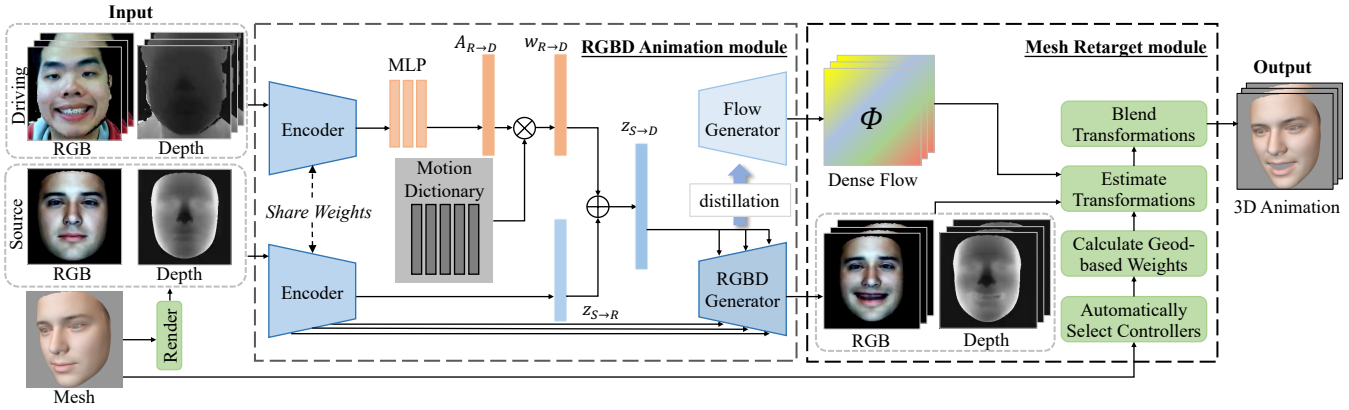

Figure 1: An overview of our proposed framework VFA. We generate 3D facial animation with the source mesh and captured RGBD video as input in an end-to-end manner. Our model consists of an RGBD animation module and a mesh retarget module. The RGBD animation module encodes source image S to $z_{S\to R}$ and encodes facial motion from driving frame D to $w_{R\to D}$ using the motion dictionary $\mathcal{D}_m$. With the composed latent code $z_{S\to D}$, the RGBD animation module generates the driven RGBD frame and estimates a dense optical flow field Φ, which can be used to warp the source image. The mesh retarget module then warp the source mesh $\mathcal{S}$ utilizing information from the animated RGBD pair and dense flow Φ to generate 3D facial animation.

then regressed the 3DMM parameters from the animated images. However, these methods typically require a large amount of paired high-accuracy 3D facial data, which is difficult to capture. Additionally, 3DMM-based methods suffer from a lack of expressiveness due to their linear nature. In our work, we propose to train our method with RGBD videos which be captured easily using a single Azure Kinect V2 camera. Our method eliminates the need for paired 3D facial data and allows arbitrary deformation by warping the facial mesh with an estimated optical flow field.

2D Facial Animation. Generating 2D facial animation, also known as face reenactment, has seen rapid progress due to advancements in deep learning. To facilitate the image-to-image translation model for facial reenactment, researchers have introduced facial structure representations as prior knowledge, such as facial landmarks, [5, 35, 40, 42], semantic label maps [28, 34] and optical flows [22, 26, 41]. However, such semantic labels for supervised learning are usually difficult to access for training and inference. A self-supervised motion transfer approach, i.e., the first-order motion model, was introduced to learn facial keypoint transformations from raw videos and warp the source image with the estimated dense optical flow [30]. Based on the first-order motion model, Hong *et al.* proposed to recover facial depth images in a self-supervised manner and leverage the depth information to generate 2D facial animation [11]. Wang *et al.* proposed a novel method called LIA to drive still 2D images via latent space navigation, which eliminates the need for explicit structure representations like keypoints and can discover high-level motion transformations in latent space [38]. However, there remains a gap between 2D face reenactment and 3D facial retargeting, since most methods treat 3D information as prior knowledge, and few focus on how to transfer facial motion in the 3D mesh. We bridge the gap by incorporating the depth information from RGBD videos and modeling facial motion in the depth field using the depth motion dictionary to generate animated RGBD frames and subsequently deform the facial mesh.

3 METHODOLOGY

We aim to animate the source 3D facial mesh $\mathcal{S}$ of a target avatar based on the facial motion from a raw RGBD video $\mathcal{D}$ captured by Azure Kinect V2. To achieve this, our proposed end-to-end framework consists of the RGBD animation module and the mesh retarget module, as depicted in Fig. 1.

The RGBD animation module is designed to model facial motion extracted from the driving frame **D** of video $\mathcal{D}$ and transfer it to the rendered image **S** from mesh $\mathcal{S}$. Additionally, the RGBD animation module estimates a dense optical flow field Φ, which can be used to establish correspondence between the source image and the animated frame. The estimated dense flow Φ will be utilized in the mesh retarget module.

The mesh retarget module deforms the source mesh $\mathcal{S}$ with detected facial landmarks as controllers and geodesic-distance-based controlling weights. The transformations of controllers are estimated using dense flow field Φ from the RGBD animation module and then are mapped to 3D world space with the animated depth frames. Finally, the transformations are blended to generate the desired 3D facial animation frame by frame. In the following, we will introduce the two modules in detail.

3.1 RGBD Animation Module

3.1.1 Encoder The RGBD animation module is inspired by the LIA method proposed by Wang *et al.* [38]. It utilizes an auto-encoder structure and consists of an encoder and two generators. The encoder encodes images to latent codes, and the RGBD generator decodes these codes and generates animated RGBD frames coarse-to-fine. Furthermore, the flow generator estimates dense optical flow fields which will be utilized in the mesh retarget module. In the following section, we proceed to discuss them comprehensively.

The encoder is designed to learn a latent code $z_{S\to D}$ to represent the motion transformation from **S** to **D**. However, as Wang *et al.* [38] point out, it is challenging to learn $z_{S\to D}$ directly from the input image pair, as the model needs to model the direction and norm

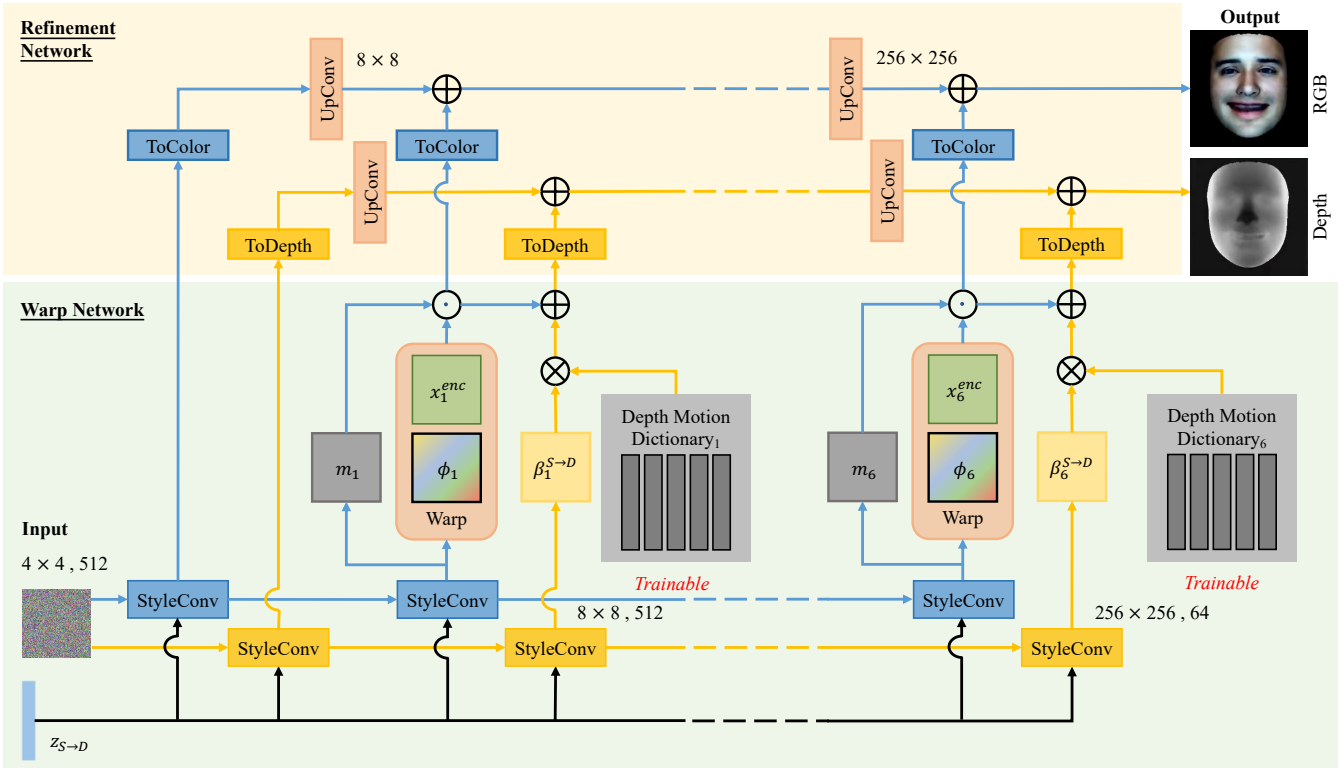

Figure 2: An overview of the generator. The generator employs a 6-level pyramid architecture, comprising the warp network and the refinement network. The warp network utilizes *StyleConv* [14] to estimate optical flow fields $\{\phi_i\}_1^6$ and masks $\{m_i\}_1^6$, and warps feature maps x_i^{enc} from the encoder. The depth motion dictionaries are introduced to model motion in the depth field. The refinement network then utilizes convolution layers to generate the animated frames in a coarse-to-fine manner.

of the vector $z_{S\to D}$ simultaneously. To overcome this challenge, we assume there exists a reference frame **R** so that the motion transformation from **S** to **D** can be decomposed as **S** $\to$ **R** $\to$ **D**. This allows us to learn the transformations **S** $\to$ **R** and **R** $\to$ **D** independently and then compose them to represent **S** $\to$ **D**. We model $z_{S\to D}$ as the target point in the latent space, which can be reached from the source point $z_{S\to R}$ along a path $w_{R\to D}$ in the latent space. Mathematically, the latent code $z_{S\to D}$ can be decomposed as $z_{S\to D} = z_{S\to R} + w_{R\to D}$.

To ensure that the learned latent codes are in the same latent space, we utilize a single encoder to encode the source image and the driving image. As depicted in Fig. 1, the encoder encodes the source image and the driving image as $z_{S\to R}$ and $z_{D\to R}$ respectively. To extract high-level motion information from $z_{D\to R}$, we propose to encode motion via Linear Motion Decomposition [38]. Specifically, we introduce a learnable orthogonal basis called the motion dictionary D_m. Each vector of the motion dictionary represents a direction $\mathbf{d_i}$ of the motion space. $z_{D\to R}$ is mapped to a magnitude vector $A_{R\to D}$ by an MLP layer. Then, the latent path $w_{R\to D}$ is obtained by linearly combining the magnitude vector $A_{R\to D}$ with the basis vector $\mathbf{d_i}$ from the motion dictionary D_m. With the latent code $z_{S\to R}$ learned from the source image **S** and the latent path $w_{R\to D}$ extracted from the driving frame **D**, we can obtain $z_{S\to D}$ which represents the transformation **S** $\to$ **D**.

3.1.2 Generator We proceed to introduce the RGBD generator and the flow generator respectively.

The general architecture of the RGBD generator is depicted in Fig. 2, which consists of the flow and refinement networks. To learn multi-scale features, the generator employs a 6-level pyramid architecture and uses skip connection between layers. The *StyleConv* [14] layer is introduced to decode $z_{S\to D}$ and estimate multiple levels of optical flow fields $\{\phi_i\}_1^6$. These optical flow fields $\{\phi_i\}_1^6$ are then used to warp the feature maps x_i^{enc} from the corresponding level of the source encoder. However, as Siarohin *et al.* [30] pointed out, the occluded parts of the source image **S** can not be recovered by simply warping the image. Consequently, we propose to estimate the masks $\{m_i\}_1^6$ along with $\{\phi_i\}_1^6$. The masks are utilized to mask the occluded region to inpaint in the refinement network. In this way, the transformed feature map is formulated as:

$$x_i' = m_i \odot f_w(x_i^{enc}, \phi_i),$$

where f_w represents the backward warping function.

The estimated optical flow $\{\phi_i\}_1^6$ provides the pixel-wise correspondence between the source and warped images in the 2D domain, but it is not enough to model motion in the depth field. When an object moves relative to the camera's z-axis, it can alter the pixel values in the depth frame, which can not be captured by the optical flow $\{\phi_i\}_1^6$. To address this issue, we introduce the depth

motion dictionaries $\{D_i^{depth}\}_1^6$ to adequately learn motion in the depth field. The basis vectors of the depth motion dictionary represent the direction of the depth motion space. By linearly combining the basis vectors with the predicted magnitude vector $\beta_i^{S\to D}$, we estimate the motion in the depth field. This allows us to obtain the feature map x_i^{depth} for generating accurate depth images. x_i^{depth} can be expressed as:

$$x_i^{depth} = m_i \odot f_w(x_i^{enc}, \phi_i) + \sum_{j=1}^{M} \beta_{i,j}^{S\to D} \mathbf{d}_{i,j}^{depth},$$

where M denotes the size of the depth motion dictionary D_i^{depth}, and $\mathbf{d}_{i,j}^{depth}$ represents the basic vectors of D_i^{depth}.

In the refinement network, we adopt a coarse-to-fine approach to generate precise RGBD results. At each layer of the refinement network, we combine the upsampled results from the previous layer with inpainted feature maps to generate images. This iterative process allows us to progressively refine the generated images in a hierarchical manner, capturing finer details and improving the overall visual quality of the outputs.

We note that the warp network predicts optical flow fields ϕ_i to warp the feature maps x_i^{enc} from the encoder. This poses a challenge for the mesh retarget module in analyzing the flow fields and accurately tracking the movement of controlling points during animation. To address this challenge, we introduce a dense flow generator to generate a dense optical flow field, denoted as Φ, which represents the pixel-wise correspondence between the input image $\mathbf{S}$ and the animated image. The dense flow generator is trained through distillation from the original generator, utilizing the warped image from the refinement network and the source image as training data. This training scheme allows the dense flow generator to generate Φ without conversion or extra training data. This approach facilitates the mesh retarget module to track the transformation of controllers to perform mesh retargeting.

3.2 Mesh Retarget Module

The design of the mesh retarget module is inspired by linear blend skinning (LBS), the most popular shape deformation algorithm for real-time animation due to its efficiency and simplicity [15, 16]. Our method modifies the vertex positions while preserving mesh connectivity to achieve accurate and consistent animation results for different target meshes. Furthermore, we determine controlling weights using geodesic distances, which maintain the mesh topology and produce natural results.

We use an open-source library, Mediapipe [23] to detect facial landmarks as controllers. These detected landmarks provide rich semantic information facilitating reasonable and transferable blend transformations. Then we compute geodesic distances on the mesh surface between controlling points and mesh vertices. For each mesh vertex, we assign the 10 nearest controllers to determine the controlling weights based on the inverse square of the geodesic distances. We must note that we use geodesic distance instead of Euclidean distance to preserve the mesh topology. Specifically, using geodesic distance as the metric ensures that the upper and lower lip vertices are not mistakenly considered neighbors. Further details on the comparison are discussed in Section 4.5.

When generating animation frame by frame, we analyze the flow generator's dense flow field Φ to estimate controller transformations. However, these estimated transformations are in the 2D screen space, while the source mesh $\mathcal{S}$ exists in 3D world space. The transformations can not be directly aggregated to deform the mesh. Therefore, to map the transformation to 3D space, we estimate the position of controller v_j utilizing the depth of its corresponding pixel and unproject it to the 3D world space using the perspective matrix. We formulate this process as follows:

$$\mathbf{v}_j = \mathbf{P}^{-1}(v_j.x, v_j.y, d(v_j), 1)^T,$$

where P denotes the perspective matrix, and $d(v_j)$ denotes the depth value of v_j's corresponding pixel in the image. We then track the movement of the controllers with the dense optical flow field Φ and estimate controller transformations in 3D world space. The deformed vertex positions can be calculated by linearly combining the transformations with the controlling weights.

The mesh retarget module plays a critical role in our proposed framework, bridging the 2D image animation problem and the 3D facial retargeting problem. By utilizing geodesic-determine controlling weights and incorporating depth information from generated frames, this module enables direct warping of the source mesh $\mathcal{S}$ without blendshapes or rigs. This integration simplifies retargeting facial motion to avatars, making our proposed framework a cost-efficient solution for creating realistic 3D facial animation.

3.3 Training Losses

In the training stage, the RGBD animation module is trained in a self-supervised manner to reconstruct the driving frame $\mathbf{D}$. To further enhance the robustness and performance of our model, we fine-tune the RGBD module based on the weights of LIA pre-trained in the VoxCeleb [25] dataset. Four losses are used to train the RGBD module: a reconstruction loss $\mathcal{L}_{rec}$, a perceptual loss $\mathcal{L}_{vgg}$, a smooth loss $\mathcal{L}_{sm}$ and a structure preserve loss $\mathcal{L}_{sp}$.

$\mathcal{L}_{rec}$ is calculated using $\mathcal{L}_1$ distance, while the perceptual loss $\mathcal{L}_{vgg}$, proposed by Johnson *et al.* [13], is calculated on multi-scales feature maps extracted from the pre-trained VGG-19 network [31].

To improve the quality of the depth image we generated, we design two depth-related losses: the smooth loss $\mathcal{L}_{sm}$ and the structure preserve loss $\mathcal{L}_{sp}$. The smooth loss is designed based on the Laplacian operator to improve the smoothness of $\hat{\mathbf{D}}$:

$$\mathcal{L}_{sm} = \mathbb{E}\left|\nabla^2\mathbf{D} - \nabla^2\hat{\mathbf{D}}\right|_2,$$

where $\nabla^2\mathbf{D}$ denotes a Laplacian operator:$\nabla^2\mathbf{D}(x,y) = \mathbf{D}(x+1,y) + \mathbf{D}(x-1,y) + \mathbf{D}(x,y+1) + \mathbf{D}(x,y-1) - 4\mathbf{D}(x,y)$

Furthermore, to preserve the original geometric structure and the depth discontinuity along the edges in depth frames, we introduce the structure preserve loss $\mathcal{L}_{sp}$ proposed by Jeon *et al.* [12]:

$$\mathcal{L}_{sp} = \mathbb{E}_p\left|\max_{q\in\Omega(p)}\left|\nabla\mathbf{D}(p)\right| - \max_{q\in\Omega(p)}\left|\nabla\hat{\mathbf{D}}(p)\right|\right|_2,$$

where $\Omega(p)$ denotes a local region in the neighborhood of p, and $\nabla\mathbf{D}(p)$ denotes the gradient calculated as: $\nabla_x\mathbf{D}(x,y) = \mathbf{D}(x+1,y) - \mathbf{D}(x-1,y)$, $\nabla_y\mathbf{D}(x,y) = \mathbf{D}(x,y+1) - \mathbf{D}(x,y-1)$. In practice, we set $\Omega(p)$ as a 5×5 window near the pixel p.

Table 1: Results of Same-identity Reconstruction. We compared our method with three state-of-the-art methods on two datasets: MMFace4D [39] and VocCeleb [25]. For all metrics except CSIM, the lower, the better.

Method	MMFace4D							VoxCeleb				
	$\mathcal{L}_1$	LPIPS	AKD	AED	CSIM	Depth $\mathcal{L}_1$	$\mathcal{L}_{sm}$	$\mathcal{L}_1$	LPIPS	AKD	AED	CSIM
FOMM [30]	10.61	0.123	2.822	0.537	0.839	0.040	0.019	12.27	0.128	2.398	0.574	0.814
OSFV [36]	9.32	0.121	2.612	0.528	0.842	0.037	0.017	11.87	0.121	**2.385**	0.562	0.822
DaGAN [11]	8.13	0.104	2.502	0.529	**0.844**	0.036	0.015	11.77	0.122	2.542	0.570	0.820
Ours	**8.04**	**0.104**	**2.312**	**0.440**	**0.844**	**0.030**	**0.015**	**11.26**	**0.119**	2.475	**0.570**	**0.820**

Table 2: Quantitative Results of Cross-identity Motion Retargeting. We compared our method with three state-of-the-art methods using three designed tasks. The lower video FID [37] indicates better generation qualities.

Method	Vox2→Vox2	MM→Vox2	MM→MM
FOMM [30]	46.86	42.55	42.29
OSFV [36]	45.18	42.81	42.18
DaGAN [11]	46.02	42.01	41.18
Ours	**44.47**	**40.87**	**40.69**

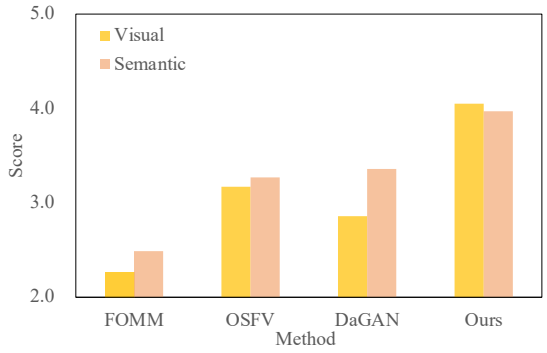

Figure 3: User Study of Motion Retargeting. We asked 20 users to evaluate the generated videos' visual quality and semantic consistency with the driving video. The score is in the range of 1-5, and a higher score denotes better.

Our full loss function while training the RGBD animation module is the combination of the four losses discussed above:

$$\mathcal{L} = \mathcal{L}_{vgg} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{sm}\mathcal{L}_{sm} + \lambda_{sp}\mathcal{L}_{sp},$$

where we use three user-define hyperparameters for balance. In practice, these parameters are set as $\lambda_{rec} = \lambda_{sm} = 200, \lambda_{sp} = 50$. It is important to note that our method exhibits robustness to different hyperparameter settings. Consequently, we do not demonstrate an ablation study examining the combined loss function.

4 EXPERIMENTS

4.1 Experiments Settings

Dataset Our model is pre-trained in the VoxCeleb [25] dataset, and fine-tuned in the **MMFace4D** dataset proposed by Wu *et al.* [39]. The MMFace4D dataset is a large-scale facial RGBD video dataset captured by Azure Kinect V2. During training, we selected 191 identities, used 16,549 videos, cropped the facial region, removed the background, resized the frames to 256×256, and normalized the frames to the range of $[-1, 1]$. For testing, we utilized the test dataset from VoxCeleb [25] and VoxCeleb2 [6] as well as RGBD videos of 41 unseen identities from MMFace4D.

Baselines Our proposed method is the first neural approach attempting to create 3D facial animation driven by raw RGBD videos in an end-to-end manner, utilizing estimated optical flow fields to transform mesh vertices and deform the facial mesh. To provide a comprehensive evaluation of our method, we compare it with three state-of-the-art optical-flow-based 2D animation methods: FOMM [30], OSFV [36] and DaGAN [11]. To animate RGBD images and drive the 3D mesh under our framework, we modify these methods and train them on the MMFace4D dataset using the loss function formulated in Sec. 3.3. Both methods are initialized with pre-trained weights on VoxCeleb [25].

4.2 Evaluate Metrics

We evaluate the performance of our model based on: (i) reconstruction fidelity using $\mathcal{L}_1$ and LPIPS metrics, (ii) generated video quality using video FID, (iii) semantic consistency using average keypoint distance (AKD), average Euclidean distance (AED) and cosine similarity (CSIM), and (iv) generated depth images quality using $\mathcal{L}_1$ and $\mathcal{L}_{sm}$ in Sec. 3.3. These metrics provide us with a comprehensive evaluation of our model.

Video FID [37], derived from Fréchet inception distance (FID), is a metric that assesses both the visual quality and temporal consistency of the generated videos. Lower video FID indicates a higher quality of the generated videos. In our experiments, we utilize a pre-trained ResNext101 [10] model to extract spatiotemporal features and compute video FID as an objective measure of video quality.

AKD aims to measure the difference between the facial landmarks of the reconstructed frame $\hat{\mathbf{D}}$ and the real frame $\mathbf{D}$. We adopt the facial landmark detection method proposed by Bulat and Tzimiropoulos [2] and compute the average distance between corresponding landmarks as AKD.

AED and CSIM [42] both evaluate the ability to preserve identity while generating videos. We extracted identity embedded features with ArcFace [7], calculated the mean Euclidean distance between the identity embeddings as AED, and the cosine similarity between the embeddings as CSIM.

4.3 Quantitative Analysis

To provide a quantitative analysis, we conduct two experiments to evaluate our framework thoroughly: same-identity reconstruction

Figure 4: RGBD results of cross-identity motion retargeting. The first column shows the source images, while the second column shows the driving frames. The following columns show the transferred results of FOMM [30], OSFV [36], DaGAN [11], and our method, respectively.

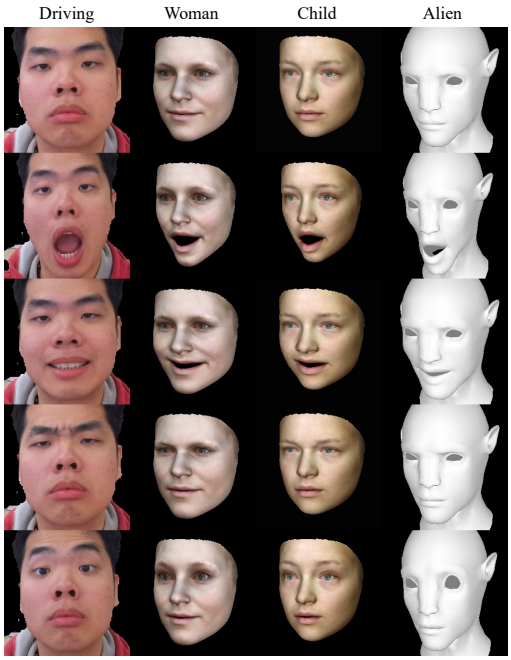

Figure 5: Qualitative results from our method. The leftmost column displays the driving frames. In contrast, the subsequent columns exhibit three target characters: a woman, a child, and an alien. The top row shows the source meshes. More results are presented in the supplementary material.

in Sec. 4.3.1 to assess the quality of our reconstruction, and cross-identity motion retargeting in Sec. 4.3.2 to evaluate the motion transfer ability of our approach.

4.3.1 Same-identity Reconstruction In this experiment, we aim to evaluate the reconstructing ability of our method. For simplicity, we focus on evaluating the quality of RGBD animation, as it directly affects the quality of mesh retargeting in our framework. We used the first frame as the source image (**S**) and the remaining frames as driving frames (**D**) to reconstruct the video. We conducted this experiment on the MMFace4D dataset and the VoxCeleb test set and reported the results in Tab. 1.

As Tab. 1 shows, our method achieves the best performance across all the metrics. Compared with the three baseline methods, our method achieves the highest reconstruction fidelity in both datasets, particularly in depth frames. This result further validates the effectiveness of the depth motion dictionaries proposed in this paper. While FOMM [30] and OSFV [36] treat the depth information as a simple image channel, and DaGAN [11] fails to model motion in the depth field, our method excels in depth reconstruction due to the depth motion dictionaries. Furthermore, our method achieves the highest scores in AKD, AED, and CSIM, indicating its ability to transfer motion while preserving the identity of the source character. These results highlight the strength of our multi-level flow-based generator.

4.3.2 Cross-identity Motion Retargeting In this experiment, we aim to assess the motion transfer ability of our method. Specifically, we used source images (**S**) and driving frames (**D**) from different video sequences, which differs from Sec. 4.3.1. We designed three tasks: driving source images from VoxCeleb2 with videos from VoxCeleb2 (Vox2→Vox2), driving source images from MMFace4D with videos from VoxCeleb2 (MM→Vox2), and driving source images from MMFace4D with videos from MMFace4D (MM→MM). It's important to note that all the images and videos used here were unseen during the training of our models, ensuring a fair evaluation.

As the ground truth animation videos were unavailable, we used video FID [44] to assess our generated videos' visual quality and temporal consistency. We randomly selected 2200 source images and driving video clips for each task to generate retargeted videos. These videos were then downsampled to the resolution of 112×112 and randomly cut to 32 frames. We computed video FID by calculating the distance between the generated data and the real data distributions sampled from the source dataset. The results are presented in Tab. 2. Our method consistently outperforms the other methods regarding video FID for all tested tasks, demonstrating superior motion transfer ability. To provide an intuitive demonstration of the performance of the four methods, we show some transferred RGBD results in Fig. 4. FOMM produced some artifacts, such as a puffy face, while OSFV generated noisy results in color and depth frames. Although DaGAN transferred the facial motion better, it did not preserve the identity well. In contrast, our method generated the most natural and clearest color frame and the cleanest and smoothest depth frame, achieving the best performance in transferring the facial motion of RGBD frames.

To further compare the effectiveness of our method with baseline methods, we conducted a user study. Each participant was asked to evaluate and rate the videos generated by the methods. Specifically,

(a) Wave-lip artifact (b) Our method

Figure 6: Comparison of mesh retargeting results. (a) Wave-lip artifacts are caused by using Euclidean distance to calculate blend weights $w_{i,j}$. (b) More natural results are obtained with our method using geodesic distance to determine $w_{i,j}$.

we randomly generated groups of videos. Each video group contained a video generated by our method and three videos generated by the three baselines. These video groups and their corresponding driving videos were presented to 20 human raters. The raters were asked to evaluate the video's visual quality and semantic consistency. As Fig. 3 reported, our method obtained the highest scores, which means that our method generates the most realistic video while transferring facial motion from driving videos.

Notably, the three baselines rely on facial keypoints transformation, so their performance may be affected by the accuracy of the keypoint detector. However, our method captures facial motion by hierarchical motion dictionaries and generates RGBD frames coarse-to-fine, which facilitates realistic motion retargeting.

4.4 Qualitative Analysis

In Figure 5, we present qualitative examples of our proposed method. Specifically, we recorded an RGBD video using the Azure Kinect V2 to drive the facial expressions of three target characters: a woman, a child, and an alien. Despite the dissimilarity between the actor and the target characters, our method generated impressive results and accurately retargeted facial motion, particularly the motion of the mouth, and transferred micro-expressions, such as eye-widening, squinting, and mouth stretching. Furthermore, our method demonstrated impressive ability in animating the alien avatar, which had a significantly different appearance from the actor and was not seen during the training phase. However, expressions such as rolling eyes, gazing, and sticking out the tongue could not be transferred to the target character, as the target mesh did not model eyes and tongue separately. Overall, our results demonstrate the potential of our method as a novel solution for generating 3D facial animation.

4.5 Ablation Study

4.5.1 Controlling Weights Calculation As discussed in Section 3.2, using geodesic distances to calculate controlling weights is crucial for producing accurate retargeted results. To further verify this assertion, we present a case study. When controlling weights are calculated using Euclidean distance, artifacts such as the wave-lip artifact can occur when the mouth is open, as illustrated in Fig. 6(a). This is because the movement of controlling points from the lower lip heavily influences the vertex of the upper lip. However, calculating blend weights using geodesic distance can avoid such artifacts, as the controlling points of the lower lip will not be considered neighbors of the vertex in the upper lip. Therefore, our method generates more natural results, as shown in Fig. 6(b).

Table 3: Ablation Study on Depth Motion Dictionary.

Size of D_i^{depth}	MMFace4D			VoxCeleb	
	$\mathcal{L}_1$	LPIPS	Depth $\mathcal{L}_1$	$\mathcal{L}_1$	LPIPS
0	8.64	0.118	0.043	11.42	0.129
5	**8.04**	**0.104**	**0.030**	**11.26**	**0.119**
10	8.71	0.116	0.036	11.69	0.126
20	8.73	0.119	0.035	11.35	0.127

4.5.2 Depth motion dictionary We provide an in-depth analysis of our design of the depth motion dictionaries $\{D_i^{depth}\}_1^6$ in the generator, as discussed in Sec. 3.1.2. We focus on whether introducing D_i^{depth} benefits the generation of RGBD frames and determine the optimal number of basis vectors that D_i^{depth} requires.

Here we performed the same task as discussed in Sec. 4.3.1, and reported the reconstruction faithfulness metrics, *i.e.*, $\mathcal{L}_1$, and LPIPS. As shown in Tab. 3, the depth motion dictionary D_i^{depth} indeed benefits the reconstruction ability of our method, especially in terms of depth image generation. Notably, when the size of D_i^{depth} is set to 5, our model achieves the best reconstruction results, which indicates that a few basic transformations can represent the depth motion space. Thus, a small depth motion dictionary is sufficient to model facial motion in the depth field.

5 CONCLUSION

In this paper, we propose a novel self-supervised framework, Versatile Face Animator, for transferring facial motion from captured RGBD videos to 3D facial meshes to create 3D facial animation. Our framework comprises two modules: a flow-based RGBD animation module that animates RGBD frames with hierarchical motion dictionaries and a mesh retarget module that performs 3D facial retargeting using blend transformations. Our end-to-end approach eliminates the need for labor-intensive and time-consuming blendshape-based methods or facial rigging techniques. Extensive experiments demonstrate that our framework is a promising and cost-efficient solution for generating 3D facial animation compared with existing literature. However, there are still some limitations to our method. The RGBD animation module may not perform well in some occluded cases, and more training data may be required to improve retarget performance for unseen avatars. Additionally, the estimation of the controller transformations and the accuracy of the generated depth frames significantly influence the realisticness of the retargeted mesh. In future work, we plan to focus on improving the quality of generated RGBD frames and the versatility of our framework for 3D facial animation production. We believe that the simplicity, efficiency, and versatility of our framework are crucial steps toward the future of the metaverse.

6 ACKNOWLEDGEMENTS

This work is supported by the National Key R&D Program of China under Grant No. 2021QY1500, the State Key Program of the National Natural Science Foundation of China (NSFC) (No.61831022).

REFERENCES

[1] Stephen W Bailey, Dalton Omens, Paul Dilorenzo, and James F O'Brien. 2020. Fast and deep facial deformations. *ACM Transactions on Graphics* 39, 4 (2020), 94–109.

[2] Adrian Bulat and Georgios Tzimiropoulos. 2017. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1021–1030.

[3] Emma Carrigan, Eduard Zell, Cédric Guiard, and Rachel McDonnell. 2020. Expression Packing: As-Few-As-Possible Training Expressions for Blendshape Transfer. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 219–233.

[4] Prashanth Chandran, Loïc Ciccone, Markus Gross, and Derek Bradley. 2022. Local anatomically-constrained facial performance retargeting. *ACM Transactions on Graphics* 41, 4 (2022), 1–14.

[5] Zhuo Chen, Chaoyue Wang, Haimei Zhao, Bo Yuan, and Xiu Li. 2022. D2Animator: Dual Distillation of StyleGAN For High-Resolution Face Animation. In *Proceedings of ACM International Conference on Multimedia*. 1769–1778.

[6] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. In *Proceedings of Interspeech*. 1086–1090.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4690–4699.

[8] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. 2021. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics* 40, 4 (2021), 1–13.

[9] Barbara Flueckiger. 2011. Computer-generated characters in Avatar and Benjamin Button. *Digitalitat und Kino. Translation from German by B. Letzler* 1 (2011), 2.

[10] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6546–6555.

[11] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. 2022. Depth-aware generative adversarial network for talking head video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3397–3406.

[12] Junho Jeon and Seungyong Lee. 2018. Reconstruction-based pairwise depth dataset for depth image enhancement using CNN. In *Proceedings of the European Conference on Computer Vision*. 422–438.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*. Springer, 694–711.

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.

[15] Ladislav Kavan. 2014. Direct skinning methods and deformation primitives. *ACM SIGGRAPH Courses* 1, 1 (2014), 11 pages.

[16] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. 2008. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics* 27, 4 (2008), 1–23.

[17] Seonghyeon Kim, Sunjin Jung, Kwanggyoon Seo, Roger Blanco i Ribera, and Junyong Noh. 2021. Deep Learning-Based Unsupervised Human Facial Retargeting. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 45–55.

[18] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*. 14 pages.

[19] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. 2014. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* 1, 8 (2014), 2.

[20] Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-based facial rigging. *ACM Transactions on Graphics* 29, 4 (2010), 1–6.

[21] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics* 36, 6 (2017), 194–210.

[22] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2018. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision*. 600–615.

[23] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172* (2019), 9 pages.

[24] Lucio Moser, Chinyu Chien, Mark Williams, Jose Serra, Darren Hendler, and Doug Roble. 2021. Semi-supervised video-driven facial animation transfer for production. *ACM Transactions on Graphics* 40, 6 (2021), 1–18.

[25] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. In *Proceedings of Interspeech*. 2616–2620.

[26] Katsunori Ohnishi, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Hierarchical video generation from orthogonal information: Optical flow and texture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32. 8 pages.

[27] Verónica Orvalho, Pedro Bastos, Frederic I Parke, Bruno Oliveira, and Xenxo Alvarez. 2012. A Facial Rigging Survey. *Eurographics (State of the Art Reports)* (2012), 183–204.

[28] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. 2019. Video generation from single semantic label map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3733–3742.

[29] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. 2018. Generating 3D faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision*. 704–720.

[30] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. In *Advances in Neural Information Processing Systems*, Vol. 32. 7137–7147.

[31] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*. 14 pages.

[32] Steven L Song, Weiqi Shi, and Michael Reed. 2020. Accurate face rig approximation with deep differential subspace reconstruction. *ACM Transactions on Graphics* 39, 4 (2020), 34–1.

[33] Luan Tran and Xiaoming Liu. 2018. Nonlinear 3d face morphable model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7346–7355.

[34] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. 2019. Few-shot Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems*, Vol. 32. 451–462.

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems*, Vol. 31. 1144–1156.

[36] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 2021. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10039–10049.

[37] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. 2020. G3AN: Disentangling appearance and motion for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5264–5273.

[38] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. 2022. Latent Image Animator: Learning to Animate Images via Latent Space Navigation. In *International Conference on Learning Representations*. 17 pages.

[39] Haozhe Wu, Jia Jia, Junliang Xing, Hongwei Xu, Xiangyuan Wang, and Jelo Wang. 2023. MMFace4D: A Large-Scale Multi-Modal 4D Face Dataset for Audio-Driven 3D Face Animation. *arXiv preprint arXiv:2303.09797* (2023), 14 pages.

[40] Zhuoqian Yang, Wentao Zhu, Wayne Wu, Chen Qian, Qiang Zhou, Bolei Zhou, and Chen Change Loy. 2020. Transmomo: Invariance-driven unsupervised video motion retargeting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5306–5315.

[41] Zili Yi, Qiang Tang, Vishnu Sanjay Ramiya Srinivasan, and Zhan Xu. 2020. Animating through warping: An efficient method for high-quality facial expression animation. In *Proceedings of ACM International Conference on Multimedia*. 1459–1468.

[42] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. 2019. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9459–9468.

[43] Juyong Zhang, Keyu Chen, and Jianmin Zheng. 2020. Facial expression retargeting from human to avatar made easy. *IEEE Transactions on Visualization and Computer Graphics* 28, 2 (2020), 1274–1287.

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 586–595.

[45] Yong Zhao, Haifeng Chen, Hichem Sahli, Ke Lu, and Dongmei Jiang. 2022. Uncertainty-Aware Semi-Supervised Learning of 3D Face Rigging from Single Image. In *Proceedings of ACM International Conference on Multimedia*. 170–179.