

---

# Image Animation with Keypoint Mask

---

Or Toledano<sup>1</sup> Yanir Marmor<sup>1</sup> Dov Gertz<sup>1</sup>

**Table 1. Motion transfer example:** Given a YouTube clip of a Thai-Chi artist (top), and an image of another one, our method transfers the one’s performance onto the other (bottom).



## Abstract

Motion transfer is the task of synthesizing future video frames of a single source image according to the motion from a given driving video. This task is challenging due to the complexity of motion representation and the unknown relations between the driving video and the source image. Despite this difficulty, this problem attracted great interests from researches at the recent years, with gradual improvements. The problem can be thought as decoupling of motion and appearance, which is often solved by extracting the motion from keypoint movement. We chose to tackle the generic, unsupervised setting, where we need to apply animation to any arbitrary object, without any domain specific model for the structure of the input. In this work, we extract the structure from a keypoint heatmap, without an explicit motion representation. Then, the structures from the image and the video are extracted to warp the image according to the video, by a deep generator.

## 1. Introduction

Take a look at Figure 1 for two video sequences. The input is a YouTube clip of a Thai-Chi artist (the source subject)

<sup>1</sup>Tel Aviv University. Correspondence to: Or Toledano <ortoledano@protonmail.com>, Yanir Marmor <yanirmr@gmail.com>, Dov Gertz <dovgertz1@gmail.com>.

performing a series of complex motions in the top row. Our algorithm’s output is shown in the bottom row. It refers to frames that appear to show a different person (the target subject) executing the same motions. The twist is that the target individual has never performed the same exact sequence of motions as the driver video. In reality, he was photographed another movement with no clear reference to the source’s actions. And, as the figure shows, the source and the target are of different races, have different backgrounds, and dress differently.

In this study, we propose a simple, however surprisingly efficient approach to video retargeting – the transfer of the movement automatically from driving video sequence through an source image for a new artificial video. Two inputs – image of the target person whose appearance we want to summarize and the other one of the source theme, the motion of which we want to impose on our target person – allow us to transfer movement between them by learning a compact motion representation. With our framework, we create a variety of videos that never even happen.

A lot of researches observe that the keypoint-based pose preserves motion signatures over time, while abstracting all possible subject identities as much as possible. We therefore use the keypoint-based pattern without motion priors. We took the keypoints and use them for animate images such a way that doesn’t need any external information about the subject or any assumption and prior about what is happening in the scene.

The main challenges we need to deal with, as they defined in (Mirsky & Lee, 2021) are generalization, temporal coherence, and occlusions.

Our paper focuses on this motion transfer problem: given a source image  $S$  and a driving video  $D$ , the goal is to synthesize a video with the identity of  $S$ , and the motion from  $D$ .

Our contribution is twofold: first, we demonstrate that removing the motion prior will function and provide better accuracy outcomes, and second, we provide a compact representation with improved efficiency.

### 1.1. Related Work:

Motion transfer has gained a lot of coverage over the last twenty years. Early approaches based on manipulating existing video footage to generate new content. They searched for frames in which the body position corresponded to the desired motion and use them to generate a new content(Bregler et al., 1997). Our approach is equally designed for videos, but rather than manipulating existing images, we learn to synthesize new movements that never was done.

A number of techniques are based on calibrated multi camera systems to scan a target player and use an adapted 3D model of the target to control their motion in a new frame(Cheung et al., 2004). Our solution instead examines the transition of movement between 2D video subjects and prevents data calibration or 3D space information.

Deep learning for reanimation has been used in many implementations in recent works and relies on more accurate input representations. Due to the synthetic rendering, an interior model and a gaze map as an input. They manage to transfer head position and facial expressions among human subjects and to produce detailed portrait videos of their facial gestures(Kim et al., 2018). Our problem is similar to that, except that our is full body movement redirected and the inputs to our model are 2D video and an image. No external data replication is used.

Latest approaches concentrate on the disentangling of appearance and activity and synthesizing of new motion videos (Tulyakov et al., 2018). Similarly, we apply our representation of motion to different target subjects to generate new motions. However, In contrast to these works, we did not use GAN but encoder-decoder approach.

In comparison to the pixel-based approaches for image animation that were common until recently, keypoint-based approaches are now thought to have the ability to achieve high performance in the field of video reanimation. Some notable works in this area are (Siarohin et al., 2020), (Wiles et al., 2018), (Siarohin et al., 2019), (Kim et al., 2019), (Balakrishnan et al., 2018), (Ma et al., 2017), (Reed et al., 2017), (Chan et al., 2019)

Like (Siarohin et al., 2020), during training, our keypoint predictor does not require any keypoint annotations. This will result in erratic results. One of the most important forces motivating the exploration of unsupervised keypoints is the equivariance loss. It forces the model to predict keypoints that are consistent with known geometric transformations.

In contrast to (Siarohin et al., 2020), our work does not depend on a strong motion prior directly, but rather on a structure mask derived from a keypoint detector of a motion-based model, such as (Siarohin et al., 2020). The idea of

using drawn keypoints as a geometry representation (structural mask) was already used in image-to-image translation works such as TransGaGa (Wu et al., 2019).

The concept of using a structural mask in the context of image animation is demonstrated in (Shalev & Wolf, 2020). However, the current work differs by basing the mask off a motion related module, which saves us the hassle of perturbing the input hoping to achieve an identity-less mask which might not even be optimal. That way, we can base more of the motion representation on the deep network, reduce our prior, and leave some space for our network to achieve better results.

## 2. Methodology

The network can be divided into two parts: obtaining the mask, and generating the synthesized frame. The generator architecture is constant amongst all of our variations, and can be described as low resolution generation from the source image, source mask and driving mask; Followed by up-scaling of the low scale synthesized prediction by passing it with the source frame to a high resolution generator. We created two different versions for the mask generator: the first, an absolute motion transfer; the second, a relative motion transfer. We chose to present the results for the first, absolute one, due to its performance. The performance drop for our second, relative version was expected because the mask contained less structural information. The first version is referred as a "keypoint heatmap," while the second is referred as a "circles mask" or "keypoints after softmax".

### 2.1. First mask version for absolute motion transfer, with warping

Our main mask for the project is obtained from carefully observing the keypoint module from (Siarohin et al., 2020). U-Net based keypoint modules of this form, work by extracting features, which pass through a *conv* layer to form a heatmap image  $K$  channels, where  $K$  is the number of keypoints. Then, a softmax is performed over the channels, and keypoints are extracted from the mean location of each heatmap channel. By carefully debugging the code, and the expectation for a segmentation map out of U-Net, we obtain Figure 1.

Summing over the channels, we get Figure 2.

which is our output mask, aka the heatmap, pre softmax mask.

### 2.2. Second mask version for relative motion transfer

We purposes an additional "circles" only mask which can be used in the context of relative motion transfer during animation, as in (Siarohin et al., 2020), which isn't possible

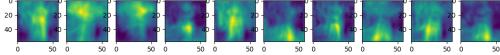


Figure 1.  $K$  channels of the keypoint detector network used in (Siarohin et al., 2020), before the softmax activation. Our main motion prior in this project.

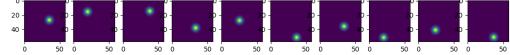


Figure 3.  $K$  channels of the keypoint detector network used in (Siarohin et al., 2020), after the softmax activation and Gaussian fit.

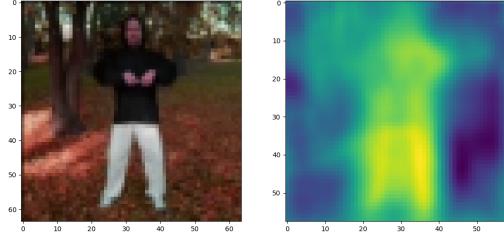


Figure 2. The sum of the  $K$  channels which is fed as a structural mask into the generator.

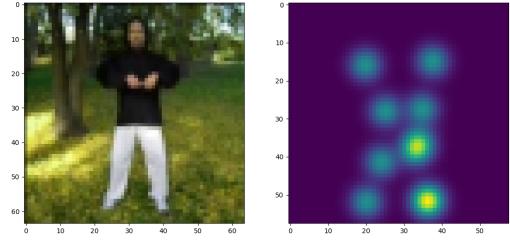


Figure 4. The sum of the  $K$  channels which is fed as a structural mask into the generator.

with the previous heatmap mask. The mask captures the image’s geometry representation (Wu et al., 2019), and by requiring it to be represented as keypoints with a center, we can use the relative coordinates for the animation. This module, though, did not do as well as our heatmap mask module (Table 4).

While relative motion transition is not always desired, this work shows that a keypoint-only-prior-based module is feasible for the task. Since the only information contained in the pair of masks is the keypoint displacement, our deep network can only attempt to approximate a zero order approximation, we can anticipate results that are more close to (Siarohin et al., 2019).

By taking the softmax over the heatmap, we get Table 3.

Summing over the channels, we get Table 4.

which is our output mask, aka the keypoints after softmax mask.

### 2.3. Architecture

Our architecture follows (Siarohin et al., 2020) without the dense motion module, after changing its keypoint generation module to return our mask. After the mask is obtained, we follow the encoder decoder approach as (Shalev & Wolf, 2020). Namely, the encoder of the low resolution generator consists of  $conv_{7 \times 7}$ ,  $batch\_norm$ ,  $relu$ , followed by six residual blocks of  $batch\_norm$ ,  $relu$ ,  $conv_{3 \times 3}$ ,  $batch\_norm$ ,  $relu$ ,  $conv_{3 \times 3}$ , (and a sum with the source). The residual blocks help to maintain the identity of the source image (He et al., 2015). The decoder consists of two blocks, each is a sequence of  $up\_sample_{2 \times 2}$ ,  $batch\_norm$ ,  $relu$ . The decoder is followed by a  $conv_{7 \times 7}$  and a  $sigmoid$  activation. For the high resolution generator, use an encoder (decoder) with five encoding (decoding) blocks, where each block is a sequence of  $conv_{3 \times 3}$ ,  $batch\_norm$ ,  $relu$   $avg\_pool_{2 \times 2}$ , and each decoding block is a sequence of  $up\_sample_{2 \times 2}$ ,  $conv_{3 \times 3}$ ,  $batch\_norm$ ,  $relu$ . We add skip connections from each of the encoding layers to its corresponding encoding layer, to form a U-Net architecture (Ronneberger et al., 2015).

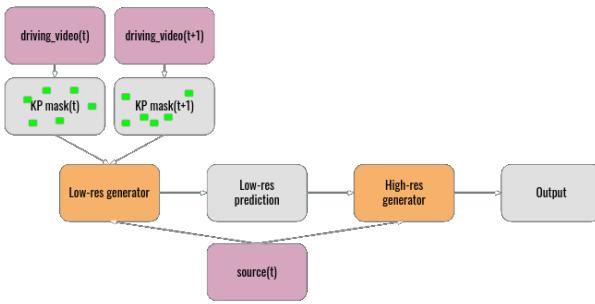


Figure 5. Architecture of the model. The keypoint mask can be either a keypoint heatmap such as in Figure 2, or drawn keypoint circles as in Figure 4.

## 2.4. Losses

We use the same perceptual loss as in (Siarohin et al., 2020) which is based on the implementation of (Wang et al., 2018). With the input driving frame  $D$  and the corresponding reconstructed frame  $\hat{D}$ , the reconstruction loss is written as:  $L_{rec}(\hat{D}, D) = \sum_{i=1}^I |N_i(\hat{D}) - N_i(D)|$ , where  $N_i(\cdot)$  is the  $i^{th}$  channel feature extracted from a specific VGG-19 layer (Simonyan & Zisserman, 2015) and  $I$  is the number of feature channels in this layer. Additionally we use this loss on a number of resolutions, forming a pyramid obtained by down-sampling  $\hat{D}$  and  $D$ , similarly to ((), ()). The resolutions are  $256 \times 256$ ,  $128 \times 128$ ,  $64 \times 64$  and  $32 \times 32$ .

## 3. Experiments

### 3.1. Datasets

The Tai-chi-HD dataset, which includes brief videos of people doing Tai-chi exercises, was used for training and evaluation. Following (Siarohin et al., 2020), 3,141 Tai-chi videos were downloaded from YouTube. The videos were cropped and resized to a resolution of  $256^2$ , while the aspect ratio was preserved. There are 3,016 training videos and 125 evaluation videos.

### 3.2. Comparison with Previous Works

In order to compare our work to previous works (Table 4) we used metrics previously used in similar papers. Average Keypoints Distance (Cao et al., 2017) (AKD) measures the average key-points distance between the generated video and the source video. Average Euclidean Distance (Zheng et al., 2019) (AED) measures the average euclidean distance between the representations of the ground-truth and generated videos in some embedding space. In addition, we added the L1 distance as well. Our AED and AKD metrics were calculated using the following repository: <https://github.com/AliaksandrSiarohin/pose-evaluation>

Table 2. Images comparison

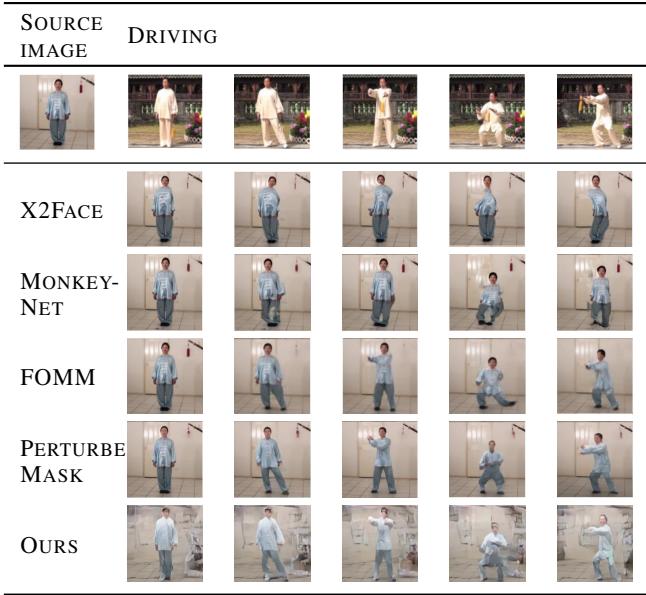


Table 3. Images comparison 2



[com/AliaksandrSiarohin/pose-evaluation](https://github.com/AliaksandrSiarohin/pose-evaluation)  
Note that those metrics aren't optimal since one can easily improve reconstruction by increasing the bottleneck, and we can see our artifacts in animation results (Table 2). However, our pose did follow the driving video well, and to improve the identity and background artifacts, we suggest some fixes in Section 4.

## 4. Future work

We would like to test our module on more datasets, and compare them to the state of the art. In addition, summing the heatmap channels might not be optimal, and there is certainly some space to try something deeper with the features extracted in the keypoint detector as an input, or feed all of

Table 4. Accuracy Metrics

METHOD	AKD	AED	L1
X2FACE	17.654	0.272	0.080
MONKEY-NET	10.798	0.228	0.077
FOMM	6.872	0.167	0.063
PERTURBED MASK	4.239	0.147	0.047
OURS (CIRCLES MASK)	14.760	0.245	0.077
OURS	5.551	0.141	0.045
IMPROVEMENT (FOMM)	19.2%	15.5%	28.5%

the channels separately into the generator. We want to experiment with mask thresholds due to the distortions in the animation background. We may also increase the number of keypoints, but that would probably be more beneficent to the second type of mask, and would increase the required GPU memory proportionally. We also thought about coloring matching keypoints with the same color to help the module to learn a motion flow.

## 5. Conclusions

We constructed a novel method for image animation by moving the need for a strong motion prior (optical flow) to the assumption of a pre-trained keypoint detector/keypoint heatmaps prior to activation, which might be based on a motion prior. By doing so, we encapsulated motion to a motion mask, which is bottlenecked by the prior training which has the keypoint bottleneck. The motion masks are then fed into a generator, which combines the appearance of the source image and the mask which represents the structure, decoupled from any appearance naturally by the assumption that during the training of the keypoint detector, the heatmap mask went into a keypoint bottleneck. After evaluation, we can conclude that our method is feasible, although with some artifacts.

## Software and Data

Detailed in our repository:

<https://github.com/or-toledano/animation-with-keypoint-mask>

## References

Balakrishnan, G., Zhao, A., Dalca, A. V., Durand, F., and Guttag, J. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8340–8348, 2018.

Bregler, C., Covell, M., and Slaney, M. Video rewrite: Driving visual speech with audio. In *Proceedings of*

*the 24th annual conference on Computer graphics and interactive techniques*, pp. 353–360, 1997.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chan, C., Ginosar, S., Zhou, T., and Efros, A. A. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5933–5942, 2019.

Cheung, G. K., Baker, S., Hodgins, J., and Kanade, T. Markerless human motion transfer. In *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, pp. 373–378. IEEE, 2004.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., Pérez, P., Richardt, C., Zollhöfer, M., and Theobalt, C. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

Kim, Y., Nam, S., Cho, I., and Kim, S. J. Unsupervised keypoint learning for guiding class-conditional video prediction. *arXiv preprint arXiv:1910.02027*, 2019.

Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., and Van Gool, L. Pose guided person image generation. *arXiv preprint arXiv:1705.09368*, 2017.

Mirsky, Y. and Lee, W. The creation and detection of deepfakes. *ACM Computing Surveys*, 54(1):1–41, Mar 2021. ISSN 1557-7341. doi: 10.1145/3425780. URL <http://dx.doi.org/10.1145/3425780>.

Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Chen, Y., Belov, D., and Freitas, N. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, pp. 2912–2921. PMLR, 2017.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

Shalev, Y. and Wolf, L. Image animation with perturbed masks, 2020.

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., and Sebe, N. Animating arbitrary objects via deep motion transfer, 2019.

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., and Sebe, N. First order motion model for image animation, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1526–1535, 2018.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. Video-to-video synthesis, 2018.

Wiles, O., Koepke, A. S., and Zisserman, A. X2face: A network for controlling face generation by using images, audio, and pose codes, 2018.

Wu, W., Cao, K., Li, C., Qian, C., and Loy, C. C. Transgaga: Geometry-aware unsupervised image-to-image translation, 2019.

Zheng, Z., Yang, X., Yu, Z., Zheng, L., Yang, Y., and Kautz, J. Joint discriminative and generative learning for person re-identification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.