Performance Driven Facial Animation
Course Notes Example:

Motion Retargeting

J.P. Lewis
Stanford University

Frédéric Pighin
Industrial Light + Magic

Introduction

When done correctly, a digitally recorded facial performance is an accurate measurement of the performer's motions. As such it reflects all the idiosyncrasies of the performer. However, often the digital character that needs to be animated is not a digital replica of the performer. In this case, the decision to use performance capture might be motivated by cost issues, the desire to use a favorite actor regardless of the intended character, or the desire to portray an older, younger, or otherwise altered version of the actor. The many incarnations of Tom Hanks in *Polar Express* illustrate several of these scenarios.

In this scenario, the recorded (source) performance has to be adapted to the target character. In this section of the course, we examine different techniques for retargeting or cross-mapping a recorded facial performance onto a digital face. We have grouped these techniques in several categories mostly as a function of whether they use a blendshape system for the source and/or the target face.

Retargeting techniques

Blendshape animation is one of the most widespread facial animation techniques. Thus, it is not surprising that many techniques consider the retargeting/cross-mapping problem in the context of a blendshape system. If we have a set of blendshapes for the performer and one for the target character that correspond to the same expressions, then once the performance is mapped onto the source blendshapes, it can be mapped onto the target character by simply reusing the same weights [10].

Blendshape mapping. It is also possible to accomplish blendshape mapping when the source and target models have blendshapes with differing functions. The slides accompanying this session point out that, provided a skilled user can produce N "corresponding" poses of the source and target models, a matrix that converts the weights from source to target representations can be found with a linear system solve. This assumes of course that linearity is adequate – which is also an assumption of the underlying blendshape representation. More importantly, however, it assumes that animation of the source model is obtainable. In the case of performance driven animation however, the source model will *not* exactly represent the performance unless the model itself is obtained directly from that performance (e.g. by principal component analysis of dense capture). Thus, transferring the performance onto the source model is at issue (and, to the extent that this mapping can be solved, why not omit the source and map the performance directly to the target?).

Choe and Ko [3] invented a very effective technique for transferring a recorded performance onto a digital character. In this framework, the target face is animated as a set of blendshapes. The goal of the algorithm

is to find for each frame of the performance the corresponding blending weights. To do this, they first create a corresponding set of blend shapes (or actuation basis) for the source face. Once this is done, the blending weights can simply be transfered from the source blendshapes to the target blendshapes. The main advantage of their approach is that it refines the source blendshapes as a function of the recorded performance. In this sense, it is tolerant to approximate modeling.

This technique starts by manually assigning a location (corresponding point) on the source model for each recorded marker. From these correspondences, a transformation (rigid transformation and scaling) is computed that maps the performance coordinate system onto the model coordinate system. This transformation takes care of the difference in orientation and scale between the performance and source models. It is estimated on the first frame and applied to all the frames of the performance. The following procedure is then applied for each frame. If a frame in the animation is considered as a vector, it can be written as a linear combination of the corresponding points in the blendshapes where the weights are the blending weights. This provides a set of linear equations where the blending weights are the unknowns. Augmented with a convexity constraint (i.e. all weights have to be non-negative and sum up to one), this system can be solved using quadratic programming. Their approach assumes that the source blendshapes can exactly represent the performance, which is generally not true of manually sculpted blendshape models. To address this issue, a geometric correction is performed by solving the same system for the position of the corresponding points. These two steps (blend weight estimation and geometric correction) are iterated until convergence. Finally, the displacement of the corresponding points are propagated to the model vertices using radial basis functions [2].

This work is presented in a muscle actuation framework where each blendshape corresponds to the actuation of a muscle or muscle group. However, it should equally apply to sets of blendshapes constructed with different philosophies.

Direct mapping to target blendshapes. The previous technique requires a set of blendshapes for the source face. Other researchers have investigated direct mappings between the source motions and the target blendshapes. For instance Buck et al. [1] developed a system for mapping 2D facial motion onto cartoon drawings. The input motion is estimated from video by tracking a sparse set of features whose configuration provides a simplified facial expression.

Their system is build on top of a library of cartoon drawings that represent key poses for different facial areas of the target character (e.g. mouth, forehead). These key drawings are blended together, much like a set of blendshapes, to create an animation. Their mapping algorithm relies on associating each key drawing with a particular configuration of the tracked features. This association is then generalized using a scattered data interpolation algorithm. The interpolation is performed using a partition of the space of potential feature configuration (i.e. simplified facial expression). The partition they use is a 2D Delaunay triangulation. To map a frame of input motion, first the triangle that contains that frame is determined; second the barycentric coordinates within the triangles are computed; finally these coordinates are used as blending weights to compute the combination of key drawings. To provide a 2D parameterization of the input space, a Principal Component Analysis is performed on some test data. The two first principal components (maximum variance) determine the reduced dimensionality space.

Tim Hawkins et al. [8] use the same technique to animate facial reflectance fields with a higher dimensional space.

Chuang and Bregler [4] described another approach to mapping a performance directly to a differing target model. The source in their technique is video, but similar thinking could be applied in mapping from three-dimensional motion capture. Important feature points on the video frames are tracked using the Eigenpoints technique in which the weights of an eigenimage fit are applied in parallel to vectors of two dimensional points associated with each of the basis image [6]. The tracked feature points in a new image frame can (after a coarse affine registration) be approximated as a linear combination of basis feature vectors, and similar basis shapes can be sculpted for the target model.

With this background, they present two ideas that lead to a robust retargeting. First, they show how to choose a basis from among the source frames. After experimenting with several plausible approaches it was found that the best basis (least error in representing the source performance) resulted from taking the source frame point vectors that result in the smallest and largest projection on the leading eigenvectors of the source performance principal component model. Secondly, they point out that reusing the source weights on a target model does not work well when the basis is large and does not exactly represent the source performance. In this situation errors in representing a novel source frame can sometimes be reduced with large counterbalanced positive and negative weights, which results in a poor shape when the weights are reused on the target. Instead, they require the weights to be non-negative. This prevents the previously described situation because negative weights are not available, and results in demonstrably better retargeting even though the error in representing the source is somewhat higher.

Geometric mapping. Using a blend shape system is not the only way to drive a synthetic face through performance capture. For instance, good results can also be achieved using radial basis functions [9]. Noh and Neumann [12] propose a different approach, called "Expression Cloning", that does not rely on blend shapes. Their technique assumes that the source performance is given as an animated mesh (i.e. the topology and motion curves for every vertex). Their goal is to transfer the deformations of the source mesh onto the target mesh.

The first step of the algorithm is to find geometric correspondences between the source and target models. This is done by computing a sparse set of correspondences that are propagated to the rest of the mesh using scattered data interpolation (radial basis functions). The sparse correspondences are determined either manually or using some face-specific heuristics.

Once the two models are brought into dense correspondence the motion vectors (offsets from the initial or rest expression) can be transferred. This transfer is performed by assigning a local coordinate system to each vertex in the source and target models. These coordinates systems are determined by the normal of the mesh at that vertex. Transferring a motion vector can then be done by changing local coordinate systems. The motion can also be locally scaled by using the ratio of locally defined bounding boxes in the two models. An additional procedure takes care of the special case of the lip contact line and prevents any spurious interactions between the two lips.

Expression/Style learning. Wang et. al. [11] describe an ambitious machine-learning based system for retargeting. A data reduction manifold learning technique (local linear embedding, LLE) is first used to derive a mapping from animated expression geometry over time to a one dimensional manifold (curve) embedded in a low-dimensional (e.g. 3D) space. They then establish correspondences between curves for a given expression over different individuals (this includes different monotonic reparameterizations of cumulative length along the curve). Once the correspondences are established, the registered curves are averaged to produce a mean manifold for the particular expression. Evolution of the expression over time now corresponds to movement along this curve.

Next a mapping from the curve back to the facial geometry is constructed. First a mapping from points on the expression curve back to the actor's changing facial geometry is obtained using an approximating variant of radial basis scattered interpolation. This mapping conflates the different "styles" of facial expressions of different people. Lastly, using the bilinear decomposition approach introduced in graphics by Chuang et. al. [5], the changing expression geometry is factored into components of facial expression and individual identity (thus, for a particular facial expression there is a linear model of how various individuals effect that frozen expression, and for any individual the evolution of the expression over time is also a linear combination of models).

Although the system deals with each facial expression in isolation, it hints at future advances in deriving useful higher level models from data.

Unexplored issues

Expression vs. motion retargeting. The different techniques we have described treat the retargeting issue as a geometric problem where each frame from the recorded performance is deformed to match the target character. Unfortunately, this might not respect the dynamics of the target character. To go beyond a straight per frame retargeting requires an approach that takes timing into account. There are basically two ways this can be tackled: using a physical approach or a data-driven approach.

A physically-based animation system could provide physical constraints for the target face. The retargeting algorithm would have to satisfy two types of constraints: matching the source performance but also respecting the physics of the target face. By weighting these constraints an animator could control how much of the source performer versus how much of the target character appears in the final animation.

Any data-driven approach must be carefully designed to minimize the "curse of dimensionality" issues introduced by additional dimensions of timing and expression. One approach might involve building a (small) database of motions for the target character. The performer could then act these same motions to create a corresponding source database. Using machine learning or interpolation techniques these matching motions could provide a time-dependent mapping from the source motion space to the target motion space.

Facial puppeteering. The human face can express a wide gamut of emotions and expressions that can vary widely both in intensity and meaning. The issue of retargeting raises the more general issue of using the human face as an animation input device not only for animating digital faces but any expressive digital object (e.g. a pen character does not have a face). This immediately raises the issue of "mapping" the facial expressions of the performer onto meaningful poses of the target character. Dontcheva et al. [7] tackles this issue in the context of mapping body gesture onto articulated character animations.

References

[1] Ian Buck, Adam Finkelstein, Charles Jacobs, Allison Klein, David H. Salesin, Joshua Seims, Richard Szeliski, and Kentaro Toyama. Performance-driven hand-drawn animation. In *NPAR 2000 : First International Symposium on Non Photorealistic Animation and Rendering*, pages 101–108, June 2000.

[2] Martin D. Buhmann. *Radial Basis Functions : Theory and Implementations*. Cambridge University Press, 2003.

[3] Byoungwon Choe, Hanook Lee, and Hyeong-Seok Ko. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation*, 12(2):67–79, May 2001.

[4] E. Chuang and C. Bregler. Performance driven facial animation using blendshape interpolation. *CS-TR-2002-02, Department of Computer Science, Stanford University*, 2002.

[5] E. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proceedings of Pacific Graphics*, 2002.

[6] M. Covell and C. Bregler. Eigen-points. In *Proc. IEEE International Conference on Image Processing*, pages vol 3 p 471–474, 1996. Lausanne, Switzerland, Sept 16-19 1996.

[7] Mira Dontcheva, Gary Yngve, and Zoran Popović. Layered acting for character animation. *ACM Transactions on Graphics*, 22(3):409–416, July 2003.

[8] Tim Hawkins, Andreas Wenger, Chris Tchou, Andrew Gardner, Fredrik Göransson, and Paul Debevec. Animatable facial reflectance fields. In *Rendering Techniques 2004: 15th Eurographics Workshop on Rendering*, pages 309–320, June 2004.

[9] J. Noh, D. Fidaleo, and U. Neumann. Animated deformations with radial basis functions. In *ACM Symposium on Virtual Realisty Software and Technology*, pages 166–174, 2000.

[10] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D.H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH 98 Conference Proceedings*, pages 75–84. ACM SIGGRAPH, July 1998.

[11] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning, and transfer of dynamic 3-d facial expressions. In *Eurographics*, 2004.

[12] Jun yong Noh and Ulrich Neumann. Expression cloning. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 277–288, August 2001.