IMAGE ANIMATION WITH REFINED MASKING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a novel approach for image-animation of a source image by a driving video, both depicting the same type of object. We do not assume the existence of pose models and our method is able to animate arbitrary objects without knowledge of the object's structure. Furthermore, both the driving video and the source image are only seen during test-time. Our method is based on a shared mask generator, which separates the foreground object from its background, and captures the object's general pose and shape. A mask-refinement module then replaces, in the mask extracted from the driver image, the identity of the driver with the identity of the source. Conditioned on the source image, the transformed mask is then decoded by a multi-scale generator that renders a realistic image, in which the content of the source frame is animated by the pose in the driving video. Our method is shown to greatly outperform the state of the art methods on multiple benchmarks. Our code and samples are attached as supplementary.

1 INTRODUCTION

The ability to reanimate a still image based on a driving video has been extensively studied in recent years. The developed methods achieve an increased degree of accuracy in both maintaining the source identity, as extracted from the source frame, and in replicating the motion pattern of the driver's frame. In addition, the recent methods also show good generalization to unseen identities and are relatively robust and have fewer artifacts than the older methods. The relative ease in which these methods can be applied out-of-the-box has led to their adoption in various visual effects.

Interestingly, some of the most striking results have been obtained with methods that are model-free, i.e., that do not rely, for example, on post-extraction models. This indicates that such methods can convincingly disentangle shape and identity from motion.

There are, however, a few aspects in which such methods are still wanting. First, the generated videos are not without noticeable artifacts. Second, some of the identity from the source image is lost and replaced by identity elements from the person in the driving video. For example, the body shape mimics that of the person in the driving frames. Third, the animation of the generated video does not always match the motion in the driver video.

Here, we propose a method that is preferable to the existing work in terms of motion accuracy, identity and background preservation, and the quality of the generated video. Our method relies on a mask-based representation of the driving pose and on explicit conditioning on the source foreground mask. Both masks (source and driver) are extracted by the same network. The driver mask goes through an additional refinement stage that acts to replace the identity information in the mask.

The reliance on masks has many advantages. First, it eliminates many of the identity cues from the driving video. Second, it explicitly models the region that needs to be replaced in the source image. Third, it is common to both source and target, thus allowing, with proper augmentation, to train only on driving videos. Fourth, it captures a detailed description of the object's pose and shape.

To summarize, our contributions are: (i) an image animation method that is based on applying a masking process to both the source image and the driving video, (ii) the method generalizes to unseen identities of the same type and is able to animate arbitrary objects, (iii) the mask generator separates the foreground object from its background and captures, in a generic way, the fine details of the object's pose and shape, (iv) conditioning the mask of the driving frame on the source frame and its mask, in order to remove the identity elements of the driving frames and introduce those of

the source frame, and (v) a comprehensive evaluation of several different applications of our method, which show a sizable improvement over the current state of the art in image animation.

2 Related Work

Much of the work on image animation relies on prior information on the animated object, in the form of explicit modeling of the the object's structure. For example, Zakharov et al. (2019); Zakharov et al. (2020) animate a source image using facial landmarks and Ren et al. (2020) developed a human-pose-guided image generator. However, in many applications, an explicit model is not available. Our method is model-free and able to animate arbitrary objects.

There are many model-free contributions in the field of image to image translation, in which an input image from one domain is mapped to an analog image from another domain. Isola et al. (2016) learn a map between two domains using a conditional GAN. Wang et al. (2018b) developed a multi-scale GAN that generates high-resolution images from semantic label maps. Huang et al. (2018) encodes images of both domains into a shared content space and a domain-specific style space. Content code of one domain is combined with the style code of the other domain, and then the image is generated using a domain specific decoder. For this class of methods, the model is not able to generalize to other unseen domains of the same category without retraining. In contrast, for a given type of model (e.g. human faces), our method is trained once, and able to generalize to unseen domains of the same type (e.g. the source and driving faces can be of any identity).

More related to our method, Wiles et al. (2018) assume a reference frame for each video, and learn a dense motion field that maps pixels from a source frame to its reference frame, and another mapping from the reference frame to the driver's frame. Siarohin et al. (2019a) extract landmarks for driving and source images of arbitrary objects, and generate motion heatmaps from the key-points displacements. The heatmaps and the source image are then processed in order to generate the final prediction. A follow-up work by Siarohin et al. (2019b) extracts a first order motion-representation, consisting of sparse key-points and local affine transformations, with respect to a reference frame. The motion representation is then processed to generate a dense motion field, from the driver's frame to the source's, and occlusion maps to mask out regions that should be impainted by the generator. Our method does not assume a reference frame, and instead of key-points, we generate objects masks, which are more informative regarding pose and shape.

Other methods, including Lorenz et al. (2019); Dundar et al. (2020) learn a part-based disentangled representation of shape and appearance, and try to ensure that local changes in appearance and shape remain local, and do not affect the overall representation.

When a source video is available, video to video translation methods may be used. (Chan et al., 2018; Kim et al., 2018) address the task of motion transfer, by utilizing the rich appearance and pose information available in the source video. Such methods learn a mapping between two domains and are able to generate realistic results, where the source video is animated by the driver video. These methods requires a large number of source frames at train time, and require a long training process for every target subject. In contrast, our model is able to animate a single source image, which is unseen during training, and employs a driving video with another novel person.

3 Method

The method consists of four autoencoders: the mask generator m, the mask refinement network r, the low resolution frame generator ℓ, and the high resolution frame generator h. The four networks transform an input source frame $\boldsymbol{s}$ and a driving frame $\boldsymbol{d}$ into the generated high resolution frame $\boldsymbol{f}$. This is done for each driving frame separately through the following process, as depicted in figure 1:

$$\boldsymbol{m}_s = m(\boldsymbol{s}) \tag{1}$$
$$\boldsymbol{m}_d = r(\mathbf{D}(\boldsymbol{s}), \boldsymbol{m}_s, \mathbf{P}_{\text{test}}(m(\boldsymbol{d}))) \tag{2}$$
$$\boldsymbol{c} = \ell(\mathbf{D}(\boldsymbol{s}), \boldsymbol{m}_s, \boldsymbol{m}_d) \tag{3}$$
$$\boldsymbol{f} = h(\boldsymbol{s}, \mathbf{U}(\boldsymbol{m}_s), \mathbf{U}(\boldsymbol{m}_d), \boldsymbol{c})\,, \tag{4}$$

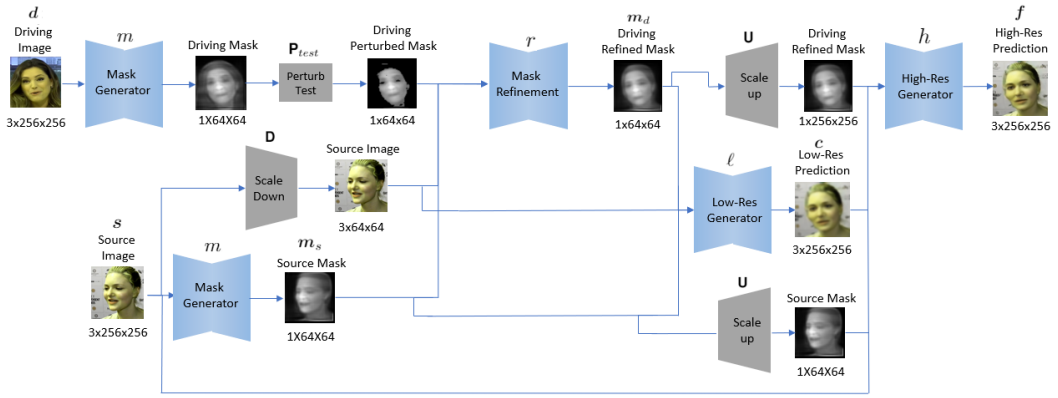

Figure 1: Overview of our method at test time. Source and driving masks $\boldsymbol{m}_s$ and $m(\boldsymbol{d})$ are generated using the mask generator m. The identity-perturbation operator $\mathbf{P}_{\text{test}}$ is then applied to the driver's mask, and along with a scaled-down version of the source's image $\mathbf{D}(\boldsymbol{s})$ and the source's mask $\boldsymbol{m}_s$, they are fed into the mask refinement network r, in order to generate the driver's refined mask $\boldsymbol{m}_d$. Next, the refined mask $\boldsymbol{m}_d$, the source's mask $\boldsymbol{m}_s$ and the scaled-down source's image $\mathbf{D}(\boldsymbol{s})$ are fed into the low-res generator ℓ, which generate the initial prediction $\boldsymbol{c}$. Finally, the scaled-up refined mask $\mathbf{U}(\boldsymbol{m}_d)$, the source image $\boldsymbol{s}$, the initial prediction $\boldsymbol{c}$ and the scaled-up source's mask $\mathbf{U}(\boldsymbol{m}_s)$ are fed into the high-res generator h, in order to generate the final prediction $\boldsymbol{f}$.

where $\boldsymbol{m}_s$ and $\boldsymbol{m}_d$ are the source mask and the driving refined mask respectively, $\mathbf{P}_{\text{test}}$ is an identity-perturbation operator applied at test time, which sets to zero pixels that are smaller than a threshold ρ, and scales-down the mask by 25%, $\boldsymbol{c}$ is the coarse (low resolution) generated frame, and $\mathbf{D}$ ($\mathbf{U}$) is a downscale (upscale) operator, implemented using a bi-linear interpolation, that transforms an image of resolution 256×256 to an image of resolution 64×64 (or vice versa). For each driver's mask, we set the threshold ρ in $\mathbf{P}_{\text{test}}$ to be the median pixel value.

The goal of this process is to generate a frame $\boldsymbol{f}$ that contains the foreground and background of the source frame $\boldsymbol{s}$, such that the pose of the foreground object in $\boldsymbol{s}$ is modified to match that of the driver frame $\boldsymbol{d}$.

The generated frame is being synthesized in a hierarchical process in which the coarse frame $\boldsymbol{c}$ is first generated using ℓ and is then refined by the network h. Both generators (ℓ and h) utilize the mask $\boldsymbol{m}_s$ to attend the foreground and background objects in the source frame and to infer the occluded regions that need to be generated.

The driver's mask $\boldsymbol{m}_d$ is the only conditioning on the frame generation process that stems from the driver's frame $\boldsymbol{d}$. It, therefore, needs to encode the pose of the foreground object in the driving frame. However, has to be done in a way that is invariant to the identity of the foreground object. For example, when reanimating a person based on a driver video of another person, a pose of a person should be given, while discarding the body shape information. Otherwise, the generated frame could have the appearance of the foreground and a body shape that mixes that of the person in the source frame and that of the person in the driving frame.

The roles of the identity perturbation operator $\mathbf{P}_{\text{test}}$, and of the refinement network r, is to remove the foreground identity of the driver's frame from the mask, and install the identity of the foreground object of the source frame. At test time, the source and driving frames are of different identities and the inputs to the network ℓ are the source frame, the source mask and the refined mask of the driver frame. Finally, the coarse generated frame is enhanced by the network h. The exact architecture of the networks is given in appendix A.

3.1 Training

The training is being performed using source videos only, i.e., the driving- and source-frame are taken from the same video. The underlying reason is that for the type of supervised loss terms we

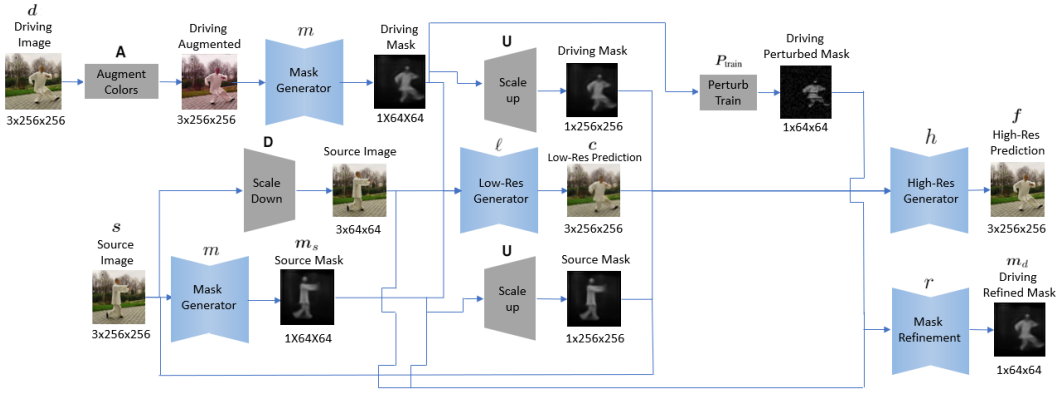

Figure 2: The method at train time. The driving image $\boldsymbol{d}$ is augmented using $\mathbf{A}$. We generate the driver's refined mask $\boldsymbol{m}_d$ as in test time, except that the operator $\mathbf{P}_{\text{train}}$ is applied instead of $\mathbf{P}_{\text{test}}$. Instead of getting the driver's refined mask $\boldsymbol{m}_d$ as in test time, the low-res and high-res generators ℓ and h are using the driver's mask $m(\mathbf{A}(\boldsymbol{d}))$ and its up-scaled version $\mathbf{U}(m(\mathbf{A}(\boldsymbol{d})))$, respectively.

use, a ground-truth target frame is required. The main challenge is, therefore, to train networks that are robust enough for accepting a driving frame $\boldsymbol{d}$ that is from another video.

As shown in Fig. 2, training involves a slightly modified pipeline, in which an augmentation $\mathbf{A}$ is applied to the driving image $\boldsymbol{d}$, and a much more elaborate perturbation $\mathbf{P}_{\text{train}}$ takes place. Additionally, the low-res and high-res generators ℓ and h are using the driver's mask $m(\mathbf{A}(\boldsymbol{d}))$ and its up-scaled version $\mathbf{U}(m(\mathbf{A}(\boldsymbol{d})))$ respectively, instead of using the refined mask $\boldsymbol{m}_d$:

$$\boldsymbol{m}_d = r(\mathbf{D}(\boldsymbol{s}), \boldsymbol{m}_s, \mathbf{P}_{\text{train}}(m(\mathbf{A}(\boldsymbol{d})))) \tag{5}$$
$$\boldsymbol{c} = \ell(\mathbf{D}(\boldsymbol{s}), \boldsymbol{m}_s, m(\mathbf{A}(\boldsymbol{d}))) \tag{6}$$
$$\boldsymbol{f} = h(\boldsymbol{s}, \mathbf{U}(\boldsymbol{m}_s), \mathbf{U}(m(\mathbf{A}(\boldsymbol{d}))), \boldsymbol{c})\,, \tag{7}$$

The augmentation $\mathbf{A}$ is a color transformation that scales the input's brightness, contrast and saturation by a random value drawn from $[0.9, 1.1]$, and shifts its hue by a random value drawn from $[-0.1, 0.1]$. $\boldsymbol{P}_{\text{train}}$ performs the following steps sequentially: (i) breaks the image vertically (horizontally) into six parts, and scales each part horizontally (vertically) by a random value drawn from $[0.75, 1.25]$. Next, it scales the entire output vertically (horizontally), by a random value drawn from $[0.75, 1.25]$. (ii) similarly to $\boldsymbol{P}_{\text{test}}$, setting to zero pixels that are lower than the threshold value ρ, and, finally (iii) adds an element-wise noise sampled from the Poisson distributions with $\lambda = 20$.

Loss Terms We train our system end to end using only two loss terms: a mask refinement loss and a perceptual reconstruction loss. At train time, the role of the mask refinement network r is to recover the driver's identity after it was reduced by the perturbation operator $\mathbf{P}_{\text{train}}$, therefore, we minimize the L_1 loss of the driver's mask $m(\mathbf{A}(\boldsymbol{d}))$ and its refined mask $\boldsymbol{m}_d$:

$$\mathcal{L}_{\text{mask}}(\boldsymbol{d}) = L_1(\boldsymbol{m}_d, m(\mathbf{A}(\boldsymbol{d})))\,. \tag{8}$$

For the image reconstruction loss of the generators ℓ and h, following Siarohin et al. (2019b) and based on the implementation of Wang et al. (2018), we minimize a perceptual loss using the pre-trained weights of a VGG-19 model. For two images $\boldsymbol{a}$ and $\boldsymbol{b}$, the reconstruction loss terms using the j^{th} layer of the pre-trained VGG model are written as:

$$\mathcal{L}_{\text{VGG}}(\boldsymbol{a}, \boldsymbol{b})_j = \text{AVG}(|\mathbf{N}_j(\boldsymbol{a}) - \mathbf{N}_j(\boldsymbol{b})|) \tag{9}$$

where AVG is the average operator and $\mathbf{N}_j(\cdot)$ are the features extracted using the j^{th}-layer of the pre-trained VGG model. For the coarse and fine predictions $\boldsymbol{c}$ and $\boldsymbol{f}$, and a driving frame $\boldsymbol{d}$, we compute the following reconstruction loss for multiple resolutions:

$$\mathcal{L}_{\text{reconstruct}} = \sum_{s} \sum_{j} \mathcal{L}_{\text{VGG}}(\boldsymbol{c_s}, \boldsymbol{d_s})_j + \mathcal{L}_{\text{VGG}}(\boldsymbol{f_s}, \boldsymbol{d_s})_j \tag{10}$$

where input image $\boldsymbol{a_s}$, has a resolution $\boldsymbol{s} \in [256^2, 128^2, 64^2]$. We use the first, the third and the fifth ReLU layers of the VGG-19 model. Note that while the VGG network was designed for a specific resolution (224^2), the first layers are fully convolutional, and can be used for an arbitrary input scale.

The combined loss is given by $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{mask}} + \lambda_2 \mathcal{L}_{\text{reconstruct}}$, for weight parameters $\lambda_1 = 100$ and $\lambda_2 = 10$. To avoid unwanted adaptation of the network m, the backpropagation of $\mathcal{L}_{\text{mask}}$ only updates the weights of the mask refinement network r. When backpropagating the second part of the reconstruction loss $\sum_s \sum_j \mathcal{L}_{\text{VGG}}(\boldsymbol{f_s}, \boldsymbol{d_s})_j$, only the top-scale generator h is updated. The Adam optimizer is employed with a learning rate of 2×10^{-4} and β values of 0.5 and 0.9. The batch size is 16 and similar to Siarohin et al. (2019b), we decay the learning rate at epochs 60 and 90, running for a total of 100 epochs. The mask refinement network r starts training after we complete the first training epoch, which is about when the outputs of the mask generator m start to be meaningful.

4 EXPERIMENTS

The training and evaluation were done using three different datasets, containing short videos of diverse objects. **Tai-Chi-HD** is a dataset containing short videos of people doing tai-chi exercises. Following Siarohin et al. (2019b), 3,141 tai-chi videos were downloaded from YouTube. The videos were cropped and resized to a resolution of 256^2, while preserving the aspect ratio. There are 3,016 training videos and 125 test videos. **VoxCeleb** is an audio-visual dataset consisting of short videos of talking faces, introduced by Nagrani et al. (2017). VoxCeleb1 is the collection used, and as pre-processing, bounding boxes of the faces were extracted and resized to 256^2, while preserving the aspect ratio. It contains an overall number of 18,556 training videos and 485 test videos. **RoboNet** contains short videos of robotic arms interacting with different objects (Dasari et al., 2019). The subset used depicts the Sawyer robot. It contains 42,880 training videos and 128 test videos. Each video consists of 30 frames and has a resolution of 256^2. We were unable to obtain the UvA-NEMO smile dataset of Dibeklioğlu et al. (2012), which was utilized in some of the earlier contributions.

We follow the evaluation process of Siarohin et al. (2019b). First, the method is quantitatively evaluated for video reconstruction, and then qualitatively for the task of image animation, where the source and driving videos are of different identities. Additionally, despite being model-free, we compare to model-based methods in the few-shot-learning scenario. In this case, our method, unlike the baseline methods, does not employ any few shot samples.

The metrics are borrowed from related work: **L1** is the L1 distance between the generated and ground-truth videos. **Average Key-points Distance (AKD)** measures the average distance between the key-points of the generated and ground-truth videos. For the Tai-Chi-HD dataset, we use the human-pose estimator of Cao et al. (2016), and for the VoxCeleb dataset we use the facial landmark detector of Bulat & Tzimiropoulos (2017). **Missing Key-points Rate (MKR)** measures the percentage of key-points that were successfully detected in the ground-truth video, but were missing in the generated video. The human-pose estimator of Cao et al. (2016) outputs for every keypoint an indicator of whether it was successfully detected. Using this indicator, we measure MKR for the Tai-Chi-HD dataset. **Average Euclidean Distance (AED)** measures the average Euclidean distance in some embedding space between the representations of the ground-truth and generated videos. Following Siarohin et al. (2019b), we employ the feature embedding of Siarohin et al. (2019a). **Structural Similarity (SSIM)** (Wang et al., 2004): we compare the structural similarity of the ground-truth and generated images. **Cosine Similarity (CSIM)** the identity similarity of the generated and ground-truth faces, by comparing the cosine similarity of embedding vectors generated by a the ArcFace face recognition network (Deng et al., 2019).

4.1 VIDEO RECONSTRUCTION

The video reconstruction benchmarks follow the training procedure in that the source and target frames are taken from the same video. For evaluation, the first frame of a test video is used as the source frame and the remaining frames of the same video as the driving frames. The goal is to reconstruct all frames of the test video, except the first.

L1, AKD, MKR and AED are compared with the state of the art model-free methods, including X2Face of Wiles et al. (2018), MonkeyNet of Siarohin et al. (2019a) and the method suggested by Siarohin et al. (2019b), which we refer to as FOMM. The results are reported in Tab. 1. Evidently, our method outperforms the baselines for each of the datasets and all metrics by a significant margin, except for the AKD measure on the VoxCeleb dataset, where accuracy was decreased by 2.7%. The most significant improvement is for the Tai-Chi-HD dataset, which is the most challenging dataset, because it consists of diverse movements of a highly non-rigid type.

Next, we follow Zakharov et al. (2019) and compare SSIM and CSIM with X2Face, Pix2PixHD (Wang et al., 2018a) and the method of Zakharov et al. (2019), which we refer to as FSAL. The baselines are evaluated in the few-shot-learning setting, where models are fine-tuned on a set of size #FT, consisting of frames of a person that was not seen during the initial meta-learning step. After the fine-tuning step, the evaluation is done on a hold-out set, consisting of unseen frames of the same person. The evaluation is done for VoxCeleb1 and the results are reported in Tab. 2. As can be seen, our method generalizes better and outperforms the baselines in SSIM and even more so in CSIM. This is especially indicative of the method's capabilities, since (i) we skip the fine-tuning step for our model (in our case #FT $= 0$), and (ii) X2Face and FSAL were designed specifically for faces, while our method is model-free and generic.

4.2 Image Animation

The task of image animation is to animate a source image using a driving video. The object (and its background) in the source and driving inputs may have different identities and appearances. To this end, we use the first frame of a source video as the source frame for encoding the appearance, and we use all frames of the driving video as the driving frames for encoding the object's motion. A video is generated where the content of the source frame is animated by the driving video.

Sample results for image animation compared to the baseline methods are shown in Fig. 4. For the VoxCeleb dataset, our method better preserves the identity of the source, and the facial expressions of the generated frames are more compatible with that of the driver's. For the Tai-Chi-HD dataset, the baseline methods tend to generate infeasible poses for the fourth generated frame, while we do not. Unlike FOMM, we well maintain environment elements, such as the stick on the top-right of the generated frame. For the RoboNet dataset, the images generated by our method are the sharpest, while we are the only method that places the generated object in the right position. It is worth noting that the samples were selected to match those of Siarohin et al. (2019a), and not by us.

Ablation The main challenge in reanimation is to replace the identity on the driver's mask to that of the source, while keeping on driver's general pose and shape. We do that in two steps: (i) the driver's identity is reduced by applying $\mathbf{P}_{\text{test}}$, (ii) we inject the source's identity using the mask refinement network r. To evaluate the roles of $\mathbf{P}_{\text{test}}$ and r, we evaluate three reduced methods: no_pert, no_ref and no_id, where the first, second, or both steps are removed, respectively.

Ablation and intermediate results generated by our pipeline are shown in Fig. 3. As can be seen, the generated masks $\boldsymbol{m}_s$ and $m(\boldsymbol{d})$ capture very accurately the object's pose and shape, and the mask refinement network r successfully applies the source's identity to the driver's mask. Comparing the generated frame $\boldsymbol{f}$ to that of FOMM, we notice that for the Tai-Chi-HD dataset, the pose of the generated body using our method, is much more compatible with that of the driver's, where the model of FOMM generates a distorted body. For the VoxCeleb dataset, using our method, the identity of the source is better preserved, as it also reveals a small portion of the teeth, as the driver does. For the Robonet dataset, unlike FOMM, our method was able to inpaint the occluded surface, including the white and blue items on the right of the generated frame.

Examining the generated frames of the ablation models shows that both steps, identity-perturbation and mask refinement, are critical for our image-animation pipeline. The frames generated by no_pert and no_id have significant traces of the driver's identity. This is especially clear for VoxCeleb, on the forehead area of no_pert and the general appearance for no_id. Similarly, for the Tai-Chi-HD dataset, the frame generated by no_ref contains traces from the driver's environment, and for the other datasets it generates distorted results.

User study To qualitatively evaluate our method and compare with existing work, we presented volunteers with a source image, a driving video, and four randomly ordered generated videos, one for

Table 1: Video reconstruction results (lower is better). Improvement is relative to FOMM.

	Tai-Chi-HD				*VoxCeleb*			*RoboNet*
Method	L1	AKD	MKR	AED	L1	AKD	AED	L1
X2Face	0.080	17.654	0.109	0.272	0.078	7.687	0.405	0.065
Monkey-Net	0.077	10.798	0.059	0.228	0.049	1.878	0.199	0.034
FOMM	0.063	6.862	0.036	0.179	0.043	**1.294**	0.140	0.027
Ours	**0.047**	**4.239**	**0.015**	**0.147**	**0.034**	1.329	**0.130**	**0.021**
Improvement	25.4%	38.2%	58.3%	17.9%	20.9%	-2.7%	7.1%	22.2%

Table 2: Results on VoxCeleb1 in the few-shot learning scenario. Unlike the other methods, we do not perform fine-tuning on the identity in the source image. #FT=number of images used for finetuning. P2PHD=Pix2PixHD.

Method	#FT	SSIM ↑	CSIM ↑
X2Face	1/8/32	0.68/0.73/0.75	0.16/0.17/0.18
P2PHD	1/8/32	0.56/0.64/0.70	0.09/0.12/0.16
FSAL	1/8/32	0.67/0.71/0.74	0.15/0.17/0.19
Ours	**0**	**0.80**	**0.70**

Table 3: Percent of selected best video samples for each method based on quality or motion fidelity. X2=X2Face. MN=Monkey-Net.

	Dataset	X2	MN	FOMM	Ours
Quality	*Tai-Chi-HD*	0%	4%	16%	80%
	VoxCeleb	0%	8%	16%	76%
	RoboNet	0%	4%	24%	72%
Motion	*Tai-Chi-HD*	0%	4%	8%	88%
	VoxCeleb	0%	0%	12%	88%
	RoboNet	4%	4%	16%	76%

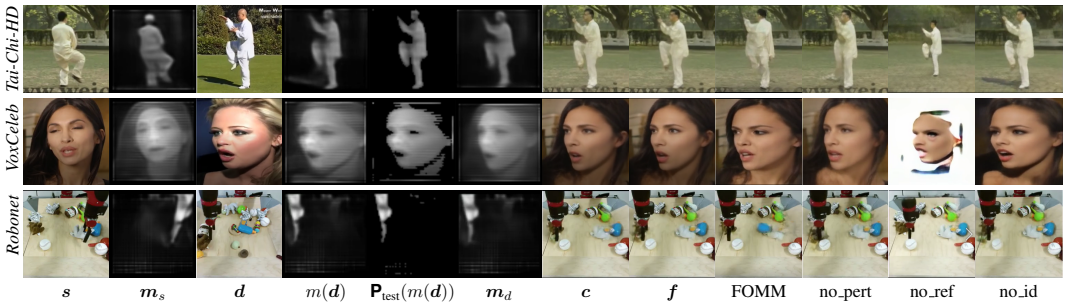

Figure 3: Intermediate results generated by our method. The final generated frame $\boldsymbol{f}$ is compared to FOMM and to ablation models. From left to right: source frame $\boldsymbol{s}$, source mask $\boldsymbol{m}_s$, driving frame $\boldsymbol{d}$, driving mask $m(\boldsymbol{d})$, perturbed driving mask $\mathbf{P}_{\text{test}}(m(\boldsymbol{d}))$, refined driving mask $\boldsymbol{m}_d$, low-res prediction $\boldsymbol{c}$, high-res prediction $\boldsymbol{f}$, FOMM result, and the ablations: no_pert, which drops $\mathbf{P}_{\text{test}}$, no_ref, which omits the mask refinement r, and no_id, which omits both.

each baseline method. They were asked to (i) select the most realistic animation of the source image, and (ii) select the video with the high fidelity to the driver video. For each of the $n = 25$ participants, we repeated the experiment three times, each time using a different dataset and a random test sample.

The results are reported in Tab. 3 and are highly consistent with the video reconstruction results, indicating that the quality and the animation of the videos generated by our method contain fe artifacts and are better synchronized with the driver videos.

5 Conclusions

A novel method for conditionally reanimating a frame is presented. It utilizes a masking mechanism as a means for encoding pose information. By properly augmenting and refining the masks, we are able to effectively extract both the source and the driving masks, while accurately capturing the shape and foreground/background separation of the first, and recovering an identity-free pose from the latter. Our results outperform the state of the art by a sizable margin on the available benchmarks.

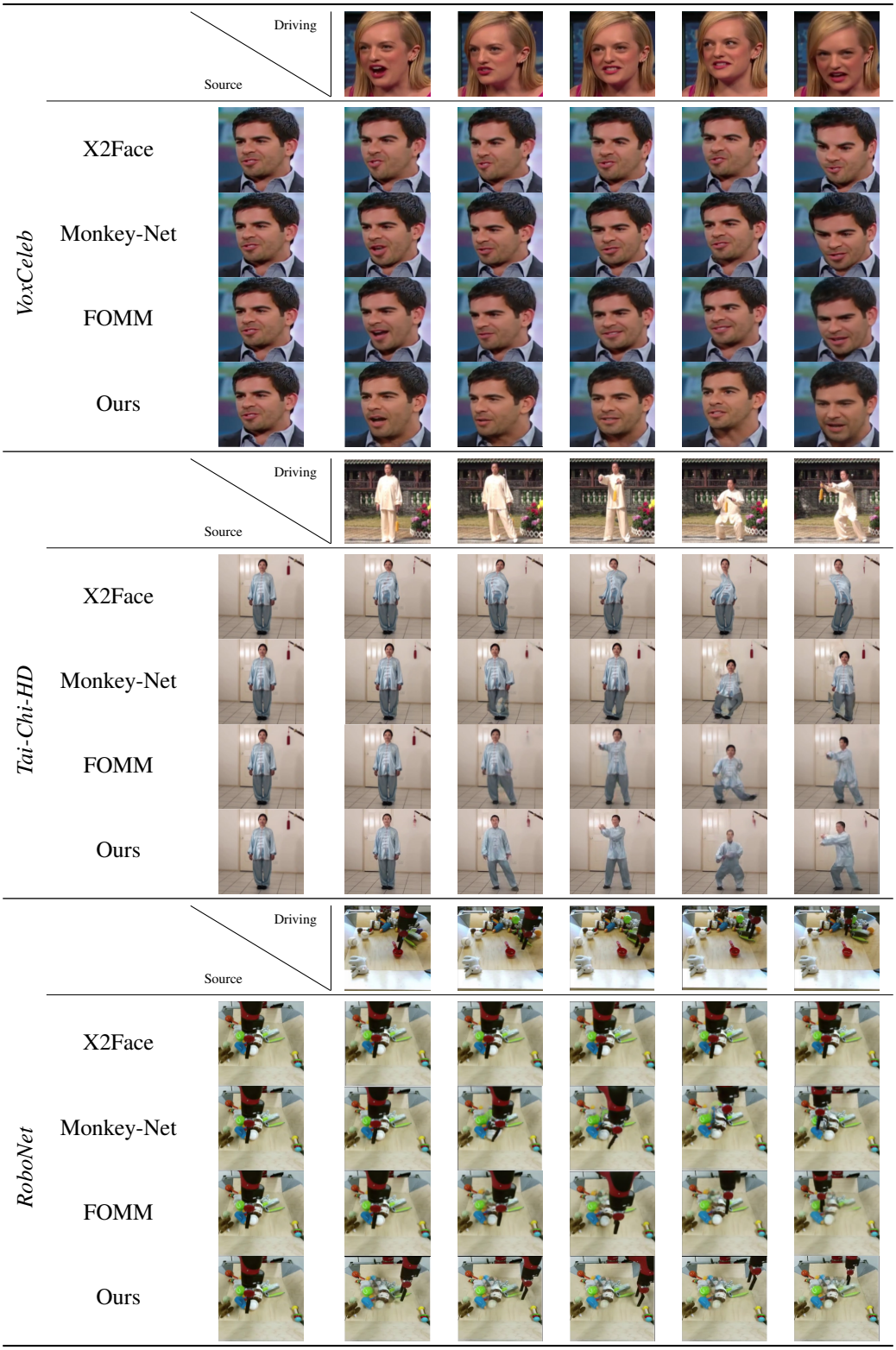

Figure 4: Sample results on the three benchmarks. We use the exact same samples as evaluated by FOMM (Siarohin et al., 2019b).

REFERENCES

Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). *arXiv e-prints*, art. arXiv:1703.07332, March 2017.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *arXiv e-prints*, art. arXiv:1611.08050, November 2016.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody Dance Now. *arXiv e-prints*, art. arXiv:1808.07371, August 2018.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-Scale Multi-Robot Learning. *arXiv e-prints*, art. arXiv:1910.11215, October 2019.

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

Hamdi Dibeklioğlu, Albert Ali Salah, and Theo Gevers. Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *European Conference on Computer Vision*, pp. 525–538. Springer, 2012.

Aysegul Dundar, Kevin J. Shih, Animesh Garg, Robert Pottorf, Andrew Tao, and Bryan Catanzaro. Unsupervised Disentanglement of Pose, Appearance and Background from Images and Videos. *arXiv e-prints*, art. arXiv:2001.09518, January 2020.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal Unsupervised Image-to-Image Translation. *arXiv e-prints*, art. arXiv:1804.04732, April 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv e-prints*, art. arXiv:1611.07004, November 2016.

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 Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10955–10964, 2019.

A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7690–7699, 2020.

Aliaksandr Siarohin, Stephane Lathuiliere, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

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

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-Video Synthesis. *arXiv e-prints*, art. arXiv:1808.06601, August 2018.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018a.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2Face: A network for controlling face generation by using images, audio, and pose codes. *arXiv e-prints*, art. arXiv:1807.10550, July 2018.

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

Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars. *arXiv e-prints*, art. arXiv:2008.10174, August 2020.

A ARCHITECTURE

Following Siarohin et al. (2019b), the mask generator m, the mask refinement network r and the high-res generator h have the same encoder-decoder architecture, followed by a $conv_{7\times7}$ layer and a $sigmoid$ activation. The encoder (decoder) consists of five encoding (decoding) blocks, where each encoding block is a sequence of $conv_{3\times3} - relu - batch\_norm - avg\_pool_{2\times2}$, and each decoding block is a sequence of $up\_sample_{2\times2} - conv_{3\times3} - batch\_norm - relu$. Only for the high-res generator h we add skip connections from each of the encoding layers, to its corresponding decoding layer, to form a U-Net architecture.

The encoder of the low-res generator ℓ consists of $conv_{7\times7} - batch\_norm - relu$, followed by six residual blocks, where each block consists of $batch\_norm - relu - conv_{3\times3} - batch\_norm - relu - conv_{3\times3}$. The decoder consists of two blocks, where each block is a sequence of $up\_sample_{2\times2} - conv_{3\times3} - batch\_norm - relu$. The decoder is followed by a $conv_{7\times7}$ layer and a $sigmoid$ activation.