MACQUARIE
University
SYDNEY·AUSTRALIA

From Image Animation to Emotion Imitation

Peizhen Li

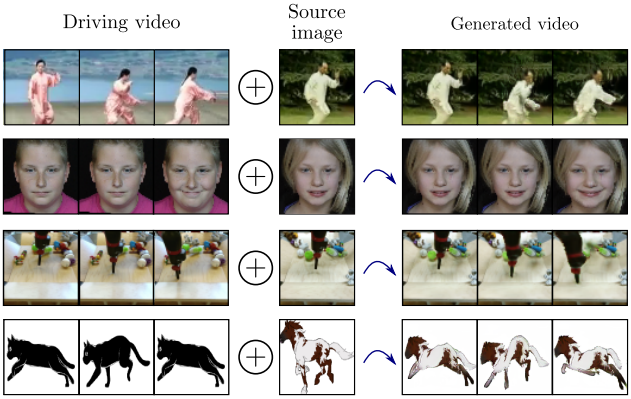
Oct 17, 2024

Outline

- Introduction
  - Image Animation
  - Video Reconstruction
  - Portrait Animation
- Motion Transfer from Video to Ameca
  - Challenges & Solutions
- Expected Results & Discussion

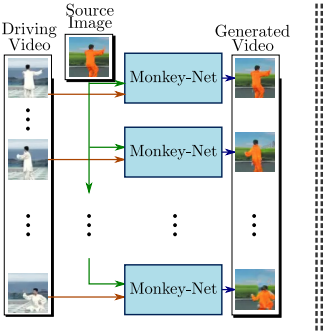
Image Animation

- Task formulation
  1. Given a source image with a target object and a driving video
  2. Animate the target object according to the motion of the driving video

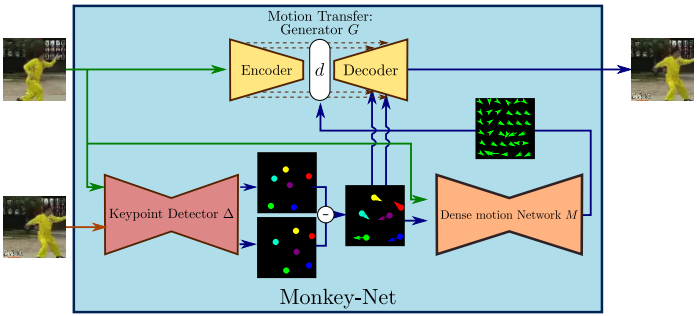


Monkey-Net

- A deep architecture that decouples appearance and motion information
- Keypoint detector Δ, dense motion network M and image generator G



(a) Image Animation



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

Unsupervised Keypoint Detection

1. Estimate K heatmaps $H_k \in [0,1]^{H \times W}$ for each input image
2. Fit a Gaussian on each detection confidence map:

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

3. Expected keypoint coordinates $\mathbf{h}_k \in \mathbb{R}^2$ and its covariance Σ_k:

$$\mathbf{h}_k = \sum_{p \in \mathcal{U}} H_k[p]p; \ \Sigma_k = \sum_{p \in \mathcal{U}} H_k[p](p - \mathbf{h}_k)(p - \mathbf{h}_k)^T. \quad (2)$$

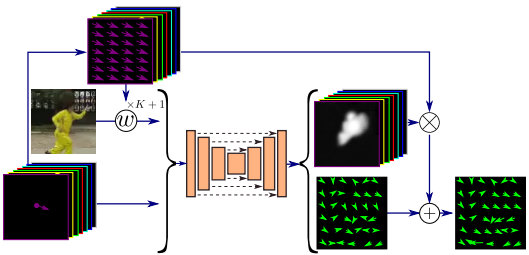
4. For $x, x^{'} \in \mathcal{X} \Rightarrow H = \{H_k\}_{k=1..K}, H^{'} = \{H_k^{'}\}_{k=1..K}, \dot{H} = H^{'} - H.$

From Sparse Keypoints to Dense Optical Flow

- Locally rigid assumption

$$\mathcal{F}_{\text{coarse}} = \sum_{k=1}^{K+1} M_k \otimes \rho(\mathbf{h}_k),\ \mathcal{F} = \mathcal{F}_{\text{coarse}} + \mathcal{F}_{\text{residual}}. \tag{3}$$

$$M_k \in \mathbb{R}^{H \times W},\ \rho(\cdot) \in \mathbb{R}^{H \times W \times 2},\ \mathcal{F} \in \mathbb{R}^{H \times W \times 2}.$$



Generator Network with Deformation

Reconstruct $x^{'}$ from $x,\ \Delta(x) = H$ and $\Delta(x^{'}) = H^{'}$.

1. Appearance feature extraction: $\xi_r \in \mathbb{R}^{H_r \times W_r \times C_r},\ 1 \leq r \leq R$
2. Feature map warping: $\xi_r^{'} = f_w(\xi_r, \mathcal{F})$
3. Feed $\xi_r^{'} \oplus \dot{H}_r$ to the decoder

```
def deform_input(self, inp, deformations_absolute):
    bs, d, h_old, w_old, _ = deformations_absolute.shape
    _, _, _, h, w = inp.shape
    deformations_absolute = deformations_absolute.permute(0, 4, 1, 2, 3)
    deformation = F.interpolate(deformations_absolute, size=(d, h, w), mode=self.interpolation_mode)
    deformation = deformation.permute(0, 2, 3, 4, 1)
    deformed_inp = F.grid_sample(inp, deformation)
    return deformed_inp
```

Bilinear interpolation

Network Training

- Adversarial loss:

$$\begin{aligned}\mathcal{L}_{\text{gan}}^{D}(D) &= \mathbb{E}_{x' \in \mathcal{X}}[(D(x^{'} \oplus H^{'}) - 1)^2] \\ &+ \mathbb{E}_{(x,x') \in \mathcal{X}^2}[D(\hat{x}^{'} \oplus H^{'})^2], \\ \mathcal{L}_{\text{gan}}^{G}(G) &= \mathbb{E}_{(x,x') \in \mathcal{X}^2}[(D(\hat{x}^{'} \oplus H^{'}) - 1)^2].\end{aligned} \tag{4}$$

- Feature matching loss:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{(x,x') \in \mathcal{X}^2}[||D_i(\hat{x}^{'} \oplus H^{'}) - D_i(x^{'} \oplus H^{'})||_1]. \tag{5}$$

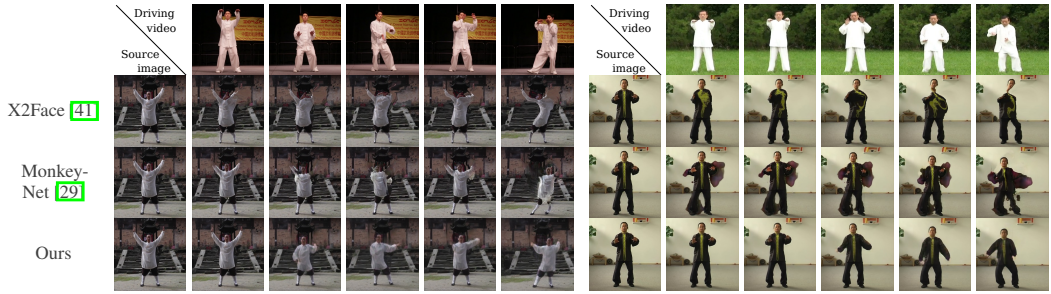
- Total training loss:

$$\mathcal{L}_{\text{tot}} = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{gan}}^{G}. \tag{6}$$

FOMM

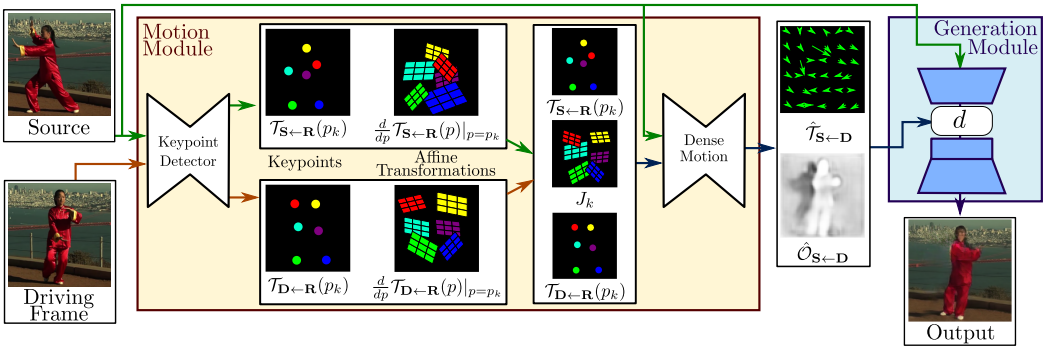
Weakness of Monkey-Net:

- Poorly models object appearance transformations in the keypoint neighborhoods
  - — Poor generation quality in the case of large object pose changes



FOMM

- Self-learned keypoints together with local affine transformations to model complex motions
- Occlusion-aware generator to indicate object parts that are not visible in the source image
- Extended equivariance loss for keypoints detector training



Local Affine Transformations for Motion Field Approximation

- Dense motion field:

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}} : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \text{ (a.k.a. backward optical flow)}, \ \mathbf{D} \in \mathbb{R}^{3\times H\times W}, \ \mathbf{S} \in \mathbb{R}^{3\times H\times W}.$$

- Approximate by the first order Taylor expansion:

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

$$\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p) \simeq \left\{\left\{\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_1), \frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\middle| p = p_1\right\}, \cdots \left\{\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_k), \frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\middle| p = p_K\right\}\right\}.$$

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}} = \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}} \circ \mathcal{T}_{\mathbf{R}\leftarrow\mathbf{D}} = \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}} \circ \mathcal{T}^{-1}_{\mathbf{D}\leftarrow\mathbf{R}}. \tag{7}$$

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}(z) \approx \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) + J_k(z - \mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)). \tag{8}$$

$$J_k = \left(\frac{d}{dp}\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p)\middle| p = p_k\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p)\middle| p = p_k\right)^{-1}. \tag{9}$$

Combining Local Motions

- Compute heatmaps:

$$\mathbf{H}_k(z) = \exp\left(\frac{(\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k) - z)^2}{\sigma}\right) - \exp\left(\frac{(\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) - z)^2}{\sigma}\right). \tag{10}$$

- Dense motion prediction:

$$\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}(z) = \mathbf{M}_0 z + \sum_{k=1}^{K} \mathbf{M}_k(\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) + J_k(z - \mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k))). \tag{11}$$

- Compare to Monkey-Net:

$$\mathcal{F}_{\text{coarse}} = \sum_{k=1}^{K} M_k \otimes \rho(\mathbf{h}_k),\ M_k \in \mathbb{R}^{H\times W},$$
$$\mathcal{F} = \mathcal{F}_{\text{coarse}} + \mathcal{F}_{\text{residual}}.$$

Occlusion-aware Image Generation and Training Losses

- Feature warping with occlusion map:

$$\xi^{'} = \hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}} \odot f_w(\xi, \hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}),\ \xi \in \mathbb{R}^{H^{'}\times W^{'}},\ \hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}} \in \mathbb{R}^{H^{'}\times W^{'}}. \tag{12}$$

- Reconstruction loss:

$$L_{rec}(\hat{\mathbf{D}}, \mathbf{D}) = \sum_{i=1}^{I} \left| N_i(\hat{\mathbf{D}}) - N_i(\mathbf{D}) \right|. \tag{13}$$

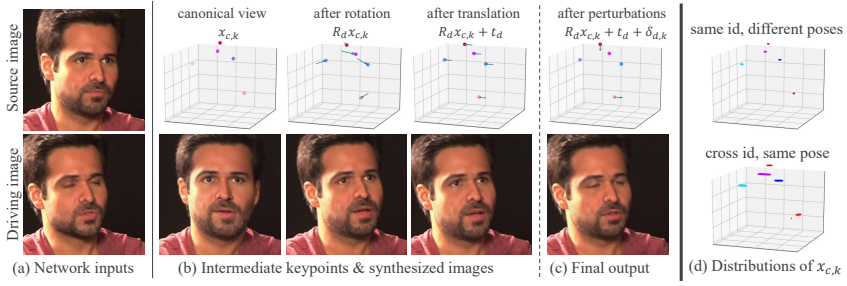
- Impose equivariance constraint:

$$\begin{aligned} \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}} &\equiv \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}} \circ \mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}, \\ \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_k) &\equiv \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}} \circ \mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p_k), \\ \left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\middle| p = p_k\right) &\equiv \left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}}(p)\middle| p = \mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p_k)\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p)\middle| p = p_k\right). \end{aligned} \tag{14}$$

Face vid2vid

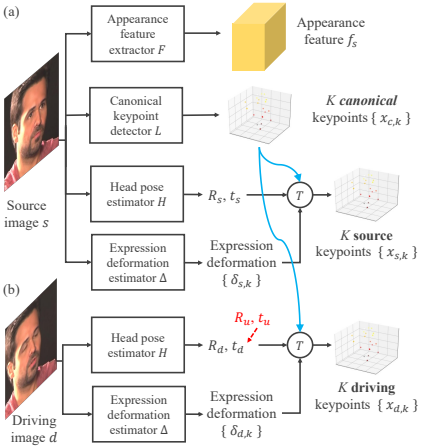
- Avoid estimating Jacobians like FOMM
- A novel keypoint representation

$$x_{s,k} = T(x_{c,k}, R_s, t_s, \delta_{s,k}) \equiv R_s x_{c,k} + t_s + \delta_{s,k},$$
$$x_{d,k} = T(x_{c,k}, R_d, t_d, \delta_{d,k}) = R_d x_{c,k} + t_d + \delta_{d,k}. \quad (15)$$



Source and Driving Feature Extraction

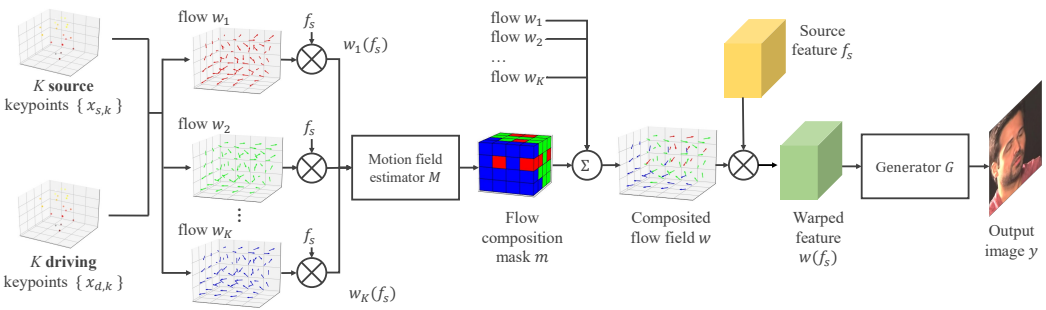
- One-shot free-view video synthesis



Video Synthesis

- Warp the source feature f_s using flow field w and feed the result to the image generator G
- Training loss:

$$\mathcal{L}_P(d, y) + \mathcal{L}_G(d, y) + \mathcal{L}_E(\{x_{d,k}\}) + \mathcal{L}_L(\{x_{d,k}\}) + \mathcal{L}_H(R_d, \bar{R}_d) + \mathcal{L}_\Delta(\{\delta_{d,k}\}). \quad (16)$$



LivePortrait

- Upgraded network architecture
- Scalable motion transformation

$$\begin{cases} x_s = s_s \cdot (x_{c,s} R_s + \delta_s) + t_s, \\ x_d = s_d \cdot (x_{c,s} R_d + \delta_d) + t_d. \end{cases} \tag{17}$$

- Landmark-guided implicit keypoints optimization

$$\mathcal{L}_{\text{guide}} = \frac{1}{2N} \sum_{i=1}^{N} (\text{Wing}(l_i, x_{s,i,:2}) + \text{Wing}(l_i, x_{d,i,:2})). \tag{18}$$

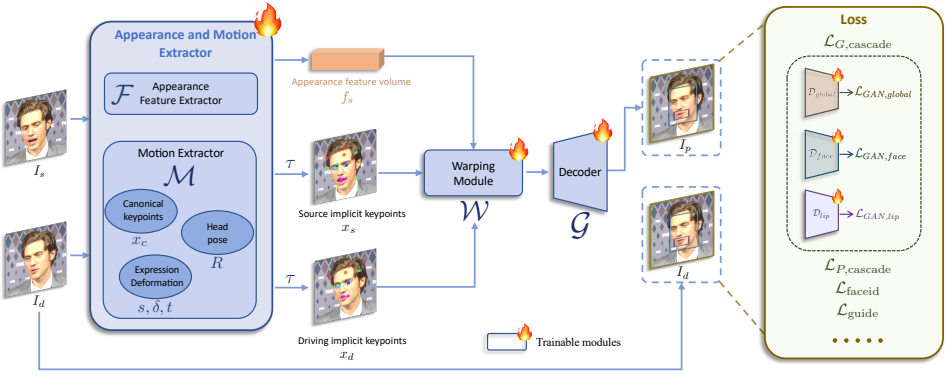
- Total training objective of the first stage

$$\mathcal{L}_{base} = \mathcal{L}_E + \mathcal{L}_L + \mathcal{L}_H + \mathcal{L}_\Delta + \mathcal{L}_{P,\text{cascade}} + \mathcal{L}_{G,\text{cascade}} + \mathcal{L}_{\text{faceid}} + \mathcal{L}_{\text{guide}}. \tag{19}$$

arXiv 2024: LivePortrait: Efficient Portrait Animation with Stitching and Retargeting Control

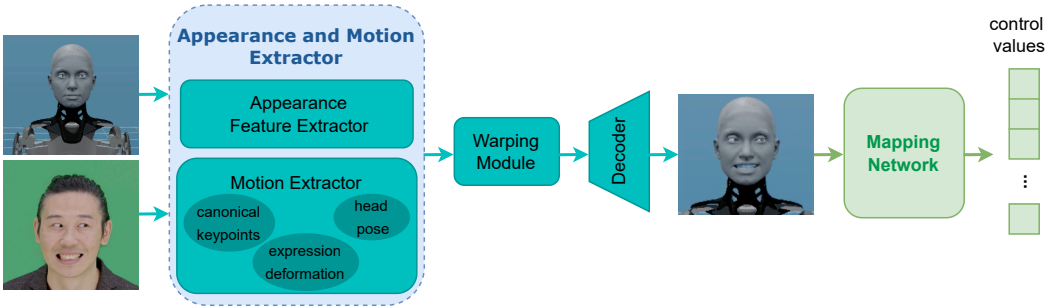
Pipeline of the Base Model Training

- The appearance and motion extractor $\mathcal{F}$ and $\mathcal{M}$, the warping module $\mathcal{W}$, and the decoder $\mathcal{G}$ are optimized



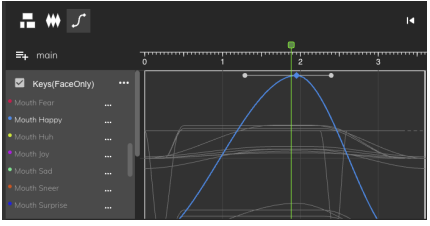
Motion Transfer from Video to Ameca

- Image space: Appearance of Ameca + motion of driving video
- Image space → action space: Generate control values for the robot given animated images



Challenges & Solutions

- Challenges
  1. How to generate action/control values from animated image?
  2. Lack of training data
- Solutions
  1. Imitation learning[1]: $\pi : \mathcal{S} \times \mathcal{Z} \rightarrow \mathcal{A}$.
  2. Construct (Image, Control Values) pairs using Animator tool



[1] BC-Z: Zero-Shot Task Generalization with Roboitc Imitation Learning

Expected Results & Discussion

- Driving video (captured by the on-robot camera) $\rightarrow$ imitated facial expression
- Alternative: control value generation from feature space
- Additional notes: interpolation and Bezier curves

Thank you very much!
Q&A

peizhen.li1@hdr.mq.edu.au