

Attachment-Based Character Deformation

Nick Toothman

University of California, Davis
njtoothman@ucdavis.edu

Michael Neff

University of California, Davis
mpneff@ucdavis.edu

ABSTRACT

While advancements have made it easier to work with digital characters, it remains difficult to author animations that display the free and highly expressive shape change that characterize hand-drawn animation. We present a deformation method that combines skeletal control and free shape change in a single framework, along with an intuitive, sketch-based interface. By finding attachment points between the mesh and skeleton, we enable configurable skeleton and surface-based deformations, and avoid common skinning artifacts. Use of sketch-based interfaces and graphics hardware make both skeletal and mesh deformation simple to control and fast enough for interactive use.

CCS CONCEPTS

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

KEYWORDS

shape control, deformation, skinning, animation

ACM Reference format:

Nick Toothman and Michael Neff. 2017. Attachment-Based Character Deformation. In *Proceedings of SCA '17, Los Angeles, CA, USA, July 28-30, 2017*, 2 pages.

DOI: 10.1145/3099564.3108161

1 INTRODUCTION

Skeleton-based deformation (skinning) methods typically make it easier to control character rigs, but the types of deformations available are often limited by the rig. Surface-based deformation offers more shape control, but lacks anatomical support. To address this, we present an attachment-based character deformation (ABCD) method designed to support free and easy shape change while still maintaining the advantages of skeletal abstraction. In particular, we wish to support shape change that may or may not be physically correct, as the artist desires, but enhances the artist’s expressive power. Our approach consists of: a new skinning representation that addresses classic skinning artifacts while also allowing shape changes to be encoded; a set of tools for controlling shape change; and a sketch-based interface to make it intuitive to author skeleton and surface deformations in a single framework. Employing GPU acceleration, the approach is suitable for real-time use.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SCA '17, Los Angeles, CA, USA

© 2017 Copyright held by the owner/author(s). 978-1-4503-5091-4/17/07...\$15.00
DOI: 10.1145/3099564.3108161

2 DEFORMATION

Formally, for a vertex v , the attachment point $a_v = p + t_v(c - p)$ is a position along the bone between parent and child joint positions p and c parameterized by $t_v \in [0, 1]$. Use of the t_v is analogous to the endpoint weights for stretchable and twistable bones, but in our method, they effectively replace traditional skin weights [Jacobson and Sorkine 2011]. The scale vector $s_v = v - a_v$ defines the vertex relative to its attachment point. Computing the skinned attachment point as $a'_v = p' + t_v(c' - p')$ incorporates changes in bone position and scale. With this, the rigid skinned vertex position v' is found as: $v' = a'_v + R(p)s_v = a'_v + s'_v$, where $R(p)$ is the parent joint’s rotation transform composed of angle p_θ and axis $p_{\vec{x}}$.

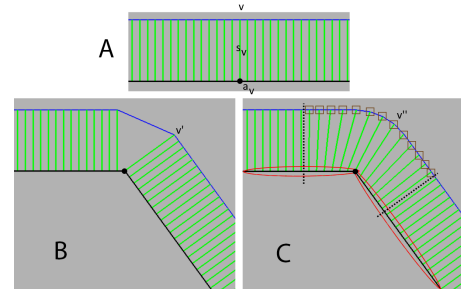


Figure 1: Cross-section of the top half of a cylinder mesh (blue) with one joint and two bones (black). Scale vectors are green; attachment points are circled in red. Dotted lines mark the joint’s influence size. A) bind pose, B) rigid deformation, C) incremental rotation.

Vertices near joints are subject to additional rotation around their attachment points to smooth away discontinuities from rigid skinning. The bone’s joints, p and c , have user-set influence values $p_i, c_i \in [0, 1]$ that denote the range of vertices subject to incremental rotation as distance to the joint decreases. Because p_i, c_i , and t_v are parameterized by bone length, computing the incremental rotation weight is straightforward: $p_w = \max(0, \min(1 - \frac{t_v}{p_i}, 1))$, $c_w = \max(0, \min(\frac{t_v - c_i}{1 - c_i}, 1))$.

Vertices can experience at most half of the joint’s rotation. If vertices on the opposite side receive half of the rotation, but in the opposite direction, the skinning result is smooth. Using an angle-axis rotation matrix constructor $\text{Rot}(\theta, \vec{x})$, for each joint, the potential incremental rotation matrices are $\Delta_p = \text{Rot}(-\frac{p_w p_\theta}{2}, p_{\vec{x}})$ and $\Delta_c = \text{Rot}(\frac{c_w c_\theta}{2}, c_{\vec{x}})$. When $p_w = p_c = 0$, no incremental rotation occurs. Users may choose to overlap influence regions from the parent and child joints, in which a weighted average rotation matrix is used. Otherwise, the non-zero weight determines the final choice of incremental rotation matrix Δ . Using Δ , the result of this stage is $v'' = a'_v + \Delta s'_v = a'_v + s''_v$.

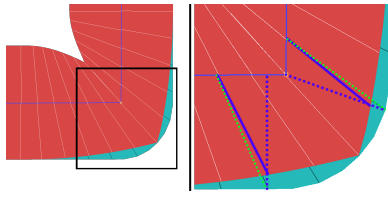


Figure 2: Adjusting scale vector length around joint bends. The default joint deformation is shown in red with the adjusted result shown in teal. Extending the scale vector length (dotted blue) to match its initial length (solid blue) for the final result (dotted green).

By default, this method only rotates s_v . As a result, large bends will generate a pointed tip artifact. This may be desirable or unwanted, depending on the application. The artifact can be eliminated by changing the length of s'_v based on the amount of bend and proximity to the joint, as shown in Figure 2.

Representing vertices as $v = a_v + s_v$ for skinning allows us to change the values of s_v to enable surface-level deformations. Given an edited vertex position v_e , we can compute a new scale vector $s_{v_e} = v_e - a_v$ and use this instead of s_v , encoding the surface deformation into the skinning computation. We use the Laplacian Surface Editing (LSE) and as-rigid-as-possible (ARAP) methods to create deformations, but any surface-based technique is applicable [Sorkine and Alexa 2007; Sorkine et al. 2004]. By interpolating between different sets of s_{v_e} for a mesh region, we can also create surface animations.

3 ATTACHMENT BINDING

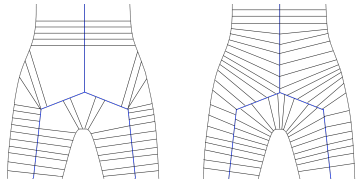


Figure 3: Cross-section of attachment binding without smoothing (left) and with (right).

ABCD produces better deformations when the attachment binding satisfies two requirements: minimize the distances between vertices and bones, and preserve vertex adjacency. Projecting vertices onto the nearest bone satisfies the first requirement, but also creates large gaps between attachment points (see Figure 3). To satisfy the second requirement, after the initial projection, we alternately apply Laplacian smoothing to the attachment points and reproject them onto the skeleton. This process is fast and effective, but too many iterations can violate the first requirement.

4 INTERFACE

To support rapid exploration and simplicity, we use sketch-based tools to drive deformations (see Figure 4). Skeleton posing involves drawing a stroke to select joints and establish a baseline reference, then drawing a target stroke to approximate the desired pose.

This follows the line-of-action stroke mapping presented in differential blending, but with more explicit user control over the mapping [Öztireli et al. 2013].

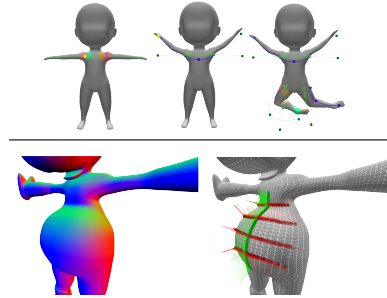


Figure 4: Top: posing the character with input strokes. The baselines (green) select joints and serve as references for the target strokes (blue). Bottom: deferred rendering enables real-time surface querying and surface drawing.

For surface deformations, we initially used a technique that deforms regions based on input strokes outside the mesh border, allowing the user to focus solely on shaping the silhouette [Zimmermann et al. 2007]. This is a powerful technique, but it is limited to vertices that can be detected with image-based edge detection. In our system, we permit the user to draw directly on the mesh surface. We achieve this with deferred rendering techniques for fast surface querying.

5 CONCLUSION

ABCD offers smooth skinning and native mesh deformation in a unified framework with great potential for creating expressive animation in real-time, ranging from naturalistic to exaggerated cartoon styles, with performance comparable to established methods. It performs quite well with or without bone stretch, twist, length adjustment, or surface edits. Future work includes improvements on skinning behavior in complex skeleton regions, more sophisticated attachment binding, improved shape control between skeleton and surface deformations, and scale vector dynamics to support collision, bulge, and movement propagation.

REFERENCES

Alec Jacobson and Olga Sorkine. 2011. Stretchable and Twistable Bones for Skeletal Shape Deformation. *ACM Trans. Graph.* 30, 6, Article 165 (Dec. 2011), 8 pages. DOI: <https://doi.org/10.1145/2070781.2024199>

A. Cengiz Öztireli, Ilya Baran, Tiberiu Popa, Boris Dalstein, Robert W. Sumner, and Markus Gross. 2013. Differential Blending for Expressive Sketch-based Posing. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '13)*, 155–164. DOI: <https://doi.org/10.1145/2485895.2485916>

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (SGP '07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 109–116. <http://dl.acm.org/citation.cfm?id=1281991.1282006>

O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. 2004. Laplacian Surface Editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP '04)*, 175–184. DOI: <https://doi.org/10.1145/1057432.1057456>

Johannes Zimmermann, Andrew Nealen, and Marc Alexa. 2007. SilSketch: Automated Sketch-based Editing of Surface Meshes. In *Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling (SBIM '07)*, 23–30. DOI: <https://doi.org/10.1145/1384429.1384438>