FaceBaker: Baking Character Facial Rigs with Machine Learning

Sarah Radzihovsky
radz@pixar.com
Pixar Animation Studios

Fernando de Goes
fernando@pixar.com
Pixar Animation Studios

Mark Meyer
mmeyer@pixar.com
Pixar Animation Studios

ABSTRACT
Character rigs are procedural systems that deform a character’s shape driven by a set of rig-control variables. Film quality character rigs are highly complex and therefore computationally expensive and slow to evaluate. We present a machine learning method for approximating facial mesh deformations which reduces rig computations, increases longevity of characters without rig upkeep, and enables portability of proprietary rigs into a variety of external platforms. We perform qualitative and quantitative evaluations on hero characters across several feature films, exhibiting the speed and generality of our approach and demonstrating that our method outperforms existing state-of-the-art work on deformation approximations for character faces.

CCS CONCEPTS
• Computing methodologies → Machine learning.

KEYWORDS
deep learning, character rigs, mesh deformation, rig simplification

INTRODUCTION
The use of film quality rigs in production poses three main challenges. First, high quality character rigs require costly deformation computations to solve for the shape of the character mesh given the animation controls. Second, although there is a desire to use high quality characters outside of our proprietary software (Presto), it is infeasible to port our computationally intensive rigs into external environments. Lastly, film quality rigs are often challenging to technically maintain and therefore difficult to reuse in new projects.

A previous attempt by Kanyuk et al. [2018] to simplify complex Presto character rigs was done by extracting a skeleton from the rig and solving for linear blend skinning weights with a smoothing term to most appealingly approximate the deformations. The skeletal skinning is adjusted with corrective shapes that are driven by rig-control variables using a sparse weight interpolant. The work of Bailey et al. [2018] also uses machine learning to approximate rig deformations. Their approach aims to overcome nonlinear body poses by splitting the mesh deformation into linear and nonlinear, letting the linear portion be computed directly from transformations of the rig’s underlying skeleton and leveraging deep learning to approximate the more cumbersome nonlinear deformations. Neither method, however, can handle facial animation.

Unlike body deformations, face deformations rely mostly on rig controls rather than the underlying skeleton, and each face vertex is affected by a much larger number of rig parameters, leading to a difficult learning problem with a high-dimensional input being mapped to each vertex. We tackle this challenging problem with a purely data-driven approach, providing a fast, portable, and long-lasting solution for approximating such face poses.

METHOD
Data Representation: Arguably the most straightforward representation of a mesh deformation is the per-vertex translation of a mesh from its rest position, relative to object space. We also experimented with representing mesh deformations in terms of the deformation gradients used to move each mesh face from its rest to posed state, however, this generally proved to generate similar results.

Training Data: For our experiments, we relied on four different types of training data: (1) film shots, (2) rig calisthenics, (3) single rig-control excitations, and (4) combinations of regional expressions. Single rig-control excitations are created by individually firing each rig-control variable uniformly between its minimum and maximum range with some refinement. These excitation shapes help the network decouple the contribution of each rig-control variable uniformly between its minimum and maximum range with some refinement. These excitation shapes help the network decouple the contribution of each rig-control variable from more global facial motions. Combinations of regional facial expressions (brows, mouth, eyes, and lids) also supplement the model with examples of localized poses that cannot be recreated by simply combining the shapes created by single rig-control excitations.

Architecture: Batches of rig-control variables are first fed into 8 dense layers of width 256, into a final...
dense layer that scales to the size of the deformation representation. The last layer’s weights are fixed to a set of the most important blendshapes selected by Principal Component Analysis (PCA) such that the blendshapes cover 99.9% of the variance in the data. Providing the network with precomputed blendshapes reduces the size and complexity of the problem. The 8th layer’s width is equal to the number of components in this PCA set. To combat diminishing gradients, we bolster our network with skip connections that help propagate the signal through each layer by adding the signal from the \( i \)th layer to the signal exiting \((i + 1)\)th layer, as shown in Figure 2.

Figure 2: We added skip connections to our architecture to help propagate early signals to deeper layers in order to avoid diminishing gradients. © Disney/Pixar.

3 RESULTS AND EVALUATION

To evaluate the accuracy of our deep learning predictions, we measure the per point mean squared error of the approximation as well as the largest single vertex error. We compare our results against the true rig deformations and combined linear blendshapes corresponding to each animation control, as shown in Figure 1.

Application: Rig Variant. Our method lends itself as an attractive rig variant due to being fast and much more lightweight than most film quality rigs without noticeable loss in deformation quality. To use our method as a lightweight rig variant, we assume that shot data cannot be relied on as available training data because the variant is used to create shots. This assumption reduces the amount of data the variant model can learn from, thereby worsening its generality and the quality of most rig deformation approximations.

Application: Backlot. The compact and universal nature of the pretrained model resulting from our approach also serves as a suitable way to preserve a character rig with little maintenance cost, which we refer to as “backlot”. For the purposes of baking a character’s rig for backlot, we assume there are many shots, exemplifying the character’s rig in motion, available as training and validation data. This additional source of data gives this model many more examples to correspond to each animation control, as shown in Figure 1.

Memory and Time. Training time for the rig variant model takes 7 hours for characters with 6,000 vertices and 500 rig-control variables. The backlot model trains on all available data (which increases the number of training and validation examples by a factor of 4) and takes 20 hours to train on characters with similar complexity. Once trained, the inference time to approximate the character’s mesh for each pose is on average 5 ms for both models (all clocked on a 2.3 GHz Intel Xeon E5-2699).

### 4 CONCLUSIONS

We present a data-driven framework for approximating rig functions with a more durable, portable, and lightweight system. The learned model has augmented our pipeline, enabling artists to easily reuse characters, incorporate characters in projects on external platforms (e.g. VR), and broaden the expressivity of crowds and other characters requiring simplified rig variants.

### REFERENCES