COMPOSITION OF MOTION FROM VIDEO ANIMATION THROUGH LEARNING LOCAL TRANSFORMATIONS

Michalis Vrigkas[⋆] *Virginia Tagka*[†] *Marina E. Plissiti*[†] *Christophoros Nikou*[†]

[⋆]Dept. Communication and Digital Media, University of Western Macedonia, 52100 Kastoria, Greece
[†]Dept. Computer Science and Engineering, University of Ioannina, 45110, Ioannina, Greece

ABSTRACT

In this work, we solve the problem of motion representation in videos, according to local transformations applied to specific keypoints extracted from static the images. First, we compute the coordinates of the keypoints of the body or face through a pre-trained model, and then we introduce a convolutional neural network to estimate a dense motion field through optical flow. Next, we train a generative adversarial network that exploits the previous information to generate new images that resemble as much as possible the target frames. To reduce trembling and extract smooth movements, our model incorporates a low-pass spatio-temporal Gaussian filter. Results indicate that our method provides high performance and the movement of objects is accurate and robust.

***Index Terms*—** Video-synthesis, Animation, GANs, Gaussian filter, Dense motion

1. INTRODUCTION

Nowadays, animation constitutes the main domain in graphics and visual effects for various applications in film and video game production. One of the most interesting tasks in animation is the implementation of models reproducing a video from static images. Traditional approaches for motion picture representation and video redefinition designed for specific areas such as human faces [1, 2, 3, 4], human silhouettes [5, 6, 7], and gestures [8] and required detailed prior knowledge of the objects that are represented. As these technologies are constantly improving, the range of possible applications is increasing [9].

In [10], realistic results were produced, based on 3D face models using optimization reconstruction algorithms. The representation of motion can also be treated as a transfer problem from one optical field to another. Isola *et al.* [11] investigated conditionally competitive neural networks in image-to-image transfer problems.

Wang *et al.* [7] relied on image-to-image transfer technique [11] and transfered human motion extending it to a video-to-video transfer problem using a GAN model. In this direction, Bansal *et al.* [12] extended these methods with conditionally generative adversarial networks (Recycle-GAN), incorporating spatio-temporal data to improve video-to-video transmission by one sector to another.

Models which do not require prior information, have also been proposed, such as X2Face [13]. This model can regulate a source face using another face as a guide-frame to create a new frame, that will have the same identity as the source frame but the posture and facial expression will be similar to the contents of the guide frame. Siarohin *et al.* [14] introduced the Monkey-Net, a self-monitoring frame for representing the motion of arbitrary objects using sparse trajectories of keypoints. In a more recent study [15], the first order motion model is proposed. Similar to Monkey-Net [14], sparse orbits of keypoints are calculated in an unsupervised manner but on the contrary with Monkey-net, the movement of objects is modeled in the neighborhood of each keypoint through local affine transformations. Furthermore, a generator is introduced, which adopts an occlusion mask to indicate the parts of the objects that may be produced through image deformation, and it also extends the loss of symmetry commonly used for feature detector training to improve the assessment of local affine transforms [16, 17].

Finally, in [18] the animation of people in clothing as a function of body pose is implemented. The learning is achieved directly from scans, complex clothing is modeled and pose-dependent details for realistic animation are produced. Extensions of the synthesis of videos for the extraction of 3D models [19] have also been reported.

In our work, we have included objects such as human bodies and faces. The motion representation is based on videos that act as guides and the images are transformed according to the movements of the objects in the corresponding frames of a driving video. Note that there is no prior information or ground-truth data for the target object, as the establishment of ground truth data is a limitation in this kind of applications. To address this problem, we investigate alternatives for the training of neural networks in an unsupervised manner. In the present work, we have developed convolutional neural networks and a generative adversarial network, which are trained to create new images that will construct the video animation of our target. Our method can produce videos of static objects that move in a similar way to the movement of another object in a video guide, with high performance.

2. METHODS

2.1. Keypoint Detection

To cope with the complex movement of the target object, we use a keypoint representation consisting of a set of keypoints and their local affine transformations. We compute the keypoints that represent specific parts of the body or face (e.g., hand, shoulder, and eyes). To this end, we used the pre-trained model of Fang *et al.* [20] that recognizes and computes the positions of multiple human bodies even if the boundary frames of the target objects are inaccurate.

The VGG SSD-512 model was used as a human detector since it has an excellent performance in object recognition. The stacked hourglass model [21] was also used as a human pose estimator and the ResNet-18 network as a tracking network.

2.2. Video Motion Composition Using Local Transformations

During training, pairs of images of $H \times W$ dimension that represents two random frames from the same video are given as input to the proposed model. One frame is denoted as the source image

$\mathbf{S} \in \mathbb{R}^{3\times H\times W}$ and the other as the driving image $\mathbf{D} \in \mathbb{R}^{3\times H\times W}$. Using the driving image as ground truth data, the network is trained to produce a new target image that represents the same motion. Next, the keypoints of the two images are used to estimate the relative motion of the pixels. The motion network is modeled by a function $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that assigns each pixel in the driving frame $\mathbf{D}$ to its corresponding position in the source frame $\mathbf{S}$. The function $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}$ is also referred as backward optical flow from the driving frame $\mathbf{D}$ to the source frame $\mathbf{S}$.

In both transformations, we obtain sparse trajectories from keypoint movements. The dense motion network combines local affine transformations to obtain the dense motion field $\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}$. Also, the motion network computes an occlusion mask $\hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}}$ which shows which parts of the frame $\mathbf{D}$ can be reconstructed by distorting the source image $\mathbf{S}$ and which parts should be inpainted based on the surrounding context. Finally, the generator computes a new image of the source image object that contains the same motion patterns as the driving video. We use a generator network G that distorts the source image according to $\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}$ and inpaints the parts of the image that are not visible in the source image. In Fig. 1 an overview of the proposed approach is presented.

The motion module computes the backward optical flow $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}$ from a driving frame $\mathbf{D}$ to a source image $\mathbf{S}$. To process frames $\mathbf{D}$ and $\mathbf{S}$ independently, we assume that there is an abstract reference frame $\mathbf{R}$. This frame is an abstract concept that is never explicitly computed. The abstract frame is used because, during the test, the model takes pairs of images of heterogeneous visual representations. To estimate the transformation $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}$, we have to estimate $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}$ and $\mathcal{T}_{\mathbf{R}\leftarrow\mathbf{D}}$. Given a frame $\mathbf{X}$, we compute each transformation $\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}$ at the keypoints. Assume that $p_1, ..., p_K$ are the coordinates of the keypoints in the reference frame $\mathbf{R}$. The motion function $\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}$ is represented by its values at each keypoint p_k:

$$\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p) = \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_k) + \left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)|_{p=p_k}\right)(p-p_k) + o(||p-p_k||)\,. \quad (1)$$

To compute the transformation $\mathcal{T}_{\mathbf{R}\leftarrow\mathbf{X}} = \mathcal{T}^{-1}_{\mathbf{X}\leftarrow\mathbf{R}}$ we assume that $\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}$ is locally one-to-one at each keypoint neighborhood. Our purpose is to estimate the transformation $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}$ near the keypoint z_k in frame $\mathbf{D}$, provided that z_k is the pixel location of the corresponding pixel to location p_k in frame $\mathbf{R}$. First, we compute the transformation $\mathcal{T}_{\mathbf{R}\leftarrow\mathbf{D}}$ near the keypoint z_k in the driving frame $\mathbf{D}$. Then, $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}$ is estimated using the Taylor expansion as follows:

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}(z) \approx \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) + (z - \mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k))\,. \quad (2)$$

In fact, the transformations $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}$ and $\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}$ represent the keypoints that have already been estimated. Thus, $p_k = \mathcal{T}_{\mathbf{R}\leftarrow\mathbf{D}}(z_k)$. Then we compute the transformation $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}$ near p_k in the reference frame $\mathbf{R}$. The transformation $\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}$ is obtained as follows:

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}} = \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}} \circ \mathcal{T}_{\mathbf{R}\leftarrow\mathbf{D}} = \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}} \circ \mathcal{T}^{-1}_{\mathbf{D}\leftarrow\mathbf{R}}\,. \quad (3)$$

2.3. Image Inpainting Under Occlusions

The source image $\mathbf{S}$ is not pixel-to-pixel aligned to the driving image $\hat{D}$. Thus, to address this misalignment, we use a distortion technique similar to [14, 22]. After applying two down-sampling convolutional blocks to the prediction estimated by the hourglass model [21], we obtain a feature map $\xi \in \mathbb{R}^{H\times W}$. Next, we distort the map ξ according to the transformation $\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}$. In the presence of occlusions in the source image $\mathbf{S}$, the optical flow may not be able to produce the image $\hat{D}$. Parts of the source image $\mathbf{S}$ cannot be recovered by simply distorting and transforming it and must be repainted.

An occlusion map $\hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}} \in [0,1]^{H'\times W'}$ is estimated that represent the pixels that should be inpainted. The occlusion map limits the effect of certain keypoints that correspond to the occluded parts.

$$\xi' = \hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}} \odot f(\xi, \hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}})\,, \quad (4)$$

where $f(\cdot,\cdot)$ denotes the distortion function and $\odot$ denotes the Hadamard product.

2.4. Training Losses

Perceptual loss: The perceptual loss [23] relies on high-order features performed by a pre-trained VGG-19 network used for image classification. During training the perceptual loss measures image similarities which are more reliable than losses estimated per pixel.

$$\mathcal{L}_{per}(\hat{\mathbf{D}}, \mathbf{D}) = \sum_{i=1}^{I} \|N_i(\hat{\mathbf{D}}) - N_i(\mathbf{D})\|\,, \quad (5)$$

where $\mathbf{D}$ is the driving frame, $\hat{\mathbf{D}}$ is the corresponding reconstructed frame, $N_i(\cdot)$ is the i-th channel feature of a specific VGG-19 layer, and I is the number of feature channels at this layer.

Keypoint matching loss: Keypoint matching loss helps to stabilize the training. Consider two images $\mathbf{I}$ and $\mathbf{J}$ which are frame images derived from the same video and $\hat{\mathbf{J}}$ the generated image.

$$\mathcal{L}_{km} = \mathbb{E}_{(\mathbf{I},\mathbf{J})}\left[\|D_i(\hat{\mathbf{J}} \oplus \mathbf{O}') - D_i(\mathbf{J} \oplus \mathbf{O}')\|_1\right]\,, \quad (6)$$

where D_i indicates the i-th layer of feature extraction in the discriminator and $\mathbf{O}'$ denotes the occlusion map.

Discrimination loss: We use the discriminator in which is given as input the occlusion map of the keypoints in the image $\mathbf{I}$ merged either with the true image $\mathbf{J}$ or the generated image $\hat{\mathbf{J}}$. The generator is trained to reconstruct the $\mathbf{J}$ frame from the coordinates of the keypoints and the $\mathbf{I}$ image.

$$\mathcal{L}^D_{GAN} = \mathbb{E}_{\mathbf{J}}\left[(1 - D(\mathbf{J} \oplus \mathbf{O}'))^2\right] + \mathbb{E}_{(\mathbf{I},\mathbf{J})}[D(\hat{\mathbf{J}} \oplus \mathbf{O}')^2\Big]\,, \quad (7)$$

where $\oplus$ indicates the union along the channel axis.

Generator loss: Similar to the discriminator loss, we compute the generator loss as follows:

$$\mathcal{L}^G_{GAN} = \mathbb{E}_{(\mathbf{I},\mathbf{J})}\left[\Big(D(\hat{\mathbf{J}} \oplus \mathbf{O}') - 1\Big)^2\right]\,. \quad (8)$$

The total loss is computed as the sum of all the individual losses:

$$\mathcal{L}_{tot} = \lambda\mathcal{L}_{per} + \lambda\mathcal{L}_{km} + \mathcal{L}^D_{GAN} + \mathcal{L}^G_{GAN}\,, \quad (9)$$

where λ controls the importance of the two terms in the total loss. In our experiments, $\lambda = 10$ according to [24].

2.5. Gaussian Filtering in the Field of Space-Time

To improve the visual results of our experiments, we incorporated in our model, after the generation module, Gaussian filtering in space and time. We used a low-pass spatio-temporal Gaussian filter on the time-axis of all generated video sequences.

$$\mathcal{G}(x,y,t) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x^2+y^2+t^2)}{2\sigma^2}}\,, \quad (10)$$

where σ is the standard deviation. The Gaussian filter removes noise and blurs the movements of the object of interest along time. It acts as a motion stabilizer by eliminating the trembling of the image and thus a smoother motion of the video objects is feasible.

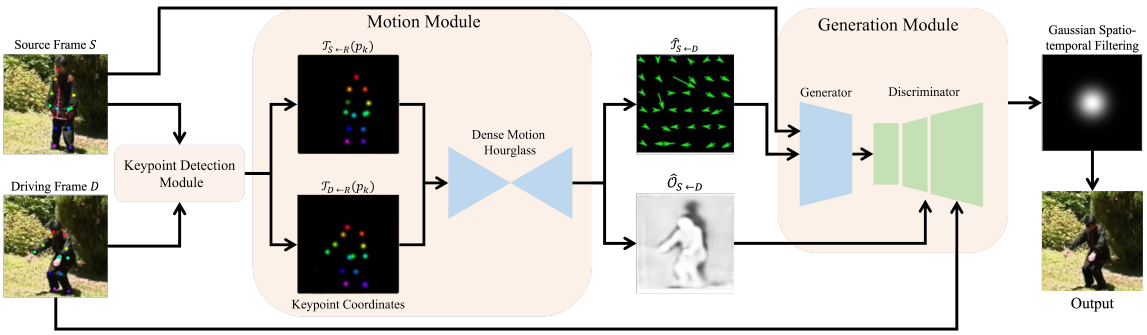

Fig. 1: Overview of the proposed motion network. A source image $\mathbf{S}$ and a frame of the driving video $\mathbf{D}$ are given as inputs to the model. The keypoint detection module computes keypoints in an unsupervised manner. The dense motion module generates the optical flow $\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}$ and estimates the occlusion map $\hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}}$. The generation module estimates the target image and a Gaussian filter is applied to the output target.

3. EXPERIMENTAL RESULTS

3.1. Datasets

Tai-Chi-HD: The Tai-Chi-HD dataset [25] includes 252 videos for training and 28 videos for testing. Some of them have split into different clips and we finally collected 1072 training videos and 86 test videos. We used 150 epochs for training and 13 keypoints.

Fashion Video: The Fashion Video dataset [26] includes 499 videos for training and 99 videos for testing. The number of epochs in training set is 100 and we computed 13 keypoints.

VoxCeleb: VoxCeleb dataset [27] contains 22, 496 videos, some of them have been split into different clips, so we finally collected 831 training videos and 506 test videos. The number of epochs in training set is 150 and 11 keypoints were computed.

3.2. Implementation Details

All our experiments were performed on an AMD Ryzen 5 2400G octa-core processor with Radeon Vega Graphics with 15.6 GB of RAM at 1517,709 MHz and an Nvidia Titan V GPU graphics card with 12 GB of memory. The Gaussian filter of size 3×3 and standard deviation $\sigma = 2.5$ was used for all datasets.

For, both the Tai-Chi-HD and VoxCeleb datasets, the total number of training epochs was 150 and for the Fashion Video dataset the total number of training epochs was 100, respectively. Also, the Adam optimizer with $2e-4$ training rate was used to train the neural network and the batch size was set to eight. Finally, we applied a decay by a factor of 0.1 in training rate after the 60th and 90th epochs.

Performance Metrics: The performance of our method is estimated following the protocol proposed by Siarohin *et al.* [15] since there are no ground-truth data. In our case, we reconstruct the input videos by combining the first video frame with the representations of the movements in each subsequent frame. The metrics that were used are i) the pixel loss (or L_1), ii) the average Euclidean distance (AED), and ii) the structural similarity index (SSIM) [28].

3.3. Comparison with State of the Art

In Table 1, we observe the results of our method, in terms of L_1, AED, and SSIM for all datasets. As it can seen, the incorporation of the Gaussian filter in our model leads to better performance.

We compared our model with two state-of-the-art models, the Monkey-Net [14] and the First Order Motion Model (FOMM) [15]

Table 1: Experimental results for Tai-Chi-HD [25], Fashion Video [26], and VoxCeleb datasets [27].

Dataset	***Metric***	***w/ GF***	***w/o GF***
Tai-Chi-HD [25]	$\boldsymbol{L_1}$ (%)	5.63	5.79
	AED (%)	15.50	15.52
	SSIM (%)	76.50	75.46
Fashion Video [26]	$\boldsymbol{L_1}$ (%)	2.26	2.48
	AED (%)	9.88	9.70
	SSIM (%)	92.38	91.52
VoxCeleb [27]	$\boldsymbol{L_1}$ (%)	5.40	5.65
	AED (%)	17.31	17.35
	SSIM (%)	82.75	80.71

Table 2: Comparison of the different models on the Tai-Chi-HD [25] and VoxCeleb [27] datasets.

Method	**Tai-Chi-HD** [25]		**VoxCeleb** [27]	
	$\boldsymbol{L_1}$ (%)	**AED** (%)	$\boldsymbol{L_1}$ (%)	**AED** (%)
Monkey-Net [14]	7.7	22.8	4.9	19.9
FOMM [15]	6.3	17.9	4.3	14.0
Ours w/o GF	5.8	15.5	5.7	17.4
Ours w/ GF	5.6	15.5	5.4	17.3

Quantitative results are reported to Table 2. We observe that our full model corresponds better than FOMM [15] in Tai-Chi-HD data set. Both the L_1 pixel loss and the AED are smaller than the corresponding ones of FOMM and Monkey-Net values. Regarding the VoxCeleb dataset, our model presents comparable results with the other two models. FOMM results were slightly better but our model exhibits better results in terms of AED, compared to Monkey-Net.

Figure 2 illustrates a qualitative comparison between the proposed and state-of-the-art methods for the Tai-Chi-HD dataset. In comparison with the competitor methods, both variants of the proposed method (i.e., w/ and w/o GF) manage to produce better visual results, where every part of the body is animated separately.

Figure 3 shows the quality results at a random instance of the videos for all three datasets. The black regions on the occlusion mask indicate the parts that cannot be retrieved from the source image and need to be inpainted. The processing time mainly depends

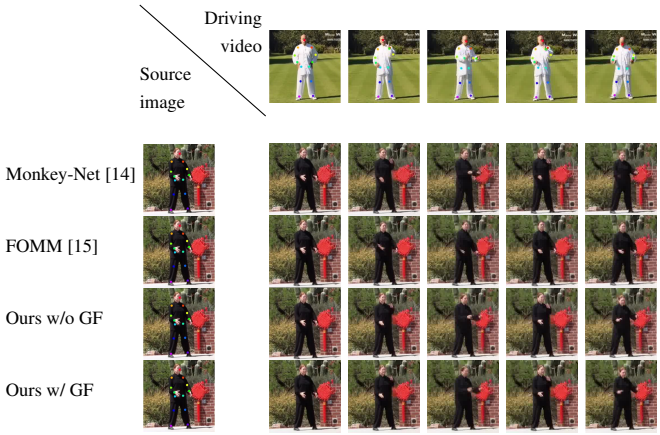

Fig. 2: Qualitative comparison with state of the art for the Tai-Chi-HD [25] dataset.

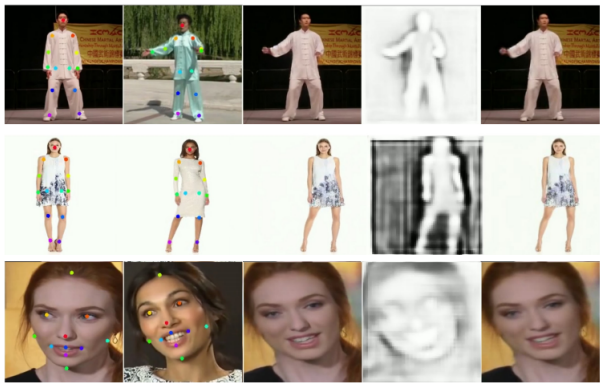

Fig. 3: Indicative qualitative results of our method. Rows (upper to lower): Tai-Chi-HD [25], Fashion Video [26], and VoxCeleb [27]. Columns (left to right): source frame, guide frame, resulted frame, occlusion mask, extracted frame with Gaussian filter.

on the size of the available dataset and the computational sources that are used for the training of the neural network. The average training time for Tai-Chi-HD and Fashion Video datasets is 120 hours and for the VoxCeleb dataset is 192 hours.

Ablation Studies: Furthermore, according to the ablation study of [15], we compared several versions of our model: (i) baseline (BL), (ii) baseline with occlusion mask (OCC), (iii) pyramid (PYR), and (iv) Full. In the first two cases, the model is trained without and with the occlusion mask respectively, with images of size 256×256. In the pyramid version, the training follows the baseline model, but with images of four different resolutions (256×256, 128×128, 64×64, 32×32). The full version combines the last three aforementioned models.

In Table 3, the ablative results for the Tai-Chi-HD dataset are contained. The best results are obtained with the full model w/ GF for $L_1 = 5.63\%$, $AED = 15.50\%$, and $SSIM = 76.50\%$. It must be noted that for the metrics L_1 and SSIM, the results are better with the incorporation of the Gaussian filter for all the models.

In Table 4, we obtain similar results for the Fashion Video data set. From the ablation study, the full model provides the best results with the incorporation of the Gaussian filter $L_1 = 2.26\%$, $AED = 9.88\%$, and $SSIM = 92.38\%$. Also, the SSIM index

Table 3: Quantitative ablation study for Tai-Chi-HD dataset [25] with (w/ GF) and without (w/o GF) Gaussian filter.

Model	L_1 (%)		AED (%)		SSIM (%)	
	w/ GF	*w/o GF*	*w/ GF*	*w/o GF*	*w/ GF*	*w/o GF*
BL	5.91	6.07	15.79	15.82	76.26	75.25
OCC	5.75	5.91	15.53	15.64	76.65	75.67
PYR	5.66	5.83	15.61	15.40	76.10	74.98
Full	5.63	5.79	15.50	15.52	76.50	75.46

Table 4: Quantitative ablation study for Fashion Video data set [26] with (w/ GF) and without (w/o GF) Gaussian filter.

Model	L_1 (%)		AED (%)		SSIM (%)	
	w/ GF	*w/o GF*	*w/ GF*	*w/o GF*	*w/ GF*	*w/o GF*
BL	2.49	2.65	10.30	12.00	92.16	91.25
OCC	2.46	2.62	9.95	11.60	92.28	91.37
PYR	2.41	2.57	9.90	9.80	92.32	91.43
Full	2.26	2.48	9.88	9.70	92.38	91.52

Table 5: Quantitative ablation study for VoxCeleb dataset [27] with (w/ GF) and without (w/o GF) Gaussian filter.

Model	L_1 (%)		AED (%)		SSIM (%)	
	w/ GF	*w/o GF*	*w/ GF*	*w/o GF*	*w/ GF*	*w/o GF*
BL	5.62	5.77	17.40	17.52	81.97	79.40
OCC	5.51	5.69	17.33	17.46	82.77	80.32
PYR	5.59	5.71	17.38	17.50	82.34	80.10
Full	5.40	5.65	17.31	17.35	82.75	80.71

is close to one, which means that images are almost identical to the target images.

Finally, in Table 5, the results in the VoxCeleb data set are shown. We may conclude that the model with the Gaussian filter outperforms the model without it, in terms of all metrics.

4. CONCLUSIONS

In this paper, we proposed a video motion representation method where the object of study can move in the same way as an object moves in a driving video. In contrast to other works (e.g., [14, 15]), our model does not require the computation of Jacobian matrices for the estimation of local transformations of the movements between the source and the driving video. Moreover, an additional step of low-pass Gaussian spatio-temporal filtering improved the results remarkably in all three datasets and also acted as a stabilizer factor leading thus to smoother motion in the output video without any flickering.

Acknowledgments: The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research. We acknowledge support of this work by the projects "Dioni: Computing Infrastructure for Big-Data Processing and Analysis" (MIS No. 5047222), which is implemented under the Action "Reinforcement of the Research and Innovation Infrastructure", and and "Bessarion" (T6YB -00214), which is implemented under the call "Open Innovation in Culture", both funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

5. REFERENCES

[1] M. C. Doukas, S. Zafeiriou, and V. Sharmanska, "Headgan: One-shot neural head synthesis and editing," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14378–14387.

[2] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10039–10049.

[3] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *arXiv*, 2019, arXiv:1905.08233.

[4] M. Vrigkas, C. Nikou, and I.A. Kakadiaris, "Exploiting privileged information for facial expression recognition," in *Proc. 9th IAPR/IEEE International Conference on Biometrics*, Halmstad, Sweden, June 2016, pp. 1–8.

[5] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," *arXiv*, 2019, arXiv:1808.07371.

[6] A. Shysheya, E. Zakharov, K. A. Aliev, R. Bashirov, E. Burkov, K. Iskakov, A. Ivakhnenko, Y. Malkov, I. Pasechnik, D. Ulyanov, A. Vakhitov, and V. Lempitsky, "Textured neural avatars," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 2019, pp. 2387–2397.

[7] T.C. Wang, M.-Y. Liu, J.Y. Zhu, G.n Liu, A. Tao, K. Jan, and B. Catanzaro, "Video-to-video synthesis," in *Proc. Conference on Neural Information Processing Systems*, Montreal, Canada, 2018, vol. 31.

[8] H. Tang, W. Wang, D. Xu, Y. Yan, and N. Sebe, "GestureGAN for hand gesture-to-gesture translation in the wild," in *Proc. 26th ACM International Conference on Multimedia*, 2018, pp. 774–782.

[9] T. Wang, N. Sarafianos, M.H. Yang, and T. Tung, "Animatable neural radiance fields from monocular RGB-D," *arXiv*, 2022.

[10] M. Zollhöfer, J. Thies, D. Bradley, P. Garrido, T. Beeler, P. Péerez, M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3D face reconstruction, tracking, and applications," *Computer Graphics Forum*, 2018.

[11] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5967–5976.

[12] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-GAN: Unsupervised video retargeting," in *Proc. European Conference in Computer Vision*, Munich, Germany, September 2018, pp. 122–138.

[13] O. Wiles, A. S. Koepke, and A. Zisserman, "X2Face: A network for controlling face generation by using images, audio, and pose codes," in *Proc. European Conference in Computer Vision*, Munich, Germany, September 2018, pp. 690–706.

[14] A. Siarohin, S. Lathuilire, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 2377–2386.

[15] A. Siarohin, S. Lathuilire, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Proc. Conference on Neural Information Processing Systems*, December 2019.

[16] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, "Unsupervised learning of object landmarks through conditional image generation," in *Proc. Conference on Neural Information Processing Systems*, Montreal, Canada, 2018.

[17] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. Metaxas, "Learning to forecast and refine residual motion for image-to-video generation," in *Proc. European Conference on Computer Vision*, 2018, pp. 387–403.

[18] G. Tiwari, N. Sarafianos, and G. Tung, T.and Pons-Moll, "Neural-gif: Neural generalized implicit functions for animating people in clothing," in *Proc. International Conference on Computer Vision*, October 2021.

[19] G. Yang, M. Vo, N. Neverova, D. Ramanan, A. Vedaldi, and H. Joo, "BANMo: Building animatable 3D neural models from many casual videos," *arXiv*, 2021, arXiv:2112.12761.

[20] H. Fang, S. Xie, Y. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE International Conference on Computer Vision*, Hawaii, HNL, October 2017, pp. 2353–2362.

[21] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. European Conference on Computer Vision*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds., 2016, pp. 483–499.

[22] A. Grigorev, A. Sevastopolsky, A. Vakhitov, and V. Lempitsky, "Coordinate-based texture inpainting for pose-guided human image generation," in *Proc. Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 12135–12144.

[23] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. European Conference on Computer Vision*, 2016, pp. 694–711.

[24] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzar, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8798–8807.

[25] Z. Geng, C. Cao, and S. Tulyakov, "3D guided fine-grained face manipulation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 9821–9830.

[26] P. Zablotskaia, A. Siarohin, B. Zhao, and L. Sigal, "DwNet: Dense warp-based network for pose-guided human video generation," *arXiv*, 2019, arXiv:1910.09139.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[28] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.