Practice and Theory of Blendshape Facial Models

J.P. Lewis[1], Ken Anjyo[2], Taehyun Rhee[1], Mengjie Zhang[1], Fred Pighin[3], Zhigang Deng[4]

[1]Victoria University, Wellington, New Zealand
[2]OLM Digital
[3]Google, Inc.
[4]University of Houston

Abstract

"Blendshapes", a simple linear model of facial expression, is the prevalent approach to realistic facial animation. It has driven animated characters in Hollywood films, and is a standard feature of commercial animation packages. The blendshape approach originated in industry, and became a subject of academic research relatively recently. This report describes the published state of the art in this area, covering both literature from the graphics research community, and developments published in industry forums. We show that, despite the simplicity of the blendshape approach, there remain open problems associated with this fundamental technique.

Introduction

The face has always held a particular interest for the computer graphics community: its complexity is a constant challenge to our increasing ability to model, render, and animate lifelike synthetic objects. A variety of approaches to facial animation have been pursued, including:

- physically-based models, which approximate the mechanical properties of the face such as skin layers, muscles, fatty tissues, bones, etc. [SNF05],
- parametric models [Par74, Par91], in which custom deformation algorithms defined specifically for the face are implemented,
- approaches using proprietary deformers of commercial packages, such as "cluster deformers" [Tic09],
- generic, low-level meshes driven by dense motion capture [EYE, GGW*98, BPL*03, Mov09, BHPS10],
- principal component analysis (PCA) models obtained from scans or motion capture [ZSCS04],
- approaches based on spatial interpolation [BBA*07] or interpolation in an abstract "pose" or expression space [LCF00, BLB*08, LH09, RHKK11],
- "blendshape" models, which are referred to with several other names (refer to the Terminology section), and
- hybrid approaches [KMML10].

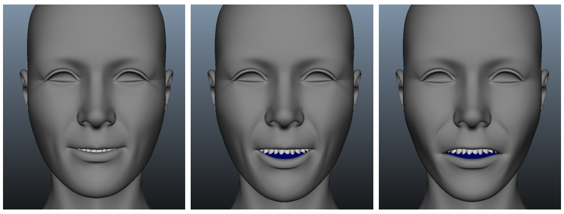


Figure 1: *Blendshapes are an approximate semantic parameterization of facial expression. From left to right, a half smile, a smile, and a (non-smiling) open-mouth expression. While the smile and open-mouth expressions are most similar in terms of geometric distance, the smile is closer to the half-smile in parameter distance (distance=0.36) than it is to the open-mouth expression (distance=1.34). Please enlarge to see details.*

See [OBP*12, DN07, PW08, NN99] for further overview of facial animation approaches.

Among these choices, blendshapes remain popular due to their combination of simplicity, expressiveness, and interpretability. Blendshape facial animation is the predominant choice for realistic humanoid characters in the movie industry. The approach has been used for lead characters in movies such as *The Curious Case of Benjamin Button* [Flu11], *King Kong* [SG06], *The Lord of the Rings* [Sin03], *Final Fantasy: The Sprits Within*, and *Stuart Little*. Even

© The Eurographics Association 2014.

DOI: 10.2312/egst.20141042

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

when more sophisticated approaches to facial modeling are used, blendshapes are sometimes employed as a base layer over which nonlinear or physically based deformations are layered.

A blendshape model generates a facial pose as a linear combination of a number of facial expressions, the blendshape "targets". By varying the weights of the linear combination, a range of facial expressions can be expressed with little computation. The set of shapes can be extended as desired to refine the range of expressions that the character can produce. In comparison with other representations, blendshapes have two major advantages:

- Blendshapes are a *semantic parameterization*: the weights have intuitive meaning for the animator as the strength or influence of the various facial expressions (Figure 1). Other linear models such as PCA do not provide this (section 7.7).
- To some extent blendshapes force the animator to stay "on model", that is, arbitrary deformations are not possible (Fig. 3). While this could be seen as limiting the artist's power, it helps ensure that the facial character is consistent even if animated by different individuals. It also enforces a division of responsibility between the character modeler and animator.

Although the blendshape technique is conceptually simple, developing a blendshape face model is a large and labor intensive effort at present. To express a complete range of realistic expressions, digital modellers often have to create large libraries of blendshape targets. For example the character of Gollum in the *Lord of the Rings* had 946 targets [Rai04]. Generating a reasonably detailed model can be as much as a year of work for a skilled modeler, involving many iterations of refinement.

The remainder of this report is organized as follows. Section 1 collects the industry terminology of blendshapes. Section 2 presents a brief history, though most related literature will be discussed in relevant later sections. Section 3 describes blendshapes from a linear algebra point of view, including recent variants such as "combination" blendshapes. Section 4 surveys methods of constructing blendshape models. Section 5 reviews interaction and animation techniques including performance-driven and direct manipulation approaches. Section 6 considers blendshapes as a high-dimensional interpolation problem. Section 7 considers blendshapes as a parameterization, and contrasts this approach with those based on principal component analysis. Section 8 considers applications and extensions, including blendshape-based expression cloning, model transfer, and refinement of models.

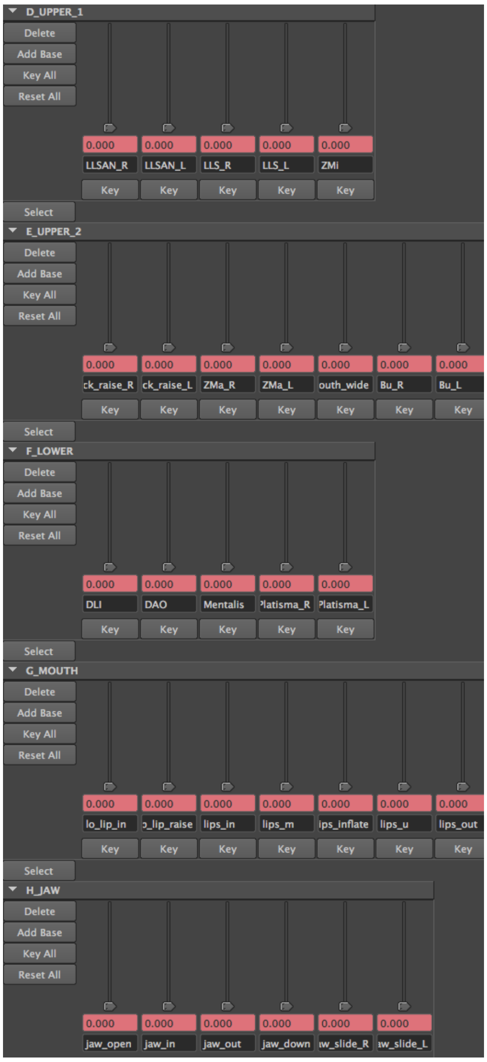


Figure 2: *Screenshot of a portion of the blendshape slider interface for a professionally created model (the complete slider interface does not fit on the computer display). The relatively simple model has 45 sliders.*

1. Terminology

The "blendshapes" term was introduced in the computer graphics industry, and we follow that definition: blendshapes are linear facial models in which the individual basis vectors are not orthogonal but instead represent individual facial expressions. The individual basis vectors have been referred to as *blendshape targets* and *morph targets*, or (confusingly) as *shapes* or *blendshapes*. The corresponding weights are often called *sliders*, since this is how they appear in the user interface (Fig. 2). A *morphable model* [BV99] is also a linear facial model, though it may focus on identity rather than ex-

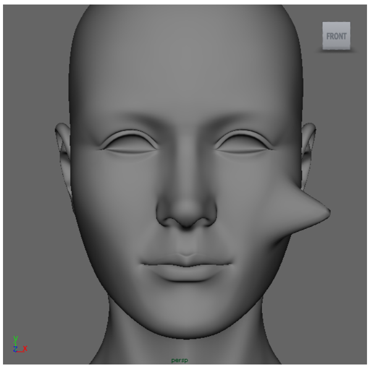


Figure 3: *Blendshapes prevent the artist from making "improper" edits such as this.*

pression, and its underlying basis is orthogonal rather than semantic.

From an artist's point of view, the interpretability of the blendshape basis is a defining feature. To manage the scope of this survey we will not attempt to fully survey techniques that make use of an orthogonal basis. Since the distinction is less important from a mathematical and programming point of view, however, relevant concepts that have to date only been employed with orthogonal models will be mentioned.

2. History

The origin of the blendshape approach is not generally associated with an academic publication, though it was well known in the computer graphics industry by the 1980s. Although Fred Parke is known for his pioneering work on the alternate parametric approach to facial modeling [Par72, Par74], he experimented with linear blending between whole face shapes [Par]. By the late 1980s the "delta" or offset blendshape scheme became popular [Bei05] and appeared in commercial software [Ber87,Els90]. In this variant a neutral face shape is designated and the remaining shapes are replaced by the differences between those shapes and the neutral shape. This results in localized control when the differences between the target shape and the neutral face are restricted to a small region, although it relies on the modeler to produce shapes with this property.

This idea was extended to a segmented face where separate regions are blended independently [Kle89], thus guaranteeing local control. A standard example is the segmentation of a face into an upper region and a lower region: the upper region is used for expressing emotions, while the lower region expresses speech [DBLN06].

While blendshape targets are most often considered as time-independent facial expressions, it is also possible to view individual blendshapes as being situated at particular times in the animation, and to simply cross-fade between them

$$\begin{bmatrix} f_{1x} \\ f_{1y} \\ f_{1z} \\ f_{2x} \\ f_{2y} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{nz} \end{bmatrix} = \begin{bmatrix} x & x & \cdots & x \\ y & y & \cdots & y \\ z & z & \cdots & z \\ x & x & \cdots & x \\ y & y & \cdots & y \\ \vdots & | & \cdots & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_n \\ | & | & \cdots & | \\ | & | & \cdots & | \\ | & | & \cdots & | \\ | & | & \cdots & | \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

Figure 4: *Vector-matrix expression of a blendshape model.*

to produce the final animation. This time-dependent blendshape approach provides the character designer or modeler full control and guarantees that particular expressions can be achieved at important points in the animation, but it requires the construction of many blendshapes that may not be reusable at other points in the animation. Some animations have combined the time-dependent and time-independent blendshape approaches [Zha01].

Additional literature on blendshapes will be mentioned in appropriate sections of the remainder of the report.

3. Algebra and Algorithms

Some insight and ease of discussion can be had by viewing blendshapes as a simple vector sum. To be concrete, consider a facial model composed of $n = 100$ blendshapes, each having $p = 10000$ vertices, with each vertex having three components x,y,z. By "unrolling" the numbers composing each blendshape into a long vector $\mathbf{b}_k$ in some order that is arbitrary (such as xxxyyyzzz, or alternately xyzxyzxyz) but consistent across the individual blendshapes (Fig. 4), the blendshape model is expressed as

$$\mathbf{f} = \sum_{k=0}^{n} \mathbf{b}_k w_k \tag{1}$$

or using matrix notation

$$\mathbf{f} = \mathbf{B}\mathbf{w} \tag{2}$$

where $\mathbf{f}$ is the resulting face, in the form of a 30000×1 vector, $\mathbf{B}$ is a $m = 30000 \times 100$ matrix ($m = 3p$) whose column vectors, $\mathbf{b}_k$, are the individual blendshapes (30000×1 vectors), and $\mathbf{w}$ are the weights (a 100×1 vector). We take $\mathbf{b}_0$ to be the blendshape target representing the neutral face. Thus blendshapes can be considered simply as adding vectors. This linear algebra viewpoint will be used to describe various issues and algorithms.

© The Eurographics Association 2014.

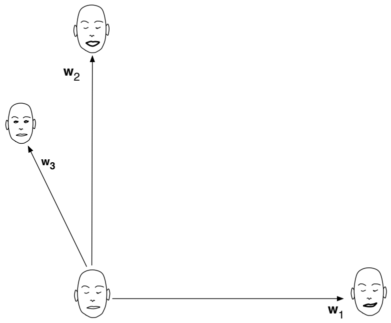


Figure 5: *The basic delta blendshape scheme can be visualized as situating targets at vertices of a hypercube that share an edge with the neutral face at the origin.*

Equation (2) represents the global or "whole-face" blendshape approach. In this approach scaling all the weights by a multiplier causes the whole head to scale. Overall scaling of the head is more conveniently handled with a separate transformation, however. To eliminate undesired scaling the weights in equation (2) may be constrained to sum to one. Additionally the weights can be further constrained to the interval $[0,1]$, as described in section 7.5.

3.1. Delta blendshape formulation

In the local or "delta" blend shape formulation, one face model $\mathbf{b}_0$ (typically the resting face expression) is designated as the "neutral" face shape, and the remaining targets $\mathbf{b}_k, \quad k = 1 \ldots N$ are replaced with the difference $\mathbf{b}_k - \mathbf{b}_0$ between the kth face target and the neutral face:

$$\mathbf{f} = \mathbf{b}_0 + \sum_{k=1}^{n} w_k(\mathbf{b}_k - \mathbf{b}_0) \tag{3}$$

(with $\mathbf{b}_0$ being the neutral shape). We denote this as

$$\mathbf{f} = \mathbf{b}_0 + \mathbf{B}\mathbf{w} \tag{4}$$

(note that we are reusing variable names from equation (2)). In this formulation the weights are conventionally limited to the range $[0,1]$, although there are exceptions to this convention. For example the Maya [Tic09] blendshape interface allows the [0,1] limits to be overridden by the artist if needed.

If the difference between a particular blend shape $\mathbf{b}_k$ and the neutral shape is confined to a small region, such as the left eyebrow, then the resulting parameterization offers intuitive localized control.

The delta blendshape formulation is used in popular packages such as Maya, and our discussion will assume this variant if not otherwise stated. Many comments apply equally (or with straightforward conversion) to the whole-face variant.

A blendshape model can be considered as placing targets at (some of) the vertices of a n-dimensional hypercube, with the origin being the neutral shape, and hypercube edges representing weights on the corresponding targets (Figure 5). In this and following figures, a small face image represents the collection of vertex components of a particular blendshape target (A more veridical representation would have a collection of m plots $u = f(w_1, w_2)$ where u represents a particular (x,y, or z) component of one of the vertices).

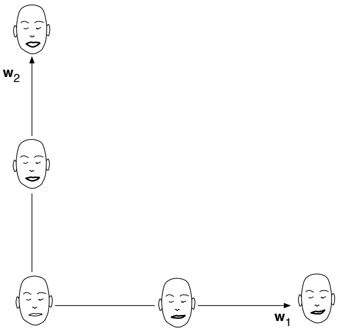


Figure 6: *Blendshape targets can be situated at intermediate locations, resulting in piecewise linear interpolation to the full target.*

3.2. Intermediate shapes

As an individual weight in Eq. (4) varies from zero to one, the moving vertices on the face travel along a line. To allow more fidelity, production blendshape implementations such as that in Maya [Tic09] allow targets to be situated at intermediate weight values, giving piecewise linear interpolation. This is shown schematically in Figure 6.

3.3. Combination blendshapes

Another blendshape variant, described in [Osi07, Ver], adds additional "correction" shapes that become active to the extent that particular pairs (or triples, etc.) of weights are active.

This approach might be notated as

$$\begin{aligned}\mathbf{f} = \mathbf{f}_0 &+ \mathbf{b}_1 w_1 + \mathbf{b}_2 w_2 + \mathbf{b}_3 w_3 + \cdots \\ &+ \mathbf{b}_{1,5}\, w_1\, w_5 + \mathbf{b}_{2,13}\, w_2\, w_{13} + \cdots \\ &+ \mathbf{b}_{2,3,10}\, w_2\, w_3\, w_{10} + \cdots\end{aligned}$$

© The Eurographics Association 2014.

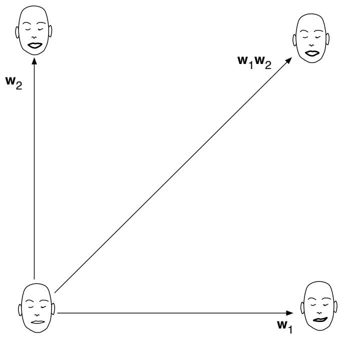


Figure 7: *Schematic illustration of the "combination blendshape" idea [Osi07]. A correction shape (top right) is added with weight* $w_1 \cdot w_2$.

Here the first line is equivalent to equation (4). A term $\mathbf{b}_{1,5}\, w_1\, w_5$ is a bilinear "correction" shape that is fully added only when w_1 and w_5 are both one, and is completely off if either is zero. The irregular numbering $1,5$ is intended to indicate that these corrections are only needed for particular pairs (or triples, quadruples) of shapes such as shape 1 and shape 5. For example, the eyebrow and mouth corner are spatially well separated, so it is unlikely that any correction shape would be needed for this pair of shapes. A schematic visual representation of this approach is shown in Figure 7. The combination targets are situated at (some of) the diagonals of the blendshape hypercube.

The majority of the blendshape targets in modern professional models with hundreds of targets are these combination shapes. As a rule of thumb, the primary targets (those situated at the hypercube vertices that are neighbors of the neutral shape) may number 100 shapes, whereas the number of combination shapes may be several hundred or more [Rai04]. The combination blendshape idea should be distinguished from the on-line "correction" shapes that have been a subject of recent research (section 8.2). Correction shapes modify or add to the linear blendshape basis, whereas combination shapes can be seen as a second-order term in a Taylor series in the blendshape weights (section 6.3).

The combination blendshape scheme is not ideal from an interpolation point of view. When the facial expression travels along the (hyper)diagonal toward a 2nd order correction, the correction is added in as $s^2\mathbf{b}_{i,j,\ldots}$ where $s^2 = w_i w_j$ is the fractional distance along the diagonal. Thus the correction has little effect over most of the range and then appears relatively suddenly. The problem is exacerbated with 3rd and higher order corrections. This can be partially addressed by placing intermediate shapes along the diagonal.

3.4. Hybrid rigs

In a blendshape model the jaw and neck are sometimes handled by alternate approaches. For example, since the motion of the jaw has a clear rotational component, the `jaw-open` target is often augmented by linear blend skinning [OBP*12]. The eyelids are another area that is sometimes handled by alternate rigging approaches, again due to the rotational motion. [OBP*12] is a recent survey of facial rigging techniques.

4. Constructing Blendshapes

There are several approaches for creating blendshapes. A skilled digital artist can deform a base mesh into the different shapes needed to cover the desired range of expressions. Alternatively, the blend shapes can be directly scanned from a real actor or a sculpted model. A common template model can be registered to each scan in order to obtain vertex-wise correspondences across the blendshape targets. Methods to register scans (and register a generic template to scans) include [LDSS99, ARV07, SE09, WAT*11, ACP03, ASK*05].

In [PHL*98] blendshape targets are rapidly constructed with minimal manual assistance from multiple pictures of an actor. [BV99] fits a morphable model (PCA model of both the geometry and texture) to a single image, resulting in an estimate of the geometry and texture of the person's face. Typically the geometry of a facial model is fairly coarse, with fine scale details such as wrinkles and freckles represented via textures, bump or normal maps, or recent techniques such as [MJC*08, BLB*08]. This decomposition makes good use of graphics hardware, and the choice of relatively coarse geometry in facial model capture and tracking applications can also be motivated from bias-variance considerations in model fitting.

In concept, a dynamic mesh obtained from dense motion capture can be decomposed into a linear model using principal component analysis or other approaches. However, the PCA models lack the interpretability of blendshapes. This will be discussed further in sections 7.7 and 8.3.

Blendshape models can also be constructed by transferring the expressions from an existing source model to a target model with different proportions. These model transfer approaches are described in section 8.1.

We informally refer to the set of mesh vertices and edges as a topology. Given an existing set of blendshape models with common topology, it is possible to create new faces as weighted combinations of the existing models. This may be suitable for producing background or "crowd" characters.

© The Eurographics Association 2014.

This approach is somewhat limited however: consider the case of two models, one of which has big eyes and a small mouth, and a second which has small eyes and a big mouth. Using a global linear combination of these two models, it is not possible to produce a new face with small eyes and a small mouth. A further issue is that (by a form of the central limit theorem) the blend of a number of faces will tend towards a Gaussian distribution, and a convex combination will have less distinct features than the basis faces. Allowing different linear combinations in different regions of the face is an obvious approach to increasing the diversity of the generated faces. This generates a new problem, however, in that the constructed regions will have discontinuities at their boundaries. Blending between regions is a poor solution, in that it is not obvious what the transition region should be, and in cases where the regions are quite different the transition may look unnatural. [MBL12] solved these issues by blending in the gradient domain (thereby providing continuity) and solving a Poisson problem to generate the composite face.

5. Animation and Interaction Techniques

Animating with blendshape requires specifying weights for each frame in the animation. Animation techniques can be broadly divided into performance-driven animation techniques, keyframe animation, and direct manipulation. Performance-driven animation is commonly used to animate characters different from the actor, so expression cloning techniques will also be surveyed here. The section will conclude with a brief survey of alternative editing techniques.

5.1. Keyframe animation

Blendshape models have traditionally been animated using keyframe animation of the weights (sliders). Commercial packages such as Maya provide spline interpolation of the weights and allow the tangents to be set at keyframes. As an approximate figure, professional animation requires a keyframe roughly every three frames. Many animators prefer that keyframes include keys for all targets, rather than putting keys on each curve independently.

5.2. Performance-driven animation

In performance-driven facial animation, the motion of a human actor is used to drive the face model [Wil90, CDB02, BBPV03, PL06, PL05]. Whereas keyframe animation is commonly used in animated films with stylized characters, performance-driven animation is commonly used for visual-effects movies in which the computer graphics characters interact with filmed characters and backgrounds. Because blendshapes are the common choice for realistic facial models, blendshapes and performance-driven animation are frequently used together. The general literature on face tracking in general spans several decades and a complete survey is beyond the scope of this report. We will concentrate on performance capture methods that drive a blendshape rig. Technique that drive a low-level representation such as a mesh or a mesh driven by skinning [Wil90, GGW*98, BPL*03, BHPS10] will not be surveyed.

Performance capture methods can be classified into those that use 3D motion capture information as input [CLK01, DCFN06] versus methods that do model-based tracking of video [PSS99, BBPV03, CXH03, RHKK11, BGY*13, CWLZ13]. Another distinction is whether a PCA basis [BBPV03] or blendshape basis [PSS99, CK01, CLK01, CDB02] is used. [DCFN06] uses a PCA basis for the motion capture which is then retargeted to a blendshape basis through a nonlinear radial basis mapping [DCFN06]. [TDlTM11] uses overlapping local PCA models.

Model-based tracking of blendshapes solves for the blendshape weights at each frame. Typically the weights are constrained to the range $0 \ldots 1$. When the source motion to match is available in the form of 3D motion capture, this is a constrained linear problem that can be solved with quadratic programming [CK01,CLK01,JTDP03]. When model-based tracking is used to match images from a video, the perspective nonlinearity requires the use of nonlinear optimization (unless weak perspective is employed). [PSS99] allowed soft constraints with a Levenberg-Marquardt algorithm.

With the popularity and affordability of low-cost commercial depth cameras (e.g., Microsoft's Kinect), researchers have developed a number of techniques to utilize such cameras for performance driven facial animation. One approach does real-time tracking and transfers the facial movement to a user-specific blendshape face model that is manually constructed at the offline stage [WLGP09, WBLP11]. Recent advances include online modeling of user-specific blendshape faces (without the offline step) and introduction of adaptive corrective shapes at runtime for high-fidelity performance driven facial animation applications [BGY*13, LYYB13, BWP13]. These basis adaptation approaches are briefly surveyed in section 8.2.

5.3. Expression cloning

In *expression cloning techniques* [NN01, SP04], the motion from one facial model (the "source") is retargeted to drive a face (the "target") with significantly different proportions. Expression cloning is frequently the goal of performance-driven animation. For example, an adult actor may produce the motion for a younger or older person (as in the movies *The Polar Express* and *The Curious Case of Benjamin Button*) or a non-human creature (as in *Avatar* and the Gollum character in the *Lord of the Rings* movies). A related problem is that of creating a full target face model, given the source face but only limited samples of the target, usually only the neutral shape. This problem is discussed in section 8.1.

[NN01] introduced the expression cloning problem. This

© The Eurographics Association 2014.

approach requires only a generic animated facial mesh for the source and makes no assumption of a blendshape or other representation. It establishes a mapping by finding corresponding pairs of points on the source and target models using face-specific heuristics. [VBPP05] discovers a tensor basis that spans both expression and identity in different dimensions. Identity can be flexibly manipulated in this approach, however it does not use a blendshape basis.

The dominant industry approach to expression cloning is what might be termed "parallel parameterization" [SG06, HIWZ05, LWP10]: source and target blendshape models are constructed to have the same number of targets, with the same semantic function (typically FACS inspired [SG06]). The blendshape weights are then simply copied from source to target. This approach is simple, and allows great flexibility in developing the cross-mapping. For example, one could imagine a `smile` morph target for a lizard character in which the mouth corners move backward rather than upward. The parallel parameterization approach may have been first demonstrated by SimGraphics in the 1990s [Wil01].

If the source and target models already exist, but do not share a parameterization, it may be possible to learn a cloning function given sufficient examples of corresponding poses. In a linear version of this idea, there need to be $p \geq n$ corresponding poses if the models contain n blendshape targets. Let $\mathbf{w}_k$ be the blendshape weights for the source, and $\mathbf{v}_k$ be the blendshape weights for the target, for each pair k of corresponding poses. Gather $\mathbf{w}_k$ and $\mathbf{v}_k$ as the columns of matrices $\mathbf{W}, \mathbf{V}$ of dimension $n \times p$. Then an expression cloning matrix $\mathbf{E}$ of dimension $n \times n$ that maps $\mathbf{w}$ to $\mathbf{v}$ can be found,

$$\begin{aligned}
\mathbf{W} &= \mathbf{E}\mathbf{V} \\
\mathbf{W}\mathbf{V}^T &= \mathbf{E}\mathbf{V}\mathbf{V}^T \\
\mathbf{E} &= \mathbf{W}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T)^{-1}
\end{aligned}$$

This simple *linear expression cloning* approach has its limitations – in particular in that the mapping is linear (as is also the case with parallel parameterization).

Most existing expression cloning algorithms do not consider adapting the temporal dynamics of the motion to the target character, and instead assume that if each individual frame can be transferred correctly, the resulting motion will be correct. This will tend to be the case if the source and target are of similar proportions.

There are several scenarios in which the temporal dynamics of face movement should be considered however. One case is where the target cannot reproduce the full range of movement of the source model. These limits commonly occur when a blendshape model is driven directly by motion capture. They also can occur even when the source is a blendshape model. For example, the target model might allow `jaw-open` to range up to 1, but it may be that the results look unnatural if `smile` is also active with a value of more than 0.7. This situation can be handled with an expression that sets the limit on the `jaw-open` as `jaw-open-limit = 1 - 0.3 * smile`.

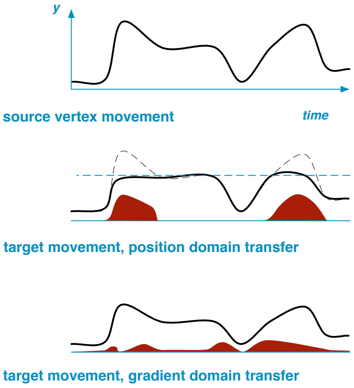


Figure 8: *The "movement matching" principle in [SLS*12]. The target cannot fully reproduce source movement (top). Attempting to best reproduce the position of each frame results in clipping when the pose is not achievable (dashed line in middle figure). Instead, the movement matching principle attempts to match the temporal derivatives, thereby reproducing the shape of the motion (bottom). The red shaded areas indicate the magnitude and distribution of the matching error.*

In this situation, [SLS*12] argue that reproducing the source on a per-frame basis results in unnatural motion when the target motion limit is reached. They propose that it is better to preserve the overall shape of the motion, rather than matching the position of each frame independently. This objective is formulated by saying that the temporal derivatives (rather than positions) should be matched in a least squares sense. This leads to a space-time Poisson equation that is solved for the target blendshape motion.

More generally, most current expression cloning techniques require that the target expression for a particular frame be a function of the source expression for that frame only. More powerful expression cloning techniques may require looking at adjacent frames in order to allow anticipation and coarticulation-like effects to be produced. An open problem is the case in which the target motion should differ from that of the source is when the target has significantly different proportions or size from the source. The human mouth moves very quickly during speech – for example the mouth can change from a fully open to a fully closed position in two adjacent video frames. Transferring this rapid motion to a large and non-humanoid character such as a whale would likely give implausible looking results.

On the other hand, we recall the antropomorphic principal that the target character is usually humanoid if not human – if the character needs to be perceived by human audiences, it needs to express facial emotion in human-like ways.

Retargeting of motion capture requires determining the coordinate frame of the skull. The motion of this frame is removed, and the remaining motion of the face is recorded as the facial expression. One open problem is that the rigid coordinate frame of the skull is not easily determined, and if it is poorly estimated subsequent analysis may conflate head motion with expression change. The issue is that people cannot naturally produce expressions without simultaneously moving the head. One approach to this problem is to attempt to find at least three relatively stationary points on the face, and estimate the rigid transform from these – typical candidates are the corners of the eyes and the nose tip. However, some people slightly move these points (relative to the skull) while making extreme expressions. Another solution is to identify the head motion using a rigid hat. However vigorous movement or particular expressions (such as raising the eyebrows strongly) may cause the hat to move slightly. Facial expressions can be very subtle (consider the geometric difference between a face expressing the two emotions "calm" and "contempt"). While the rigid hat solution is most often used in practice, an even more accurate solution would be desirable.

5.4. Partially-automated animation

In practice, performance-driven animation is rarely used without subsequent manual adjustment. On reason for this is lack of fidelity or errors in the motion capture process. For example, marker-based systems typically place markers around the outside of the mouth are not able to track the inner contour of the lips ([BGY*13] is a recent exception). Similarly, most motion capture systems do not track the eyes or eyelids.

There is another important reason for editing performance-driven animation: changes in the acting may be required. This may be because a performance that is automatically transferred to a different (e.g. non-human) character may not convey the intended emotion. As well, a movie director can request changes in the performance. For these reasons, a viable performance-capture system must allow for subsequent manual editing by artists. This is a major reason why existing performance capture approaches use a blendshape representation.

Subsequent editing of motion capture presents a further problem: motion capture produces weight curves with a key at every frame. This is too "dense" for artists to easily edit. [SSK*11, LA09] introduced an optimal curve simplification technique using dynamic programming. With a GPU implementation, it can produce roughly an 80% reduction in sample density with little or no visible difference in the resulting curve.

5.5. Direct manipulation

Blendshapes have traditionally been animated with keyframe animation or by motion capture. Although inverse kinematics approaches to posing human figures have been used in animation for several decades, analogous inverse or *direct manipulation* approaches for posing faces and setting keyframes have appeared only recently. In these approaches, rather than editing the underlying parameters (as in forward kinematics, and keyframe animation), the artist directly moves points on the face surface and the software must solve for the underlying weights or parameters that best reproduce that motion.

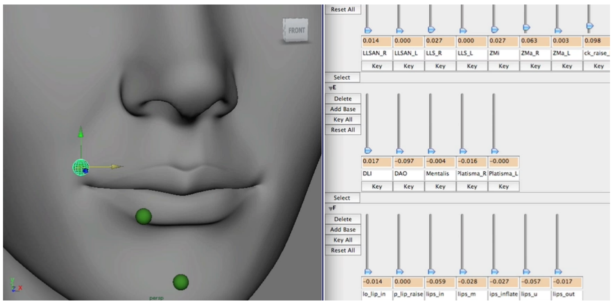
The evident challenge for direct manipulation of faces is that it can be a very under-constrained inverse problem – similar to inverse kinematics, but more so. In moving the hand of the character using inverse kinematics, for example, the animator specifies a goal point (3 degrees of freedom), and animation system must solve for on the order of 10 degrees of freedom representing the joint angles from the hand through the shoulder. In a professional blendshape model, the analogous number of unknown weights can be 100 or more. Solving the inverse problem for direct manipulation blendshapes then means that we find a discrete function (i.e., a vector $\Delta\mathbf{w}$) that satisfies the constraint given by a pin-and-drag manipulation [YN03] onto a 3D face model. This requires an appropriate prior to regularize the inverse solution. The resultant weights are then (usually automatically) interpolated to make a whole animation.

It is important to note that professional animation requires providing both direct manipulation and access to the underlying parameters (sliders). Intuitively, this is because some edits are simply harder to accomplish using direct manipulation. In fact it is easy to argue on mathematical grounds that slider manipulation is necessarily more efficient for some edits, whereas the converse is also true – direct manipulation is necessarily more efficient for other edits. Briefly, this is because of the spreading effect of a multiplication by a non-identity matrix. In direct manipulation the blendshape weights are in a pseudoinverse relationship to the manipulated points, and columns of the pseudoinverse tend to have a number of non-zero values [LA10].

5.5.1. Direct manipulation of PCA models

The underconstrained direct manipulation inverse problem was first solved by several approaches that use an underlying PCA representation. [ZSCS04] allow direct face editing using local and adaptive radial basis blends of basis shapes. The shapes are obtained by tracking high quality stereo data with a template mesh. [ZLG*06] develop a hierarchical segmented principal component (PCA) model. User-directed movement of a particular point on the face is propagated to the rest of the face by projecting the altered point vector into the subspace and iterating this procedure over the remainder of the hierarchy. [MA07] learn a PCA subspace of facial

© The Eurographics Association 2014.



0.014	0.000	0.027	0.000	0.027	0.063	0.003	0.098
LLSAN_R	LLSAN_L	LLS_R	LLS_L	ZMi	ZMa_R	ZMa_L	ck_raise_

0.017	-0.097	-0.004	-0.016	-0.000
DLI	DAO	Mentalis	Platisma_R	Platisma_L

-0.014	0.000	-0.059	-0.028	-0.027	-0.057	-0.017
lo_lip_in	p_lip_raise	lips_in	lips_m	ips_inflate	lips_u	lips_out

Figure 9: *Screenshot of a direct manipulation interface in operation. (Left panel) selecting a point on the model surface creates a manipulator object termed a pin. These can be dragged into desired positions, and the system solves for the slider values (right panel) that cause the face to best match the pinned positions.*

poses. This is used to bypass computation of a majority of face points, by "PCA imputation" wherein a subset of points is computed and fit and the same linear combination is used to estimate the locations of the remaining points. [LD08] use a local, hierarchical PCA face model; facial editing is performed with a constrained version of weight propagation [ZLG*06]. This provides local control while also allowing natural cross-region correlations. [LCXS09] develop direct dragging and stroke-based expression editing on a PCA model obtained from motion capture data, and include a statistical prior on the space of poses.

These PCA approaches are good solutions if the model will be manipulated exclusively with direct manipulation, and this is the most appropriate choice for novice users. Since professional animation also requires access to the underlying sliders however, this in turn necessitates the use of an underlying blendshape rather than PCA representation due to the lack of interpretability of the PCA basis (section 7.7). While it is easy to interconvert between PCA and blendshape models (section 7.8), doing so requires having a blendshape model.

5.5.2. Direct manipulation of blendshapes

[ZSCS04] included a direct manipulation algorithm in their facial capture system. It used a basis of selected frames from a captured performance, which does not meet our definition of "blendshape", however it is also not a PCA basis. They introduced an interesting regularization for the direct manipulation inverse problem, in which the basis meshes most similar to the desired constraints are weighted more heavily. This is an effective approach to extending the span of a model with a limited number of shapes (see Figure 6 (d),(e) in [ZSCS04]), though with a well developed model this property might be considered undesirable.

The inverse problem can be avoided by using a fully constrained approach, exactly as would be used for performance driven animation. In this approach the artist interacts with manipulators that serve the same role as motion capture markers. The manipulators cover the face and are moved one at a time, with the others remaining stationary. The first published approach to direct manipulation of blendshape models [JTDP03] used this approach.

While constraining the face with a full set of manipulators avoids the inverse problem, it can also increase the required number of edits since no part of the face is free to move without intervention from the artist. Formulating direct manipulation as an underconstrained inverse problem allows many parts of the face to move during each edit, but requires a sensible regularization to make this useful (the previous fully constrained version of the problem can be recovered as a special case by adding sufficient constraints). [LA10] started with the principle that moving a particular part of the face should cause the remainder of the face to change as little as possible – a principle of "least surprise". To embody this in an algorithm, they observe that the blendshape model itself is designed as a semantic parameterization, that is, the targets are sculpted so that facial expressions can be described by the combination of n sliders, each with approximately equal effect on the facial expression. This is in contrast to PCA, where the higher coefficients by definition have smaller influence. Thus the change in facial expression is to a first approximation represented by the change in weights, as demonstrated in Figure 1. In this figure a Euclidean distance on the control vertices indicates that the full smile and open-mouth expressions are most similar, but the distance between the blendshape weight vectors correctly indicates that the smile is more similar to the half-smile.

[SILN11] presents a direct manipulation system suitable for use in animation production, including treatment of combination blendshapes and non-blendshape deformers. They add an improved regularization term that better handles the common case where the artist repeatedly moves a single slider over the same range of values in order to understand its effect. The nonlinear components of their production-quality rig are handled with a combination of nonparametric regression (for the jaw) and a derivative free nonlinear optimizer. [ATJ12] describes an extension of the direct manipulation approach [LA10], which allows more efficient edits using a simple prior learned from facial motion capture. This system also allows the artist to select between three different modes at any time during editing: sliders, regular, and learned direct manipulation (see section 7.8). [NVW*13] show direct manipulation of an automatically created local linear model. This work is discussed in section 8.3.

5.6. Further interaction techniques

[PHL*98] proposes a painterly interface for creating facial expressions. The interface has three components: a canvas for designing a facial expression, a brush interface that

© The Eurographics Association 2014.

let the user selects the intensity and decay of the strokes, and a palette where the colors have been replaced by facial expressions. When a stroke is applied to the facial canvas, weights from the selected facial expression are transferred and blended. When completed the facial canvas can be added to the facial palette and selected to design more complex expressions.

While direct manipulation offers advantages over the traditional slider editing, a more fluid or "sketch based" interface [MAO*11] might be preferable for both novice users and for previsualization of professional animation. Development of a sketch-based system that interoperates with an underlying blendshape basis is an open problem.

6. Facial Animation as an Interpolation Problem

Blendshapes are perhaps the simplest approach to facial animation imaginable, and limitations of the linear model are evident. In this section we discuss blendshapes in abstract as a problem of interpolation, and consider whether a better approach may be possible.

6.1. Linear interpolation

The linear nature of blendshapes affects the animation in some cases. In the interpolation from one target to another, two weights change in a convex combination, and the movement of each changing vertex is necessarily along a line. If the two weights are not in an affine (sum-to-one) combination, the movement is constrained to a plane, etc. More generally, the blendshape scheme constrains movement to a n dimensional subspace of the $3m$-dimensional ambient space.

6.2. Blendshapes as a high dimensional interpolation problem

In abstract, facial animation is an interpolation problem of the form

$$f : \mathbf{R}^n \rightarrow \mathbf{R}^{3p}$$

that maps a set of n animation control parameters (such as $n \approx 100$ for blendshape sliders) to the $3p$ values, where p is the number of control vertices ("points") of the 3D face model.

A scattered interpolation scheme might seem an ideal solution to the problem of interpolating a number of targets in a high dimensional space, since the sculpted faces could be placed at arbitrary (scattered) desired locations in the parameter space $\mathbf{w}$ (Figure 10). In a radial basis function (RBF) approach the kernel could be chosen to as the Green's function of a differential operator, resulting in smooth interpolation of the data. This formulation would also separate the number of targets from the dimensionality of the space.

Unfortunately, high-dimensional interpolation is known to be intrinsically difficult [Caf98, Gar]. The Green's function corresponding to the differential operator family ∇^{2s} is defined as [Duc76, LPA10]

$$R(\mathbf{x}) \propto \begin{cases} |\mathbf{x}|^{2s-n} \log |\mathbf{x}| & \text{if } 2s-n \text{ is an even integer,} \\ |\mathbf{x}|^{2s-n} & \text{otherwise} \end{cases} \qquad (5)$$

for smoothness order s and space dimension n.

This requires a condition $2s > n$ in order to avoid having a singularity at the origin. For typical cases in graphics applications, where $n = 2$ or 3 and $s = 2$, the kernel functions work well. However, in our context, we need to consider a low-order smoothness $s = 2$ in the case of our 100-dimensional "face space". The kernel would then be like

$$R(\mathbf{x}) \propto |\mathbf{x}|^{-50}$$

which has a singularity at the origin and is (for numerical purposes) zero elsewhere. A practical RBF interpolation scheme could not be built on these assumptions. More generally, we expect that any approximation scheme will for a given approximation error require a number of samples exponential in the dimension due to the curse of dimensionality [HTF09], unless the scheme can identify that the data lives on a lower-dimensional manifold.

Thus, we have the open problem of interpolating in a high (e.g. $n = 100$) dimensional space. One possibility would be to dramatically increase the order of smoothness s. While this has not been explored, it can be noted that in other applications in computer graphics $\mathcal{C}^2$ smoothness has often proven sufficient, and at present we have no reason to believe that the motion of the face between expressions is extremely smooth.

6.3. Blendshapes as a tangent space

Equation 4 resembles a vector-valued Taylor series expansion about the neutral face, i.e.,

$$f(\mathbf{w}) = f(0) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \cdot \mathbf{w}$$

with $f(0) \equiv \mathbf{b}_0$ and the Jacobian $\left[\frac{\partial f_i}{\partial w_j}\right] \equiv \mathbf{B}$. In abstract geometric terms, we might consider blendshapes to be the tangent space (about the neutral face) of the n-dimensional face "manifold" embedded in a m-dimensional ambient space. As we move from one point to another along this 100-dimensional tangent space, the location in the 30000 dimensional ambient space also changes.

This comparison to a Taylor series suggests limitations of the blendshape approach, and one wonders whether an alternative approach is possible. The blendshape approach requires the artist to sculpt n shapes at all the locations in weight space $\mathbf{w}_i = \delta_{i,k}$ for $k = 1 \ldots n$ (the vertices of the hypercube connected by an edge to the neutral shape, (Figure 5), i.e. the "one-ring" of the neutral). It is not possible for the artist to specify shapes at an arbitrary location such as

© The Eurographics Association 2014.

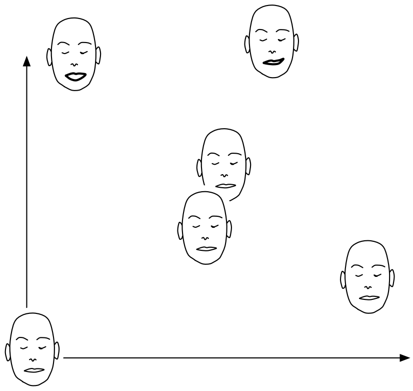


Figure 10: *Blendshape schemes require that targets are placed at constrained locations, i.e. the vertices of a "weight hypercube" (Fig. 7). It would be preferable to allow targets to be placed anywhere in face space, allowing the sculpting effort to be directed specifically where it is needed.*

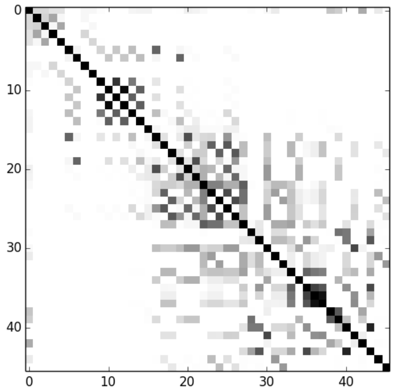


Figure 11: *Mutual coherence plot for the 46-target blendshape model shown in Figure 9 and other figures. The i, j entry is the covariance between the i-th and j-th blendshape targets, i.e.* $\frac{\mathbf{b}_i^T \mathbf{b}_j}{\|\mathbf{b}_i\| \|\mathbf{b}_j\|}$.

$w = 0.3, 0.7, 0.1, \cdots$ (Figure 10). If the facial model is incorrect at an arbitrary location, current systems require the artist to modify a number of targets so that their weighted sum reduces the desired correction, while simultaneously not disturbing other face poses. This is an time-consuming iterative refinement procedure.

[SSK*12] described a hybrid approach in which a basic blendshape model is augmented with additional nonlinear corrections. The corrections are interpolated by a radial basis function scheme inspired by weighted pose space deformation [KM04], with the underlying blendshape weights defining the pose space. This approach allows shapes to be placed as needed at any pose of the model (Figure 10) and the interpolation is smooth and free of artifacts such as the quadratic ramp-up that occurs with combination shapes (section 3.3). The use of the weighted form of pose space deformation [KM04] partially addresses the curse of dimensionality inherent in this high dimensional problem, by breaking the global interpolation problem into a collection of softly coupled local problems.

7. The Blendshape Parameterization

Despite the simplicity of the blendshape representation, there are a number of associated issues, and in fact blendshapes provide an interesting 'workshop" for discussing general issues of representation and parameterization. These issues will be surveyed in this section.

7.1. Lack of orthogonality

The major distinguishing characteristic of blendshapes relative to the more common principal component representation is that the shapes are not orthogonal (Figure 11). This has the advantage of interpretability (section 7.6). It has the disadvantage that the parameters are not independent, and so adjusting a parameter can degrade the effects obtained with previous edits. [LMDN05] addressed this problem with a user-interface technique in which the artist can "pin" particular points representing desirable aspects of the current facial expression, and subsequent edits occur in the approximate null-space of these constraints.

7.2. Blendshape models are not unique

There are an infinite number of different blendshape models that can produce the same range of animation as a particular blendshape model. Intuitively, this is similar to the fact that an infinite number of vector pairs span the plane, and given two such vectors (analogous to a particular "model"), another pair can be constructed as weighted combinations of the original vectors - for example the sum and difference of the original pair is one such basis. Given a particular blendshape model $\mathbf{B}$, an arbitrary non-singular $n \times n$ matrix $\mathbf{R}$ and its inverse can be inserted between the $\mathbf{B}$ and the weights without changing anything:

$$\mathbf{f} = \mathbf{B}\left(\mathbf{R}\mathbf{R}^{-1}\right)\mathbf{w}$$

Then $\mathbf{BR}$ is a new blendshape basis with corresponding weights $\mathbf{R}^{-1}\mathbf{w}$ that produces the same range of motion as $\mathbf{B}$.

© The Eurographics Association 2014.

7.3. Equivalence of whole-face and delta blendshape formulations

Proponents of various blendshape approaches are outspoken in industry forums regarding the proposed advantages of each particular approach. While working in the entertainment industry, one of the authors heard emphatic claims that the delta form is the most powerful form of blendshape, or alternately that using targets modeled after the FACS poses [SG06, ER97] is the only approach that produces all and only the full set of valid face shapes. In fact it is simple to show that, while these techniques have their respective advantages, they are equivalent in expressive power and the desired range of expressions does not uniquely specify a blendshape model.

The delta formulation equation (4) and the whole-face form equation (2) can be seen to be equivalent (in the terms of the range of shapes produced) by rewriting equation (1) as

$$\begin{aligned}\mathbf{f} &= \sum_{k=0}^{n} w_k \mathbf{b}_k \\ &= w_0\mathbf{b}_0 + \sum_{k=1}^{n} w_k \mathbf{b}_k \\ &= w_0\mathbf{b}_0 + \sum_{k=1}^{n} w_k \mathbf{b}_k - \sum_{k=1}^{n} w_k \mathbf{b}_0 + \sum_{k=1}^{n} w_k \mathbf{b}_0 \\ &= \left(\sum_{k=0}^{n} w_k\right)\mathbf{b}_0 \quad + \quad \sum_{k=1}^{n} w_k (\mathbf{b}_k - \mathbf{b}_0) \qquad (6)\end{aligned}$$

If the whole-face weights are convex (as is generally the case) this exactly recovers the delta-face formulation (3).

It is intuitive to think of local blendshapes as having more power for a given number of targets. For example, if there are n_1 shapes for the mouth and lower face, n_2 for the right eye and brow, and n_3 for the left eye and brow, then we may be tempted to consider that the resulting system would require $n_1 \cdot n_2 \cdot n_3$ whole-face shapes to have equivalent power. In fact this is incorrect, as suggested by equation (6) above. As an analogy, consider a pixel (sample) basis and a Fourier basis. The former is maximally local, yet spans the same space as the latter.

As an example, consider a blendshape model that has these targets: left-eye-closed, right-eye-closed. In the delta scheme, creating a model with both eyes closed requires corresponding weights (1,1). In the whole-face scheme, setting the weights to (1,1) would cause the head to scale, whereas setting them to (0.5,0.5) will give a the result of two half-closed eyes. However if we notate the delta blendshapes as b_1, b_2, and the corresponding whole-face targets as $B_1 = b_1 + n, B_2 = b_2 + n$, simple algebra gives the result that the desired closed-eye expression in delta form, $b_1 + b_2 + n$, is equivalent to $B_1 + B_2 - n$. Note that this is not a convex weight combination.

7.4. Global versus local control

In general, both global and local specification of shape deformation may be desirable. Global specification is desirable when the modeler is given a picture or sculpted maquette of a complete head that they must match with a computer model. Modeling a set of heads with various facial expressions is a more natural task than modeling the corresponding "delta" shapes such as the displacements governing eyebrow movement. Global specification is also used in some animation scenarios, such as the time-dependent blendshape modeling approach mentioned in section 2.

On the other hand, many animation tasks are more easily approached if local control is available. For example, increasing the width of the mouth is more easily accomplished if only one or a few blend shapes affect the mouth region than in the situation where every basis vector affects all regions of the face including the mouth. While producing the desired effect should be possible in an equivalent system of non-localized blendshapes (equation (6)), the global effect of each blendshape combined with their interaction with other shapes (see section 7.1) results in a tedious trial and error process for the artist. Fortunately, equation (6) points out that converting between whole-shape and delta formulations is a simple matter. Because of this equivalence and the simplicity of converting between the whole-face and delta formulations, it is not necessary to restrict oneself to the choice of one representation over the other – the user interface can allow the artist to select between the whole-face and delta forms according to the particular task.

As noted above, local control can be obtained with the delta blendshape formulation if the changes in the target faces are restricted to small areas. This may be difficult to obtain in some common modeling methodologies, however, as when the target faces are digitized from physical models. We also noted that local control can be guaranteed by segmenting the face into separate regions each of which has an independent set of blend shapes [Kle89]. Unfortunately the ideal segmentation may be difficult to choose in advance, particularly because designing blend shapes is a trial-and-error process, with many iterations of modeling corrections typically being required. [LD08] approaches the problem with a hierarchical (PCA) basis, thereby providing both local control and cross-region correlations. Automated creation of localized blendshapes is a goal of several research efforts [JTDP03, DCFN06, NVW*13]; these approaches are discussed elsewhere in this report.

7.5. Convex combination of shapes

Whole-face blendshape interpolation can be restricted to convex combinations by enforcing the following constraints on the weights

$$\begin{aligned} \textstyle\sum_{k=1}^{n} w_k &= 1 \\ w_k &\geq 0, \quad \text{for all } k. \end{aligned} \qquad (7)$$

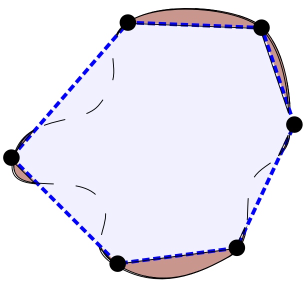


Figure 12: *The space of valid face shapes, represented abstractly as the curved shaded region, is approximated as a convex combination of a number of blendshapes lying on the boundary of the region (black circles). Some regions of the space are not reachable with these blendshapes. This can only be addressed by sculpting blendshapes that lie outside of the valid face space. This is an unnatural task for the modeller.*

These constraints guarantee that the blendshape model lies within the convex hull of the blendshapes. This is a reasonable first assumption, but it is desirable to relax it. By analogy with the convex hull containing a two-dimensional face space, it is likely that targets sufficient to span a broad range of facial expressions must themselves lie outside the valid range of expressions (Fig. 12). Because it is somewhat unnatural to ask an artist to sculpt targets that are slightly beyond the range of plausible expressions, it is often desirable to slightly relax the constraint in equation (7).

Constraining the weights to sum-to-one results in an inconvenient parameterization in which the model has n user parameters for $n-1$ degrees of freedom, and any weight can be expressed as a linear combination of the other weights. In practice it means that the blending weights cannot be modified independently (e.g. using sliders) without violating the constraint. One solution is to normalize the weights after each modification. From the user interface point of view, this has the undesirable consequence that changing a particular weight will cause other weights that were not explicitly altered to change as well. Animators are not novice computer users, however, and can learn to anticipate this behavior.

7.6. Semantic parameterization

The blendshape basis has meaning by construction: blendshape targets have simple and definable functions such as `raise-right-eyebrow`. This allows the effect of particular targets to be predicted and remembered, thereby reducing trial-and-error exploration during animation.

Recent literature in several fields explores the idea that sparse, positive, non-orthogonal, and redundant bases are better able to encode aspects of the meaning of a signal. Examples of this literature include non-negative matrix factorization [LS99], sparse coding for image processing [Ela10], and modeling of biological information processing [OF96].

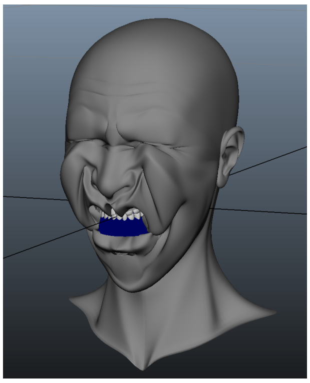


Figure 13: *Blendshapes appear to function as a sparse basis. This figure shows a professionally created model with 45 targets, all set to one. Combinations of several (perhaps up to five or so) targets produce useful expressions, but the combination of many targets produces unusable shapes.*

We note that blendshapes share the qualities of being a non-orthogonal and sparse representation. The blendshape weights are (usually) positive, but the basis is not redundant. A well-constructed blendshape model produces reasonable facial expressions when a few weights (up to five or so) are non-zero, but the models fail when many weights are active (Figure 13). Fig. 14 compares the sparsity of the blendshape encoding to a PCA encoding. The blendshape weights are usually either large or zero, and relatively few weights are active at any point. The PCA representation of the animation has a large number of very small weights. These small weights would be laborious to specify using keyframe animation.

7.7. PCA is not interpretable

While the first few basis vectors discovered by PCA are often interpretable (for example, the first eigenvector typically reflects the jaw-opening motion), the remaining basis vectors are notoriously difficult to interpret. In this section we explain this lack of interpretability in three ways:

- by intuitive argument: a target such as `raise-right-mouth-corner` is obviously not orthogonal to `jaw-open` (the jaw-open motion pulls the mouth corner down slightly).
- by demonstration: Figure 15 shows several eigenvectors from a professionally created facial animation, (visualized

© The Eurographics Association 2014.

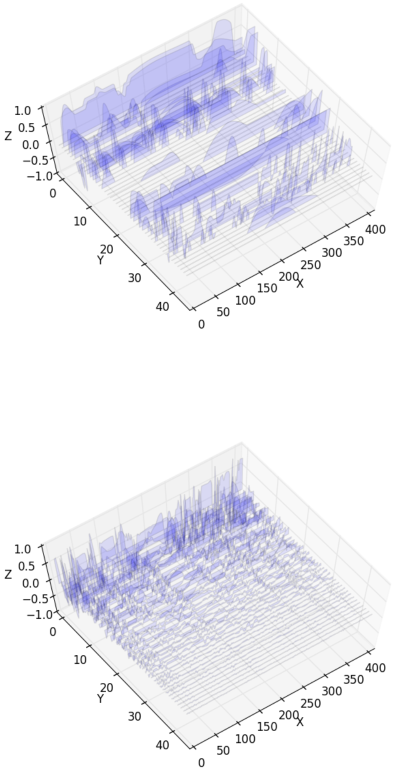


Figure 14: *Comparison of blendshape (top) and PCA coefficients encoding (bottom) of the same 405-frame animation (X-axis) of a 45-dimensional (Y-axis) professionally authored face model. The blendshape coefficients are visibly sparser.*

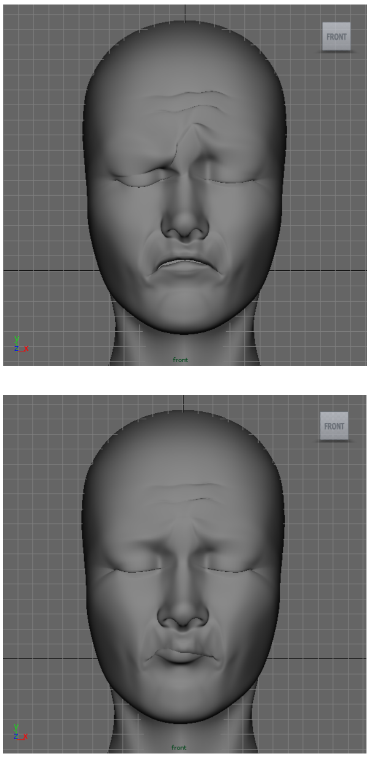


Figure 15: *PCA basis vectors are difficult to interpret and remember. These are the 9th and 10th eigenvectors from a professionally produced facial animation.*

with the mean added as face meshes). The deformations are global and hard to describe (or remember).

- By mathematical arguments:

  1. (An informal version of the Courant nodal theorem): The eigenvector corresponding to the second smallest eigenvalue must be orthogonal to first eigenvector. If the first eigenvector is everywhere positive (or everywhere negative), in order to be orthogonal, the second eigenvector must have both positive and negative regions over the support of the positive part of the first eigenvector (and similarly for regions corresponding to the negative part of the first eigenvector). Thus we see that each eigenvector must have more oscillations than the previous. Note that this argument follows only from the orthogonality of the basis, and thus applies equally to PCA variants such as weighted PCA.

  2. The eigenvectors are linear combinations of all the variables (this is a motivation for sparse PCA schemes). PCA is the orthogonal basis that minimizes the squared reconstruction error. By the "grouping effect" of least squares [ZH], if a group of correlated variables contributes to an eigenvector, their contribution tends to be distributed evenly across all variables.

PCA is also quite weak as a means of characterizing or modeling data (Figure 16). The advantages of PCA may not outweigh the loss of interpretability for some applications.

The PCA technique belongs to the unsupervized learning algorithms category. There has been little research that explore other techniques in this category. An exception is the work done by [CFP03] on extracting a linear model from facial motion capture data. They use Independent Component Analysis (ICA), which tries to extract linear components that are statistically independent, a stronger property than uncorrelated components used by PCA. They show that the extracted components can be categorized in broad motion groups such as speech, emotion, eyelid. The components can then be used for coarse motion editing such as exaggeration.

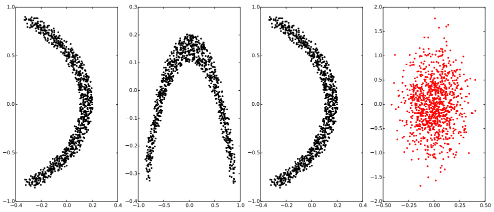


Figure 16: *PCA is a weak "model" of data. From left to right: a synthetic data set, the PCA* coefficients *of this data, the* rotated *PCA coefficients, and random points having the same covariance as the data. While the two eigenvectors and corresponding eigenvalues capture the spread of the data, all the structure ends up in the coefficients. In this two dimensional example the coefficients* $\mathbf{c} = \mathbf{U}^T\mathbf{f}$ *are simply a rotation of the original data points* **f***, since* **U** *is orthogonal.*

7.8. Conversion between blendshape and PCA representations

A blendshape representation can be equated to a PCA model that spans the same space:

$$\mathbf{B}\mathbf{w} + \mathbf{f}_0 = \mathbf{U}\mathbf{c} + \mathbf{e}_0 \tag{8}$$

where **U** and **c** are the PCA eigenvectors and coefficients, and $\mathbf{f}_0$ and $\mathbf{e}_0$ are the neutral face and mean face respectively. The weights can be interconverted as

$$\mathbf{w} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{U}\mathbf{c} + \mathbf{e}_0 - \mathbf{f}_0)$$
$$\mathbf{c} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T(\mathbf{B}\mathbf{w} + \mathbf{f}_0 - \mathbf{e}_0)$$

Note that the matrices here (e.g. $(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{U}$) can be precomputed and are of size $n \times n$. The vectors $(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{e}_0 - \mathbf{f}_0)$ can also be precomputed. Thus converting from weights to coefficients or vice versa is a simple affine transform that can easily be performed at interactive rates on consumer machines. A blendshape software system can thus internally convert operations into a PCA representation if this is advantageous.

7.9. Probability of a blendshape expression

Various applications require or can benefit from knowing the "probability" of a blendshape expression. The probability and norm can be used to identify outliers in tracking, and particularly to regularize the inverse problem in direct manipulation facial editing [ATJ12]. The Gaussian density leads to simple MAP (maximum a posteriori) computation, so this approach is widely used in many applications.

The correspondence of blendshapes and PCA representations (equation 8) gives a simple means to assign a probability to a blendshape expression. The expectation of the square of an individual PCA coefficient is the corresponding eigenvalue:

$$\begin{aligned}\mathbb{E}[c_i^2] &= \mathbb{E}[\mathbf{u}_i^T\mathbf{f}\mathbf{f}^T\mathbf{u}_i] \\ &= \mathbf{u}_i^T\mathbb{E}[\mathbf{f}\mathbf{f}^T]\mathbf{u}_i = \mathbf{u}_i^T\mathbf{C}\mathbf{u}_i \\ &= \mathbf{u}_i^T\lambda_i\mathbf{u}_i \\ &= \lambda_i \quad \text{because } \|\mathbf{u}_i\| = 1\end{aligned}$$

where **f** is a vector representing the face (or other data) with the data mean removed, $\mathbf{u}_i$ is a particular eigenvector and λ_i is the corresponding eigenvalue.

Since the eigenvalues are variances, the multivariate normal density with these variances can be used to assign a probability to a facial expression:

$$P(\mathbf{c}) = \exp\left(\sum_i \frac{c_i^2}{\lambda_i}\right) = \exp\left(-\mathbf{c}^T\Lambda^{-1}\mathbf{c}\right)$$

This also generates a "face norm" $\|\mathbf{f}\|_{\mathcal{B}}$

$$= \mathbf{c}^T\Lambda^{-1}\mathbf{c} = (\mathbf{f}^T\mathbf{U})(\mathbf{U}^T\mathbf{C}^{-1}\mathbf{U})(\mathbf{U}^T\mathbf{f}) = \mathbf{f}^T\mathbf{C}^{-1}\mathbf{f} = \|\mathbf{f}\|_{\mathcal{B}}^2$$

The form $\mathbf{f}^T\mathbf{C}^{-1}\mathbf{f}$ is the multidimensional equivalent of the f^2/σ^2 that appears in the one-dimensional Gaussian $\exp(-f^2/\sigma^2)$.

There is an important but rarely acknowledged issue with assigning a Gaussian probability to face models however [LMAR14]: MAP seeks the mode of the posterior Gaussian. In high dimensions the Gaussian is a heavy tailed distribution, and the mode is an atypical point – the interior of the density has almost no volume, and (contrary to some published statements) typical faces drawn from this density will not lie near the mean (Figure 17).

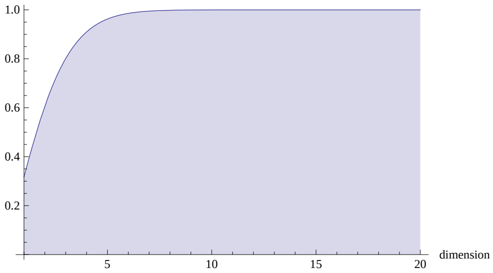


Figure 17: *Probablility that a sample from a unit variance Gaussian lies outside the unit hypersphere for various dimensions.*

7.10. Compressing blendshapes

While the blendshape representation provides compression of an animation, further compression is desirable for animation editing, and is required for games. As an example for discussion, a blendshape model with 1000 targets, each with 10000 vertices represented with four-byte floats, would

© The Eurographics Association 2014.

require 120 megabytes of memory. In the delta blendshape form most targets are localized and are zero at most vertices, so this size can be reduced using a suitable sparse matrix data structure.

While these figures indicate that a reasonably detailed model is easily accommodated in the memory of current processors, there are two reasons for needing additional compression. First, it is desirable (and required in games) that the scene includes the character body and background complete with textures. As well, some scenes may have multiple characters. A more important reason is that the matrix-vector multiply $\mathbf{Bw}$ in (2) is memory-bound on both current CPUs and GPUs.

The obvious approach to compressing a blendshape model is to apply principal component analysis, retaining only the eigenvectors corresponding to the largest eigenvalues. As a rule of thumb, PCA can provide 10:1 or greater compression of many natural signals with little visible change in the signal. PCA does not work as well for blendshape models, however, because blendshapes are already a compressed representation (an animation of any length requires storage of only the n basis vectors, and n weights per frame). In several tests on available models, we found that the compression rates obtainable without introducing visible degradation are as small as 3:1.

While blendshape models are resistant to PCA compression, they nevertheless have considerable structure and smoothness that can be exploited. [SILN11] observe that it is possible to re-order the blendshape matrix $\mathbf{B}$ to expose large low-rank blocks. Placing these in "off diagonal" positions allows application of hierarchically semi-separable (HSS) algorithms [XCGL10]. These approaches produce a hierarchical compressed representation by compressing off-diagonal blocks and then recursively processing the diagonal blocks, and they provide a fast and parallelizable matrix-vector multiplication. Using a HSS representation [SILN11] obtained on the order of 10:1 compression and similar speed increases.

8. Further Applications

8.1. Model transfer

Section 5.3 described algorithms for transferring the motion from one model to another. This subsection describes the related problem of constructing a target model that in some sense is equivalent to a given source model, given only a limited sample of the target expressions such as the neutral face. We term this problem *model transfer.*

Deformation transfer [SP04] is a leading approach for constructing a target model by model transfer. It requires a fully constructed blendshape model for the source, but only a neutral model for the target. This approach first finds the deformation gradient between each triangle of the neutral pose source model $\mathbf{b}_0$ and the corresponding triangle in one of the source blendshape expressions $\mathbf{b}_k, k \geq 1$ (The "deformation gradient" is the Jacobian of the function that deforms the source triangle from its neutral position). Then, given a non-neutral expression on the source model, deformation transfer finds triangles on a target expression so that the target deformation gradient matches the equivalent Jacobian on the source model in a least squares sense. [BSPG06] points out that deformation transfer is a form of Poisson equation.

Since the original deformation transfer does not consider collisions, it may result in self-collisions particularly around the eyelids and lips. [Sai13] inserted virtual triangles into the eye and mouth openings of the mesh to prevent this problem. They also add a new term to the optimization that causes the Laplacian of target mesh to resemble that of the source in areas that are most compressed or stretched, reducing a tendency to crumple in these areas.

[LWP10] is a technique designed specifically for the blendshape model transfer problem. This approach allows the artist to guide the model transfer process by specifying a small number of example expressions and corresponding approximate expected blendshape weights for these expressions. Since only a small number of example expressions are provided, construction of the full target basis is an underdetermined problem. This is solved by using the deformation transfer of the source as a regularization energy in their optimization.

8.2. Blendshape refinement

Often a blendshape model will not exactly match the desired motion. One variant of this problem is when the motion causes the model to take on an expression that reveals undesirable interactions between the blendshape targets. In this case artists can resculpt the model or add corrective combination shapes as discussed in section 3.3. A second form of the problem is when the model matches the motion in a least squares sense but with a large residual. To handle this case [JTDP03] fit the residual with a radial basis function scattered interpolation. [CLK01] used an alternating least squares optimization to solve for positions of the basis vertices corresponding to the markers. This correction was then applied to the remaining vertices using radial basis interpolation. [KSSN11] addresses the refinement problem by augmenting the basis with new targets for frames with high residuals. The correction uses (bi)harmonic interpolation of the tracked displacements [BBA*07].

While some of the previous methods optimize over all frames in a sequence, a focus of recent research is methods that can accomplish on-line refinement of the blendshape basis. [LYYB13] introduced performance driven animation system from live color-depth video stream. The initial blendshapes of the actor's face are created using deformation transfer. Then, additional corrective PCA models refine the

© The Eurographics Association 2014.

actor-specific expressions on the fly using incremental PCA based learning. [BGY*13] presents a system to refine animation curves and produce additional correctives from a set of blendshapes along with 2D features such as markers on the face and contours around eyelids and lips. Every frame is optimized using 2D marker constraints, 3D bundle constraints, and contour constraints. [BWP13] combine a PCA model of identity with a blendshape model of expressions obtained through deformation transfer from a generic template model. Since a particular person's expressions are not exactly captured in this basis, they add a correction in the form of the low-frequency eigenvectors of the graph Laplacian of the face mesh. This correction basis can fit a smooth residual from the blendshape basis while rejecting the noise from the RGB-D camera used in their system.

8.3. Discovery of blendshapes

Creating a realistic blendshape model typically requires sculpting on the order of 100 blendshape targets, and hundreds more shapes if the combination shapes scheme is used (section 3.3). Each target must be designed to capture its intended role such as approximating the activation of a particular muscle, while simultaneously minimizing undesirable interactions with other shapes. This is a labor intensive and iterative effort.

It would be ideal if one could start with dense motion capture of a sufficiently varied performance, and then automatically or semi-automatically convert this into a blendshape model. Doing PCA on the dense motion capture might be a first step toward this goal, however as pointed out earlier, the PCA basis vectors are global and lack the necessary semantics. Given a PCA model $\mathbf{f} = \mathbf{U}\mathbf{c} + \mathbf{m}$ with $\mathbf{U}$ being the eigenvectors and $\mathbf{m}$ the mean shape, the discovery problem can be formulated as finding a "recombination" matrix $\mathbf{R}$ such that the new basis $\mathbf{U}\mathbf{R}$ in an equivalent model $\mathbf{f} = (\mathbf{U}\mathbf{R})(\mathbf{R}^{-1}\mathbf{c}) + \mathbf{n}$ is more sparse [LMN04].

[NVW*13] addresses this blendshape discovery problem by minimizing $||\mathbf{M} - \mathbf{B}\mathbf{W}||^2$ subject to a sparsity penalty on the basis $\mathbf{B}$, where $\mathbf{M}$ is the sequence of scans or motion capture of the face, and $\mathbf{W}$ are the corresponding (unknown) weights. Rather than minimizing $||\mathbf{B}||_1$ to promote sparsity, they use an ℓ_1 norm over the length (ℓ_2 norm) of each vertex. In other words, each vertex in the basis is encouraged to be zero, but if the vertex is not zero then there is no further penalization of its components [BJMO12]. The results outperform PCA, ICA, and several other algorithms and allow intuitive direct manipulation editing.

While [NVW*13] is a significant advance, further developments may be possible on this important problem. It is likely that artists will prefer to guide the blendshape construction rather than relying on a fully automatic process, so an ideal solution must accelerate the artist's process without taking away control.

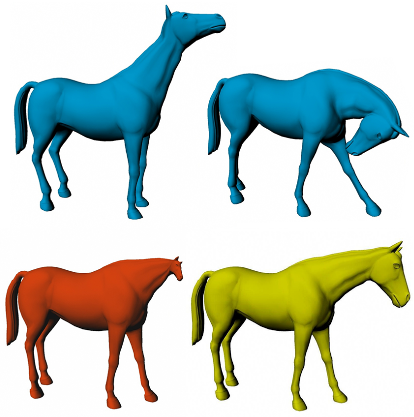


Figure 18: *From* A Blendshape Model that Incorporates Physical Interaction *[MWF*12]. Top row, two meshes to interpolate. Bottom left, linear interpolation. Bottom right, interpolation of edge lengths followed by a mass-spring solve.*

8.4. Alternatives and related techniques

We conclude by briefly mentioning two techniques that accomplish nonlinear blending of target shapes. While these are outside of the industry definition of blendshapes, they point the way toward more powerful techniques. [SZGP05] obtains the deformation gradients of a model in several poses, and blends them using a scheme that interpolates rotation separately from scale. Specifically, they split the Jacobian into rotation and symmetric factors using the polar decomposition and then do linear interpolation in the rotation Lie algebra using the exponential map. The symmetric factor is linearly interpolated directly.

[MWF*12] interprets the original target meshes as a mass spring model and linearly blends edge lengths rather than geometry. This simple approach is able to produce reasonable rotational motion (Figure 18) as well as contact and collision effects.

9. Conclusion

"Blendshapes" are at present the leading approach to realistic facial animation. While most algorithms in graphics industry software and practice can be traced back to original research publications, blendshapes are unusual in that both the original idea and some recent developments [Osi07] originated outside of academic forums. Despite its popularity, the technique has both unresolved limitations and associated open problems.

Acknowledgements

We thank Hiroki Itokazu and Bret St. Clair for model preparation, and Lance Williams, Fred Parke, Craig Reynolds, and

© The Eurographics Association 2014.

Thad Beier for information on the early history of blendshapes.

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph. 22*, 3 (July 2003), 587–594. 5

[ARV07] AMBERG B., ROMDHANI S., VETTER T.: Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8. 5

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (July 2005), 408–416. 5

[ATJ12] ANJYO K., TODO H., J.: A practical approach to direct manipulation blendshapes. *J. Graphics Tools 16*, 3 (2012), 160–176. 9, 15

[BBA*07] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M.: Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. 26*, 3 (2007), 33. 1, 16

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating faces in images and video. *Computer Graphics Forum(Proceedings of Eurographics 2003) 22*, 3 (2003). 6

[Bei05] BEIER T.: personal communication. Hammerhead Productions, 2005. 3

[Ber87] BERGERON P.: 3-d character animation on the symbolics system. *SIGGRAPH Course Notes: 3-D Character Animation by Computer* (July 1987). 3

[BGY*13] BHAT K. S., GOLDENTHAL R., YE Y., MALLET R., KOPERWAS M.: High fidelity facial animation capture and retargeting with contours. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2013), SCA '13, ACM, pp. 7–14. 6, 8, 17

[BHPS10] BRADLEY D., HEIDRICH W., POPA T., SHEFFER A.: High resolution passive facial performance capture. *ACM Trans. Graph. 29*, 4 (July 2010), 41:1–41:10. 1, 6

[BJMO12] BACH F., JENATTON R., MAIRAL J., OBOZINSKI G.: Convex optimization with sparsity-inducing norms. In *Optimization for Machine Learning*, Sra S., Nowozin S., Wright S., (Eds.). MIT Press, 2012. 17

[BLB*08] BICKEL B., LANG M., BOTSCH M., OTADUY M. A., GROSS M.: Pose-space animation and transfer of facial details. In *SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2008), Eurographics Association, pp. 57–66. 1, 5

[BPL*03] BORSHUKOV G., PIPONI D., LARSEN O., LEWIS J. P., TEMPELAAR-LIETZ C.: Universal capture: image-based facial animation for "the matrix reloaded". In *Proceedings of the SIGGRAPH 2003 conference on Sketches & applications* (2003), ACM Press, pp. 1–1. 1, 6

[BSPG06] BOTSCH M., SUMNER R., PAULY M., GROSS M.: Deformation transfer for detail-preserving surface editing. In *Proceedings of Vision, Modeling, and Visualization (VMV)* (2006), pp. 357–364. 16

[BV99] BLANZ T., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proceedings of ACM SIGGRAPH* (Aug. 1999), ACM SIGGRAPH, pp. 187–194. 2, 5

[BWP13] BOUAZIZ S., WANG Y., PAULY M.: Online modeling for realtime facial animation. *ACM Trans. Graph. 32*, 4 (July 2013), 40:1–40:10. 6, 17

[Caf98] CAFLISCH R.: Monte carlo and quasi-monte carlo methods, 1998. 10

[CDB02] CHUANG E., DESHPANDE H., BREGLER C.: Facial expression space learning. In *Proceedings of Pacific Graphics* (2002). 6

[CFP03] CAO Y., FALOUTSOS P., PIGHIN F.: Unsupervised learning for speech motion editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 225–231. 14

[CK01] CHOE B., KO H.-S.: Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *Computer Animation and Social Agents (CASA)* (2001). 6

[CLK01] CHOE B., LEE H., KO H.-S.: Performance-driven muscle-based facial animation. *J. Visualization and Computer Animation 12*, 2 (2001), 67–79. 6, 16

[CWLZ13] CAO C., WENG Y., LIN S., ZHOU K.: 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July 2013), 41:1–41:10. 6

[CXH03] CHAI J.-X., XIAO J., HODGINS J.: Vision-based control of 3d facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 193–206. 6

[DBLN06] DENG Z., BAILENSON J., LEWIS J. P., NEUMANN U.: Perceiving visual emotions with speech. In *Proc. of the 6th Int'l Conf. on Intelligent Virtual Agents (IVA) 2006* (August 2006), pp. 107–120. 3

[DCFN06] DENG Z., CHIANG P. Y., FOX P., NEUMANN U.: Animating blendshape faces by cross-mapping motion capture data. In *Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)* (Mar. 2006), pp. 43–48. 6, 12

[DN07] DENG Z., NOH J.: Computer facial animation: A survey. In *Data-Driven 3D Facial Animation* (2007), Springer, pp. 1–28. 1

[Duc76] DUCHON J.: Math. comp. *Interpolation des Fonctions de Deux Variables Suivant le Principe de la Flexion des Plaques Minces 10* (1976), 5–12. 10

[Ela10] ELAD M.: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer Publishing Company, Incorporated, 2010. 13

[Els90] ELSON M.: "Displacement" facial animation techniques. *SIGGRAPH Course Notes: State of the Art in Facial Animation* (1990). 3

[ER97] EKMAN P., ROSENBERG E.: *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. Series in affective science. Oxford University Press, 1997. 12

[EYE] EYETRONICS: Shapesnatcher. http://www.eyetronics.com. 1

[Flu11] FLUECKIGER B.: Computer-generated characters in *Avatar* and *Benjamin Button*. In: H. Segeberg (ed.) *Digitalitat und Kino*. Translation from German by B. Letzler, 2011. 1

[Gar] GARCKE J.: Sparse grid tutorial. http://www.math.tu-berlin.de/ garcke/paper/sparseGridTutorial.pdf. 10

[GGW*98] GUENTER B., GRIMM C., WOOD D., MALVAR H., PIGHIN F.: Making faces. In *SIGGRAPH 98 Conference Proceedings* (July 1998), ACM SIGGRAPH, pp. 55–66. 1, 6

[HIWZ05] HYNEMAN W., ITOKAZU H., WILLIAMS L., ZHAO X.: Human face project. In *ACM SIGGRAPH 2005 Courses* (New York, NY, USA, 2005), SIGGRAPH '05, ACM. 7

[HTF09] HASTIE T., TIBSHIRANI R., FRIEDMAN J.: *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag, New York, NY, 2009. 10

[JTDP03] JOSHI P., TIEN W. C., DESBRUN M., PIGHIN F.: Learning controls for blend shape based realistic facial animation. *Eurographics/SIGGRAPH Symposium on Computer Animation (SCA)* (2003). 6, 9, 12, 16

[Kle89] KLEISER J.: A fast, efficient, accurate way to represent the human face. In *SIGGRAPH '89 Course Notes 22: State of the Art in Facial Animation* (1989). 3, 12

[KM04] KURIHARA T., MIYATA N.: Modeling deformable human hands from medical images. In *Proceedings of the 2004 ACM SIGGRAPH Symposium on Computer Animation (SCA-04)* (2004), pp. 357–366. 11

[KMML10] KOMOROWSKI D., MELAPUDI V., MORTILLARO D., LEE G. S.: A hybrid approach to facial rigging. In *ACM SIGGRAPH ASIA 2010 Sketches* (New York, NY, USA, 2010), SA '10, ACM, pp. 42:1–42:2. 1

[KSSN11] KIM P. H., SEOL Y., SONG J., NOH J.: Facial Retargeting by Adding Supplemental Blendshapes. In *Pacific Graphics* (2011), pp. 89–92. 16

[LA09] LEWIS J., ANJYO K.: Identifying salient points. In *SIGGRAPH Asia short papers* (2009). 8

[LA10] LEWIS J., ANJYO K.: Direct manipulation blendshapes. *Computer Graphics and Applications (special issue: Digital Human Faces) 30*, 4 (2010), 42–50. 8, 9

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: A unified approach to shape interpolation a nd skeleton-driven deformation. In *Siggraph 2000, Computer Graphics Proceedings* (2000), Akeley K., (Ed.), Annual Conference Series, ACM Press / ACM SIGGRAPH / Addison Wesley Longman. 1

[LCXS09] LAU M., CHAI J., XU Y.-Q., SHUM H.-Y.: Face poser: Interactive modeling of 3d facial expressions using facial priors. *ACM Trans. Graph. 29*, 1 (2009), 1–17. 9

[LD08] LI Q., DENG Z.: Orthogonal blendshape based editing system for facial motion capture data. *IEEE Computer Graphics and Applications (CG&A) 28*, 6 (Nov. 2008), 76–82. 9, 12

[LDSS99] LEE A., DOBKIN D., SWELDENS W., SCHRÖDER P.: Multiresolution mesh morphing. In *Proceedings of SIGGRAPH 99* (August 1999), pp. 343–350. 5

[LH09] LEE G. S., HANNER F.: Practical experiences with pose space deformation. In *ACM SIGGRAPH ASIA 2009 Sketches* (New York, NY, USA, 2009), SIGGRAPH ASIA '09, ACM, p. 43. 1

[LMAR14] LEWIS J., MO Z., ANJYO K., RHEE T.: Probable and improbable faces. In *Mathematical Progress in Expressive Image Synthesis I* (2014), Springer. 15

[LMDN05] LEWIS J., MOOSER J., DENG Z., NEUMANN U.: Reducing blendshape interference by selected motion attenuation. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM Press, pp. 25–29. 11

[LMN04] LEWIS J., MO Z., , NEUMANN U.: Ripple-free local bases by design. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (2004), pp. 684–688. 17

[LPA10] LEWIS J., PIGHIN F., ANJYO K.: Scattered data interpolation for computer graphics. SIGGRAPH Asia Course, http://portal.acm.org, 2010. 10

[LS99] LEE D., SEUNG H. S.: Learning the parts of objects by non-negative matrix factorization. *Nature*, 401 (1999), 788–791. 13

[LWP10] LI H., WEISE T., PAULY M.: Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010) 29*, 3 (July 2010). 7, 16

[LYYB13] LI H., YU J., YE Y., BREGLER C.: Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics 32*, 4 (July 2013). 6, 16

[MA07] MEYER M., ANDERSON J.: Key point subspace acceleration and soft caching. *ACM Trans. Graph. 26*, 3 (2007), 74. 8

[MAO*11] MIRANDA J. C., ALVAREZ X., ORVALHO J. A., GUTIERREZ D., SOUSA A. A., ORVALHO V.: Sketch express: Facial expressions made easy. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling* (New York, NY, USA, 2011), SBIM '11, ACM, pp. 87–94. 10

[MBL12] MA W.-C., BARBATI M., LEWIS J. P.: A facial composite editor for blendshape characters. In *Proceedings of the Digital Production Symposium* (New York, NY, USA, 2012), DigiPro '12, ACM, pp. 21–26. 6

[MJC*08] MA W.-C., JONES A., CHIANG J.-Y., HAWKINS T., FREDERIKSEN S., PEERS P., VUKOVIC M., OUHYOUNG M., DEBEVEC P.: Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph. 27*, 5 (Dec. 2008), 121:1–121:10. 5

[Mov09] MOVA LLC: Contour reality capture. *http://www.mova.com* (2009). 1

[MWF*12] MA W.-C., WANG Y.-H., FYFFE G., CHEN B.-Y., DEBEVEC P.: A blendshape model that incorporates physical interaction. *Computer Animation and Virtual Worlds 23*, 3-4 (2012), 235–243. 17

[NN99] NOH J., NEUMANN U.: A survey of facial modeling and animation techniques. In *USC Technical Report 99-705* (1999), U. Southern California. 1

[NN01] NOH J., NEUMANN U.: Expression cloning. In *SIGGRAPH 2001, Computer Graphics Proceedings* (2001), Fiume E., (Ed.), ACM Press / ACM SIGGRAPH, pp. 277–288. 6

[NVW*13] NEUMANN T., VARANASI K., WENGER S., WACKER M., MAGNOR M., THEOBALT C.: Sparse localized deformation components. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia) 32*, 6 (2013). 9, 12, 17

[OBP*12] ORVALHO V., BASTOS P., PARKE F., OLIVEIRA B., , ALVAREZ X.: A facial rigging survey: State of the art report. In *Eurographics* (2012), pp. 183–204. 1, 5

[OF96] OLSHAUSEN B. A., FIELD D. J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature 13 June*, 381 (1996), 607–609. 13

[Osi07] OSIPA J.: *Stop Staring: Facial Modeling and Animation Done Right, 2nd Ed.* Sybex, 2007. 4, 5, 17

[Par] PARKE F.: personal communication. Texas A&M University. 3

[Par72] PARKE F.: Computer generated animation of faces. *Proceedings ACM annual conference.* (Aug. 1972). 3

[Par74] PARKE F.: *A parametric model for human faces*. PhD thesis, University of Utah, Salt Lake City, Utah, Dec. 1974. UTEC-CSc-75-047. 1, 3

[Par91] PARKE F.: Control parametrization for facial animation. In *Computer Animation 91*, Thalmann N. M., Thalmann D., (Eds.). Springer-Verlag, Tokyo, 1991, pp. 3–14. 1

[PHL*98] PIGHIN F., HECKER J., LISCHINSKI D., SZELISKI R., SALESIN D.: Synthesizing realistic facial expressions from photographs. In *SIGGRAPH 98 Conference Proceedings* (July 1998), ACM SIGGRAPH, pp. 75–84. 5, 9

[PL05] PIGHIN F., LEWIS J.: Digital face cloning. SIGGRAPH Course, http://portal.acm.org, 2005. 6

[PL06] PIGHIN F., LEWIS J.: Performance-driven facial animation. SIGGRAPH Course, http://portal.acm.org, 2006. 6

[PSS99] PIGHIN F., SZELISKI R., SALESIN D.: Resynthesizing facial animation through 3d model-based tracking. In *Proceedings, International Conference on Computer Vision* (1999). 6

[PW08] PARKE F. I., WATERS K.: *Computer Facial Animation*. AK Peters Ltd, 2008. 1

[Rai04] RAITT B.: The making of Gollum. Presentation at U. Southern California Institute for Creative Technologies's *Frontiers of Facial Animation* Workshop, August 2004. 2, 5

[RHKK11] RHEE T., HWANG Y., KIM J. D., KIM C.: Real-time facial animation from live video tracking. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 215–224. 1, 6

[Sai13] SAITO J.: Smooth contact-aware facial blendshapes transfer. In *Proceedings of the Symposium on Digital Production* (New York, NY, USA, 2013), DigiPro '13, ACM, pp. 7–12. 16

[SE09] SCHNEIDER D., EISERT P.: Fast nonrigid mesh registration with a data-driven deformation prior. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on* (Sept 2009), pp. 304–311. 5

[SG06] SAGAR M., GROSSMAN R.: Facial performance capture and expressive translation for King Kong. In *SIGGRAPH 2006 Sketches* (2006). 1, 7, 12

[SILN11] SEO J., IRVING G., LEWIS J. P., NOH J.: Compression and direct manipulation of complex blendshape models. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 164:1–164:10. 9, 16

[Sin03] SINGER G.: The Two Towers: Face to face with Gollum. Animation World Network, March 2003. 1

[SLS*12] SEOL Y., LEWIS J., SEO J., CHOI B., ANJYO K., NOH J.: Spacetime expression cloning for blendshapes. *ACM Trans. Graph. 31*, 2 (Apr. 2012), 14:1–14:12. 7

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (July 2005), 417–425. 1

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 399–405. 6, 16

[SSK*11] SEOL Y., SEO J., KIM P. H., LEWIS J., NOH J.: Artist friendly facial animation retargeting. *ACM Transactions on Graphics (TOG) 30*, 6 (2011), 162. 8

[SSK*12] SEOL Y., SEO J., KIM P. H., LEWIS J. P., NOH J.: Weighted pose space editing for facial animation. *The Visual Computer 28*, 3 (2012), 319–327. 11

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *ACM Trans. Graph. 24*, 3 (July 2005), 488–495. 17

[TDlTM11] TENA J. R., DE LA TORRE F., MATTHEWS I.: Interactive region-based linear 3d face models. In *ACM SIGGRAPH 2011 papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 76:1–76:10. 6

[Tic09] TICKOO S.: *Autodesk Maya 2010: A Comprehensive Guide*. CADCIM Technologies, 2009. 1, 4

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. 426–433. 7

[Ver] VERNON C.: Introduction to Python scripting for Maya artists. chadvernon.com. 4

[WAT*11] WILSON C. A., ALEXANDER O., TUNWATTANAPONG B., PEERS P., GHOSH A., BUSCH J., HARTHOLT A., DEBEVEC P.: Facial cartography: interactive high-resolution scan correspondence. In *ACM SIGGRAPH 2011 Talks* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 8:1–8:1. 5

[WBLP11] WEISE T., BOUAZIZ S., LI H., PAULY M.: Real-time performance-based facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011) 30*, 4 (July 2011). 6

[Wil90] WILLIAMS L.: Performance-driven facial animation. In *SIGGRAPH 90 Conference Proceedings* (Aug. 1990), vol. 24, pp. 235–242. 6

[Wil01] WILLIAMS L.: personal communication. Disney Feature Animation, 2001. 7

[WLGP09] WEISE T., LI H., GOOL L. V., PAULY M.: Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)* (ETH Zurich, August 2009), Eurographics Association. 6

[XCGL10] XIA J., CHANDRASEKARAN S., GU M., LI X. S.: Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications 17*, 6 (2010), 953–976. 16

[YN03] YAMANE K., NAKAMURA Y.: Natural motion animation through constraining and deconstraining at will. *IEEE Tran. Visualization and Computer Graphics 9*, 3 (2003), 352–360. 8

[ZH] ZOU H., HASTIE T.: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*, 2, 301–320. 14

[Zha01] ZHAO X.: personal communication. Disney Feature Animation, 2001. 3

[ZLG*06] ZHANG Q., LIU Z., GUO B., TERZOPOULOS D., SHUM H.-Y.: Geometry-driven photorealistic facial expression synthesis. *IEEE Transactions on Visualization and Computer Graphics 12*, 1 (2006), 48–60. 8, 9

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH* (New York, NY, USA, 2004), ACM, pp. 548–558. 1, 8, 9

© The Eurographics Association 2014.