A Pose Space for Squash and Stretch Deformation

Richard Roberts
School of Engineering and
Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: robertrich2@myvuw.ac.nz

Byron Mallett
School of Design
Victoria University of Wellington
Wellington, New Zealand
Email: byronated@gmail.com

***Abstract*—'Squash & Stretch' is a fundamental technique in traditional animation, in which the shape of a character or object is intentionally distorted in order to accentuate its movement. Unfortunately the fluid deformation required by the squash & stretch technique is difficult to achieve using current tools, because the development of computer graphics algorithms has been largely driven by the goal of producing photo-realistic images and movement. We contribute a novel technique in which the squash & stretch effect can be introduced and controlled simply by specifying examples of the desired effect at any point along an animation timeline. The examples are associated with the pose of the model, where 'pose' is defined as selected parameters of the underlying animation such as position, velocity, and acceleration. Unlike keyframe animation, which would require a squash or stretch to be specified every time that it occurs, our approach automatically produces the desired effect whenever the pose of the model approaches one of the trained poses. And unlike rigging-based approaches, which require the artist to specify various weight and expressions, our approach allows the desired deformations to be produced simply by sculpting them.**

I. Introduction

In traditional animation, the narrative is emphasised through animation principles such as pacing, squash & stretch, arc, and more [1]. In the *squash & stretch* technique, the shape of a character or object is intentionally distorted in order to emphasise its motion or emotion. For example, a character that is accelerating may be depicted with exaggerated elongation in the direction of motion. Squash & stretch is lauded as the most important principle in the art of animation [2]. Once understood, the convention assists in many aspects of animation including, but not limited to, timing, anticipation, and follow through [1].

Contemporary computer graphics tools facilitate the creation of realistic motion and have been developed to meet demands from the film and gaming industries for photorealistic effects. Unfortunately, the artistic prose of motion has largely been ignored as these tools have been developed, with exceptions reviewed in Section II. For instance, skeleton-driven deformation techniques are commonly used in the animation pipeline. While these approaches allow posing of an articulated character, directly distorting the shape of an object is not possible. The desired deformations can be approximated with simple 'rigging' expressions, such as specifying a uniform scaling in the direction of motion that is proportional to the velocity or acceleration. This formulaic approach is both unintuitive and limited in its expressiveness.

978-1-4799-0883-7/13/$31.00

Few tools are available that aid in the task of producing emotionally expressive motion for characters [3]. A framework for exaggerating an animation, through direct manipulation of present objects themselves, would resemble traditional animation practice. However, developing an appropriate interface for squash & stretch is not trivial, as computer graphics artists are familiar with skeletal interfaces and sculpting, and a radical change to the toolset could be disruptive. Our approach appears as a straightforward and light-weight addition to the existing animation pipeline.

Our approach is distinguishable in that it does not reduce the desired deformation to a simple formula, but instead enables direct control to the aesthetic of the motion. The artist simply moves to a point in the sequence and reshapes the model as required (the term 'sequence' is used to denote an animation produced through any existing technique, such as manual key-framing, skeletal deformation, or motion capture). The deformed shape is associated with the current generalised pose, meaning a set of attributes selected by the artist, such as the position, speed, direction of motion, or other existing attributes of the animation. The deformation reappears automatically whenever the character or object approaches a similar pose. The resulting system is simple and intuitive, making use of sculpting skills familiar to the artist, while automating repetitive aspects of the work.

Our design addresses the following goals:

- We target an artist-supervised workflow. While it may be beneficial to follow emotion analysis approaches such as [4] and [5], we argue that these issues should be at the concern and discretion of the animator.
- The method should allow iterative refinement, which is central to the animation process [6].
- It should allow arbitrary deformations, rather than being limited to simple mathematical expressions.
- The deformations should be directly and intuitively specified, rather than relying on a large number of indirect parameters such as per-vertex weights on transforms (i.e. skinning weights).
- It should not require repeatedly specifying the same information, as would be the case if the squash & stretch distortion were implemented with keyframe animation.

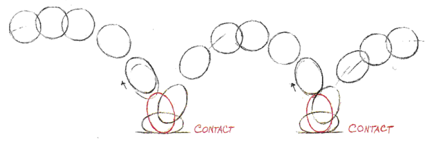


Fig. 1: The classic example squash & stretch: a bouncing ball stretching proportionate to its acceleration (see [6, pp. 93–95]).

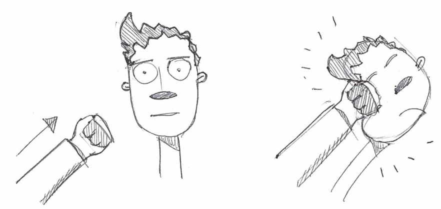


Fig. 2: Squash & stretch is used to emphasise the impact of a punch. (see also [6, pp. 280]).

II. Related Work

In this section we briefly describe the use of squash & stretch in traditional animation, followed by a survey of relevant computer graphics literature. Readers seeking a detailed review of the squash & stretch technique should refer to [7], and to [1] for explanation on how characters (such as the Pixar lamp) use the convention in the context of 3D animation.

A. Squash & Stretch in Traditional Animation

Computer graphics tools have been engineered with focus on the production of photorealism [8]. Because of this, contemporary computer-based systems discourage many stylistic conventions, such as squash & stretch, while traditional animation pipelines facilitate them (due to their hand-drawn nature). For example, Ratatouille (a well known Pixar movie), avoids both motion capture and skeletal deformation interfaces for these reasons [9].

Whether subtle or extreme deformation is used, the squash & stretch convention is considered to be a fundamental visual means of exaggerating a desired message. In Figure 2 it is used to emphasise the impact of a punch, and in 3 it conveys force behind the striking of a piano key. While the technique is primarily used for emphasis, it can simply elucidate motion; [6, pp. 93–95] introduce the classic example of a bouncing ball (Figure 1). As this example serves as a standard test for new animators, we recreate it to demonstrate our approach (see Section IV). Many examples of the squash & stretch technique can be seen in the *Warner Brother Golden Collection* DVD, which features cartoons such as Bugs Bunny and Hubie and Bertie.

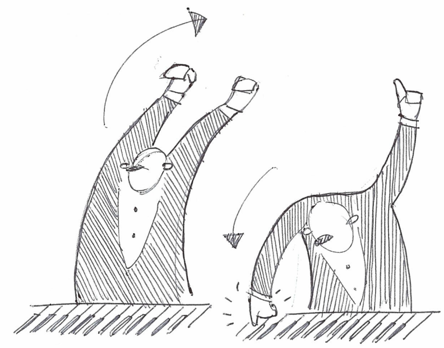


Fig. 3: The force behind a character's hand striking a key is exaggerated (see also [6, pp. 282]).

B. Computer Graphics

In this subsection, existing approaches that directly relate to squash & stretch will be reviewed in detail, while the surrounding topics of emotive analysis of motion and emotive enhancement to motion are outlined.

The "cartoon animation filter" [10], directly address the squash & stretch problem in the contemporary 3D environment. They present a formulaic approach by exploiting the shape of an inverted Laplacian of Gaussian filter. Actions in a sequence are shifted and smoothed when convolved with the filter, producing automatic anticipation and follow-through: two key poses in squash & stretch. Further development by [11] parameterises the time-shift component of this filter, enabling higher-level control over where the key poses occur in the new sequence. The technique is applied to motion capture data enabled through the use of a system for animating a character whose skeletal features are not pre-defined; presented by [12]. The cartoon animation filter provides a method for automatically manipulating motion, however, aesthetic preference varies between characters, their animators, and their directors. We argue that more control should be given to the artist to enable a higher level of artistic refinement of the output animation.

Some research poses the exaggeration of motion as a classification problem. [13] describes an approach to identify and emphasise the most expressive features in motion capture data. This is parallel to [14] and [15], who introduce a model to exaggerate emotive sequences using signal-processing techniques, where emotionally charged motions can be quantified and reformed. For example, the "angry" component of a walk motion can be used to add anger to an otherwise neutral run. [4] take a different approach, outlining a technique for parameterising motions in terms of verbs and adverbs that represent the state of a sequence. These attributes are used to give an animator interactive or programmatic control over the motion. The benefit of such classification approaches is that the focus is to enhance artistic control over motion, as opposed to automating a stylisiation process.

Other approaches feature procedural systems, often developed in order to contribute a robust tool that enables fast manipulation of a sequence. [16] describe a technique that supports both an animator's ability to abstract motion, and the efficiency of procedurally constructing the animation. Using a collision-based system, an animated model can be automatically deformed. The animator effects the shape's deformation through a parameterized interface; perhaps to appear light and rigid or soft and heavy. [5] present a two-step retargeting technique. First, motion is extracted from a given sequence to a two-dimensional space; quantifying movement from robotic to expressive, and aesthetic style from photo-realistic to abstract. The space can then be used to retarget the motion to other characters. Procedural techniques are necessary to manage the progressing complexity of models and objects, often focused on enabling efficient generation of content. Many tools allow parameterised input, but lack novelty when iteratively refinement is necessary. [3] recognise this issue and present a high-level framework for categorizing motion, deriving terminology from a study of motion in performing arts literature. The notion of "character sketches" are used to store aesthetic preferences of motion.

Other notable research includes [17], who contribute a method for direct manipulation of 2D characters, a parallel to the 3D interface of the Pose Space Deformation (PSD) algorithm that we adopt; and a robust augmentation to skeletal deformation that enables an animator to manipulate a character's pose to be more cartoon-like while preserving spatial coherence [9].

III. METHOD

The design of our approach considers two central ideas: that the aim of cartooning is to manipulate motion to emphasise primary actions, while minimising irrelevant movements [10], and that iteration is central to an animator's discourse (see the pose-to-pose technique in [6]). We discuss our system in generality, followed by a brief review of the implementation. See the accompanying resources for detailed overview of the interface.

A. Iterative Direct Manipulation

We formulate and solve the squash & stretch problem using Pose Space Deformation (PSD). This is an 'example based' deformation technique, in which the desired deformation is generated from artist-specified examples. Unlike keyframe animation and spline interpolation, however, the examples are interpolated as a function of 'pose' rather than time or space.

In the prototypical application of generating creature skinning, pose is defined as the concatenated degrees of freedom of relevant parts of the creature's skeleton. The artist specifies deformations at arbitrary poses, leading to a scattered interpolation problem that was solved using Radial Basis functions in [18].

In our problem, using the creature's skeletal pose is not sufficient, and is inappropriate for non-articulated objects such as a bouncing ball. Instead, we define 'pose' with a small number of relevant dimensions such as position, velocity, and acceleration. The velocity and acceleration are derived using the finite difference method. The user can select which parameters form the pose space, and can restrict the position and direction parameters to a plane or line if necessary (X, Y, and Z dimensions can be toggled individually). Several particular examples will be presented in Section IV.

The collection of dimensions (the 'pose space') need only be selected once (though it can be changed at any point), and should be different for each distinct type of squash & stretch deformation that will be applied.

Once the pose space is defined, the artist iteratively refines the existing animation by moving to a point in time, deforming the object as desired, reviewing the result, and moving on to any other places that need attention. The deformed shape is automatically associated with the pose variables (velocity, etc.) at that point in time. It will be termed a *pose instance*.

The underlying software must then interpolate these pose instances as a function of the pose at each frame in the sequence. We note that by formulating the interpolation as a function of pose rather than as a function of time, pose-dependent deformations (such as of a walking character or bouncing ball) need only be authored once, rather than each time that pose occurs. On the other hand, the artist can add additional pose instances at any point if more refinement is desired.

B. Implementation

The dimensionality of the required interpolation is $\mathbb{R}^p \rightarrow \mathbb{R}^{3v}$, where p is the dimensionality of the pose space, and v is the number of vertices, control points, or degrees of freedom on the model. A scattered interpolation method is required, since the poses selected for deformation are arbitrary rather than on any regular grid. Among the variety of possible scattered interpolation methods [19] we use the Radial Basis function (RBF) approach. This provides smooth and controllable interpolation and a simple linear system implementation. The RBF interpolation for a particular component (x,y, or z) of a vertex can be written

$$\hat{d}(\mathbf{p}_i) = \sum_k^N w_k \phi(\|\mathbf{p}_i - \mathbf{p}_k\|) = d_i, \text{ for } 1 \leq i \leq N$$

where $\hat{d}(\mathbf{p}_i)$ is the desired deformation at pose $\mathbf{p}_i$, equaling the artist-specified deformation d_i, and $\phi()$ is the radial basis function. We use the $\phi(\|\mathbf{p}_i - \mathbf{p}_k\|) = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_k\|^2}{2\sigma^2}\right)$. N is the total number of poses. This leads to a linear system in the weights w_k,

$$\begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \cdots \\ \phi_{2,1} & \phi_{2,2} & \cdots & \\ \phi_{3,1} & \cdots & & \\ \vdots & & & \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \end{bmatrix}$$

where $\phi_{j,k} \equiv \phi(\|\mathbf{p}_j - \mathbf{p}_k\|)$.

The system for other vertex components and vertices is similar, having different w_k and d_i. Note however that the $\mathbf{\Phi}$ matrix is in common to all vertices, so the linear system cost is minor (N may be on the order of 3-10 for a single animated shot of several seconds in length).

In our implementation we interpolate the difference between the deformed pose instance and original shape at that pose. With the use of the Gaussian function for $\phi()$, this causes the deformation to automatically and smoothly fade away as the pose changes. The σ parameter can be tuned to globally adjust the extent of the squash & stretch effect, however it can

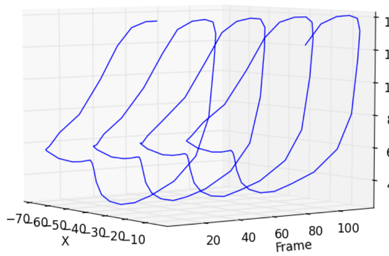


Fig. 4: X and Z position of the right-hand over the frames of the Walkcycle sequence.

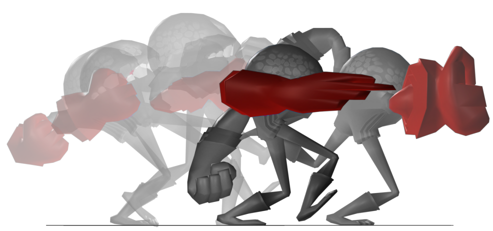


Fig. 5: Transparent view of the pose instances used for the punch sequence.

also be limited in a more controlled fashion by placing "null" pose instances (i.e. shapes with no deformation) at desired poses.

As noted above, the significant benefit to using the pose to drive the interpolation is that the deformation will be automatically repeated in similar situations. In the example of the bouncing ball, Figure 1, the desired deformation can be specified for just one bounce. Since the state of motion for other bounce is nearly identical, the deformation will be replicated to a degree. The velocity is greater in the first bounce (for which poses are defined) and decreases in the following bounce. If more shorter bounces are added, the squash & stretch effect will gradually die out as the ball comes to a rest (see the accompanying material for a video example).

IV. Results

We introduce the implementation, Figure 6a using the classic example: a ball being deformed as it bounces (Figure 1). Before the ball contacts the ground it is stretched (anticipation), squashed when in full contact with the ground, and stretched again when entering the next phase (follow-through). The ball's height, direction of motion, and speed are factors that differentiate these three situations. To create the bouncing ball animation, each of the desired deformations are set at the appropriate frames on the first bounce, and associated with the corresponding pose (defined as the height, direction of motion, and speed tuple).

A secondary action is introduced as a pose instance. The interpolation is updated automatically once the new instance is added, demonstrated in Figure 6b. Secondary actions are another traditional animation convention, refer to discussion on "pose-to-pose" and "secondary-actions" in [1] and [6].

To demonstrate the benefit of our approach in more complex cases, we provide two further examples using an Alien character: *Walkcycle*, an energetic walk cycle, and *Punch*, a clumsy punch. See the accompanying material for videos of these sequences.

To stylise the Walkcycle animation we used three pose instances at each step: Figures 7a and 7b. The first instance (centre) is created and squashed to exaggerate the weight of the character shifting. Stretched poses are then added to act as the anticipation (left) and follow-through (right). Figures 7c and 7d compare the sequence with and without squash & stretch.

The deformation is based on motion of the right hand. Figure 4 shows the X and Z position of the hand throughout the sequence. Using the right hand motion as input to the PSD is ideal as it is both repetitive and differentiates between the "left-foot forward" and "right-foot forward" steps; the rigid transform of the character cannot. Attaching the particle to eye-position would also be suitable, but the motion of the hand is greater and therefore enables a wider pose space. Deferring decisions such as this to the animator enables better creative control.

The Punch animation was created using five pose instances: null, inflation of the hand whilst drawing back, null (again), stretch as the fist gains momentum forward, and finally squash as it slows (Figure 5).

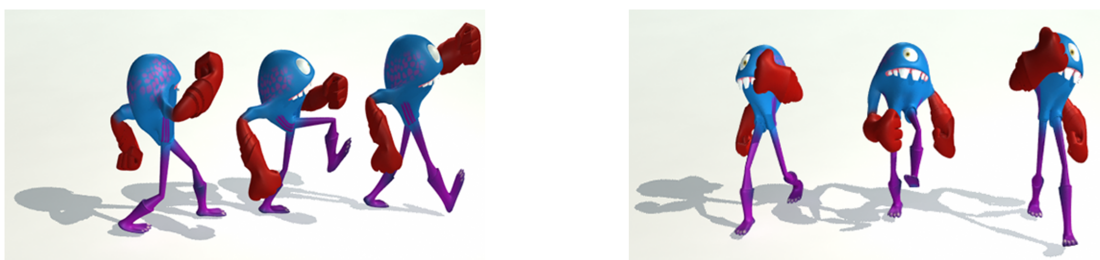
These examples document how the PSD-based squash & stretch algorithm enables an interactive and intuitive approach to stylising motion, thereby bringing traditional animation conventions into contemporary computer animation practice.

V. Conclusion

We argue that the role of computer graphics software is to facilitate creativity while increasing productivity, leaving aesthetic matters in full control of the artist. In this spirit, we have presented a novel approach for stylising animation using the squash & stretch convention, focused on enabling a familiar environment for an experienced animator. Our approach allows direct specification and manipulation of the stylisation at any point in the animation, but automates the repetitive specification of similar shapes that would be required with keyframe animation. This provides the desired balance between automation of, and artistic control over, the deformation process. By pursuing control instead of complete automation the system provides both quality and robustness. The tool is designed to facilitate the squash & stretch convention in 3D animation, but is also suitable for any type of motion exaggeration where direct manipulation is useful.

(a) Poses for anticipation, squash, and follow-through. (b) A wobble (secondary action) is added to the sequence.

Fig. 6: Recreating the classic example squash & stretch example.



(a) Pose instances (left-foot forward) (b) Pose instances (right-foot forward)

(c) Comparison of technique. Left: with, right: without (d) Comparison of technique. Left: with, right: without

Fig. 7: *Walkcycle*, squash & stretch is used to convey a walk cycle as more energetic.

A. Limitations

The selection of pose dimensions introduces a small initial cost to using the system, perhaps foreign to a novice artist. This issue can be resolved by experimenting with the pose space, such as exploring a variety of widths for the Gaussian kernel or selecting different sets of inputs for the space. The real-time feedback enables the user to gain a high-level understanding of the framework.

In cases where the object only travels a small distance, the scene will become cluttered if not managed appropriately, as duplicate meshes of the original character or model are used for each pose instance. Showing only the relevant poses may prove to be beneficial.

Due to the focus on maximising artist control, we do not automate the setup of our system. While the selection of appropriate dimensions for the pose space requires some technical understanding, selecting these dimensions is certainly simpler than writing formulas involving them, as is required in the rigging processes used currently. The possible set of pose variables is open-ended. For example, for a dragon character, one might want the squash & stretch effect to be triggered by the fire breathing (exhaling) actions specified by the animator. An automated setup approach would probably exclude creative associations such as this. On the other hand, the dependency graph structure of some current commercial animation software often makes it possible to indicate relationships such as these simply by connecting attributes in the interface.

B. Future Work

Pose Space Deformation thinly veils a view of animation as scattered interpolation in a suitable space. Abstracting this idea further might enable its application to many traditional animation conventions, as well as other artistic styles seen in comic books and graphic novels.

Acknowledgments

The authors would like to acknowledge Ian Loh for producing Figures 2 and 3, and Mangesh Kondalkar who created the "Boxer Alien" character.

References

[1] J. Lasseter, "Principles of traditional animation applied to 3d computer animation," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 35–44, Aug. 1987. [Online]. Available: http://doi.acm.org/10.1145/37402.37407

[2] K. Roy, *How to Cheat in Maya 2014 Tools and Techniques for Character Animation*. Burlington, MA: Focal Press, 2013.

[3] M. Neff and E. Fiume, "Aer: aesthetic exploration and refinement for expressive character animation," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '05. New York, NY, USA: ACM, 2005, pp. 161–170. [Online]. Available: http://doi.acm.org/10.1145/1073368.1073391

[4] C. Rose, M. F. Cohen, and B. Bodenheimer, "Verbs and adverbs: Multidimensional motion interpolation," *IEEE Computer Graphics and Applications*, vol. 18, no. 5, pp. 32–40, Sep. 1998. [Online]. Available: http://dx.doi.org/10.1109/38.708559

[5] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande, "Turning to the masters: motion capturing cartoons," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 399–407, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566595

[6] R. Williams, *The Animator's Survival Kit*. London: Faber, 2001.

[7] F. Thomas, *The Illusion of Life : Disney Animation*. New York: Hyperion, 1995.

[8] M. Haller, C. Hanl, and J. Diephuis, "Non-photorealistic rendering techniques for motion in computer games," *Comput. Entertain.*, vol. 2, no. 4, pp. 11–11, Oct. 2004. [Online]. Available: http://doi.acm.org/10.1145/1037851.1037869

[9] Y. Savoye, "Stretchable cartoon editing for skeletal captured animations," in *SIGGRAPH Asia 2011 Sketches*, ser. SA '11. New York, NY, USA: ACM, 2011, pp. 5:1–5:2. [Online]. Available: http://doi.acm.org/10.1145/2077378.2077385

[10] J. Wang, S. M. Drucker, M. Agrawala, and M. F. Cohen, "The cartoon animation filter," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1169–1173, Jul. 2006. [Online]. Available: http://doi.acm.org/10.1145/1141911.1142010

[11] J.-Y. Kwon and I.-K. Lee, "The squash-and-stretch filter for character animation," in *ACM SIGGRAPH ASIA 2009 Posters*, ser. SIGGRAPH ASIA '09. New York, NY, USA: ACM, 2009, pp. 11:1–11:1. [Online]. Available: http://doi.acm.org/10.1145/1666778.1666789

[12] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen, "Real-time motion retargeting to highly varied user-created morphologies," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 27:1–27:11, Aug. 2008. [Online]. Available: http://doi.acm.org/10.1145/1360612.1360626

[13] J. W. Davis and V. S. Kannappan, "Expressive features for movement exaggeration," in *ACM SIGGRAPH 2002 conference abstracts and applications*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 182–182. [Online]. Available: http://doi.acm.org/10.1145/1242073.1242195

[14] K. Amaya, A. Bruderlin, and T. Calvert, "Emotion from motion," in *Graphics Interface*, May 1996, pp. 222–229.

[15] M. Unuma, K. Anjyo, and R. Takeuchi, "Fourier principles for emotion-based human figure animation," in *Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 91–96. [Online]. Available: http://doi.acm.org/10.1145/218380.218419

[16] S. Chenney, M. Pingel, R. Iverson, and M. Szymanski, "Simulating cartoon style animation," in *Proceedings of the 2nd international Symposium on Non-photorealistic Animation and Rendering*, ser. NPAR '02. New York, NY, USA: ACM, 2002, pp. 133–138. [Online]. Available: http://doi.acm.org/10.1145/508530.508553

[17] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1134–1141, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073323

[18] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 165–172. [Online]. Available: http://dx.doi.org/10.1145/344779.344862

[19] J. Lewis, F. Pighin, and K. Anjyo, "Scattered data interpolation for computer graphics," SIGGRAPH Asia Course, http://portal.acm.org, 2010.