Motion Representations for Articulated Animation

Aliaksandr Siarohin[1*], Oliver J. Woodford[*], Jian Ren[2], Menglei Chai[2] and Sergey Tulyakov[2]
[1]DISI, University of Trento, Italy, [2]Snap Inc., Santa Monica, CA
aliaksandr.siarohin@unitn.it, {jren,mchai,stulyakov}@snap.com, *Work done while at Snap Inc.

Abstract

We propose novel motion representations for animating articulated objects consisting of distinct parts. In a completely unsupervised manner, our method identifies object parts, tracks them in a driving video, and infers their motions by considering their principal axes. In contrast to the previous keypoint-based works, our method extracts meaningful and consistent regions*, describing locations, shape, and pose. The regions correspond to semantically relevant and distinct object parts, that are more easily detected in frames of the driving video. To force decoupling of foreground from background, we model non-object related global motion with an additional affine transformation. To facilitate animation and prevent the leakage of the shape of the driving object, we disentangle shape and pose of objects in the region space. Our model[1] can animate a variety of objects, surpassing previous methods by a large margin on existing benchmarks. We present a challenging new benchmark with high-resolution videos and show that the improvement is particularly pronounced when articulated objects are considered, reaching 96.6% user preference vs. the state of the art.*

1. Introduction

Animation—bringing static objects to life—has broad applications across education and entertainment. Animated characters and objects, such as those in Fig. 1, increase the creativity and appeal of content, improve the clarity of material through storytelling, and enhance user experiences.

Until very recently, animation techniques necessary for achieving such results required a trained professional, specialized hardware, software, and a great deal of effort. Quality results generally still do, but vision and graphics communities have attempted to address some of these limitations by training data-driven methods [36, 6, 25, 11, 10] on object classes for which prior knowledge of object shape and pose can be learned. This, however, requires ground truth pose and shape data to be available during training.

Recent works have sought to avoid the need for ground truth data through *unsupervised* motion transfer [39, 27, 28]. Significant progress has been made on several key challenges, including training using image reconstruction as a loss [39, 27, 28], and disentangling motion from appearance [19]. This has created the potential to animate a broader range of object categories, without any domain knowledge or labelled data, requiring only videos of objects in motion during training [27]. However, two key problems remain open. The first is how to represent the parts of an articulated or non-rigid moving object, including their shapes and poses. The second is given the object parts, how to animate them using the sequence of motions in a driving video.

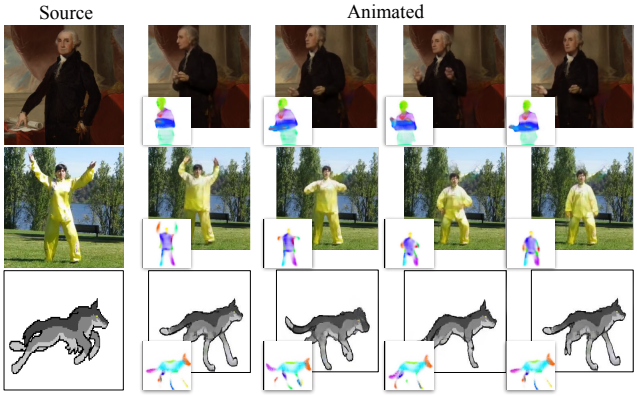


Figure 1: Our method animates still source images via unsupervised region detection (inset).

Initial attempts used end-to-end frameworks [39, 28] to first extract unsupervised keypoints [19, 17], then warp a feature embedding of a source image to align its keypoints with those of a driving video. Follow on work [27] further modelled the motion around each keypoint with local, affine transformations, and introduced a generation module that both composites warped source image regions and inpaints occluded regions, to render the final image. This enabled a variety of creative applications[2], for example needing only one source face image to generate a near photo-realistic animation, driven by a video of a different face.

However, the resulting unsupervised keypoints are de-

[1]Our source code is publicly available at https://github.com/snap-research/articulated-animation.

[2]E.g. a music video in which images are animated using prior work [27].

tected on the boundary of the objects. While points on edges are easier to identify, tracking such keypoints between frames is problematic, as any point on the boundary is a valid candidate, making it hard to establish correspondences between frames. A further problem is that the unsupervised keypoints do not correspond to semantically meaningful object parts, representing location and direction, but not shape. Due to this limitation, animating articulated objects, such as bodies, remains challenging. Furthermore, these methods assume static backgrounds, i.e. no camera motion, leading to leakage of background motion information into one or several of the detected keypoints. Finally, absolute motion transfer, as in [27], transfers the shape of the driving object into the generated sequence, decreasing the fidelity of the source identity. These remaining deficiencies limit the scope of previous works [27, 28] to more trivial object categories and motions, especially when objects are articulated.

This work introduces three contributions to address these challenges. First, we redefine the underlying motion representation, using *regions* from which first-order motion is *measured*, rather than regressed. This enables improved convergence, more stable, robust object and motion representations, and also empirically captures the shape of the underpinning object parts, leading to better motion segmentation. This motion representation is inspired by Hu moments [8]. Fig. 3(a) contains several examples of region vs. keypoint-based motion representation.

Secondly, we explicitly model background or camera motion between training frames by predicting the parameters of a global, affine transformation explaining non-object related motions. This enables the model to focus solely on the foreground object, making the identified points more stable, and further improves convergence. Finally, to prevent shape transfer and improve animation, we disentangle the shape and pose of objects in the space of unsupervised regions. Our framework is self-supervised, does not require any labels, and is optimized using reconstruction losses.

These contributions further improve unsupervised motion transfer methods, resulting in higher fidelity animation of articulated objects in particular. To create a more challenging benchmark for such objects, we present a newly collected dataset of TED talk speakers. Our framework scales better in the number of unsupervised regions, resulting in more detailed motion. Our method outperforms previous unsupervised animation methods on a variety of datasets, including talking faces, taichi videos and animated pixel art being preferred by 96.6% of independent raters when compared with the state of the art [27] on our most challenging benchmark.

2. Related work

Image animation methods can be separated into supervised, which require knowledge about the animated object during training, and unsupervised, which do not. Such knowledge typically includes landmarks [4, 40, 24, 12], semantic segmentations [22], and parametric 3D models [11, 32, 9, 20, 18]. As a result, supervised methods are limited to a small number of object categories for which a lot of labelled data is available, such as faces and human bodies. Early face reenactment work [32] fitted a 3D morphable model to an image, animating and rendering it back using graphical techniques. Further works used neural networks to get higher quality rendering [16, 38], sometimes requiring multiple images per identity [11, 23]. A body of works treats animation as an image-to-image [30] or video-to-video [36, 6, 25] translation problem. Apart from some exceptions [35], these works further constrain the problem to animating a single instance of an object, such as a single face [16, 2] or a single human body [6, 25, 36], requiring retraining [2, 6, 25] or fine-tuning [40] for each new instance. Despite promising results, generalizing these methods beyond a limited range of object categories remains challenging. Additionally, they tend to transfer not only the motion but also the shape of the driving object [16, 40].

Unsupervised methods address some of these limitations. They do not require any labelled data regarding the shape or landmarks of the animated object. Video-generation-based animation methods predict future frames of a video, given the first frame and an animation class label, such as "make a happy face", "do jumping jack", or "play golf" [33, 26, 7]. A further group of works re-target animation from a driving video to a source frame. X2Face [39] builds a canonical representation of an input face, and generates a warp field conditioned on the driving video. Monkey-Net [28] learns a set of unsupervised keypoints to generate animations. Follow-up work substantially improves the quality of animation by considering a first order motion model (FOMM) [27] for each keypoint, represented by regressing a local, affine transformation. Both of these works apply to a wider range of objects including faces, bodies, robots, and pixel art animations. Empirically, these methods extract keypoints on the boundary of the animated objects. Articulated objects such as human bodies are therefore challenging, as internal motion, for example, an arm moving across the body, is not well modeled, producing unconvincing animations.

This work presents an unsupervised method. We argue that the limitations of previous such methods in animating articulated objects is due to an inability of their internal representations to capture complete object parts, their shape and pose. X2Face [39] assumes an object can be represented with a single RGB texture, while other methods find keypoints on edges [28, 27]. Our new region motion representation resembles the construction of a motion history image whose shape is analyzed using principal components (namely Hu moments [8]). In [8] the authors construct motion descriptors by computing temporal image differences, aggregate them into motion history images, and use Hu moments to build a motion recognition system. In this manner, blob

statistics are used to discriminate between different actions.

3. Method

We propose three contributions over FOMM [27], namely our PCA-based motion estimation (Sec. 3.2), background motion representation (Sec. 3.3) and animation via disentanglement (Sec. 3.6). To make the manuscript self-contained, we first give the necessary technical background on the original first order motion representation [27] (Sec. 3.1).

3.1. First Order Motion Model

FOMM [27] consists of the two main parts: motion estimation and image generation, where motion estimation further contains coarse motion estimation and dense motion prediction. Coarse motion is modelled as sparse motions between separate object parts, while dense motion produces an optical flow along with the confidence map for the entire image. We denote by $\mathbf{S}$ and $\mathbf{D}$ the source and the driving frames extracted from the same video respectively.

The first step is to estimate coarse motions from $\mathbf{S}$ and $\mathbf{D}$. Motions for each object part are represented by affine transformations, $A^k_{\mathbf{X}\leftarrow\mathbf{R}} \in \mathcal{R}^{2\times 3}$, to an abstract, common reference frame, $\mathbf{R}$; $\mathbf{X}$ is either $\mathbf{S}$ or $\mathbf{D}$. Motions are estimated for K distinct parts. An encoder-decoder *keypoint* predictor network outputs K heatmaps, $\mathbf{M}^1, .., \mathbf{M}^K$ for the input image, followed by softmax, s.t. $\mathbf{M}^k \in [0,1]^{H\times W}$, where H and W are the height and width of the image respectively, and $\sum_{z\in\mathcal{Z}} \mathbf{M}^k(z) = 1$, where z is a pixel location (x, y coordinates) in the image, the set of all pixel locations being $\mathcal{Z}$, and $\mathbf{M}^k(z)$ is the k-th heatmap weight at pixel z. Thus, the translation component of the affine transformation (which is the last column of $A^k_{\mathbf{X}\leftarrow\mathbf{R}}$) can be estimated using softargmax:

$$\mu^k = \sum_{z\in\mathcal{Z}} \mathbf{M}^k(z) z. \tag{1}$$

In FOMM [27] the remaining affine parameters are regressed per pixel and form 4 additional channels $P^k_{ij} \in \mathcal{R}^{H\times W}$, where $i \in \{0,1\}, j \in \{0,1\}$ indexes of the affine matrix $A^k_{\mathbf{X}\leftarrow\mathbf{R}}$. The latter is estimated using weighted pooling:

$$A^k_{\mathbf{X}\leftarrow\mathbf{R}}[i,j] = \sum_{z\in\mathcal{Z}} \mathbf{M}^k(z) P^k_{ij}(z). \tag{2}$$

We refer to this way of computing motion as *regression-based*, where affine parameters are *predicted* by a network and pooled to compute $A^k_{\mathbf{X}\leftarrow\mathbf{R}}$. Motion between $\mathbf{D}$ and $\mathbf{S}$ for part k is then computed via the common reference frame:

$$A^k_{\mathbf{S}\leftarrow\mathbf{D}} = A^k_{\mathbf{S}\leftarrow\mathbf{R}} \begin{bmatrix} A^k_{\mathbf{D}\leftarrow\mathbf{R}} \\ 0\ 0\ 1 \end{bmatrix}^{-1}. \tag{3}$$

Given the coarse motion, the next step is to predict the optical flow and the confidence map. Since the transformations between $\mathbf{D}$ and $\mathbf{S}$ are known for each part, the task is to combine them to obtain a single optical flow field. To this end, the flow predictor selects the appropriate coarse transformation for each pixel, via a weighted sum of coarse motions. Formally, $K+1$ assignment maps are output, one per region, plus background, which FOMM [27] assumes is *motionless*, and softargmax is applied pixelwise across them, s.t. $\mathbf{W}^k \in [0,1]^{H\times W}$, $\mathbf{W}^0$ corresponds to the background, and $\sum_{k=0}^{K} \mathbf{W}^k(z) = 1,\ \forall z$. Optical flow per pixel, $\mathbf{O}(z) \in \mathbb{R}^2$, is then computed as:

$$\mathbf{O}(z) = \mathbf{W}^0(z) z + \sum_{k=1}^{K} \mathbf{W}^k(z) A^k_{\mathbf{S}\leftarrow\mathbf{D}} \begin{bmatrix} z \\ 1 \end{bmatrix}. \tag{4}$$

With such a model, animation becomes challenging when there is even slight background motion. The model automatically adapts by assigning several of the available keypoints to model background as shown in the first row of Fig. 3(a).

A confidence map, $\mathbf{C} \in [0,1]^{H\times W}$, is also predicted using the same network, to handle parts missing in the source image. Finally $\mathbf{S}$ is passed through an encoder, followed by warping the resulting feature map using the optical flow (Eq. 4) and multiplied by the confidence map. A decoder then reconstructs the driving image $\mathbf{D}$.

At test time FOMM [27] has two modes of animating $\mathbf{S}$: *standard* and *relative*. In both cases the input is the source image $\mathbf{S}$ and the driving video $\mathbf{D}_1, \mathbf{D}_2, .., \mathbf{D}_t, .., \mathbf{D}_T$. In the *standard* animation the motion between source and driving is computed frame-by-frame using Eq. (3). For *relative* animation, in order to generate a frame t the motion between $\mathbf{D}_1$ and $\mathbf{D}_t$ is computed first and then applied to $\mathbf{S}$. Both of these modes are problematic when the object in question is articulated, as we show in Sec. 3.6.

3.2. PCA-based motion estimation

Accurate motion estimation is the main requirement for high-quality image animation. As mentioned previously, FOMM regresses the affine parameters. This requires higher capacity networks, and generalizes poorly (see Sec. 4.1). We propose a different motion representation: *all* motions are *measured* directly from the heatmap $\mathbf{M}^k$. We compute the translation as before, while in-plane rotation and scaling in x- and y-directions are computed via a principal component analysis (PCA) of the heatmap $\mathbf{M}^k$. Formally, the transformation $A^k_{\mathbf{X}\leftarrow\mathbf{R}} \in \mathbb{R}^{2\times 3}$, of the k^{th} region from the reference frame to the image is computed as:

$$\mu^k = \sum_{z\in\mathcal{Z}} \mathbf{M}^k(z) z, \tag{5}$$

$$U^k S^k V^k = \sum_{z\in\mathcal{Z}} \mathbf{M}^k(z) \left(z - \mu^k\right) \left(z - \mu^k\right)^{\mathsf{T}} \text{ (SVD)}, \tag{6}$$

$$A^k_{\mathbf{X}\leftarrow\mathbf{R}} = \left[U^k S^{k\,\frac{1}{2}}, \mu^k\right]. \tag{7}$$

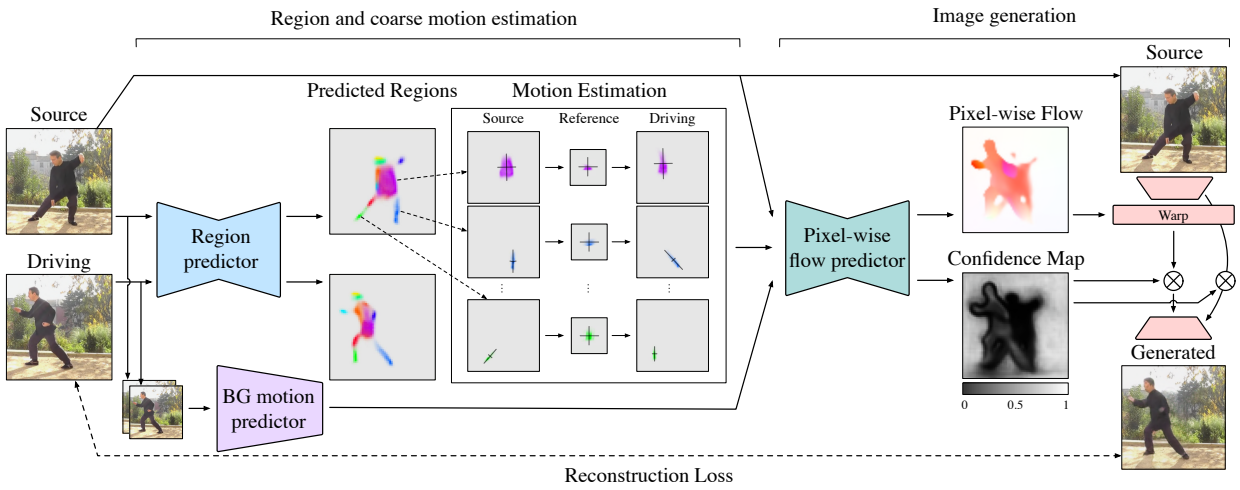


Figure 2: **Overview of our model.** The region predictor returns heatmaps for each part in the source and the driving images. We then compute principal axes of each heatmap, to transform each region from the source to the driving frame through a whitened reference frame. Region and background transformations are combined by the pixel-wise flow prediction network. The target image is generated by warping the source image in a feature space using the pixel-wise flow, and inpainting newly introduced regions, as indicated by the confidence map.

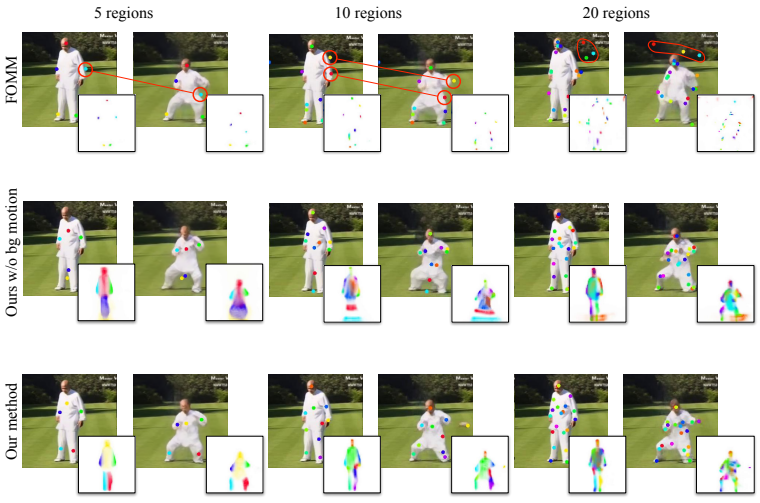
Here the singular value decomposition (SVD) approach to computing PCA [34] is used, Eq. (6) decomposing the covariance of the heatmap into unitary matrices U^k and V^k, and S^k, the diagonal matrix of singular values. We call this approach *PCA-based*, in contrast to *regression-based* for Eq. (1). Despite using the same region representation and encoder here, the encoded regions differ significantly (see Fig. 3(a)), ours mapping to meaningful object parts such as the limbs of an articulated body, due to our novel foreground motion representation, described above. Note that in our *PCA-based* approach, shear is not captured, therefore our transform is not fully affine, with only five degrees of freedom instead of six. Nevertheless, as we later show empirically, it captures sufficient motion, with shear being a less significant component of the affine transform for this task. The reference frame in both is used only as an intermediate coordinate frame between the source and driving image coordinate frames. However, here (in contrast to FOMM) it is not in fact abstract, corresponding to the coordinate frame where the heatmap is whitened (i.e. has zero mean and identity covariance); see Fig. 2. $A^k_{\mathbf{S}\leftarrow\mathbf{D}}$ is computed per Eq. (3).

3.3. Background motion estimation

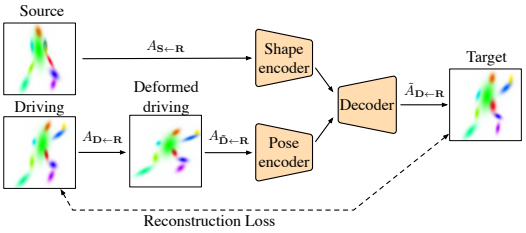
Background occupies a large portion of image. Hence even small background motion between frames, e.g. due to camera motion, negatively affects the animation quality. FOMM [27] does not treat background motion separately, therefore must model it using keypoints. This has two negative consequences: (i) additional network capacity is required, since several keypoints are used to model the background instead of the foreground; (ii) overfitting to the training set, since these keypoints focus on specific parts of the background, which may not appear in the test set. Hence, we additionally predict an affine background transformation, $\mathbf{A}^{\mathbf{0}}_{\mathbf{S}\leftarrow\mathbf{D}}$, using an encoder network assuming $\mathbf{S}$ and $\mathbf{D}$ as input and predicting six real values, $a_1, .., a_6$, such that $\mathbf{A}^{\mathbf{0}}_{\mathbf{S}\leftarrow\mathbf{D}} = \left[\begin{smallmatrix} a_1, a_2, a_3 \\ a_4, a_5, a_6 \end{smallmatrix}\right]$. Since our framework is unsupervised, the background network can include parts of the foreground into the background motion. In practice this does not happen, since it is easier for the network to use a more appropriate *PCA-based* motion representation for the foreground. It is also simpler for the network to use $\mathbf{S}$ and $\mathbf{D}$ to predict background movement, instead of encoding it in the heatmaps modelling the foreground. We verify this empirically, demonstrating that the proposed motion representations can separate background and foreground in a completely unsupervised manner (see Fig. 6 and Sec. 4.5 for comparisons).

3.4. Image generation

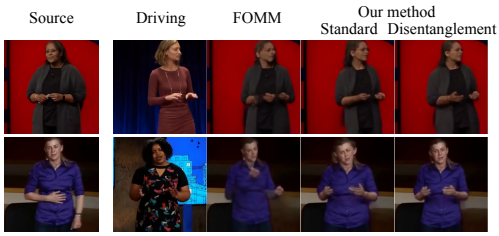
Similarly to FOMM [27], we render the target image in two stages: a pixel-wise flow generator converts coarse motions to dense optical flow, then the encoded features of the source are warped according to the flow, followed by inpainting the missing regions. The input of the dense flow predictor is a $H \times W \times (4K + 3)$ tensor, with four channels per region, three for the source image warped according to the region's affine transformation, and one for a heatmap of the region, which is a gaussian approximation of $\mathbf{M}^k$, and a further three channels for the source image warped according to the background's affine transformation. In contrast to FOMM, which uses constant variances, we estimate covariances from heatmaps. Our dense optical flow is given



(a) Comparisons of motion/part representations



(b) Transformation-based shape and pose disentanglement



(c) Comparisons of FOMM, standard and animation via disentanglement

Figure 3: **Motion representation and disentanglement.** (a) A comparison of part regions estimated by regression-based (FOMM [27]) and PCA-based (our method) motion representations. Each row shows a generated sequence along with the detected keypoints and region heatmaps (inset). (b) Framework for animating motion driving frame whilst retaining shape from the source frame, by disentangling shape and pose from motion representations. (c) Qualitative animation results of articulated objects, using FOMM [27], and our method using standard and disentangled motion transfer (sec. 3.6).

	5 regions			10 regions			20 regions		
	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	(AKD, MKR)	AED
FOMM [27]	0.062	(7.34, 0.036)	0.181	0.056	(6.53, 0.033)	0.172	0.062	(8.29, 0.049)	0.196
Ours w/o bg	0.061	(6.67, 0.030)	0.175	0.059	(**5.55**, **0.026**)	0.165	0.057	(5.47, **0.026**)	0.155
Ours	**0.049**	(**6.04**, **0.029**)	**0.162**	**0.047**	(5.59, 0.027)	**0.152**	**0.046**	(**5.17**, **0.026**)	**0.141**

Table 1: Comparing our model with FOMM [27] on TaiChiHD (256), for $K = 5$, 10 and 20. (Best result in bold.)

by (cf. Eq. (4)):

$$\mathbf{O}(z) = \sum_{k=0}^{K} \mathbf{W}^k(z) A^k_{\mathbf{S}\leftarrow\mathbf{D}} \begin{bmatrix} z \\ 1 \end{bmatrix}. \tag{8}$$

The predicted optical flow and confidence map are used as per FOMM [27]. However in contrast to FOMM [27], but similar to Monkey-Net [28], here deformable skip connections [30] are used between the encoder and decoder.

3.5. Training

The proposed model is trained end-to-end using a reconstruction loss in the feature space of the pretrained VGG-19 network [15, 37]. We use a multi-resolution reconstruction loss from previous work [27, 31]:

$$\mathcal{L}_{\text{rec}}(\hat{\mathbf{D}}, \mathbf{D}) = \sum_{l} \sum_{i} \left| \mathrm{V}_i(\mathrm{F}_l \odot \hat{\mathbf{D}}) - \mathrm{V}_i(\mathrm{F}_l \odot \mathbf{D}) \right|, \tag{9}$$

where $\hat{\mathbf{D}}$ is the generated image, V_i is the i^{th}-layer of the VGG-19 pretrained network, and F_l is a downsampling operator. Per FOMM [27], we also use an equivariance loss,

$$\mathcal{L}_{\text{eq}} = \left| A^k_{\mathbf{X}\leftarrow\mathbf{R}} - \tilde{A} A^k_{\tilde{\mathbf{X}}\leftarrow\mathbf{R}} \right|, \tag{10}$$

where $\tilde{\mathbf{X}}$ is image $\mathbf{X}$ transformed by $\tilde{A}$, and $\tilde{A}$ is some random geometric transformation. The final loss is the sum of terms, $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{eq}}$.

3.6. Animation via disentanglement

Image animation using both *standard* and *relative* methods has limitations. The *standard* method directly transfers object shape from the driving frame into the generated video, while *relative* animation is only applicable to a limited set of inputs, e.g. it requires that objects be in the same pose in the source $\mathbf{S}$ and initial driving $\mathbf{D}_1$ frames. To address this, we learn disentangled shape and pose encoders, as shown in Fig. 3(b). The pose encoder takes in the set of *driving* motions, $\{A^k_{\mathbf{D}\leftarrow\mathbf{R}}\}_{k=1}^K$, while the shape encoder takes in the set of *source* motions, $\{A^k_{\mathbf{S}\leftarrow\mathbf{R}}\}_{k=1}^K$. A decoder then uses the concatenated latent representations (each in $\mathbb{R}^{64}$) of these two encoders, to produce a set of modified driving motions,

$\{\tilde{A}^k_{\mathbf{D}\leftarrow\mathbf{R}}\}_{k=1}^K$ encoding the motion of the former and the shape of the latter. These are then used to render the output.

The encoders and the decoder are implemented using fully connected layers, and trained separately from (and after) other blocks, using an $\mathcal{L}_1$ reconstruction loss on the motion parameters. As with earlier training, source and driving frames come from the same video (i.e. the object has the same shape), therefore to ensure that shape comes from the correct branch, random horizontal and vertical scaling deformations are applied to the driving motions during training, as shown in Fig. 3(b). This forces shape information to come from the other (shape) branch. However, since the shape branch has a different pose, the pose must still come from the pose branch. Thus shape and pose are disentangled. The deformations are not applied at test time.

4. Evaluation

We now discuss the datasets, metrics and experiments used to evaluate the proposed method. Later we compare with prior work, as well as ablate our contributions.

4.1. Toy Motion Representation Experiment

To demonstrate the benefit of the proposed PCA-based motion representation, we devise an experiment on rotated rectangles (see Sup. Mat.): the task is to predict the rotation angle of a rectangle in an image. To fully isolate our contribution, we consider a supervised task, where three different architectures learn to predict angles under the $\mathcal{L}_1$ loss. The first, a Naive architecture, directly regresses the angle using an encoder-like architecture. The second is Regression-based, as in FOMM [27]. The third uses our PCA-based approach (see Sup. Mat.). Test results are presented in Fig. 5, against training set size. The Naive baseline struggles to produce meaningful results for any size of training set, while Regression-based performance improves with more data. However, the PCA-based approach significantly improves accuracy over the Regression-based one, being over an order of magnitude better with a large number of samples. This shows that it is significantly easier for the network to infer geometric parameters of the image, such as angle, using our proposed PCA-based representation.

4.2. Benchmarks

We evaluate our method on several benchmark datasets for animating human faces, bodies and animated cartoons. Each dataset has separate training and test videos. The datasets are as follows:

- *VoxCeleb* [21] consists of interview videos of different celebrities. We extract square, face regions and downscale them to 256×256, following FOMM [27]. The number of frames per video ranges from 64 to 1024.
- *TaiChiHD* [27] consists of cropped videos of full human bodies performing Tai Chi actions. We evaluate on two resolutions of the dataset: 256×256 (from FOMM [27]), and a new, 512×512 subset, removing videos lacking sufficient resolution to support that size.
- *MGif* [28] is a dataset of *.gif* files, that depicts 2D cartoon animals. The dataset was collected using google searches.
- *TED-talks* is a new dataset, collected for this paper in order to demonstrate the generalization properties of our model. We cropped the upper part of the human body from the videos, downscaling to 384×384. The number of frames per video ranges from 64 to 1024.

Since video animation is a relatively new problem, there are not currently many effective ways of evaluating it. For quantitative metrics, prior works [39, 28, 27] use video reconstruction accuracy as a proxy for image animation quality. We adopt the same metrics here:

- $\mathcal{L}_1$ error is the mean absolute difference between reconstructed and ground-truth video pixel values.
- *Average keypoint distance* (AKD) and *missing keypoint rate* (MKR) evaluate the difference between poses of reconstructed and ground truth video. Landmarks are extracted from both videos using public, body [5] (for TaiChiHD and TED-talks) and face [3] (for VoxCeleb) detectors. AKD is then the average distance between corresponding landmarks, while MKR is the proportion of landmarks present in the ground-truth that are missing in the reconstructed video.
- *Average Euclidean distance* (AED) evaluates how well identity is preserved in reconstructed video. Public re-identification networks for bodies [13] (for TaiChiHD and TED-talks) and faces [1] extract identity from reconstructed and ground truth frame pairs, then we compute the mean $\mathcal{L}_2$ norm of their difference across all pairs.

4.3. Comparison with the state of the art

We compare our method with the current state of the art for unsupervised animation, FOMM [27], on both reconstruction (the training task) and animation (the test-time task). We used an extended training schedule compared to [27], with 50% more iterations. To compare fairly with FOMM [27], we also re-trained it with the same training schedule. Also, for reference, we include comparisons with X2Face [39] and Monkey-Net [28] on video reconstruction.

Reconstruction quality Quantitative reconstruction results are reported in Table 2. We first show that our method reaches state-of-the-art results on a dataset with non-articulated objects such as faces. Indeed, when compared with FOMM [27] on VoxCeleb, our method shows on-par results. The situation changes, however, when articulated objects are considered, such as human bodies in

Figure 4: **Qualitative comparisons.** We show representative examples of articulated animation using our method and FOMM [27], on two datasets of articulated objects: TED-talks (left) and TaiChiHD (right). Zoom in for greater detail.

	TaiChiHD (256)			TaiChiHD (512)			TED-talks			VoxCeleb			MGif
	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	AKD	AED	$\mathcal{L}_1$
X2Face	0.080	(17.65, 0.109)	0.27	-	-	-	-	-	-	0.078	7.69	0.405	-
Monkey-Net	0.077	(10.80, 0.059)	0.228	-	-	-	-	-	-	0.049	1.89	0.199	-
FOMM	0.056	(6.53, 0.033)	0.172	0.075	(17.12, 0.066)	0.203	0.033	(7.07, 0.014)	0.163	0.041	**1.27**	0.134	0.0223
Ours	**0.047**	(**5.58**, **0.027**)	**0.152**	**0.064**	(**13.86**, **0.043**)	**0.172**	**0.026**	(**3.75**, **0.007**)	**0.114**	**0.040**	1.28	**0.133**	**0.0206**

Table 2: Video reconstruction: comparison with the state of the art on five different datasets. For all methods we use $K = 10$ regions. (Best result in bold.)

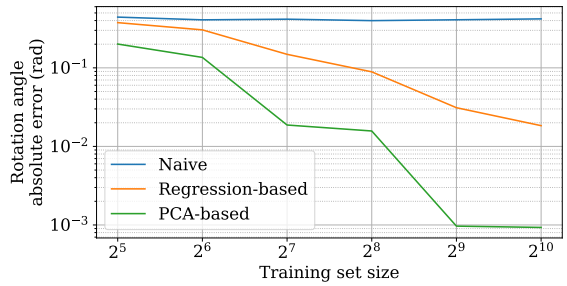


Figure 5: Mean test-time absolute rotation error, as a function of training set size.

TaiChiHD and TED-talks datasets, on which our improved motion representations boost all the metrics. The advantage over the state of the art holds at different resolutions, for TaiChiHD (256), TaiChiHD (512) and TED-talks, as well as for different numbers of selected regions (discussed later).

Animation quality Fig. 3(a,c) & 4 show selected and representative animations respectively, using our method and FOMM [27], on articulated bodies. Here for FOMM [27] we use the *standard* method, while ours uses animation via disentanglement. The results show clear improvements, in most cases, in animation quality, especially of limbs.

Animation quality was evaluated quantitatively through a user preference study similar to that of [27]. AMT users were presented with the source image, driving video, and the output from our method and FOMM [27], and asked which of the two videos they preferred. 50 such videos were evaluated, by 50 users each, for a total of 2500 preferences per study. The results further support the reconstruction scores in Tab. 2. When the animated object is not articulated (VoxCeleb), the method delivers results comparable to the previous work, e.g. 52% preference in favour of our method. However, when bodies are animated (TaiChiHD & TED-talks), FOMM [27] fails to correctly detect and animate the articulated body parts such as hands. Our method renders them in the driving pose even for extreme cases, leading to a high preference in favor of it (see Tab. 3, middle column). Moreover since it is not possible to demonstrate the benefits of the animation via disentanglement using reconstruction metrics, we run an additional user study to compare our method with *standard* animation and animation via disentanglement. Since animation via disentanglement preserves shape of the object much better (see Fig. 3(c)), users prefer it more often. It especially pronoun in case of the TED-talks dataset, since shape of the objects differs more in that case (see Tab. 3, last column).

Dataset	Our vs FOMM (%)	Our vs Standard (%)
TaiChiHD (256)	83.7%	53.6%
TED-talks	96.6%	65.6%

Table 3: User study: second column - the proportion (%) of users that prefer our method over FOMM [27]; third column - the proportion (%) of users that prefer animation via disentanglement over *standard* animation for our model.

	$\mathcal{L}_1$	(AKD, MKR)	AED
No pca or bg model	0.060	(6.14, 0.033)	0.163
No pca	0.056	(9.58, 0.034)	0.206
No bg model	0.059	(**5.55**, **0.026**)	0.165
Full method	**0.048**	(5.59, 0.027)	**0.152**

Table 4: Ablation study on TaiChiHD (256) dataset with $K = 10$. (Best result in bold.)

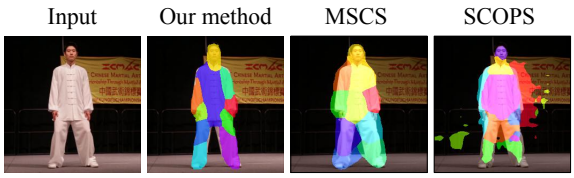


Figure 6: Qualitative co-part segmentation comparison with recent methods.

Finally, we applied animation from a TED-talks video to a photograph of George Washington, shown in Fig. 1, demonstrating animation of out-of-domain data.

4.4. Ablations

In order to understand how much benefit each of our contributions bring, we ran a number of ablation experiments, detailed in Tab. 4.

PCA-based vs. regression-based representations First we compare the PCA-based motion model with the previous, regression-based one [27]. From the qualitative heatmap depictions in Fig. 3(a), we observe that the regression-based method localizes one edge of each corresponding part, while our method predicts regions that roughly correspond to the segmentation of the object into its constituent, articulated parts. This meaningful segmentation arises completely unsupervised.

From Tab. 1 we note that adding the PCA-based representation alone (second row) had marginal impact on the $\mathcal{L}_1$ score (dominated by the much larger background region), but it had a much larger impact on other metrics, which are more sensitive to object-part-related errors on articulated objects. This is corroborated by Tab. 4. Moreover, we observed that when our PCA-based formulation is not used, the network encodes some of the movement into the background branch, leading to significant degradation of keypoint quality, which in turn leads to degradation of the AKD and AED scores (No-pca, Tab. 4).

We intuit that PCA-based estimation both captures regions and improves performance because it is much easier for the convolutional network to assign pixels of an object part to the corresponding heatmap than to directly regress motion parameters to an abstract reference frame. This is borne out by our toy experiment (sec. 4.1). In order to estimate the heatmap, it need only learn all appearances of the corresponding object part, while regression-based networks must learn the joint space of all appearances of a part in all possible geometric configurations (e.g. rotated, scaled etc.).

One of the most important hyper-parameters of our model is the number of regions, K. The qualitative and quantitative ablations of this parameter are shown in Fig. 3(a) and Tab. 1 respectively. We can observe that, while the regression-based representation fails when the number of keypoints grows to 20, our PCA-based representation scales well with the number of regions.

Modeling background motion Background motion modeling significantly lowers $\mathcal{L}_1$ error (see Tab. 4, Full method vs. No bg Model). Since background constitutes a large portion of the image, and $\mathcal{L}_1$ treats all pixels equally, this is to be expected. AED was also impacted, suggesting that the identity representation captures some background appearance. Indeed, we observe (Fig. 3(a), *second row*) that having no background model causes a reduction in region segmentation quality. However, since AKD & MKR metrics evaluate object pose only, they are not improved by background modelling.

4.5. Co-part Segmentation

While designed for articulated animation, our method produces meaningful object parts. To evaluate this capability of our method, we compare it against two recent unsupervised co-part segmentation works: MSCS [29] and SCOPS [14]. Following MSCS, we compute *foreground segmentation IoU* scores on TaiChiHD. Despite not being optimized for this task, our method achieves superior performance reaching 0.81 *IoU* vs. 0.77 for MSCS [29] and 0.55 for SCOPS [14]. See Fig. 6 for qualitative results.

5. Conclusion

We have argued that previous unsupervised animation frameworks' poor results on articulated objects are due to their representations. We propose a new, PCA-based, region motion representation, that we show both makes it easier for the network to learn region motion, and encourages it to learn semantically meaningful object parts. In addition, we propose a background motion estimation module to decouple foreground and background motion. Qualitative and quantitative results across a range of datasets and tasks demonstrate several key benefits: improved region distribution and stability, improved reconstruction accuracy and user perceived quality, and an ability to scale to more regions. We also introduce a new, more challenging dataset, TED-talks, for benchmarking future improvements on this task.

While we show some results on out of domain data (Fig. 1), generalization remains a significant challenge to making this method broadly practical in articulated animation of inanimate objects.

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition. 2016. 6

[2] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[3] 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). In *ICCV*, 2017. 6

[4] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics*, 2014. 2

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 6

[6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 2

[7] A Clark, J Donahue, and K Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019. 2

[8] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, 1997. 2

[9] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[10] Oran Gafni, Lior Wolf, and Yaniv Taigman. Vid2game: Controllable characters extracted from real-world videos. *arXiv preprint arXiv:1904.08379*, 2019. 1

[11] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3d guided fine-grained face manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[12] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[13] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017. 6

[14] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, 2019. 8

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, 2016. 5

[16] 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*, 2018. 2

[17] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In *Proceedings of the Neural Information Processing Systems Conference*, 2019. 1

[18] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[19] 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*, 2019. 1

[20] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. pagan: real-time avatars using dynamic textures. *ACM Transactions on Graphics*, 2018. 2

[21] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 6

[22] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[23] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[24] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a face: Towards arbitrary high fidelity face manipulation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[25] Jian Ren, Menglei Chai, Sergey Tulyakov, Chen Fang, Xiaohui Shen, and Jianchao Yang. Human motion transfer from poses in the wild. *arXiv preprint arXiv:2004.03142*, 2020. 1, 2

[26] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2

[27] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Proceedings of the Neural Information Processing Systems Conference*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[28] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5, 6

[29] Aliaksandr Siarohin, Subhankar Roy, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Motion-supervised co-part segmentation. In *Proceedings of the International Conference on Pattern Recognition*, 2020. 8

[30] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 5

[31] Hao Tang, Dan Xu, Wei Wang, Yan Yan, and Nicu Sebe. Dual generator generative adversarial networks for multi-domain image-to-image translation. In *ACCV*, 2018. 5

[32] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[33] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[34] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003. 4

[35] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Proceedings of the Neural Information Processing Systems Conference*, 2019. 2

[36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proceedings of the Neural Information Processing Systems Conference*, 2018. 1, 2

[37] 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*, 2017. 5

[38] Wei Wang, Xavier Alameda-Pineda, Dan Xu, Pascal Fua, Elisa Ricci, and Nicu Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[39] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision*, 2018. 1, 2, 6

[40] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2