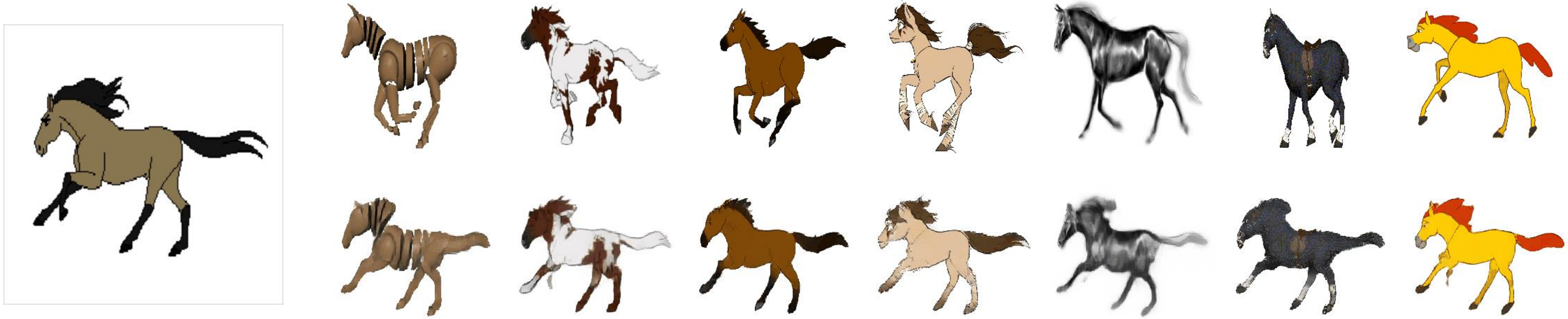
Keypoints and deformation

Paper

First Order Motion for Image Animation (FOMM)

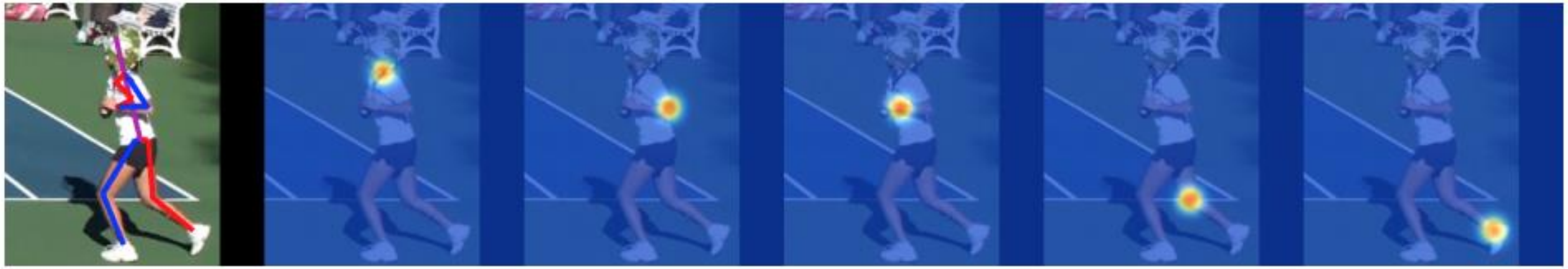
Motion-supervised Co-part Segmentation (다음에...)

What is FOMM ?

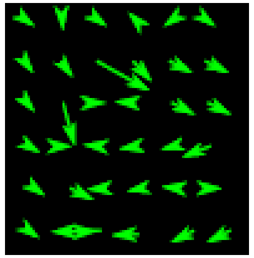


What we need to make it ?

1. Keypoints



2. Optical flow



$\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}$

* source가 target과 같은 자세가 되도록 바꾸어 주는 flow, $(R^{2\times H'\times W'})$

Compare to past work

Past work

1. 직접 key points를 지정하여 학습시킴

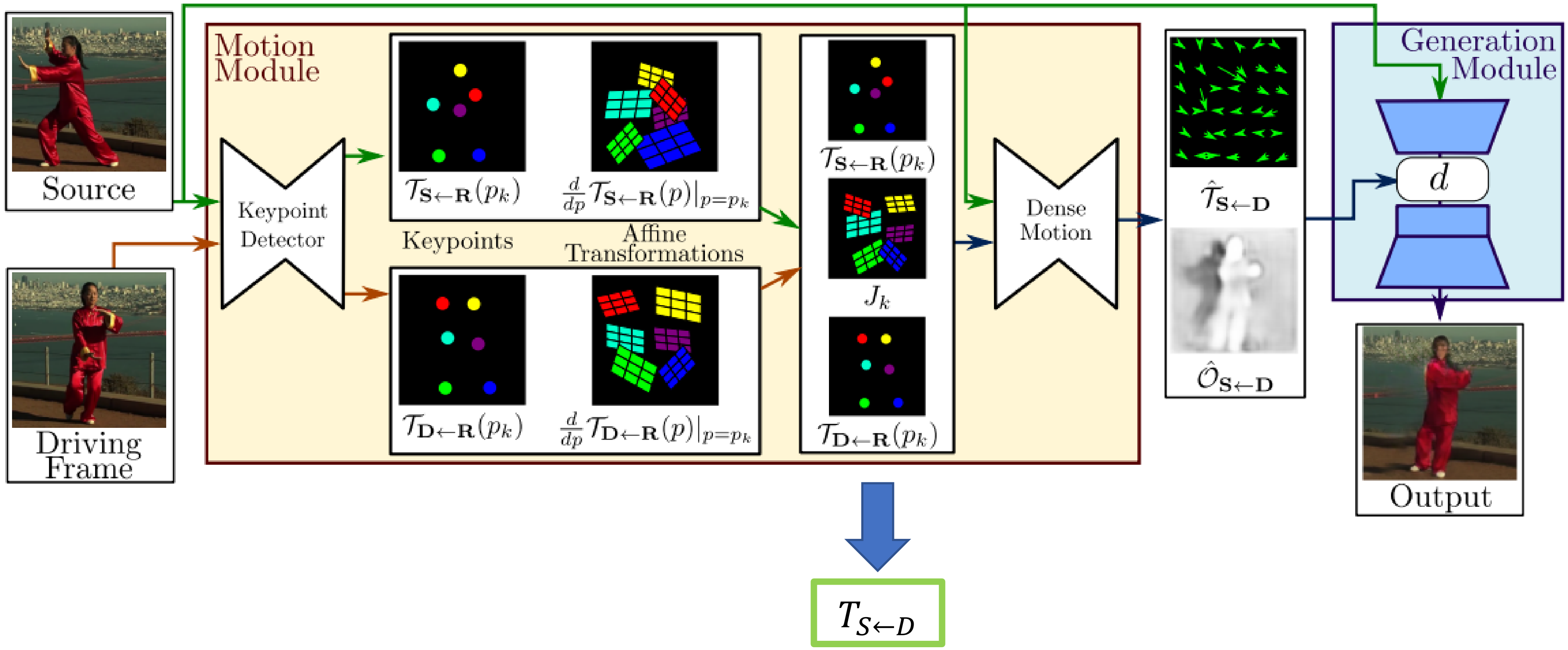
2. 단순히 target과 source의 key point 거리차를 통해 optical flow를 추정하였음.

FOMM

1. Key point의 개수만 지정하여 네트워크 스스로 적합한 key points를 찾게 한다. (Self-supervised learning, idea from Monkey-Net)

2. Taylor expansion을 사용하여 기존 optical flow 추정 방식을 더 정교하게 만들었다. (Affine Transformation)

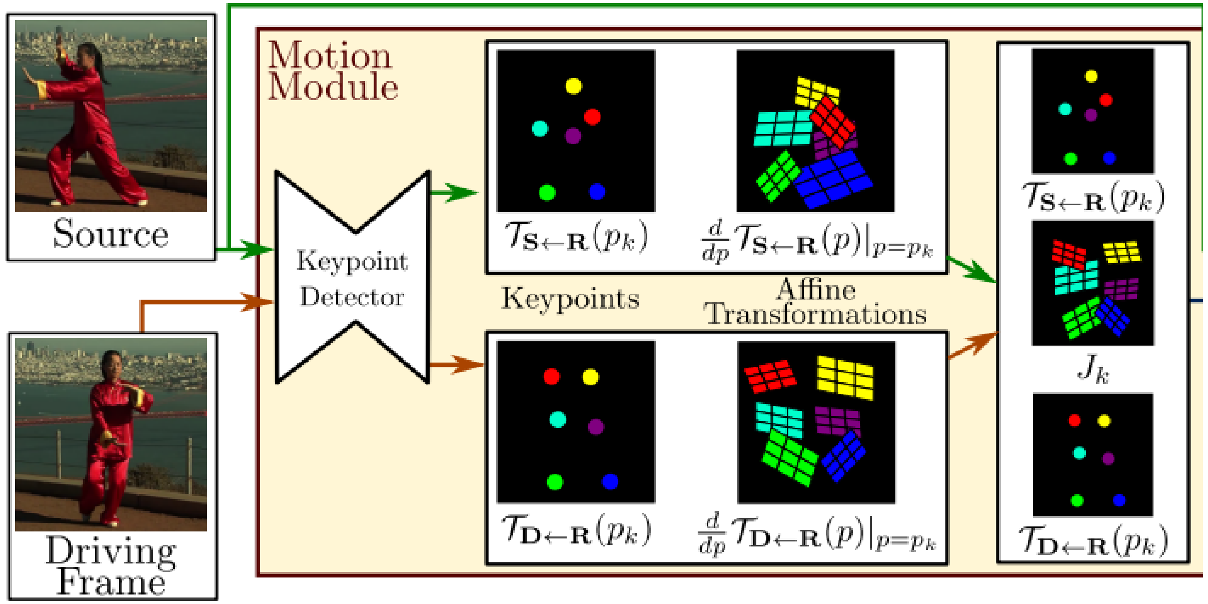
Network



Source
Driving Frame
Motion Module
Keypoint Detector
$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k)$
$\frac{d}{dp}\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p)|_{p=p_k}$
Keypoints
Affine Transformations
$\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)$
$\frac{d}{dp}\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p)|_{p=p_k}$
$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k)$
J_k
$\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)$
Dense Motion
$\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}$
$\hat{\mathcal{O}}_{\mathbf{S}\leftarrow\mathbf{D}}$
Generation Module
d
Output
$T_{S\leftarrow D}$

Network (Motion Module)

Keypoint Detector (Hourglass network)	To find $T_{S \leftarrow D}(z)$



1. R: reference frame
2. Backward optical flow 사용
3. $T_{X \leftarrow R}(p) = T_{X \leftarrow R}(p) + \left(\frac{d}{dp} T_{X \leftarrow R}(p)|_{p=p_k}\right)(p - p_k)$

* Taylor expansion을 통해 $T_{S \leftarrow D}(z)$의 근사를 찾아준다.

Network (Motion Module)

Keypoint Detector

- $T_{S\leftarrow D}(z) = T_{S\leftarrow D}(z_k) + \left(\frac{d}{dp}T_{S\leftarrow D}(z)|_{z=z_k}\right)(z - z_k)$

$$= T_{S\leftarrow R}\circ T_{R\leftarrow D}(z_k) + \left(\frac{d}{dp}T_{S\leftarrow R}\right)\left(\frac{d}{dz}T_{R\leftarrow D}\right)\left(z - T_{D\leftarrow R}(p_k)\right)$$

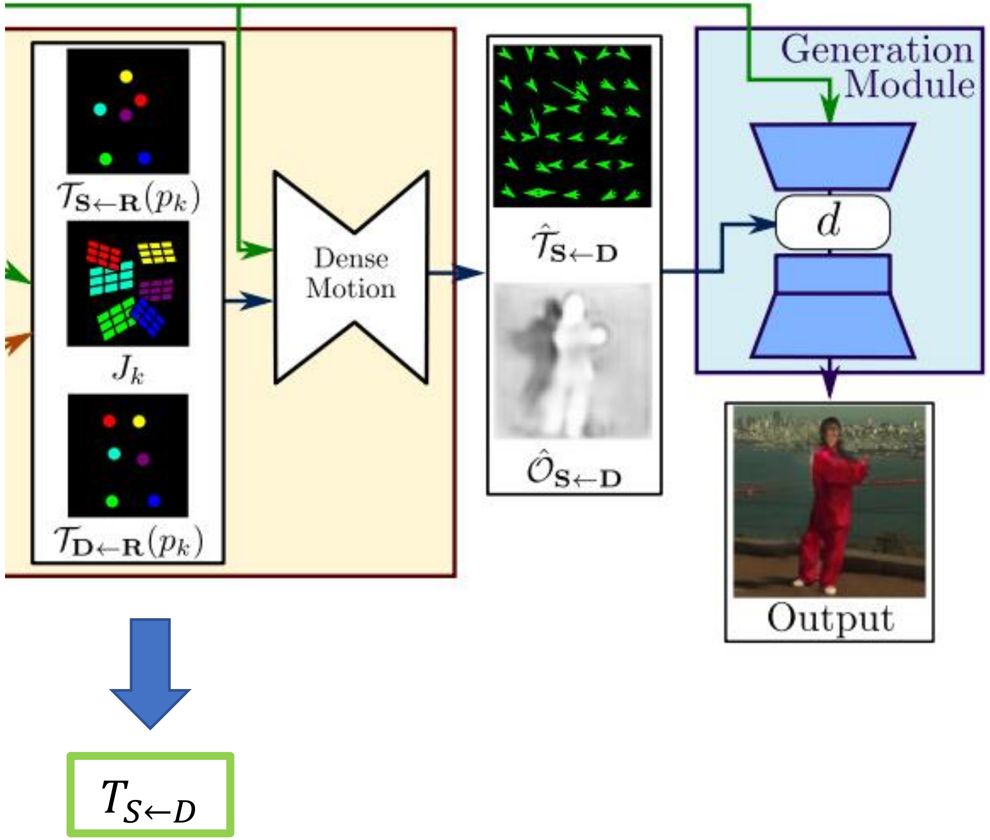
$$= T_{S\leftarrow R}(p_k) + \left(\frac{d}{dp}T_{S\leftarrow R}(p)|_{p=p_k}\right)\left(\frac{d}{dp}T_{D\leftarrow R}(p)|_{p=p_k}\right)^{-1}\left(z - T_{D\leftarrow R}(p_k)\right)$$

$$= T_{S\leftarrow R}(p_k) + J_k\left(z - T_{D\leftarrow R}(p_k)\right)$$

* Driving frame에서 source로의 keypoints로의 이동을 reference frame의 keypoints(p_k)로 표현함

Network (Motion & Generation Module)

Dense Motion (U-net) & Generation Module(encoder-decoder)



Input

1. $T_{S\leftarrow D}(z)$
2. $H_k(z) = \exp\left(\frac{(T_{D\leftarrow R}(p_k)-z)^2}{\sigma}\right) - \exp\left(\frac{(T_{S\leftarrow R}(p_k)-z)^2}{\sigma}\right)$
3. k개의 $T_{S\leftarrow D}(z)$에 의해 transform 된 이미지

Output

1. $\hat{T}_{S\leftarrow D}$: Feature map d에 대한 optical flow
2. $\hat{O}_{S\leftarrow D}$(Occlusion map): Source가 움직임에 따라 나타나게 되는 가려졌던 배경의 범위를 알려준다. (Inpainting 되어야 하는 부분)

Network (Motion & Generation Module)

Dense Motion & Generation Module

- $\hat{T}_{S\leftarrow D}(z) = M_o z + \sum_{k=1}^{K} M_k(T_{S\leftarrow D}(z))$
  $= M_o z + \sum_{k=1}^{K} M_k(T_{S\leftarrow R}(p_k) + J_k(z - T_{D\leftarrow R}(p_k))$

  *M을 mask로 각 transform에 대한 범위를 지정해준다. Dense motion 은 이를 공부한다.

- $\hat{O}_{S\leftarrow D} \in [0,1]^{H\prime \times W\prime}$는 transformed feature map을 만드는 역할을 한다. 드러나게 되는 부분을 지워 generation module에서 inpainting이 되어야 하는 부분을 알려주는 역할을 한다.

- $\xi' = \hat{O}_{S\leftarrow D} \odot f_w(\xi, \hat{T}_{S\leftarrow D})$가 generation module의 디코더에 입력될 feature map이 된다.

  * ξ': 최종적으로 디코더에 입력될 feature map
  * ξ : 인코더를 통해 나온 feature map
  * f_w: back warping function

Loss functions

1. Reconstruction Loss (based on perceptual loss)

   $L_{rec}(\widehat{D}, D) = \sum_{i=1}^{I} \left| N_i(\widehat{D}) - N_i(D) \right|$, (20 loss terms in total & VGG-19)

2. Imposing Equivariance Constraint

   (2.1) $Loss1 = ||T_{X \leftarrow R} - (T_{X \leftarrow Y} \circ T_{Y \leftarrow R})||$

   (2.2) $Loss2 = ||1 - \left(\frac{d}{dp} T_{X \leftarrow R}(p)|_{p=p_k}\right)^{-1} \left(\frac{d}{dp} T_{X \leftarrow Y}(p)|_{p=T_{X \leftarrow Y}(p_k)}\right) \left(\frac{d}{dp} T_{Y \leftarrow R}(p)|_{p=p_k}\right)||$

Loss functions (Imposing equivariance constraint)

- Loss 1

$$by\ using\quad T_{S\leftarrow R} \equiv T_{S\leftarrow D}\circ T_{D\leftarrow R}$$

$$Loss1 = \|T_{S\leftarrow R} - (T_{S\leftarrow D}\circ T_{D\leftarrow R})\|$$

- Loss 2

$$by\ using\quad T_{S\leftarrow R}(p_k) = T_{S\leftarrow R}(p) - \boxed{\left(\frac{d}{dp}T_{S\leftarrow R}(p)|_{p=p_k}\right)}(p-p_k)$$

$$= T_{S\leftarrow D}\circ T_{D\leftarrow R}(p_k) = T_{S\leftarrow R}(p) - \boxed{\left(\frac{d}{dp}T_{S\leftarrow D}(p)|_{p=T_{S\leftarrow D}(p_k)}\right)\left(\frac{d}{dp}T_{D\leftarrow R}(p)|_{p=p_k}\right)}(p-p_k)$$

$$Loss2 = ||1 - \left(\frac{d}{dp}T_{S\leftarrow R}(p)|_{p=p_k}\right)^{-1}\left(\frac{d}{dp}T_{S\leftarrow D}(p)|_{p=T_{S\leftarrow D}(p_k)}\right)\left(\frac{d}{dp}T_{D\leftarrow R}(p)|_{p=p_k}\right)||$$

Motion transfer at test stage

$$T_{S_1 \leftarrow S_t}(z) = T_{S_1 \leftarrow R}(p_k) + J_k\left(z \boxed{- T_{S_1 \leftarrow R}(p_k) + T_{D_1 \leftarrow R}(p_k)} - T_{D_t \leftarrow R}\,(p_k)\right)$$

$$p_k = T_{R \leftarrow S_t}(z_k) \quad J_k = \left(\frac{d}{dp} T_{D_1 \leftarrow R}(p)|_{p=p_k}\right)\left(\frac{d}{dp} T_{D_t \leftarrow R}(p)|_{p=p_k}\right)^{-1}$$

* D_1과 S_1이 비슷하거나 같은 pose여서 key points의 위치가 유사하다고 생각한다.

A.3 Transferring Relative Motion

In order to transfer only relative motion patterns, we propose to estimate $\mathcal{T}_{\mathbf{S}_t \leftarrow \mathbf{R}}(p)$ near the keypoint p_k by shifting the motion in the driving video to the location of keypoint p_k in the source. To this aim, we introduce $\mathcal{V}_{\mathbf{S}_1 \leftarrow \mathbf{D}_1}(p_k) = \mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{R}}(p_k) - \mathcal{T}_{\mathbf{D}_1 \leftarrow \mathbf{R}}(p_k) \in \mathbb{R}^2$ that is the 2D vector from the landmark position p_k in $\mathbf{D}_1$ to its position in $\mathbf{S}_1$. We proceed as follows. First, we shift point coordinates according to $-\mathcal{V}_{\mathbf{S}_1 \leftarrow \mathbf{D}_1}(p_k)$ in order to obtain coordinates in $\mathbf{D}_1$. Second, we apply the transformation $\mathcal{T}_{\mathbf{D}_t \leftarrow \mathbf{D}_1}$. Finally, we translate the points back in the original coordinate space using $\mathcal{V}_{\mathbf{S}_1 \leftarrow \mathbf{D}_1}(p_k)$. Formally, it can be written:

$$\mathcal{T}_{\mathbf{S}_t \leftarrow \mathbf{R}}(p) = \mathcal{T}_{\mathbf{D}_t \leftarrow \mathbf{D}_1}\left(\mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{R}}(p) - \mathcal{V}_{\mathbf{S}_1 \leftarrow \mathbf{D}_1}(p_k)\right) + \mathcal{V}_{\mathbf{S}_1 \leftarrow \mathbf{D}_1}(p_k)$$

Now, we can compute the value and Jacobian in the p_k:

$$\mathcal{T}_{\mathbf{S}_t \leftarrow \mathbf{R}}(p_k) = \mathcal{T}_{\mathbf{D}_t \leftarrow \mathbf{D}_1} \circ \mathcal{T}_{\mathbf{D}_1 \leftarrow \mathbf{R}}(p_k) - \mathcal{T}_{\mathbf{D}_1 \leftarrow \mathbf{R}}(p_k) + \mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{R}}(p_k)$$

and:

$$\left(\frac{d}{dp}\mathcal{T}_{\mathbf{S}_t \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right) = \left(\frac{d}{dp}\mathcal{T}_{\mathbf{D}_t \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{D}_1 \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right)^{-1}\left(\frac{d}{dp}\mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right).$$

Now using Eq. (6) and treating $\mathbf{S}_1$ as source and $\mathbf{S}_t$ as driving frame, we obtain:

$$\mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{S}_t}(z) \approx \mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{R}}(p_k) + J_k(z - \mathcal{T}_{\mathbf{S} \leftarrow \mathbf{R}}(p_k) + \mathcal{T}_{\mathbf{D}_1 \leftarrow \mathbf{R}}(p_k) - \mathcal{T}_{\mathbf{D}_t \leftarrow \mathbf{R}}(p_k)) \qquad (13)$$

with

$$J_k = \left(\frac{d}{dp}\mathcal{T}_{\mathbf{D}_1 \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{D}_t \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right)^{-1}. \qquad (14)$$

Note that, here, $\left(\frac{d}{dp}\mathcal{T}_{\mathbf{S}_1 \leftarrow \mathbf{R}}(p)\Big|_{p=p_k}\right)$ canceled out.

Contributions(summary)

1. Monkey-Net: zeroth order model로 optical flow를 근사하기 때문에 물체의 모습이 흐트러진다(leaking)는 약점이 있다.

   >> local affine transformation 제시

2. "Occlusion-aware generator" 소개: occlusion mask를 만들어 드러나게 되어 inpainting이 필요한 부분을 알려주는 역할을 한다.

3. Key point detector의 훈련에 사용되는 equivariance loss를 확장한다. 이는 local affine transformation의 측정을 정확히 하기 위함이다.

Limitations

1. Driving frame과 source간의 움직임이 크면 아직 leaking이 일어난다.
2. 배경과 대상의 분리가 불분명할 때가 있다.

끝:3