Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation

J. P. Lewis*, Matt Cordner, Nickson Fong

Centropolis

Abstract

Pose space deformation generalizes and improves upon both shape interpolation and common skeleton-driven deformation techniques. This deformation approach proceeds from the observation that several types of deformation can be uniformly represented as mappings from a *pose space*, defined by either an underlying skeleton or a more abstract system of parameters, to displacements in the object local coordinate frames. Once this uniform representation is identified, previously disparate deformation types can be accomplished within a single unified approach. The advantages of this algorithm include improved expressive power and direct manipulation of the desired shapes yet the performance associated with traditional shape interpolation is achievable. Appropriate applications include animation of facial and body deformation for entertainment, telepresence, computer gaming, and other applications where direct sculpting of deformations is desired or where real-time synthesis of a deforming model is required.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid and object modeling I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: Animation, Deformation, Facial Animation, Morphing, Applications.

1 Introduction

Free form deformation has been approached from several distinct perspectives. As an abstract and general problem, good methods have been obtained both using the well known technique that bears this name [32, 12, 17] and other kinematic surface deformation techniques, and with physical models that simulate the time evolution of a membrane or solid.

The animation of human and creature skin deformation is arguably the most common and important application of free form deformation in computer graphics. While such creature animation can be considered a special case of general free form deformation, its importance and difficulty have lead researchers to propose a number of domain-specific algorithms that will be reviewed in Section 2.

The problem of realistic facial animation is being actively and successfully addressed by image-based and hybrid techniques. These techniques are not yet suitable for all applications, however: while a purely image-based approach can achieve very realistic images, this advantage may be lost if one needs to introduce geometry and surface reflectance in order to re-light characters to match preexisting or dynamically computed environments. Film and entertainment applications require fanciful creatures that fall outside the scope of image-based approaches.

Some of the most impressive examples of geometry-based (as opposed to image-based) human and creature animation have been obtained in the entertainment industry. These efforts traditionally use shape interpolation for facial animation and a standard but variously-named algorithm that we will term skeleton subspace deformation (SSD) for basic body deformation [25, 9]. While shape interpolation is well-liked by production animators, it is not suitable for skeleton-driven deformation. On the other hand SSD produces characteristic defects and is notoriously difficult to control.

These issues, which will be detailed in the next section, lead us to look for a more general approach to surface deformation. We consider the following to be desirable characteristics of a skeleton-based surface deformation algorithm:

- The algorithm should handle the general problem of skeleton-influenced deformation rather than treating each area of anatomy as a special case. New creature topologies should be accommodated without programming or considerable setup efforts.
- It should be possible to specify arbitrary desired deformations at arbitrary points in the parameter space, with smooth interpolation of the deformation between these points.
- The system should allow *direct manipulation* of the desired deformations [33].
- The locality of deformation should be controllable, both spatially and in the skeleton's configuration space (pose space).
- In addition, we target a conventional animator-controlled work process rather than an approach based on automatic simulation. As such we require that animators be able to visualize the interaction of a reasonably high-resolution model with an environment in real time (with 'high resolution' defined in accord with current expectations). Real time synthesis is also required for applications such as avatars and computer games.

Our solution, termed *pose space deformation*, provides a uniform and expressive approach to both facial skin deformation and skeleton-driven deformation. It addresses the previously mentioned drawbacks of shape interpolation and SSD while retaining the simplicity and performance associated with these techniques.

The next section reviews various approaches to free form deformation and describes shape interpolation and skeleton subspace deformation algorithms. The pose space deformation algorithm requires well behaved and efficient scattered data interpolation in high dimensional spaces; Section 3 considers this issue. The pose-space deformation algorithm itself is described in Section 4; examples and applications are shown in the last section.

*zilla@computer.org

2 Background

Recent research has delivered significant improvements in many areas of character animation, including surface representation, model capture, performance capture, and hybrid (partially image-based) rendering approaches. In this literature review we focus specifically on milestones in the surface deformation models and necessarily omit other important contributions.

2.1 Surface Deformation Models

Continuous deformation of a character skin was first addressed in Parke's pioneering facial animation work [26]. In this work, control vertices were deformed by custom algorithmic implementation of carefully selected high-level parameters ('raise-upper-lip', etc.).

Komatsu [13] and Magnenat-Thalmann et. al. [23] demonstrated human body deformation driven by an underlying skeleton. The region and shape of deformation is algorithmically defined in each of these approaches. Magnenat-Thalmann et. al. developed algorithms for each of the various joints in the hand. The discussion in Komatsu focuses on the elbow and shows how the skin crease on the acute side can be obtained by a suitable algorithmic manipulation of the surface control vertices. The algorithms in this early work do not suffer the 'collapsing elbow' characteristic of the SSD algorithm (below). On the other hand, the algorithms are specific to particular types of joints and are perhaps too simple to portray the complexity and individual variability of real anatomy.

The short film *Tony de Peltrie* [3] popularized the use of shape interpolation for facial animation. Forsey [11] describes a character-oriented deformation scheme in which the bending of a smooth surface can be controlled by anchoring levels of a multi-resolution spline surface to the underlying skeleton. These efforts are distinguished from the previous purely algorithmic approaches in giving the modeler control of and responsibility for the deformation.

The specification and animation of surface deformation remains an active area of investigation [17, 10]. The Wires technique [22] is one interesting recent contribution; this approach is notable in providing a direct manipulation interface in a form immediately familiar to sculptors (armatures).

2.2 Multi-Layered and Physically Inspired Models

Chadwick, Haumann, and Parent [7] introduced a multi-layered and physically inspired approach to skin deformation. In their model a free-form deformation abstractly represents underlying body tissues and mediates skin movement. Chadwick et. al. demonstrated expressive three-dimensional cartoon characters but deformation of a realistic character was not shown.

Other researchers have investigated modeling the underlying body tissues in greater depth [27, 24, 8, 35]. Most recently, several groups have undertaken ambitious efforts to produce anatomically inspired multi-layered models of animals and humans with considerable verisimilitude. Nedel and Thalmann [19] simulate the surface deformation of muscles using spring mesh dynamics; a modeled skin cross section is reshaped by a ray-casting procedure that finds the maximum displacement of the underlying tissue. Several papers by Wilhelms and coworkers have shown anatomically representative human and animal models. In Wilhelms and Van Gelder [36] several classes of muscles are algorithmically modeled with attention to volume conservation; skin is a spring mesh anchored to underlying tissue or bone in appropriate areas. Scheepers et. al. [31] produced convincing representations of muscles as well as preliminary but promising skin deformation.

2.3 Common Practice

In recent years character animation has moved beyond being a research topic and sophisticated deforming characters routinely appear in films and on television. Various techniques are employed, including manually animated FFDs and custom procedural approaches in the spirit of [26, 23, 13]. Arguably the most common practice in character animation (as reflected in commercial software, animation books and courses, and some custom software) is founded on the twin techniques of shape interpolation and SSD [18, 9].

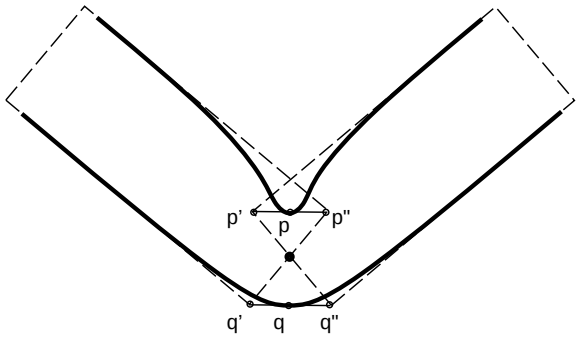

Figure 1: The skeleton subspace deformation algorithm. The deformed position of a point p lies on the line $p'p''$ defined by the images of that point rigidly transformed by the neighboring skeletal coordinate frames, resulting in the characteristic 'collapsing elbow' problem (solid line).

2.3.1 Shape Interpolation

Shape interpolation (also called shape blending and multi-target morphing) is probably the most widely used approach to skin deformation for facial animation [3, 18, 9]. Surface control vertices are simply an animated linear combination (not necessarily convex, i.e., individual weights can be greater than one or less than zero) of the corresponding vertices on a number of key shapes S_k: $\sum_{k=0} w_k S_k$. A variation of this technique uses a single base shape S_0 and a number of delta shapes, $S_0 + \sum_{k=1} w_k(S_k - S_0)$. By writing the delta shape form as $(1 - \sum_1 w_k)S_0 + \sum_1 w_k S_k$ it is clear that the space of achievable shapes is identical in both variations.[1] An attractive feature of shape interpolation is that the desired expressions can be directly specified by sculpting.

The limitations of shape interpolation. Given the popularity and effectiveness of this simple approach, it would be desirable to employ it on regions of the body other than the face. The blending of rigid shapes is inconsistent with regions of the body that are bending under the action of an underlying skeleton, however. Of course the key shapes could be deformed to the moving articulated figure using some other algorithm, but this defeats the purpose of proposing shape interpolation as the means of obtaining the deformation in question.

Shape interpolation also has some drawbacks for its intended role of facial animation. For one, the interpolation is not always smooth. Consider interpolating from a smile (shape A) to a neutral pose (B) and then to a frown (C). An individual vertex travels in a straight line between A and B and again in a line between B and C. Selecting smoothly changing weights with $dw/dt = 0$ at the key shapes merely causes the deformation to "ease in" and stop at each key pose before continuing on – the time derivative of control point motion is smooth, but the motion path itself is only piecewise linear (parametric versus geometric continuity). In practice animators object to the linear nature of the interpolation [34] and have sometimes compensated by sculpting new key shapes as often as every three to five frames [38]. These comments will be revisited in the discussion of the pose space approach later in the paper.

[1] Provided that the weights sum to one. This is enforced in the delta shape formulation. It is not enforced in the (non-delta) shape interpolation formulation as written, but weights that do not sum to one are a separate effect – they cause the face to change overall scale.

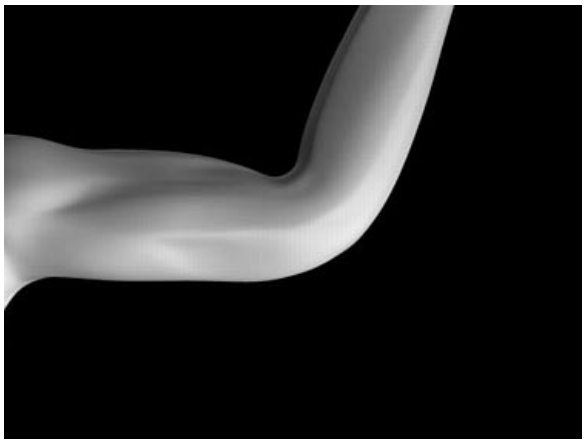

Figure 2: The 'collapsing elbow' in action, c.f. Figure 1.

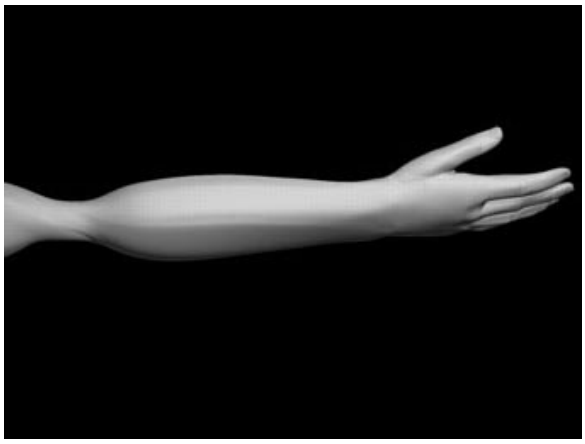

Figure 3: The forearm in the 'twist' pose, as in turning a door handle, computed by SSD. As the twist approaches 180° the arm collapses.

2.3.2 Skeleton-Subspace Deformation

This simple algorithm has been repeatedly conceived and appears in commercial software packages under several rather uninformative names such as skinning, enveloping, etc. The algorithm is unpublished but is subsumed by more general published schemes such as [23]. The position of a control vertex $\mathbf{p}$ on the deforming surface of an articulated object lies in the subspace defined by the rigid transformations of that point by some number of relevant skeletal coordinate frames (Figure 1). This may be notated

$$\bar{\mathbf{p}} = \sum w_k L_k(\mathbf{p})\, \mathbf{p}$$

(in more detail)

$$\bar{\mathbf{p}} = \sum w_k L_k^{\delta} {L_k^0}^{-1} L_{\mathbf{p}}^0\, \mathbf{p}$$

where L_p^0 is the transform from the surface containing $\mathbf{p}$ to the world coordinate system, L_k^0 is the transform from the stationary skeletal frame k to the world system (${L_k^0}^{-1} L_p^0$ together represent $\mathbf{p}$ in the coordinate system of skeletal frame k), and L_k^{δ} expresses the moving skeletal frame k in the world system. The deformation is controlled by the user through the weights w_k.

SSD is fairly versatile. For example, secondary animation effects such as muscle bulging and swelling of the chest can be achieved by variably weighting the surface to an abstract "bone" whose translation or scale is manually animated.

The limitations of SSD. The first major shortcoming of SSD results directly from the fact that the deformation is restricted to the indicated subspace. In common situations such as shoulders and elbows the *desired* deformation does not lie in this subspace, hence no amount of adjusting the algorithm weights will produce good results. This fact leads to considerable frustration by users of the algorithm – the character of the deformation changes as the weights are changed, sometimes sustaining the incorrect assumption that *some* combination of weights will produce good results. In fact, the SSD algorithm can be easily identified in animations by its characteristic 'collapsing joint' defect (Figures 1, 2).

This problem is extreme in the case of simulating the twist of a human forearm (the pose taken in turning a door handle, Figure 3). In this case the subspace basis consists of surface points rigidly transformed by the forearm frame (no axis rotation) and the wrist frame (axis rotation). With a rotation of 180 degrees this line crosses the axis of the arm, i.e., the forearm collapses entirely as the SSD weights transition at some point from the forearm to wrist frames.

A second difficulty with SSD is that, unlike shape interpolation, it does not permit direct manipulation; artists instead directly or indirectly edit the meshes of weights w_k (for each control vertex on a surface there is one weight per skeletal frame that affects the vertex). SSD algorithms consequently have the reputation for being tedious and difficult to control. Artists with a poor understanding of the underlying algorithm have difficulty distinguishing between results that can be further improved by adjusting weights and results that cannot be improved since the desired result lies outside the achievable subspace, resulting in the impression of unpredictability ("sometimes adjusting the weights helps, sometimes it doesn't").

In some cases the SSD defects can be manually corrected using FFDs and other techniques, and one could consider a scheme whereby these fixes are procedurally invoked as the skeleton articulates. But although FFDs work well (and have a direct manipulation algorithm [12]) the layered FFDs do not reduce the difficulty in adjusting the underlying SSD. The algorithm introduced in the subsequent sections removes the need for such layered fix-it approaches and permits direct specification of the desired deformations.

2.3.3 Unified Approaches

Several published algorithms and commercial packages combine aspects of skeleton-driven deformation and shape interpolation in ways that anticipate our approach. In the pioneering work of Burtnyk and Wein, two dimensional characters were animated using a polygonal rubber sheet that afforded both skeletal and local deformation control [6]. Van Overveld described a two-dimensional animation system in which animation is controlled by a skeleton and character deformation is driven from this skeleton through a scattered interpolation [20]. This work is similar in spirit to ours but differs in that it used the image plane as a global interpolation domain rather than introducing a pose space. Litwinowicz and Williams's system [16] is also a precedent and introduced sophisticated scattered interpolation (again in the image domain). Several papers consider animation (and indeed image synthesis in general) as a special case of neural net learning and interpolation/extrapolation [14, 15, 21]. While this viewpoint is valid, in practice it is perhaps excessively general, for example, a skeleton is merely learned rather than being an intrinsic part of the model. While employed at Industrial Light and Magic the first author of the present paper developed a system that attempted to blend shape interpolation and SSD algorithms; a small portion of it remains in use in their well known *Caricature* animation system. Drawbacks of this work included both a complicated dependence on the details of SSD and its overall conception as a "correction" to SSD. Some commercial

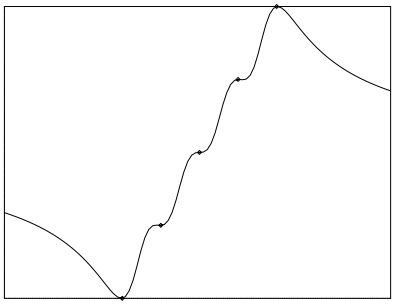

Figure 4: Shepard's interpolant operating on a set of colinear points. The derivative is zero at the data points, and the curve extrapolates to the average of the data values.

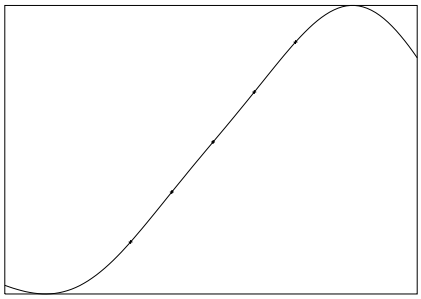

Figure 5: Radial basis functions $\phi(x) = \exp(-x^2/2\sigma^2), \sigma = 10$ interpolating the same set of colinear points as in Figure 4. A different y scale is used to fit the curve. The curve extrapolates to zero.

packages allow blending between two sculpted deformations as a function of a single-joint rotation, thereby combining shape interpolation and skeleton-driven deformation in a limited but useful setting.

2.4 Kinematic or Physical Simulation?

The *depth of simulation* is a prevalent issue in computer graphics, albeit one that is not always consciously considered. Early approaches to animation were purely kinematic; an emphasis on physically based modeling appeared in the literature later. Recent sophisticated approaches allow a hybrid of animator-controlled and physically governed animation as needed. In rendering we perhaps see the opposite trend – much of the literature a decade ago focused on ever deeper simulations of reality, whereas 'shallower' image-based approaches are attracting attention at present.

Similarly, in character deformation both deep and shallow approaches have their place. Deep models promise universally accurate simulation, and the importance of representing humans justifies the needed effort. The authors of these approaches acknowledge that producing anatomically plausible models is a daunting task, however.

Pose space deformation is a shallow, purely kinematic approach to deformation (i.e. without reference to underlying forces, mass, volume), and it has consequent disadvantages. In particular, accuracy is reliant on the modeler/animator rather than being guaranteed by the simulation. On the other hand, our algorithm has clear advantages with respect to simplicity and generality, direct manipulation, real-time synthesis, and other criteria listed in the introduction.

3 Deformation as Scattered Interpolation

In abstract, we wish to express the deformation of a surface as a function of either the pose of an underlying skeleton, or equivalently as a function of some other set of parameters such as the {*smile, raise-eyebrow,...*} controls desirable in facial animation. We also wish to directly sculpt the desired deformation at various points in the parameter space, rather than working in a more abstract space such as the coefficients on various coordinate frames as required by the SSD algorithm.

A scattered data interpolation method is required because deformations will be sculpted at arbitrary (rather than regularly spaced) poses. Since this interpolation is central to our application (the results of the interpolation will be directly visible in the animating deformation), we will consider the available scattered interpolation approaches before settling on a candidate.

3.1 Shepard's Method

Shepard's method [1, 2] is a frequently employed scattered data interpolation scheme in computer graphics. In this method the interpolated value is a weighted sum of the surrounding data points normalized by the sum of the weights,

$$\hat{d}(\mathbf{x}) = \frac{\sum w_k(\mathbf{x}) d_k}{\sum w_k(\mathbf{x})}$$

with weights set to an inverse power of the distance: $w_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\|^{-p}$. (This is singular at the data points $\mathbf{x}_k$ and should computed as $(\|\mathbf{x} - \mathbf{x}_k\| + \epsilon)^{-p}$). With $p > 1$ the interpolation surface is once differentiable. Unfortunately this simple scheme has some potentially undesirable properties. Far from the data the weights will be approximately the same, $\hat{d}(\infty) = w_\infty \sum d_k / w_\infty \sum 1 = \sum d_k / N$, i.e. the interpolated surface converges to the average of the data values. A serious drawback for some applications is that the derivative of the surface is zero at the data points (Figure 4).

3.2 Radial Basis Functions

Radial basis functions [28, 29] have become a popular choice for scattered interpolation. The interpolant is a linear combination of nonlinear functions of distance from the data points:

$$\hat{d}(\mathbf{x}) = \sum_k^N w_k \phi(\|\mathbf{x} - \mathbf{x}_k\|) \tag{1}$$

If N values of d are available then the weights can be easily solved by a linear system; this can be derived either by least squares fit or by subspace projection. Taking the latter approach, we reconsider the available data points as a single point $\mathbf{d}$ in an N dimensional space, and consider $\phi_k() = \phi(\|\mathbf{x}_j - \mathbf{x}_k\|)$ as the kth basis vector. The best approximation to $\mathbf{d}$ in the space spanned by $\phi_k()$ occurs (in direct analogy with the three-dimensional case) when the weights are such that the error $\mathbf{d} - \mathbf{\Phi w}$ (with $\phi_k()$ comprising the columns of $\mathbf{\Phi}$) is orthogonal to each of the $\phi_k()$:

$$\mathbf{\Phi}^T(\mathbf{\Phi w} - \mathbf{d}) = \mathbf{0}$$

so (the so-called "normal equation")

$$\mathbf{\Phi}^T \mathbf{\Phi w} = \mathbf{\Phi}^T \mathbf{d}$$

can be solved for the familiar

$$\mathbf{w} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{d}$$

A least squares approach leads to the identical result.

Any nonlinear function $\phi()$ will interpolate the data, including odd choices such as $\phi(x) = x$ (which is nonlinear since $x = \|\mathbf{x} - \mathbf{x}_k\|$ is the argument), provided that the columns of $\mathbf{\Phi}$ are independent. On the other hand a smooth $\phi()$ will result in a

smooth interpolant (a weighted sum of continuous functions is continuous). In fact radial basis functions have a universal convergence property similar to Fourier series, though the convergence definition is different.

The preceding description maps a k-dimensional input space (arbitrary k) to a one dimensional range, i.e., it is the k-dimensional version of a height field. Surfaces can of course be interpolated by allowing different combinations of the same basis functions in different dimensions, i.e., vector valued w_k. The distance $|||$ can be generalized to Mahalanobis distance (effectively rotating and stretching the basis function) [4].

3.3 Energy Functionals and Non-Convex Methods

Various visual reconstruction schemes can be adapted for scattered data interpolation. In these schemes the interpolated or approximated surface is found as the minimum of a functional such as

$$\int |\hat{d}(\mathbf{x}) - d(\mathbf{x})|^2 d\mathbf{x} + \lambda P(\hat{d})$$

where the first term penalizes deviation of the surface $\hat{d}$ from the available data d and the second regularizing term votes for surface smoothness e.g. by integrating the squared second derivative of the surface. With small λ many of these schemes can serve as scattered data interpolants; reference [5] is a good introduction to these approaches.

In some of the most powerful formulations of scattered interpolation the regularizer is considered to hold everywhere except at an unknown set of edges – this is the piecewise-smooth prior desirable in image reconstruction. Since the unknown edges may exist (or not exist) at any location in the domain, all combinations of possible edge locations must be considered and the interpolation cost is prima facie exponential in the surface resolution.

4 Pose Space Deformation

The crux of our approach is the identification of an appropriate space for defining deformations. As discussed above, the interpolation domain is (a subset of) the pose space of an articulated character, or equivalently the space defined by some set of parameters such as facial controls.

In concept the range of the interpolation function could simply be the desired movement of the surface control vertices. To make the job easier for the interpolation we instead interpolate the desired *deviation* of a surface vertex (expressed in the local frame) from its initially computed position (the rigidly transformed position in the case of an articulated model). Several reasons for this choice will be mentioned shortly.

Thus the deforming surface is defined by $\mathbf{p} + \vec{\delta}$ with $\mathbf{p}$ moved rigidly by the skeleton or other underlying system, and

$$\vec{\delta} = f_{\text{interp}}(\text{configuration})$$

where *configuration* is the configuration of the set of joints or parameters controlled by the animator.

Our scheme can be bootstrapped on top of an existing software system: the model is posed as desired and the desired surface at that pose is sculpted. Our algorithm computes the difference between the initial and resculpted model at that pose. This 'deformation' is associated with the joints or other parameters that have moved from their default positions to create the particular pose. One or more deformations will then be interpolated in this subspace using a scattered data approach.

We now have enough criteria to select a particular interpolation scheme. Although it would be desirable to allow deformations to change both continuously and discontinuously with respect to the pose space, creature deformations that are discontinuous with respect to pose seem unlikely. As such the expensive energy functional and non-convex schemes are not necessary. In addition we want $\vec{\delta}$ to approach zero away from the data, and the width of this falloff should be selectable.

Together these comments support $\phi_k(\mathbf{x}) = \exp(\frac{-(\|\mathbf{x}-\mathbf{x}_k\|)^2}{2\sigma^2})$ as one possible choice of radial basis (Figure 5). Gaussian radial basis functions are reputed to be well behaved and our experience supports this judgement. Gaussian radial basis functions with adjustable placement and σ are discussed in the neural net literature and optimizing over these parameters is possible. This issue does not arise in our application, however, since the animator decides where in the parameter space to sculpt a pose (effectively deciding the basis function placement). The falloff σ is also specified explicitly by the animator, as described below.

4.1 Algorithm Summary

The steps in a pose space deformation (PSD) algorithm will now be described consecutively.

Definitions. A pose is defined as the configuration of any *pose controls* (joints or abstract manipulators) that have changed from their default values. An abstract manipulator is a UI control or arbitrary piece of geometry whose movement will control the interpolation of some deformation, such as a muscle bulge or a desired facial attribute such as "happiness." A self-relative configuration of the controls is actually considered, for example, an elbow involves two skeletal frames but only one joint angle.

The *pose space* is the space spanned by the variations of these controls. If $n = 2$ pose controls are active and each has three degrees of freedom then a $3(n-1)$ pose space is defined, and the particular position of the controls defines a point in that space.

Sculpt. The artist first positions some set of pose controls and then sculpts a deformation for that pose. The artist also assigns a falloff (Gaussian σ), either as a symmetric radius across all controls or to each control individually (axis stretched falloff).

Define $\vec{\delta}$(pose). Any control vertices that have moved from their rest position are found. This is done in the local coordinate frame, i.e., rigid body articulated motion results in zero $\vec{\delta}$. The $\vec{\delta}$ values for the deformed vertices are computed (again in the local coordinate system) and they are saved in a database together with their corresponding location in a pose space. (At the boundary of several surface patches there may be shared vertices that need to be coincident to maintain surface continuity. Unlike some SSD implementations interpolation in pose space by definition cannot separate such vertices).

Solve. When several such deformations have been saved (or when the artist is ready to try animating) it is necessary to solve the interpolation problem. For each control vertex that was moved during sculpting there are now one or more $\vec{\delta}$ values at points in the pose space. Note that the dimension of the pose space can vary across vertices, for example, a particular vertex might be modified in three sculpted deformations but a neighboring vertex might have been modified in only two deformations. The interpolation is done independently for each control vertex (but see additional details below); in our experience using patch surfaces this has not been problematic. Singular $\mathbf{\Phi}^T\mathbf{\Phi}$ is interpreted as a user error; in practice this has turned out to be the result of saving new deformations without moving any pose controls rather than a result of actual numerical problems.

Synthesis. The model is now moved to an arbitrary pose. The location in pose space is determined from the concatenated relative degrees of freedom of the pose controls (simply interpreted as independent dimensions). For each deforming control vertex a $\vec{\delta}$ is interpolated from the delta values at the stored poses using Eq. (1).

Evaluate and Repeat. At this point the model interpolates through the previously defined deformation(s). The most recently defined deformation may extend too far (or not far enough) in pose space, however. There is a rich literature of schemes for optimizing radial basis parameters including σ [4]. On the other hand, animators consider detailed control of the animation to be part of their craft and are quite happy to have interpolation parameters exposed to them. We have found that this potentially abstract parameter is comprehensible so long as it is possible to explore the effect of different values. At a minimum axis-aligned scaling of the falloff should be available; we have not experimented with Mahalanobis rotation of the basis. Based on the evaluation the artist may decide to sculpt additional poses as needed to achieve the desired motion.

A detail that was omitted previously will now be mentioned: when a deformed vertex is found the associated pose space is determined as described above. If there are previous deformations of this vertex in the same pose space then the new deformation is simply another point to interpolate. The new deformation's pose space may, however, be different from the previous spaces associated with the vertex! In such a case a new pose space is started, and the $\vec{\delta}$ is computed as a delta from the previous layered PSD synthesis rather than from the base model. This ensures that the previous deformations are interpolated while allowing the artist complete freedom in determining the extent of the deformation and the associated pose controls. While there is an issue of commutativity, in our experience artists consider this *iterative layered refinement* to be a natural process.

In the preceeding discussion we have not described the representation of rotations and other transformations. This is a well known issue; well behaved transformations are fundamental and are hopefully addressed early in the development of any character animation system.

4.2 Cost

With n poses three matrices of size n must be inverted for each surface control vertex. Typically n will be between 1 and 10, say, so this cost is small. Also it is incurred at a convenient time – during setup (as a pose is saved) rather than during synthesis.

For synthesis, the basis function $\phi(x)$ can be implemented by interpolated table lookup and the `sqrt` required in the Euclidean distance can be composed with $\phi(x)$ in the table. The cost of Eq. (1) is then not much greater than the cost of traditional shape interpolation, i.e., real time synthesis is possible with significant models on current machines.

5 Applications and Discussion

5.1 PSD for Skeleton-Driven Deformation

An articulated model such as a human will typically have a number of different deformation subspaces, each with one or several deformations; the deformations in different subspaces may overlap spatially e.g. to simulate the influence of different muscles. The deformations needed for an elbow, for example, will be interpolated in the one-dimensional subspace defined by the elbow joint angle. Deformations in a shoulder area will need to consider two or more degrees of freedom. The neck/chest/leg blend area of many quadrupeds is a more complex case – the motion of the skin surface in these regions may depend on the relative configuration of several leg bones as well as the rib cage and possibly the neck region of the spine. PSD handles all these cases simply and uniformly.

Figures 8 and 9 are a simple comparison of PSD and SSD algorithms in action on human elbow and shoulder regions.

5.2 PSD for Facial Animation

The application of PSD to facial animation is best described by comparison with shape interpolation (SI).

- In both approaches a set of key shapes (or delta shapes) are sculpted. The same set of shapes can be used in both approaches.
- Whereas shape interpolation is (despite the name) a *superposition* of a set of shapes, PSD *interpolates* among these shapes.
- The animator's task in PSD is to choose the interpolation path (and adjust interpolation parameters such as falloff if desired). With SI the animator's task is to choose the interpolation path but also solve the problem of representing this path by using a set of (non-orthogonal!) basis shapes. In practice this has been considered the major difficulty in applying SI when high quality animation demands large numbers of basis shapes [38].
- In shape interpolation the key shapes and the animation parameter space are one and the same – the keys define the axes of the animation parameter space. In PSD the key shapes are positioned as desired in a space of desired dimensionality.

  One can assign each shape in PSD to a separate dimension, exactly as with SI. On the other hand, PSD allows one to sculpt intermediate expressions (half-smile) and situate them half-way along the relevant (full-smile) axis. Similarly a sculpted pose that represents the simultaneous activation of several parameters (e.g. *happy but surprised*, or *smiling with a wink*) can simply be saved at the appropriate location in the pose space. Psychological research has shown that human facial expressions are largely described by two "emotional" axes [30] (Figure 6); this two-dimensional space would be a convenient high-level pose space for controlling facial animation.
- The PSD interpolation is smooth if so desired.

To illustrate these comments consider Figure 7, which abstractly represents both SI and PSD with an identical set of expressions (*neutral*, *half-smile*, *full-smile*, *frown*). In the SI side of the diagram expressions are arranged as independent (but not orthogonal) dimensions as required by SI. In the PSD diagram the expressions are situated in an expression space having a happy-unhappy axis; a second axis (arousal) and an expression (*delighted*) on that axis are added to show a multidimensional space.

As illustrated, a PSD path from *neutral* to *half-smile* to *full-smile* is monotonic, as might be expected; the motion of a surface point over this interpolation is also smooth. To interpolate these emotions using SI requires a zig-zag pattern of weights: the *half-smile* weight goes from zero to one, and then back to zero as the *full-smile* weight goes from zero to one.The motion of a surface point would also be piecewise linear using SI in this scenario.

5.3 PSD for Secondary Animation

Switches and dials. Additional "dimensions" of deformation can be added at any time by adding a new parameter and associating additional poses with the movement of this parameter. For example, a limb can be modeled in a particular pose both in an unloaded state and with muscles sculpted to express carrying a heavy load. The 'heavy' pose can be associated with the 'on' state of an abstract parameter (e.g. an isolated bone moved into the vertical position); light and heavy loads can then be controlled by flipping this switch. Similarly one can imagine setting up a dial that causes the character to morph; this would of course require a significant set of additional deformation poses.

6 Conclusions

Pose space deformation is not the last word in surface deformation for character animation; high quality anatomically based models are certainly preferable. Nevertheless both anatomically based and

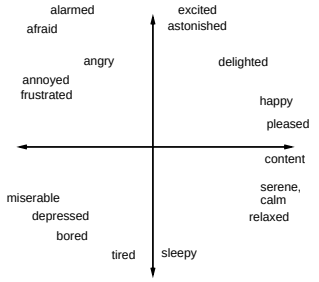

Figure 6: Schematic diagram of emotion space obtained by multi-dimensional scaling from pairwise similarity ratings, simplified from [30].

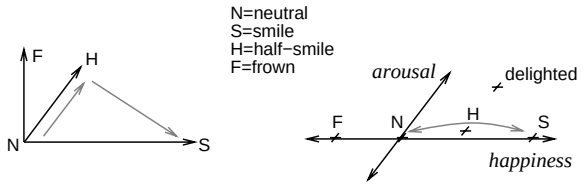

Figure 7: Abstract comparison of shape interpolation versus pose space deformation using the same set of facial expressions.

purely kinematic models have their place. In the current computer animation culture animators generally practice their craft by direct and exhaustive specification of the desired motion combined with quick evaluation using real-time playback. Deeper simulation approaches intrinsically take away some of this control, and animators often argue (rightly or not) that automated processes are inferior or will not produce a human feel. The performance of current anatomically based models prohibits animation preview and other real-time applications such as telepresence and gaming (one published result is several orders of magnitude slower than real time), and the effort needed to produce an anatomically accurate model is not always justified, nor even appropriate if the model is of a fanciful creature whose surface appearance may be inconsistent with any plausible internal anatomy in any case.

PSD unifies and improves upon two techniques that have been common graphics practice for more than a decade. This relatively simple algorithm uniformly handles a variety of deformation situations ranging from a simple elbow to secondary animation. The setup cost of the algorithm is insignificant, and the synthesis cost is only slightly more than that of shape interpolation, so real-time synthesis is possible at effective resolutions on current hardware. We expect that this algorithm will be a useful complement to current techniques.

Acknowledgements

The authors wish to thank Sean Jenkins, David Miya, Amir Nasrabadi, Steven Puri, Kosta Saric, Steffen Wild, Lance Williams, Raymond Yeung, and the anonymous reviewers.

References

[1] R. Barnhill, R. Dube, and F. Little, Properties of Shepard's Surfaces. *Rocky Mountain J. Math.*, vol.13, 1983, pp. 365-382.

[2] T. Beier and S. Neely, Feature-Based Image Metamorphosis. *Computer Graphics* vol. 26, no. 2 (Proc. SIGGRAPH 92), pp 35-42.

[3] P. Bergeron and P. Lachapelle, Controlling Facial Expression and Body Movements in the Computer Generated Short 'Tony de Peltrie'. *SIGGRAPH 85 Tutorial Notes*, ACM, 1985.

[4] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford: Clarendon, 1995.

[5] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, 1988.

[6] N. Burtnyk and M. Wein, Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation, *Comm. ACM*, vol. 19, no. 10 (October 1976), pp. 564-569.

[7] J. Chadwick, D. Haumann, and R. Parent, Layered Construction for Deformable Animated Characters. *Computer Graphics* vol. 23 no. 3 (Proc. SIGGRAPH 89), pp. 243-252.

[8] D. T. Chen and D. Zeltzer, Pump It Up: Computer Animation Based Model of Muscle Using the Finite Element Method. *Computer Graphics* vol. 26, (Proc. SIGGRAPH 92), pp. 89-98.

[9] *Cinefex*, Riverside, CA, various issues, e.g. vol. 66 (June 1996), p. 52 (*Dragonheart*); vol. 64 (Dec 1995), p. 62 (*Jumanji*).

[10] T. DeRose, M. Kass, and T. Truong, Subdivision Surfaces in Character Animation. *Proc. SIGGRAPH 98*, ACM, pp. 85-94.

[11] D. Forsey, A Surface Model for Skeleton-Based Character Animation. In *Second Eurographics Workshop on Animation and Simulation*, Vienna, Austria, 1991, pp. 55-73.

[12] W. M. Hsu, J. F. Hughes, and H. Kaufman, Direct Manipulation of Free-Form Deformations, *Computer Graphics* vol. 26 (Proc. SIGGRAPH 92), pp. 177-184.

[13] K. Komatsu, "Human Skin Model Capable of Natural Shape Variation," *The Visual Computer*, vol. 4, no. 3, 1988, pp. 265-271.

[14] J. P. Lewis, Creation by Refinement: A Creativity Paradigm for Gradient Descent Learning Networks. *International Conf. on Neural Networks.* San Diego, 1988, II: 229-233.

[15] J. P. Lewis, Probing the Critic: Approaches to Connectionist Pattern Synthesis. *IEEE International Joint Conference on Neural Networks* Seattle, July 1991.

[16] P. Litwinowicz and L. Williams, Animating Images with Drawings, *Proc. SIGGRAPH 94*, pp. 409-412.

[17] R. MacCracken and K. Joy, Free-Form Deformations with Lattices of Arbitrary Topology. *Proc. SIGGRAPH 96*, pp. 181-180.

[18] G. Maestri, *Digital Character Animation 2*, Vol 1. New Rider, Indianapolis, 1999. ISBN 1056205-930-0.

[19] L. Nedel and D. Thalmann, Modeling and Deformation of the Human Body Using an Anatomically-Based Approach, *www preprint.*

[20] C. W. A. M. van Overveld, A Technique for Motion Specification in Computer Animation, *Visual Computer*, vol. 6 (1990), p. 106-116.

[21] T. Poggio and R. Brunelli. A novel approach to graphics. AI Memo 1354, C.B.I.P Paper 71, MIT, 1992.

[22] K. Singh and E. Fiume, Wires: A Geometric Deformation Technique. *Proc. SIGGRAPH 98*, ACM, pp. 405-414.

[23] N. Magnenat-Thalmann, R. Laperriere, and D. Thalmann. Joint-Dependent Local Deformations for Hand Animation and Object Grasping. *Proc. Graphics Interface*, 1988, pp. 26-33.

[24] N. Magnenat-Thalmann and D. Thalmann. Human Body Deformations using Joint-Dependent Local Operators and Finite Element Theory. In N. Badler, B. Barsky, and D. Zeltzer, eds., *Making Them Move: Mechanics, Control, and Animation of Articulated Figures* San Mateo, CA: Morgan Kaufmann, 1991, pp. 243-262.

[25] *Maya* Alias/Wavefront, Santa Barbara, CA, 1998.

[26] F. I. Parke, Parameterized Models for Facial Animation. *IEEE Computer Graphics and Applications*, vol. 2, no. 9, November 1982, pp. 61-68.

[27] S. Pieper, Physically-Based Animation of Facial Tissue for Surgical Simulation, *SIGGRAPH 89 Tutorial Notes: State of the Art in Facial Animation*, ACM, 1989.

[28] J. D. Powell, The Theory of Radial Basis Function Approximation. Cambridge University Numerical Analysis Report, 1990.

[29] M. J. D. Powell, Radial Basis Functions for Multivariable Interpolation: A Review. In J. Mason and M. Cox, Eds., *Algorithms for Approximation*, Oxford: Clarendon, pp. 143-167.

[30] J. A. Russel, A Circomplex Model of Affect. *J. Personality and Social Psychology*, vol. 39, p. 1161-1178, 1980.

[31] F. Scheepers, R. Parent, W. Carlson, and S. May, Anatomy-Based Modeling of the Human Musculature. *Proc. SIGGRAPH 97*, ACM, pp. 163-172.

[32] T. Sederberg and S. Parry, Free Form Deformations of Solid Geometric Models. *Computer Graphics*, vol. 20 no. 4, (Proc. SIGGRAPH 86), pp. 150-161.

[33] B. Shneiderman, The Future of Interactive Systems and the Emergence of Direct Manipulation. *Behaviour and Information Technology*, 1, pp. 237-356.

[34] Pat Taylor, Disney/Dream Quest *Mighty Joe Young* facial animation, *personal communication*.

[35] R. Turner and D. Thalmann, The Elastic Surface Layer Model for Animated Character Construction. in N. M. Thalmann and D. Thalmann, eds., *Proc. Computer Graphis International*, New York: Springer Verlag, 1993, pp. 399-412.

[36] J. Wilhelms and A. Van Gelder, Anatomically Based Modeling. *Proc. SIGGRAPH 97*, pp. 173-180.

[37] G. Wyvill, C. McPheeters, and B. Wyvill, Animating Soft Objects. *Visual Computer*, 2, 235-242, 1986.

[38] Xinmin Zhao, Disney *Dinosaur* project, *personal communication.*

Figure 8a. Comparison of PSD and SSD on an animating shoulder – PSD using only two sculpted poses.

Figure 8b. SSD on an animating shoulder. The shoulder area is especially problematic for SSD due to the large range of rotational movement.

Figure 9. Comparison of PSD (at left) and SSD on the extreme pose of an elbow.

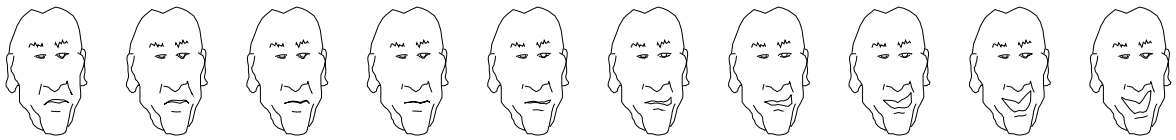

Figure 10. Smooth interpolation of four expressions (frown, neutral, smirk, smile) arranged along a single axis in a pose space, c.f. the discussion of Figure 7.