



# Artist-Friendly Tools for Realistic Knitted Cloth

Curtis Andrus  
Animal Logic  
Vancouver, BC, Canada  
curtis.andrus@animallogic.ca

Burak Ertekin  
Animal Logic  
Vancouver, BC, Canada  
burak.ertekin@animallogic.ca

Stefan Feess  
Animal Logic  
Vancouver, BC, Canada  
stefan.feess@animallogic.ca

Guillaume Pernin  
Animal Logic  
Vancouver, BC, Canada  
guillaume.pernin@animallogic.ca



Figure 1: Example of knitted garments worn by ALab Stoats created with Weave.

## ABSTRACT

In this talk, we discuss the knitting system we incorporated into our proprietary woven fabric generation tool, Weave. To make our system production-usable for a large number of characters, we implemented several tools to automate the generation and placement of knitted patterns. This allows artists to create knitted garments with a few simple steps using techniques they are already familiar with, while taking advantage of features in our existing cloth tools.

## CCS CONCEPTS

• Computing methodologies → Parametric curve and surface models.

## KEYWORDS

Procedural Geometry, Knit, Houdini, Weave, Fabric

### ACM Reference Format:

Curtis Andrus, Burak Ertekin, Stefan Feess, and Guillaume Pernin. 2023. Artist-Friendly Tools for Realistic Knitted Cloth. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '23 Talks)*, August 06–10, 2023, Los Angeles, CA, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3587421.3595465>

## 1 INTRODUCTION

Weave had an existing knitting option, but it was limited to copying a single loop over the entire garment. Our upcoming shows, however, required a much wider variety of knitted garments so we

needed to implement a system that allowed us to create complex patterns with ease and also scalability.

We started with stitch meshes, an approach to knitting proposed by [Yuksel et al. 2012]. In a stitch mesh, knitted loops are represented by faces on a mesh; simple loops like knits/purls are represented as quads, and more complicated loops like increases/decreases are represented by faces with additional edges for incoming/outgoing loops. While other approaches have been successful (such as [Walker et al. 2022]), we found stitch mesh’s close connection to real-world knitting an appealing feature.

Unfortunately, the tools described for manipulating stitch meshes require several manual steps specific to a given model, so they are impractical for artists working on a large number of characters. In the next section, we describe how we’ve built a toolset to automate a significant amount of the stitch mesh creation process.

## 2 KNITTING WORKFLOW

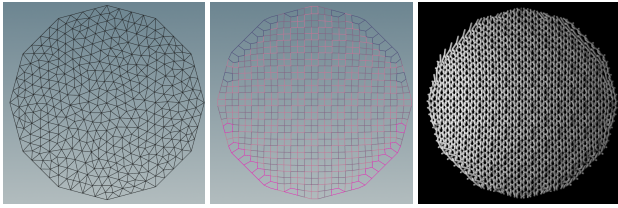
Our tools work by automatically generating a stitch mesh (called a *knit mesh* in our system to avoid confusion with an existing feature). First, a base knit mesh is generated from the input garment mesh.

Artists then pick from a set of pre-defined patterns and apply these patterns to the knit mesh using our procedural Houdini SOP nodes. The cloth is then rendered with our proprietary renderer, Glimpse, where our Weave plug-in converts the knit mesh to renderable yarn curves and ensures they follow any motion applied to the original mesh.

### 2.1 Initialization

We initialize our knit mesh by re-meshing the input geometry into a quad-dominant mesh with rows and columns aligned to UV directions, matching the original tiling generated by Weave. This initialization sets up attributes describing the knitting directions, which are used by pattern placement nodes. We provide the option

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 '23 Talks, August 06–10, 2023, Los Angeles, CA, USA  
© 2023 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0143-6/23/08.  
<https://doi.org/10.1145/3587421.3595465>



**Figure 2: Weave’s knitting workflow from input mesh to knit mesh to rendered yarn curves. More details can be seen in our supplemental video.**

to subdivide specific regions of the knit mesh to have different density levels of knitting across a garment.

## 2.2 Pattern Nodes

There exists an infinity of stitch patterns, these being composed of simple purl, knit, or more complicated stitches. To integrate some of these patterns onto our knit mesh, we have developed two nodes that facilitate the process, both relying on our knit mesh annotation (described in the supplemental material) to use row and column numbers. In order to avoid a dependency on the base mesh’s shape or topology, we use textures to drive the placement of patterns. This lets Surfacing artists work with familiar texture painting tools, and avoids the need for any manual updates if small changes are made to the model (UVs are fairly stable in our pipeline).

**Quad Pattern Fill** lets the artist specify a combination of knits and purls via a simple string description (e.g "0011" for the 2x2 rib pattern used in Figure 3a), which will be applied to a region of the knit mesh. Modular arithmetic on the row/column number is then used to determine the final pattern placement.

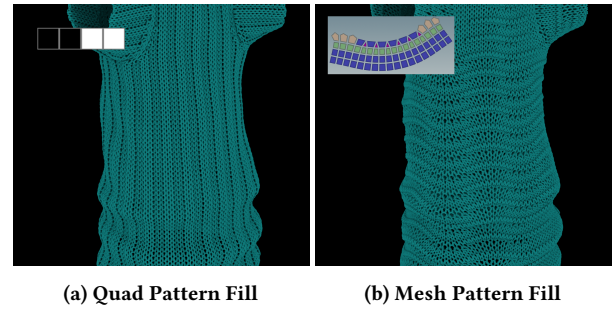
**Mesh Pattern Fill** allows artists to stamp a pre-defined pattern over a region of the knit mesh, replacing the quad-dominant topology with the pattern’s topology (Figure 3b). To ensure that no cracks or holes are generated, we require that the region of the mesh be locally grid-like (i.e. all quads with well-organized rows/columns) and that edge points of the pattern’s knit mesh are mappable to these grid coordinates. To control the placement, artists have several options such as left/right/center-alignment and spacing.

## 2.3 Pattern Library and Tile Generation

The *Pattern Library* defines a collection of out of the box patterns ready to be placed on a model by using *Mesh Pattern Fill*, along with a visual selection box to quickly select the fitting pattern. To support the generation of multiple stitch types, we built a Houdini node to procedurally generate new stitches. These stitches are stored as curves in USD and mean-value coordinates are also written out so we can quickly deform the reference tile to face on the knit mesh.

## 2.4 Post-Processing

To get a more natural looking mesh and to approach the relaxation of the stitches, we used Houdini’s Vellum solver in combination with smoothing operations to emulate the *Mesh-based Relaxation* step from Yuksel et al.. The high computational demand of the



**Figure 3: Knitted Meshes created with our pattern fill tools.**

second step of Yuksel et al.’s relaxation called *Yarn-level Relaxation* prohibits its use in our current workflow.

## 2.5 Rendering

The knit mesh is then passed to our render procedural, which will generate curves representing the yarn. On top of this, options from the original Weave such as fuzz, displacement or scraggle are still available to obtain a unique look [Smith et al. 2018]. Our supplemental material contains further details on yarn generation and shading capabilities.

## 3 CONCLUSIONS AND FUTURE WORK

By automating a few key steps in the knit mesh creation process we have been able to achieve a very scalable workflow for our knitting tools. This workflow lets artists design knitted garments with intuitive options, quickly going from the starting mesh to trying different patterns on the knit mesh, to having the rendered garment visualized.

Adding new patterns to our library currently requires several manual steps that are tedious and prevent the quick addition of new knitting patterns. A lead could be the automatic generation of the geometry representing a knitting pattern from a traditional knitting chart. Our current workflow does not allow the construction of cables or the correct relaxation of lace patterns. Finding a way to support them in our system would be a good future step.

## ACKNOWLEDGMENTS

We would like to thank Don Boogert for his early work on the render procedural, Beau Parkes for integrating the knitting tools into the Animal Logic pipeline and Valeria Lanfranchi for setting up example materials on the sweater and rendering the teaser figure.

## REFERENCES

- Bryan Smith, Roman Fedotov, Sang N. Le, Matthias Frei, Alex Latyshev, Luke Emrose, and Jean Pascal leBlanc. 2018. Simulating Woven Fabrics with Weave. In *ACM SIGGRAPH 2018 Talks* (Vancouver, British Columbia, Canada) (*SIGGRAPH '18*). Association for Computing Machinery, New York, NY, USA, Article 12.
- Megan Walker, Rachele Bellini, Cory Sivek, and Shyh-Chyuan Huang. 2022. Clothing Suite: Interactively Design Complex Garments. In *ACM SIGGRAPH 2022 Talks* (Vancouver, BC, Canada) (*SIGGRAPH '22*). Association for Computing Machinery, New York, NY, USA, Article 32.
- Cem Yuksel, Jonathan M Kaldor, Doug L James, and Steve Marschner. 2012. Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–12.