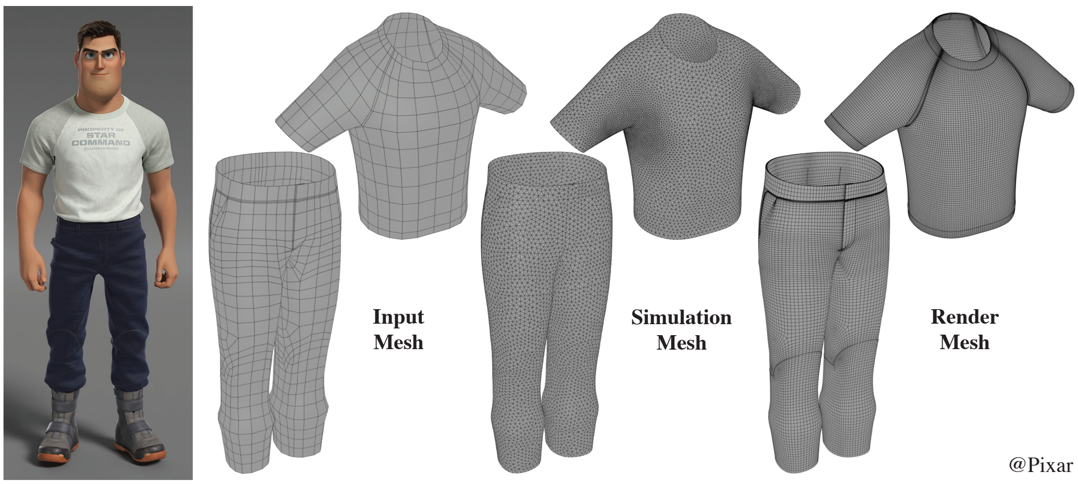
Revamping the Cloth Tailoring Pipeline at *Pixar*

Christine Waggoner
cmw@pixar.com
Pixar Animation Studios
USA

Fernando de Goes
fernando@pixar.com
Pixar Animation Studios
USA



ABSTRACT

This work presents the most recent updates to the cloth tailoring pipeline at Pixar. We start by reviewing the evolution of cloth authoring tools used at Pixar from 2001 to the present day. Motivated by previous approaches, we introduce a structured workflow for cloth tailoring that manages multiple mesh versions concurrently. In our implementation, artists interact primarily with a low-resolution quad-dominant mesh, which defines the garment look as well as setups for rigging and simulation. Our system then converts this coarse input model into a triangulated mesh for simulation and a quadrangulated subdivision surface for rendering. To this end, we developed a new remeshing tool that outputs surface triangulations with adaptive resolution and conforming to edge constraints. We also devised procedural routines to generate render meshes by applying fold-over thickness, refining the mesh, and inserting seams. In addition, we introduced a suite of algorithms for transferring input attributes onto the derived meshes, including UV shells, face colors, crease edges, and vertex weights. Our revamped pipeline was deployed on Pixar's feature films *Turning Red* and *Lightyear*, producing hundreds of high-quality garment meshes.

ACM Reference Format:
Christine Waggoner and Fernando de Goes. 2022. Revamping the Cloth Tailoring Pipeline at *Pixar*. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '22 Talks), August 07-11, 2022.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3532836.3536252

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).
SIGGRAPH '22 Talks, August 07-11, 2022, Vancouver, BC, Canada
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9371-3/22/08.
https://doi.org/10.1145/3532836.3536252

1 INTRODUCTION

The first foray into cloth tailoring at Pixar began on *Monsters Inc.* (2001) to create Boo's t-shirt. Back then, Maya Classic Cloth was employed to construct 2D panels which were then gathered into Delaunay triangulations and used both for simulation and rendering. In subsequent feature films until *Up* (2009), several in-house tools were developed to assist the panel-based workflow, which was internally referred to as *Edna Design*. However, this 2D approach was often scrutinized as a complex system in which art references were manually deconstructed into panels instead of modeled using more familiar 3D tools. Moreover, there was an increasing interest in separating the generation of simulation and render meshes so that smoother shapes with fewer details could be simulated, while rendering visually rich garments with seams and fold-overs.

To address these issues, a new 3D workflow coined as *Cloth 3D* (aka *C3D*) was developed for cloth tailoring in *Toy Story 3* (2010). In this system, garments were designed directly in 3D using typical Maya modeling tools, but were also equipped with constrained edge sets indicating details to be decimated from the simulation mesh or refined by the render mesh. Simulation and render meshes were tessellated separately based on a front advancing strategy that stitches regular hexagonal lattices together over the 3D surface aligned to input edge constraints. This hex-meshing approach thus attempted to produce uniform triangles with valence six vertices to be rendered as Loop subdivision surfaces. Additionally, auxiliary UVs were required as part of the input model in order to copy attributes into the tessellated meshes through panel-based transfers.

Despite the extensive usage for almost a decade, many components in C3D were still cumbersome and often required manual

Figure 1: This example shows simulation meshes for a coat garment (left) tessellated by our method versus the previous hex-based approach (right). ©Pixar.

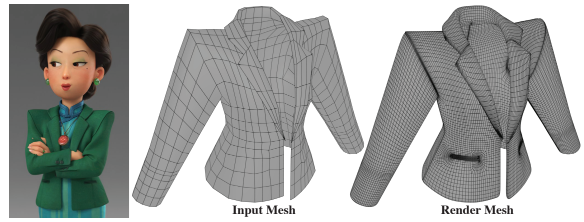


Figure 2: This example shows a blazer garment (left) produced by our procedural workflow that refines the input mesh (middle) into a quadrangulated render mesh with seams and fold-over thickness (right). ©Pixar.

workarounds. For instance, the hex-meshing method was notorious to introduce badly shaped triangles and irregular vertices clumped near edge constraints (Figure 1), thus degrading the simulation performance and the rendering quality. These artifacts were especially problematic for rendering shiny garments such as the supersuits on *Incredibles 2* (2018), which ended up resolved by hand-crafting quad render meshes aside of the C3D pipeline. The need for modeling fine details in the input mesh and later decimated them in the simulation mesh was also undermining, adding laborious data management to the artist. Even transfer schemes sometimes failed to preserve structured attributes such as edge and face sets.

2 CONTRIBUTIONS

For the production of *Turning Red* and *Lightyear*, both released in 2022, we developed a series of improvements onto the C3D system that resolved the limitations discussed in §1. We present next the implementation details for these revamped components.

Simulation mesh: Once the input mesh is 3D modeled, we define the smooth baseline shape targeted by our tessellator by applying a few subdivision steps. Our remeshing algorithm is related to the scheduling method proposed by Botsch and Kobbelt [2004] that alternates edge splits, collapses, and flips, followed by vertex relaxations. In our experiments, we have observed that five iterations are sufficient to produce high-quality triangle meshes. For splits and collapses, we sort mesh edges into a priority queue based on the difference between the edge length and a target value. To enable adaptive resolution, we set the target value by interpolating minimum and maximum length values using painted vertex weights. Edge flips are performed sequentially for every edge whose sum of opposite angles is greater than π, thus favoring Delaunay triangulations. If UVs are available, we compute a relaxed vertex by averaging its neighbors in UV space and then lifting the point back to the input subdivided mesh. Otherwise, we relax each vertex in 3D and project it to the closest point onto the input subdivided mesh. Importantly, our approach preserves constrained edge sets selected by the user and/or extracted from UV shells. To this end, we label a vertex as an anchor if it is incident to one or more than two constrained edges. We then rearrange constrained edges into chains forming loops or connecting pairs of anchors. We identify a collapsible constrained edge when both its vertices are exclusively incident to the same chain. We also update the relaxation step by sliding vertices incident to a single chain along its corresponding curve. In the case of edge flips, we simply skip constrained edges. At last, edge splits are computed with no modification. The teaser figure shows simulation meshes with adaptive resolution and conforming to constrained edges produced by our method, while Figure 1 compares our tessellator versus the hex-meshing scheme.

Render mesh: Orthogonal to the simulation mesh, our pipeline can also generate render meshes represented by quadrangulated tessellations defining Catmull-Clark subdivision surfaces. To construct render meshes, we devised procedural rules that refine the input model through a stack of modeling and deformation operations. For instance, artists can author geometric seams by specifying edge sets in the input mesh and then interactively edit seam parameters such as thickness, normal offset, and left-right asymmetry. We can also make garments double-sided by extruding selected face sets adjacent to the input mesh boundaries and then offsetting corresponding vertices based on user prescribed weights. Moreover, a live preview of these procedural recipes is provided, thus allowing artists to quickly iterate on the garment look. Once the preview is satisfactory, we complete the render mesh by transferring remaining attributes available in the input mesh. Figure 2 shows a render mesh assembled by our approach, including various seams and a double-sided thick collar. In the teaser figure, we compare the shape and connectivity differences between our input and render meshes.

Transfer Tools: In our system, we decorate input meshes with various attributes to be consumed by downstream departments. While transferring these attributes to render meshes can be computed using subdivision stencils, transfers to simulation meshes require custom routines. We exploit our tessellator to transfer attributes set as meshing constraints (e.g., UVs, edge sets, weights). For face sets and colors, we propose a stochastic transfer that samples attributes densely over the input mesh and then reconstructs them in the simulation mesh by counting projected samples per simulation face. To transfer remaining edge sets, we first detect continuous edge chains within each set, construct a temporary face set for every chain side, and then repeat the face set transfer in order to identify the transferred edges. Similarly, we transfer additional UV shells by first processing UV cuts via the edge set approach and then copying UV coordinates within each transferred shell.

REFERENCES

M. Botsch and L. Kobbelt. 2004. A Remeshing Approach to Multiresolution Modeling. In *Proc. of the Eurographics/ACM SIGGRAPH Symp. on Geometry Processing*. 185–192.