Face2Face$^\rho$: Real-Time High-Resolution One-Shot Face Reenactment

Kewei Yang[1], Kang Chen[1](✉), Daoliang Guo[1], Song-Hai Zhang[2], Yuan-Chen Guo[2], and Weidong Zhang[1]

[1] NetEase Games AI Lab, Hangzhou, People's Republic of China
{yangkewei,ckn6763,guodaoliang,zhangweidong02}@corp.netease.com
[2] Tsinghua University, Beijing, People's Republic of China

Abstract. Existing one-shot face reenactment methods either present obvious artifacts in large pose transformations, or cannot well-preserve the identity information in the source images, or fail to meet the requirements of real-time applications due to the intensive amount of computation involved. In this paper, we introduce Face2Face$^\rho$, the first **R**eal-time **H**igh-resolution and **O**ne-shot (RHO, ρ) face reenactment framework. To achieve this goal, we designed a new 3DMM-assisted warping-based face reenactment architecture which consists of two fast and efficient sub-networks, i.e., a u-shaped rendering network to reenact faces driven by head poses and facial motion fields, and a hierarchical coarse-to-fine motion network to predict facial motion fields guided by different scales of landmark images. Compared with existing state-of-the-art works, Face2Face$^\rho$ can produce results of equal or better visual quality, yet with significantly less time and memory overhead. We also demonstrate that Face2Face$^\rho$ can achieve real-time performance for face images of 1440×1440 resolution with a desktop GPU and 256×256 resolution with a mobile CPU.

Keywords: Face reenactment · One-shot · Real-time · High-resolution

1 Introduction

Face reenactment is the task of synthesizing realistic images of a source actor, with head poses and facial expressions synchronized with a specified driving actor. Such technology has great potential for media and entertainment applications. Traditional reenactment solutions [1] typically rely on costly CG techniques to create a high-fidelity digital avatar for the source actor, and to transfer the facial movements of the driving actor onto the digitized avatar via motion capture systems. To bypass the expensive graphics pipeline, image-based methods have been proposed, which synthesize reenacted faces using image retrieval

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19778-9_4.

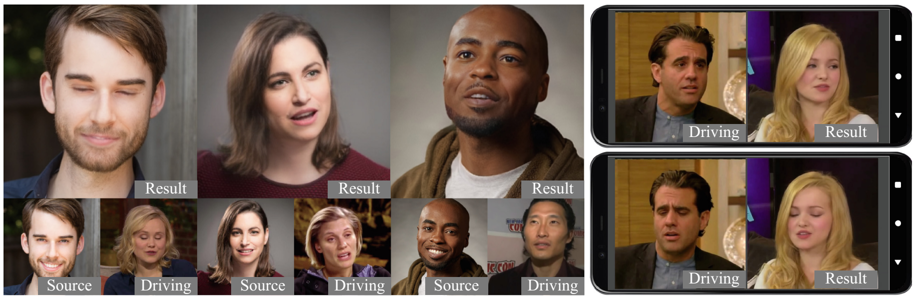

Fig. 1. One-shot face reenactment results synthesized by Face2Face$^\rho$. Left: 1440×1440 resolution images generated at 25 fps on a desktop GPU (Nvidia GeForce RTX 2080Ti); the source images are from the internet. Right: 256×256 resolution images generated at 25 fps on a mobile CPU (Qualcomm Kryo 680). Notice that the time overhead of all required computations (e.g., facial landmark detection, shape and expression regression, etc.) for each frame is accounted in the fps calculation, not just the synthesis module.

and blending algorithms [8,35,38], generative adversarial networks [16,17,19,46] or neural textures [36,37]. Nevertheless, all these methods require a considerable amount of video footage (i.e., several minutes or hours) of the source actor, which is often infeasible in practical application scenarios.

Hence, few-shot/one-shot solutions have been developed, which can animate an unseen actor with only a small number of example face images. The key idea behind these methods is to decouple an actor's facial appearance and motion information with two separate encodings, allowing the network to learn the facial appearance and motion priors from a large collection of video data in a self-supervised fashion. Based on the way how motion information is encoded in the network, classic algorithms can be divided into two categories [24,49], i.e., warping-based and direct synthesis. Warping-based methods [2,32–34,45] learn to warp the source face based on an explicitly estimated motion field, while direct synthesis methods [5,27,50,52] encode both the appearance and motion information into some low-dimensional latent representation and then synthesize the reenacted image by decoding from the corresponding latent codes. Although both are capable of producing photo-realistic reenactment results, each approach has its advantages and disadvantages. Warping-based techniques work perfectly for a small range of motion, but may easily break when large pose transformations appear. On the contrary, direct synthesis solutions have better robustness against large pose changes, but the overall fidelity of the synthesized faces tends to be lower than those produced by warping-based methods, e.g., even the identity of the source actor may not be well-preserved [7].

In this context, the main focus of later researches in one-shot face reenactment is to achieve high-fidelity reenactment results for a wide range of poses. For instance, Meshry et al. [24] relieve the identity preserving problem in direct synthesis approaches by additionally employing an encoding concerning actors' facial layouts, Wang et al. [43] incorporate 3D motion fields in the warping-based methods to improve the performance on large poses, and Doukas et al. [7] propose a hybrid framework by injecting warped appearance features into a typical direct synthesize backbone so that the two types of methods can complement each other. Although these strategies improve the overall quality of generated images, they also significantly increase the networks' computational complexity. To our best knowledge, none of the current state-of-the-art one-shot approaches can be successfully adapted to meet the requirements of real-time applications, especially on mobile devices. The only algorithm that reports having barely achieved 25 fps inference speed is [49], but given the additional computational cost for facial landmark detection, GPU-CPU copy operations, etc., it still fails to support a real-time face reenactment system. In addition, [49] sacrifices too much quality for speed, as the quality produced results are clearly below state-of-the-art.

In this paper, we tackle a new challenging problem in one-shot face reenactment, i.e., building a real-time system that can produce results of state-of-the-art quality. To achieve this goal, we introduce Face2Face$^\rho$, the first real-time high-resolution and one-shot face reenactment framework. Specifically, Face2Face$^\rho$ can reenact faces at 25 fps for images of 1440×1440 resolution on a desktop GPU and 256×256 resolution on a mobile CPU (see Fig. 1). The key idea behind this framework is that warping-based backbones have better potentials in building a light-weighted reenactment framework, since the warped feature maps already depict most parts of the target face, leaving the generator a relatively easier task (i.e., refining and in-painting) to learn. In this spirit, we present a new way to combine the warping-based and direct synthesis approaches, i.e., injecting the 3D head pose encodings into a light-weighted u-shaped warping-based backbone. Further, we introduce a novel hierarchical motion prediction network that coarse-to-finely estimates the required motion fields based on different scales of landmark images. Such design achieves over 3 times speedup in both rendering and motion field prediction without damaging the visual quality or prediction precision. Our experiments also demonstrate that Face2Face$^\rho$ has the ability to perform state-of-the-art face reenactment with significantly less time and memory overhead than existing methods.

2 Related Work

We briefly review previous few-shot/one-shot methods in this section. As mentioned before, existing methods can be literately categorized into warping-based methods, direct synthesis methods, and hybrid methods.

Warping-Based Methods. These methods represent the pose and expression transformations using explicit motion fields, and then learn to warp and synthesize the target faces base on the estimated motion fields. Early methods [45] typically conduct warping directly on the source image, which often leads to unnatural head deformation. A more recent warping-based framework is introduced by Siarohin et al. [32] which performs warping on the latent feature maps and uses relative keypoint locations in the driving images to estimate motion fields. Followup works use a first-order approximation for keypoints [33] or keyregions [34] to improve the accuracy and robustness of motion field estimation, but still follow the relative motion transfer scheme. However, relative motion transfer would lead to obvious flaws if the initial driving face and the source face are different in head poses or expressions. Therefore, 3D morphable face models [3] which can explicitly estimate pose and expression information of 2D faces are incorporated to allow absolute motion transfer [9,47,48]. Nonetheless, previous warping-based methods share a common limitation, i.e., they only work well for a limited range of head poses. The latest work [43] overcomes this limitation to some extent by lifting the 2D motion fields to the 3D space. However, expensive operators like 3D convolution also make it unsuitable for real-time or mobile applications.

Direct Synthesis Methods. Zakharov et al. [50] introduce the first direct synthesis method, which projects both the appearance and motion into latent feature space and synthesizes target faces by decoding from corresponding latent codes. This framework also demonstrates that it is possible to directly synthesize reasonable reenactment results without an explicit warp field. However, [50] uses driving faces' landmarks for computing motion encodings. As facial landmarks are person-specific, motion encodings extracted from driving faces' landmarks would also contain some identity-related information, causing noticeable identity gaps in the synthesized face. Later works try to solve this problem by eliminating the driving actors' identity information from the motion codes. Various strategies have been proposed, e.g., FReeNet [52] trains a landmark converter to adapt the motion of an arbitrary person to the target person in the latent space, LPD [5] applies pose augmentation to improve cross-person reenactment performance, DAE-GAN [51] utilizes a deforming autoencoder [31] to learn pose-invariant embedded faces, etc. The latest method [24] additionally involves an encoding for the actors facial layouts, which can further relieve the identity preservation problem in some sense. Overall, direct synthesis methods can handle a much wider range of head pose changes, however, the fidelity of the generated images is typically lower than their warping-based counterparts under the same condition, because the high-frequency details may easily be lost when projecting the source face into a low-dimensional appearance embedding space. Besides, direct synthesis backbones also tend to be slower than warping-based ones, because the motion fields provide strong priors about the motion information, while direct synthesis methods learn everything from scratch.

Hybrid Methods. Inspired by FS-VID2VID [41], hybrid methods [7,49] that combine warping and synthesis have been proposed. Bi-layer [49] combines them in a parallel fashion, i.e., a direct synthesis branch to generate low-frequency components of the face image and a warping-based branch to add high-frequency details. Such combination makes the fastest one-shot reenactment network ever (i.e., 25 fps inference speed for 256×256 resolution images on mobile GPU), however, its quality of results are clearly below state-of-the-art. Moreover, it is still not fast enough to support real-time face reenactment applications, because other required operations like facial landmark detection and GPU-CPU copy operations also cost nonignorable computational time. HeadGAN [7] combines them in another way, injecting the warped appearance features into a direct synthesis backbone. Such design achieves the best visual performance among

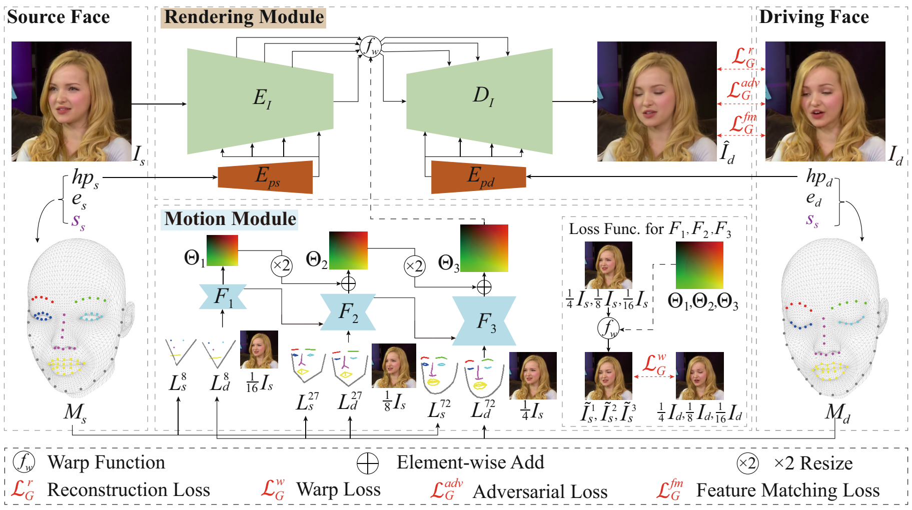

Fig. 2. Training pipeline of Face2Face$^\rho$. We first regress the 3DMM coefficients from the source image I_s and driving image I_d, and reconstruct three different scales of landmark images (i.e., L_s^n and L_d^n). The landmark images and the resized source images are fed to the motion module to coarse-to-finely predict facial motion fields, i.e., Θ_1, Θ_2 and Θ_3. Θ_3 and I_s are sent to the rendering module to generate the reenacted image $\hat{I}_d$. Head pose information is also injected into the encoder E_I and decoder D_I to improve the performances on large poses. The image sizes are adjusted for better illustration.

all existing algorithms but also makes HeadGAN [7] one of the most complex frameworks, unusable for time-critical applications. Face2Face$^\rho$ also follows the hybrid scheme, but we combine warping and synthesis in a new way. Specifically, we inject the pose encodings which are normally used in direct synthesis methods, into a warping-based generator backbone. We demonstrate that such architecture allows us to build a better one-shot face reenactment framework, which produces state-of-the-art results with significantly less time and memory overhead.

3 Method

The training pipeline of Face2Face$^\rho$ is illustrated in Fig. 2. For each pair of source and driving face images, we calculate their shape coefficients, expression coefficients, head poses, and landmark images with the help of 3DMM (Sect. 3.1), and then, the rendering module learns to synthesize the reenacted face image based on the source image, the estimated motion field, and the source/driving head pose pairs (Sect. 3.2), while the motion module learns to coarse-to-finely recover the motion fields from three different scales of landmark images (Sect. 3.3). Training and inference details are described in Sect. 3.4.

3.1 3DMM Fitting

Similar to previous methods [7,48], our method also relies on 3DMM [3] to disentangle face shape $s \in \mathbb{R}^{50}$, expression $e \in \mathbb{R}^{51}$ and head pose $hp \in \mathbb{R}^6$. We

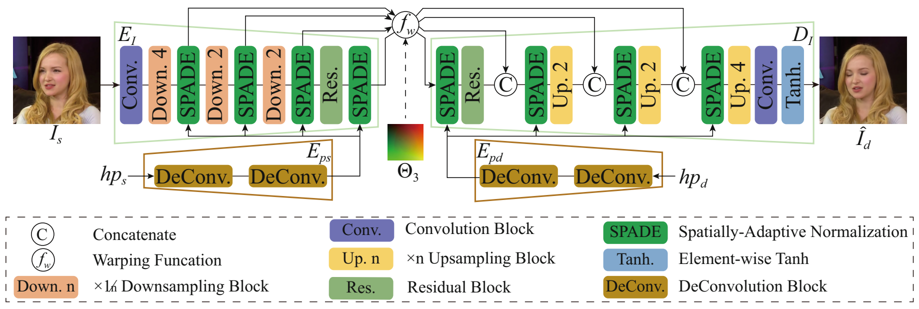

Fig. 3. Architecture of rendering module.

fit 3DMM to the source and driving image based on [44], yielding two vectors $[hp_s, e_s, s_s]$ and $[hp_d, e_d, s_d]$, where the subscripts s and d denote source and driving respectively. Each vector corresponds to a 3D face mesh in the 3DMM space. However, unlike previous methods, our framework does not require the entire face mesh to be constructed, thus, we only use the fitting 3DMM information to compute the locations of a set of pre-specified 3D keypoints (i.e., M_s and M_d) on the 3DMM face template. In our implementation, 72 keypoints were selected (as shown in Fig. 4) which are a subset of the 106-point facial landmark specification [20]. Using the standard 68-point specification [4] (i.e., two fewer points at the boundary of each eye) is also OK. It is also worth noting that, M_d is calculated using $[hp_d, e_d, s_s]$ rather than $[hp_d, e_d, s_d]$ to eliminate the interference of driving actor's identity.

3.2 Rendering Module

As mentioned before, the network structure of our rendering module is derived from a warping-based backbone. Previous warping-based methods [32–34, 43] support at most 4 times downsampling. Further downsampling layers would quickly reduce the network's performance, resulting in intensive time and memory consumption when dealing with high-solution images. To address this problem, we present a much more effective rendering network architecture that supports 16 times downsampling without damaging the quality of results. The detailed structure of the rendering module is shown in Fig. 3, which is a u-shaped encoder-decoder architecture with a warp field applied to each connected feature map between the image encoder E_I and decoder D_I. Since each downsampling operation would discard some detail, we add skip connections to compensate for the information loss, as suggested in other high-speed rendering networks [23, 54]. Such design allows the encoder to employ two additional downsampling blocks, making the network 3 times faster than previous warping-based backbones.

The major drawback of standard warping-based methods is that they can only handle a limited range of pose changes. Since using pose information to supervise the synthesis process proves to be an effective way to improve the robustness against large poses in recent direct synthesis approaches [5, 50], we

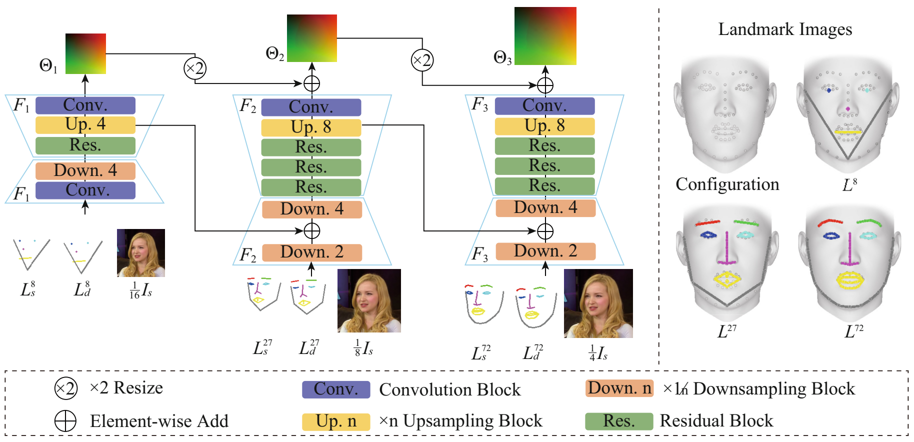

Fig. 4. Left: architecture of motion module, three scales of motion fields (i.e., $\frac{1}{16}$, $\frac{1}{8}$ and $\frac{1}{4}$ the size of the input image) are estimated from coarse to fine. Right: different scales of landmark images, top left shows the pre-configured 72 keypoints on the mesh template, and keypoints of the coarse-scale landmark images are subsets of these points.

inject pose encodings into the image encoder E_I and decoder D_I. The commonly used inputs to account for head poses are 2D facial landmarks [41,49,50], 3D facial landmarks [43] and rasterized 3DMM face meshes [7]. However, producing or processing such input information would cost non-negligible computational time. Alternatively, we directly provide the head rotations and translations estimated in the 3DMM fitting process, as pose encodings, to the rendering network. The 6-dimensional head pose vectors are reshaped into square matrices by deconvolution (i.e., E_{ps} and E_{pd}), and injected into the rendering backbone via SPADE [28]. Actually, AdaIn [11] can also be utilized for the same purpose, but the MLP layers therein tend to be more time-consuming. As no noticeable differences can be found between these two strategies in our experiments, we adopt SPADE [28] in our implementation.

3.3 Motion Module

The precision of the estimated motion field is of vital importance to a warping-based pipeline. As fitting 3DMM meshes have proven to be the best guidance information for such purpose [7,48], our motion module also follows a 3DMM-assisted fashion. However, different from previous 3DMM-assisted approaches which require the building and/or rendering of the entire face mesh, our method only needs to track the 3D positions of a small set of pre-configured 3D keypoints on the 3DMM template mesh, since a reasonable number of sparse keypoints are already sufficient to reveal the global motion field [32,33,43]. The tracked sparse 3D keypoints are then projected to the image space and transformed into a face sketch, which we refer to as a landmark image. The main advantage of such design is that it successfully avoids soft rasterizing (i.e., a very costly operation) the whole mesh, ensuring all inputs of the motion module to be quickly generated even on mobile CPUs.

The classic backbone for predicting the motion field is the single-scale hourglass network [32–34], effective but low-efficient. Inspired by recent optical flow estimation algorithms (e.g., [12]) which progressively produce high-resolution optical flow maps based on lower ones, we apply a similar coarse-to-fine strategy to motion field estimation, and successfully increase the inference speed of the motion branch by 3.5 times without sacrificing precision. As shown in Fig. 4, the motion module predicts three scales of motion fields Θ_1, Θ_2 and Θ_3, with three sub-networks F_1, F_2 and F_3 respectively. Each sub-network takes a different scale of the source and landmark images, and the feature maps after the upsampling block in each sub-network are accumulated to the subsequent sub-network to help convergence, as suggested in [42]. Notice that the number of keypoints n used when generating different scales of landmark images (i.e., L_s^n and L_d^n) also follows a coarse-to-fine scheme, where $n = 8, 27, 72$ (see right of Fig. 4).

3.4 Training and Inference

We train Face2Face$^\rho$ on the VoxCeleb dataset [26], which contains over 20k video sequences of different actors. For each video frame, we pre-compute its 3DMM coefficients as described in Sect. 3.1. Notice that the shape coefficients for the same video are forced to be identical in the 3DMM fitting process. Training pairs of source image I_s and driving image I_d are sampled from the same video. The rendering and motion module are jointly trained in an adversarial way for 380k iterations with a batch size of 6. We use an Adam [18] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively. The learning rate is 0.0002 for the first 330k iterations and then linearly decays to 0. We adopt the commonly used training loss configuration in face reenactment [5,7,9,33,41,43], and the loss for the generator G (i.e., rendering and motion module) is calculated as:

$$\mathcal{L}_G = \lambda_G^r \mathcal{L}_G^r + \lambda_G^w \mathcal{L}_G^w + \lambda_G^{adv} \mathcal{L}_G^{adv} + \lambda_G^{fm} \mathcal{L}_G^{fm}, \tag{1}$$

where $\mathcal{L}_G^r$, $\mathcal{L}_G^w$, $\mathcal{L}_G^{adv}$ and $\mathcal{L}_G^{fm}$ denote the reconstruction loss, warp loss, adversarial loss, and feature matching loss respectively. The corresponding balancing weights λ's are set to 15, 500, 1 and 1, which are determined by grid searching.

The reconstruct loss $\mathcal{L}_G^r$ ensures the output image looks similar to the ground truth. Specifically, we adopt the perceptual loss of Wang et al. [43] to measure the distance between the driving image I_d and the synthesized image $\hat{I}_d$:

$$\mathcal{L}_G^r(I_d, \hat{I}_d) = \sum_{i=1}^{N} \left\| VGG_i(I_d) - VGG_i(\hat{I}_d) \right\|_1, \tag{2}$$

where $VGG_i(\cdot)$ is the i^{th} feature layer extracted from a pre-trained VGG-19 network [15] and $N = 5$. We have tried adding an additional loss using a pre-trained VGGFace network [29], but no noticeable improvements in identity preservation can be observed. The reason is that the 3DMM already disentangles the identity (shape), expression and head pose, as discussed in [7,17,47].

Similar to [7,31,45], we also adopt a warping loss $\mathcal{L}_G^w$ to force the motion module to learn correct motion fields, which is defined as:

$$\mathcal{L}_G^w = \sum_{i=1}^{3} \left\| I_d - \widetilde{I}_s^i \right\|_1 , \tag{3}$$

where $\widetilde{I}_s^i = f_w(I_s, \Theta_\mathrm{i})$ denotes the warped source image according to the i^{th} motion field Θ_i. $f_w(\cdot,\cdot)$ is the warping function, which is implemented using a bilinear sampler. We downsample I_d to match the size of each warped image.

The generator is trained by minimizing an adversarial loss $\mathcal{L}_G^{adv}$ as well, which is calculated by using the PatchGAN discriminator D [13] and the least-square GAN objective [21]. According to [21], $\mathcal{L}_G^{adv}$ is defined as:

$$\mathcal{L}_G^{adv} = (D(\hat{I}_d, L_d^{72}) - 1)^2, \tag{4}$$

where L_d^{72} is the top-scale landmark image of the driving image. We also add a feature matching loss $\mathcal{L}_G^{fm}$ to help stabilize the training process [42]. The discriminator D is optimized by minimizing:

$$\mathcal{L}_{adv}^D = (D(I_d, L_d^{72}) - 1)^2 + D(\hat{I}_d, L_d^{72})^2. \tag{5}$$

During inference, the source actor's the 3DMM coefficients $[hp_s, e_s, s_s]$, landmark images L_s^n and encodings from E_I and E_{ps} in the rendering module can all be pre-computed and recorded in the offline phase, while the others are computed online (e.g., the driving actor's 3DMM coefficients $[hp_d, e_d, s_d]$, motion fields Θ_{1-3}, etc.). The inference speed is fast enough to support real-time applications on both PC and mobile devices. Demos are provided in the supplementary video.

4 Evaluation

In this section, we compare Face2Face$^\rho$ with the state-of-the-art works and perform an ablation study to evaluate some of our key design choices.

4.1 Comparisons

The state-of-the-art one-shot methods which we chose as the baselines are FS-VID2VID [41], Bi-layer [49], LPD [5], FOMM [33], MRAA [34] and HeadGAN [7]. For a fair comparison, all methods were implemented using PyTorch [30], and trained on the VoxCeleb dataset (i.e., randomly sampled 3M pairs for training and 3k pairs for testing). The authors' official implementations were used for Bi-layer, LPD, FOMM, MRAA and FS-VID2VID, while HeadGAN was implemented by ourselves since its source code was not publicly available. All models are trained using their recommended training configurations in the papers.

Table 1. Complexity comparisons. The inference time (Inf.) and GPU memory (Mem.) are measured on an Nvidia GeForce RTX 2080Ti GPU with FP32 and FP16 mode respectively. The "-" indicates unmeasurable, due to GPU memory limits or not supporting FP16 mode.

Method	256×256			512×512			1024×1024		
	MACs↓ $\times10^9$	Inf.(ms)↓ FP32/FP16	Mem.(GB)↓ FP32/FP16	MACs↓ $\times10^9$	Inf.(ms)↓ FP32/FP16	Mem.(GB)↓ FP32/FP16	MACs↓ $\times10^9$	Inf.(ms)↓ FP32/FP16	Mem.(GB)↓ FP32/FP16
FS-VID2VID	40.8	42.8/–	2.4/–	163.5	86.8/–	3.4/–	653.9	296.1/–	10.5/–
Bi-layer	3.8	5.6/5.4	0.9/0.9	15.3	12.6/10.0	1.1/1.0	61.2	36.2/25.8	1.9/1.6
LPD	30.3	11.8/10.1	1.3/1.0	121.2	27.9/18.5	1.7/1.3	484.9	95.6/53.9	3.2/2.6
FOMM	53.7	15.4/–	1.4/–	173.5	38.5/–	1.65/–	694.0	138.8/–	3.2/–
MRAA	53.7	15.6/–	1.4/–	173.5	39.1/–	1.65/–	694.0	145.9/–	3.2/–
HeadGAN	69.9	16.0/10.1	1.1/0.9	279.8	57.4/35.2	1.4/1.2	1119.4	224.3/147.2	3.1/2.3
Face2Face$^\rho$	**1.9**	**5.3/4.8**	**0.9/0.9**	**9.2**	**10.9/9.5**	**1.0/0.9**	**37.0**	**27.1/18.9**	**1.4/1.2**

Method	1440×1440			1536×1536			2048×2048		
	MACs↓ $\times10^9$	Inf.(ms)↓ FP32/FP16	Mem.(GB)↓ FP32/FP16	MACs↓ $\times10^9$	Inf.(ms)↓ FP32/FP16	Mem.(GB)↓ FP32/FP16	MACs↓ $\times10^9$	Inf.(ms)↓ FP32/FP16	Mem.(GB)↓ FP32/FP16
FS-VID2VID	–	–/–	–/–	–	–/–	–/–	–	–/–	–/–
Bi-layer	120.3	73.1/55.7	2.9/2.2	138.0	77.2/58.6	3.2/2.4	245.5	125.6/100.3	4.5/3.4
LPD	960.1	185.8/106.9	5.1/4.0	1091.2	198.9/114.3	5.6/4.3	1939.2	322.2/185.1	7.8/6.0
FOMM	1372.3	291.8/–	5.1/–	1561.4	310.4/–	5.7/–	2776.8	541.8/–	7.9/–
MRAA	1372.3	304.2/–	5.1/–	1561.4	327.6/–	5.8/–	2776.8	564.9/–	8.0/–
HeadGAN	2213.6	485.6/275.3	5.2/3.8	2518.6	518.4/295.6	5.8/4.2	4478.5	858.4/490.4	7.9/5.7
Face2Face$^\rho$	**71.8**	**52.1/32.5**	**1.8/1.6**	**82.8**	**56.7/34.8**	**2.0/1.8**	**146.2**	**95.8/60.2**	**2.7/2.4**

Notice that we omitted the side-by-side comparisons with some earlier works, e.g., X2Face [45] and FSTH [50], because their performances have been extensively compared and proved to be below our baselines [5,7,33,49]. Besides, a small-scale qualitative comparison with other state-of-the-art methods and also a many-shot method [36] are provided in the supplementary document.

Computational Complexity. We first evaluate the time and spatial complexity of each reenactment backbone in the inference phase. The time complexity is measured by the inference time and the number of multiply-accumulate operations (MACs), while the spatial complexity is measured by the run-time GPU memory overhead. The results shown in Table 1 demonstrate that Face2Face$^\rho$ is overwhelmingly superior in terms of computational complexity. Setting the speed requirement for real-time applications to 25 fps and considering the additional time (i.e., around 3–6 ms) required for some necessary auxiliary modules, only Bi-layer and LPD can handle 512×512 resolution images, and only Bi-layer still survives when the resolution increases to 1024×1024. In contrast, the resolution limit for Face2Face$^\rho$ is 1440×1440. We provide more high-resolution (i.e., 1440×1440) results synthesized by Face2Face$^\rho$ in the supplementary document.

Notice that there are currently no available high-resolution datasets that are suitable for the face reenactment task. So the high-resolution version of Face2Face$^\rho$ was trained using the upscaled images from VoxCeleb (i.e., 512×512), which inevitably limits the performance of Face2Face$^\rho$ on some true high-definition images. However, compared with the faked 1440×1440 results produced by simply upscaling the outputs of a 512×512 model, the results generated by a 1440×1440 model clearly have better visual quality (see supplementary document). Therefore, despite the absence of high-resolution datasets, supporting high-resolution images is still beneficial.

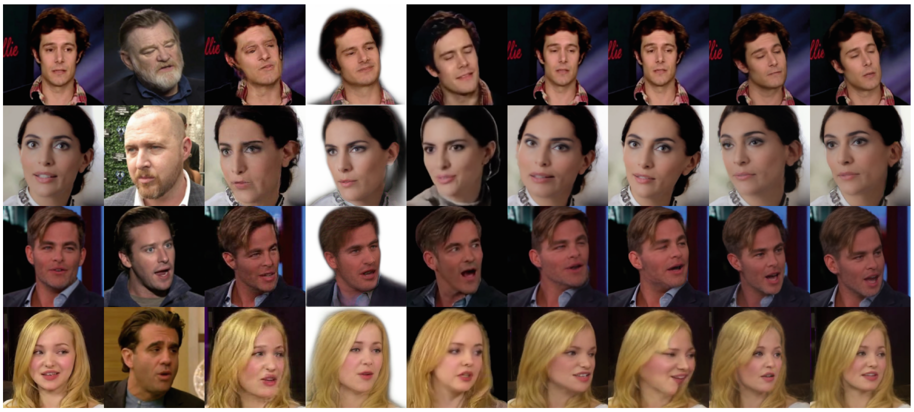

Fig. 5. Qualitative comparisons with the baselines, on the task of reenactment.

Reenactment. Next, we compare the quality of reenactment results. Since no ground truth is available for this task, we use FID [10] to measure image realism and use CSIM [6] to measure the identity gaps. We also adopt Average Rotation Distance (ARD) and Action Units Hamming distance (AU-H) used in [7] to evaluate the accuracy of the pose and expression transfer respectively. The statistics shown in Table 2 demonstrate that Face2Face$^\rho$ achieves state-of-the-art quality in each measurement. Figure 5 highlights some results where the source and the driving images have relatively large head pose differences, i.e., the head pose changes are from 30°–45° in either roll, pitch or yaw. Notice that the most recent works [7, 43] that focus on pose editing typically only allow pose changes up to around 30°, so such range of pose variance can already be considered as *large* in one-shot face reenactment. From the results, we can see that methods HeadGAN and Face2Face$^\rho$ are clearly better than their counterparts. Methods with low CSIM scores (i.e., FS-VID2VID and Bi-layer) cannot correctly preserve the source actor's identity, while methods with high FID scores (i.e., Bi-layer and LPD) tend to generate blurry results. Besides, unnatural deformations can be observed in the results of two warping-based methods (i.e., MRAA and FOMM) when head pose deformations are large (e.g., last two rows of Fig. 5). Furthermore, since they both follow a relative motion transfer scheme, which cannot correctly transfer head pose when there are large pose gaps between the initial driving face and source face (e.g., first two rows of Fig. 5). Most importantly, the results demonstrate that Bi-layer sacrifices too much quality for fast inference speed, whereas Face2Face$^\rho$ achieves even faster inference speed without compromising quality. Notice that the CSIM/FID of FOMM and MRAA in Table 2 are inconsistent with their visual quality in Fig. 5, because Fig. 5 only highlights the cases of large head pose transformations while these scores are averaged over all 3k test pairs. The dynamic results shown in the supplementary video are better in line with these statistics. Besides, the video also demonstrates that our method

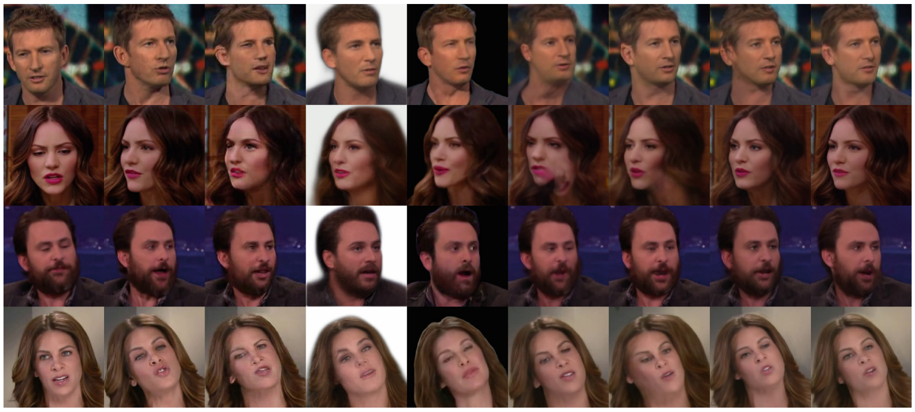

Fig. 6. Qualitative comparisons with the baselines, on the task of reconstruction.

Table 2. Quantitative comparisons with the baselines.

Method	Reenactment				Reconstruction	
	FID↓	CSIM↑	ARD↓	AU-H↓	LPIPS↓	AKD↓
FS-VID2VID	57.3	0.571	3.43	0.29	0.167	3.20
Bi-layer	101.8	0.534	3.32	0.26	0.171	2.56
LPD	81.9	0.675	4.89	0.23	0.165	2.79
FOMM	46.8	0.736	7.61	0.27	0.160	2.44
MRAA	49.4	**0.741**	7.73	0.28	0.155	2.45
HeadGAN	48.1	0.690	3.06	0.23	0.137	1.49
Face2Face$^\rho$	**44.5**	0.729	**2.82**	**0.22**	**0.123**	**1.48**

Table 3. Quantitative comparisons with the ablation study cases.

Method	Reenactment				Reconstruction	
	FID↓	CSIM↑	ARD↓	AU-H↓	LPIPS↓	AKD↓
r_1+p_0+m_0	72.5	0.599	3.13	0.25	0.169	1.64
r_2+m_0	51.7	0.692	3.83	0.26	0.164	1.74
r_0+p_1+m_0	50.9	0.702	3.53	0.26	0.158	1.68
r_0+p_2+m_0	**43.4**	0.722	2.85	0.23	0.123	1.45
r_0+p_3+m_0	45.1	0.713	**2.81**	0.24	0.127	**1.44**
r_0+p_0+m_1	45.3	0.715	3.17	0.32	0.131	1.48
Face2Face$^\rho$	44.5	**0.729**	2.82	**0.22**	**0.123**	1.48

presents good temporal coherency though we do not explicitly enforce it in the training process (i.e., as did in [41]). This is achieved by a simple but effective strategy, i.e., applying bilateral filtering on the fitted 3DMM coefficients.

Reconstruction (Self-reenactment). Finally, we compare the results in the self-reenactment task (see Fig. 6), where the source and driving images are of the same actor. Since the driving image can also be viewed as the ground truth in this scenario, the reconstruction quality can be directly measured by comparing the similarity between the driving image and the synthesized image. In our experiments, the AlexNet-based LPIPS metric [53] and the average keypoint distance (AKD) [33] were adopted for such purpose. Statistics are also shown in Table 2, which demonstrate Face2Face$^\rho$ achieves the best scores in this task.

4.2 Ablation Study

In this section, we evaluate how some key choices would affect the performance of Face2Face$^\rho$. For clarity purposes, we denote the proposed rendering module,

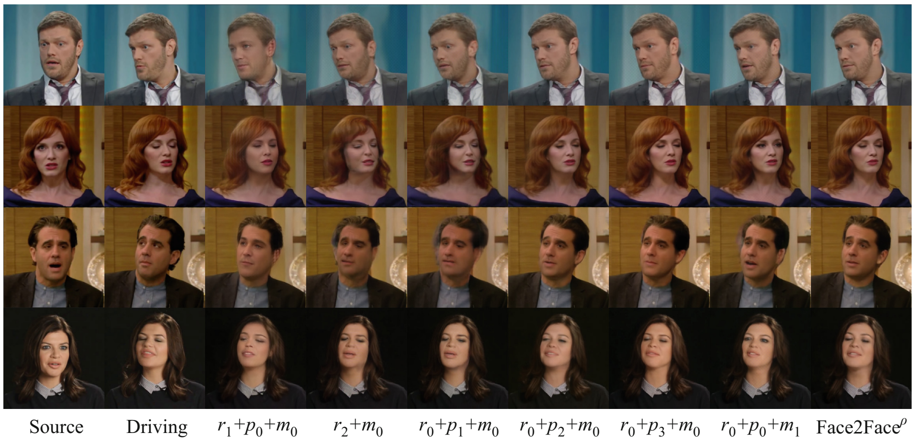

Fig. 7. Qualitative comparisons with the ablation study cases.

3DMM head pose encoding network and motion module as r_0, p_0 and m_0 respectively. Then, the alternatives to be studied can be defined as follows. 1) r_1: r_0 without skip connections, 2) r_2: r_0 without head pose injection, 3) p_1: 2D landmarks for pose encoding, 4) p_2: 3D landmarks for pose encoding, 5) p_3: depth map for pose encoding, 6) m_1: single-scale motion estimation network used in [32,33], with the number of channels reduced to match the complexity of m_0.

The results of different combinations are shown in Table 3, Fig. 7 and the supplementary video. We can see that r_1 and r_2 tend to have problems in large head pose changes, expressions generated by m_1 are often inconsistent with the driving actor (see the eyes and mouths in the 2^{nd} and 3^{rd} rows of Fig. 7)), and the performance of head pose encoding p_0 is almost equal to p_2 and p_3, despite p_2 and p_3 consume more computational time (i.e., 2–5 ms). Notice that using 2D landmarks for pose encoding p_1 actually does not help in handling large poses. Besides, we also conduct another ablation study to assess the significance of each loss term in the supplementary document.

4.3 Limitation

While our framework is generally robust, it shares some common limitations with other state-of-the-art one-shot methods. As illustrated in Fig. 8, some visible artifacts may appear in the background regions when the head transformations are large, and abnormal objects (e.g., hands or microphone) in the foreground may lead to blurry results.

5 Mobile Device Deployment

We use MNN [14] to deploy Face2Face$^\rho$ on mobile devices (see the supplementary video for a mobile demo). The 3DMM fitting and landmark image generation

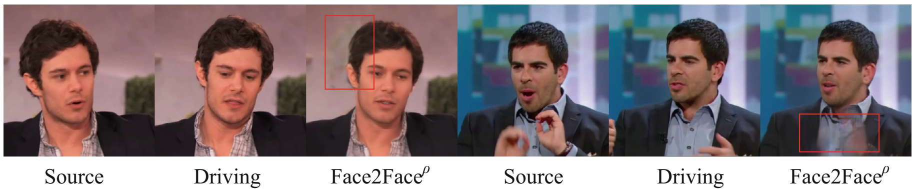

Fig. 8. Example limitations. Left: large pose modification sometimes cause visible artifacts in the background. Right: abnormal objects (e.g., hands or microphone) in the foreground may lead to blurry results.

Table 4. Running time for different resolution images on Mobile CPU (Kryo 680, 4 threads) and GPU (Adreno 660, FP16).

Step	256×256 CPU/GPU (ms)	384×384 CPU/GPU (ms)	512×512 CPU/GPU (ms)
3DMM fitting	6.9	7.0	7.1
Landmark image generation	0.1	0.1	0.2
Motion field estimation	5.6/4.1	11.2/5.7	16.3/8.9
Feature warping	1.5/0.5	4.0/0.6	8.9/0.7
Decoding	25.1/12.7	59.9/27.5	110.7/49.4
GPU-CPU communication	–/3.8	–/16.7	–/29.1
Total	39.2/28.1	82.2/57.6	143.2/95.4

run on the CPU, while all other steps run on the CPU or the GPU. The running time for each step is listed in Table 4. As can be seen, Face2Face$^\rho$ only requires a Mobile CPU to achieve real-time performance for images of 256×256 resolution. Theoretically, Face2Face$^\rho$ supports higher resolution images when running on the GPU, but the expensive GPU-CPU communication operations limit its ability.

6 Conclusion

In this paper, we present Face2Face$^\rho$, the first real-time high-resolution one-shot face reenactment framework that consists of two fast and efficient networks: a u-shaped rendering network and a hierarchical coarse-to-fine motion network. Experimental results demonstrate that Face2Face$^\rho$ achieves state-of-the-art quality, while significantly reducing the computational complexity. Specifically, it can run in real-time for generating face images of 1440×1440 resolution with a desktop GPU and 256×256 resolution with a mobile CPU.

Besides, our method supports using a single photo to generate an inauthentic video, so it has the potential to be used for illegal activities. A prime example of this is the growing misuse of DeepFakes [25,39]. However, as more and more people can access such techniques, the threat has also become more well-aware. Accordingly, image forensic techniques [22,40] have been proposed. As there is no guarantee to detect all fake images by these methods, it is strictly prohibited to use our method to generate and publish unauthorized videos.

Acknowledgements. We thank Weitong Yao for his help in presenting our supplementary video.

References

1. Alexander, O., et al.: The digital emily project: achieving a photorealistic digital actor. IEEE Comput. Graphics Appl. **30**(4), 20–31 (2010)
2. Averbuch-Elor, H., Cohen-Or, D., Kopf, J., Cohen, M.F.: Bringing portraits to life. ACM TOG **36**(6), 1–13 (2017)
3. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: SIGGRAPH, pp. 187–194 (1999)
4. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In: ICCV, pp. 1021–1030 (2017)
5. Burkov, E., Pasechnik, I., Grigorev, A., Lempitsky, V.: Neural head reenactment with latent pose descriptors. In: CVPR, pp. 13786–13795 (2020)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In: CVPR, pp. 4690–4699 (2019)
7. Doukas, M.C., Zafeiriou, S., Sharmanska, V.: HeadGAN: one-shot neural head synthesis and editing. In: ICCV, pp. 14398–14407 (2021)
8. Garrido, P., et al.: VDub: modifying face video of actors for plausible visual alignment to a dubbed audio track. Comput. Graph. Forum **34**(2), 193–204 (2015)
9. Ha, S., Kersner, M., Kim, B., Seo, S., Kim, D.: MarioNETte: few-shot face reenactment preserving identity of unseen targets. In: AAAI, pp. 10893–10900 (2020)
10. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS, pp. 6626–6637 (2017)
11. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV, pp. 1501–1510 (2017)
12. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: RIFE: real-time intermediate flow estimation for video frame interpolation. arXiv preprint arXiv:2011.06294 (2020)
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-Image translation with conditional adversarial networks. In: CVPR, pp. 1125–1134 (2017)
14. Jiang, X., et al.: MNN: a universal and efficient inference engine. In: MLSys (2020)
15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
16. Kim, H., et al.: Neural style-preserving visual dubbing. ACM TOG **38**(6), 1–13 (2019)
17. Kim, H., et al.: Deep video portraits. ACM TOG **37**(4), 1–14 (2018)
18. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
19. Koujan, M.R., Doukas, M.C., Roussos, A., Zafeiriou, S.: Head2Head: video-based neural head synthesis. In: FG, pp. 16–23 (2020)
20. Liu, Y., et al.: Grand challenge of 106-point facial landmark localization. In: ICMEW, pp. 613–616. IEEE (2019)
21. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: ICCV, pp. 2794–2802 (2017)

22. Marra, F., Gragnaniello, D., Cozzolino, D., Verdoliva, L.: Detection of GAN-generated fake images over social networks. In: MIPR, pp. 384–389. IEEE (2018)
23. Martin-Brualla, R., et al.: LookinGood: enhancing performance capture with real-time neural re-rendering. ACM TOG **37**(6), 1–14 (2018)
24. Meshry, M., Suri, S., Davis, L.S., Shrivastava, A.: Learned spatial representations for few-shot talking-head synthesis. In: ICCV, pp. 13829–13838 (2021)
25. Mirsky, Y., Lee, W.: The creation and detection of deepfakes: a survey. ACM Comput. Surv. **54**(1), 1–41 (2021)
26. Nagrani, A., Chung, J.S., Zisserman, A.: VoxCeleb: a large-scale speaker identification dataset. In: INTERSPEECH, pp. 2616–2620 (2017)
27. Nirkin, Y., Keller, Y., Hassner, T.: FSGAN: subject agnostic face swapping and reenactment. In: ICCV, pp. 7184–7193 (2019)
28. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR, pp. 2337–2346 (2019)
29. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC (2015)
30. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NIPS, pp. 8024–8035 (2019)
31. Shu, Z., Sahasrabudhe, M., Alp Güler, R., Samaras, D., Paragios, N., Kokkinos, I.: Deforming autoencoders: unsupervised disentangling of shape and appearance. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 664–680. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_40
32. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: Animating arbitrary objects via deep motion transfer. In: CVPR, pp. 2377–2386 (2019)
33. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: Advances in Neural Information Processing Systems, pp. 7135–7145 (2019)
34. Siarohin, A., Woodford, O.J., Ren, J., Chai, M., Tulyakov, S.: Motion representations for articulated animation. In: CVPR, pp. 13653–13662 (2021)
35. Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: Synthesizing Obama: learning lip sync from audio. ACM TOG **36**(4), 1–13 (2017)
36. Thies, J., Elgharib, M., Tewari, A., Theobalt, C., Nießner, M.: Neural voice puppetry: audio-driven facial reenactment. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12361, pp. 716–731. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58517-4_42
37. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: image synthesis using neural textures. ACM TOG **38**(4), 1–12 (2019)
38. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: real-time face capture and reenactment of RGB videos. In: CVPR, pp. 2387–2395 (2016)
39. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., Ortega-Garcia, J.: Deepfakes and beyond: a survey of face manipulation and fake detection. Information Fusion **64**, 131–148 (2020)
40. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: CNN-generated images are surprisingly easy to spot... for now. In: CVPR, pp. 8695–8704 (2020)
41. Wang, T., Liu, M., Tao, A., Liu, G., Catanzaro, B., Kautz, J.: Few-shot video-to-video synthesis. In: NIPS, pp. 5014–5025 (2019)
42. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: CVPR, pp. 8798–8807 (2018)

43. Wang, T.C., Mallya, A., Liu, M.Y.: One-shot free-view neural talking-head synthesis for video conferencing. In: CVPR, pp. 10039–10049 (2021)
44. Weng, Y., Cao, C., Hou, Q., Zhou, K.: Real-time facial animation on mobile devices. Graph. Models **76**(3), 172–179 (2014)
45. Wiles, O., Koepke, A.S., Zisserman, A.: X2Face: a network for controlling face generation using images, audio, and pose codes. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 690–706. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_41
46. Wu, W., Zhang, Y., Li, C., Qian, C., Loy, C.C.: ReenactGAN: learning to reenact faces via boundary transfer. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 622–638. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_37
47. Yao, G., et al.: One-shot face reenactment using appearance adaptive normalization. In: AAAI, pp. 3172–3180 (2021)
48. Yao, G., Yuan, Y., Shao, T., Zhou, K.: Mesh guided one-shot face reenactment using graph convolutional networks. In: ACM MM, pp. 1773–1781 (2020)
49. Zakharov, E., Ivakhnenko, A., Shysheya, A., Lempitsky, V.: Fast bi-layer neural synthesis of one-shot realistic head avatars. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12357, pp. 524–540. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58610-2_31
50. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: ICCV, pp. 9459–9468 (2019)
51. Zeng, X., Pan, Y., Wang, M., Zhang, J., Liu, Y.: Realistic face reenactment via self-supervised disentangling of identity and pose. In: AAAI, pp. 12757–12764 (2020)
52. Zhang, J., et al.: FReeNet: multi-identity face reenactment. In: CVPR, pp. 5326–5335 (2020)
53. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR, pp. 586–595 (2018)
54. Zhang, R., et al.: Real-time user-guided image colorization with learned deep priors. ACM TOG **36**(4), 1–11 (2017)