Pose-weight Interpolation: a Lateral Approach to Pose-based Deformations

Arthur Gregory
DreamWorks Animation
USA

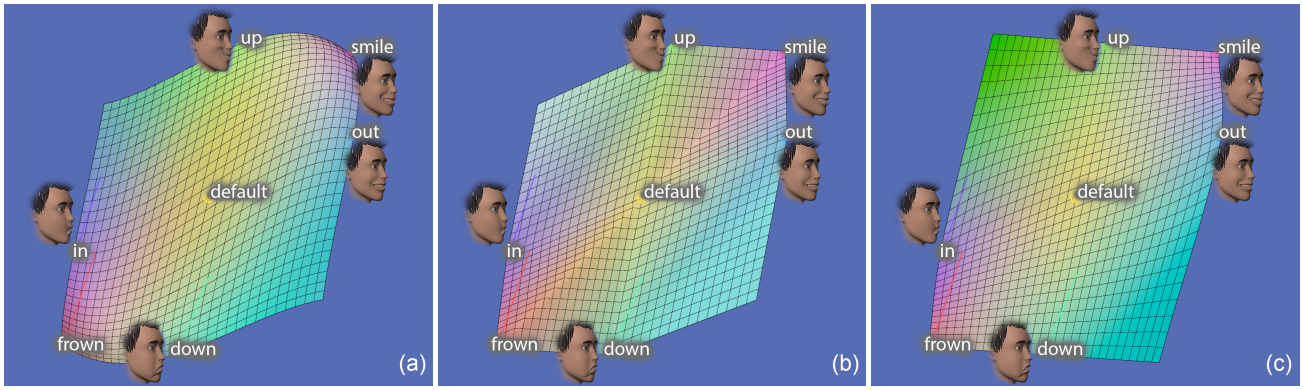

Figure 1: Plot of Pose-weight Interpolation techniques.
(a) Gaussian Radial Basis Functions. (b) Simplex. (c) Our method, Constrained Weight Smoothing.

ABSTRACT

Sculpting character deformations that stay on-model for an arbitrary pose is a non-trivial task. Example based methods are desirable, depending on how few sculpted examples they require. After experimenting with various methods, we find the results are lacking from an artistic point of view. The core problem comes down to a matter of Pose-weight Interpolation, for which we present a novel, artist-friendly solution, Constrained Weight Smoothing. CWS computes Pose-weights on an n-dimensional mesh in pose-space such that weights for an arbitrary pose can be evaluated in O(1) time.

ACM Reference Format:
Arthur Gregory. 2021. Pose-weight Interpolation: a Lateral Approach to Pose-based Deformations. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '21 Talks), August 09-13, 2021.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450623.3464632

1 INTRODUCTION

Pose-based Deformations are often applied in conjunction with other deformation techniques so that sculpted example target models are only required where the existing deformations fail to meet the mark. [Lewis et al. 2000] used Scattered Data Interpolation (SDI) to blend between the targets. This treats the problem like a black box. For an arbitrary pose, there is no simple way to visualize which targets contribute to the result, which makes it difficult for artists to improve. Instead, we divide the problem into Pose-weight Interpolation and target blending, the latter of which can be handled by a weighted sum of the target models multiplied by the Pose-weights. This separation provides a simpler, more direct experience for the artist, and allows for an intuitive visualization of the interpolation, as shown in Figure 1.

SDI can still be used for Pose-weight interpolation if the desired Pose-weights are substituted for the target model, at a each pose. Unfortunately, this exacerbates the problem of overshoot, for which smoothly-interpolating SDI techniques, such as Gaussian Radial Basis Functions, tend to suffer. For example, the rise and fall of the surface plotted between the default pose and smile in Figure 1a tends to cause a distracting wobble in the deformations as a character's mouth is animated (shown in the accompanying video). Although various parameters can be added to tune the result [Lee 2009], the trade-offs are difficult to manage.

To avoid overshoot and deliver more predictable results, [Bengio and Goldenthal 2013] proposed Simplical Interpolation, which uses barycentric coordinates to interpolate between poses on n-simplices constructed in pose-space using the Delaunay algorithm. For C, The Simplical Interpolation suffers from discontinuities along edges between simplices, such as the edge between the default pose and smile in Figure 1b, which causes a hitch in the animation (shown in the accompanying video).

Machine Learning techniques, such as Neural Networks, require too many examples. Pose-weight interpolation is used for fantastical and stylized characters where performance capture is not an option. In practice, our solution rarely requires more than two poses per control.

SIGGRAPH '21 Talks, August 09-13, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8373-8/21/08.
https://doi.org/10.1145/3450623.3464632

2 METHOD

Our solution, Constrained Weight Smoothing, was inspired by surface fairing approaches such as [Desbrun et al. 1999] and [Taubin 1995]. The key difference is that instead of smoothing mesh vertex positions, we smooth weights assigned to the vertices of a static, n-dimensional mesh in pose space.

In the context of deforming a digital character, we define a pose, P, as an n-dimensional subset of local transformations of an underlying skeleton in combination with other controls, such as such as smile or frown. The problem is to compute the Pose-weights for an arbitrary pose C representing the fractional contribution of each of the example poses, P_i, on C, such that $W(P, C) = [w_0, w_1, \ldots, w_m]$.

2.1 Initialization

The initialization step only requires recomputation during the character rigging process when P changes by adding, editing, or removing poses. We construct a mesh, M, as a regular grid spanning P, where V is the set of vertex indices $\{1, 2, \ldots, K\}$. The geometric realization of the mesh, $X\colon V \rightarrow \mathbb{R}^n$, is a mapping from the vertex indices to their locations in the n-dimensional pose space. $W\colon V \rightarrow \mathbb{R}^m$ maps vertices to weight multipliers for each of the m poses. Let $S(k) \subset V$ be the set of the vertices adjacent to vertex k. For shorthand, we use w_k to describe the weights of vertex k.

Similar to the Laplacian operator used to approximate membrane energy in surface fairing techniques, we use a discrete function to compute the energy of the mesh. Instead of using positions, we express energy in terms of the Pose-weights:

$$E(M) = \|D(W)\|^2 = \sum_{k \in V} |D(w_k)|^2 \tag{1}$$

where

$$D(w_k) = \sum_{a \in S(k)} \mu_{ak}(w_a - w_k), \tag{2}$$

and μ_{ak} are the weights satisfying $\sum_{a \in S(k)} \mu_{ak} = 1$, and $\mu_{ak} = 0$ when $a \notin S(k)$. In matrix form, we have

$$E(M) = W^T(U^T U)W \tag{3}$$

where $U = (\mu_{ak})_{KxK}$. Hence, U is a sparse matrix.

For each pose, P_i, the nearest vertex k is weighted entirely P_i and constrained by imposing $D(w_k) = 0$ during a minimization of $E(M)$. M is constructed at a resolution that guarantees a unique closest-vertex per pose.

There are many approaches for minimizing discrete energy. For regular meshes, such as ours, the explicit methods discussed by [Taubin 1995] are relatively efficient. Implicit integration allows for larger time steps, which makes it faster in applications such as [Desbrun et al. 1999]. However, this advantage must be weighed against the additional cost of solving a linear system.

Minimizing Equation (1) tends to flatten the Pose-weights, which can cause abrupt changes in the interpolation around P_i. This is similar to the mesh shrinkage caused by Laplacian Smoothing. [Kobbelt et al. 1998] alleviated this problem by using the second order Laplacian to minimize the change in curvature over the mesh. Following their lead, we minimize $D^2(W)$ where

$$D^2(w_k) = \sum_{a \in S(k)} \mu_{ak}(D(w_a) - D(w_k)). \tag{4}$$

2.2 Weight Interpolation

While posing a character, w_c can be evaluated in $O(1)$ time by using a hash function to look up the mesh cell containing C, and then linearly interpolating the Pose-weights cached on the cell's vertices during the initialization step (Section 2.1). The weights are offset, based on inverse-distance, such that w_c interpolates P.

3 DISCUSSION

The weights computed by CWS transition smoothly between poses without overshoot or popping. This reduces the number of example targets that must be sculpted to meet the required level of art direction. The main drawback is that the mesh size increases exponentially with the dimension of the pose space. However, this has not been an issue, in practice, because for most operations the pose space is two-dimensional. E.g., mouth corner in-out and up-down are interpolated independent of other facial controls. Since Pose-weights are interpolated independent from the resolution of the character model, they are easy to visualize and manipulate. Furthermore, Pose-weights can drive more than just a weighted sum of target models, such as surface relaxations or simulation parameters.

ACKNOWLEDGMENTS

This work stands on the shoulders of two giants: Brent Watkins and Michael Hutchinson. Additional thanks goes to Terran Boylan, Paul DiLorenzo, David Drell, Barry Fowler, Rob O'Neill, Andrew Pearce, Dan Weston, Jeff Woo, and the anonymous SIGGRAPH reviewers.

REFERENCES

Julien Cohen Bengio and Rony Goldenthal. 2013. Simplicial Interpolation for Animating the Hulk. In *ACM SIGGRAPH 2013 Talks (SIGGRAPH '13)*. ACM, New York, NY, USA, Article 7, 1 pages. https://doi.org/10.1145/2504459.2504468

Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. 1999. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 317–324. https://doi.org/10.1145/311535.311576

Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. 1998. Interactive Multi-resolution Modeling on Arbitrary Meshes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 105–114. https://doi.org/10.1145/280814.280831

Gene S. Lee. 2009. Evaluation of the Radial Basis Function Space. In *ACM SIGGRAPH ASIA 2009 Sketches* (Yokohama, Japan) *(SIGGRAPH ASIA '09)*. ACM, New York, NY, USA, Article 42, 1 pages. https://doi.org/10.1145/1667146.1667199

J. P. Lewis, Matt Cordner, and Nickson Fong. 2000. 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 (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 165–172. https://doi.org/10.1145/344779.344862

Gabriel Taubin. 1995. A Signal Processing Approach to Fair Surface Design. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 351–358. https://doi.org/10.1145/218380.218473