Neural Facial Deformation Transfer

Prashanth Chandran, Loïc Ciccone, Gaspard Zoss, Derek Bradley

DisneyResearch|Studios, Switzerland

prashanthchandran.pc@gmail.com, gaspard.zoss@gmail.com, loic.ciccone,derek.bradley@disneyresearch.com

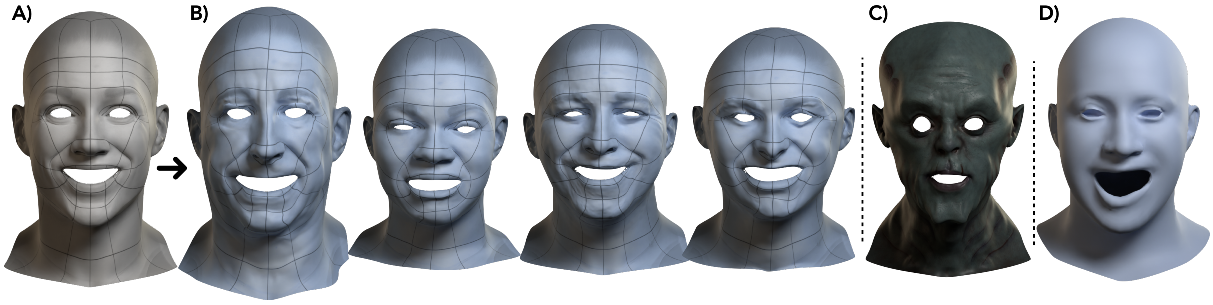


Figure 1: *We present Neural Facial Deformation Transfer (NFDT); a method to transfer facial expressions in high fidelity from a template character (A), to unseen target characters (B), humanoid creatures (C), and even across varying mesh topologies (D).*

Abstract

We address the practical problem of generating facial blendshapes and reference animations for a new 3D character in production environments where blendshape expressions and reference animations are readily available on a pre-defined template character. We propose Neural Facial Deformation Transfer (NFDT); a data-driven approach to transfer facial expressions from such a template character to new target characters given only the target's neutral shape. To accomplish this, we first present a simple data generation strategy to automatically create a large training dataset consisting of pairs of template and target character shapes in the same expression. We then leverage this dataset through a decoder-only transformer that transfers facial expressions from the template character to a target character in high fidelity. Through quantitative evaluations and a user study, we demonstrate that NFDT surpasses the previous state-of-the-art in facial expression transfer. NFDT provides good results across varying mesh topologies, generalizes to humanoid creatures, and can save time and cost in facial animation workflows.

CCS Concepts

• ***Computing methodologies*** → ***Shape modeling; Animation;***

1. Introduction

Highly skilled animation artists invest a great amount of time and effort to create believable facial animations for both animated films and high end visual effects in live-action films. A common first step in the facial animation workflow for a new character is the creation of expression blendshapes, and reference animations that establish the range of motion of the character. In production environments where artists work on one project after another, sculpting facial blendshapes and creating reference animations from scratch for each character can be a repetitive and time consuming endeavour. So to help with this task across projects, artists usually have access to a library of common expression blendshapes and reference animations that are defined on a template character. This library of facial shapes and animations is built and continuously refined over time, and is used to bootstrap the facial animation workflow for a new character. When an artist sets forth animating a new character, they can start by quickly transferring over blendshapes and reference animations from the template character library to the rest (neutral) shape of the new character, and proceed to refine them from there. In this work, we present a simple, and practical data-driven approach to address the transfer of facial expressions from a template character to a new 3D target character that is defined only by its neutral shape.

2. Related Work

Previous work to transfer facial expressions from one character to another can be categorized into geometric and data-driven approaches. Geometric approaches like Deformation Transfer [SP04] compute local deformations between a rest and a deformed template shape and transfer those to a new character in the rest shape. Another example of a geometric approach is example based facial rigging (EBFR) [LWP10] where blendshapes for a new character are optimised for, given a template blendshape basis and exemplar expression shapes of the new character. The primary challenge with geometric approaches is that they struggle to capture the unique style in which individuals perform similar expressions by activating their facial muscles in different ways. Data-driven approaches for facial expression transfer on the other hand, can capture these idiosyncrasies by learning a disentangled representation of identity and expression from a large collection of 3D shapes using supervised or unsupervised learning [QSA*23, AGK*22, WLL*23, YZC*24]. Most of these data-driven approaches, however, represent facial expressions using low-dimensional blendweights that cannot explain subtle facial expressions faithfully.

3. Method

Our approach called Neural Facial Deformation Transfer (NFDT) marries these two paradigms with a data-driven approach to deformation transfer wherein the expression to be transferred to a target character is provided directly as an expression shape of the template character. Importantly, NFDT operates without a rig inversion step (or an encoder) that bounds the template expression to a predefined basis. As a result our method can transfer arbitrary facial expressions from a template character, which may be obtained through conventional blendshape based animation, manual sculpting or any other means. NFDT therefore has the low setup cost, and flexibility benefits of geometric methods like Deformation Transfer, while adapting to subject specific idiosyncrasies due to it's data-driven nature.

At a high level, NFDT requires two inputs (see Fig. 3) i) a deformed template shape performing some facial expression ii) a target shape in the neutral expression, and produces as output the deformed target shape in the same expression as the template character. We learn this expression transfer through supervised learning, and require pairs of corresponding shapes of the template and target characters in the same expression for training. In our work, we will first describe a strategy to automatically create such a training dataset from pre-existing 3D databases with registered scans of real human subjects (Section 3.1). Then we present our decoder-only transformer network (Section 3.2) that leverages the generated dataset for facial deformation transfer of unseen subjects, without reducing the expression to a low dimension latent space.

3.1. Training Data Generation

Our training dataset should contain template and target character shape pairs in the same facial expression. To build such a dataset, we propose to transfer facial expressions captured from real (target) subjects to a pre-defined and fixed template character. While this problem may seem similar to facial expression transfer at first, transferring expressions from varying real target characters to a fixed template character allows for two practical advantages. First we can leverage pre-existing registered 3D facial databases with static and dynamic expressions of hundreds of subjects [CBGB20] for training. Secondly, we can leave the choice of template character to the user's discretion. In practice, this allows users to set this template character to the one for which blendshape expressions and reference animations are already available in a production environment. This also allows us to leverage the template character's blendshapes as its identity prior while creating the dataset as we will see. For easy identification, we depict the fixed template character as gray colored meshes, and the target character as blue colored meshes as seen in Fig. 2 across all our results.

Let $\mathcal{T}_0$ be the neutral template character shape, and let $\mathcal{S}_0$ and $\mathcal{S}_P$ be the neutral and deformed shapes of a target character. As we assume access to the template character's blendshapes, we also build a patch blendshape model of the template character following [CCGB22]. Further we assume that the template and target characters share the same number of vertices and connectivity. We can therefore compute the per-vertex expression displacements between $\mathcal{S}_P$ and $\mathcal{S}_0$ and transfer them to the template character $\mathcal{T}_0$ following standard delta transfer to get an initial estimate of template character $\widetilde{\mathcal{T}_P}$ in same expression.

$$\widetilde{\mathcal{T}_P} = \mathcal{T}_0 + (\mathcal{S}_P - \mathcal{S}_0) \tag{1}$$

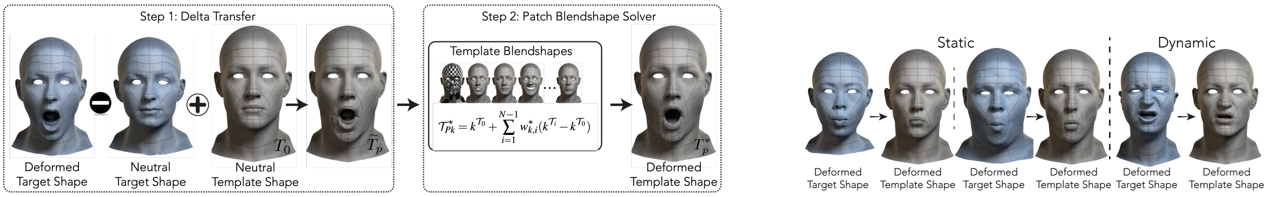
As the result of this initial delta transfer is prone to geometric artifacts and identity changes, we project the estimated shape $\widetilde{\mathcal{T}_P}$ onto the patch blendshape model of the template character. Concretely, we solve for patch blendweights $w^*_{k,i}$ using a regularized patch blendshape solver [CCGB22] with 3D position constraints to match the result of the delta transfer $\widetilde{\mathcal{T}_P}$ as closely as possible while staying true to the template character's identity. This patch blendshape solver enforces geometric consistency between neighbouring skin patches and additionally imposes anatomical constraints to result in plausible facial shapes, which makes it possible to use its results as ground truth for our method.

$$\mathcal{T}^*_{Pk} = k^{\mathcal{T}_0} + \sum_{i=1}^{N-1} w^*_{k,i}(k^{\mathcal{T}_i} - k^{\mathcal{T}_0}) \tag{2}$$

Here k is the patch index of a patch blendshape model with N shapes. Repeating this for all k patches gives us the final deformed template patch $\mathcal{T}^*_P$ performing the same expression as $\mathcal{S}_P$. We do this for every expression of all subjects in the 3D database [CBGB20] to result in the final training dataset for NFDT.

3.2. Network Architecture

To learn from the generated dataset, we design a topology agnostic, decoder-only transformer network to transfer expressions from a deformed template shape to the rest shape of a target character. As illustrated in Fig. 3, we adapt the Shape Transformer [CZG*22] for this purpose and provide corresponding vertices from the deformed template shape $\mathcal{T}^*_P$ and the neutral target shape $\mathcal{S}_0$ as input tokens to the shape transformer. Both input tokens are treated as displacements from the neutral template shape $\mathcal{T}_0$ to keep the network inputs small, and are individual processed by a template and target MLP respectively. At the other end, the output MLP predicts expression displacements for the given target character with respect to the given neutral target shape. The network is trained using a L2 loss on the predicted deformed target shape, using the adam optimizer for 200 epochs. We note that other geometry backbones [SACO22, AGK*22] could be readily used here in place of the Shape Transformer.



(a) *Our two step dataset generation consists of delta transfer step to get an initial estimate of the deformed template character $\widetilde{\mathcal{T}}_P$, followed by a patch blendshape solver to result in the final deformed template shape $\mathcal{T}_P^*$.*

(b) *Expressions transferred to the template character following our approach for both static and dynamic expressions from a 3D face dataset [CBGB20].*

Figure 2: *(a) Our dataset generation procedure (b) Examples of template/target expression pairs generated by our method.*

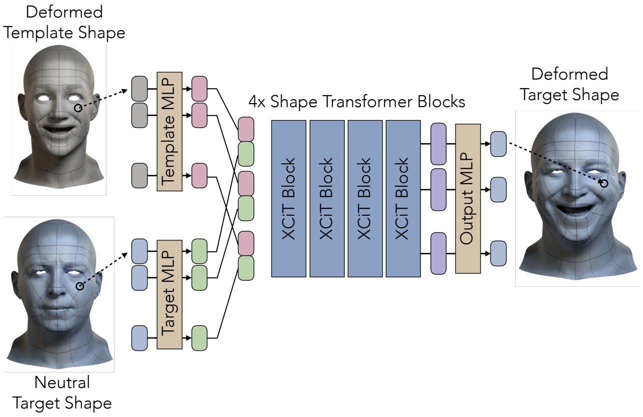


Figure 3: *Our architecture consists of pointwise template/target MLPs that serve as learned position encoding on the input positions. We use 4 shape transformer blocks with a feature dimension of 256. The pointwise output MLP maps the tokens back to 3D displacements, which are applied on top of the neutral shape to result in the final deformed target shape. At inference time, our network can run on a single Nvidia 3090Ti GPU at around 15 FPS.*

4. Results

We now demonstrate the usefulness of our method through various results and practical applications in facial animation. We also refer to our supplemental video for animation results.

Quantitative Evaluation. Recollect that our method only requires the neutral shape of the target character to transfer expressions from a deformed template shape. This is the same requirement as for popular retargeting techniques like delta transfer and deformation transfer [SP04]. In Fig. 4, we show comparisons of transferring static expressions from our template shape to a few unseen target characters using delta transfer and deformation transfer. Our method provides results that are closest to actual target character's expressions. In Table 1, we also perform a quantitative evaluation on 5 animations from test subjects, and compare NFDT to two recent approaches in geometric [CCGB22] and data-driven [CZG*22] facial animation retargeting. Again our method achieves the lowest error.

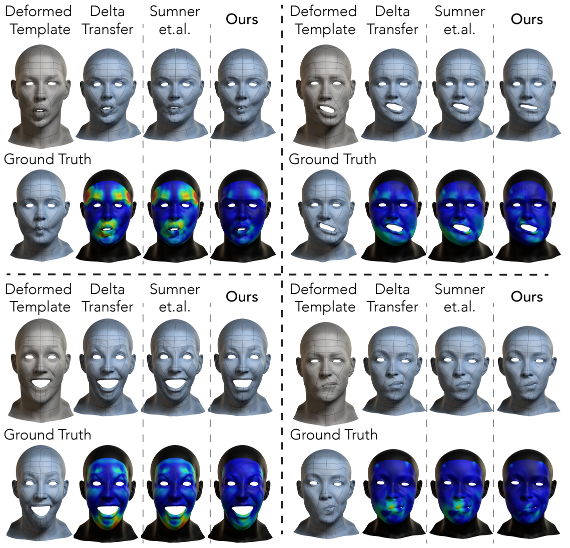


Figure 4: *Our method captures identity specific expression deformations better than widely used methods like Delta Transfer and the method of Sumner et al. [SP04]. Scale 0 mm 10 mm*

Table 1: *Reconstruction errors (in mm) for unseen animations*

Method	Mean	Median	Std. Dev.
Anatomical Model [CCGB22]	2.89	3.16	1.16
Shape Transformer [CZG*22]	1.87	1.99	0.28
NFDT (Ours)	**1.70**	**1.83**	**0.58**

User Study. For the same set of results from Table 1, we also conducted a user study to qualitatively gauge the performance of our method. Participants were shown facial animations generated by the three methods and were asked which of them was the most/least similar to the ground truth target animation. Participants also answered the same questions for a second time when only the template character's animation was shown instead of the ground truth. The ordering of the samples was randomized for each question. A total of 41 participants took part in our user study, whose results are consolidated in Fig. 5. Most participants rated our method NFDT to have results that are most similar to the ground truth animation.

Nonlinear Expression Interpolation. NFDT can capture subject specific nonlinear expression deformations and therefore can be used as a shape prior to augment linear blendshape animation. In

Proceedings published by Eurographics - The European Association for Computer Graphics.

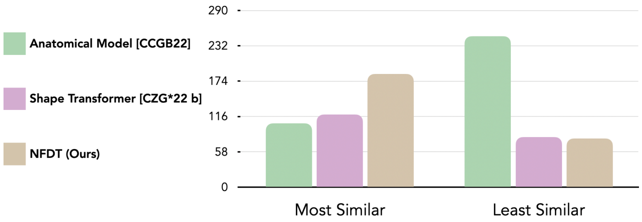


Figure 5: *A majority of the participants found that our method NFDT was the most similar to ground truth/template animation. Participants also largely rated the geometric approach [CCGB22] as least similar to the ground truth/reference animation.*

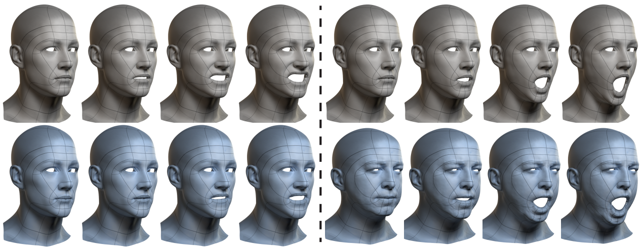


Figure 6: *Even when the template expressions (top row) are created via linear interpolation of expressions, NFDT can capture nonlinear effects in target subject's expression (bottom row), such as the opening of the mouth. Results for two target characters are shown.*

Fig. 6, we show two examples of transferring a linear interpolation between two template expressions to different target characters.

Performance Generation. NFDT can be used to transfer reference blendshapes and ROM animations to unseen target subjects, and even generalizes to humanoid creatures (see Fig. 1 and our supplemental video).

Application on FLAME Topology. As NFDT is built using only point-wise MLPs and Shape Transformer blocks, we can apply our trained model to varying target mesh topologies at inference time, without fine-tuning. In Fig. 7 we show facial expression transfers in the FLAME mesh topology. This ability also allows us to leverage off-the-shelf algorithms to obtain a neutral target shape from a single in-the-wild image, and then apply NFDT on the recovered neutral target geometry to quickly setup blendshapes and reference animations for real life characters (see Fig. 8).

5. Conclusion

We propose Neural Facial Deformation Transfer (NFDT), a method to transfer facial expressions and animations from a template character to unseen target characters given only their neutral 3D shape. In comparison to previous work, NFDT does not require a rig inversion step to describe template expressions within a pre-defined blendshape basis and can therefore faithfully represent arbitrary expression shapes produced by artists. Unlike standard Deformation Transfer [SP04], NFDT can also capture target specific expression deformations, and can operate on arbitrary topologies without requiring a mapping between them. While occasionally resulting in minor artifacts, NFDT has the potential to save cost and time in animation workflows.

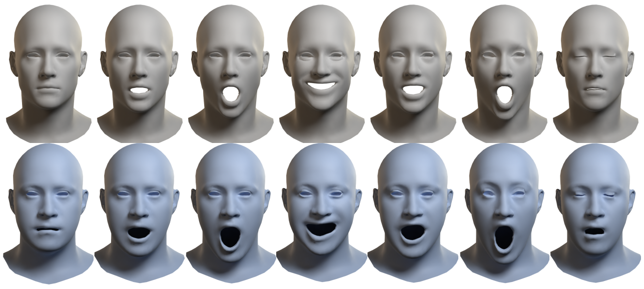


Figure 7: *NFDT can be applied to varying mesh topologies at test time without any fine-tuning, like that of the FLAME model.*

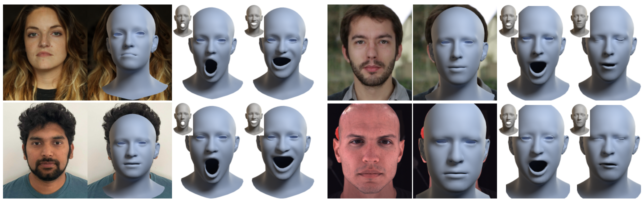


Figure 8: *By recovering the neutral shape of a target character from a single RGB image using off-the-shelf methods, NFDT can be used to generate target blendshapes and reference animations easily, and thereby saving time and cost in production environments.*

References

[AGK*22] Aigerman N., Gupta K., Kim V. G., Chaudhuri S., Saito J., Groueix T.: Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *SIGGRAPH* (2022). 2

[CBGB20] Chandran P., Bradley D., Gross M., Beeler T.: Semantic deep face models. In *International Conference on 3D Vision (3DV)* (Los Alamitos, CA, USA, nov 2020), IEEE Computer Society, pp. 345–354. doi:10.1109/3DV50981.2020.00044. 2, 3

[CCGB22] Chandran P., Ciccone L., Gross M., Bradley D.: Local anatomically-constrained facial performance retargeting. *ACM Trans. Graph. 41*, 4 (jul 2022). doi:10.1145/3528223.3530114. 2, 3, 4

[CZG*22] Chandran P., Zoss G., Gross M., Gotardo P., Bradley D.: Shape transformers: Topology-independent 3d shape models using transformers. *Computer Graphics Forum 41*, 2 (2022), 195–207. doi:https://doi.org/10.1111/cgf.14468. 2, 3

[LWP10] Li H., Weise T., Pauly M.: Example-based facial rigging. *ACM Trans. Graph. 29*, 4 (July 2010). doi:10.1145/1778765.1778769. 1

[QSA*23] Qin D., Saito J., Aigerman N., Thibault G., Komura T.: Neural face rigging for animating and retargeting facial meshes in the wild. In *SIGGRAPH 2023 Conference Papers* (2023). 2

[SACO22] Sharp N., Attaiki S., Crane K., Ovsjanikov M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph 41*, 3 (2022). doi:https://doi.org/10.1145/3507905. 2

[SP04] Sumner R. W., Popović J.: *Deformation Transfer for Triangle Meshes*, 1 ed. Association for Computing Machinery, New York, NY, USA, 2004. 1, 3, 4

[WLL*23] Wang J., Li X., Liu S., Mello S. D., Gallo O., Wang X., Kautz J.: Zero-shot pose transfer for unrigged stylized 3D characters. In *CVPR* (June 2023). 2

[YZC*24] Yang L., Zoss G., Chandran P., Gross M., Solenthaler B., Sifakis E., Bradley D.: Learning a generalized physical face model from data. *ACM Trans. Graph. 43*, 4 (July 2024). doi:10.1145/3658189. 2

Proceedings published by Eurographics - The European Association for Computer Graphics.