Motion Representations for Articulated Animation: Supplementary Material

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.

In this supplementary material we report additional details of the toy experiment in Sec. 1. In Sec. 2 we provide additional details for the co-part segmentation experiment. We provide additional implementation details in Sec. 3. Additionally in Sec. 4 we visually demonstrate the ability of the model to control the background. Finally in Sec. 5 we describe the TED-talks data collection procedure.

1. Toy Experiment Details

The rotated rectangles dataset consists of images of rectangles randomly rotated from $0°$ to $90°$, along with labels that indicate the angle of rotation. The rectangles have different, random colors. Visual samples are shown in Fig. 1.

We tested three different networks: Naive, Regression-based and PCA-based. The Naive network directly predicts an angle from an image using an encoder and a fully-connected layer. Regression-based is similar to FOMM [5]; the angle is regressed per pixel an using hourglass network, and pooled according to heatmap weights predicted using the same hourglass network. PCA-based is our method, described in Sec. 3.2 (in the main paper). We predict the heatmap using an hourglass network, PCA is performed according to Eq. (6) (in the main paper), and the angle is computed from matrix U as $\arctan(U_{10}/U_{00})$.

Each of the networks was trained, on subsets of the dataset of varying sizes, to minimize the $\mathcal{L}_1$ loss between predicted and ground truth rotation angle. All models were trained for 100 epochs, with batch size 8. We used the Adam optimizer, with a learning rate of 10^{-4}. We varied the size of the training set from 32 to 1024. Results, on a separate, fixed test set of size 128, were then computed, shown in Fig. 5 (in the main paper).

2. Co-part segmentation details

To perform co-part segmentation we use $\mathbf{M}^k$. A pixel z is assigned to the part that has the maximum heatmap response for that pixel, i.e. $\text{argmax}_k \mathbf{M}^k(z)$. Moreover, since our region predictor did not explicitly the predict background region, we assign pixel z to the background iff $\sum_k \mathbf{M}^k(z) < 0.001$. We demonstrate additional qualitative comparisons with MSCS [6] and SCOPS [1] in Fig. 2. It shows that our method produces more meaningful co-part segmentations compared to SCOPS [1], and separates the foreground object from the background more accurately than MSCS [6].

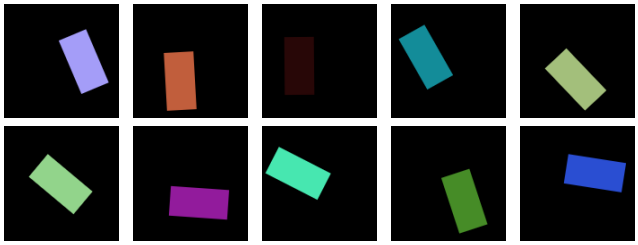

Figure 1: Examples of synthetic rectangle dataset.

Similarly to MSCS [6], we can exploit the generated segmentations by performing a part swap. In Fig. 3 we copy the cloth from the person in the source image on to the person in the driving video.

3. Implementation details

For a fair comparison, in order to highlight our contributions, we mostly follow the architecture design of FOMM [5]. Similar to FOMM, our region predictor, background motion predictor and pixel-wise flow predictor operate on a quarter of the original resolution, e.g. 64×64 for 256×256 images, 96×96 for 384×384 and 128×128 for 512×512. We use the U-Net [4] architecture with five "convolution - batch norm - ReLU - pooling" blocks in the encoder and five "upsample - convolution - batch norm - ReLU" blocks in the decoder for both the region predictor and the pixel-wise flow predictor. For the background motion predictor, we use only five block encoder part. Similarly to FOMM [5], we use the Johnson architecture [2] for image generation, with two down-sampling blocks, six residual-blocks, and two up-sampling blocks. However, we add skip connections that are warped and weighted by the confidence map. Our method is trained using Adam [3] optimizer with learning rate $2e-4$ and batch size 48, 20, 12 for 256×256, 384×384 and 512×512 resolutions respectively. During the training process, the networks observe 3M source-driving pairs, each

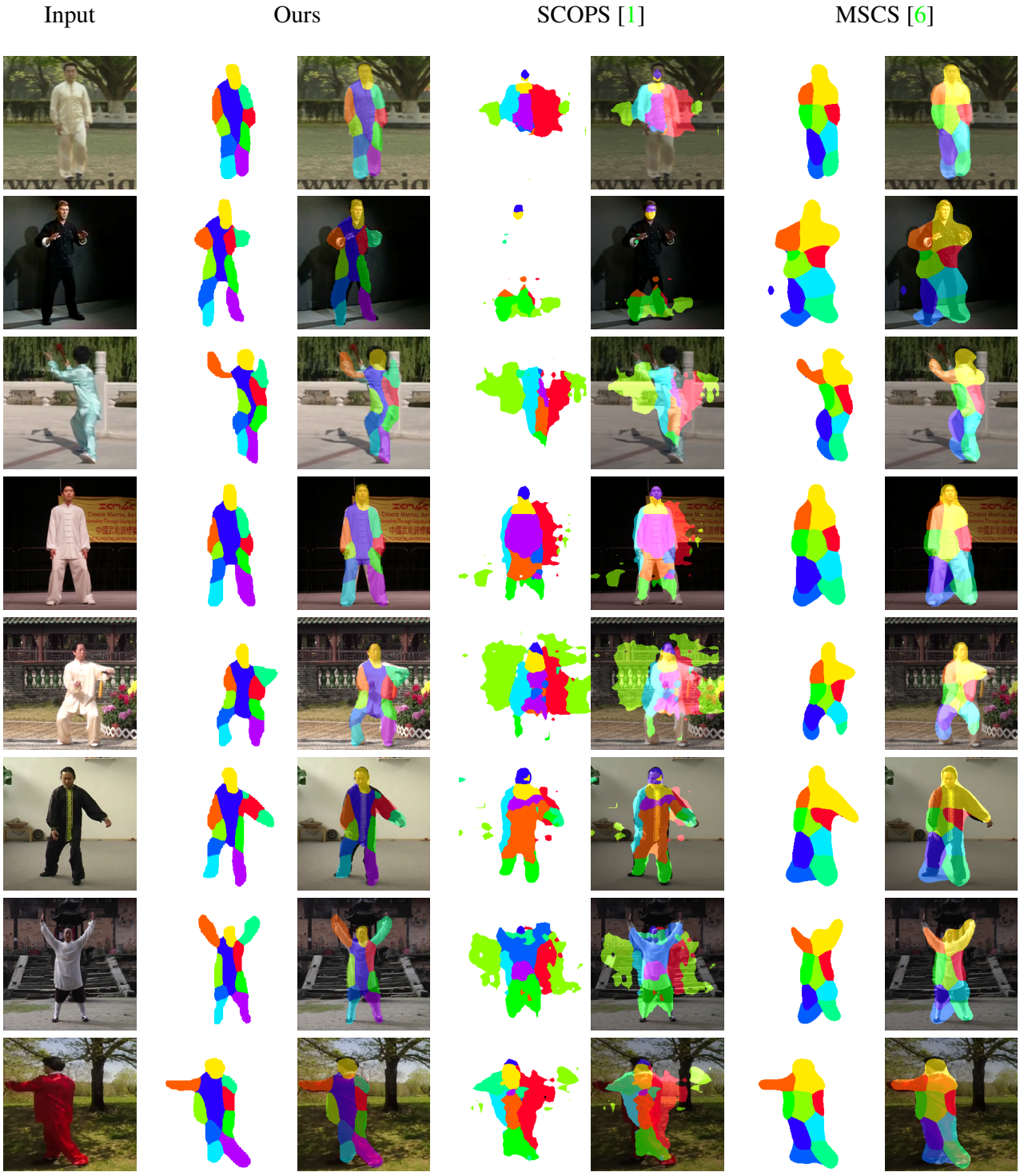

Figure 2: Additional qualitative co-part segmentation comparisons with recent methods. First column is an input. In next columns, for every method segmentation mask and image with overlayed segmentation are shown.

pair selected at random from a random video chunk, and we drop the learning rate by a factor of 10 after 1.8M and 2.7M pairs. We use 4 Nvidia P100 GPUs for training.

The shape-pose disentanglement network consists of 2 identical encoders and 1 decoder. Each encoder consists of 3 "linear - batch norm -ReLU" blocks, with the number of

Figure 3: Examples of cloth swap performed using our model. First column depicts sources from which cloth is taken, while the first row shows a driving video to which we put the cloth. Rest demonstrates images generated with our model.

hidden units equal to 256, 512, 1024, and another linear layer with the number of units equal to 64. The decoder takes a concatenated input from the encoders and applies 3 "linear - batch norm - ReLU" blocks, with sizes 1024, 512, 256. The network is trained on 1M source-driving pairs, organized in batches of 256 images. We use the Adam optimizer with learning rate $1e-3$ and drop the learning rate at 660K and

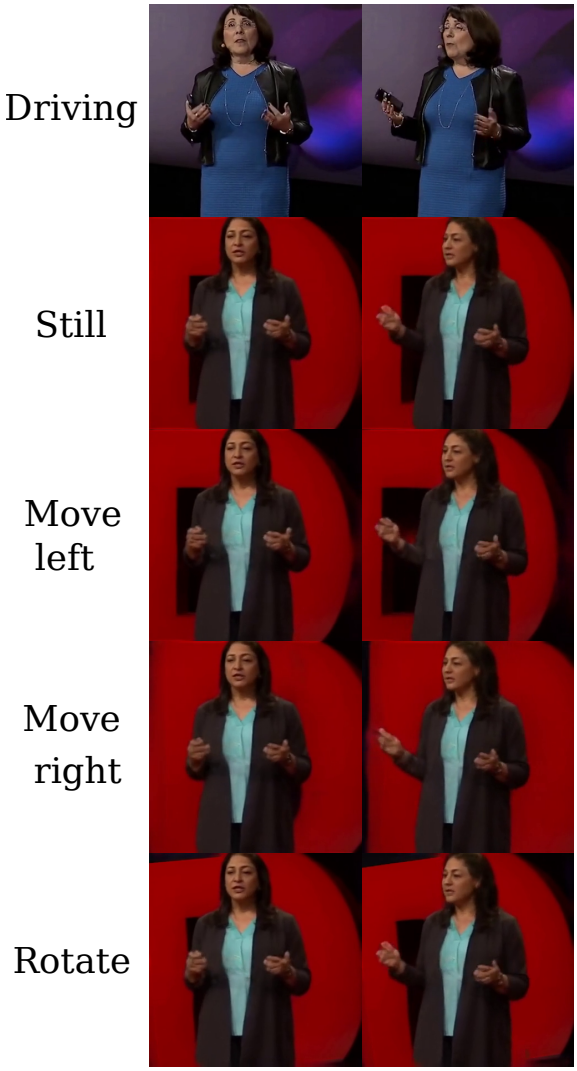

Figure 4: Visualizations of background movement. From top to bottom we show driving frame, still background, background that moves left, moves right and rotates counterclockwise.

880K pairs.

4. Background movement

The primary purpose of background modelling is to free up the capacity of the network to better model the object during training from video. For animating articulated objects from a static image at test time, background motion is usually not desired. Thus, though we estimate background motion in the driving video, we set it to zero during animation. However, nothing in our framework prevents us from controlling camera motion. Below we show a still background, then move it left, right, and rotate counterclockwise.

5. TED-talks dataset creation

In order to create the TED-talks dataset, we downloaded 3,035 YouTube videos, shared under the "CC BY – NC – ND 4.0 International" license[1], using the query "TED talks". From these initial candidates, we selected the videos in which the upper part of the person is visible for at least 64 frames, and the height of the person bounding box was at least 384 pixels. After that, we manually filtered out static videos and videos in which a person is doing something other than presenting. We ended up with 411 videos, and split these videos in 369 training and 42 testing videos. We then split each video into chunks without significant camera changes (e.g. with no cuts to another camera), and for which the presenter did not move too far from their starting position in the chunk. We cropped a square region around the presenter, such that they had a consistent scale, and downscaled this region to 384×384 pixels. Chunks that lacked sufficient resolution to be downscaled, or had a length shorter than 64 frames, were removed. Both the distance moved and the region cropping were achieved using a bounding box estimator for humans [7]. Using this process, we obtained 1,177 training video chunks and 145 test videos chunks.

References

[1] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, 2019. 1, 2

[2] 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. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 1

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[5] 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

[6] 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. 1, 2

[7] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

[1] This license allows for non-commercial use.