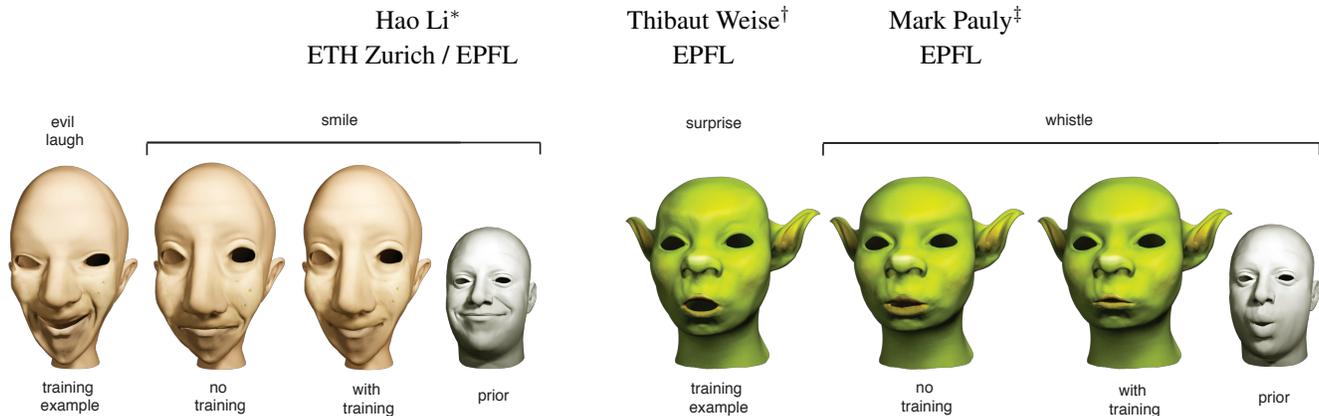


# Example-Based Facial Rigging



**Figure 1:** Example-based facial rigging allows transferring expressions from a generic prior to create a blendshape model of a virtual character. This blendshape model can be successively fine-tuned toward the specific geometry and motion characteristics of the character by providing more training data in the form of additional expression poses.

## Abstract

We introduce a method for generating facial blendshape rigs from a set of example poses of a CG character. Our system transfers controller semantics and expression dynamics from a generic template to the target blendshape model, while solving for an optimal reproduction of the training poses. This enables a scalable design process, where the user can iteratively add more training poses to refine the blendshape expression space. However, plausible animations can be obtained even with a single training pose. We show how formulating the optimization in gradient space yields superior results as compared to a direct optimization on blendshape vertices. We provide examples for both hand-crafted characters and 3D scans of a real actor and demonstrate the performance of our system in the context of markerless art-directable facial tracking.

**Keywords:** facial animation, rigging, blendshape animation

## 1 Introduction

Building a facial model parameterization or *rig* is an essential element for animating CG characters in feature films and computer games. Due to their intuitive controls, facial rigs based on blendshape models are particularly popular among artists for creating realistic looking facial animations. Since they are art-directable, blendshape parameterizations are often used for retargeting detailed recordings of facial performances to digital faces that differ strongly from the source model. For instance, an artist has maximum control over the appearance of wrinkles and folds for a particular facial

pose, as opposed to pure dynamic muscle rigs. However, hundreds of separately sculpted shapes are typically needed to achieve realism. The ability to both efficiently generate a complete customized facial rig and automatically adjust blendshapes to match the specific look of the actor’s expressions (while retaining the controller semantics) is thus an important asset for the artist.

This paper introduces a framework that automatically creates optimal blendshapes from a set of example poses of a digital face model. A predefined blendshape rig of a generic face is used as a prior to determine the semantics of each blendshape expression that we solve for. While in a traditional setting a precise pose needs to be provided for every blendshape, we only require a reduced set of example poses with a rough initial guess on the blending weights.

We propose an interleaved optimization that refines the blending weights and solves for the optimal blendshapes in two alternating steps. We regularize the optimization with meaningful blendshape expressions transferred from a generic face to accurately capture both the example poses and the semantics of the individual blendshapes. Expression transfer is achieved by mapping the deformation gradients between the neutral and blendshape pose triangles of the generic face to the target mesh triangles. Since we optimize for all blendshapes simultaneously, weighting local variations between the blendshapes in neutral and deformed pose is crucial in the regularization to prevent semantically incorrect blendshapes. We introduce an optimization that operates directly in gradient space in order to efficiently solve for blendshapes with semantics that corresponds to those of a generic facial rig prior.

We illustrate the versatility of our system with applications to art-directable rigging for sculpted virtual characters and automated rigging from 3D scans of a real actor. A key aspect of our approach is that the blendshape reconstruction can be edited and adapted iteratively by either generating additional training expressions or adapting the blending weights of the example poses. This provides full control over the resulting blendshape model and facilitates easy integration into existing workflows such as facial tracking. Without our technique, an artist would have to adapt each blendshape to match all desired input examples.

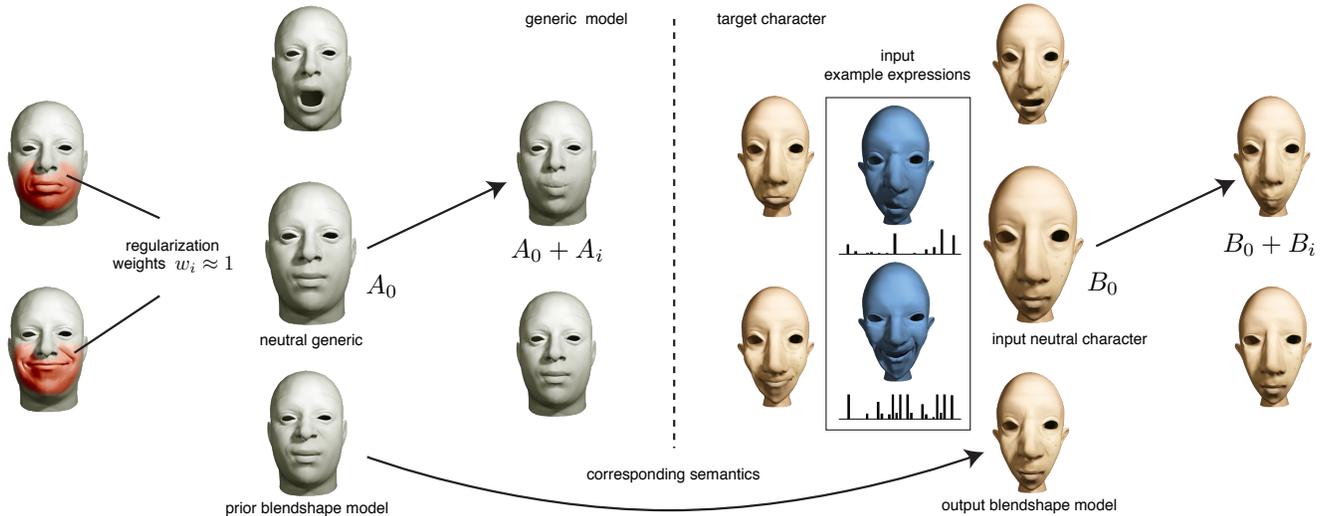
## 2 Related Work

A large variety of different methods for facial rigging have been proposed in the past. Some are based on skeletons and joints

\*E-mail:hao@inf.ethz.ch

†E-mail:thibaut.weise@epfl.ch

‡E-mail:mark.pauly@epfl.ch



**Figure 2:** Conceptual overview of our method. The generic template, illustrated with a subset of the blendshapes (left), serves as a geometry and motion prior for an actor-specific blendshape model (right). The optimization solves for the target blendshapes such that a set of example expressions are best reproduced while maintaining the semantic correspondence between template and target models. We ensure semantically correct transfer of expressions using additional per vertex regularization weights in our optimization (shown in red).

[Magnenat-Thalmann et al. 1988], physically-based muscle models [Waters 1987], linear blendshapes [Bergeron and Lachapelle 1985], or combinations thereof. Skeleton-based rigs are most often employed for full-body animation due to their intuitive control for articulated motion. While skeletons are often used for face animation of cartoon characters, this approach is less suited to produce detailed facial expressions that exhibit wrinkles and folds. Automatic rigging using skeleton-embedding was proposed by [Baran and Popović 2007], but with the focus on full-body animation.

Physically-based muscle models are well suited for creating realistic expression dynamics and secondary motions [Sifakis et al. 2005]. However, artistic control can be difficult to achieve. [Terzopoulos and Waters 1990] proposed a semi-automatic rigging of a muscle-based model to image data. Similarly, [Kähler et al. 2001] fit a complex anatomical model to partial 3D scan data. [Orvalho et al. 2008] introduced a general rigging method by transferring a generic facial rig to 3D input scans or hand-crafted models.

Linear blendshape models [Bergeron and Lachapelle 1985] provide a good compromise between realism and control. However, hundreds of blendshapes are usually necessary to capture realistic facial expression and are often used to mimic the effect of facial muscle groups as described by Ekman’s *Facial Action Coding System* [1978]. In particular, *FACS* decomposes facial behavior into 46 basic poses which are often complemented with a multitude of combined expressions and visemes. Building such a linear facial rig for highly realistic animation was recently demonstrated by [Alexander et al. 2009], though each blendshape was still hand-crafted by animators. [Pighin et al. 1998] build a rig automatically from photographs, and, similarly, [Zhang et al. 2004; Weise et al. 2009] from 3D scan data where all facial expressions are required as input. Automatic creation of facial models using (multi-) linear PCA models was proposed in [Blanz and Vetter 1999; Blanz et al. 2003; Vlasic et al. 2005], though the resulting linear blendshapes are not necessarily meaningful for facial animation control. To circumvent this problem, we propose to use a predefined generic blendshape rig as a semantic prior. This has the benefit that only a subset of expressions is sufficient to build a complete model, and moreover, the resulting blendshapes match the controller semantics of the prior.

Linear blendshape models are especially suited for retargeting [Chuang 2004]. An overview of current methods is given by [Pighin and Lewis 2006]. Choe and Ko [2005] proposed optimizing a generic predefined blendshape rig to fit sparse motion capture data of an actor. Liu et al. [2008] extended this method using expression cloning [Noh and Neumann 2001] as a prior to handle under-constrained cases where less training data is available than the number of blendshapes. In this work, we focus on building a blendshape model that has the same semantics as the input model. This is achieved by formulating the optimization problem in deformation gradient space [Sumner and Popović 2004].

### 3 Optimization

Our goal is to produce a full set of blendshapes from a user-provided handcrafted character or scanned 3D model in neutral expression. The user can specify an arbitrary number of additional expressions to refine the model toward the specific geometry and motion characteristics of the actor. The algorithm then determines the optimal blendshapes that best reproduce the input examples, while preserving the controller semantics by matching the deformation gradients of a generic blendshape model rig (c.f. Figure 2). For complex input expressions, such as an angry face, it can be difficult to determine the blending weights for a given example pose and those values can vary substantially for different characters. Thus, in addition to computing the optimal blendshapes, we also solve for blending weights given a rough initial guess provided by the user.

We assume a generic blendshape model is given as a set of meshes  $\mathcal{A} = \{A_0, \dots, A_n\}$ , where  $A_0$  is the rest pose and the  $A_i, i > 0$  are additive displacements. Expressions can be generated as  $T_j = A_0 + \sum_{i=1}^n \alpha_{ij} A_i$ , where  $\alpha_{ij}$  are the blending weights of pose  $T_j$ .

Our method is general in the sense that we can process input from various sources. In the case of 3D scans, we align the generic rest pose  $A_0$  to the input shapes using the non-rigid registration method of [Li et al. 2009]. This produces a set  $\mathcal{S} = \{S_1, \dots, S_m\}$  of complete meshes with connectivity of the prior model and shape of the respective scan. For hand-crafted models, we either perform

the same registration operation or directly sculpt from the rest pose. We call these meshes *training poses*.

Our goal is to compute a new blendshape model  $\mathcal{B} = \{B_0, \dots, B_n\}$  that matches the geometry and motion of the actor. Thus we need to find blendshapes  $B_i$  and corresponding weights  $\alpha_{ij}$  such that the training poses are faithfully reproduced, i.e.,  $S_j \approx B_0 + \sum_{i=1}^n \alpha_{ij} B_i$ . To solve this bi-linear problem, we need to address two main challenges: Firstly, how can we compute the target blendshapes  $B_i$ , if only very few training poses are given, i.e., when the problem is under-constrained ( $m < n$ )? And secondly, how can we achieve the right controller semantics, i.e., ensure that similar weight settings lead to semantically similar expressions for both the template and the target blendshape models?

Our solution proceeds iteratively by alternating between two steps: step A keeps the blending weights  $\alpha_{ij}$  fixed and optimizes for the blendshapes, while step B keeps blendshapes fixed and solves for the optimal weights. As an important means of control, the user establishes a semantic correspondence between each training pose  $S_j$  and the generic template. For this purpose, the user selects appropriate blending weights on the template to model a pose  $T_j$  that roughly corresponds to the training pose  $S_j$ . This yields (approximate) weights  $\alpha_{ij}^*$  that provide initial values for step A of the optimization and semantic constraints for step B. We show in Section 4 that the  $\alpha_{ij}^*$  can be intuitively determined by the user and do not need to be very accurate. Typically, the blending weights of only a few but sufficiently expressive poses (usually not more than 4) need to be manually activated in the beginning for each training pose.

**A: Optimizing Blendshapes.** To be able to reconstruct target blendshape models from few training poses, we incorporate additional constraints derived from the expression space of the template. The idea is to preserve the motion characteristics of the template by mapping the relative change between rest pose and blendshapes from the template to the target. This relative change can be encoded effectively using deformation gradients [Sumner and Popović 2004]. For a triangle  $t$  with vertices  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ , we define a local frame as the  $3 \times 3$  matrix  $\mathbf{M}_t = [\mathbf{v}_3 - \mathbf{v}_1, \mathbf{v}_2 - \mathbf{v}_1, \mathbf{n}]$ , where  $\mathbf{n} = (\mathbf{v}_3 - \mathbf{v}_1) \times (\mathbf{v}_2 - \mathbf{v}_1)$  is the triangle normal vector. The deformation gradient that maps a source triangle  $s$  to a target triangle  $t$  is then given as  $\mathbf{G}_{s \rightarrow t} = \mathbf{M}_t \cdot \mathbf{M}_s^{-1}$ .

One of the key insights of this paper is that we can formulate the blendshape optimization in gradient space and reconstruct the final blendshapes from the local triangle frames. As we show in Section 4, this leads to significant improvements compared to a direct optimization of blendshape vertex positions. Since the following optimization is performed independently for each triangle, we omit triangle indices and write e.g.,  $\mathbf{M}_i^B$  for the (unknown) frame of each triangle in blendshape  $B_i$ .

For the actor’s rest pose  $B_0$  and each of the training poses  $S_j$ , we can compute the frames  $\mathbf{M}_0^B$  and  $\mathbf{M}_j^S$ , respectively. To faithfully reproduce the training poses, we define the fitting energy

$$E_{\text{fit}} = \|\mathbf{M}_j^S - (\mathbf{M}_0^B + \sum_{i=1}^n \alpha_{ij} \mathbf{M}_i^B)\|_F^2$$

which measures the deviation of the training poses  $S_j$  in the space of triangle frames from the best possible reconstruction in the unknown blendshape model. To account for insufficient training data we postulate that the deformation gradients of actor blendshapes  $B_i$  and template blendshapes  $A_i$  should be similar. Since the  $A_i$  and  $B_i$  for  $i > 0$  are additive displacements, this means that  $\mathbf{G}_{B_0 \rightarrow B_0 + B_i} \approx \mathbf{G}_{A_0 \rightarrow A_0 + A_i}$ . We can write  $\mathbf{G}_{B_0 \rightarrow B_0 + B_i} =$

$(\mathbf{M}_0^B + \mathbf{M}_i^B)(\mathbf{M}_0^B)^{-1}$  and define the regularization energy as

$$E_{\text{reg}} = \sum_{i=1}^n w_i \|\mathbf{M}_i^B - \mathbf{M}_i^{A*}\|_F^2$$

where the  $\mathbf{M}_i^{A*} := \mathbf{G}_{A_0 \rightarrow A_0 + A_i} \cdot \mathbf{M}_0^B - \mathbf{M}_0^B$  can be computed from the template blendshapes and the target rest pose. We incorporate additional regularization weights  $w_i$  as an essential means for maintaining the semantics of the generic prior. If a triangle of the template blendshape moves a little or not at all, we want to ensure that the same holds for the reconstructed target blendshape. However, if the template blendshape exhibits a strong motion, we want to allow the target deformation gradients to deviate more from the template prior to account for geometric and motion differences of the two characters. Our experiments showed that evaluating the regularization weights as  $w_i = ((1 + \|\mathbf{M}_i^A\|_F)/(\kappa + \|\mathbf{M}_i^A\|_F))^\theta$  with  $\kappa = 0.1$  and  $\theta \geq 1$  adequately guides the optimization toward these semantics. We use  $\theta = 2$  for all our results, yet similar results are obtained with other values. Note that constraining the vertices using the regularization weights does not limit the range of expressions for the target character, since other complementary blendshapes will be activated by the optimization to achieve a specific expression.

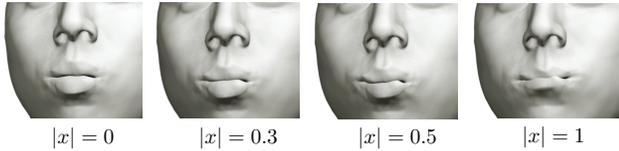
We combine both energy terms to yield the global energy  $E_A = E_{\text{fit}} + \beta E_{\text{reg}}$ , where  $\beta$  is a parameter that allows balancing fitting and regularization. Due to the cross-product in the definition of the normal vector that constitutes the third column in the matrix  $\mathbf{M}_i^B$ , the energy  $E_A$  is non-linear in the vertex positions. Fortunately, as shown in [Botsch et al. 2006], we can safely *ignore* the normal component for the reconstruction and only solve for the linear components, i.e., the first two columns of the matrices  $\mathbf{M}_i^B$ . Thus minimizing  $E_A$  amounts to simply solving a linear system. Given the  $\mathbf{M}_i^B$ , we can reconstruct the vertex positions of each blendshape using a least-squares optimization as described in [Sumner and Popović 2004]. To prevent undesirable drifting, we constrain all vertices that are stationary in a template blendshape to remain fixed in the corresponding target blendshapes as well.

**B: Optimizing Weights.** Given the computed set  $\mathcal{B}$  of blendshapes, we can solve for the optimal weights  $\alpha_{ij}$  to reconstruct the training poses  $S_j$  using least-squares fitting. We include the user-specified weights  $\alpha_{ij}^*$  as soft constraints and define the energy  $E_B$  as a function of the unknowns  $\alpha_{ij}$  as

$$E_B = \sum_{k=1}^N \|\mathbf{v}_k^{S_j} - (\mathbf{v}_k^{B_0} + \sum_{i=1}^n \alpha_{ij} \mathbf{v}_k^{B_i})\|_2^2 + \gamma \sum_{i=1}^n (\alpha_{ij} - \alpha_{ij}^*)^2$$

where  $\mathbf{v}_k^{S_j}$  and  $\mathbf{v}_k^{B_i}$  are the vertices of the training pose  $S_j$  resp. the blendshapes  $B_i$ , and  $N$  is the total number of vertices. The parameter  $\gamma$  balances fitting and regularization. Note that even for  $\gamma = 0$ , the resulting weights are likely to match the semantics of the template controllers, since the regularization energy  $E_{\text{reg}}$  of the blendshape optimization couples corresponding template and target blendshapes. However, the weights  $\alpha_{ij}^*$  allow the user to adapt the controller semantics and thus control the resulting expression space.

Since blendshape weights are typically constrained between zero and one, we use quadratic programming to solve the constrained system. Moreover, when manipulating or fine-tuning blendshape sliders, artists often impose *activation* constraints to disallow pairs of blendshapes to simultaneously contribute to a pose. For instance, a mouth which lies exactly on the reflective symmetry plane of the face is often constrained to not squeeze to the left and to the right at the same time. This can be formulated as non-linear constraints of the form  $\alpha_{ij} \alpha_{kj} = 0$  for two mutually exclusive blendshapes  $B_i$



**Figure 3:** When the initial blending weights are perturbed by  $\pm x$ , the fitting quality using the optimized blendshape model start to decrease when  $|x| > 0.3$ .

and  $B_k$ . We replace these non-linear constraints by corresponding non-linear penalty terms and apply a second optimization to update the blending weights  $\alpha_{ij}$  using a solver for non-linear least-squares problems with linear constraints [Coleman and Li 1996].

## 4 Evaluation

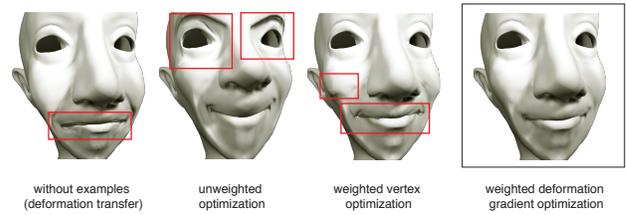
To evaluate our method, we created two sets of training poses using geometric modeling tools and two sets using a 3D scanner [Weise et al. 2007]. The generic template model  $\mathcal{A}$  is taken from the book *Stop Staring* [Osipa 2007] and consists of 11K vertices, which is considered high for artists to start sculpting from. While our model is triangulated from a subdivision quad mesh, we still retain a one-to-one correspondence between our mesh and the initial model. The facial rig includes 29 different blendshapes with 6 pairs of modes that must not be activated simultaneously. For 14 training poses, our unoptimized implementation requires 45 seconds per iteration. Approximately equal computation time is spent on blendshape optimization, reconstruction, and alpha optimization respectively.

In a typical setting, the artist mainly controls the parameters  $\beta$  and  $\gamma$  to adjust the output blendshapes. When  $\beta \gg 1$  the resulting blendshape model is close to the results achieved via pure deformation transfer. In this case, even when  $\gamma = 0$ , no visible artifacts were observed in any of our examples. When  $\beta$  is closer to 0.1, the resulting blendshape is able to accurately capture the input examples, but its quality can be sensitive for  $\beta \ll 0.1$ . In particular, when  $\gamma = 0$  some artifacts can appear for some blendshapes, but these are prevented when  $\gamma$  is large enough. For all our results, we simply apply 10 iterations of alternating blendshape and weight optimizations, with  $\beta = 0.5$  and  $\gamma = 1000$  for the first iteration. The weights are gradually decreased to  $\beta = 0.1$  and  $\gamma = 100$  in the last iteration. Weight scheduling ensures robustness to local minima while enabling detailed adaption to the input after optimization.

Our optimization is robust to variations in the initial selection of the blending weights  $\alpha_{ij}^*$ . We perturbed the user-provided initial values by randomly adding a value between  $-x$  and  $x$ . Up to  $|x| = 0.3$ , we did not obtain any noticeable differences in the reconstructed blendshapes for all examples. Figure 3 shows the impact of increasing variations of random  $\alpha_{ij}^*$  when fitting the kiss expression with the optimized blendshape model.

Figure 4 demonstrates the importance of the weights  $w_i$  in the regularization energy  $E_{\text{reg}}$ . Without weighting, the optimization creates a combination of semantically separate blendshapes, i.e., mixes undesirable eyebrow motion into the smile blendshape. However, using a weighted optimization when solving for the vertex positions directly leads to artifacts, as each vertex is considered independently. These artifacts are absent in our method as the optimization of the deformation gradient is followed by a subsequent blendshape reconstruction step.

Figure 6 illustrates the influence of the parameter  $\beta$  on the blendshape energy  $E_A$ . While the fitting improves with decreasing weight, at around  $\beta = 0.05$  over-fitting occurs that leads to artifacts in the reconstructed blendshapes. We found that  $\beta = 0.1$  is



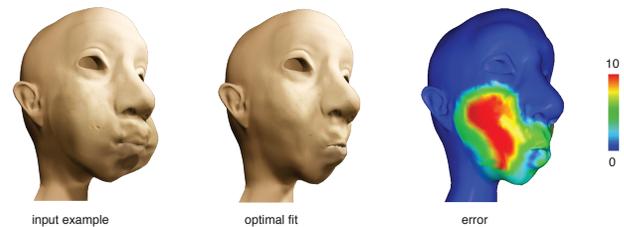
**Figure 4:** Different blendshape optimization methods. Without training data, the reconstructed blendshapes correspond to pure deformation transfer. Without weighting, undesirable mixing of blendshape modes occurs, noticeable in the motion of the eye brows in the smile blendshape. The optimization formulated in vertex space leads to visible artifacts, while our approach avoids these errors and achieves the desired semantic separation of blendshapes.

a good compromise between accuracy and robustness for both 3D scan data and hand-crafted models.

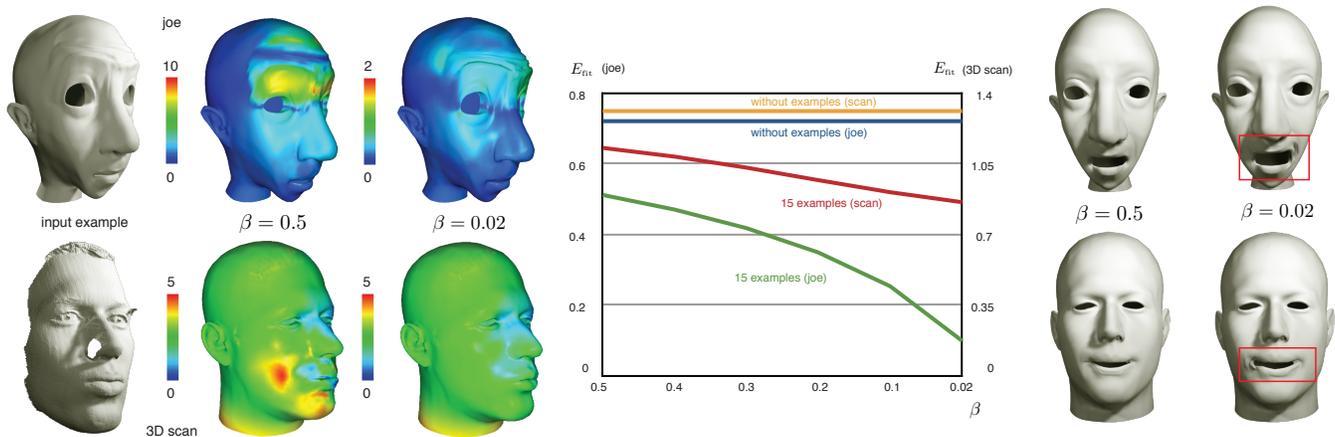
Figure 7 provides qualitative results for three complex expressions of two cartoon characters (see also Figure 1) and one model derived from 3D scans. Without training examples our method effectively performs deformation transfer on the blendshapes, which results in expressions that mimic the poses of the generic template. With more training examples, the expressions adapt closer to the characteristics of the target model while still conforming to the same controller semantics. For instance, our method automatically includes the wrinkles of the *joe* model that appear in the training examples. The automatically reconstructed blendshape rigs may best be appreciated in the accompanying video where an animation sequence from the template rig is applied to each rigged face.

We also tested our algorithm in the context of markerless facial tracking (Figure 8) by using a generic model that contains all 46 FACS poses and 28 supplemental expressions [Stahlberg 2010]. We used the system described in [Weise et al. 2009] and replaced the PCA model with our optimized rig. Now our approach enables artists to intuitively tweak blending weights after tracking. Also, our technique demonstrates that very few training poses (17) are sufficient to accurately express a dense facial expression space. Without examples, the blendshapes are not expressive enough.

**Limitations.** Our method assumes training examples to semantically correspond to valid blendshape combinations of the generic rig. The *blow* expression in Figure 5 cannot be represented by the prior model and therefore the optimization fails to fit this expression. However, we can easily detect the case when poses are missing in the generic model by verifying if  $E_{\text{fit}}$  exceeds a certain threshold. Semantically differing expressions would thus need to be added as additional blendshapes. Currently, the algorithm is not fast



**Figure 5:** Not all training poses can be expressed by the reconstructed blendshape model for the given semantics. In such cases, additional blendshapes are required in the prior to introduce more degrees of freedom.



**Figure 6:** Influence of the parameter  $\beta$  that balances fitting term  $E_{fit}$  (in mm) and regularization term  $E_{reg}$ . Effectively,  $\beta$  controls the relation between deformation transfer from the generic template and example poses provided by the user. Over-fitting can occur when  $\beta$  is too small.

enough for interactive rates. The algorithmic complexity scales linearly with the number of training examples and mesh vertices, but the non-linear solver has cubic complexity in the number of blendshapes. More sophisticated solvers and an optimized GPU implementation may allow artists to get direct feedback on the facial rig while sculpting the example expressions. When only using a few example expressions, only those blendshapes are being optimized that influence these expressions. Every other created blendshape will look like deformation transfer, which creates plausible deformations, but may not catch the exact expressions of the character.

## 5 Conclusion and Future Work

Example-based facial blendshape rigging is intended to increase the productivity of professional artists and allow even inexperienced users to quickly generate actor-specific blendshape models. Instead of solely relying on sculpting and fine-tuning every single blendshape to match the intended expressions of an animation, our method only requires a small subset of these expression. This enables a scalable design process and effective reuse of existing rigs.

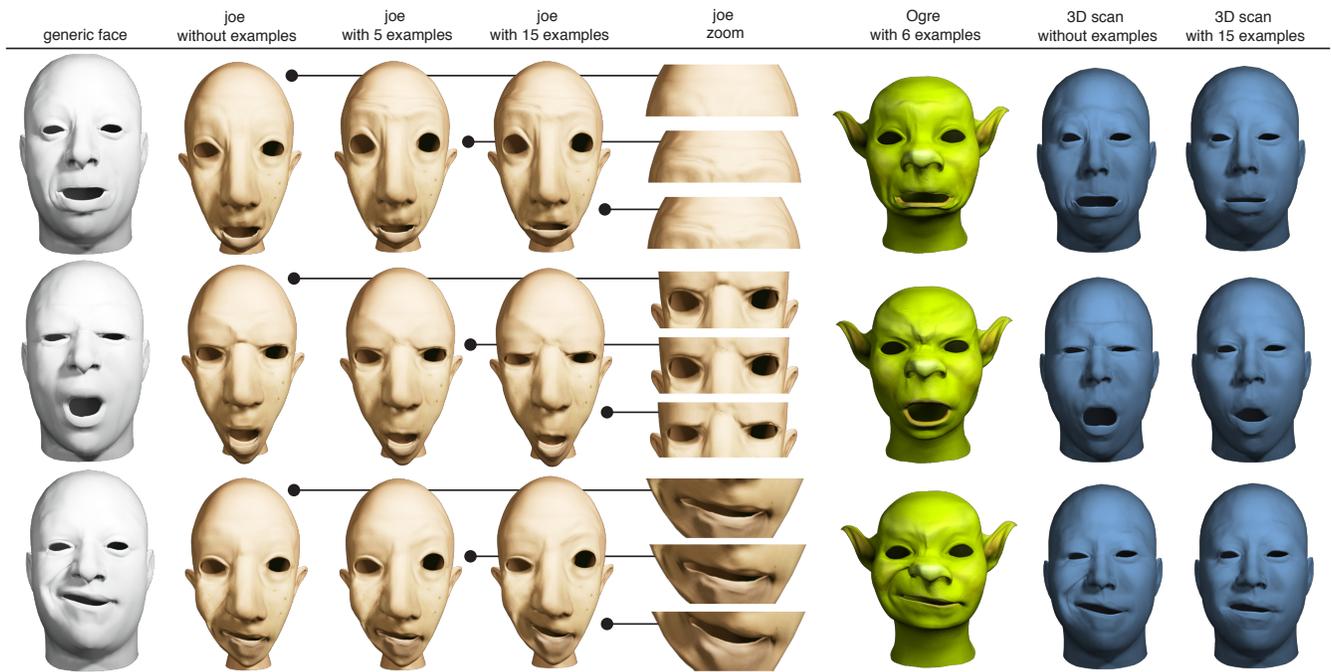
Key contribution of this paper is the formulation of the blendshape optimization in gradient space. In combination with appropriate weighting schemes, we obtain a consistent integration of expression transfer and reconstruction of example poses to yield high-quality customized blendshape rigs with pre-defined controller semantics.

In the future, we want to investigate how combining multiple template models improves expression transfer by providing a more general prior that can more closely adapt to the target model. In order to handle input examples that cannot be expressed by a given generic template model due to insufficient blendshapes, we would also like to explore methods that suggest supplemental expressions of the generic model that can maximize orthogonality.

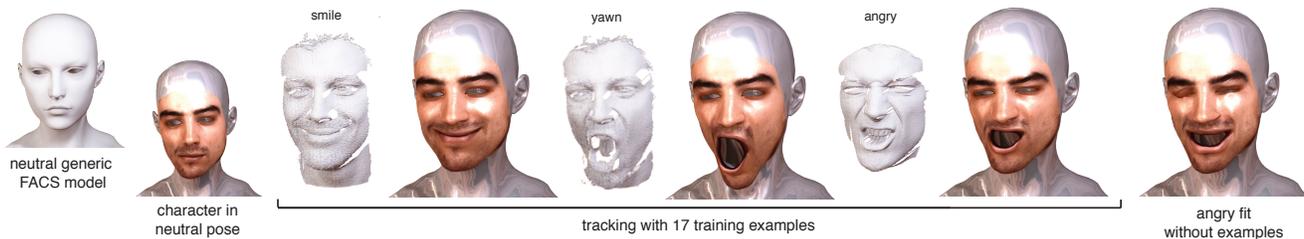
**Acknowledgements.** We thank Polina Tolkacheva and Tricia Barrett for sculpting the CG characters and evaluating our rigging technology. Special thanks go to Jason Osipa for allowing us to use the generic head model, Severin Stalder for his facial scans, and Mario Botsch, Carsten Stoll, and Pieter Peers for fruitful discussions. We are grateful to Kiran Bhat and Chen Shen for industry feedback, Volker Helzle and Clemens Sielaff for providing the generic model with FACS poses, and Krystle de Mesa for proof-reading the paper. This research is supported by Swiss National Science Foundation grant 20001-112122.

## References

- ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, M., AND DEBEVEC, P. 2009. The digital emily project: photoreal facial modeling and animation. In *SIGGRAPH '09 Courses*.
- BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.* 26, 3, 72.
- BERGERON, P., AND LACHAPPELLE, P. 1985. Controlling facial expressions and body movements in the computer generated animated short 'Tony de Peltrie'. In *SIGGRAPH '85 Tutorial Notes, Advanced Computer Animation Course*.
- BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH '99*.
- BLANZ, V., BASSO, C., POGGIO, T., AND VETTER, T. 2003. Reanimating faces in images and video. In *EUROGRAPHICS '03*.
- BOTSCH, M., SUMNER, R., PAULY, M., AND GROSS, M. 2006. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling, Visualization 2006*, 357–364.
- CHOE, B., AND KO, H.-S. 2005. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *SIGGRAPH '05 Courses*.
- CHUANG, E. 2004. *Analysis, Synthesis, and Retargeting of Facial Expressions*. PhD thesis, Stanford University.
- COLEMAN, T. F., AND LI, Y. 1996. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* 6, 2, 418–445.
- EKMANN, P., AND FRIESEN, W. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press.
- KÄHLER, K., HABER, J., AND PETER SEIDEL, H. 2001. Geometry-based muscle modeling for facial animation. In *Proc. Graphics Interface 2001*.
- LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2009)* 28, 5.
- LIU, X., MAO, T., XIA, S., YU, Y., AND WANG, Z. 2008. Facial animation by optimized blendshapes from motion capture data. *Comput. Animat. Virtual Worlds* 19, 3-4, 235–245.



**Figure 7:** Without training examples, our method simply transfers the expression dynamics of the generic face toward the actor. With more training expressions, the reconstructed blendshape model adapts toward the geometry and motion characteristics of the actor.



**Figure 8:** Art-directable facial tracking. Our optimized blendshapes can be directly used for facial tracking by solving for optimal blending weights for each frame of a 3D scan sequence. Compared to PCA approaches, our method allows intuitive control over the blending weights.

MAGENAT-THALMANN, N., LAPERRIÈRE, R., AND THALMANN, D. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface '88*, 26–33.

NOH, J.-Y., AND NEUMANN, U. 2001. Expression cloning. In *Proc. SIGGRAPH '01*.

ORVALHO, V. C. T., ZACUR, E., AND SUSIN, A. 2008. Transferring the rig and animations from a character to different face models. *Comput. Graph. Forum* 27, 8, 1997–2012.

OSIPA, J. 2007. *Stop Staring: Facial Modeling and Animation Done Right*. Sybex, Second Edition.

PIGHIN, F., AND LEWIS, J. P. 2006. Facial motion retargeting. In *SIGGRAPH '06 Courses*.

PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI, R., AND SALESIN, D. H. 1998. Synthesizing realistic facial expressions from photographs. In *Proc. SIGGRAPH '98*.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.* 24, 3, 417–425.

STAHLBERG, S., 2010. Nikita real-time character. Filmakademie Baden-Wuerttemberg / Institute of Animation's R&D Labs.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2004)* 23, 3.

TERZOPOULOS, D., AND WATERS, K. 1990. Physically-based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation* 1, 73–80.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Trans. Graph.* 24.

WATERS, K. 1987. A muscle model for animation three-dimensional facial expression. In *Proc. SIGGRAPH '87*.

WEISE, T., LEIBE, B., AND GOOL, L. V. 2007. Fast 3d scanning with automatic motion compensation. In *Proc. CVPR'07*.

WEISE, T., LI, H., GOOL, L. V., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *Proc. SCA'09*.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conf. on Comp. Graphics*.