ManVatar : Fast 3D Head Avatar Reconstruction Using Motion-Aware Neural Voxels

Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, Yebin Liu

Department of Automation, Tsinghua University

Abstract

With NeRF widely used for facial reenactment, recent methods can recover photo-realistic 3D head avatar from just a monocular video. Unfortunately, the training process of the NeRF-based methods is quite time-consuming, as MLP used in the NeRF-based methods is inefficient and requires too many iterations to converge. To overcome this problem, we propose ManVatar, a fast 3D head avatar reconstruction method using Motion-Aware Neural Voxels. ManVatar is the first to decouple expression motion from canonical appearance for head avatar, and model the expression motion by neural voxels. In particular, the motion-aware neural voxels is generated from the weighted concatenation of multiple 4D tensors. The 4D tensors semantically correspond one-to-one with 3DMM expression bases and share the same weights as 3DMM expression coefficients. Benefiting from our novel representation, the proposed ManVatar can recover photo-realistic head avatars in just 5 minutes (implemented with pure PyTorch), which is significantly faster than the state-of-the-art facial reenactment methods.

1. Introduction

Facial reenactment and head avatar reconstruction from a monocular video have been research hotspots recently, which have a very broad application prospect in VR/AR, digital human, holographic communication, webcast, etc.

However, current methods cannot reconstruct 3D head avatars in minutes, which has become a main limitation of applications. Recent works [11, 12, 15, 45] can recover photo-realistic 3D head avatars using easily available data, such as a monocular video. These methods can be divided into two categories: geometry-based methods and NeRF-based methods. Geometry-based [15, 19, 45] methods can obtain well-defined 3D face geometry. For example, Neural

Project Page: https://www.liuyebin.com/manvatar/manvatar.html

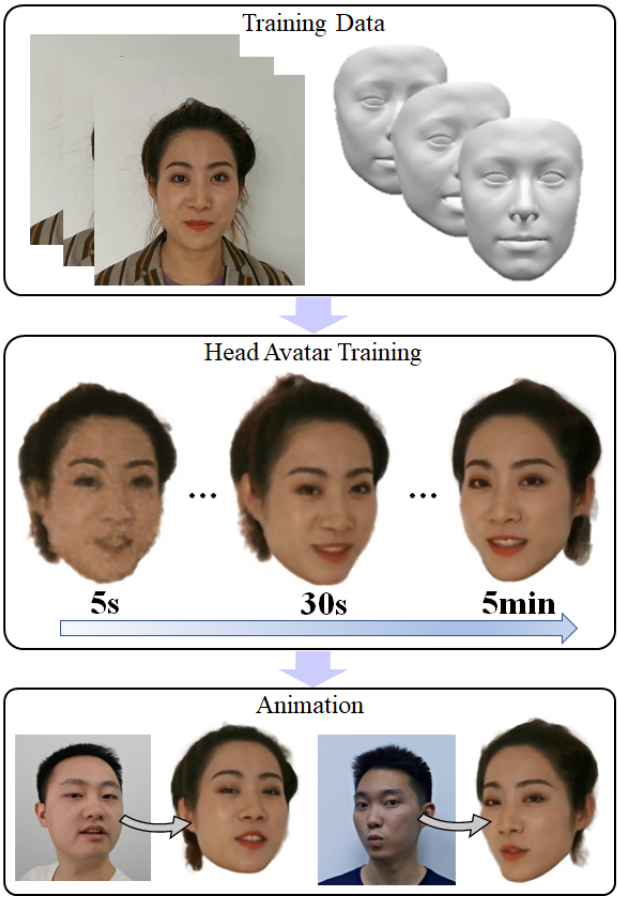

Figure 1. we propose ManVatar, a fast 3D head avatar reconstruction method. Given a monocular video, our method can recover photo-realistic head avatar in 5 minutes.

Head Avatar [15] constructs the avatar model through non-rigid deformation of the mesh template but is limited by the topology of template itself. IMAvatar [45] solves this problem by optimizing an implicit signed distance field upon a mesh template. These methods model the motion by linear blend skinning of template, which leads to slow training and rendering speed. NeRF-based methods [11, 12, 16, 25] achieve photo-realistic and view-consistent rendering and

are not limited by the topology and coarse expressiveness of face templates. NeRFace [11] proposes to use an expression conditioned dynamic NeRF to model a head avatar and generates photo-realistic portrait images. However, training a NeRF model often takes hours or even days. The concurrent work on NeRFBlendShape [12] reduces the training time consumption to 20 minutes by introducing the multi-level voxel field representation with multi-resolution hash tables storing features. However, the coupling of motion and appearance still limits their efficiency.

On the other end of the spectrum, a number of recent approaches [27, 32, 34] propose to use explicit voxel data structures to represent a static NeRF scene, and their experiments prove that this representation plays a dramatic role in accelerating NeRF training. Meanwhile, some other methods extend explicit voxel representations to dynamic NeRF. The traditional D-NeRF [31] represents dynamic scenes by decoupling deformation field and canonical appearance. TineuVox [10] further accelerates D-NeRF by introducing a voxel grid to model the canonical component. But the deformation field of TineuVox is still built by a deep MLP, which still takes much time to converge. Based on these methods, it is still not trivial to accelerate dynamic NeRF training in head avatar reconstruction. Specifically, as the human face motions are more complex, a deep MLP to model the complex motions is usually inevitable. As a result, it takes much longer time for the MLP to converge during model training.

To overcome the above challenges, we propose ManVatar, a fast 3D head avatar reconstruction method using Motion-Aware Neural Voxels. Our approach can achieve 5-minute 3D head avatar reconstruction which means the training can be completed immediately just after collecting the training data. The key idea consists of two points: First, inspired by previous dynamic NeRF reconstruction approaches, we for the first time decouple the complex expression-related motion from the canonical appearance in NeRF-based head avatar reconstruction. We further adopt an efficient voxel-based representation, instead of deep MLPs, for both the appearance and the motion field. Note that NeRFBlendshape [12] only introduces the voxel-based representation but does not decouple the motion and appearance. Second, we utilize the prior information provided by 3DMM expression bases to model the expression-related motion, so that we can reconstruct detailed motion using only a tiny MLP and voxel grids. Specifically, we use an expression-conditioned neural voxel grid to describe the motion field, and further decompose this neural voxel grid as a linearly weighted concatenation of multiple neural voxel grids bases. The number of the voxel grids bases is the same as the dimension of 3DMM expression bases and the weights of the concatenation is exactly the 3DMM expression coefficients. We define this representation as motion-aware neural voxels. For the canonical appearance, we simply use a single neural voxel grid to represent the static appearance of the avatar. Overall, our method can reconstruct a photo-realistic 3D head avatar in 5 minutes implemented by pure Pytorch code. Both quantitative and qualitative experiments demonstrate the superiority of our method in training speed while preserving a comparable rendering result. To conclude, in this paper, we present:

- The first NeRF-based head avatar reconstruction method that decouples complex expression-conditioned motion from static appearance.
- Motion-Aware Neural Voxels representation for modeling expression-related motion.
- 5 minutes 3D head avatar reconstruction using a monocular portrait video.

2. Related Works

Portrait Video Synthesis. In the past period of time, a large number of portrait video synthesis methods have been proposed. The earliest approaches [22, 35–37, 39] track to reconstruct textured face mesh with expressions and then re-render it to synthesis portrait images with desired expressions. Areas such as mouth, hair and background need to be blended later. Such methods heavily rely on accurate tracking methods, high-quality reconstructed mesh models, and photo-realistic rendering techniques, etc, which severely limits the broad applications. Warping-based methods [1, 13, 29, 33, 40, 42] model the 2D motion flow map as a common representation for expression transfer, but fail to deal with extreme head poses or expressions. Some recent approaches [8, 38] lift the 2D motion flow map into 3D space to overcome such artifacts to a certain extent. Based on the widely-used facial parametric models [14, 23], template-based methods [6, 7, 20, 21] that combine face template and neural rendering has gradually become the mainstream. These methods first render coarse images using face templates with controllable expression and pose coefficients, and then utilize convolutional neural networks to generate mouths, hair and rich details to synthesize high-quality portrait images. Other similar methods [5, 43] propose to use semantic maps or landmarks for coarse-level representation instead of face models.

Monocular 3D Head Avatar Reconstruction. It has always been a challenging task to reconstruct 3D full head avatar from monocular video. On the one hand, it's difficult to accurately capture the complex expression-related dynamic motion with current tracking methods. On the other hand, the detailed geometry of hair and wrinkles cannot be fully recovered. Early methods [2, 3, 17, 18, 28] use blendshape-based templates to fit portraits in input videos to model human heads. For parts such as eyeball and mouth

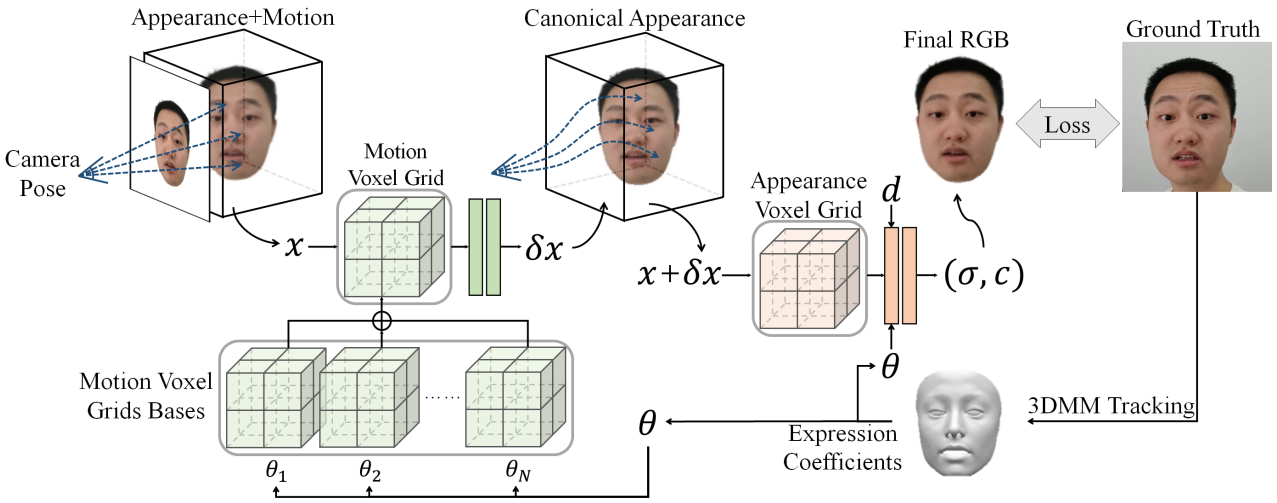

Figure 2. Overview. Given a portrait video, we first track the expression and head pose using a 3DMM template. After the pre-processing, given expression coefficients, we use motion voxel grid bases to represent motions caused by each expression basis and sum them weighted as an entire motion voxel grid. The entire motion voxel grid and the following 2-layer MLP will then transfer an input point x to $x + \delta x$ by adding all expression-related deformations. Finally, we will query point $x + \delta x$ in the appearance voxel grid and generate a final portrait image using volumetric rendering.

interior that are difficult to model by the templates, these methods require additional post-processing or leverage neural networks to learn geometry or textures [15, 19]. Inspired by the 3D scene reconstruction method IDR [41] based on implicit representation [30], IMAvatar [45] proposes to optimize an implicit signed distance field and a color field to represent the head model based on the FLAME model, and then render images via ray tracing. As NeRF [26] representation shows strong ability to synthesis high-fidelity photo-realistic images, NeRFace [11] proposes to use a MLP-based dynamic NeRF, which is conditioned by 3DMM expression coefficients to model a head avatar. Furthermore, NeRFBlendShape [12] use a voxel grid with multi-resolution hash embedding to replace the MLP, and model dynamic NeRF by linear combination of multiple NeRF basis one-to-one corresponding to semantic blendshape coefficients. In the field of the audio-driven avatar, latest methods [16, 25] also leverage dynamic NeRF as the representation of head avatars. NeRF representation is used in our method as well, but compared with the previous methods, we make great progress in accelerating training speed by both using explicit representation and the decomposition of dynamic motion and static appearance.

Training Acceleration for NeRF. As vanilla NeRF [26] usually takes hours or even days to complete the training of a static scene, a lot of works focus on speeding up NeRF training process. DVGO [34] proposes to accelerate NeRF training by directly replacing most MLPs with voxel grids, which significantly reduces the time required for training convergence. Plenoxels [32] goes a step further and proposes to only use a sparse grid with density and spherical harmonic coefficients at each voxel without any neural network. TensoRF [4] proposes to decompose the voxel grid into sum of vector-matrix outer products. It reduces the size of the model while ensuring training efficiency and rendering quality. Instant-NGP [27] introduces multi-resolution hash embedding for the voxel grid structure. Combined with customized CUDA implementation, they enable extremely fast NeRF training. These methods mainly focus on static scenes reconstruction. For dynamic scenes, TiNeuVox [10] proposes to replace the MLPs for canonical scene in D-NeRF [31] with an explicit voxel grid. Different from [10] which still represent the time-dependent deformation field with a MLP, our method propose to utilize the explicit data structure for both the static content and the expression-dependent deformation field.

3. Overview

As shown in Fig. 2, our approach reconstructs a head avatar model from a monocular portrait video. During the data preprocessing stage, we first perform 3DMM tracking to obtain the expression coefficients and head poses from the portrait video, which is used as input to our method. At the training stage of expression-related motion learning, we first establish Motion Voxel Grid (MVG) and a 2-layer MLP to represent the expression-related 3D motion. The bases of MVG share the same dimension as the 3DMM expression bases. And the same expression coefficients can be used as weights to concatenate the MVG bases. Then, at the training stage of canonical appearance reconstruction, an appearance voxel grid and another 2-layer MLP are

introduced to represent the basic appearance of the target video. Specifically, given an input coordinate x, we then transfer it to $x + \delta x$ by MVG with the following MLP and query point $x + \delta x$ in the appearance voxel grid to get its density and color. Note that the expression coefficients and view directions are also fed into the following MLP to generate expression-related and view-dependent changes. Finally, after volumetric rendering, a consistency constraint is imposed between the rendered portrait images and the corresponding ground-truth frames. During the inference stage, we can generate photo-realistic portrait images given only expression coefficients and head poses.

4. Method

ManVatar can generate a fast 3D head avatar by decoupling complex expression motion from the canonical appearance and voxel-based representation. In this section, we will introduce our voxel-based representation and the training process.

4.1. Representation

Generally, previous methods [11, 12, 16, 25] formulate NeRF-based head avatar as a expression-dependent NeRF model:

$$(c, \sigma) = \Phi(x, d, \theta). \tag{1}$$

Given a query point x with view direction d and expression coefficients θ, the color and the density denoted by c and σ respectively are computed for volumetric rendering [26].

However, their practice of mixing the dynamics expression motion with the appearance of the human head brings obstacles to the fast convergence of avatar learning. In contrast, our method decouples the NeRF-based head avatar into the canonical appearance and expression motion based on the physical assumption that a face geometry with expressions can be deformed from the neutral face geometry. Specifically, we define $\Phi(\cdot)$ as expression-independent NeRF and additionally introduce an expression-dependent deformation $\Omega(\cdot)$, which can be formulated as:

$$(c, \sigma) = \Phi(x + \delta x, d), \tag{2}$$

$$\text{with} \quad \delta x = \Omega(x, \theta). \tag{3}$$

Note that $\Phi(\cdot)$ represents a static NeRF and does not contain the expression θ in its parameters. Since the degrees of freedom of the parameters are greatly reduced, the training speed can be significantly improved as shown in our experiments.

To further improve training efficiency, we enhance our representation with an explicit voxel-based strategy. However, using a single voxel grid is infeasible to model the dynamic expression motion. Previous methods [10, 31] utilize deep MLPs to alleviate this issue but lead to a quite time-consuming convergence process. In our solution, we take the expression prior from 3DMM and use its expression coefficients to condition multiple voxel grids, which is the key to efficient and effective avatar creation.

Expression Motion. In the traditional PCA-based 3DMM [14], the face model can be deformed through a linear combination of its expression PCA bases. To leverage the expression prior of 3DMM and the expressive power of neural voxels, we use the expression coefficients as the weights to concatenate multiple Motion Voxel Grid (MVG) bases into an entire MVG to represent the expression motion. Specifically, each dimension of MVG bases uses the same expression coefficient $\theta \in \mathbb{R}^N$ as the corresponding 3DMM expression basis, where N denotes the dimension of 3DMM expression bases. From another point of view, MVG bases can be considered as "neural" expression bases for head motion, which is able to present more detailed motion in a wider range (including the ears and hair regions). The process can be formulated as:

$$\boldsymbol{V}_d(\theta) = \theta^1 \boldsymbol{V}_d^1 \oplus \theta^2 \boldsymbol{V}_d^2 \oplus \cdots \oplus \theta^N \boldsymbol{V}_d^N, \tag{4}$$

where $\{\boldsymbol{V}_d^1, \boldsymbol{V}_d^2, \cdots, \boldsymbol{V}_d^N\} \in \mathbb{R}^{N \times C_d \times L_d \times L_d \times L_d}$ denotes the MVG bases in the latent space. $\boldsymbol{V}_d(\theta) \in \mathbb{R}^{NC_d \times L_d \times L_d \times L_d}$ denotes the MVG and $\theta = \{\theta^1, \theta^2, \cdots, \theta^N\} \in \mathbb{R}^N$ denotes the expression coefficients derived from 3DMM. C_d denotes the channel number of voxel features in each voxel grid base.

Finally, for a query point x, we adopt multi-distance interpolation [10] to sample the corresponding feature vector v_d and feed it into a 2-layer MLP f_d to predict the motion offsets of input points. This process can be formulated as:

$$v_d = T_K(x, \boldsymbol{V}_d(\theta)) \tag{5}$$

$$\delta x = f_d(v_d), \tag{6}$$

where $T_K(\cdot)$ denotes K-distance interpolation [10] and δx denotes the motion offset of a query point x. We have described in detail how we formulate the expression-dependent deformation $\Omega(\cdot)$ in Eq.3 using Motion-Aware Neural Voxels representation.

Canonical Appearance. Since we model the canonical appearance $\Phi(\cdot)$ as a static NeRF, we just directly represent the canonical appearance field with a single appearance voxel grid $\boldsymbol{V}_a \in \mathbb{R}^{C_a \times L_a \times L_a \times L_a}$ and a two-layer MLP F_a, where C_a denotes the channel number of voxel features and L_a denotes the resolution of the appearance voxel grid. The explicit appearance voxel grid $\boldsymbol{V}_a$ contains information for appearance in its per-voxel features, while the two-layer MLP F_a translates the sampled features to the color and the density.

Specifically, for each canonical query point x, its corresponding feature vector v_a is first sampled from the appearance voxel grid $\boldsymbol{V}_a$ by multi-distance interpolation [10]. Then, we feed this feature vector together along with the

Figure 3. We visualize our learned canonical static appearance and expression motion.

view direction d into the 2-layer MLP F_a to predict the color and the density:

$$(c(x), \sigma(x)) = F_a(\gamma(v_a), \gamma(d), \theta), \tag{7}$$

where $c(\cdot)$ and $\sigma(\cdot)$ denote the RGB value and the density value of query point x respectively, $\gamma(\cdot)$ is the positional encoding [26] mapping the feature vector v_a and the view direction d to their periodic formulation. As it is difficult to model dynamic details of the human face such as wrinkles given only motion information, we additionally feed the expression coefficients θ to F_a to describe dynamic appearance.

4.2. Training

For more effective training, we first remove the background [24], the neck, and the body part of the human body [46] from each video frame during the data preprocessing phase. Then we detect face landmarks [9] and obtain expression coefficients and head poses by tracking 3DMM models on each frame. During training, we optimize the voxel grids and MLPs with direct photometric supervision. Moreover, we empirically find that the motion-aware neural voxels may also learn to model the information of canonical appearances such as global constant offset and non-zero offsets in static areas without regularization. Hence, we add a regularization term to punish all non-zero offsets of sampled points. Meanwhile, as all the offsets in the motion voxel grid are pushed to zeros, the canonical appearance tends to present a neutral expression. The total loss function can be formulated as:

$$\mathcal{L} = \sum_{r \in \mathcal{R}} ||I(r) - I_{gt}(r)||_1 + \lambda \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}(r)} ||\delta(r(t))||_2, \tag{8}$$

where $\mathcal{R}$ denotes the sampled rays in a batch. I_{gt} denotes the preprocessed ground-truth image, $\mathcal{T}(r)$ denotes the distances set of the sampled points on rays r, and λ denotes the weight of the regular term.

5. Experiments

5.1. Implementation Details

We implement our whole framework with pure PyTorch. For expression coefficients, we directly use the first 32 expression bases of Basel Face Model [14]. For appearance voxel grid of canonical appearance, we set the channel number of feature as 4 and resolution as 32^3. For expression motion, the number of MVG bases is set as 32 (equal to the number of expression coefficients). The channel number of the features is set as 2 and the resolution of each grid is set as 64^3. All the MLPs in our framework are 2-layer with 64 neurons for each hidden layer. We uniformly set frequency number as 4 for positional encoding. For multi-distance interpolation, we set K as 3. For loss function, we set $\lambda = 0.01$.

During optimization, we use Adam optimizer. The initial learning rate is set as 1×10^{-2} for MVG bases and appearance voxel grid, and 1×10^{-3} for all the tiny MLPs. At the 500 and 2000 iterations, we reduce the learning rate by a third. For ray sampling, we sample 4096 rays and 64 points along each ray in each iteration. The batch size is set as 1. Thus, the time consumption of one iteration is 34ms on one RTX 3090 GPU. We train a model for 10000 iterations in total. For the first 6000 iterations, we use training images with 256×256 resolution and for the last 4000 iterations, we use 512×512 resolution.

We collected 8 training videos for our experiments. with 4 of them are from HDTF dataset [44], 1 from NeRFBlendShape [12]. We additionally collect 3 videos by a handhold mobile phone. Each video contains about 2000-4000 frames. Meanwhile, we collect several videos as source videos for facial reenactment task.

5.2. Comparisons on Render Quality

We conduct qualitative and quantitative comparisons on render qualities between our ManVatar and three state-of-the-art methods DVP (Deep Video Portraits) [20], IMAvatar [45], and NeRFace [11] on both self animation task and facial reenactment task. DVP does not reconstruct the full head model, but directly synthesizes 2D images through an image-to-image translation network with a controllable coarse 3DMM model as guidance. IMAvatar reconstructs an implicit signed distance field based on the FLAME model. NeRFace reconstructs a NeRF head model with 3DMM expression coefficients as condition. In our experiments, we spend enough time for training to ensure complete convergence for all the methods. Respectively, **1 day** for IMAvatar [45], **1 day** for DVP [20], **12 hours** for NeRFace [11], and **5 minutes** for ManVatar. Since the concurrent work on NeRFBlendShape [12] doesn't release their code, we cannot make comparisons with them. In the next section, we only make qualitative comparisons on training speed to their demo video.

Qualitative comparisons on self animation task are shown in Fig. 4. The results validate that ManVatar achieves the highest render quality while the training time is far less than the other methods. IMAvatar reconstructs an implicit model based on a FLAME template, yet the expressiveness

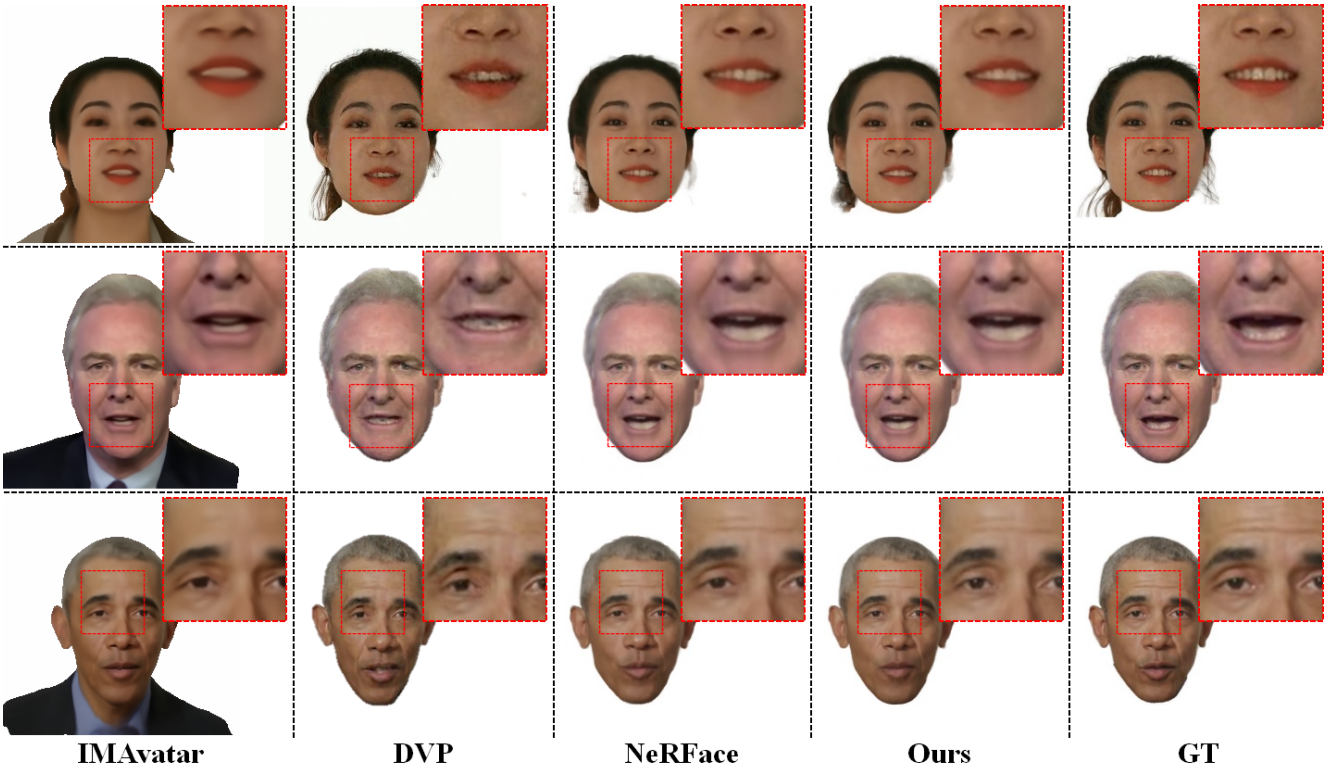

Figure 4. Qualitative comparisons between ManVatar and other three state-of-the-art methods on self re-animation task. From left to right: IMAvatar [45], DVP [20], NeRFace [11], ManVatar and Ground Truth. Our approach is able to converge and learn details within a short time.

is insufficient. Therefore, they can hardly learn person-specific expression details. DVP inherits the GAN framework and relies on a 2D convolutional network to generate images. But in many cases, the generated details are not appropriate. NeRFace's performance is comparable to our ManVatar, but the training time is much longer.

Method	MSE ↓	PSNR ↑	SSIM	LPIPS
DVP [20]	0.0047	24.2	0.89	0.072
IMavatar [45]	0.0041	24.6	0.92	0.131
NeRFace [11]	0.0015	30.2	**0.96**	**0.038**
ManVatar	**0.0014**	**30.4**	**0.96**	**0.038**

Table 1. Quantitative evaluation results of ManVatar and other three state-of-the-art methods on self animation task.

Table 1 shows the quantitative evaluation results. We evaluate on four metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structure Similarity Index (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). Our ManVatar achieves the best results in MSE and PSNR metrics, and comparable results to NeRFace on both SSIM and LPIPS metrics.

Next, we compare ManVatar with these three state-of-the-art methods on facial reenactment task. Given a source video, we obtain the corresponding camera pose and expression coefficients by template fitting, and use them to animate the pre-trained avatars. Qualitative results are shown in Fig. 5. In cases where the expression from the source video is out of the distribution in the training data, NeRFace might produce floating artifacts. Benefiting from our decoupling of the dynamic and the static elements, ManVatar maintains better stability.

5.3. Comparisons on Training Speed

We qualitatively compare the training speed of our ManVatar and two other NeRF-based methods: NeRFBlendShape [12] and NeRFace [11]. We train each model from scratch, and render the corresponding image at 5s, 15s, 30s, 1min, 2min. Since our model has almost reached complete convergence within the first 2 minutes, We do not visualize the render results after 2 minutes. NeRFBlendshape claims that their method takes 20 minutes to converge. In our experiments, we found that NeRFace actually only takes a few hours to converge. Note that, We use the subject in the demo video of NeRFBlendShape for comparison.

Qualitatively comparisons are shown in Fig. 8. Our model converges with very high efficiency. In the first 30 seconds of training, our model can complete most of the

Figure 5. Qualitative results of ManVatar and three other state-of-the-art methods on facial reenactment task. From left to right: DVP [20], IMAvatar [45], NeRFace [11] and ManVatar.

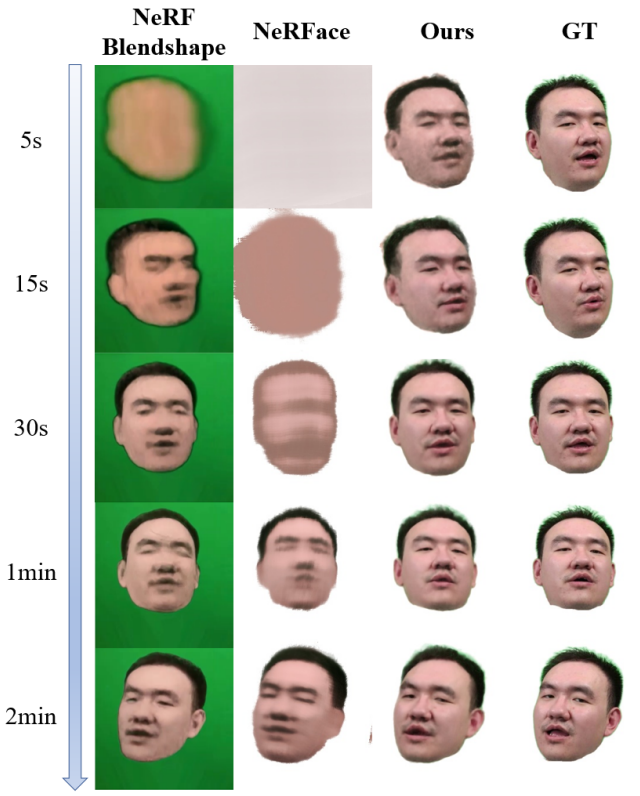

Figure 6. We make qualitative comparisons on training speed among NeRFace [11], NeRFBlendShape [12] and our ManVatar. Our model converges rapidly within the first 2 minutes.

convergence after very few iterations. At the time of 2 minutes, our model has almost converged (5 minutes for the complete convergence). In contrast, much more training time is required for the other two methods.

5.4. Ablation Study

Compared to previous methods, we introduce two major contributions in the representation for NeRF-based head avatar: decoupling expression motion from canonical appearance; modeling the expression motion by motion-aware neural voxels representation. In the following, we ablate these two components.

Without Decoupling. This baseline no longer decouples expression motion and canonical appearance, but directly uses weighted concatenation of multiple voxel grid bases to represent the whole dynamic head avatar. Given N dimension expression coefficients, we also establish N corresponding voxel grids basis but the weighted concatenation of the voxel grids directly represent the final expressive head avatar. A 2-layer percepron is followed to predict the color and the density values. In our expriments, we set the resolution of the voxel grid bases as 64^3, which is equal to resolution of the appearance voxel grid in our method.

MLP Deformation. In this baseline, we directly use a MLP to implicitly model the expression motion. Specifically, for a query point, we feed the coordinate concatenated with the expression coefficients in to a MLP with positional encoding to obtain the offset. In our expriments, we use a 4-layer percepron with 128 neurons for each hidden layer. For the canonical appearance, we still use a neural voxel grid with a 2-layer percepron.

In this section, we evaluate our full method ManVatar, NeRFace [11] and the two variant baselines above. Quantitative results of these four methods are shown in Fig. 7. Qualitative results of ManVatar and two baselines are shown in Fig. 8. As baseline without decoupling and NeRFace do not decouple the expression motion from the static appearance, their training efficiency is restricted. Note that

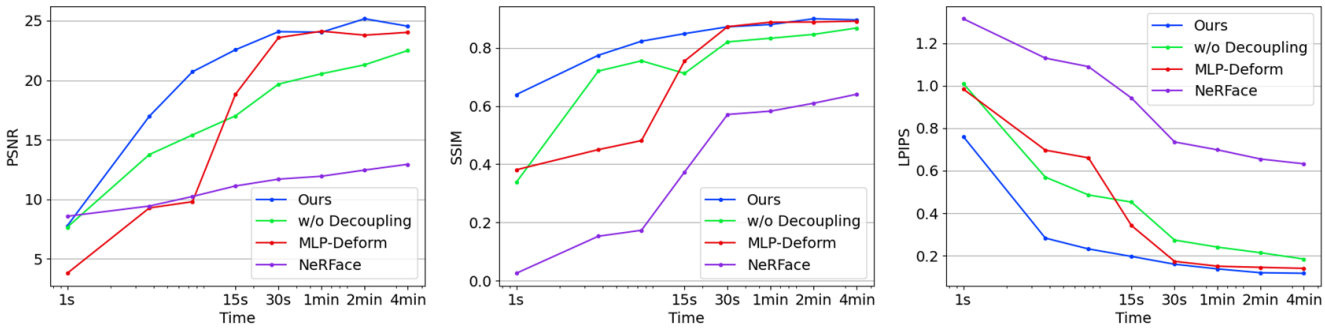

Figure 7. We quantitatively evaluate the convergence of ManVatar, baseline without decoupling, baseline using MLP to model the motion and NeRFace on self-animation task at different points in time. We evaluate on 3 metrics: PSNR, SSIM and LPIPS.

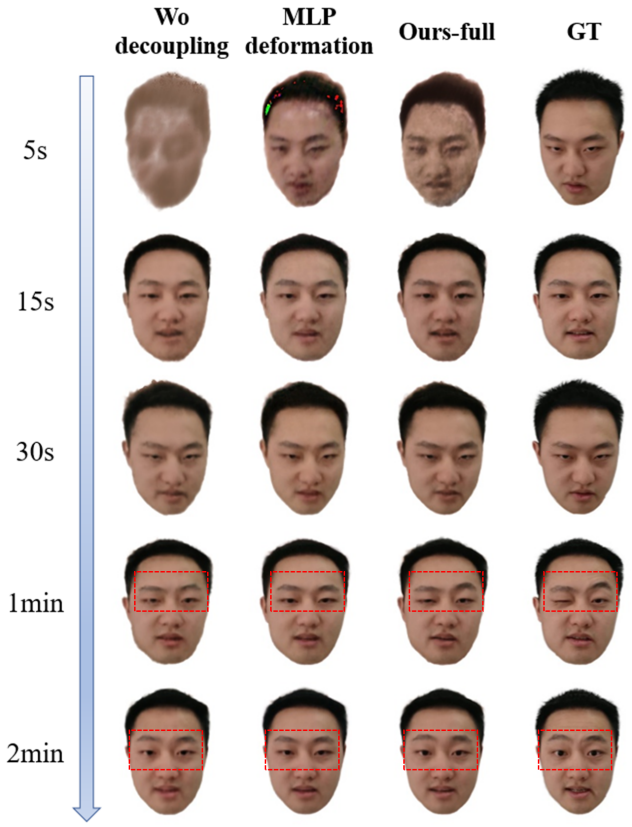

Figure 8. Qualitative results of two baselines and our ManVatar on self-animation task at different points in time. Our full method ManVatar can learn both high-quality appearance and expression motion faster.

due to the voxel-based representation, the training speed of this baseline is still significantly faster than NeRFace. MLP-deformation baseline benefits from decoupling and converges faster than the previous two methods. However, it can be observed from the images corresponding to 1min or 2min in the qualitative results that only voxel-based canonical appearance converges to the average state rapidly, while MLP-based motion converges slowly. Our ManVatar combines the advantages of both decoupling and voxel-based representation to achieve the best training efficiency.

6. Limitation

Although we accelerate the training speed of NeRF-based head avatar to almost instant, there is no significant improvement in rendering quality. We believe that this bottleneck is caused by the fitting errors in data preprocessing. Specifically, the camera poses and expression coefficients obtained by fitting a face template are not accurate enough. Thus, artifacts appear when the rendering viewpoint is away from the front view or the expression is out of the distribution of the training data. On the other hand, limited by the face template, our approach can only handle human heads, without the ability to reconstruct other regions such as neck, upper body and long hair. In the future, we will try to use more general parameterized representation to solve this problem.

7. Conclusion

In this paper, we have presented ManVatar, a fast 3D head avatar reconstruction method using motion-aware neural voxels. ManVatar have decoupled complex expression-related motion from the static appearance for the first time in head avatar reconstruction works. The proposed neural-voxel-based representation of both the canonical appearance and the motion field is able to greatly accelerate the training process. Furthermore, we have also demonstrated superiority and efficiency of the proposed motion-aware neural voxels in reconstructing complex expression-related motion. In future work, on-line creating a 3D head avatar during live capturing process might become feasible, which can greatly reduce the cost of generating digital human faces. We believe ManVatar will inspire the following facial reenactment researches, and the proposed motion-aware neural voxels could further broaden the applications of dynamic NeRF.

References

[1] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. Bringing portraits to life. *ACM Trans. Graph.*, 36(6), nov 2017. 2

[2] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *ACM Trans. Graph.*, 34(4), jul 2015. 2

[3] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.*, 35(4), jul 2016. 2

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022. 3

[5] Zhuo Chen, Chaoyue Wang, Bo Yuan, and Dacheng Tao. Puppeteergan: Arbitrary portrait animation with semantic-aware appearance transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[6] Michail Christos Doukas, Mohammad Rami Koujan, Viktoriia Sharmanska, Anastasios Roussos, and Stefanos Zafeiriou. Head2head++: Deep facial attributes re-targeting. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(1):31–43, 2021. 2

[7] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14398–14407, October 2021. 2

[8] Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. Megaportraits: One-shot megapixel neural head avatars. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 2663–2671, New York, NY, USA, 2022. Association for Computing Machinery. 2

[9] emilianavt. Openseeface. https://github.com/emilianavt/OpenSeeFace. 5

[10] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. *arxiv:2205.15285*, 2022. 2, 3, 4

[11] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Niessner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021. 1, 2, 3, 4, 5, 6, 7

[12] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *arXiv preprint arXiv:2210.06108*, 2022. 1, 2, 3, 4, 5, 6, 7

[13] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided gans for single-photo facial animation. *ACM Trans. Graph.*, 37(6), dec 2018. 2

[14] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schoenborn, and Thomas Vetter. Morphable face models - an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82, 2018. 2, 4, 5

[15] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18653–18664, June 2022. 1, 3

[16] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5784–5794, October 2021. 1, 3, 4

[17] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6), nov 2017. 2

[18] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph.*, 34(4), jul 2015. 2

[19] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *European Conference of Computer vision (ECCV)*, 2022. 1, 3

[20] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Trans. Graph.*, 37(4), jul 2018. 2, 5, 6, 7

[21] Mohammad Rami Koujan, Michail Christos Doukas, Anastasios Roussos, and Stefanos Zafeiriou. Head2head: Video-based neural head synthesis. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 16–23, 2020. 2

[22] Kai Li, Feng Xu, Jue Wang, Qionghai Dai, and Yebin Liu. A data-driven approach for facial expression synthesis in video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 57–64. IEEE, 2012. 2

[23] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6), nov 2017. 2

[24] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance, 2021. 5

[25] Xian Liu, Yinghao Xu, Qianyi Wu, Hang Zhou, Wayne Wu, and Bolei Zhou. Semantic-aware implicit neural audio-driven video portrait generation. *CoRR*, abs/2201.07786, 2022. 1, 3, 4

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 4, 5

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 2, 3

[28] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. Pagan: Real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37(6), dec 2018. 2

[29] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[31] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020. 2, 3, 4

[32] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2, 3

[33] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[34] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2, 3

[35] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6), oct 2015. 2

[36] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2395, 2016. 2

[37] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, page 426–433, New York, NY, USA, 2005. Association for Computing Machinery. 2

[38] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[39] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Trans. Graph.*, 30(4), jul 2011. 2

[40] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[41] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2492–2502. Curran Associates, Inc., 2020. 3

[42] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. *arxiv:2203.04036*, 2022. 2

[43] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[44] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3661–3670, 2021. 5

[45] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I m avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13545–13555, June 2022. 1, 3, 5, 6, 7

[46] zllrunning. face-parsing.pytorch. https://github.com/zllrunning/face-parsing.PyTorch. 5