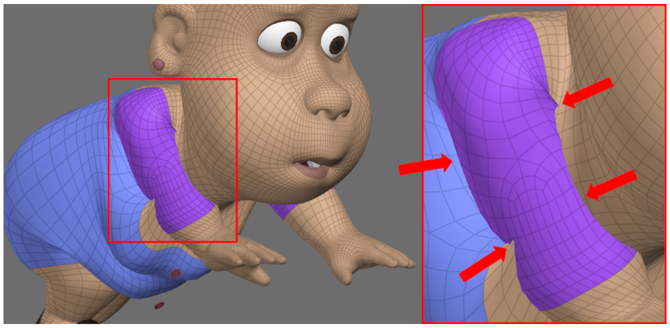
Clean Cloth Inputs: Removing Character Self-Intersections With Volume Simulation

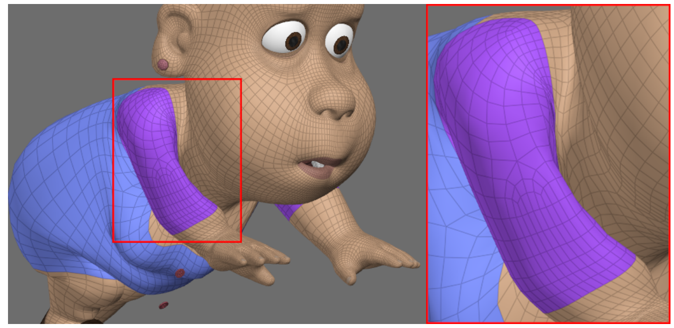
Audrey Wong
Pixar Animation Studios

David Eberle*
Pixar Animation Studios

Theodore Kim
Pixar Animation Studios



(a) Before self-intersection removal



(b) After self-intersection removal

Figure 1: Character from *Bao* before and after volume simulation was performed to clean up self-intersections. Before, the arm badly inter-penetrates the neck and torso. After, there is an ample gap for cloth to slide through. ©Disney/Pixar

ABSTRACT

Simulation artists frequently work with characters that self-intersect. When these characters are sent as inputs to a cloth simulator, the results can often contain terrible artifacts that must be addressed by tediously sculpting either the input characters or the output cloth. In this talk, we apply volume simulation to character meshes and remove self-intersections before they are sent to the cloth simulator. The technique has successfully dealt with very challenging animation scenarios in a production setting, and was applied to all the characters in the short film *Bao*.

ACM Reference Format:
Audrey Wong, David Eberle, and Theodore Kim. 2018. Clean Cloth Inputs: Removing Character Self-Intersections With Volume Simulation. In *Proceedings of SIGGRAPH '18 Talks*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3214745.3214786

1 OUR APPROACH

Animated characters frequently self-intersect at armpits, chins, and elbows, and cloth simulations run over these non-physical inputs often produce unacceptable results. To address this problem, we convert a character's surface mesh into a volume mesh, run a volume simulation to remove all self-intersections, and use the resulting intersection-free surface as the input to the cloth simulator. The approach was successfully applied to every character in *Bao*.

Tetrahedral volume simulation was added to our proprietary cloth simulator Fizt, which is based on the method of Baraff and Witkin [1998]. The tetrahedra leverage a novel and efficient implementation of the As-Rigid-As-Possible (ARAP) model [Alexa et al. 2000]. We chose this over more volume-preserving models because it does not lead to large, visible deformations outside the contact region, which helps maintain the animator's intended silhouette.

In order to maintain stability, the eigenvalues of the force gradient of ARAP must be projected to greater than or equal to zero. Our implementation uses a fast, new, analytic formulation that allows this projection to be computed directly for each tetrahedron [Anonymous 2018]. No numerical eigendecomposition is necessary, because the entire eigensystem is available in closed form.

2 OUR PIPELINE

To clean up the input to our cloth simulation, we have created a pipeline that first creates a tetrahedral mesh from the surface mesh of each character. For each frame in the animation, we warp the surface points of the tetrahedral mesh to match those of the character mesh, and then relax the interior tetrahedra using Projective Dynamics (PD) [Bouaziz et al. 2014] with collisions ignored at this stage. In order to ensure that the result of the volume simulation is as close as possible to the input mesh, we use rest-state retargeting (see e.g. [Kautzman et al. 2012]) so that the rest state of the tetrahedral mesh is updated to the PD result.

The tetrahedral mesh is graded, so vertices are sparse along the interior and dense towards the surface. Distance fields are used to extract a thin core of interior points centered along the main axes of each limb and torso. These points are then hard-constrained ("nailed") and the full volume Fizt simulation is run, including collision processing. Fully constrained regions are automatically omitted, allowing us to reduce the size of the linear system. While the character rig can be evaluated at subframes, we typically compute

*Now at Adobe Systems

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

ACM ISBN 978-1-4503-5820-0/18/08.
https://doi.org/10.1145/3214745.3214786

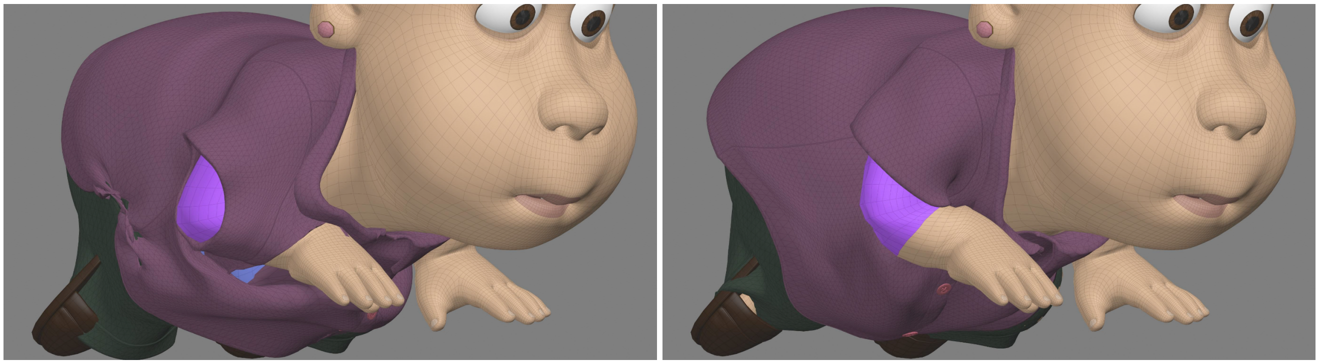


Figure 2: Cloth results before (left) and after (right) our volume simulation pass. Shoes are not simulated. ©Disney/Pixar

the retargeted rest-state once per frame. These rest shapes are then linearly interpolated across each time step during the simulation.

The result of the PD relaxation can contain inverted elements, so we take care to ensure that rest states are not retargeted to inverted configurations. First, we detect if the element's volume has become negative. If so, we swap the pointers of the first and last particles referenced by our element prior to computing forces and their Jacobians. According to our element particle ordering convention, this swap is sufficient to ensure the new rest state will not be inverted. When we are done with the element computation for this alternate ordering, we swap the particles back and adjust the results to obtain ones consistent with the original ordering.

In order to better preserve the original animated shape, we disable gravity. We also use a feature called "killCTM" (Current Transform Matrix) to adjust the amount of translation and rotation that the simulation experiences. In practice, we removed all of this motion. However, these will not remove all undesirable secondary motions arising from violent local deformations. Thus, we also rely on Rayleigh damping of the ARAP model, as well as a feature called "kinSprings". Each dynamic particle has a kinematic position interpolated from the original animation, and we attach a damped spring between the particle and this position. These can then be used to help the simulation track certain regions.

During simulation, we detect and resolve collisions of the surface geometry. Proximity queries are performed at the beginning of each step and generate penalty forces. Penalty forces are assigned a stiffness that is much greater than the internal forces. Continuous collision detection and impulse responses are used to add further robustness after the implicit integration step. Our simulation also uses the Global Intersection Analysis technique from Baraff et al. [2003], which can separate disjoint closed contour regions that started in intersection, and also recover from rare failures that arise during continuous collision response.

Artists can control the deformation by varying the stiffness of the ARAP model. If one side of a contact region contains the majority of the animation performance, such as the face, then we can use nail constraints to exclude it from the volume simulation. This is identical to 3-DOF constraint described in Baraff and Witkin [1998].

The results of our volume simulation are used to deform the character's collision geometry. This is then used as the final input for our cloth simulations. Figure 1 illustrates the benefits of using our approach on a variety of challenging characters in a recent short, *Bao*. The short necks and the squat, round bodies that were integral to the character design in *Bao* exacerbated many of the self intersection issues we had previously seen.

Table 1: The performance of Fizt's volume simulation pass over a variety of characters.

Model	# of tetrahedra	Avg. time per frame
Abuelita (Coco)	381,732	6.56 secs
Bob (Incredibles 2)	47,040	1.80 secs
Mom (Bao)	429,871	4.59 secs
Dad (Bao)	427,716	3.60 secs
Cindy (Bao)	372,612	4.81 secs
Son (Bao)	484,775	6.90 secs

McAdams et al. [2011], mentioned the use of volume simulation for character de-penetration as one possible application. To our knowledge, their method did not update the rest shape and also attempted to preserve volume. Both of these are likely to cause deformations that our animators would find objectionable.

Even with tetrahedral meshes with hundreds of thousands of elements, our simulations take on average 4-7 seconds per frame using 4-6 threads. The performance on several production examples are given in Table 1. Finally, though we have focused on self-intersections, we can apply the same process to resolve collisions with the external environment. This process could be used to clean up the inputs to a cloth simulation for this general case as well.

REFERENCES

M. Alexa, D. Cohen-Or, and D. Levin. 2000. As-rigid-as-possible Shape Interpolation. In *Proceedings of SIGGRAPH*. 157–164.

Anonymous. 2018. Manuscript on fast, analytic ARAP. (under review). (2018).

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of SIGGRAPH*. 43–54.

David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling Cloth. *ACM Trans. Graph.* 22, 3 (July 2003), 862–870.

S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.* 33, 4, Article 154 (2014), 11 pages.

Ryan Kautzman, Jiayi Chong, and Patrick Coleman. 2012. Stable, Art-Directable Skin and Flesh Using Biphasic Materials. In *ACM SIGGRAPH Talks*. 1.

A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (2011), 12 pages.