
Blendshape Facial Animation

Ken Anjyo

Abstract

Blendshapes are a standard approach for making expressive facial animations in the digital production industry. The blendshape model is represented as a linear weighted sum of the target faces, which exemplify user-defined facial expressions or approximate facial muscle actions. Blendshapes are therefore quite popular because of their simplicity, expressiveness, and interpretability. For example, unlike generic mesh editing tools, blendshapes approximate a space of valid facial expressions.

This article provides the basic concepts and technical development of the blendshape model. First, we briefly describe a general face rig framework and thereafter introduce the concept of blendshapes as an established face rigging approach. Next, we illustrate how to use this model in animation practice, while clarifying the mathematical framework for blendshapes. We also demonstrate a few technical applications developed in the blendshape framework.

Keywords

Computer facial animation • Face rig • Blendshapes • Retarget • Deformer • Facial motion capture • Performance capture

Contents

State of the Art	2146
Blendshape Applications	2147
Blendshape Practice	2148
Formulation	2148
Examples and Practice	2150

K. Anjyo (✉)
OLM Digital, Setagaya, Tokyo, Japan
e-mail: anjyo@acm.org

Techniques for Efficient Animation Production	2151
Direct Manipulation	2151
Use of PCA Models	2152
Blendshape Creation, Retargeting, and Transfer	2153
Conclusion	2154
Cross-References	2154
References	2155

State of the Art

Digital characters now appear not only in films and video games but also in various digital contents. In particular, facial animation of a digital character should then convey emotions to it, which plays a crucial role for visual storytelling. This requires a digital character animation process as well as its face rigging process (i.e., the setup process) to be very intensive and laborious.

In this article, we define *face rig* as the pair of a deformer and its user interface (manipulator). The deformer means a mathematical model of deforming a face model's geometry for making animation. The user interface provides animators a toolset of manipulating the face model, based on the deformer. In a production workplace, however, they usually use several deformers at a time, so that the user interface in practice should be more complicated, yet sophisticated, rather than the user interface that we will mention in later sections for blendshapes.

A variety of the face rig approaches have been developed. Physics-based models provide the rigorous and natural approaches, having several applications not only in the digital production industry but also in medical sciences, including surgery simulations. The physics-based approaches for computer graphic applications approximate the mechanical properties of the face, such as skin layers, muscles, fatty tissues, bones, etc. Although the physics-based methods may be powerful in making realistic facial animations, artists are then required to have a certain amount of knowledge and experiences regarding background physics. This is not an easy task.

On one hand, several commercial 3D CG packages provide proprietary face rig approaches, such as “cluster deformers” (see Tickoo (2009)), which allow the artist to specify the motion space using a painting operation for making 3D faces at key frames.

The blendshapes offer a completely different face rig approach. A blendshape model generates face geometry as a linear combination of a number of face poses, each of which is called a blendshape target. These targets typically mean individual facial expressions, shapes that approximate facial muscle actions or FACS (Facial Action Coding System (Ekman and Friesen 1978)) motions. These targets are predefined (designed) by the artist. The blendshapes are therefore parameterized with the weights of the targets, which gives an intuitive and simple way for the artist to make animation. The interface is called *sliders* and used to control the weights. Figure 1 presents such a slider interface example and a simple editing result for a blendshape model.

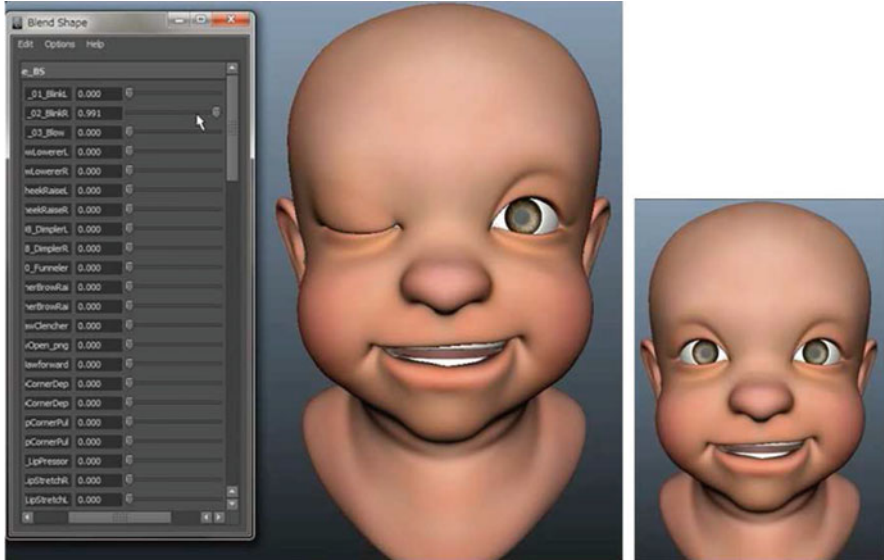


Fig. 1 Blendshapes user interface example. *Left*: The slider box and a 3D face model under editing, where the slider box gives a partial view of the blendshape sliders. This is because, in general, the number of sliders is too large to see all sliders at a time. Instead a desired slider can be reached by scrolling the slider box. The 3D face model shows an edited result with the slider operation for right eye blink, *right*: the face model before the slider operation

The use of motion capture data has become a common approach to make animation of a digital character. As is well known, the original development of motion capture techniques was driven by the needs of life science community, where the techniques are mainly used for the analysis on a subject’s movement. In the digital production industry, facial motion capture data may be used as an input for the synthesis of realistic animations. The original data will then be converted to a digital face model and edited to obtain desired facial animations. Some of the face rig techniques are therefore indispensable in the converting (retargeting) and editing processes.

Blendshape Applications

As mentioned earlier, several face rig techniques are used together for practical situations. Even when more sophisticated approaches to facial modeling are used, blendshapes are often employed as a base layer over which physically based or functional (parameterized) deformations are layered.

In digital production studios and video game companies, they need to develop a sophisticated system that should fully support the artists for efficient and high-quality production of visual effects and character animation. The role of blendshape

techniques may therefore be a small portion of the system, but is still crucial. Here we briefly describe a few state-of-the-art applications that use blendshape techniques:

- **Voodoo.** This system has been developed in Rhythm & Hues Studios over years, which deals mainly with animation, rigging, matchmove, crowds, fur grooming, and computer vision (see Fxguide (2014)). The system provides several prodigious face rigging tools using blendshapes. For example, many great shots in the film *Life of Pi* in 2012 were created with this system.
- **Fez.** This is the facial animation system developed in ILM (Bhat et al. 2013; Cantwell et al. 2016; CGW 2014), which involves an FACS implementation using blendshape techniques. It has contributed to recent films, such as *Warcraft* and *Teenage Mutant Ninja Turtles*, in 2016.
- **Face Plus.** This is a plug-in for Unity, which is a cross-platform game engine. This plug-in enables us to construct a facial capture and animation system using a web camera (see Mixamo (2013) for details). Based on the blendshape character model created by an artist, the system provides real-time facial animation of the character.

In the following sections, we describe the basic practice and mathematical background of the blendshape model.

Blendshape Practice

The term “blendshapes” was introduced in the computer graphics industry, and we follow the 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 (more roughly) as shapes or blendshapes. The corresponding weights are often called sliders, since this is how they appear in the user interface (as shown in Fig. 1). Creating a blendshape facial animation thus requires specifying weights for each frame of the animation, which has traditionally been achieved with key frame animation or by motion capture.

In the above discussion, we use a basic mathematical term “vectors.” This section starts with explaining what the vectors mean in making 3D facial models and animations. We then illustrate how to use the blendshapes in practice.

Formulation

We represent the face model as a column vector \mathbf{f} containing all the model vertex coordinates in some order that is arbitrary (such as $xyzxyzxyz$, or alternately $xxxyyyzzz$) but consistent across the individual blendshapes. For example, let us

consider the face model composed of $n = 100$ blendshapes, each having $p = 1000$ vertices, with each vertex having three components x, y, z . Similarly, we denote the blendshape targets as vectors \mathbf{b}_k , so the blendshape model is represented as

$$\mathbf{f} = \sum_{k=0}^n w_k \mathbf{b}_k, \quad (1)$$

where \mathbf{f} is the resulting face, in the form of a $m = 30,000 \times 1$ vector ($m = 3p$); the individual blendshapes $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ are $30,000 \times 1$ vectors; and w_k denotes the weight for \mathbf{b}_k ($1 \leq k \leq n$). We then put \mathbf{b}_0 as the neutral face. Blendshapes can therefore be considered simply adding vectors.

Equation (1) may be referred to as the global or “whole-face” blendshape approach. The carefully sculpted blendshape targets appeared in Eq. (1) then serve as interpretable controls; the span of these targets strictly defines the valid range of expressions for the modeled face. These characteristics differentiate the blendshape approach from those that involve linear combinations of uninterpretable shapes (see a later section) or algorithmically recombine the target shapes using a method other than that in Eq. (1). In particular, from an artist’s point of view, the interpretability of the blendshape basis is a definitive feature of the approach.

In the whole-face approach, scaling all the weights by a multiplier causes the whole head to scale, while scaling of the head is more conveniently handled with a separate transformation. To eliminate undesired scaling, the weights in Eq. 1 may be constrained to sum to one. Additionally the weights can be constrained to the interval $[0,1]$ in practice.

In the local or “delta” blendshape formulation, one face model \mathbf{b}_0 (typically the resting face expression) is designated as the neutral face shape, while the remaining targets $\mathbf{b}_k (1 \leq k \leq n)$ in Eq. (1) are replaced with the difference $\mathbf{b}_k - \mathbf{b}_0$ between the k -th face target and the neutral face:

$$\mathbf{f} = \mathbf{b}_0 + \sum_{k=1}^n w_k (\mathbf{b}_k - \mathbf{b}_0). \quad (2)$$

Or, if we use matrix notation, Eq. (2) can be expressed as:

$$\mathbf{f} = \mathbf{B}\mathbf{w} + \mathbf{b}_0, \quad (3)$$

where \mathbf{B} is an $m \times n$ matrix having $\mathbf{b}_k - \mathbf{b}_0$ as the k -th column vector, and $\mathbf{w} = (w_1, w_2, \dots, w_n)^T$ is the weight vector.

In this formulation, the weights are conventionally limited to the range $[0,1]$, while there are exceptions to this convention. For example, the Maya blendshape interface allows the $[0,1]$ limits to be overridden by the artist if needed. If the difference between a particular blendshape \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.



Fig. 2 Target face examples. *From left:* neutral, smile, disaffected, and sad

The delta blendshape formulation is used in popular packages such as Maya (see Tickoo (2009)), and our discussion will assume this variant if not otherwise specified. Many comments apply equally (or with straightforward conversion) to the whole-face variant.

Examples and Practice

Next, we show a simple example of the blendshape model, which has 50 target faces. The facial expressions in Fig. 1 were also made with this simple model. A few target shapes of the model are demonstrated in Fig. 2, where the leftmost image shows its neutral face. Using the 50 target shapes, the blendshape model provides a mixture of such targets.

As mentioned above, the blendshape model is conceptually simple and intuitive. Nevertheless, professional use of this model further requires a large and labor-intensive effort of the artists, some of which are listed as follows:

- Target shape construction
 - To express a complete range of realistic expressions, digital modelers often have to create large libraries of blendshape targets. For example, the character of Gollum in *The Lord of the Rings* had 946 targets (Raitt 2004). Generating a reasonably detailed model can be as much as a year of work for a skilled modeler, involving many iterations of refinement.
 - A skilled digital artist can deform a base mesh into the different shapes needed to cover the desired range of expressions. Alternatively, the blendshapes 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.
- Slider control (see Fig. 1)
 - To skillfully and efficiently use the targets, animators need to memorize the function of 50 to 100 commonly used sliders. Then locating a desired slider isn't immediate.

- A substantial number of sliders are needed for high-quality facial animation. Therefore the complete set of sliders does not fit on the computer display.
- Animation editing
 - As a traditional way, blendshapes have been animated by key frame animation of the weights. Commercial packages provide spline curve interpolation of the weights and allow the tangents to be specified at key frames.
 - Performance-driven facial animation is an alternative way to make animation. Since blendshapes are the common approach for realistic facial models, blendshapes and performance-driven animation are frequently used together (see section “[Use of PCA Models](#),” for instance). We then may need an additional process where the motion captured from a real face is “retargeted” to a 3D face model.

Techniques for Efficient Animation Production

In previous sections, we have shown that the blendshapes are a conceptually simple, common, yet laborious facial animation approach. Therefore a number of developments have been made to greatly improve efficiency in making blendshape facial animation. However, in this section, let us restrict ourselves to describe only a few of our work, while we also mention some techniques related to blendshapes and facial animation. To know more about the mathematical aspect of blendshape algorithms, we would recommend referring to the survey (Lewis et al. 2014).

Direct Manipulation

In general, interfaces should provide both direct manipulation and editing of underlying parameters. While direct manipulation usually provides more natural and efficient results, parameter editing can be more exact and reproducible. Artists might therefore prefer it in some cases.

While inverse kinematic approaches to posing human figures have been used for many years, analogous inverse or direct manipulation approaches for posing faces and setting key frames have emerged quite recently. In these approaches, the artist directly moves points on the face surface model, and the software must solve for the underlying weights or parameters that best reproduce that motion, rather than tuning the underlying parameters.

Here we consider the cases where the number of sliders is considerably large (i.e., well over 100) for a professional use of the blendshape model. Introducing a direct manipulation approach would then be a legitimate requirement. To achieve this, we solve the inverse problem of finding the weights for given point movements and constraints.

In Lewis and Anjyo (2010), this problem is regularized by considering the fact that facial pose changes are proportional to slider position changes. The resulting approach is easy to implement and can cope with existing blendshape models.

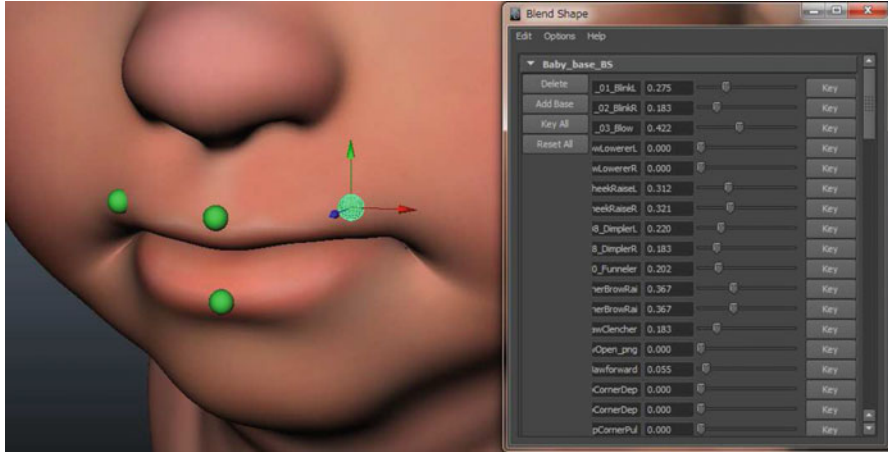


Fig. 3 Example of direct manipulation interface for blendshapes

Figure 3 shows such a direct manipulation interface example, where selecting a point on the face model surface creates a manipulator object termed a pin, and the pins can be dragged into desired positions. According to the pin and drag operations, the system solves for the slider values (the right panel in Fig. 3) for the face to best match the pinned positions. It should then be noted that the direct manipulation developed in Lewis and Anjyo (2010) can interoperate with the traditional parameter-based key frame editing. As demonstrated in Lewis and Anjyo (2010), both direct manipulation and parameter editing are indispensable for blendshape animation practice. There are several extensions of the direct manipulation approach. For instance, a direct manipulation system suitable for use in animation production has been demonstrated in Seo et al. (2011), including treatment of combination blendshapes and non-blendshape deformers. Another extension in Anjyo et al. (2012) describes a direct manipulation system that allows more efficient edits using a simple prior learned from facial motion capture.

Use of PCA Models

In performance-driven facial animation, the motion of human actor is used to derive the face model. Whereas face tracking is a key technology for the performance-driven approaches, this article focuses on performance capture methods that drive a face rig. The performance capture methods mostly use PCA basis or blendshape basis.

We use principal component analysis (PCA) to obtain a PCA model for the given database of facial expression examples. As usual each element of the database is represented as an $m \times 1$ vector \mathbf{x} . Let \mathbf{U} be an $m \times r$ matrix consisting of the r eigenvectors corresponding to the largest eigenvalues of the data covariance matrix. The PCA model is then given as:

$$\mathbf{f} = \mathbf{U}\mathbf{c} + \mathbf{e}_0, \quad (4)$$

where the vector \mathbf{c} means the coefficients of those eigenvectors and \mathbf{e}_0 denotes the mean vector of all elements \mathbf{x} in the database. Since we usually have $r \ll m$, the PCA model gives a good low-dimensional representation of the facial models \mathbf{x} . This also leads us to solutions to statistical estimation problems in a maximum a posteriori (MAP) framework. For example, in Lau et al. (2009), direct dragging and stroke-based expression editing are developed in this framework to find an appropriate \mathbf{c} in Eq. (4).

The PCA approaches are useful if the face model is manipulated only with direct manipulation. Professional animation may also require slider operations, so that the underlying basis should be of blendshapes, rather than PCA representation. This is due to the lack of interpretability of the PCA basis (Lewis and Anjyo 2010).

A blendshape representation (3) can be equated to a PCA model (4) that spans the same space:

$$\mathbf{B}\mathbf{w} + \mathbf{b}_0 = \mathbf{U}\mathbf{c} + \mathbf{e}_0. \quad (5)$$

We know from Eq. (5) that the weight vector \mathbf{w} and the coefficient vector \mathbf{c} can be interconverted:

$$\mathbf{w} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{U}\mathbf{c} + \mathbf{e}_0 - \mathbf{b}_0) \quad (6)$$

$$\mathbf{c} = \mathbf{U}^T(\mathbf{B}\mathbf{w} + \mathbf{b}_0 - \mathbf{e}_0), \quad (7)$$

where we use the fact that $\mathbf{U}^T\mathbf{U}$ is an $r \times r$ unit matrix in deriving the second Eq. (7). We note that the matrices and vectors in Eqs. (6) and (7), such as $(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{U}$ and $(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{e}_0 - \mathbf{b}_0)$, can be precomputed. Converting from weights to coefficients or vice versa is thus a simple affine transform that can easily be done at interactive rates. This will provide us a useful direct manipulation method for a PCA model, if the model can also be represented with a blendshape model.

Blendshape Creation, Retargeting, and Transfer

Creating a blendshape model for professional animation requires sculpting on the order of 100 blendshape targets and adding hundred more shapes in several ways (see Lewis et al. (2014), for instance). Ideally, the use of dense motion capture of a sufficiently varied performance should contribute to efficiently create such a large number of blendshape targets. To achieve this, several approaches have been proposed, including a PCA-based approach (Lewis et al. 2004) and a sparse matrix decomposition method (Neumann et al. 2013).

Expression cloning approaches (Noh and Neumann (2001); Sumner and Popović (2004), for instance) are developed for retargeting the motion from one facial model (the “source”) to drive a face (the “target”) with significantly different proportions.

The expression cloning problem was posed in Noh and Neumann (2001), where the solution was given as a mapping by finding corresponding pairs of points on the source and target faces using face-specific heuristics. The early expression cloning algorithms do not consider adapting the temporal dynamics of the motion to the target, which means that they work well if the source and target are of similar proportions. The movement matching principle in Seol et al. (2012) provides an expression cloning algorithm that can cope with the temporal dynamics of face movement by solving a space-time Poisson equation for the target blendshape motion.

Relating to expression cloning, we also mention *model transfer* briefly. This is the case where the source is a fully constructed blendshape model and the target consists of only a neutral face (or a few targets). Deformation transfer (Sumner and Popović 2004) then provides a method of constructing the target blendshape model, which is mathematically equivalent to solving a certain Poisson equation (Botsch et al. 2006). We also have more recent progresses for the blendshape model transfer, including the one treating with a self-collision issue (Saito 2013) and the technique allowing the user to iteratively add more training poses for blendshape expression refinement (Li et al. 2010).

Conclusion

While the origin of blendshapes may lie outside academic forums, blendshape models have evolved over the years along with a variety of advanced techniques including those described in this article. We expect more scientific insights from visual perception, psychology, and biology will strengthen the theory and practice of the blendshape facial models.

In a digital production workplace, we should also promote seamless integration of the blendshape models with other software tools to establish a more creative and efficient production environment.

Cross-References

- ▶ [Depth Sensor-Based Facial and Body Animation Control](#)
- ▶ [Example-Based Skinning Animation](#)
- ▶ [Laughter Animation Generation](#)
- ▶ [Visual Speech Animation](#)

Acknowledgments I would like to thank J. P. Lewis for mentoring me over the years in the field of computer facial animation research and practice. Many thanks go to Ayumi Kimura for her fruitful discussions and warm encouragements in preparing and writing this article. I also thank Gengdai Liu and Hideki Todo for their helpful comments and creation of the images in Figs. 1, 2, and 3.

References

- Anjyo K, Todo H, Lewis JP (2012) A practical approach to direct manipulation blendshapes. *J Graph Tools* 16(3):160–176
- Bhat K, Goldenthal R, Ye Y, Mallet R, Koperwas M (2013) High fidelity facial animation capture and retargeting with contours. In: *Proceedings of the 12th ACM SIG-GRAPH/Eurographics Symposium on Computer Animation*, 7–14
- Botsch M, Sumner R, Pauly M, Gross M (2006) Deformation transfer for detail-preserving surface editing. In: *Proceedings of Vision, Modeling, and Visualization (VMV)*, 357–364
- Cantwell B, Warner P, Koperwas M, Bhat K (2016) ILM facial performance capture, In *ACM SIGGRAPH2016 Talks*, 26:1–26:2
- CGW web page (2014) <http://www.cgw.com/Publications/CGW/2014/Volume-37-Issue-4-Jul-Aug-2014-/Turtle-Talk.aspx>
- Ekman P, Friesen W (1978) *Facial action coding system: manual*. Consulting Psychologists Press, Palo Alto
- Fxguide web page (2014) <https://www.fxguide.com/featured/voodoo-magic/>
- Lau M, Chai J, Xu Y-Q, Shum H-Y (2009) Face poser: interactive modeling of 3D facial expressions using facial priors. *ACM Trans Graph* 29(1), 3:1–3:17
- Lewis JP, Anjyo K (2010) Direct manipulation blendshapes. *IEEE Comput Graph Appl* 30(4): 42–50
- Lewis JP, Mo Z, Neumann U (2004) Ripple-free local bases by design. In: *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 684–688
- Lewis JP, Anjyo K, Rhee T, Zhang M, Poghin F, Deng Z (2014) Practice and theory of blend-shape facial models. *Eurographics 2014 (State of the Art Reports)*, 199–218
- Li H, Weise T, Pauly M (2010) Example-based facial rigging. *ACM Trans Graph* 29(3), 32:1–32:6
- Mixamo web page (2013) <https://www.mixamo.com/faceplus>
- Neumann T, Varanasi K, Wenger S, Wacker M, Magnor M, Theobalt C (2013) Sparse localized deformation components. *ACM Trans Graph* 32(6), 179:1–179:10
- Noh J, Neumann U (2001) Expression Cloning. In: *SIGGRAPH2001, Computer Graphics Proceedings*, ACM Press/ACM SIGGRAPH, 277–288
- Raitt B (2004) The making of Gollum. Presentation at U. Southern California Institute for Creative Technologies’s *Frontiers of Facial Animation Workshop*, August 2004
- Saito J (2013) Smooth contact-aware facial blendshape transfer. In: *Proceedings of Digital Production Symposium 2013 (DigiPro2013)*, ACM, 7–12
- Seo J, Irving J, Lewis JP, Noh J (2011) Compression and direct manipulation of complex blendshape models. *ACM Trans Graph* 30(6), 164:1–164:10
- Seol Y, Lewis JP, Seo J, Choi B, Anjyo K, Noh J (2012) Spacetime expression cloning for blendshapes. *ACM Trans Graph* 31(2), 14:1–14:12
- Sumner RW, Popović J (2004) Deformation transfer for triangle meshes. *ACM Trans Graph* 23(3): 399–405
- Tickoo S (2009) *Autodesk maya 2010: a comprehensive guide*. CADCIM Technologies, Schererville