Image Animation / Video Re-targeting

2020.01.09

Junsoo Lee

Deep Video Generation vs Object (Image) Animation

- Deep Video Generation:
  - This task usually takes conditional information as input that comprises **categoriacal labels** or **static images** and produces video frames of desired actions.
- Object (Image) Animation / Video Re-targeting:
  - Image Animation from a driving video can be interpreted as the problem of *transferring motion information* from one domain to another.

Deep Video Generation vs Object (Image) Animation

- Deep Video Generation:
  - This task usually takes conditional information as input that comprises **categoriacal labels** or **static images** and produces video frames of desired actions.

It is a more challenging task since image animation requires **decoupling motion and content information**, as well as a recombining them.

- Object (Image) Animation / Video Re-targeting :
  - Image Animation from a driving video can be interpreted as the problem of *transferring motion information* from one domain to another.

Previous work: Image Animation

- Displaced dynamic expression regression for real-time facial tracking and animation, TOG’14
- Face2face: Real-time face capture and reenactment of rgb videos, CVPR’16
- Recycle-GAN: Unsupervised video retargeting, ECCV’18
- Everybody dance now, ECCV’18
- X2face: A network for controlling face generation using image, audio, and pose codes, ECCV’18
- Animating Arbitary Object via Deep Motion Transfer, CVPR’19
- First Order Motion Model for Image Animation, Neurips’19

Previous work: Image Animation

- Displaced dynamic expression regression for real-time facial tracking and animation, TOG'14
- Face2face: Real-time face capture and reenactment of rgb videos, CVPR'16

- Recycle-GAN: Unsupervised video retargeting, ECCV'18
- Everybody dance now, ECCV'18

- X2face: A network for controlling face generation using image, audio, and pose codes, ECCV'18
- Animating Arbitary Object via Deep Motion Transfer, CVPR'19
- First Order Motion Model for Image Animation, Neurips'19

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

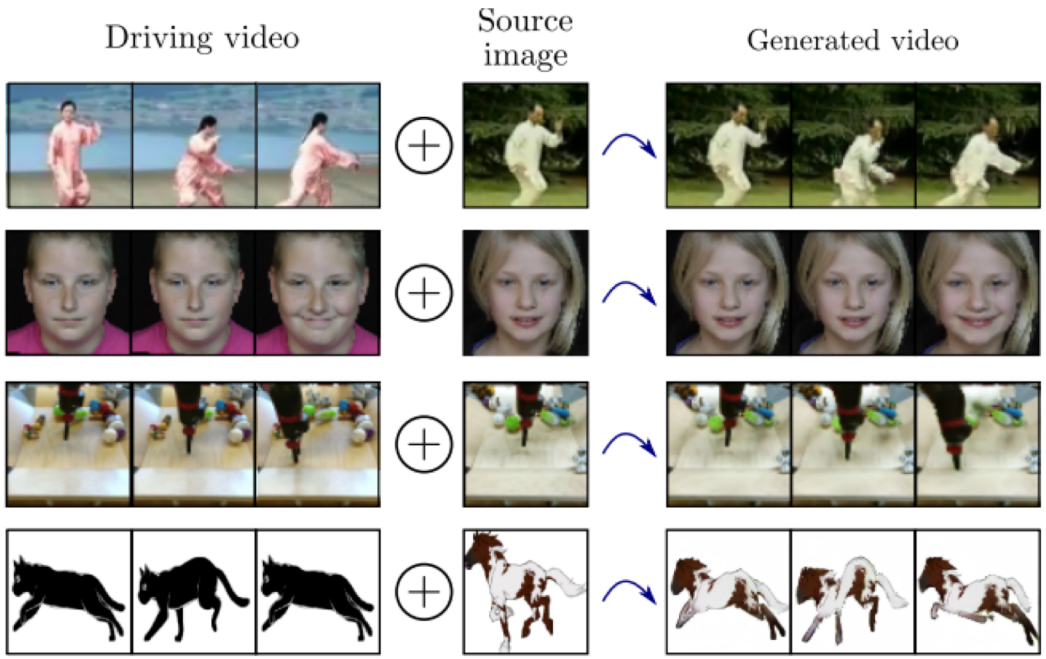


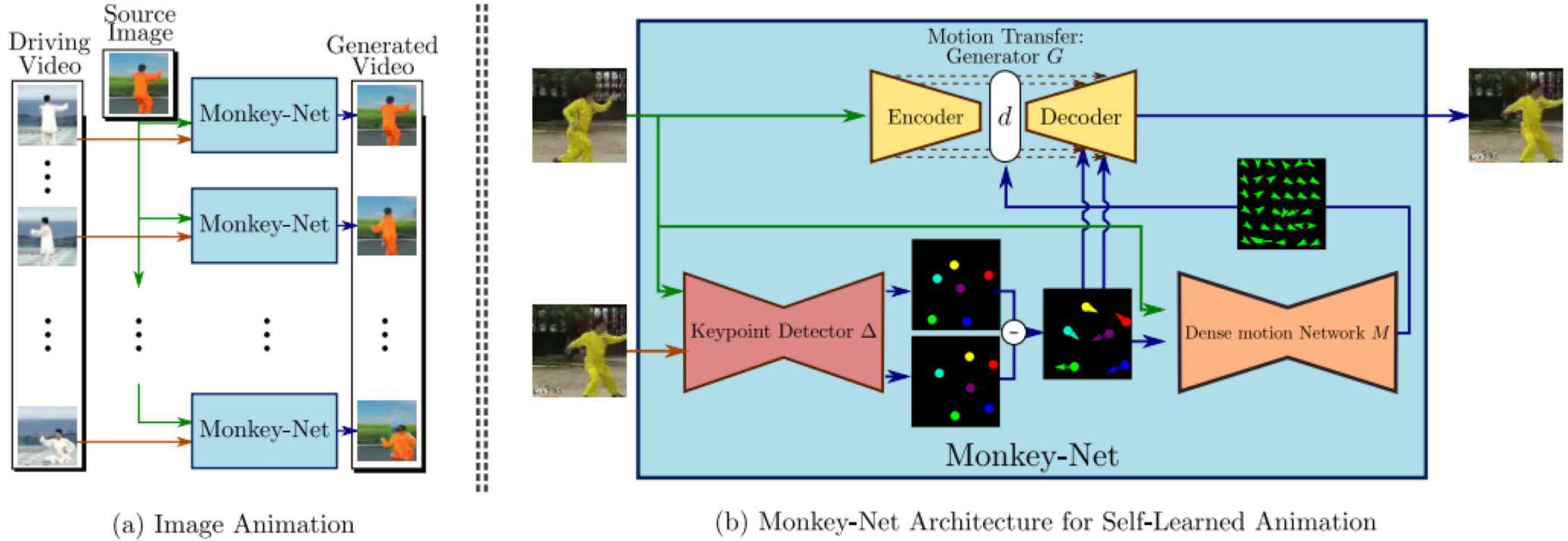
Figure 1: Our deep motion transfer approach can animate arbitrary objects following the motion of the driving video.

- The objective of this work is to animate an object based on the motion of a similar object in a driving video.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Authors propose learning a latent representation of an object category in a self-supervised way.
- This approach is not designed for specific object category, but rather is effective in animating arbitrary objects.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19



(a) Image Animation

(b) Monkey-Net Architecture for Self-Learned Animation

Figure 2: A schematic representation of the proposed motion transfer framework for image animation. At testing time (Fig. (a)), the model generates a video with the object appearance of the source image but with the motion from the driving video. Monkey-Net (Fig. (b)) is composed of three networks: a motion-specific keypoint detector Δ, a motion prediction network M and an image generator G. G reconstructs the image $\boldsymbol{x}'$ from the keypoint positions $\Delta(\boldsymbol{x})$ and $\Delta(\boldsymbol{x}')$. The optical flow computed by M is used by G to handle misalignments between $\boldsymbol{x}$ and $\boldsymbol{x}'$. The model is learned with a self-supervised learning scheme.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- 1) Keypoint Detector Δ:
- 2) Dense Motion prediction network M:
- 3) Motion Transfer network G:

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- 1) Keypoint Detector Δ:

$$\boldsymbol{h}_k = \sum_{p \in \mathcal{U}} H_k[p]p;\ \Sigma_k = \sum_{p \in \mathcal{U}} H_k[p](p - \boldsymbol{h}_k)(p - \boldsymbol{h}_k)^\top \quad (1)$$

  - Unsupervised keypoint detection inspired by
  - Estimating K heatmaps $H_k \in [0,1]^{H \times W}$, one for each keypoint
  - To model the keypoint location confidence, authors fit a Gaussian on each detection confidence map.

$$\forall p \in \mathcal{U}, H_k(\mathbf{p}) = \frac{1}{\alpha} exp\left(-(\mathbf{p} - \boldsymbol{h}_k)\Sigma_k^{-1}(\mathbf{p} - \boldsymbol{h}_k)\right) \quad (2)$$

  - The keypoint detector Δ is trained with a generator G together according to the following objective: G should be able to reconstruct x' from the keypoint location $\Delta(x)$, $\Delta(x')$, and x

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

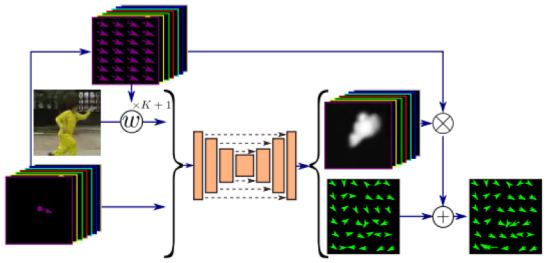


Figure 3: A schematic representation of the adopted part-based model for optical flow estimation from sparse representation. From the appearance of the first frame and the keypoints motion, the network M predicts a mask for each keypoint and the residual motion (see text for details).

$$\mathcal{F}_{\text{coarse}} = \sum_{k=1}^{K+1} M_k \otimes \rho(h_k)$$

$$\mathcal{F} = \mathcal{F}_{\text{coarse}} + \mathcal{F}_{\text{residual}}$$

- 2) Dense Motion prediction network M (from sparse keypoints to desne optical flow):
  - The task of predicting a **dense optical flow** only from the displacement of a few keypoints and the appearance of the first frame is challenging.
  - First, estimating masks $M_k \in \mathbb{R}^{H \times W}$ that segment the object in rigid parts corresponding to each keypoints
  - $\rho(\cdot) \in \mathbb{R}^{H \times W \times 2}$ is the operator that returns a tensor by repeating the input vector H x W times.
  - Since a standard U-net cannot handle large pixe2pixel misalignment between the input and the output images, a deformation module (warping function $f_w(\cdot,\cdot)$) is proposed according to $\mathcal{F}$.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Network Training:
  - Adversarial loss
    - $$\mathcal{L}^{D}_{\text{gan}}(D) = \mathbb{E}_{\boldsymbol{x}' \in \mathcal{X}}[(D(\boldsymbol{x}' \oplus H') - 1)^2] + \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2}[D(\hat{\boldsymbol{x}}' \oplus H'))^2]$$
    - $$\mathcal{L}^{G}_{\text{gan}}(G) = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2}[(D(\hat{\boldsymbol{x}}' \oplus H') - 1)^2]$$
  - Feature matching loss
    - $$\mathcal{L}_{\text{rec}} = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}')}\left[\|D_i(\hat{\boldsymbol{x}}' \oplus H') - D_i(\boldsymbol{x}' \oplus H'))\|_1\right]$$
  - Total loss
    - $$\mathcal{L}_{\text{tot}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \mathcal{L}^{G}_{\text{gan}}$$

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Generation Procedure:
  - At test time, the network receives a driving video and a source image.
  - In order to generate t^{th} frame, Δ estimates the keypoint locations h_k^s in the source image.
  - Similarly, it estimates the keypoint locations h_k^1 and h_k^t from first and the t^{th} frames of the driving video.
  - The keypoints in the generated frame are given by:

$$h_k^{s\prime} = h_k^s + (h_k^t - h_k^1)$$

  - And then, they are encoded as heatmaps using the covariance matrices.
  - Finally, the heatmaps are given to the dense motion M and the generator G together with the source image.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Results:

Figure 5: Qualitative results for image animation on the Nemo dataset: X2face (2-nd row) against our method (3-rd row).

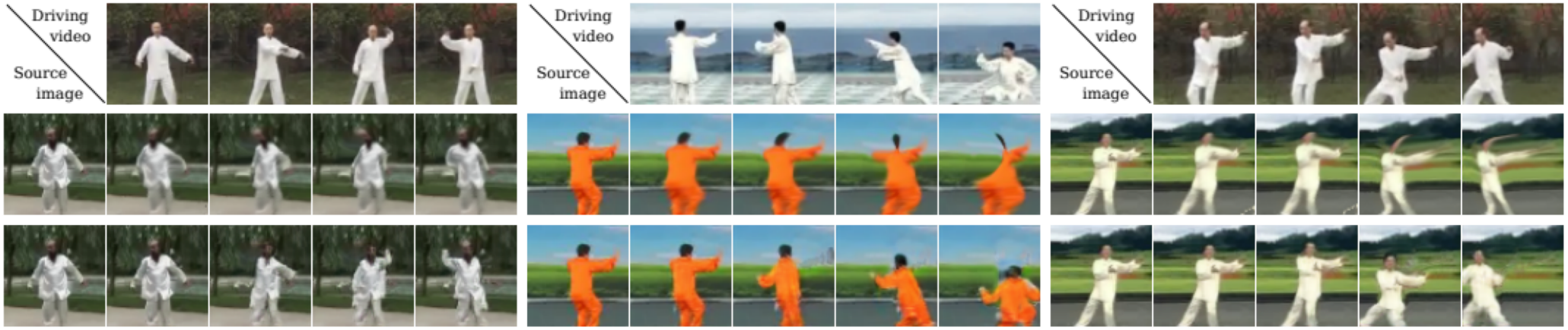


Figure 6: Qualitative results for image animation on the *Tai-Chi* dataset: X2face (2-nd row) against our method (3-rd row)

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Metric:
  - **L1** : In the case of the video reconstruction task where the ground truth video is available, we can compare the average L1 distance between frames.
  - **Average Keypoint Distance (AKD)** : In order to evaluate whether the motion of the generated video, authors employ externally pre-trained human-pose estimator for Tai-Chi dataset and the facial landmark detector for Nemo dataset
  - **Missing Keypoint Rate (MKR)** : In the case of the Tai-Chi dataset, the pre-trained human-pose estimator returns also a binary label for each keypoint indicating whether the keypoints were successfully detected. MKR is the percentage of keypoints that are detected from the ground-truth frame but not in the generated one.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Metric:
  - **Average Euclidean Distance (AED)** : AED is the feature-based metric used in [1] that consists in computing the distance between a feature representation of the ground-truth and the generated video frames. Authors employ the person re-id network for Tai-Chi and the facial identification network for Nemo.
  - **Frechet Inception Distance (FID)** : To evaluate the quality of individual frames
  - **User study**

[1] A variational u-net for conditional appearance and shape generation, CVPR'18

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Quantitative Results:

	Tai-Chi			Nemo			Bair
	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	AKD	AED	$\mathcal{L}_1$
X2Face	0.068	(4.50, 35.7%)	0.27	0.022	0.47	0.140	0.069
Ours	**0.050**	**(2.53, 17.4%)**	**0.21**	**0.017**	**0.37**	**0.072**	**0.025**

Table 1: Video reconstruction comparisons

Tai-Chi	*Nemo*	*Bair*
85.0%	79.2%	90.8%

Table 4: User study results on image animation. Proportion of times our approach is preferred over X2face [50].

Tai-Chi			
	FID	AED	MKR
MoCoGAN [43]	54.83	0.27	46.2%
Ours	**19.75**	**0.17**	**30.3%**

Nemo		
	FID	AED
MoCoGAN [43]	51.50	0.33
CMM-Net [49]	27.27	0.13
Ours	**11.97**	**0.12**

Bair	
	FID
MoCoGAN [43]	244.00
SV2P [2]	57.90
Ours	**23.20**

Table 3: Image-to-video translation comparisons.

Animating Arbitary Object via Deep Motion Transfer, CVPR'19

- Ablation Study:

	Tai-Chi		
	$\mathcal{L}_1$	(AKD, MKR)	AED
No $\mathcal{F}$	0.057	(3.11, 23.8%)	0.24
No $\mathcal{F}_{\text{residual}}$	0.051	(2.81, 18.0%)	0.22
No $\mathcal{F}_{coarse}$	0.052	(2.75, 19.7%)	0.22
No Σ_k	0.054	(2.86, 20.6%)	0.23
No $\boldsymbol{x}$	0.051	(2.71, 19.3%)	**0.21**
Full	**0.050**	**(2.53, 17.4%)**	**0.21**

Table 2: Video reconstruction ablation study *TaiChi*.

Figure 4: Qualitative ablation evaluation of video reconstruction on *Tai-Chi*.

Datasets:

- Previously used for video generation:
  - Tai-Chi dataset:
    - Clips downloaded from YouTube composed of 4500 video
    - The video length varies from 32 to 100 frames.
  - BAIR robot pushing:
    - Videos collected by a Sawyer robotic arm pushing a variety of objects over a table
    - It contains 40960 and 256 videos for train and test respectively.
  - UvA-NEMO Smile dataset:
    - a facial dynamics composed of 1240 video
    - Faces are aligned using the OpenFace library and each video has 32 frames.