Neural Implicit Morphing of Faces

Guilherme Schardong[1,*] Tiago Novello[2,*] Hallison Paz[2] Iurii Medvedev[1] Vinícius da Silva[3]
Luiz Velho[2] Nuno Gonçalves[1,4]

[1] Institute of Systems and Robotics, University of Coimbra
[2] Institute of Pure and Applied Mathematics
[3] Tecgraf, Pontifical Catholic University of Rio de Janeiro
[4] Portuguese Mint and Official Printing Office

Abstract

***Face morphing** is a problem in computer graphics with numerous artistic and forensic applications. It is challenging due to variations in pose, lighting, gender, and ethnicity. This task consists of a **warping** for feature alignment and a **blending** for a seamless transition between the warped images. We propose to leverage **coord-based neural networks** to represent such warpings and blendings of face images. The results of our experiments indicate that our method is competitive with both classical and generative models. The source-code for our method may be found at: https://github.com/schardong/ifmorph*

1. Introduction

Image warping is a continuous transformation mapping points of the image support to points in a second domain. The process of warping an image has applications ranging from correcting image distortions caused by lens or sensor imperfections [5] to creating distortions for artistic/scientific purposes [4]. Warping finds a special application in creating *image morphings* [6], where it is used to align corresponding features. By gradually aligning the image features using the warping, we obtain a smooth transition between them.

We assume the warpings to be parameterized by smooth maps. Besides obtaining smooth transitions, this allows us to use its derivatives to constrain the deformation, such as approximating it as a minimum of a *variational problem*. Feature alignment can be specified using *landmarks* to establish correlations between two images.

In this work, we use *coord-based neural networks*, which we call *neural warpings*, to parameterize image warpings. This approach enables us to calculate the derivatives in closed form, eliminating the need for discretization. We also employ a time parameter, to represent smooth transitions. By incorporating the derivatives into the loss function, we can regularize the network and easily add constraints by summing additional terms. To train a neural warping, we propose a *loss function* consisting of two main terms. First, a *data constraint* ensures that the warping fits the given keypoint correspondences. Second, we *regularize* the neural warping using the *thin-plate* energy to minimize distortions.

We use neural warping to model *time-dependent* morphings of face images, thus aligning the image features over time. Afterward, we explore the flexibility of coord-based neural networks to define two blending techniques. First, we blend the aligned image warpings in the *signal domain* using point-wise interpolation. Second, we propose to blend the image warpings in the *gradient-domain* of the signals. For this, we introduce another network to represent the morphing and train it to satisfy the corresponding variational problem. An extended version of this work presents additional experiments and ablation studies to corroborate our claims [14].

Our contributions can be summarized as follows:

- The introduction of a time-dependent **neural warping** which encodes in a single network the *direct* and *inverse* transformations needed to align two images along time. We use the warping to transport the images and their derivatives from the initial states to intermediate times.
- The neural network is **smooth**, both in space and time, enabling the use of its derivatives in the loss function. We exploit it to define an implicit regularization using the *thin-plate* energy which penalizes distortions. Thus, the landmarks follow a path that minimizes this energy instead of a straight line, as in classical approaches.
- The neural warping model is **compact**. We achieved accurate warping using a MLP composed of a single hidden layer with 128 neurons, although our ablation studies indicate that smaller networks would work for specific cases.

[*] These authors contributed equally to this work.
Project page: https://schardong.github.io/ifmorph

- We blend the resulting aligned image warpings to define a time-dependent **morphing**, distinguishing it from current methods that focus on a single blend. For the case of blending in the gradient-domain, we use another neural network (**neural morphing**).

2. Related Works

The first algorithms for face morphing were simple *cross-dissolves*, i.e., pixel interpolation between target images [16]. However, the resulting morphings are substandard unless the images are aligned, resulting in artifacts. To overcome this, *mesh-based* alignment was used before the interpolation stage, shifting the complexity to the image alignment. Beier and Neely [2] further refined the process using line correspondences and an interface to align them. Liao et al. [10] exploited halfway domains, *thin-plate* splines, and *structural similarity* to create a discrete vector field to warp the images.

The above morphing approaches are landmark-based, as is ours. Recently, generative methods, such as StyleGANs [7–9] and diffAE [13], have also been used to interpolate between faces. In contrast to these methods, ours is *smooth* in both time and space, as we have a differentiable curve tracking the path of each image point during warping. Moreover, our approach exploits the recent *implicit neural representations*, which employ coord-based neural networks [15] to parameterize the images. Hence, we eliminate the need for interpolation and image resampling. This approach has also been used in the context of generative models [1] and multiresolution image representation [11].

Furthermore, by implicitly representing the images, we obtain their *derivatives in closed form* through automatic differentiation, which is not possible with previous landmark and generative approaches. This allows efficient use of the gradient during the training/analysis. Moreover, composing the warping and images results in the warped images with gradients given by the product of the warping Jacobian and the image gradient.

An important step in our warping is the incorporation of the time variable as input of the neural warping. Combined with the above advantages, this enables the creation of continuous, smooth, and compact warpings. This also allows us to constrain the landmark paths over time by minimizing distortions, unlike classical methods.

Our morphing approach disentangles the warping from the blending. This allows for the use of different blendings, such as Poisson image blending.

3. Methodology

3.1. Background and Notation

We represent an *image* by a function $\mathrm{I}:\Omega\subset\mathbb{R}^2\to\mathcal{C}$, where Ω is the image *support* and $\mathcal{C}$ is the *color space*, and parameterize it using a (coord-based) neural network $\mathrm{I}_\theta:\mathbb{R}^2\to\mathcal{C}$ with parameters θ. To train the *neural image* I_θ such that it approximates I, we can optimize $\int_\Omega(\mathrm{I}-\mathrm{I}_\theta)^2\,dx$. This work explores *coord-based neural networks* to morph *neural images* using a novel neural *warping* approach.

We assume that a coord-based neural network is a *sinusoidal* multilayer perceptron (MLP) $f_\theta(p):\mathbb{R}^n\to\mathbb{R}^m$ defined as the composition $f_\theta(x)=W_d\circ f_{d-1}\circ\cdots\circ f_0(x)+b_d$ of d *sinusoidal layers* $f_i(x_i)=\sin(W_ix_i+b_i)=x_{i+1}$, where $W_i\in\mathbb{R}^{n_{i+1}\times n_i}$ are the weight matrices, and $b_i\in\mathbb{R}^{n_{i+1}}$ are the biases. The union of these parameters defines θ. The integer d is the *depth* of f_θ and n_i are the layers *widths*.

The MLP f_θ is smooth because its layers are composed of smooth maps, and we can compute its derivatives in closed form using automatic differentiation. This property plays an important role in our method since it allows using derivatives for implicit regularization of the warpings and morphings.

3.2. Overview of Neural Morphing

This section introduces the *neural morphing* of two images. It consists of a *neural warping* to align the features of the image and a *neural blending* of the resulting warped images.

Specifically, let $\mathrm{I}_0,\mathrm{I}_1:\mathbb{R}^2\to\mathcal{C}$ be two neural images, we represent their *neural morphing* using a (time-dependent) neural network $\mathcal{I}:\mathbb{R}^2\times[0,1]\to\mathcal{C}$ subject to $\mathcal{I}(\cdot,i)=\mathrm{I}_i(\cdot)$, for $i=0,1$. Thus, for each t we have an image $\mathcal{I}(\cdot,t)$, and varying t results in a video interpolating I_i. To define the morphing $\mathcal{I}$, we **disentangle** the spatial deformation (*warping*), used to align the corresponding *features* of I_i along the time, from the *blending* of the resulting warped images.

For the warping, we use pairs of *landmarks* $\{p_j,q_j\}$, with $j\in\mathbb{N}^*$ being the *landmark index*, sampled from the domains of I_0 and I_1 providing feature correspondences. Then, we seek a warping $\mathbf{T}:\mathbb{R}^2\times[-1,1]\to\mathbb{R}^2$ satisfying the *data constraints*:

- The curves $\mathbf{T}(p_j,t)$ and $\mathbf{T}(q_j,t-1)$, with $t\in[0,1]$, has p_j and q_j as end points;
- For each $t\in(0,1)$, we require $\mathbf{T}(p_j,t)=\mathbf{T}(q_j,t-1)$.

Thus, the values $\mathrm{I}_0(p_j)$ and $\mathrm{I}_1(q_j)$ can be blended along the path $\mathbf{T}(p_j,t)$. In points $x\neq p_j$, we employ the well-known *thin-plate* energy to force the transformations to be as affine as possible. The resulting network $\mathbf{T}$ deforms I_i along the time resulting in the *warpings* $\mathcal{I}_i:\mathbb{R}^2\times[0,1]\to\mathcal{C}$ defined as:

$$\mathcal{I}_i(x,t):=\mathrm{I}_i\big(\mathbf{T}(x,i-t)\big). \tag{1}$$

Given a point (x,t), to evaluate x in I_i we move it to time $t=i$, for $i=0,1$, which is done by $x_i:=\mathbf{T}(x,i-t)$. Note that for x_0 and x_1, we need the inverse and direct of $\mathbf{T}$ since it employs negative and positive time values.

Then we obtain the image values by evaluating $\mathrm{I}_i(x_i)$. Moreover, we can move a vector v_i at x_i to x, at time t, considering the product $v_i\cdot\mathrm{Jac}(\mathbf{T}(x,i-t))$, where Jac is the Jacobian. In Section 3.4, we use such property and consider $v_i=\nabla\mathrm{I}_i(x_i)$ to blend the images in the *gradient domain*.

We blend the resulting aligned warpings $\mathcal{I}_i$ to define the desired morphing $\mathcal{I}:\mathbb{R}^2\times[0,1]\to\mathcal{C}$. We use two blending approaches: a simple linear interpolation $\mathscr{I} = (1-t)\mathscr{I}_0 + t\mathscr{I}_1$, and blending in the *gradient domain* using the Poisson equation.

The following steps summarize the procedure of morphing two images I_i:

- Extract **key points** $\{p_j, q_j\}$ in the domains of the face images I_0 and I_1, providing feature correspondence.
- Define and train the **neural warping** $\mathbf{T}: \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ to align the key points $\{p_j, q_j\}$ while penalizing distortions using the thin-plate energy. This produces the image warpings $\mathscr{I}_i$ that align the features of I_i along time;
- Blend $\mathscr{I}_i$ to define the **morphing** $\mathscr{I}: \mathbb{R}^2 \times \mathbb{R} \to \mathcal{C}$ of I_i. Besides linear interpolation, we also consider a representation for $\mathscr{I}$ using a sinusoidal MLP and exploit its flexibility to train in the *gradient domain*.

3.3. Neural warping

This section presents the *neural warping*, a neural network that aligns features of the target images along time. Precisely, we model it using a sinusoidal MLP $\mathbf{T}: \mathbb{R}^2 \times [-1, 1] \to \mathbb{R}^2$, and require the following properties:

- $\mathbf{T}(\cdot, 0)$ is the *identity* (Id);
- For each $t \in [-1, 1]$, we have that $\mathbf{T}_{-t}$ is the *inverse* of $\mathbf{T}_t$.

The corresponding deformation of an image $\mathrm{I}: \mathbb{R}^2 \to \mathcal{C}$ by $\mathbf{T}$ is defined using $\mathscr{I}(\cdot, t) = \mathrm{I} \circ \mathbf{T}(\cdot, -t)$ which uses the inverse $\mathbf{T}_{-t}$ of $\mathbf{T}_t$. That is one of the reasons we require the inverse property. In fact, if $\mathbf{T}$ holds such a property, there is no need to invert the *direct* warp $\mathbf{T}_t$, which is a difficult task in general. For simplicity, we say that $\mathscr{I}$ is a *warping* of I. Note that at $t = 0$, we have $\mathscr{I}(\cdot, 0) = \mathrm{I}$ because $\mathbf{T}(\cdot, 0) = \mathrm{Id}$. Thus, $\mathscr{I}$ evolves the initial image I along time.

We could avoid using the inverse map $\mathbf{T}_{-t}$ by employing a sampling $\{\mathrm{I}_{ij}\}$ of I on a regular grid $\{x_{ij}\}$ of the image support. Then, $\{\mathrm{I}_{ij}\}$ are samples of the warped image $\mathrm{I} \circ \mathbf{T}_{-t}$ at points $\{\mathbf{T}_t(p_{ij})\}$. However, this approach has the drawbacks of resampling $\mathrm{I} \circ \mathbf{T}_{-t}$ in a new regular grid which can result in *holes* and relies on interpolation techniques. Our method avoids such problems since it will be trained to fit the property $\mathbf{T}_t \circ \mathbf{T}_{-t} = \mathrm{Id}$ for $t \in [-1, 1]$.

Observe that, for each t, the map $\mathbf{T}_t$ approximates a *diffeomorphism* since it is a smooth sinusoidal MLP with an inverse also given by a sinusoidal MLP $\mathbf{T}_{-t}$ since $\mathbf{T}_t \circ \mathbf{T}_{-t} = \mathrm{Id}$.

3.3.1 Loss function

Let $\mathrm{I}_0, \mathrm{I}_1: \mathbb{R}^2 \to \mathcal{C}$ be neural images and $\{p_j, q_j\}$ be the *source* and *target* points sampled from the supports of I_0 and I_1 that provide feature correspondences. Let $\mathbf{T}: \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ be a sinusoidal MLP, we train its parameters θ so that $\mathbf{T}$ approximates a warping aligning the key points p_j and q_j along time. For this, we use the following loss functional.

$$\mathscr{L}(\theta) = \mathscr{W}(\theta) + \mathscr{D}(\theta) + \mathscr{T}(\theta). \tag{2}$$

Where $\mathscr{W}(\theta)$, $\mathscr{D}(\theta)$, $\mathscr{T}(\theta)$ are the *warping*, *data*, and *thin-plate* constraints. $\mathscr{W}(\theta)$ requires the network $\mathbf{T}$ to satisfy the identity and inverse properties of the warping definition.

$$\mathscr{W}(\theta) = \underbrace{\int_{\mathbb{R}^2} \|\mathbf{T}(x, 0) - x\|^2 dx}_{\text{Identity constraint}} + \underbrace{\int_{\mathbb{R}^2 \times \mathbb{R}} \big\|\mathbf{T}\big(\mathbf{T}(x, t), -t\big) - x\big\|^2 dxdt}_{\text{Inverse constraint}}. \tag{3}$$

The *identity* constraint forces $\mathbf{T}_0 = \mathrm{Id}$ and, the *inverse* constraint asks for $\mathbf{T}_{-t}$ to be the inverse of $\mathbf{T}_t$ for all $t \in \mathbb{R}$.

The *data constraint* $\mathscr{D}(\theta)$ is responsible for forcing $\mathbf{T}$ to move the source points p_j to the target points q_j such that their paths match along time. For this, we simply consider:

$$\mathscr{D}(\theta) = \int_{[0,1]} \|\mathbf{T}(p_j, t) - \mathbf{T}(q_j, 1 - t)\|^2 \, dt \tag{4}$$

Note that $\mathscr{D}$ is asking for $\mathbf{T}(p_j, 1) = q_j$ and $\mathbf{T}(q_j, -1) = p_j$ because at the same time $\mathscr{W}$ is forcing the identity property. Moreover, it forces $\mathbf{T}(p_j, t) = \mathbf{T}(q_j, 1 - t)$ along time, thus, as observed at the beginning of this section, this is the required property for the key points $\{p_j, q_j\}$ be aligned along time. Since we assume $\mathbf{T}$ to be a sinusoidal MLP, the resulting warping provides a smooth deformation that moves the source points to the target points.

However, $\mathscr{D}$ does not add restrictions on points other than the source and target points. Even assuming $\mathbf{T}$ to be smooth the resulting warping would need some regularization, such as minimizing distortions. For this, we propose a *regularization* which penalizes distortions of the transformations $\mathbf{T}_t$ using the well-known the *thin-plate* energy [3, 5]:

$$\mathscr{T}(\theta) = \int_{\mathbb{R}^2 \times \mathbb{R}} \|\mathbf{Hess}\,(\mathbf{T})\,(x, t)\|_F^2 \, dxdt. \tag{5}$$

$\mathscr{T}$ regularizes $\mathbf{T}$ and works as a bending energy term penalizing deformation, at each space-time point (x, t), based on the derivatives of $\mathbf{T}$. This helps eliminate global effects that may arise from considering only data and warping constraints. It is important to note that we have incorporated the time variable into the thin-plate energy $\mathscr{T}$.

By using a sinusoidal MLP to model $\mathbf{T}$ and training it with $\mathscr{W}$ while regularizing with the thin-plate energy, we achieve robust warpings, see Fig 1 for an alignment between two images.

3.4. Neural Blending

Let $\mathrm{I}_i: \mathbb{R}^2 \to \mathcal{C}$ be two neural images and $\mathbf{T}: \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ be a neural warping aligning their features. Specifically, the images I_i are deformed by $\mathbf{T}$ along time and Eq 1 gives the corresponding warpings $\mathscr{I}_i(x, t) = \mathrm{I}_i\big(\mathbf{T}(x, i - t)\big)$. Then, we blend $\mathscr{I}_i$ or their derivatives to construct a morphing $\mathscr{I}: \mathbb{R}^2 \times \mathbb{R} \to \mathcal{C}$ of the initial images I_i. A naive blending

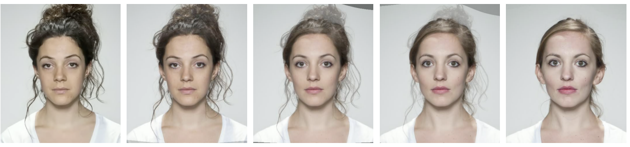

Figure 1. A neural warping $\mathbf{T}$ continuously aligning two face images along time. We use $\mathbf{T}$ to create their aligned warpings $\mathscr{J}_i$. The morphing $(1-t)\mathscr{J}_0+t\mathscr{J}_1$ was sampled at $t=0, 0.25, 0.5, 0.75, 1$.

approach could be defined directly from $\mathscr{J}_i$ by interpolating using $\mathscr{J}(x,t) = (1-t)\mathscr{J}_0(x,t) + t\mathscr{J}_1(x,t)$. Thus, at $t=0$ and $t=1$, we obtain $\mathscr{J}_0$ and $\mathscr{J}_1$, respectively (See Fig 1). Note that $\mathscr{J}$ is a smooth function both in time and space.

3.4.1 Blending in the gradient domain

Interpolating I_i does not allow us to keep parts of one of the images unchanged during the morphing, e.g. the complement region of the face. To address these issues, inspired by the *Poisson image editing* technique [12], we propose to blend I_i by solving a *boundary value problem* in $\mathbb{R}^2 \times \mathbb{R}$ to handle smooth animations and model $\mathscr{J}$ by a neural network.

We use the Jacobians $\text{Jac}(\mathscr{J}_i)$ of the warpings $\mathscr{J}_i$ to train $\mathscr{J}$. We restrict the morphing support to $S{=}[-1,1]^2{\times}[0,1]$, with $[-1,1]^2$ representing the image domain and $[0,1]$ is the time interval. Let $\Omega \subset S$ be an open set used for blending $\mathscr{J}_i$, such as the interior of the face path, and let $\mathscr{J}^*\colon S \to \mathbb{R}$ be a known function on $S-\Omega$ (it could be either $\mathscr{J}_0$ or $\mathscr{J}_1$). Finally, let U be a matrix field obtained by blending $\text{Jac}(\mathscr{J}_i)$, for example, $U = (1-t)\text{Jac}(\mathscr{J}_0) + t\text{Jac}(\mathscr{J}_1)$. A common way to extend $\mathscr{J}^*$ to Ω is by solving:

$$\min\int_{\Omega}\|\text{Jac}(\mathscr{J})-U\|^2 dxdt \text{ subject to } \mathscr{J}|_{S-\Omega}=\mathscr{J}^*|_{S-\Omega}. \tag{6}$$

We propose to use this variational problem to define the following loss function to train the parameters θ of $\mathscr{J}$.

$$\mathscr{M}(\theta)=\underbrace{\int_{\Omega}\|\text{Jac}(\mathscr{J})-U\|^2\, dxdt}_{\mathscr{C}(\theta)}+\underbrace{\int_{S-\Omega}(\mathscr{J}-\mathscr{J}^*)^2 dxdt}_{\mathscr{B}(\theta)}. \tag{7}$$

The *cloning term* $\mathscr{C}(\theta)$ fits $\mathscr{J}$ to the primitive of U in Ω, and the *boundary constraint* $\mathscr{B}(\theta)$ forces $\mathscr{J}=\mathscr{J}^*$ in $S{-}\Omega$. Thus, $\mathscr{M}$ trains $\mathscr{J}$ to *seamless clone* the primitive of U to $\mathscr{J}^*$ in Ω. Unlike classical approaches that rely on pixel manipulation, seamless cloning operates on the image gradients.

Since the images I_i contain faces and $\mathbf{T}$ aligns their features, we define Ω as the path of the facial region over time. Specifically, let Ω_0 be the region containing the face in I_0, define Ω by warping Ω_0 along time using $\mathbf{T}$, i.e., $\Omega = \cup_{t\in[0,1]}\mathbf{T}_t(\Omega_0)$. Note that the deformation of Ω_0 uses the direct deformation $\mathbf{T}_t$ while the warped image $\mathscr{J}_0$ uses the inverse $\mathbf{T}_{-t}$. The use of both inverse/direct deformations encoded in our neural warping avoids the need to compute inverses at inference time. Finally, for each t, $\mathbf{T}$ aligns the faces I_i in the region $\mathbf{T}_t(\Omega_0)$. Thus, $\mathscr{M}$ trains $\mathscr{J}$ to morph the face in I_0 into the face in I_1 while cloning the result to $\mathscr{J}_0$ on $S-\Omega$.

Besides choosing U as a linear interpolation of $\text{Jac}(\mathscr{J}_i)$, which we call the *averaged seamless cloning* case, we could choose $U = \text{Jac}(\mathscr{J}_1)$ and $\mathscr{J}^* = \mathscr{J}_0$. So, the resulting loss function $\mathscr{M}$ forces $\mathscr{J}$ to *seamless clone* the face $\mathscr{J}_1$ to the corresponding region of $\mathscr{J}_0$.

It may be desirable to combine features of $\mathscr{J}_i$, however an interpolation of $\text{Jac}(\mathscr{J}_i)$ can lead to loss of details. To avoid it, we extend the approach in [12], which allows mixing the features of both images. At each (x,t), we retain the stronger of the variations in the warpings by choosing $U{=}\text{Jac}(\mathscr{J}_0)$ if $\|\text{Jac}(\mathscr{J}_0)\| > \|\text{Jac}(\mathscr{J}_1)\|$, and $U{=}\text{Jac}(\mathscr{J}_1)$, otherwise. The resulting loss function $\mathscr{M}$ forces $\mathscr{J}$ to learn a *mixed seamless clone* of $\mathscr{J}_i$. Fig 2 shows examples of neural blending.

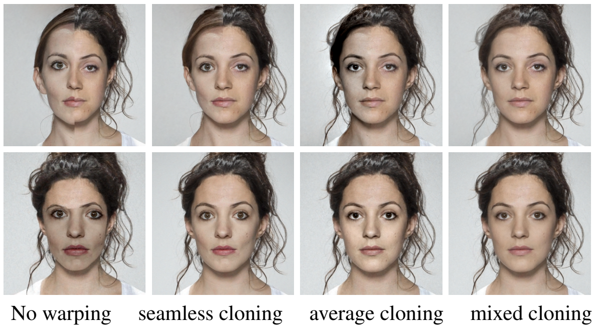
No warping seamless cloning average cloning mixed cloning

Figure 2. Comparing different neural blendings of two faces I_i. Line 1/2 shows examples of cloning the half-space region of I_1 into I_0. In Column 1 we do not align the image landmarks, the remaining columns use our neural warping for the alignment. Column 2 uses $U{=}\text{Jac}(\mathscr{J}_1)$ and $\mathscr{J}^*{=}\mathscr{J}_0$ in the neural blending. Columns 3 and 4 applies the mixed and normal seamless clone respectively.

4. Conclusions

We proposed a face morphing by leveraging coord-based neural networks. We exploited their smoothness to add energy functionals to warp and blend target images seamlessly without the need of derivative discretizations.

Our method ensures continuity in both space and time coordinates, resulting in a smooth transition between images. By operating on a smooth representation of the underlying images, we eliminate the need for pixel interpolation/resampling.The seamless blending of the target images is achieved through the integration of energy functionals, ensuring their harmonious clone. The resulting morphs exhibit a high level of visual fidelity and maintain the overall structure and appearance of the target faces.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 2

[2] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. *ACM SIGGRAPH computer graphics*, 26(2):35–42, 1992. 2

[3] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989. 3

[4] Robert Carroll, Aseem Agarwala, and Maneesh Agrawala. Image warps for artistic perspective manipulation. In *ACM SIGGRAPH 2010 papers*, pages 1–9. 2010. 1

[5] Chris A Glasbey and Kantilal Vardichand Mardia. A review of image-warping methods. *Journal of applied statistics*, 25(2):155–171, 1998. 1, 3

[6] Jonas Gomes, Lucia Darsa, Bruno Costa, and Luiz Velho. *Warping & morphing of graphical objects*. Morgan Kaufmann, 1999. 1

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8107–8116. Computer Vision Foundation / IEEE, 2020.

[9] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 852–863. Curran Associates, Inc., 2021. 2

[10] Jing Liao, Rodolfo S Lima, Diego Nehab, Hugues Hoppe, Pedro V Sander, and Jinhui Yu. Automating image morphing using structural similarity on a halfway domain. *ACM Transactions on Graphics (TOG)*, 33(5): 1–12, 2014. 2

[11] Hallison Paz, Daniel Perazzo, Tiago Novello, Guilherme Schardong, Luiz Schirmer, Vinicius da Silva, Daniel Yukimura, Fabio Chagas, Helio Lopes, and Luiz Velho. Mr-net: Multiresolution sinusoidal neural networks. *Computers & Graphics*, 2023. 2

[12] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 4

[13] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[14] Guilherme Schardong, Tiago Novello, Hallison Paz, Iurii Medvedev, Vinícius da Silva, Luiz Velho, and Nuno Gonçalves. Neural implicit morphing of face images, 2024. 1

[15] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[16] George Wolberg. Image morphing: a survey. *The visual computer*, 14(8):360–372, 1998. 2