SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

THE PREMIER CONFERENCE & EXHIBITION ON COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

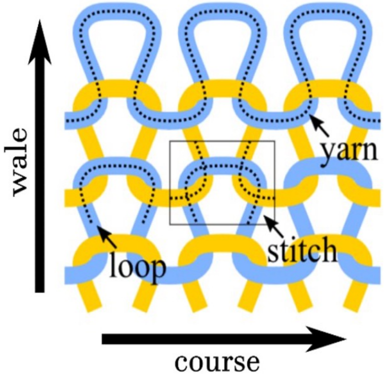
Helix-Free Stripes for Knit Graph Design

Rahul Mitra[1], Liane Makatura[2], Emily Whiting[1], Edward Chien[1]

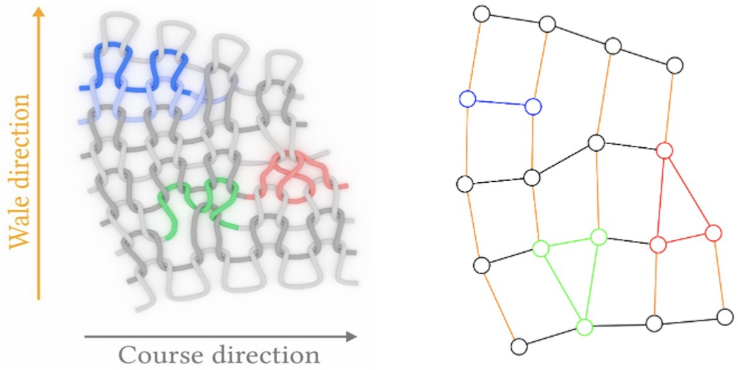
1 BOSTON UNIVERSITY

2 Massachusetts Institute of Technology

Knit Graph Abstraction

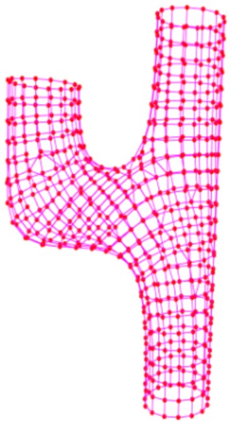


Flat geometry[1]



Stitch irregularities induce curvature[2]

Must be helix-free!



Goal: Given a 3D mesh, generate a **helix-free** knit graph over it

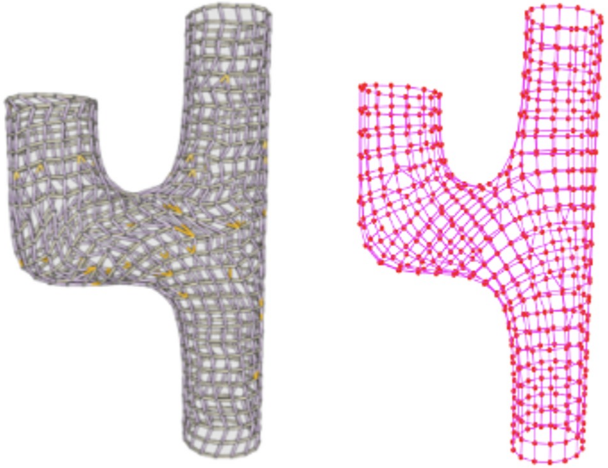
[1] from ***Visual Knitting Machine Programming (2019)***
[2]from ***Knit Sketching: from Cut and Sew Patterns to Machine-Knit Garments (2021)***

Positioning our work

- Our goal (similar to Autoknit [Naryanan et al. 2018]):
  - Automatic generation of machine-knittable graphs from input triangle meshes
  - **Precise control over helices**

- 1-form-based framework produces more globally-informed stitch patterns

- Optimization allows incorporation of linear user-specified constraints



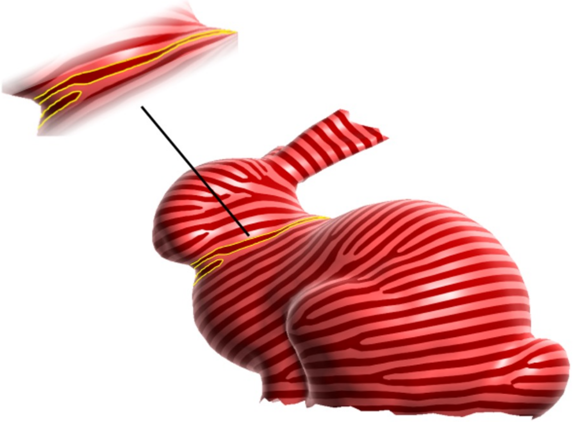
1

[1]from ***https://github.com/textiles-lab/autoknit***

Positioning our work

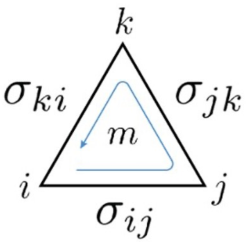
- We follow a stripes-based methodology
  - Evenly-spaced stripes ↔ evenly-spaced course rows and wale columns

- KnitKit [Nader et al. 2021] is only other work to consider this
  - They intersect stripes produced by [Knoppel et al. 2015], which often contain helices
  - Removal attempted via quad mesh operations [Bommes et al. 2011], but no guarantees
  - Our constraints can be used to guarantee helix removal

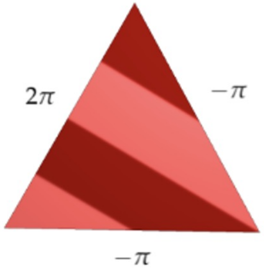


σ : The Stripe Texturing Function

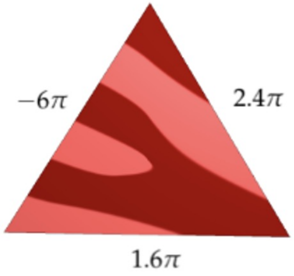
- 1 form $\sigma : E \to \mathbb{R}$ a discretization of a vector field



$$(d_1\sigma)_m = \sigma_{ij} + \sigma_{jk} + \sigma_{ki}$$



- If $(d_1\sigma)_m = 0$ vector field is integrable to a local linear function
  - Striping red if pixel value $\in (0, \pi)(mod\ 2\pi)$, pink if value $\in (\pi, 2\pi)(mod\ 2\pi)$



- If $(d_1\sigma)_m = (2\pi)k, k \in \mathbb{Z}^{|F|}$, vector field non-integrable
  - Can still get local function from triangle to S^1
  - Allows for global striping

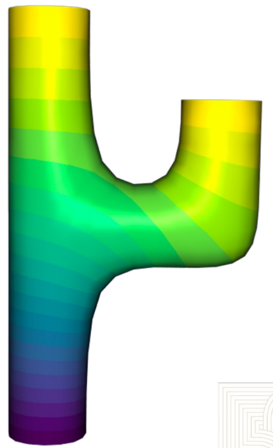
Optimizing for σ directly

- We optimize directly for σ
  - [Noma et al. 2022] optimize in the space of 1-forms but do not explore knitting applications

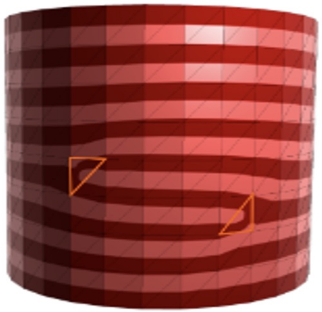
- The gradient of a harmonic interpolation $h : V \to \mathbb{R} \in [0,1]$ guides the 1-form optimization
  - **Linear constraints** achieve all the desiderata

$$\underbrace{||W(\sigma_c - \omega_c)||^2}_{\text{Quadratic minimization objective for course rows}}$$

$$(\omega_c)_{ij} = \underbrace{\frac{1}{2}\left(\frac{(\nabla h)_l}{||(\nabla h)_l||} + \frac{(\nabla h)_r}{||(\nabla h)_r||}\right) \cdot \mathbf{e}_{ij}}_{\text{integral of normalized } \nabla h \text{ along } e_{ij}}$$

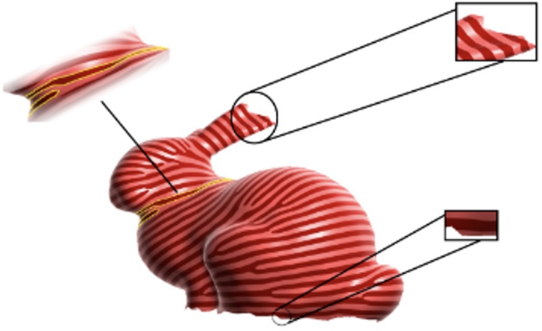


Constraints Examples

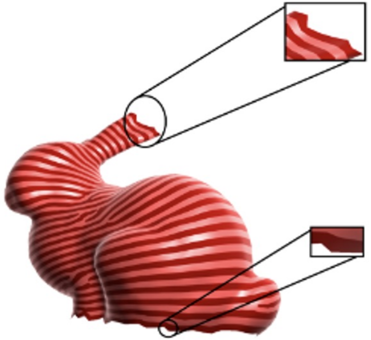


Stripe irregularity/short row placement

$$(d_1\sigma)_m = (2\pi)k, k \in \mathbb{Z}^{|F|}$$



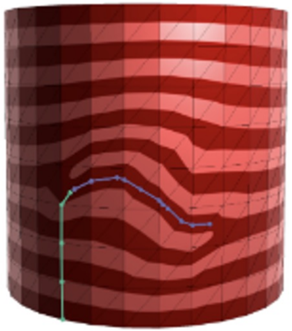
Direct use of [Knoppel et al. 2015]



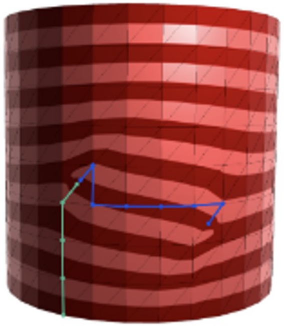
Tightness to boundaries through stripe alignment

$$\sigma_c|_{\partial M} = 0$$

Constraints Examples

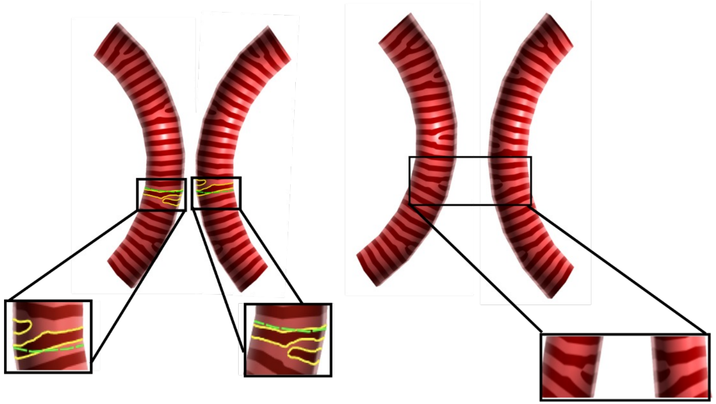


Stripe alignment constraint (blue)



$\int_{\gamma_{\text{open}}} \sigma = 0$

Open level set constraint (dark blue)



$\int_{\gamma_{\text{closed}}} \sigma = 0$

Separating helix ends with γ loops eliminates them

This level of granular control not achievable with **linear constraints** in [Knoppel et al. 2015]

Optimization problems and strategies

SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

Both: Quadratic Mixed Integer problems with linear constraints

Missing: Wale constraints

Strategy 1:

$$\begin{aligned} \min_{\sigma_c} \quad & \|W(\sigma_c - \omega_c)\|^2 \\ \text{subject to} \quad & \sigma_c\big|_{\partial M} = 0, \\ & d_1\sigma_c = (2\pi)\mathbf{k} \\ & + \textbf{linear user constraints} \end{aligned}$$

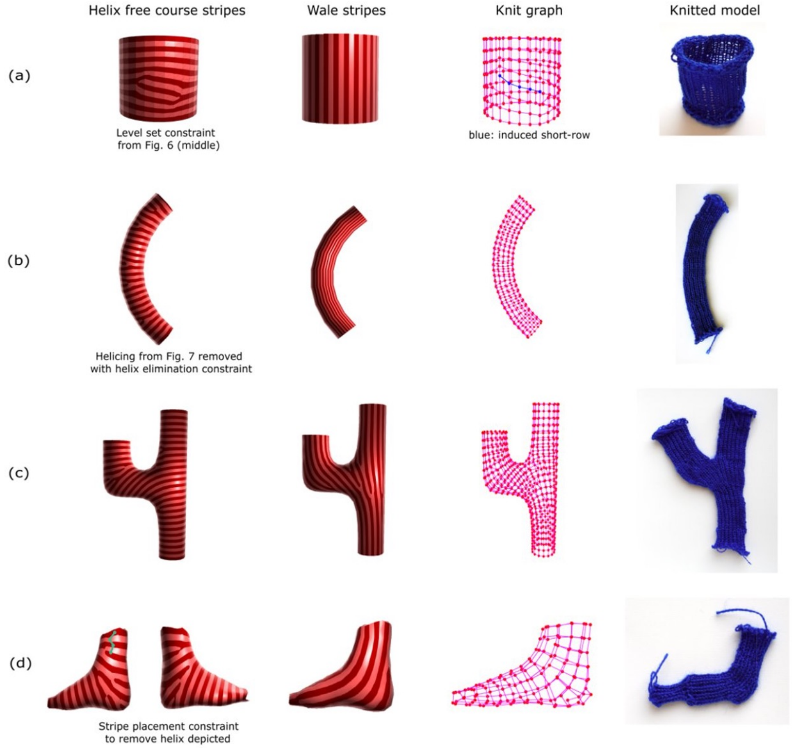
Integer variables ~ O(#boundaries)

Strategy 2:

$$\begin{aligned} \min_{\sigma_c, \mathbf{k}} \quad & \|W(\sigma_c - \omega_c)\|^2 \\ \text{subject to} \quad & \sigma_c\big|_{\partial M} = 0, \\ & d_1\sigma_c = (2\pi)\mathbf{k} \\ & + \textbf{linear user constraints} \end{aligned}$$

Integer variables ~ O(#faces)

Fabricated Results



Thanks to Yuxuan Mei in Adriana Schulz's group at UW for help with fabricating!

Conclusion

- Presented a **stripes-based framework** for global stitch structure specification, operating in the space of **discrete 1-forms**

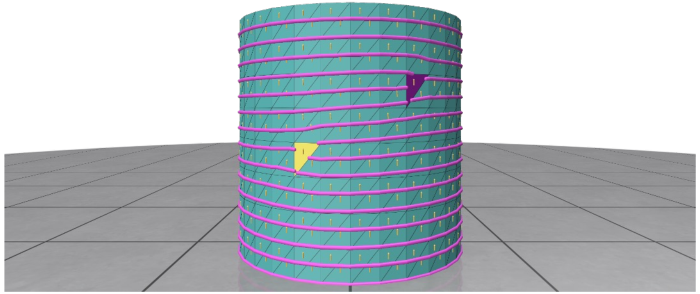
- Results in quadratic mixed-integer optimization with **linear** constraints for:
    - **Removal of helices**
    - Placement of stitch irregularities
    - Alignment of course rows and wale columns to feature curves

- Constraints are directly incorporated into a **global structure optimization**
    - Most previous works incorporate user constraints via post-processing

Future and ongoing work

- We would like to develop a UI for this framework
  - Inspired by many amazing UI-focused works: [Naryanan et al. 2019], [Jones et al. 2022], [Kaspar et al. 2019, 2021]
  - Or incorporate some of our insights into these works

- A more efficient strategy for stitch irregularity placement



- Joint optimization of course rows and wale columns

- Development of a fully form-based pipeline, which allows for exact helicing
  - Potentially avoids the need for a knit graph tracing procedure

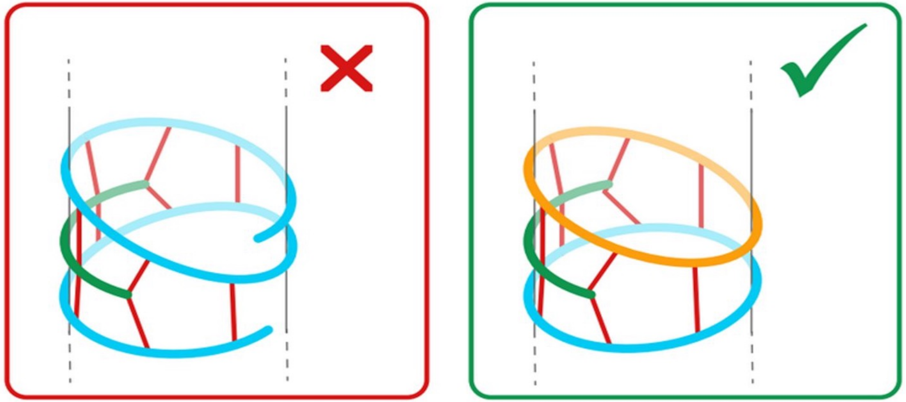
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

Thank you for listening!

(Slides that follow are for clarifying question)

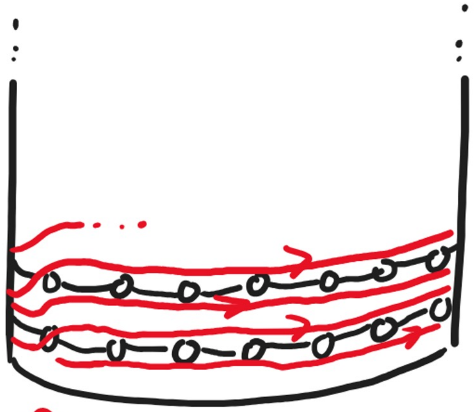
Why do knit graphs need to be helix-free?

- Knitting schedulers [Naryanan et al. 2018] trace the graph in a natural spiralling/helical pattern
  - Helices in the generated graph would result in an incorrect amount (i.e., too much) helicing

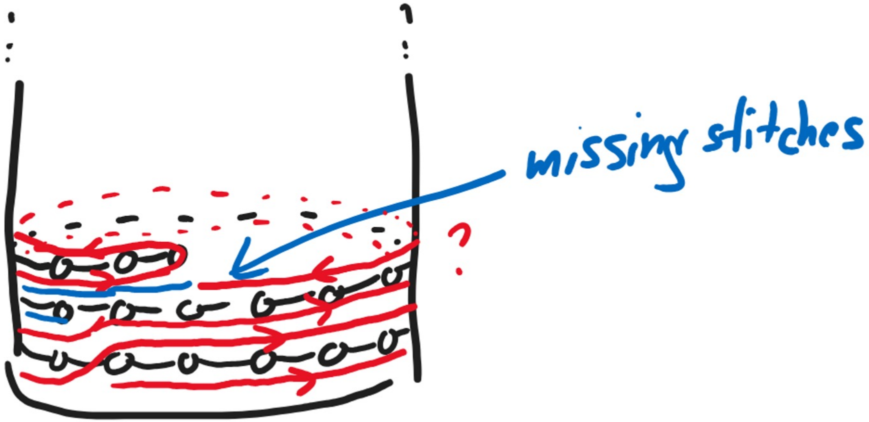


[1]from **Automatic Machine Knitting of 3D Meshes(2018)**

Why is helicing bad?



jumps up a course
row when it returns
to achieve helicing

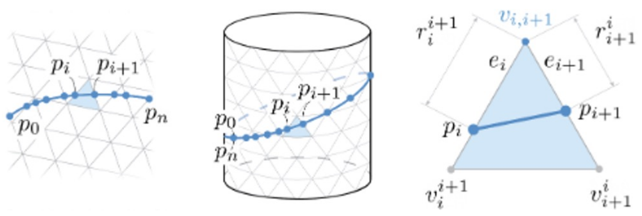


Upshot: helicing by the
exact right amount is hard

Level Set (path integral constraints)

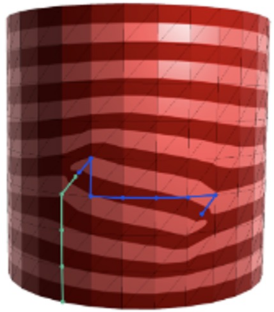
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

- Can specify path integral of σ along a specified polyline γ
  - Constrains same stripe level set to pass through polyline endpoints
  - Non-trivial constraints applied to the wale stripes to ensure stripes "close up" on the boundary



$$\int_\gamma \sigma = \sum_{i=0}^{n-1} -s_i^{i+1}\sigma_{e_i} r_i^{i+1} + s_{i+1}^i \sigma_{e_{i+1}} r_{i+1}^i = C, C \in \mathbb{R}$$

- When $C = 0$ and γ is a closed loop of mesh edges, ensures that a stripe level set does not cross the loop without returning back to its side of origin i.e., helix removal



Level set constraint in blue

Constraints in the wale direction

$$(\omega_w)_{ij} = \frac{1}{2}\left(\frac{(\nabla h)_l^T}{||(\nabla h)_l^T||} + \frac{(\nabla h)_r^T}{||(\nabla h)_r^T||}\right) \cdot \mathbf{e}_{ij}$$

$$\int_{(\partial M)^i} \sigma_w = (2\pi)k^i$$

Specified on N-1 boundaries

- Need to only consider irregularity placement and possible stripe alignment constraints
  - No helices in the wale direction

Runtime statistics

Table 1: Run time statistics.

Model	Strategy	#V	#F	# Int. Vars.	MIP Solve Time, s
Curved Cylinder (Fig. 7)	S2	54	96	97	< 5
Hemisphere (Fig. 9(e))	S2	224	384	385	40
Sock (Fig. 1)	S2	279	538	98	21
Sock (Fig. 9(d))	S1	279	538	2	< 1
Cylinder (Fig. 6, middle)	S1	270	580	3	< 1
Cactus (Fig. 9(c))	S2	391	736	738	35
Bunny (Fig. 4, right)	S2	2669	5228	5230	287

Skipping tracing & exact helicing example