Neural Implicit Morphing of Face Images

Guilherme Schardong, Tiago Novello, Hallison Paz, Iurii Medvedev, Vinícius da Silva, Luiz Velho and Nuno Gonçalves

ISR-UC, IMPA, PUC-Rio, INCM

16/04/2024

Outline

Image Warping: What is it and how can we use it?

- Morphing = Warping + blending
- Warping maps the pixels of an image to a different geometry
  - **Perspective corrections**, lens distortion corrections, geometric transformations, morphing
- We “move” pixels in an image to a different location in another (or the same) image

Image Warping: What is it and how can we use it?

- Morphing = Warping + blending
- Warping maps the pixels of an image to a different geometry
  - Perspective corrections, **lens distortion corrections**, geometric transformations, morphing
- We "move" pixels in an image to a different location in another (or the same) image

Image Warping: What is it and how can we use it?

- Morphing = Warping + blending
- Warping maps the pixels of an image to a different geometry
  - Perspective corrections, lens distortion corrections, **geometric transformations**, morphing
- We "move" pixels in an image to a different location in another (or the same) image

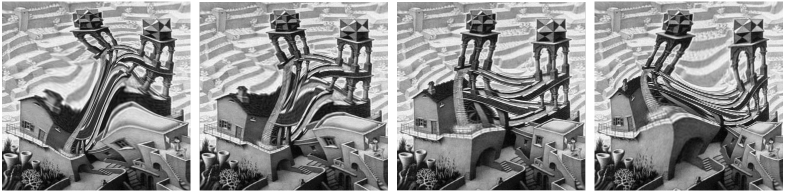


Image Warping: What is it and how can we use it?

- Morphing = Warping + blending
- Warping maps the pixels of an image to a different geometry
  - Perspective corrections, lens distortion corrections, geometric transformations, **morphing**
- We "move" pixels in an image to a different location in another (or the same) image

Image Warping: How to apply it to faces?

- For faces, warping is used to align features (eyes, nose, mouth, ears, ...)
    - Even under perfect studio conditions, faces are still “misaligned” by nature

- How can we align them?

Image Warping: How to apply it to faces?

- For faces, warping is used to align features (eyes, nose, mouth, ears, ...)
  - Even under perfect studio conditions, faces are still "misaligned" by nature

- How can we align them?

Find correspondences between every pixel in the images and warp them

Face Warping: Issues and Motivation

- Matching every single pixel does not scale well
- Plus: How to find the correspondences between every single pixel?
  - Some have no correspondences at all (signs, scars, tattoos, ...)
- Additionally, we don't really need to match all pixels, only a sufficient subset of them
  - Then we simply transform the others based on this subset
- How to find/define a subset of pixels?

Face Warping: Affine Transformation

- We can use facial landmarks as a viable subset of pixels:
  - use landmark detectors[1,2] or;
  - manually mark them
  - or both
- The matching part is trickier
- We have a correspondence between landmarks, but what about the other pixels?
  - How do we move them to their new locations?

[1] Kazemi and Sullivan, One Millisecond Face Alignment with an Ensemble of Regression Trees, CVPR 2024

[2] Lugaresi *et al.*, MediaPipe: A Framework for Building Perception Pipelines, arXiv, 2019

Face Warping: Affine Transformation

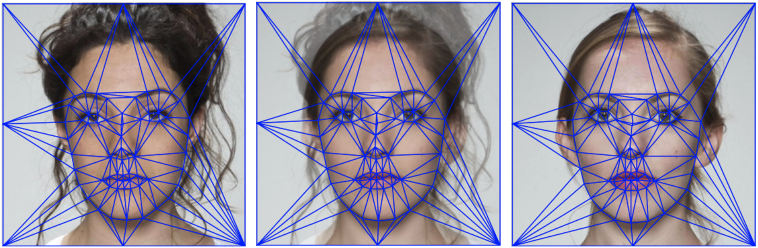
- We can use facial landmarks as a viable subset of pixels:
  - use landmark detectors[1,2] or;
  - manually mark them
  - or both
- The matching part is trickier
- We have a correspondence between landmarks, but what about the other pixels?
  - How do we move them to their new locations?
- We can think of each landmark as influencing a region around it (thoughts of Voronoi)
- When we move a landmark, any pixels in this region must be moved as well

[1] Kazemi and Sullivan, One Millisecond Face Alignment with an Ensemble of Regression Trees, CVPR 2024

[2] Lugaresi *et al.*, MediaPipe: A Framework for Building Perception Pipelines, arXiv, 2019

Face Warping: Affine Transformation

- It make sense to think that neighboring landmarks will influence each other
- Thus, we can triangulate landmarks to define the regions
- As the landmarks move, so will the triangles (and any pixels in them)



Face Warping: Affine Transformation

- It make sense to think that neighboring landmarks will influence each other
- Thus, we can triangulate landmarks to define the regions
- As the landmarks move, so will the triangles (and any pixels in them)

Good and Bad of warping by affine transformation

- Simple to implement
  - Delaunay triangulation is done once and shared for all faces
  - Landmark detection is done through 3rd party libraries
  - Image blending is done by linear interpolation as well
- Affine transforms may not be enough to align faces at all intermediate times
  - May lead to weird face proportions
  - Ghosting artifacts due to pose differences
  - It is, by nature, discrete

Motivation

- Warping by affine transforms is not that bad, but can we improve it?
    1. More specifically: Can we improve it by performing a non-linear warping?
    2. Additionally, if the warping is **smooth**, we can exploit energy functionals proposed in the literature. Is it possible to produce a **continuous** and **smooth** warping?
- Neural networks may be used as a non-linear transformation
- More specifically: Sinusoidal neural networks

SIRENs

- Sinusoidal Representation Networks (SIRENs) are simply multilayer-perceptrons with sine activation functions
- Sine was the first non-linearity proposed to be used with neural networks[3]
  - But were plagued by convergence issues
- Recent renaissance due to more in-depth studies[4]
- Eventually, better initialization methods made them feasible[5]
- Sines are C^{∞}, thus **smooth**
  - Thus, we can calculate the (high-order) derivatives of output w.r.t the input

[3] Lapedes and Farber, Nonlinear signal processing using neural networks: Prediction and system modelling, 1987

[4] Parascandolo *et al.*, Taming the waves: sine as activation function in deep neural networks, 2016

[5] Sitzmann *et al.*, Implicit neural representations with periodic activation functions, 2020

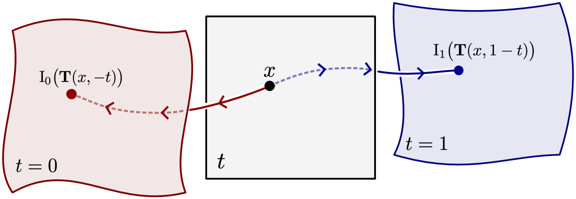
On the definition and intuition of $\mathbf{T}$

We define a transform $\mathbf{T} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$, $x \in \mathbb{R}^2$ and $t \in [-1, 1]$, such that:

- $\mathbf{T}(x, 0) = x$ (identity property)
- $\mathbf{T}(\mathbf{T}(x, t), -t) = x$ (inverse property)

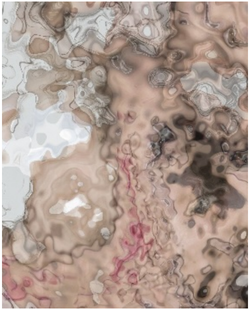
Given the landmarks p_i in I_0 and their correspondences q_i in I_1:

- $\mathbf{T}(p_i, t) = \mathbf{T}(q_i, 1 - t)$ (landmark matching property)



On the need for regularization

We want **T** to be non-rigid, but such "non-rigidity" must be under control, or this may happen:



A way to achieve this is by introducing the **Thin-plate** property for regularization:

- $\min \ \|\mathbf{Hess}\,(\mathbf{T})\,(x,t)\|_F^2$

Formal definition: Alignment Loss

$$\mathscr{L}(\theta) = \lambda_1 \mathscr{W}(\theta) + \lambda_2 \mathscr{D}(\theta) + \lambda_3 \mathscr{T}(\theta). \tag{1}$$

$\mathscr{W}(\theta)$, $\mathscr{D}(\theta)$, $\mathscr{T}(\theta)$ are the *warping*, *data*, *thin-plate* constraints

$$\mathscr{W}(\theta) = \underbrace{\int_{\mathbb{R}^2} \|\mathbf{T}(x,0) - x\|^2 \, dx}_{\text{Identity constraint}} + \underbrace{\int_{\mathbb{R}^2 \times \mathbb{R}} \left\| \mathbf{T}\Big(\mathbf{T}(x,t), -t\Big) - x \right\|^2 dxdt}_{\text{Inverse constraint}} . \tag{2}$$

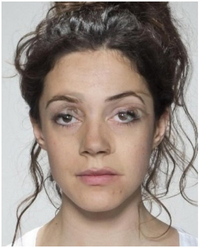
$$\mathscr{D}(\theta) = \int_{[0,1]} \|\mathbf{T}(p_i, t) - \mathbf{T}(q_i, 1-t)\|^2 \, dt \tag{3}$$

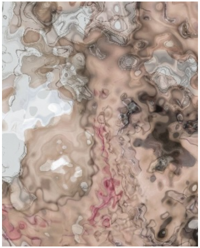
$$\mathscr{T}(\theta) = \int_{\mathbb{R}^2 \times \mathbb{R}} \|\mathbf{Hess}\,(\mathbf{T})\,(x,t)\|_F^2 \, dxdt \tag{4}$$

Effects of each loss term

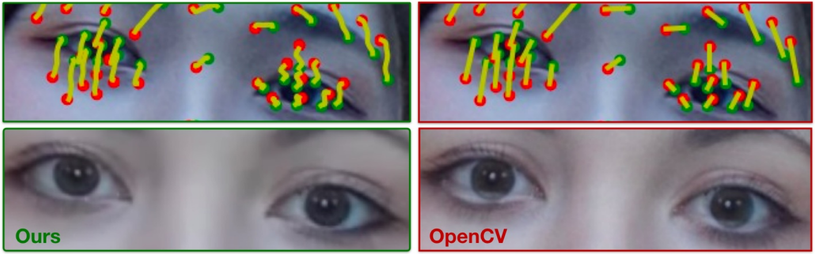

no inv


no id


no $\mathscr{D}(\theta)$

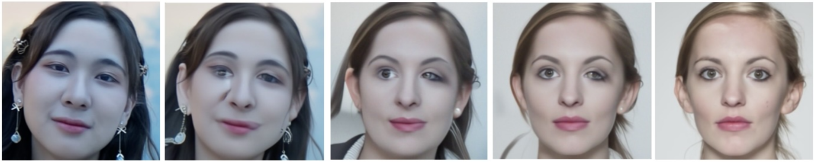

no $\mathscr{T}(\theta)$

Results: Differences in warping paths

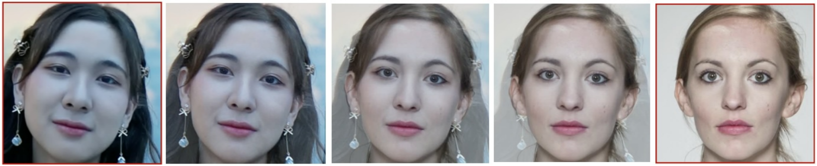


Results: Dealing with rotations

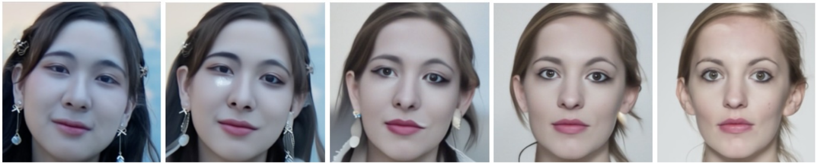
Pure diffAE



Neural warping + linear blending [Ours]

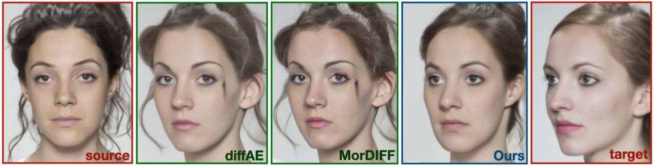


Neural warping + diffAE [Ours]



Results: Dealing with poses, expressions and occlusions

Different poses



Different expressions

source | linear+Poisson blending | generative blending | target

Eyes occlusion + faces in the wild

source | linear+ Poisson blending | generative blending | target

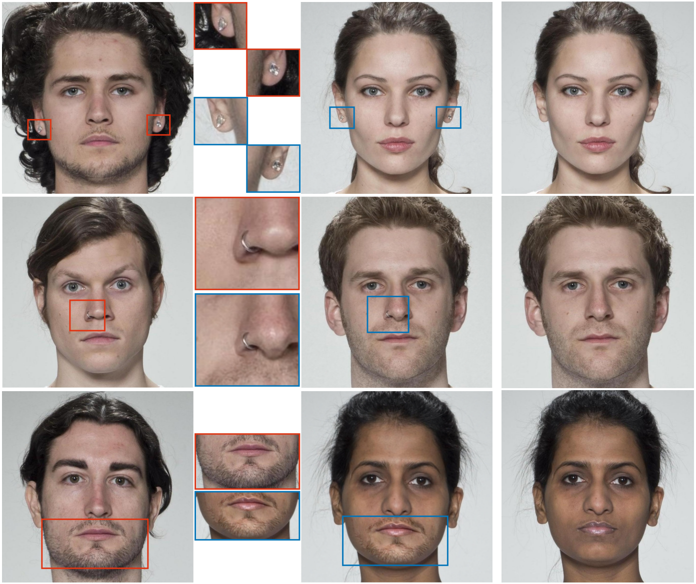
Results: Morphings between different ethnicities and genders

Results: Morphings between different ethnicities and genders

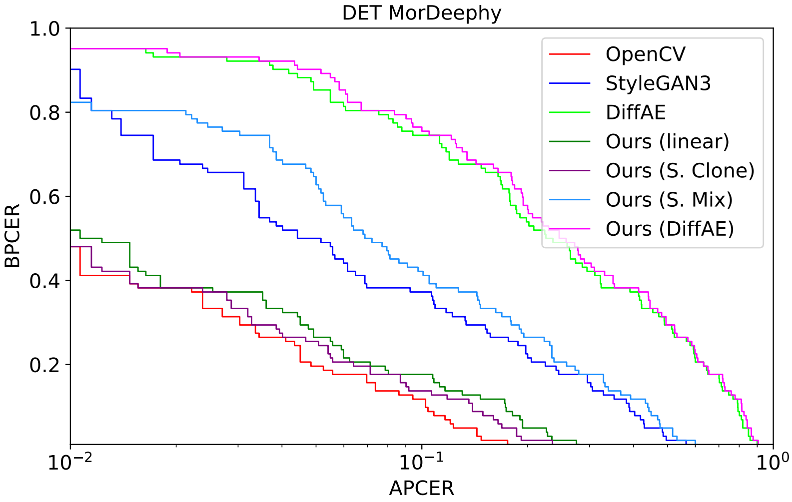
Results: Morphings between different ethnicities and genders

Results: Morphings between different ethnicities and genders

Results: Feature transference



Results: Morphing Attack Detection Effectiveness



Results: Morphing failures

- Needs more time to converge. But why?
  - Maybe the initialization is not good enough

Conclusions and Future Directions

- Introduction of classic energy functionals to use with neural networks is fairly straightforward
  - They are still very powerful and yield good results
- We can exploit the network smoothness to use its derivatives on the loss
- Separation of problems adds flexibility, since we can "mix and match" various techniques
- Investigate why some cases take so long to converge

Thank you