Simulating Cloth Using Bilinear Elements

Eston Schweickart
Weta Digital
Wellington, New Zealand

Xiao Zhai
Weta Digital
Wellington, New Zealand

ABSTRACT

The most widely used cloth simulation algorithms within the computer graphics community are defined exclusively for triangle meshes. However, assets used in production are often made up of non-planar quadrilaterals. Dividing these elements into triangles and then mapping the displacements back to the original mesh results in faceting and tent-like artifacts when quadrilaterals are rendered as bilinear patches. We propose a method to simulate cloth dynamics on quadrilateral meshes directly, drawing on the well studied Koiter thin sheet model [Koiter 1960] to define consistent elastic energies for linear and bilinear elements. The algorithm elides the need for artifact-prone geometric mapping, and has computation times similar to its fully triangular counterpart.

KEYWORDS

cloth simulation, physically based animation

ACM Reference Format:
Eston Schweickart and Xiao Zhai. 2021. Simulating Cloth Using Bilinear Elements. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '21 Talks), August 09-13, 2021.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450623.3464675

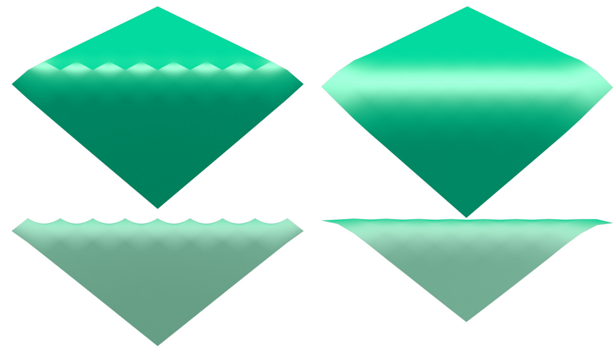

Figure 1: Left: Simulating a regular triangle grid with displacements mapped to bilinear patches results in faceting and tent-like artifacts in high curvature regions. Right: we simulate bilinear patches directly, reducing such artifacts.

1 INTRODUCTION

For visual effects and animation, the geometry used for cloth simulations is often subject to topology requirements from other stages of the production pipeline. Requests to simulate quadrilateral meshes are common, as these topologies are beneficial for modelling, subdivision, and rendering. However, most cloth simulation algorithms are defined only for triangle meshes. We may split quadrilaterals into triangles, or we may remesh the input geometry, but the output simulated mesh must have the same topology as the input in our pipeline. This means we must map the triangle displacements back to the original mesh, which can lead to faceting and tent-like artifacts from artificial curvature induced by the mapping (Fig. 1).

We propose a method for simulating cloth dynamics directly on quadrilateral meshes using bilinear elements. By simulating the same geometric primitive that will later be rendered, we sidestep the risk of introducing artifacts through geometric mapping. We've designed our method to work in a demanding environment: our elastic energies are compatible with both triangles and quadrilaterals; we support anisotropic stretching stiffness; we can guarantee a positive definite energy Hessian for robust implicit integration; and our method is approximately as computationally expensive as simulating an equivalent all-triangle mesh.

Breen et al. [1994] previously described how to simulate cloth on regular quadrilateral meshes. In fact, in other fields, simulating thin shells with quadrilateral elements is quite common; see for instance [Dai et al. 2007] and [Gruttmann and Wagner 2005]. However, unlike our algorithm, these methods commonly increase the number of degrees of freedom, require multiple quadrature points per element, or do not support features that are important to our pipeline such as anisotropic stiffness or a non-planar rest state.

2 OUR CLOTH DEFORMATION MODEL

As input, we take a manifold mesh made of quadrilaterals and triangles. The user may optionally supply a mesh of equivalent topology to represents the cloth's rest state. Additionally, the user may specify *uv*-coordinates for each vertex, which encode the orientation of material space (but not its scale, which is defined by the rest mesh).

We interpret each quadrilateral as a *bilinear patch*, which is the simplest surface defined by 4 points. Given the positions of the vertices of the element $\boldsymbol{p}_0 ... \boldsymbol{p}_3$ ordered counterclockwise, we define the patch as follows:

$$\boldsymbol{p}(s,t) = (1-s)(1-t)\boldsymbol{p}_0 + (s)(1-t)\boldsymbol{p}_1 + (s)(t)\boldsymbol{p}_2 + (1-s)(t)\boldsymbol{p}_3$$

where $(s,t) \in [0,1]^2$. It is trivial to compute the tangents $\frac{\partial \boldsymbol{p}}{\partial s}(t)$ and $\frac{\partial \boldsymbol{p}}{\partial t}(s)$, as well as the tangent gradient $\frac{\partial^2 \boldsymbol{p}}{\partial s \partial t}$, which is constant over the patch. Similarly, we can define mappings from this st-parameter space to the rest mesh and to uv-space.

As is typical in simulations of thin shells, we assume that the deformation energy of our mesh can be decomposed into the sum of membrane (in-plane) and bending (out-of-plane) components. We consider each of these modes individually in the following sections.

SIGGRAPH '21 Talks, August 09-13, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8373-8/21/08.
https://doi.org/10.1145/3450623.3464675

2.1 Membrane Deformation Energy

There are many well known membrane energy models for thin shells. In our system, we use the method proposed by Volino et al. [2009]. In this framework, strains are calculated by constructing a deformation gradient $\boldsymbol{F}$ of the mapping from 2D material space into 3D world space for each element. The Voigt form of the resulting St. Venant-Kirchhoff strain tensor is a vector $\boldsymbol{\varepsilon}_m \in \mathbb{R}^3$:

$$\boldsymbol{\varepsilon}_m = \text{Voigt}\left(\frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I})\right)$$

The entries of this vector measure stretch along the u-axis, stretch along the v-axis, and in-plane shear respectively. This is then used to define the membrane deformation energy:

$$E_m = \frac{h}{2}\int_{\overline{\Omega}} \boldsymbol{\varepsilon}_m^T \boldsymbol{K} \boldsymbol{\varepsilon}_m d\overline{A} \quad (1)$$

where h is the (constant) thickness of the cloth, $\overline{\Omega}$ is the rest surface with area element $\overline{A}$, and $\boldsymbol{K}$ is the 3×3 stiffness matrix.

Since $\boldsymbol{\varepsilon}_m$ is constant over planar elements, the integral can be calculated easily for triangles, but we need a way to compute it for bilinear patches. One option is to approximate the integral in Eq. 1 using Gaussian quadrature at the cost of computing strain values at multiple points on the patch. However, because the strains are quadratic in our degrees of freedom, and because tangents vary linearly over the patch, we can express the sum of quadratures as a linear combination of 7 values which depend only on $\frac{\partial \boldsymbol{p}}{\partial s}\left(\frac{1}{2}\right)$, $\frac{\partial \boldsymbol{p}}{\partial t}\left(\frac{1}{2}\right)$, and $\frac{\partial^2 \boldsymbol{p}}{\partial s \partial t}$. This means that we can approximate Eq. 1 to arbitrary precision while computing just 7 strain values at runtime. Collecting these new strain definitions into a new strain vector $\hat{\boldsymbol{\varepsilon}}_m$, we can approximate Eq. 1 for quadrilaterals follows:

$$E_m \approx \frac{h}{2}\sum_{q \in Q} \hat{\boldsymbol{\varepsilon}}_m^T \hat{\boldsymbol{K}}_q \hat{\boldsymbol{\varepsilon}}_m$$

where Q is the set of quadrilateral faces, and $\hat{\boldsymbol{K}}_q$ is a 7×7 modified stiffness matrix that can be precomputed for every bilinear element and accounts for the quadrature weights and rest area.

Another benefit of using a quadratic strain measure is that the energy Hessian is constant per element, meaning it can be trivially projected to be positive semi-definite by analytically computing its eigen-decomposition. This ensures that Newton's method can be reliably used for implicit integration.

2.2 Bending Deformation Energy

When calculating the bending energy of a mesh composed of non-planar elements, we must consider the bending deformations both within individual elements and between neighboring elements. We consider each of these deformations individually.

In-element bending. We use differential geometry to measure the curvature of each bilinear element. From the first and second fundamental forms, we can construct the *shape operator*, which is a linear operator (in our case, a 2×2 matrix) that describes how the normal changes at a point on a surface when moving along a particular direction in the tangent space. A common bending strain is the mean curvature, equal to the trace of the shape operator. However, this measure is often 0 at the center of bilinear patches, so we would need several quadrature points to integrate this strain over the element accurately. As an alternative, we propose a more general bending energy first described by Koiter [1960]. As noted in [Zorin 2005], this energy can be formulated as:

$$E_b = \frac{h^3}{48}\int_{\overline{\Omega}} k_1 \text{Tr}(\boldsymbol{S} - \overline{\boldsymbol{S}})^2 + k_2 \text{Tr}\left((\boldsymbol{S} - \overline{\boldsymbol{S}})^2\right) d\overline{A} \quad (2)$$

where $\boldsymbol{S}$ and $\overline{\boldsymbol{S}}$ are the shape operators of the current and rest surfaces respectively, and k_1 and k_2 are stiffness measures that can be derived from the entries of $\boldsymbol{K}$. The first term measures mean curvature, and the second term accounts for the Gaussian curvature. We have found that using a single quadrature point at the center of the patch is a sufficient approximation of Eq. 2.

Cross-element bending. There are multiple ways to calculate the shape operator for discrete meshes, though fewer options are applicable to non-triangular meshes. We use the adjoint shape operator definition in [De Goes et al. 2020], which calculates a discrete shape operator at a vertex from the normals of the adjacent faces. The operator is most accurate for regularly arranged quads, which is commonly our input. For bilinear patches, we use the normal at the center of the patch, which is consistent with how normals are defined for quadrilaterals in [De Goes et al. 2020]. This also ensures that the in-element and cross-element energies are disjoint. For consistency with the in-element energy, we use this discrete shape operator to calculate Eq. 2, now summed over all vertices.

3 DISCUSSION

We have proposed a framework for simulating cloth directly on quadrilateral meshes. Simulating the nonlinear deformation of bilinear elements reduces faceting artifacts caused by geometric mappings. To handle collisions, we split each quadrilateral into triangles for this stage, but uniting the collision and simulation surfaces is an important direction for future work. Nevertheless, we believe the proposed model is an important step forward for simulating arbitrary mesh topologies in production environments.

ACKNOWLEDGMENTS

We would like to thank the Simulation department and leadership at Weta Digital for their support.

REFERENCES

David E Breen, Donald H House, and Michael J Wozny. 1994. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 365–372.

Ke-Yang Dai, Gui-Rong Liu, and Thoi-Trung Nguyen. 2007. An n-sided polygonal smoothed finite element method (nSFEM) for solid mechanics. *Finite elements in analysis and design* 43, 11-12 (2007), 847–860.

Fernando De Goes, Andrew Butts, and Mathieu Desbrun. 2020. Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 110–1.

Friedrich Gruttmann and Werner Wagner. 2005. A linear quadrilateral shell element with fast stiffness computation. *Computer Methods in Applied Mechanics and Engineering* 194, 39-41 (2005), 4279–4300.

WT Koiter. 1960. A consistent first approximation in the general theory of thin elastic shells. *The theory of thin elastic shells* (1960), 12–33.

Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics* 28, 4, Article 105 (2009).

Denis Zorin. 2005. Curvature-based energy for simulation and variational modeling. In *International Conference on Shape Modeling and Applications 2005 (SMI'05)*. IEEE, 196–204.