LEARNING ACTIVE QUASISTATIC PHYSICS-BASED MODELS FROM DATA

Sangeetha Grama Srinivasan* Qisi Wang* Junior Rojas† Gergely Klár‡ Ladislav Kavan† Eftychios Sifakis*‡

*University of Wisconsin - Madison †University of Utah ‡Weta Digital

Abstract

Our work explores the following proposition: given a training set of example poses of an active deformable object, can we learn a low-dimensional control space that could reproduce the training set and generalize to new poses? Unlike popular machine learning methods for dimensionality reduction such as auto-encoders, we model our active objects in a physics-based way. We utilize a differentiable, quasistatic, physics-based simulation layer and combine it with a decoder-type neural network. In addition, in contrast to modeling approaches where users build anatomical models from first principles, medical literature or medical imaging, we do not presume knowledge of the underlying musculature, but learn the structure and control of the actuation mechanism directly from the input data. We present a training paradigm and several scalability-oriented enhancements that allow us to train effectively while accommodating high-resolution volumetric models, with as many as a quarter million simulation elements.

Differentiable Simulator

We adopt shape targeting [1] for our actuation model, which can be defined by extending the conventional definition of energy density function Ψ to depend not only on the deformation gradient F but also on a shape target matrix $S_t \in \mathbb{R}^{3\times3}$ for each tetrahedron:

$$\Psi(\mathbf{F}, \mathbf{S_t}) = \underset{\mathbf{R}\in SO(3)}{\operatorname{argmin}} \; \mu\|\mathbf{F} - \mathbf{R}\mathbf{S_t}\|_F^2$$

Where S_t is a rotationally-invariant descriptor of the shape that an element targets, with degrees of freedom being 6. Concatenating the shape targets of all tetrahedra into a single action vector $\mathbf{a} \in \mathbb{R}^{6u}$($u$ is the number of tetrahedra), we can reformulate the quasistatic solution χ as a function of a:

$$\chi(\mathbf{a}) = \underset{\mathbf{x}\in\mathcal{C}}{\operatorname{argmin}} \; E(\mathbf{x}, \mathbf{a}) \tag{1}$$

With boundary conditions $\mathcal{C}$ and total energy over all the elements(S), $E = \sum_{e\in S} \Psi(\mathbf{F_e}, \mathbf{St_e}) = E(\mathbf{x}, \mathbf{a})$. Given force as negative gradient of the energy with respect to vertex positions, i.e. $\mathbf{f}(\mathbf{x}) = -\nabla_x E(\mathbf{x})$, Equation 1 implies that the force at the quasistatic equilibrium configuration is zero:

$$\mathbf{f}(\chi(\mathbf{a}), \mathbf{a}) = 0 \tag{2}$$

And thus:

$$\frac{d}{d\mathbf{a}}\mathbf{f}(\chi(\mathbf{a}), \mathbf{a}) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{(\chi(\mathbf{a}),\mathbf{a})} \frac{d\chi}{d\mathbf{a}} + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right|_{(\chi(\mathbf{a}),\mathbf{a})} = 0 \tag{3}$$

To optimize the loss function, $L(\mathbf{a})$, we need to obtain a formula for $\frac{dL}{d\mathbf{a}}$, which can be done via the following derivations (omitting the explicit evaluations at $(\chi(\mathbf{a}), \mathbf{a})$ for conciseness):

$$\frac{d\chi}{d\mathbf{a}} = -\frac{\partial \mathbf{f}}{\partial \mathbf{x}}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \tag{4}$$

$$\frac{dL}{d\mathbf{a}} = \frac{dL}{d\mathbf{x}}\frac{d\chi}{d\mathbf{a}} = -\frac{dL}{d\mathbf{x}}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}^{-1}\frac{\partial \mathbf{f}}{\partial \mathbf{a}} \tag{5}$$

As a final step, we take advantage of the fact that the stiffness matrix $-\partial\mathbf{f}/\partial\mathbf{x}$ is *symmetric* to rewrite Equation 5 as:

$$\frac{dL}{d\mathbf{a}} = -\underbrace{\left(\underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}}^{-1}}_{\mathbb{R}^{3n\times3n}} \underbrace{\frac{dL}{d\mathbf{x}}^{\mathsf{T}}}_{\mathbb{R}^{3n\times1}}\right)^{\mathsf{T}}}_{\mathbb{R}^{1\times3n}} \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{a}}}_{\mathbb{R}^{3n\times6u}} \tag{6}$$

Learning Framework

We start by endowing our active model with an extremely fine-grained actuation mechanism. In essence, we allow every tetrahedral element to become an individual muscle-of-sorts, able to specify its individual target shape. A quasistatic physics-based simulator reconciles all these elemental shape targets into one consistent shape, as shown in Figure 1. Of course, real-world objects would not independently actuate at such fine granularity.

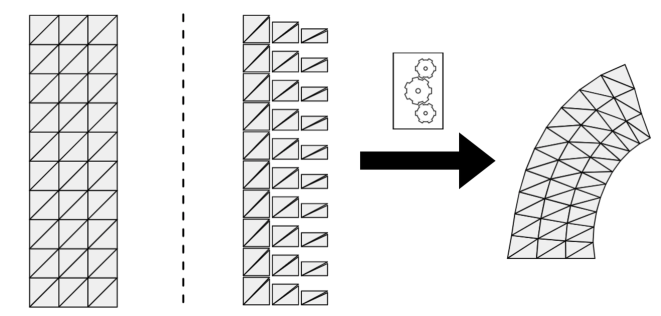

Fig. 1: Quasistatic simulator reconciles the elemental actuations into one simulated shape.

Traning Methodology

In our approach, we presume that all these fine-grained actions are a function of a lower-dimensional set of control parameters, that capture the true dimensionality of the actuation mechanism. We use a neural network to represent this mapping, and treat the control parameters as a latent vector, which will be realized through training. This learning pipeline is depicted in Figure 2.

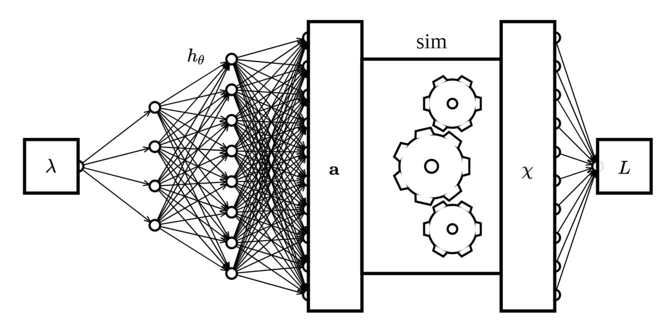

Fig. 2: Our learning pipeline is composed of a neural network coupled with a differentiable quasistatic simulator

Our first task is to generate a good initialization of the network weights representing the actuation mechanism. We do this by training a slightly different network in a supervised fashion. This network is depicted in Figure 3.

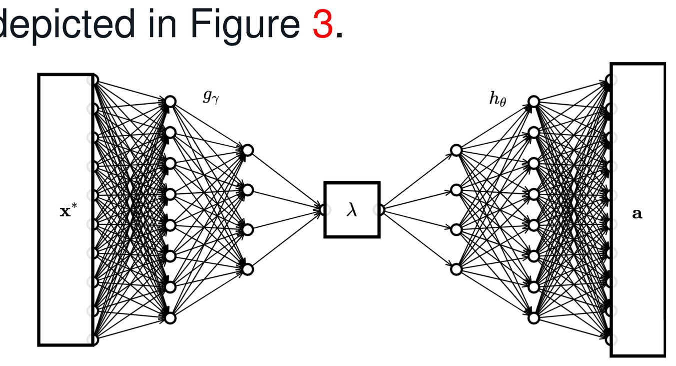

Fig. 3: Neural network trained to initialize the decoder part of the network for training with simulation

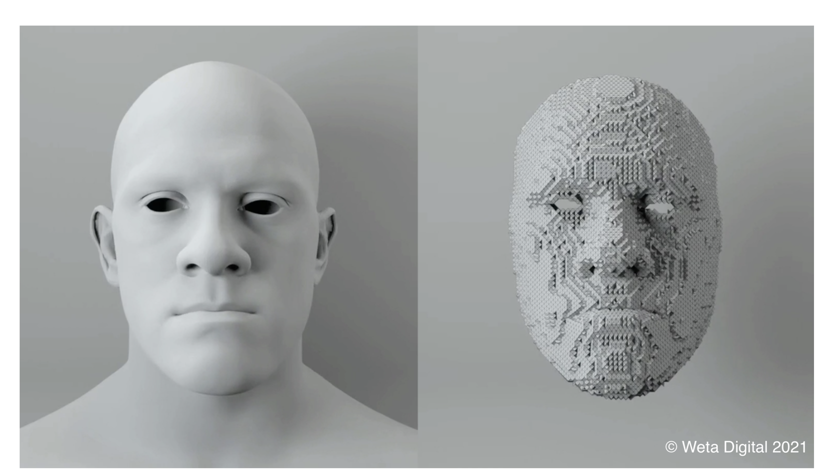

Fig. 4: Convert the surface animation data (left) into volumetric deformations (right) using embedded simulation

We use simulation to convert the surface animation data in our training set into volumetric deformations. We use an embedded simulation, as shown in Figure 4, to target the surface from the training data and obtain one possible collection of volumetric elemental targets.

Results

Using our differentiable simulator, we can produce a more accurate reconstruction of the target expression, than an auto-encoder which only operates on surface shapes and was not trained using the differentiable simulator - Figure 5.

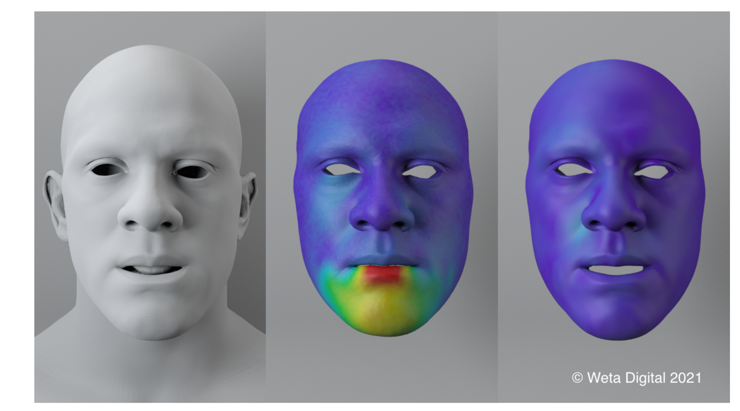

Fig. 5: Target pose (left), outputs from the auto-encoder (middle) and the end-to-end training (right)

We demonstrate the ability of our model to fit unseen data, held aside for testing. For each of the shapes in the left, we hold constant the weights of the neural network, but optimize for the latent parameters that create the best possible fit - Figure 6.

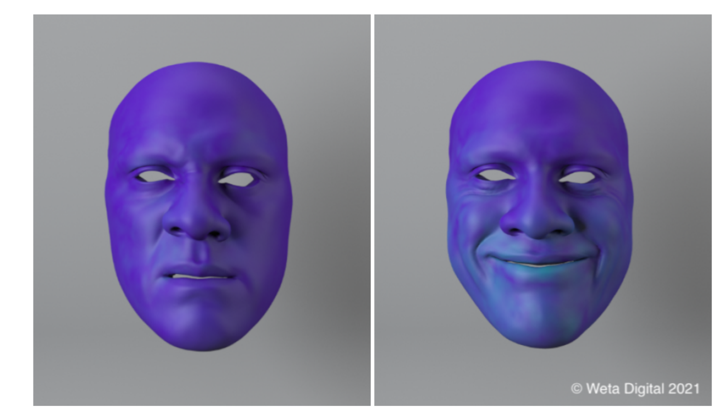

Fig. 6: Generalization of our learning pipeline to unseen poses

We demonstrate the benefit of using physics by resolving collision with teeth and self-collision on the outputs of our pipeline, as a post-processing step. For more examples, please refer to our paper.

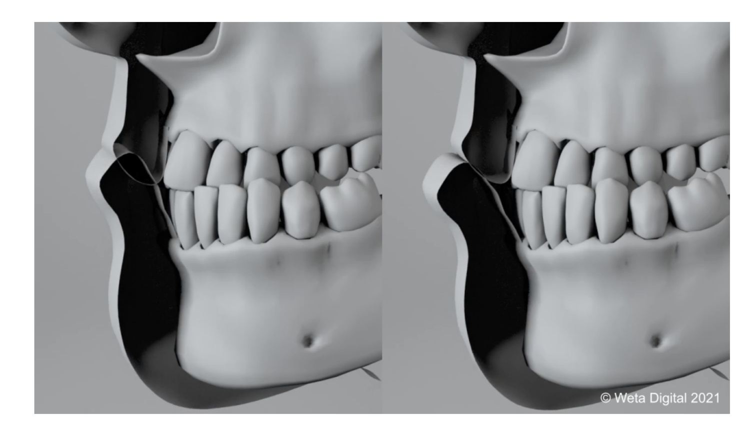

Fig. 7: Output from our pipeline (left) and post collision resolution with teeth and self-collision (right).

Acknowledgements

We thank Luca Fascione and Stephen Cullingford for valuable contributions on assets and administrative support. This research was supported in part by National Science Foundation grants CCF-1812944, IIS-1763638, IIS-1764071, IIS-2008915, IIS-2008564 and IIS-2008584.

References

[1] Gergely Klár et al. "Shape Targeting: A Versatile Active Elasticity Constitutive Model". In: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks.* SIGGRAPH '20. Association for Computing Machinery, 2020. ISBN: 9781450379717.