Patch-based Surface Relaxation

Fernando de Goes William Sheffler Michael Comet Alonso Martinez Aimei Kutt
Pixar Animation Studios
{fernando,sheffler,mcomet,alonso,aimei}@pixar.com

Abstract

From rigging to post-simulation cleanups, surface relaxation is a widely used procedure in feature animation. Over the years, Pixar has experimented with several techniques for this task, mostly based on variants of Laplacian smoothing. Notably, none of existing approaches are suited to reproduce the patch layout of a baseline mesh. This is of particular interest for modeling the span of edge flows, or for restoring the rest configuration of a mesh under large deformations. To achieve this goal, we developed a new patch-aware relaxation method for general polygonal meshes. Our approach encompasses three main contributions. We first introduce a weighting scheme that uses local decal maps to encode the structure of edge flows formed by the desirable patch layout. We then propose an update rule that transfers a reference patch arrangement to a deformed mesh. To control volume preservation, we also present a surface-constrained regime that exploits decal maps to slide points within the surface. We demonstrate the effectiveness and versatility of our tool with a series of examples from Pixar's short *Bao* and feature film *Incredibles 2*.

ACM Reference Format:

Fernando de Goes William Sheffler Michael Comet Alonso Martinez Aimei Kutt . 2018. Patch-based Surface Relaxation. In *Proceedings of SIGGRAPH Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Outline

We start by defining basic notations and describing our main steps. The 3D position of a vertex i is denoted by $\boldsymbol{x}_i$ and its incident edges are indicated by ij. We also associate the edge stencil of each vertex with a baseline patch layout with points $\widehat{\boldsymbol{x}}$. Our relaxation approach consists of Jacobi-sytle iterations that update every mesh vertex in parallel. For each vertex i, we first flatten the edge stencil of the current mesh and its corresponding patch layout (§2). This yields 2D coordinates $\boldsymbol{u}$ and $\widehat{\boldsymbol{u}}$, which we refer to as the pose and rest decal maps respectively. We use these decal maps to assign a weight w_{ij} to each oriented edge ij (§3). We then compute a displacement vector $\boldsymbol{d}_i$ per vertex i (§4) and, finally, apply an explicit half-step update: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + 1/2\, \boldsymbol{d}_i$. The user can also control volume preservation by constraining the relaxation within the surface (§5). Additionally, various patch layouts can be chosen in order to achieve different relaxation profiles. In modeling, for instance, the baseline patch is commonly set to a copy of the current mesh so that relative span spacing is kept. Rigging often uses the layout of the undeformed mesh, while post-animation and simulation cleanups can point the layout to any shape in the rig stack. Our method was integrated to Pixar animation system (Presto) both as a deformer and as a sculpting brush, and it has been used across multiple departments.

SIGGRAPH Talks, 2018

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Figure 1: Our span-aware weights relax surfaces while preserving edge flows (see nose and cheek). ©Disney / Pixar.

2 Decal Map

Our core data structure is a decal map per vertex that defines the parameterization of its edge stencil. We construct two decal maps per vertex, one for the posed mesh $\boldsymbol{x}$ and another with the rest points $\widehat{\boldsymbol{x}}$. By traversing the edges incident to the vertex i counter-clockwise, we compute a stencil parameterization using geodesic polar coordinates [Schmidt et al. 2006]. This generates 2D coordinates $\boldsymbol{u}$ that preserve the (geodesic) length of the corresponding 3D edges (i.e., $\|\boldsymbol{u}_i - \boldsymbol{u}_j\| = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$), and with angles between consecutive edges uniformly scaled to sum to 2π. For boundary vertices, we map the edge stencil to a half-disk by normalizing the angle sum to π.

3 Span-aware Weights

Since Pixar models are predominantly subdivision surfaces, the edge flows of the input meshes contain important artistic decisions carefully crafted by modelers and riggers. In order to exploit the arrangement of edge flows, we developed a new weighting heuristic that accounts for the length and angular coordinates of edge stencils encoded by decal maps. Our key insight is to set the weight w_{ij} of an oriented edge ij by evaluating the alignment of the decal map at vertex i along and across the edge ij. More concretely, for each edge ij in the decal map $\boldsymbol{u}$ of the vertex i, we construct an orthonormal frame aligned to ij and compute the largest and smallest projection of every other stencil edge to this frame. This results in a pair of positive $\overline{c}$ and negative $\underline{c}$ values estimating the flow circulation tangent to ij, and another pair $\overline{f}$ and $\underline{f}$ indicating the in- and out-fluxes perpendicular to ij. We then set the edge weight to $w_{ij} = |\underline{c}| \left(\overline{f} - \underline{f}\right)$ and normalize the stencil sum to one.

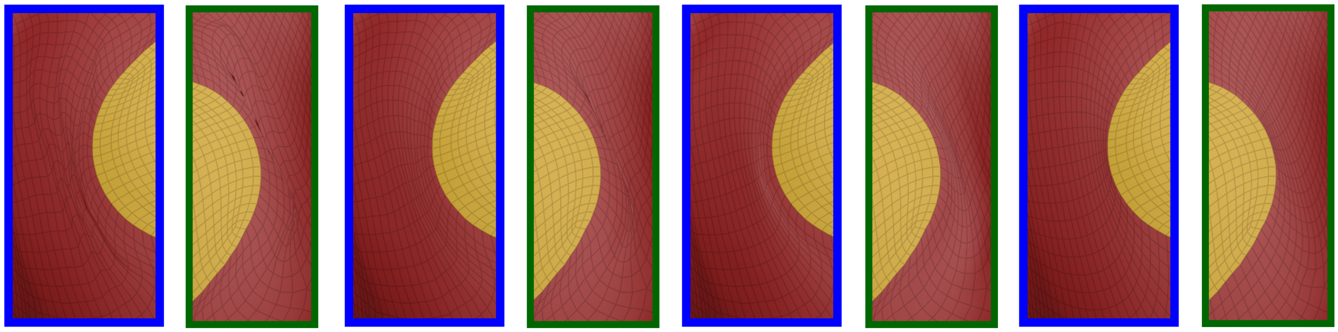

Figure 2: (Left) Close-up of areas with clumping and foldovers produced by a cloth simulation. (Left-Center) Laplacian smoothing [Desbrun et al. 1999] fails to resolve inverted elements. (Center-Right) Delta Mush [Mancewicz et al. 2014] collapses edge spans. (Right) Our approach with patches set to the rest mesh restores edge spacing and smoothness. ©Disney / Pixar.

These weights thus favor neighbors that form the shorter part of edge flows (indicated by large $|\underline{c}|$) and orthogonal to long spans (indicated by large $\overline{f}-\underline{f}$). In the special case of grid-like patches, these weights resemble a bi-linear interpolation that blends the relaxation along each span curve, as shown in Figure 1.

4 Vertex Displacement

To reproduce a desirable layout $\widehat{\boldsymbol{x}}$ in a new pose $\boldsymbol{x}$, we transfer the relative position of vertex i within $\widehat{\boldsymbol{x}}$ to $\boldsymbol{x}$. This is similar to rotated Laplacian coordinates [Lipman et al. 2004] but now computed based on decal maps. We first compute weights w_{ij} for the edges indicident to vertex i using the rest decal map $\widehat{\boldsymbol{u}}$. We utilize these weights to define the relative position vector of vertex i within the rest patch: $\widehat{\boldsymbol{v}}_i = \sum_{ij} w_{ij}\left(\widehat{\boldsymbol{x}}_j - \widehat{\boldsymbol{x}}_i\right)$. Similarly, we use the rest-based weights to compute the relative position vector for the deformed patch: $\boldsymbol{v}_i = \sum_{ij} w_{ij}\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)$. We also compute orthonormal frames to both patches and employ them to determine the rotation matrix $\boldsymbol{R}_i$ that transfers the rest vector $\widehat{\boldsymbol{v}}_i$ to the coordinate system of the posed mesh. At last, the vertex displacement is set to $\boldsymbol{d}_i = (1-\alpha)\,\boldsymbol{v}_i + \alpha \boldsymbol{R}_i \widehat{\boldsymbol{v}}_i$, where α is a scalar that blends the rest-based correction with the relaxation of the posed patch. Figure 3 shows how our tool enriches rigging by restoring local features presented in the baseline mesh. We also compare in Figure 2 the results obtained by our method versus state-of-the-art techniques.

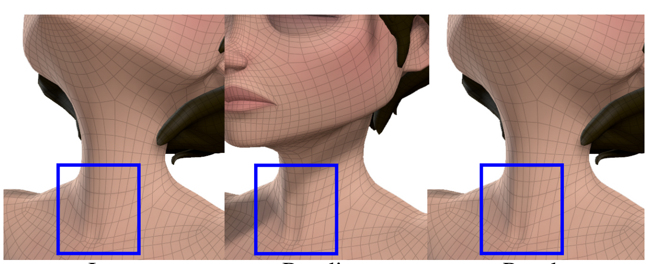

Figure 3: This example shows the neck articulation with our patch-based relaxer off (left) and on (right). We use patches from the undeformed mesh (middle) in order to capture tension features, highlighted in blue. ©Disney / Pixar.

5 Volume Control

We also developed a small modification of our method that restricts vertex displacements to the surface in order to provide volume control. The decal maps are especially suited for this task since it provides a parameterization of the edge stencil incident to a vertex, thus bypassing any normal projection. We compute surface-constrained displacements following the same steps as in §4, but now using the 2D coordinates $\boldsymbol{u}$ and $\widehat{\boldsymbol{u}}$ instead of $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$. This results in a 2D displacement vector $\boldsymbol{d}_i$ that can be added to the vertex position $\boldsymbol{u}_i$ and then lifted back to 3D by interpolating the geodesic polar coordinates $\boldsymbol{u}$. We also provide the user with a scaling factor that blends the 3D displacement with the result of the surface-constrained relaxation. Figure 4 displays the removal of mesh clumping via our patch-based and volume-aware relaxation.

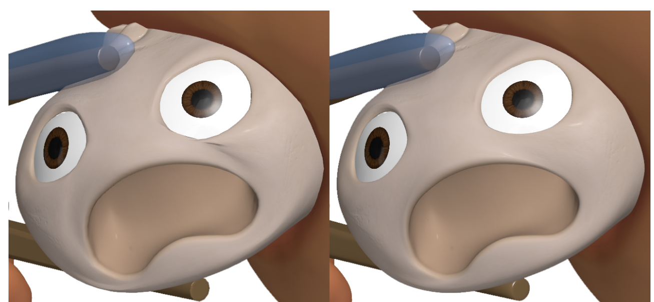

Figure 4: Our method resolves clumping while preserving volume (e.g., near the left eye). This example blended 80% of surface-based with 20% of 3D relaxation. ©Disney / Pixar.

References

M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. 1999. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *SIGGRAPH*. 317–324.

Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel. 2004. Differential Coordinates for Interactive Mesh Editing. In *Proc. of SMI*. 181–190.

J. Mancewicz, M. L. Derksen, H. Rijpkema, and C. A. Wilson. 2014. Delta Mush: Smoothing Deformations While Preserving Detail. In *DigiPro*. 7–11.

R. Schmidt, C. Grimm, and B. Wyvill. 2006. Interactive Decal Compositing with Discrete Exponential Maps. *ACM Trans. Graph.* 25, 3 (2006), 605–613.