An Implicit Physical Face Model Driven by Expression and Style - Supplemental

LINGCHEN YANG, ETH Zurich, Switzerland
GASPARD ZOSS, DisneyResearch|Studios, Switzerland
PRASHANTH CHANDRAN, DisneyResearch|Studios, Switzerland
PAULO GOTARDO, DisneyResearch|Studios, Switzerland
MARKUS GROSS, ETH Zurich, Switzerland and DisneyResearch|Studios, Switzerland
BARBARA SOLENTHALER, ETH Zurich, Switzerland
EFTYCHIOS SIFAKIS, University of Wisconsin Madison, USA
DEREK BRADLEY, DisneyResearch|Studios, Switzerland

ACM Reference Format:
Lingchen Yang, Gaspard Zoss, Prashanth Chandran, Paulo Gotardo, Markus Gross, Barbara Solenthaler, Eftychios Sifakis, and Derek Bradley. 2023. An Implicit Physical Face Model Driven by Expression and Style - Supplemental. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3610548.3618156

This supplemental document contains more details on our implementation, additional results and comparisons with other methods.

1 MAPPING FUNCTION

Given the the canonical space $\mathcal{D}_C$, the material space $\mathcal{D}_M$ of a target identity, and a bijective mapping function ϕ between these two spaces: $\phi : \mathbf{x} \in \mathcal{D}_M \rightarrow \mathbf{X} \in \mathcal{D}_C$, and $\phi^{-1} : \mathbf{X} \in \mathcal{D}_C \rightarrow \mathbf{x} \in \mathcal{D}_M$. We define the actuation tensor field on $\mathcal{D}_C$ as $\mathcal{A}$, which will be warped to $\mathcal{D}_M$ as $\tilde{\mathcal{A}}$ to deform the target face.

The energy function defined on $\mathcal{D}_C$ is

$$E = \int_{\mathcal{D}_C} \frac{1}{2} \left\| \mathbf{F}(\mathbf{X}) - \mathbf{R}^*(\mathbf{X})\mathcal{A}(\mathbf{X}) \right\|_F^2 dV \tag{1}$$

What if we directly push forward it to $\mathcal{D}_M$.

$$E = \int_{\mathcal{D}_M} \frac{1}{2} \left\| \mathbf{F}(\mathbf{x}) \frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}} - \mathbf{R}^*(\mathbf{x})\mathcal{A}(\phi(\mathbf{x})) \right\|_F^2 \left| \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{x}} \right| dv, \tag{2}$$

where $\mathbf{F}(\mathbf{x})$ is the deformation gradient measured in $\mathcal{D}_M$, This is the typical way to view the simulation in different spaces (material space and Eulerian space). However, in current setting, the canonical space $\mathcal{D}_C$ and the target space $\mathcal{D}_M$ are two independent spaces, this pushing forward is not meaningful, e.g., the deformation gradient should not be accumulated as $\mathbf{F}(\mathbf{x}) \frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}}$. To see this, we can assume the actuation tensors are all identity matrices (no induced deformation). Then, ideally, the target identity should remain undeformed. But if we use the energy function (2) to run the simulator for the target identity, it will be dragged into the canonical space, which is not what we want.

What should change is the actuation tensor, which could be decomposed into contractile directions and the magnitudes. Instead of directly warping the actuation tensor using $\mathcal{A}(\phi(\mathbf{x}))$, which is not semantically consistent as shown in Fig. 1, we couple it with the Jacobian of ϕ^{-1}, i.e., $\frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}}$. Specifically, two things are taken into consideration. First, the magnitudes should *not* change along with $\frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}}$, otherwise there will be extra deformation induced for the rest shape of the target mesh. Second, the contractile directions should correlate with $\frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}}$, e.g., to rotate consistently, thus preserving semantic meaning. To achieve this goal, we could factorize out the rotational component $\mathbf{R}_{\phi^{-1}}$ of $\frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}}$, then the warped actuation tensor is $\tilde{\mathcal{A}}(\mathbf{x}) = \mathbf{R}_{\phi^{-1}}\mathcal{A}(\phi(\mathbf{x}))\mathbf{R}_{\phi^{-1}}^\top$. The energy function for simulation is then

$$E = \int_{\mathcal{D}_M} \frac{1}{2} \left\| \mathbf{F}(\mathbf{x}) - \mathbf{R}^*(\mathbf{x})\tilde{\mathcal{A}}(\mathbf{x}) \right\|_F^2 dv \tag{3}$$

Fig. 1 and Fig. 2 show the basic idea. In practice, we don't need to train two separate networks for ϕ and ϕ^{-1}, since we have the following implicit relation

$$\frac{\partial \phi^{-1}(\mathbf{X})}{\partial \mathbf{X}} = \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{x}}^{-1}, \tag{4}$$

therefore $\mathbf{R}_{\phi^{-1}} = \mathbf{R}_\phi^{-1} = \mathbf{R}_\phi^\top$. Putting this together, we have $\tilde{\mathcal{A}}(\mathbf{x}) = \mathbf{R}_\phi^\top \mathcal{A}(\phi(\mathbf{x}))\mathbf{R}_\phi$. As shown in Fig. 4, the network architecture for ϕ is composed of 4 SIREN layers with the hyperparameter $\omega_0 = 5$, and one linear layer. For simplicity, we train such a network for each identity. The training takes 100000 iterations, with a learning rate of $1e-4$ that starts to linearly decay to 0 after 50000 iterations. The elastic regularization weight λ_e in Eq. 2 in the main paper is set to 10. To evaluate this term, we sample N_e vertices in total: the simulation vertices plus randomly sampled points (one point per element of the simulation mesh). Training takes about 1 hour. After training, the necessary information for the warp operation is evaluated once and reused in the training of the multi-identity framework. The statistics of our 6 mapping networks ϕ are shown in Fig. 3. Note that in addition to the vertex error of the explicit surface constraint,

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12…$15.00
https://doi.org/10.1145/3610548.3618156

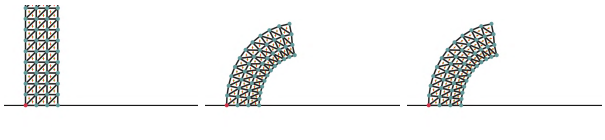

Fig. 1. Comparison of the warp methods. The first column is the template mesh with the actuation patterns (cross mark) in the canonical space. The other columns are target meshes with the actuation patterns warped with different methods. Better to zoom in to see the orientation of the cross marks.

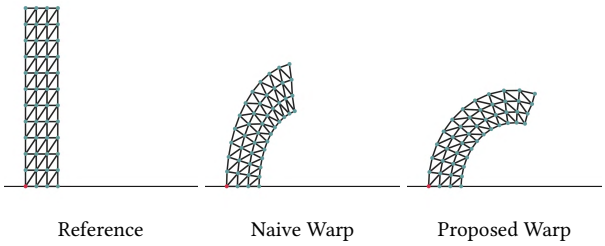

Fig. 2. The first column is the reference mesh after deformation induced by the actuation. The other columns are the target meshes after deformation induced by the actuations warped with different methods.

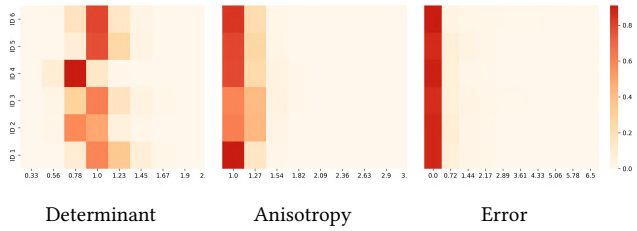

Fig. 3. The statistics of our mapping functions. The y-axis is identity number, and the x-axis is the value of the corresponding measurement. The color of the heatmap indicates the percentage of the sampled points that fall into the corresponding bin. The first column shows the heatmaps of determinant of the Jacobian matrix of the mapping functions, the second column shows the heatmaps of the ratio of the largest to the smallest singular values of the Jacobian matrix, and the third column shows the heatmaps of the vertex error.

we also report the volumetric statistics of the Jacobian of ϕ. The volumetric statistics are generated by randomly sampled the same amount of points as in evaluating the elastic regularization term.

2 MULTI-IDENTITY ARCHITECTURE AND TRAINING

Our network architecture is shown in Fig. 4. First, the input 19-dimensional expression code and 4-dimensional identity code are mapped into a higher dimensional space via two tiny MLPs (akin to learned positional encoding), and subsequently get concatenated to result in the activation code $\mathbf{z}$. $\mathbf{z}$ is the direct input to the generative transformation network $\mathcal{N}_B$, and also serves as the modulation input for the generative actuation network $\mathcal{N}_A$. We design the activation functions shown in the figure for the following considerations. Yang et al. [2022] propose to use SIREN as the backbone of $\mathcal{N}_A$, which we find to be unstable, extremely sensitive to initialization, and prone to produce noisy results. Thus, we replace all the SIREN layers with GeLU layers except the first layer which serves as the learnable positional encoding. Other positional encoding methods could also be applied here. Since GeLU activation function is unbounded from above, we use the tanh activation function to bound the modulation input. In order to add Lipschitz constraint, we augment each GeLU layer i with a Lipschitz weight normalization layer [Liu et al. 2022] that has a trainable Lipschitz bound c_i. For more details of the Lipschitz weight normalization layer, please refer to Liu et al. [2022].

Following the two-stage training strategy inspired by [Srinivasan et al. 2021; Yang et al. 2022], we commence with a plausible approximation of the actuation tensor field and jaw transformation for each target pose. This is based on passive muscle simulation [Srinivasan et al. 2021] and the tracking method delineated in [Zoss et al. 2019]. This phase aids in warming up the training without the necessity for a differentiable simulator. In the second stage, we train the network with the integration of a collision-agnostic differentiable simulator. For Eq. 7 in the main paper, we assign values of $1e-3$ to λ_{act} and $1e-6$ to λ_{lip}. We use the sampled actuation tensors used for simulation in $\mathcal{L}_{act}$. The initial stage runs for 400 epochs (roughly 16 hours), using a learning rate of $1e-4$ which linearly decays to zero after the 200th epoch. The second stage, lasting 20 epochs (approximately 30 hours), starts with a learning rate of $5e-5$ which begins a linear decay to zero right from the start. The batch size for all stages is set to 6. Notably, in the second stage, the varying simulation meshes across identities render the computational graph identity-dependent, which precludes naive identity batching. To address this, we apply a distributed data parallel strategy and train

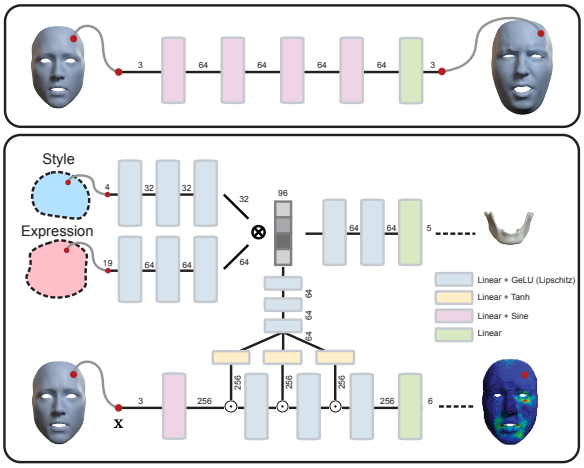

Fig. 4. The architectures of our networks. The first row shows the architecture of our mapping network ϕ. The bottom row shows the architecture of our multi-identity framework. The channel number for each layer is shown on the arrows.

our network on multiple GPUs. For the second stage, our framework takes around 6 seconds per iteration on average, covering both forward and backward passes.

2.1 Simulation

Recall that our simulation framework consists of three energy terms: shape targeting, bone attachment, and contact energies. The first two terms are based on Projective Dynamics [Bouaziz et al. 2014], where the local constraint can be generally represented as follows:

$$E_i(\mathbf{u}) = \min_{\mathbf{y}_i} \frac{\omega_i}{2} ||\mathbf{G}_i\mathbf{S}_i\mathbf{u} - \mathbf{B}_i\mathbf{y}_i||_F^2 \quad \text{s.t.} \quad C_i(\mathbf{y}_i) = 0, \tag{5}$$

where $\mathbf{u}$ denotes the simulation vertices, ω_i is a weight coefficient, and $\mathbf{y}_i$ an auxiliary variable, embodying the target position. $\mathbf{S}_i$ is a selection matrix choosing DOFs involved in E_i. $\mathbf{G}_i$ and $\mathbf{B}_i$ are designed to facilitate the distance measure. For the shape targeting energy, $\mathbf{G}_i$ maps $\mathbf{u}$ to the deformation gradient $\mathbf{F}_i$. $\mathbf{B}_i$ comes from the input actuation tensor $\mathbf{A}_i$ and $\mathbf{y}_i$ denotes the rotation matrix, projected from $\mathbf{F}_i\mathbf{A}_i$. For the bone attachment energy, $\mathbf{G}_i$ extracts the embedded bone vertex from $\mathbf{u}$, while $\mathbf{B}_i$ becomes an identity matrix and $\mathbf{y}_i$ is directly the given target position. The total energy $E(\mathbf{u})$ is the sum of all these local constraints. After converging to a local minimum, we can calculate the sensitivity matrices for the input variables of interest with implicit differentiation. For example, the sensitivity matrix of $\mathbf{u}$ with respect to $\mathbf{A}_i$ is given by:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{A}_i} = -\left(\nabla^2 E\right)^{-1} \frac{\partial \nabla E}{\partial \mathbf{A}_i}. \tag{6}$$

For collision modeling, we employ the IPC model [Li et al. 2020], which utilizes the incremental barrier energy $B(\mathbf{u})$. We set the distance threshold $\hat{d}$ to $0.001l$, where l denotes the diameter of the chosen identity. With respect to differentiable simulation, the calculation of the sensitivity matrices needs adjustment. For instance, the sensitivity matrix of $\mathbf{u}$ in relation to $\mathbf{A}_i$ is given by:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{A}_i} = -\left(\nabla^2 E + \nabla^2 B\right)^{-1} \frac{\partial \nabla_{\mathbf{u}} E}{\partial \mathbf{A}_i}. \tag{7}$$

Note that incorporating the collision model into the simulation will increase the computational cost. For each frame, simulating from the rest shape takes around 30 seconds on average. However, we find that during animation, using the previous frame as the initial state can significantly reduce the simulation time to around 6-10 seconds per frame on average.

3 OTHER EXPERIMENTAL RESULTS

Comparison with the displacement network. We demonstrate the superiority of our physics-based model by contrasting it with a displacement variant that disregards physics and directly regresses the displacement field. In order to maintain parity, the displacement field is also learned in the canonical space, employing the same Lipschitz regularization and geometric loss function as our actuation network. We utilize a similar network architecture, simply adjusting the dimension of the final layer to three.

As shown in Fig. 5, despite the displacement network's inability to manage collision, it also suffers from problems like sudden shape distortion in the lip region, further emphasizing the benefits of our physics-based approach.

We also compare the displacement network with our physics-based model in terms of style transfer task. As shown in Fig. 6, our physics-based model can better preserve the identity and expression of the source face, while the displacement network suffers from severe volume change and lip penetration in the lip region.

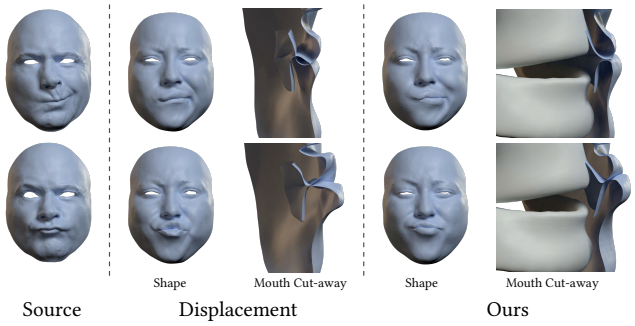

Fig. 5. Comparison between our displacement network and our actuation network in terms of retargeting.

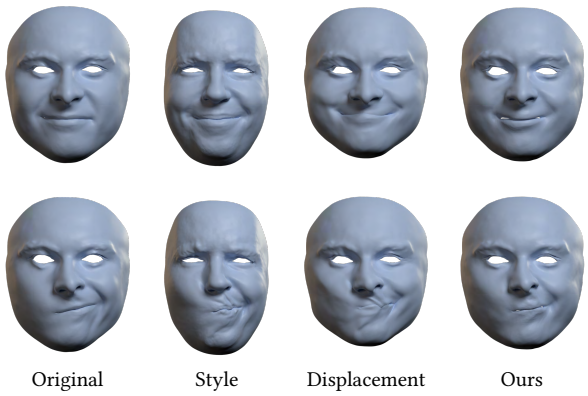

Fig. 6. Style transfer comparison between our model and the model without canonical space (Model-N). Note how the actuation field is inconsistent across the identities in Model-N for the same style.

REFERENCES

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics* 33, 4 (7 2014), 1–11. https://doi.org/10.1145/2601097.2601116

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49.

Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. 2022. Learning Smooth Neural Functions via Lipschitz Regularization. *arXiv preprint arXiv:2202.08345* (2022).

Sangeetha Grama Srinivasan, Qisi Wang, Junior Rojas, Gergely Klár, Ladislav Kavan, and Eftychios Sifakis. 2021. Learning active quasistatic physics-based models from data. *ACM Transactions on Graphics* 40, 4 (8 2021), 1–14. https://doi.org/10.1145/3450626.3459883

Lingchen Yang, Byungsoo Kim, Gaspard Zoss, Baran Gözcü, Markus Gross, and Barbara Solenthaler. 2022. Implicit neural representation for physics-driven actuated soft bodies. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–10.

Gaspard Zoss, Thabo Beeler, Markus Gross, and Derek Bradley. 2019. Accurate markerless jaw tracking for facial performance capture. *ACM Transactions on Graphics* 38, 4 (7 2019), 1–8. https://doi.org/10.1145/3306346.3323044