SINGLE SOURCE ONE SHOT REENACTMENT USING WEIGHTED MOTION FROM PAIRED FEATURE POINTS

Soumya Tripathy[*]*, Esa Rahtu*

Computer Vision Group
Tampere University
Tampere, Finland
soumya.tripathy, esa.rahtu@tuni.fi

Juho Kannala

Department of Computer Science
Aalto University
Helsinki, Finland
juho.kaanala@aalto.fi

ABSTRACT

Image reenactment is a task where the target object in the source image imitates the motion represented in the driving image. One of the most common reenactment tasks is face image animation. The major challenge in the current face reenactment approaches is to distinguish between facial motion and identity. For this reason, the previous models struggle to produce high-quality animations if the driving and source identities are different (cross-person reenactment). We propose a new (face) reenactment model that learns shape-independent motion features in a self-supervised setup. The motion is represented using a set of paired feature points extracted from the source and driving images simultaneously. The model is generalised to multiple reenactment tasks including faces and non-face objects using only a single source image. The extensive experiments show that the model faithfully transfers the driving motion to the source while retaining the source identity intact.

1. INTRODUCTION

General image reenactment and particularly facial reenactment have received plenty of attention in recent years due to numerous applications in game design, movie production, virtual reality, and interactive system design. The current state of the art models [1, 2, 3, 4, 5, 6, 7] can produce realistic talking heads of a source from a single image by imitating the facial movements from another similar looking talking video, commonly known as the driver. Impressive results often require careful selection of source and driving pairs with closely matching identities. For example, models like [3, 7, 8, 9] generate high-quality talking heads for a person who drives his own face (self-reenactment) or a face with a comparable head structure. Other models [5, 10, 6, 2, 11] require facial identity and motion representations in terms of 3D models or pretrained representation like landmarks, head poses or Action Units (AUs) [12]. These pretrained models usually require costly annotations and often fail to handle occlusions or extreme head poses.

Some of these issues are tackled in [13, 1] using unsupervisedly learned motion representation defined as a function of key points. Although the keypoint detector is obtained without explicit annotations, the learning process is driven by objective functions like equivariance loss, which encourage landmark like locations (e.g. lip corners). Figure 1 illustrates examples of detected keypoints [1], which are mostly located on facial contours. The design choice is reasonable as most of the state of the art reenactment models [3, 2, 7, 6, 14] use landmark-based motion representations. However, the landmark and contour-driven locations are prone to contain substantial shape information. One important advantage in the unsupervisedly learned keypoints is the fact that they are object agnostic and can be used to animate other objects than faces.

Face landmarks or keypoint based models generate high-quality talking heads for self reenactment, but often fail in cross-person reenactment where the source and driving image have different identities. The main reason is that landmarks are person-specific and carry facial shape information in terms of pose independent head geometry [4, 6]. Any differences of shape between source and driving heads are reflected in the facial motion (through landmarks or keypoints) and lead to a talking head that can not faithfully retain the identity of the source's person. This effect can be seen in Figure 1 for faces and in Figure 5 for non-face objects using a keypoint based reenactment model [1]. Furthermore, these models use each keypoint independently to affect the motion of its neighborhood pixels which makes the output highly dependent on the quality of the keypoints or landmarks. Any noisy keypoint prediction may severely distort the facial shape and thereby generate low-quality talking heads of the source as shown in Figure 1.

Considering the aforementioned issues in the existing reenactment models, we propose two important improvements. First, we propose a new paired feature point detector that predicts anchors on the source and driving im-

[*]Corresponding author

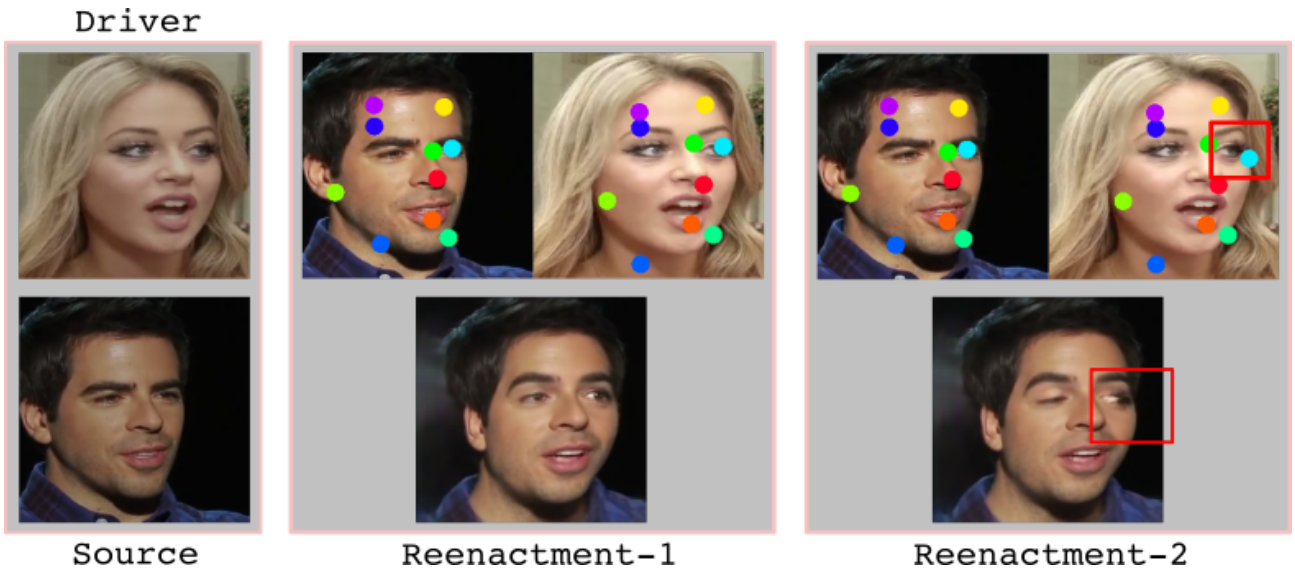

Fig. 1. Illustration of drawbacks in keypoints/landmarks based reenactment models. In both cases, the reenactment is performed using FOM [1] and the keypoints are drawn on the source and driving images. In Reenactment-1, the head structure difference between the source and driving is reflected in the output (bottom image) as the source's facial structure and identity are distorted. In the Reenactment-2, one of the key points (in the red box) is slightly displaced manually from its original position to show its effect on the output. The degradation in the output quality shows the overall system performance is highly dependant on the keypoint detectors.

ages that best describe the motion between them without imitating the landmarks. Due to lack of the physical significance like landmarks, we refer to the detected locations as feature points instead of keypoints. In previous works [1, 3, 7], the keypoint predictions (supervised and self-supervised) are extracted independently from the source and driving images. Such setup prevents optimising the locations for a specific source and driving identities leading to landmark like keypoints. In contrast, we predict the feature points using source and driving images jointly using a multi-headed co-attention layer. Hence, we call them as paired-feature-points instead of keypoints throughout the paper. The paired-feature-points encourage the detector to predict different features for different pairs without encoding the facial shapes.

Second, we propose a new motion model that predicts the motion for each source pixel using all the paired-feature-points. Here we use the Moving Least Square [15] formulation where each pixel's motion depends on all the paired-feature-points, unlike the first-order-model [1] where each corresponding keypoint is only responsible for the motion of its neighborhood pixels. Our model is less dependent on the correctness of any individual keypoint and unlikely to fail in conditions where some of the keypoints' positions are wrong due to the occlusions or changing head poses. We use a simple and robust formulation to express the motion with paired feature points. Our model is applicable to both face and non-face objects similarly to [1, 13].

We show that our paired-feature-point detector and motion model can be used to effectively reenact a face from a single image without any strong priors on the identities, initial pose, or representations (like landmarks, action units) unlike any other state-of-the-art model. In the facial cross-person reenactment, we show experimentally that our model preserves the identity better than other one-shot reenactment models. In addition, we compare the proposed model with few-shot learning based models and demonstrate improvements in pose and expression similarity. We also show qualitatively that our model works on objects other than faces, similarly to [1]. Finally, we analyze the robustness of our reenactment model with respect to the feature point locations. The results indicate that our model tolerates imperfections significantly better compared to previous works.

2. RELATED WORK

Face reenactment has seen a lot of interest from the research community in the last few years and it led to photo-realistic talking heads in [3, 9, 16, 17, 18, 7] where the source and driving identities are the same. Representation of the pose and emotion from the driving images is a key step to achieve higher quality talking faces and landmarks are used as that representation in these cases. However, the landmarks are person-specific and transfers the identity information along

with the pose and expression which leads to poor cross identity reenactment. To address this MarioNETte [2, 19] uses a landmark transformer to remove person-specific information but it requires separate hand-crafted data and model design. A few other models [10, 6, 20] use action units for reenactment as they are not person-specific. However, obtaining action units and generating photorealistic images from them with varying head poses are challenging tasks.

In X2Face[8], as an alternative to the landmark-based models, the pose and expression are unsupervisedly learned from the driving images in the form of latent codes of an encoder-decoder architecture. This approach alllows to use other modalities, like audio, as the driver for the reenactment models. However, these latent codes are difficult to be disentangled from identity-like landmarks and suffer in the cross reenactment case. In [1, 13, 21], similar unsupervised learning is used to obtain keypoints from the driving and source images from which the motion can be obtained. However, these keypoints are similar to the landmarks and don't perform well in the cross-reenactment case. In [1], they proposed to utilize the difference of keypoints between two consecutive driving frames as the motion cue for the source. Although it cancels out the shape of the driving face, it only works if the source has the same pose and expression as the first frame of the driving video. Unfortunately, such conditions put a restriction on the choice of source and driving pairs for the reenactment. A similar unsupervised model is proposed in [4] that uses few-shot learning at the inference time to train the model for each identity and requires several images of the source to have better quality results.

Apart from the data-driven models, face reenactment has also been performed using classical 3D face models like 3DMM [22] in [11]. A combination of both 3D models and learning-based models are also used as in [23, 5, 24] to create talking heads. These reenactment models require 3D model parameters as training data which is expensive and limits its application to a larger number of identities.

3. METHOD

The high-level architecture of our reenactment model is inspired by [1]. That is, we first use encoded images to predict the paired feature points and the dense warping field. The warping field is subsequently applied to the source features, which are then used to generate the output image. The most important differences to [1] are the following: 1) the motion is represented using paired-feature-points instead of keypoints extracted from individual images, and 2) the dense warping field is constructed by weighted dense motion module where the motion for each pixel is estimated by considering all paired feature points at once. These components along with the complete model are presented in the following subsections.

3.1. Overview of our model

Given a face image I_s of a source identity S, our model aims to animate it by copying the facial motion from a driver image I_d with identity D. The animated image generated from our model is called as reenacted image I_r. This process of animation involves two major steps: 1. representation of the motion difference between the source and the driving faces, and 2. applying this motion on to the source and creating a photo-realistic animation of it. In our model, the representation of this motion is obtained by understanding the motion of the feature points from the driving to the corresponding feature points of the source images. Then we use this motion to backward warp the source face in the feature space and reconstruct it using a generator with adversarial loss.

The complete block diagram of our model is presented in Figure 2. The overall steps of our models can be summarized as, 1. representing the source and driving images as a latent vector using an embedder network, 2. extracting the motion features by combining both the latent codes using attention mechanism, 3. estimating pixel-wise motion using a weighted point transformation, 4. creating a warping field from motion using an encoder-decoder architecture, and 5. finally reenacting the image using an occlusion-aware generator network.

3.2. Image Embedder

The first block of our model is an image embedder that is used to transform I_s and I_d to latent vector representations. A single encoder network is applied to both the I_s and I_d independently to map them to a common space. The goal is to use the global representations for each image that has only relevant information in obtaining the motion points. It has a series of convolution, batch norm, and average pooling layers to embed the images into vectors $l_s \in \mathbb{R}^{N\times 1024}$ and $l_d \in \mathbb{R}^{N\times 1024}$ where N is the number of user-defined motion features. We use $N = 10$ in all our experiments.

3.3. Paired-feature-point Estimation Module

Our aim is to learn feature points from l_s and l_d that can be used to express the motion between I_s and I_d. The state-of-the-art models like [1, 13] extract landmarks such as N keypoints from each image individually to capture the structure of each object (the face region in our case). Then the motion is expressed as a function of the changes in the structures between the source and driving images. The model is trained to predict the structure from each images independently without considering the connection between the source and driving pairs as in the reenactment.

Our motivation is to change the feature points depending on the particularities of the I_s and I_d together rather than considering them independently. For example, if one

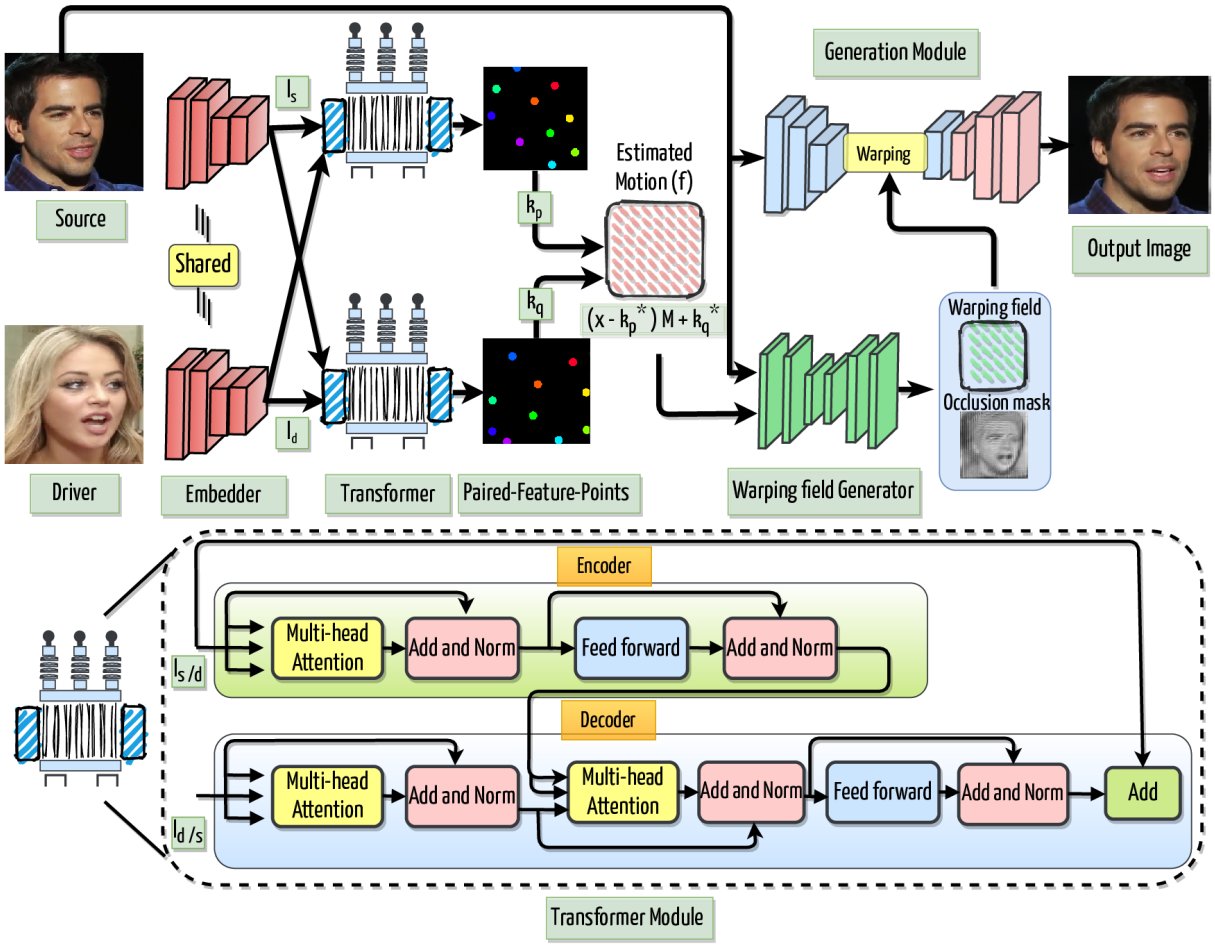

Fig. 2. The complete block diagram of the proposed reenactment model. It processes the source and driving images in five steps as 1. Encoding the images using image embedder, 2. Extracting paired-feature-points using a transformer, 3. Estimating the motion from paired-feature-points, 4. Converting the motion to the warping field and 5. Using the source image with the warping field in the generator to produce the final output. The transformer module is expanded at the bottom to showcase its building blocks in detail.

of the faces in the source-driving pair has occlusion then the feature points for the motion should be adjusted in both the images rather than predicting keypoints in individual faces which are highly erroneous in the occluded images. The module is implemented using a transformer network T that maps $T(l_s, l_d)$ and $T(l_d, l_s)$ to embedding vectors $l_{st} \in \mathbb{R}^{N\times 1024}$ and $l_{dt} \in \mathbb{R}^{N\times 1024}$ respectively. In reality, the transformer predicts the changes in the l_s and l_t such that

$$l_{st} = l_s + T(l_s, l_d), \quad l_{dt} = l_d + T(l_d, l_s) \tag{1}$$

The transformer network consists of one layer of encoder and decoder as shown in the bottom half of the Figure 2. Each of the encoder and decoder layer consists of a self attention layer and position-wise feed forward network to map the latent codes to an intermediate representation. The decoder consists of an additional attention layer that is responsible to combine the intermediate representations from both the latent codes to predict the final embedding vector. As an example, the prediction of l_{st} involves the self-attention of l_s from encoder, the self attention of l_d from decoder and additional co-attention on the output of the encoder to the output of first two layers of decoder. Due to this co-attention layer, each embedding vector l_{st} or l_{dt} is predicted by utilizing the latent codes l_s and l_d from both the images. For all the attention layers we have used a scaled-dot-product attention (A) [25] where for the given query (Q), key (K) and value (V), the A can be written as,

$$A(Q, K, V) = softmax(\frac{P(Q)P(K)^T}{\sqrt{(d_k)}})P(V) \tag{2}$$

P is the function to calculate the sinusoidal positional encoding of the embedding vectors. For the self-attention

layer, the respective latent code l_s or l_d are reused as the Q, K, and V pairs whereas for the co-attention, the output at the end of the second layer of the decoder is considered as Q and the output of the encoder is reused as K and V. We extended this attention layer to the multi-headed attention by projecting the key, query, and value 4 times with different feed-forward networks. We concatenate the attention from each of them to jointly attend the information from different projections as suggested in [25].

Finally, we reshape the predicted embeddings into $\mathbb{R}^{N\times 32\times 32}$ and pass through a softmax layer to obtain N heat-maps which are then converted to N points as k_s and k_d by extracting the mean of the heat-maps. This conversion of the embeddings to N points serve as a bottleneck and provides essential feature points for the motion calculation. The feature points learned from our model and keypoints from FOM [1] are shown in Figure 3. It is clear that our model does not imitate the landmark points like FOM [1]. One interesting aspect to note is that our source landmarks adjust themselves according to the driving image unlike FOM [1] where they are fixed. We note that in [2] the attention layers are utilized for reenactment but they are used to draw a spatial correspondence between source features and driving features to effectively transfer the style of source on to the driver's pose. Our motivation and the design structure of the attention layers are completely different from their counterparts. Our transformer architecture has similarities with the point cloud registration model in [26]. However, our model is designed for fundamentally different data and tasks.

3.4. Motion Estimation

Given the discrete motion points k_s and k_d on the I_s and I_d respectively, our goal is to predict the transformation f per each pixel point x in the driving image such that it minimises

$$\sum_n w_n |f_x(k_{dn}) - k_{sn}|^2 \tag{3}$$

where for a constant $\alpha \leq 1$, the weight w_n has the form

$$w_n = \frac{1}{|k_{dn} - x|^{2\alpha}} \tag{4}$$

The n is a counter on number of feature points and ranges from 1 to N. The equations (3) and (4) together constitutes the Moving Least Squares (MLS) [15] formulation where each point x has its own transformation f_x depending the on its distance from all other feature points. Following the derivation from [15] the final form of the f is,

$$f_x = (x - k_d^*)M + k_s^* \tag{5}$$

where M is the linear transformation matrix, $k_d^* = \frac{\sum w_n k_{dn}}{\sum w_n}$ and $k_s^* = \frac{\sum w_n k_{sn}}{\sum w_n}$. In [15], the feature points are hand-picked on the edges of the object. By considering different

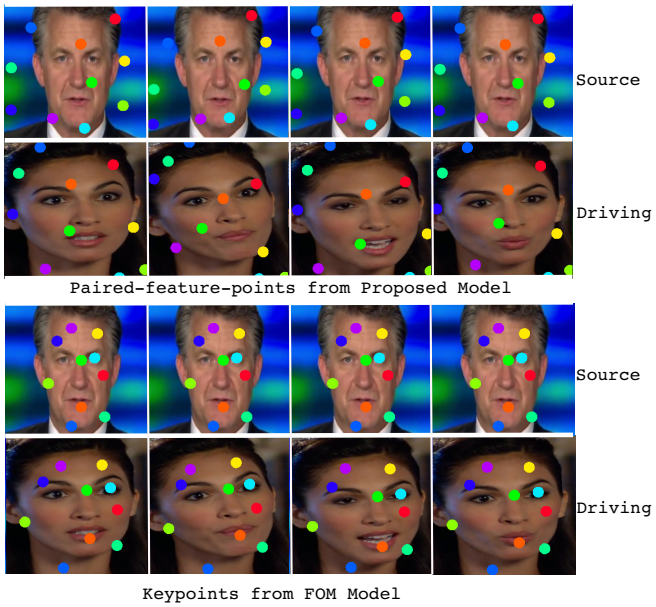

Fig. 3. Illustration of keypoints predicted from FOM[1] and paired-feature-points predicted from our model. During the reenactment, the position of the paired-feature-points are adjusted on both source and driving images depending on the pose and expression of driving image. In FOM[1], the points are predicted independently so source keypoints remain fixed throughout the process.

classes of transformation matrix M (affine, rigid and similarity transformation) a closed-form solution can be derived for f_x which gives reasonable deformations. In our case, the feature points are self-supervised and applying this closed-form, the solution can drive the feature point detector to learn the points similar to landmarks. Moreover, in the initial iterations, this closed-form solution can completely deform the images as the keypoints are not stable and can break the training of the whole network. To avoid such problems, we employed a single convolution layer to predict the transformation matrix from the weight matrix w and the heatmap representations of the k_s and k_d. We extract a single f_x for the whole image rather than learning multiple f_x for each keypoints individually like FOM [1]. Our pixel-level transformations are much stable due to the moving weight matrix unlike the [1] where each pixel's motion is highly dependant on the nearest keypoints motion. Experimentally we have shown that our motion model handles small errors in the prediction of feature points, unlike the counterparts.

3.5. Generation Module

After predicting the motion f, we utilized this to predict a warping function that realigns the source features in the generator network. To detect the occluded parts of the source face an occlusion mask is predicted during this process to

indicate in-painting region for the generator. Along with the knowledge of the occlusion, the realigned features are finally converted to the reenacted source images by the generator. The warping function is generated by a U-Net architecture which is similar to the local motion aggregation block of FOM [1] but we only consider two motions i.e for f and the background instead of $N+1$ motions in FOM.

3.6. Training Loss

We generate the training material using videos of moving objects (faces and other shapes). We randomly sample source and driving pairs from each video, which enables us to use the driving frame as pixel-wise ground truth of the intended animation of the source image. We train our model end-to-end using the perceptual loss [16] in multiple resolution of I_r and I_d. The mathematical expression for perceptual loss in each resolution can be written as

$$\mathcal{L}_p = \sum_i ||VGG_i(I_r) - VGG_i(I_d)||_1 \tag{6}$$

Where $VGG_i(.)$ stands for i^{th} channel response of pretrained VGG-19 network.

Along with the perceptual loss, we apply a loss function to the transformation function f. We apply equation (5) to equation (3) and after simplifying we express our loss function as

$$\mathcal{L}_m = \sum_n w_n |\hat{k_{dn}} M - \hat{k_{sn}}|^2 \tag{7}$$

where $\hat{k_{dn}} = k_{dn} - k_d^*$ and $\hat{k_{sn}} = k_{sn} - k_s^*$. The loss $\mathcal{L}_m$ helps in predicting the transformation matrix M which inturn predicts the per pixel motion function f_x. In order to encourage the paired-feature-points to be spread out in the image, a feature point spreading loss $\mathcal{L}_f$ is applied where the distances between the feature-points are penalised if they fall below a threshold value. Finally the adversarial loss $\mathcal{L}_{adv}$ is applied to the output image to maintain the photo realism of the reenacted image. The final loss function can be written as an weighted sum

$$\mathcal{L}_{total} = \lambda_p \mathcal{L}_p + \lambda_m \mathcal{L}_m + \lambda_f \mathcal{L}_f + \lambda_{adv} \mathcal{L}_{adv} \tag{8}$$

4. EXPERIMENTS

In this section, we assess our model in the face and non-face reenactment tasks. Moreover, we evaluate the robustness of the reenactment models with respect to the feature point (or keypoint) locations. We compare the proposed approach with the following recent works:

- *FSGAN* [7] uses direct landmarks from the driving image to reenact the source face from a single image.

Model	Test-phase training	ISIM↑	PSIM↑	ESIM↑	FID↓
LPD_{32} [4]	✓	0.80	0.67	0.92	-
LPD_1 [4]	✓	0.78	0.61	0.92	-
FSGAN [7]	×	0.39	0.78	0.91	192.01
FACEGAN [6]	×	0.49	0.84	0.88	198.75
FOM [1]	×	0.57	0.90	0.93	127.79
Proposed	×	0.70	0.80	0.93	115.54

Table 1. Quantitative comparison of our model with the state-of-the-arts. The FID scores of LPD models [4] are not calculated becasue they generate faces without the background unlike other models.

- *FACEGAN* [6] is a one-shot reenactment model that utilises a combination of source landmarks and driving AUs to reenact the source image.
- *FOM* [1] learns keypoints in a self-supervised fashion from the source and driving images. The reenactment is done using the motion extracted from these keypoints.
- *LPD* [4] learns identity and pose descriptors in a self-supervised way from videos and then combines identity of source and pose descriptors of driving image for the reenactment. They use a few-shot learning framework to finetune the model for each source identity at the test phase. Although this is different from other methods, which are identity agnostic, we include this method as a reference to our experiments. Furthermore, we use LPD with one and 32 samples of the source identity.

The baselines cover various popular features and techniques such as landmarks, AUs, keypoints, pose-descriptors, and few-shot learnings presented in the reenactment literature. We note that our model uses only one source image at the test phase and generates output without any person-specific finetuning (unlike few-shot works). All the output images have a final resolution of 256×256.

4.1. Face-reenactment

The face reenactment models are trained using talking head videos from Voxceleb [27]. All the videos are preprocessed as in [1] to obtain the source and driving frames at a resolution of 256×256. For the evaluation, we randomly sampled 40 identities (different from those in training) from Voxceleb and FaceForensic++ [28] datasets, and generated 80k reenacted images by taking the source and driving as different identity (cross-person setting).

Qualitative comparison of our model with its counterparts are shown in the cross-person setting in the Figure 4. It is clear that the landmark and keypoint-based models like FSGAN [7] and FOM [1] leak the driving facial structure to the source face in the final image. It makes the reenacted image lose the source identity. In LPD [4], there is

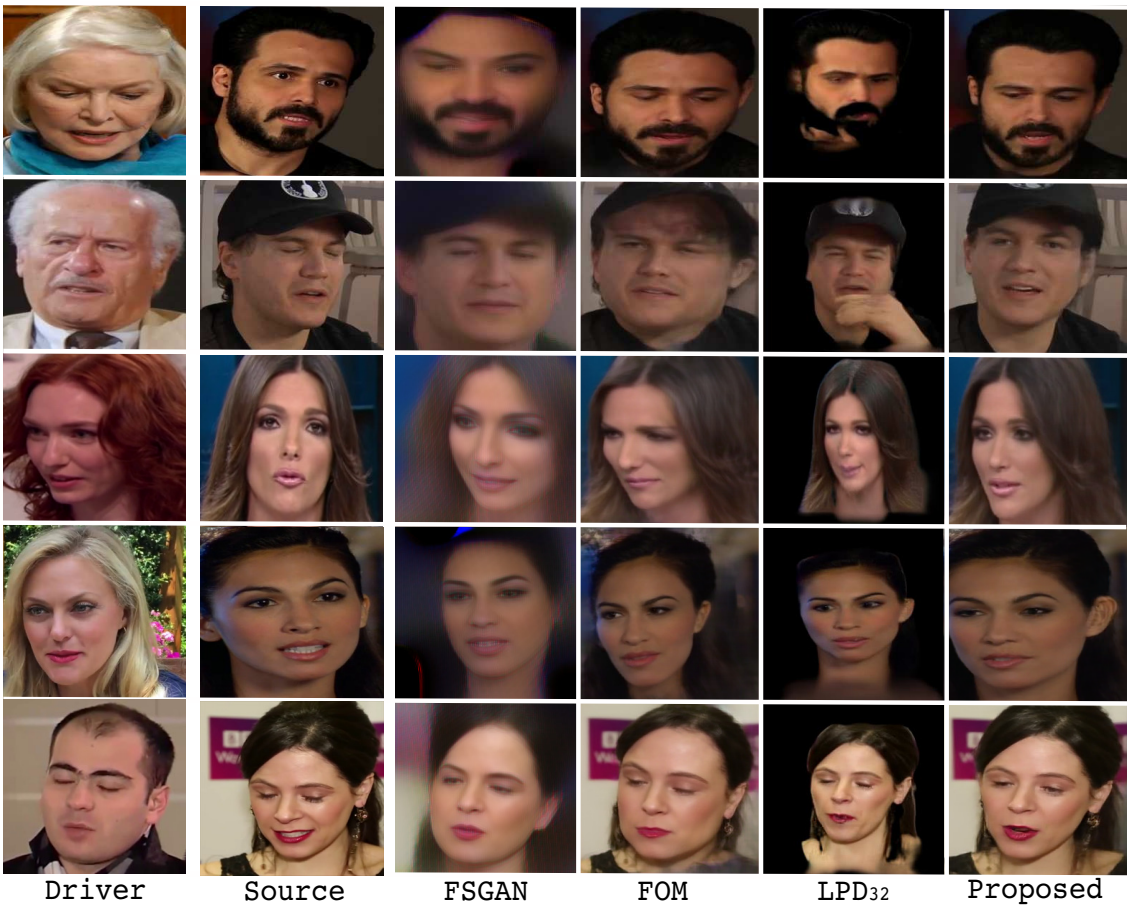

Fig. 4. Qualitative comparision of proposed model with FSGAN[7], FOM[1] and LPD[4]. Our model better reproduces the source identity, facial shape and driving motion at the output. More results can be seen in the supplementary material.

no shape leaking between source and driving but they fail to replicate the facial motion effectively as can be seen from Figure 4 (row 1 and 2). Moreover, they require multiple source images and a source-specific training step to generate good quality final images. In our method, the paired-feature-points are less sensitive to facial structure, and together with our motion model, it reproduces the source identity with driving motion at the output better than its counterparts. Additional qualitative examples are provided in the supplementary material.

Quantitative comparison of reenactment model is difficult in the cross-person setting due to lack of the exact ground truth. Nevertheless, several indirect measurements have been applied in the reenactment literature. In our experiments, we use the following metrics:

- *Identity Cosine Similarity between IMage embeddings (ISIM)*: It measures the identity similarities between source and reenacted faces by comparing the embeddings vectors from a pretrained face recognition network[29]. The higher ISIM score signifies better identity reproduction ability at the output.
- *Pose Cosine Similarities between IMages (PSIM)*: It measures the cosine similarity of head pose angles of driving and reenacted faces using a pretrained model [30]
- *Expression Cosine Similarities between IMages (ESIM)*: To measure the expression retention capability of the model, the embedding vectors from a pretrained action unit detector [31] is used for the similarity calculation.
- *Frechet-inception distance (FID)*: It measures the perceptual similarities of the generated images and training images. A lower score signifies better photo-realism of the reenacted images.

The quantitative comparisons are shown in Table 1. The proposed approach achieves the highest ISIM score among all one-shot models, which illustrates the ability to retain the source identity at the output. One of the key reasons behind the performance is the pairwise-feature-points, which are shape independent unlike the keypoints in FOM [1] and FSGAN [7]. Only the LPD models [4] achieve higher ISIM, which is understandable as they are trained with source iden-

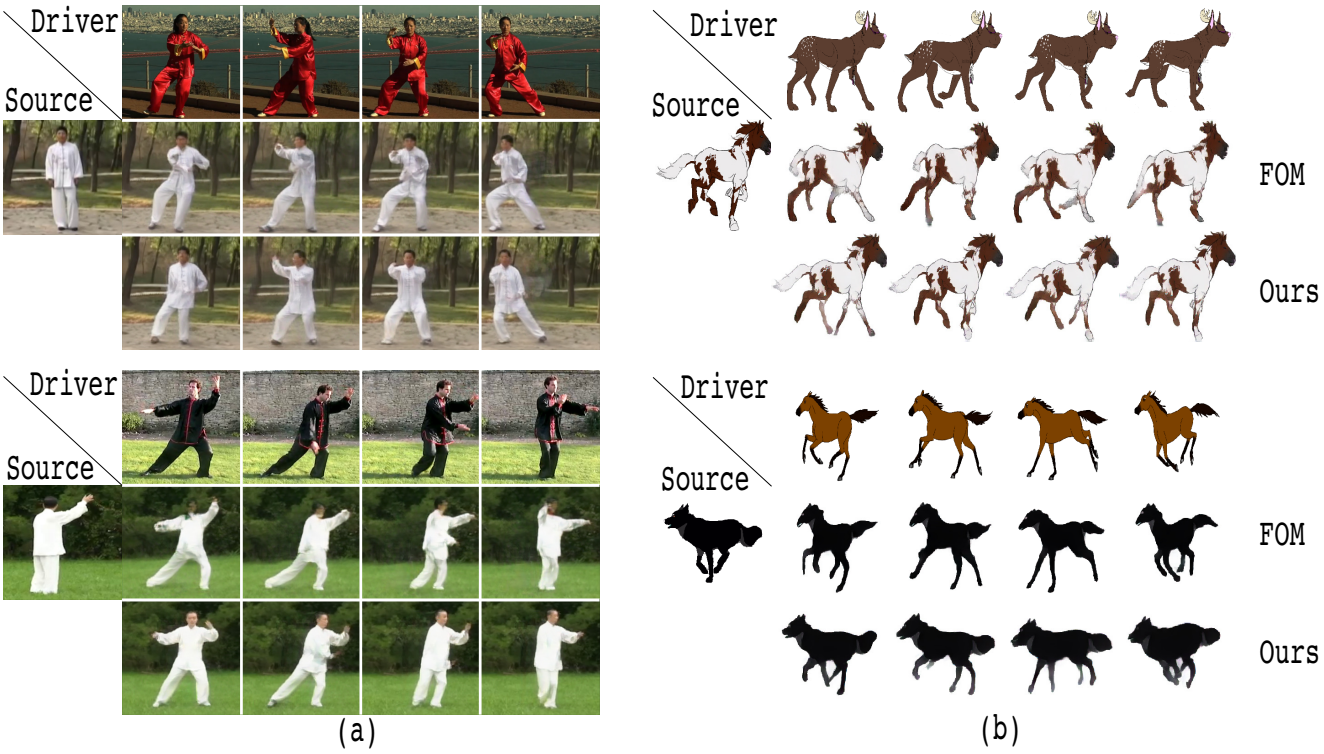

Fig. 5. Qualitative comparison of proposed model with FOM[1] on a. Tai-chi-HD [1], and b. MGif[13] datasets. Our model keeps the source shape and driver's motion intact at the output unlike FOM[1]. More results can be seen in the supplementary material.

tity at the test phase. The pose vectors for PSIM scores are calculated using an external pretrained network. The network utilises landmarks to obtain the poses, which makes it highly sensitive to facial contours. In terms of PSIM, FOM [1] and FACEGAN [6] achieve higher scores as they aim to match the landmarks between the output and the driving images. However, in this process, they hamper the image quality and identity as can be seen from the other metrics. Our model achieves a better balance between identity, pose, expression, and image quality in comparison to other models with a single source image.

Model	Without Noise	With Noise	Change
FOM [1]	1.11	2.19	97.3%
Proposed	1.31	1.61	22.9%

Table 2. Mean landmark difference scores in self-reenactment between FOM [1] and our model. Uniform noise is added to a single keypoint of driving images to analyze the stability of feature-points of our model

4.2. Stability of paired-feature-points

We have argued and qualitatively shown in Figure 3 that the proposed paired feature points are different from keypoints used in FOM [1]. To assess this, we randomly select one feature point (or keypoint) from each driving image and added uniform random noise between 0.05 to 0.5 to its location before the reenactment (point locations normalized between 0 to 1). We hypothesize that if the keypoints encode landmarks like facial structures then any distortion to it will severely distort the final image. To verify that we perform a self-reenactment experiment using 30 identities from the test set and calculate the mean landmark difference between output images and the driving images as shown in Table 2. The 97% increase in the landmark error shows that FOM [1] is highly dependant on the correctness of keypoints and less robust than our paired-feature-points in the reenactment tasks. We provide qualitative examples of the output images for both methods in the supplementary material.

4.3. Reenacting non-face objects

The proposed formulation does not make any assumptions on the reenacted object type. Therefore, the same model can be also trained without modifications to reenact other objects besides faces. To this end, we train our method using MGif [13] and Tai-chi-HD datasets [1]. We provide a few qualitative reenactment examples in Figure 5, where we compare it to FOM [1]. The proposed model is better in preserving the source object identity compared to FOM. We

provide additional examples in the supplementary material.

5. CONCLUSION

We have proposed a novel paired-feature-point detector and motion model to unsupervisedly extract the motion from the driver to reenact the source face. Our feature points are shape/identity independent and represent the motion based on the source-driving pairs, unlike its contemporaries. Our motion model predicts the motion of each source pixel based on all the feature points instead of the closest one which makes it more stable to any errors in feature point prediction. We have shown experimentally that our model produces high-quality reenactment output from a single image by keeping the desired identity, pose, expression, and photo-realism intact.

REFERENCES

[1] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019.

[2] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, "Marionette: Few-shot face reenactment preserving identity of unseen targets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[3] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9459–9468.

[4] E. Burkov, I. Pasechnik, A. Grigorev, and V. Lempitsky, "Neural Head Reenactment with Latent Pose Descriptors," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 783–13 792, June 2020.

[5] G. Yao, Y. Yuan, T. Shao, and K. Zhou, "Mesh guided one-shot face reenactment using graph convolutional networks," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1773–1781.

[6] S. Tripathy, J. Kannala, and E. Rahtu, "Facegan: Facial attribute controllable reenactment gan," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1329–1338.

[7] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7184–7193.

[8] O. Wiles, A. S. Koepke, and A. Zisserman, "X2Face: A Network for Controlling Face Generation Using Images, Audio, and Pose Codes," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11217, pp. 690–706.

[9] E. Zakharov, A. Ivakhnenko, A. Shysheya, and V. Lempitsky, "Fast bi-layer neural synthesis of one-shot realistic head avatars," in *European Conference on Computer Vision*. Springer, 2020, pp. 524–540.

[10] S. Tripathy, J. Kannala, and E. Rahtu, "Icface: Interpretable and controllable face reenactment using gans," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 3385–3394.

[11] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] P. Ekman and W. V. Friesen, "Facial action coding system: A technique for the measurement of facial movement," https://www.semanticscholar.org/paper/Facial-action-coding-system%3A-a-technique-for-the-of-Ekman-Friesen/1566cf20e2ba91ca8857c30083419bf7c127094b, 1978.

[13] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2377–2386.

[14] W. Wu, Y. Zhang, C. Li, C. Qian, and C. Change Loy, "Reenactgan: Learning to reenact faces via boundary transfer," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 603–619.

[15] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 533–540.

[16] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8798–8807.

[17] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[18] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 95:1–95:13, July 2017.

[19] J. Zhang, X. Zeng, M. Wang, Y. Pan, L. Liu, Y. Liu, Y. Ding, and C. Fan, "Freenet: Multi-identity face reenactment," in *CVPR*, 2020, pp. 5326–5335.

[20] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 818–833.

[21] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, "Unsupervised Learning of Object Landmarks through Conditional Image Generation," p. 12.

[22] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques -*

SIGGRAPH '99. Not Known: ACM Press, 1999, pp. 187–194.

[23] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[24] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, and G. Medioni, "On face segmentation, face swapping, and face perception," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 98–105.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[26] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3523–3532.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 2616–2620.

[28] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1–11.

[29] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 4685–4694.

[30] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 59–66.

[31] Y. Li, J. Zeng, S. Shan, and X. Chen, "Self-Supervised Representation Learning From Videos for Facial Action Unit Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 10 916–10 925.