Computation of Skinning Weight using Spline Interface

Seungbae Bang
Sung-Hee Lee
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
be2848@kaist.ac.kr
sunghee.lee@kaist.ac.kr

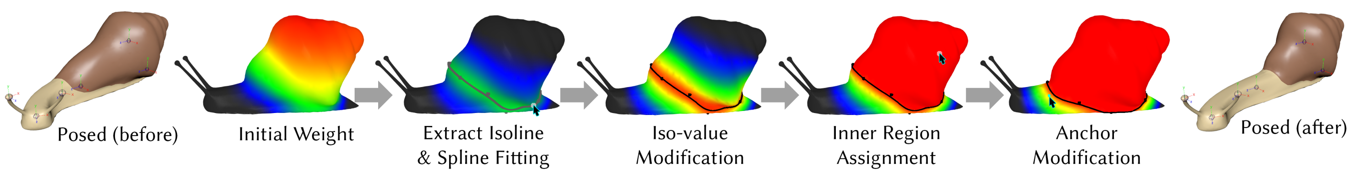

Figure 1: Left two: initial deformation and its skinning weight computed by an automatic algorithm. Our spline interface allows for convenient and intuitive editing of skinning weights by using a small number of splines defined on a mesh surface.

CCS CONCEPTS

• **Computing methodologies** → **Animation**; *Mesh geometry models*;

KEYWORDS

Character Modeling, Rigging, Skinning, Spline

ACM Reference Format:
Seungbae Bang and Sung-Hee Lee. 2018. Computation of Skinning Weight using Spline Interface. In *Proceedings of SIGGRAPH '18 Posters.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3230744.3230801

1 INTRODUCTION

Among many approaches for object and character deformation, closed-form skinning methods, such as Linear Blend Skinning (LBS) and Dual Quaternion Skinning (DQS), are widely used as they are fast and intuitive. The quality of these skinning methods highly depends on specifying appropriate skinning weights to vertices, which requires the intensive efforts of professional artists in production animation.

Software such as Maya and 3DS Max provides a paint-based interface, by which a user manually draws skinning weights on a mesh surface. This tool provides high degree of freedom in specifying the skinning weights, but is not convenient. Users need to paint repeatedly with many strokes to reach a satisfying result while frequently changing many parameters, such as brush size and smoothness.

This paper introduces a novel interface for editing skinning weights by using splines [Bang and Lee 2018]. In our interface, a spline curve defines an isoline of skinning weights. When a user manipulates the spline curves on a mesh surface, which changes the desired shape of the isolines, our method determines the skinning weights on the entire mesh such that the desired isolines are satisfied while the skinning weights between the isolines are smoothly interpolated or propagated.

SIGGRAPH '18 Posters, August 12-16, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5817-0/18/08.
https://doi.org/10.1145/3230744.3230801

2 SPLINE MODEL

We use the standard cubic B-spline model in our method. The B-spline curve is widely used in many applications because of its smoothness and local controllability. By moving an anchor point, we can selectively modify the B-spline curve without losing its geometric continuity. Spline curve that can control the skinning weight must be defined on a surface domain. We use method of [Panozzo et al. 2013], and it approximate geodesic metric using Euclidean-embedding metric.

3 WEIGHT COMPUTATION

Our primary condition to be satisfied is the spline constraint, i.e., the skinning weights of the vertices must be determined to agree with the given skinning weight of the spline. We use barycentric coordinates to interpolate the vertex skinning weights for each mesh face. Stacking the barycentric equation for a set of desired weights of spline points into a matrix constitutes a linear equlity constraint equations: $\mathbf{Ax} = \mathbf{s}^i$. The matrix $\mathbf{A}$ stacks the barycentric condition in each row, $\mathbf{x}$ is the skinning weights of all vertices, and $\mathbf{s}^i$ is a vector of the iso-value of the spline points.

The skinning weight can be computed by the minimizing the energy $E(\mathbf{x})$ that satisfies the constraints of the linear equality equation in the form:

$$\mathbf{w}^s = \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}) \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{s}^i \tag{1}$$

The decision of energy function $E(\mathbf{x})$ determines the overall shape of the skinning weight. We first introduce two types of methods that satisfy the spline constraint: the interpolation-based method which interpolate the given iso-values of splines, and diffusion

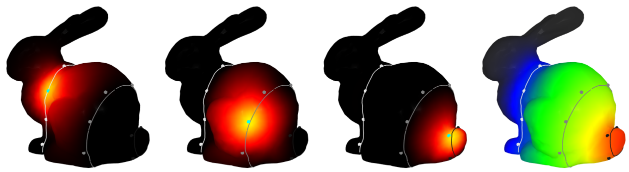

Figure 2: Anchor weight and its applied skinning weight on a bunny model. Top: The results of the hybrid method of isolines [0.1, 0.5, 0.9]. Bottom: The results of the diffusion-based method with only one isoline [0.8].

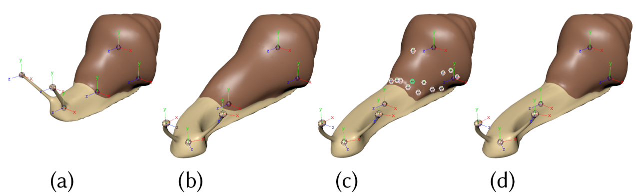

Figure 3: Snail model in a rest pose (a). Deformation result with BBW (b) and BBW with 13 additional point controllers around shell boundary (c). Our result (d).

flow method which diffuse iso-values to surrounding region. We compute the skinning weight using the diffusion flow method if there is only one iso-value spline and use both methods together if there is more than one spline by dividing region with interpolation-based method and diffusion flow method.

We introduce *anchor weight matrix* $\mathbf{K}$ where each column represents the influence of the anchor point on skinning weight of every vertex. With iso-value assigned on each of anchor point as $\mathbf{c}^i$, skinning weight can be computed as $\mathbf{w}^s = \mathbf{K}\mathbf{c}^i$. When a user moves the position of an anchor point, only vertices with non-zero anchor weight corresponding to the anchor point are affected. Therefore, we can reduce the computing region and fasten the computation time. Figure 2 shows the anchor weights and the resulting skinning weights

4 SPLINE EDITING AND FITTING

A user can dynamically build splines from scratch by picking anchor positions on the mesh through a GUI, as shown in the accompanying video. Splines can also be built by extracting isolines from the given skinning weight. Once built, our system supports anchor editing operations, such as repositioning, deletion, insertion, splitting, and merging. During the editing process, our system performs local computations for real-time user interactions.

5 EXPERIMENTS

We implemented our system as a stand-alone application using NanoGUI as the graphical user interface, and the libIGL library [Jacobson et al. 2016] for most of the geometry processing algorithms, such as Laplacian matrix construction and BBW. We used open source code provided by Panozzo et al. [2013] for computing Euclidean-embedding metric. All of the experiments were performed on a MacBook Pro with 2.8 GHz Intel Core i5 CPU and 16 GB of memory.

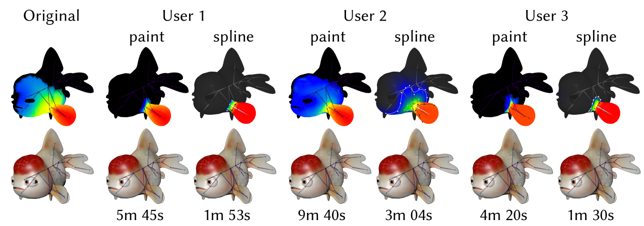

Figure 4: A fish model with the automatic skinning method using Maya (top) and its applied deformation (bottom) on the left image. Three users result on modification on original data using paint and our spline interface. On the bottom shows operation time took for complete the task.

The effectiveness of our interface is well demonstrated in a snail example (Fig. 1), which is rigged with 6 point controllers. In order to keep the shell part rigid when deforming the soft body part, ideal skinning weight should change sharply along the boundary between the two parts. We could obtain a satisfying result by generating a spline curve with high iso-value on the boundary of the shell. Automatic methods of Bounded Biharmonic Weight [Jacobson et al. 2011] that do not consider the desired deformation properties of surfaces can only obtain smoothly varying deformation as shown in Fig. 3 (b). To alleviate the problem, one needs to add significantly more controllers. In the snail example, adding 13 more controllers (Fig. 3 (c)) could reproduce similar result as ours (Fig. 3 (d)) although the shell part could not be kept perfectly rigid.

We compare our spline interface with the traditional paint-based interface by conducting a user test where a user fixes the original skinning weight obtained with an automatic method. Figure 4 (left) shows the initial skinning weight computed Maya's closest distance method. The excessively spread out weights of the fin skeleton cause unnatural deformation. A user is asked to fix this initial skinning weight with a paint interface method and our spline interface method, respectively. And despite much shorter operation time, the spline interface created a result almost similar to the result using paint based interface.

6 LIMITATION AND FUTURE WORK

Our spline interface only works on a single component mesh. However, many rigs in the production consist of multiple mesh patches. Developing a new interface which solve this problem is a nice future direction.

REFERENCES

Seungbae Bang and Sung-Hee Lee. 2018. Spline Interface for Intuitive Skinning Weight Editing (Under Minor Rrevision. *ACM Transactions on Graphics (TOG)* (2018).

Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78.

Alec Jacobson, Daniele Panozzo, et al. 2016. libigl: A simple C++ geometry processing library. (2016). http://libigl.github.io/libigl/.

Daniele Panozzo, Ilya Baran, Olga Diamanti, and Olga Sorkine-Hornung. 2013. Weighted averages on surfaces. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 60.