# Surfacing curve networks with normal control

Tibor Stanko [1,2,3]
Stefanie Hahmann [2,3]
Georges-Pierre Bonneau [2,3]
Nathalie Saguin-Sprynski [1,2]

[1] CEA, LETI, MINATEC Campus
[2] Université Grenoble Alpes
[3] CNRS (Laboratoire Jean Kuntzmann), INRIA

## Context & motivation

Shape acquisition via microsensors developed at CEA LETI is an aproach for the digitization of real objects, providing alternative to traditional 3D scanners.

The sensors measure a sequence of orthonormal frames along curves on a surface. Combined with the distances between adjacent sensor nodes, the data is used for reconstruction of the scanned network. Given such network, the goal of this work is to find the underlying surface.

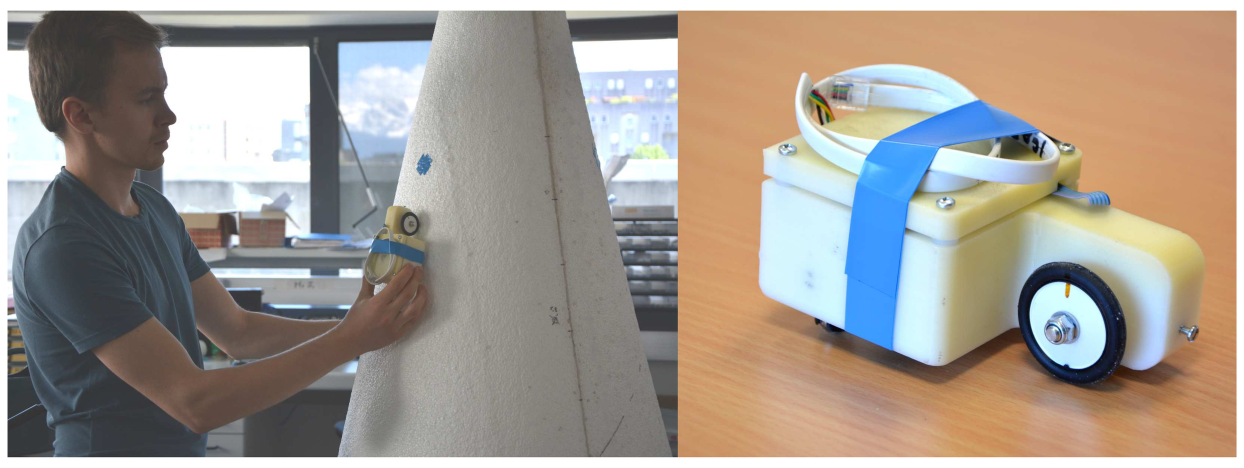


**Fig. 1**: Morphorider, a mouse-like device used for scanning surface curves. It contains a single node of sensors, and the distance is measured dynamically during acquisition.

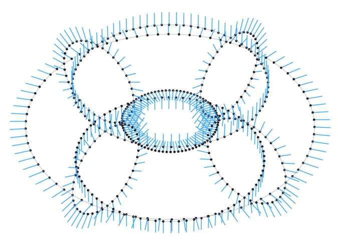
## Algorithm overview

We present mesh-based variational algorithm for surfacing arbitrary closed curve network acquired from a surface. The key ingredient is a classical differential geometry formula

$$\Delta \mathbf{v} = -\mathrm{H}\mathbf{n}$$

Laplacian of position = - mean curvature normal

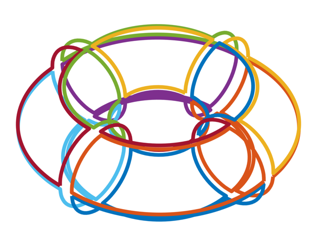
which enables compact combination of input positions and normals.



### Input data

The input is a discrete set of points with surface normals, sampled along the curve network. If needed, raw data are resampled with uniform geodesic edge length.

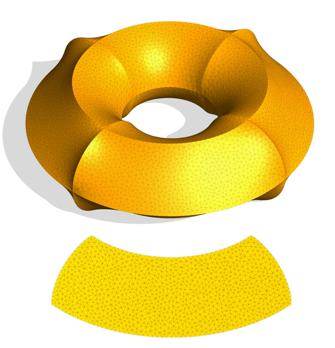
← resampled input network with surface normals



### Cycles detection

Finding cycles in a general network is a difficult problem. Exploiting our knowledge of surface normals, we perform this detection efficiently. The detected cycles then define individual surface patches.

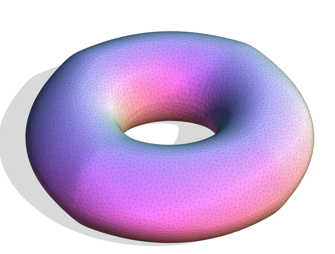
← network with detected cycles



### Meshing

With no prior mesh available, each cycle is projected into plane and triangulated. Joining individual triangulations, we obtain the topology of mesh $\mathcal{M}$. Weights for the discrete Laplacian are computed from planar triangulations.

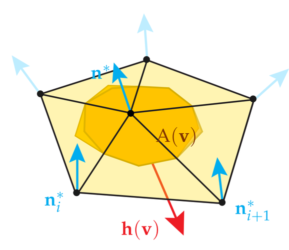
↑ tessellated cycles mapped back onto the 3D network
← planar triangulation of one of the cycles



### Initial geometry

Initial positions $\mathbf{v}^*$ and normals $\mathbf{n}^*$ for the whole mesh $\mathcal{M}$ are computed by solving two bilaplacian systems $\Delta\mathbf{v}^* = \mathbf{0}$, $\Delta\mathbf{n}^* = \mathbf{0}$ while constraining vertices along the input curves.

← mesh with initial positions colored by initial normals

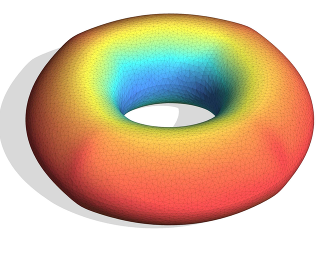



### Curvature estimation

To combine positional and normal information, mean curvature at vertex $\mathbf{v}$ with neighbours $\{\mathbf{v}_i\}_{i=1}^n$ and Voronoi area $\mathrm{A}(\mathbf{v})$ is estimated using

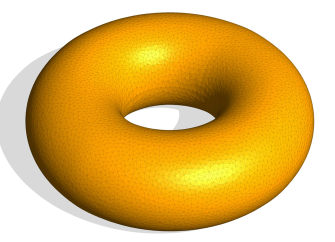
$$2\mathbf{h}(\mathbf{v}) = \oint_{\partial \mathrm{A}(\mathbf{v})} \mathbf{n} \times \mathrm{ds}$$

which we discretize via the formula



$$\mathbf{h}(\mathbf{v}) = \frac{1}{2\mathrm{A}(\mathbf{v})} \sum_{i=1}^{n} \text{normalized}\left(\mathbf{n}^* + \mathbf{n}_i^* + \mathbf{n}_{i+1}^*\right) \times \left(\mathbf{v}_{i+1}^* - \mathbf{v}_i^*\right)$$

↑ situation around mesh vertex
← mesh with initial positions colored by estimated mean curvature



### Optimization

Finally, the vertex positions are optimized by minimizing the energy

$$\mathrm{E}(\mathcal{M}) = \sum_{\mathbf{v}\in\mathcal{M}} \left\|\Delta\mathbf{v} + \mathbf{h}^*\right\|^2 \quad \text{with} \quad \mathbf{h}^* = -\left\|\mathbf{h}(\mathbf{v})\right\| \mathbf{n}^*$$

← final optimized mesh

## Results & comparison to previous methods

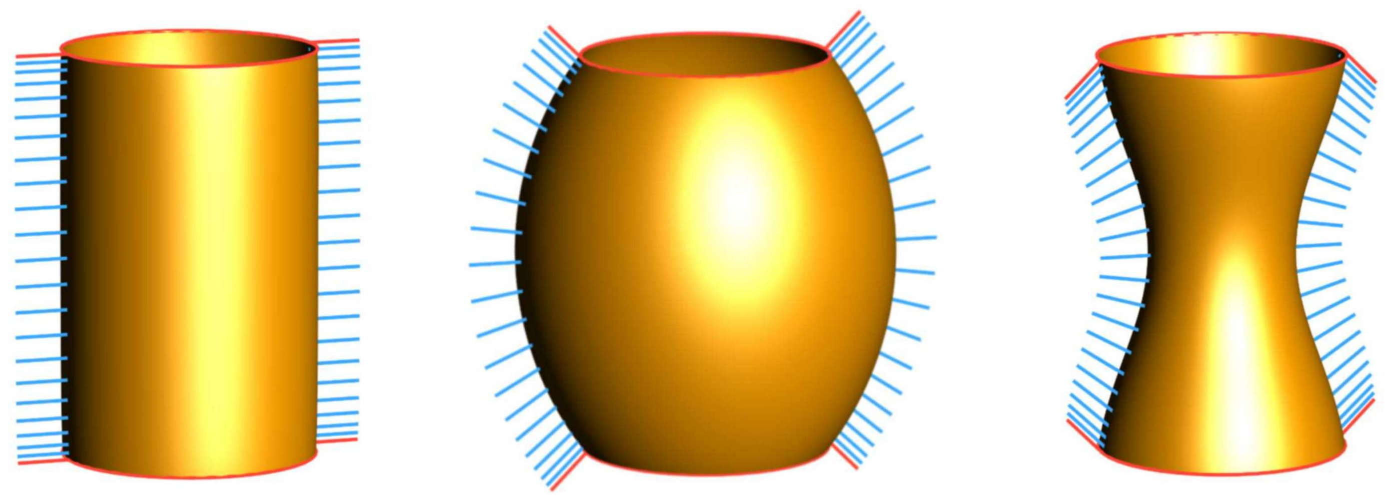


**Fig. 2**: Influence of input normals. Two circles (red) are given as input together with normal vectors of different orientation (red normals). Propagation of the input normals over the surface (blue) guides the computation of three different shapes.

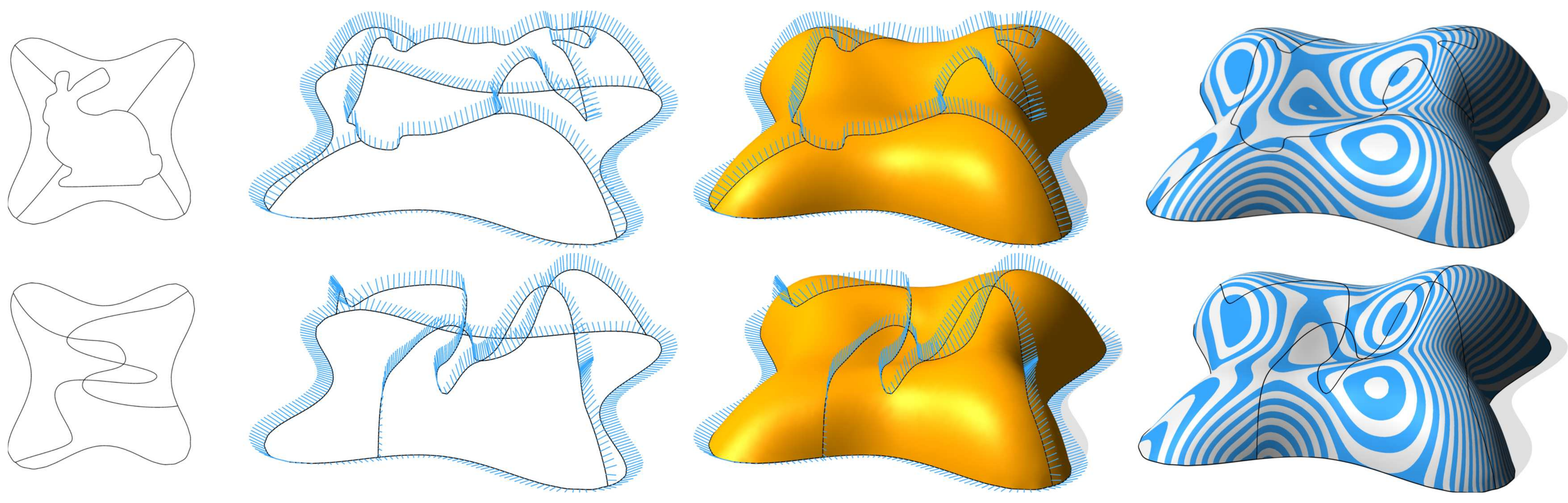


**Fig. 3**: In terms of geometry, we place no constraints on the input. Our method can handle even these challenging networks with winding curves, all while preserving the global smoothness across patches.

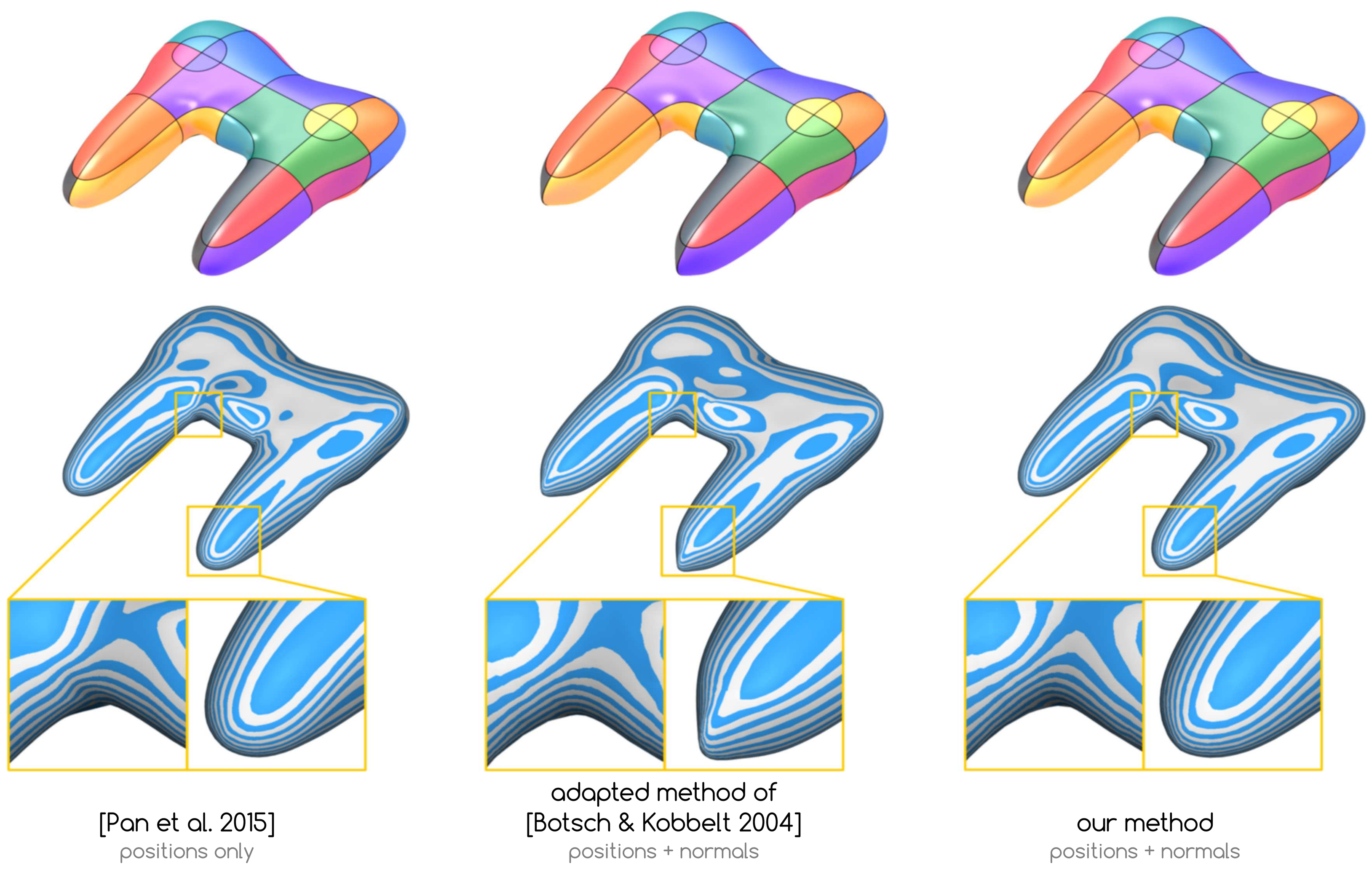



**Fig. 4**: On sketched networks, the results of our algorithm are similar to the method of [Pan et al. 2015], which assumes input curves capture flow field of the underlying surface. The isophotes on our surface vary more smoothly, suggesting higher order of continuity.

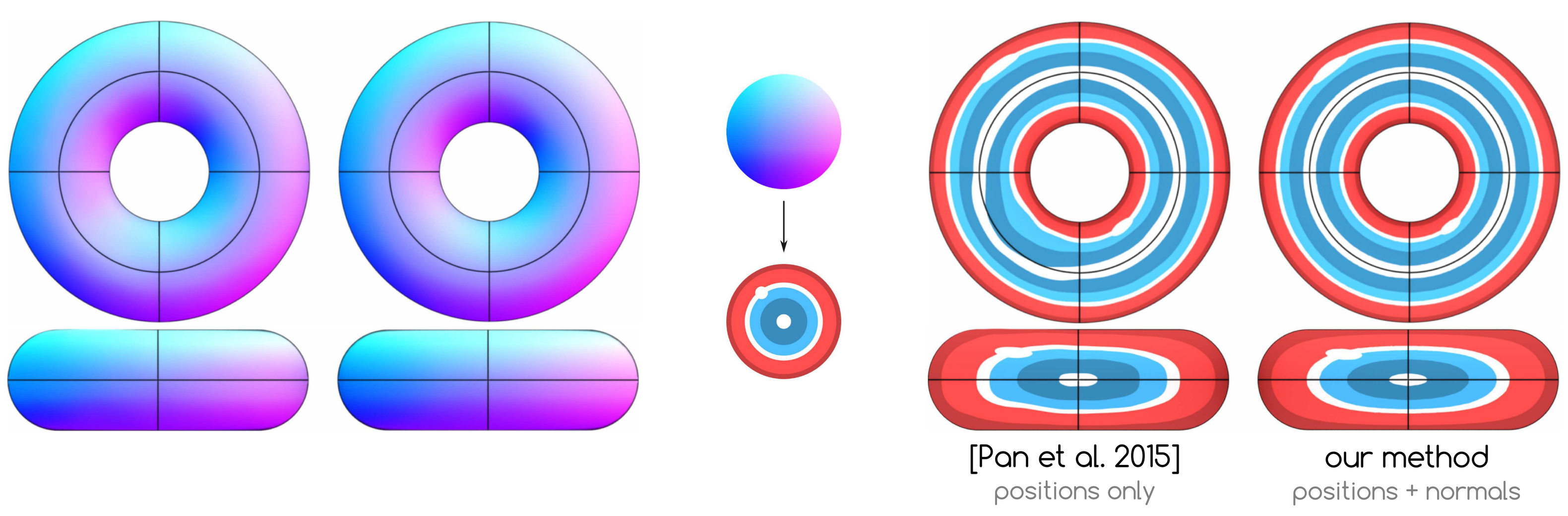



**Fig. 5**: Normal mapping with red-and-blue texture reveals important details about the surfaces: unlike [Pan et al. 2015], our method succeeds in preserving symmetry of the input network in the final surface.

## Conclusions

The presented framework is intended to serve for curve networks with normals acquired by devices equipped with microsensors. Since our current implementation runs at interactive time rates (order of 0.1s for a mesh with 10k vertices and 1k constraints), we plan to allow users to scan shapes interactively by incrementally adding curves; we therefore want to investigate how to update the optimization when the input data changes locally.

## References

This research was partially funded by the ERC advanced grant no. 291184 EXPRESSIVE.

**[Botsch, Kobbelt 2004]** Botsch M, Kobbelt L. An Intuitive Framework for Real-time Freeform Modeling. ACM Trans. Graph. (Proc. SIGGRAPH) 2004;23(3):630–4

**[Pan et al. 2015]** Pan H, Liu Y, Sheffer A, Vining N, Li CJ, Wang W. Flow-Aligned Surfacing of Curve Networks. ACM Trans. Graph. (Proc. SIGGRAPH) 2015;34(4):127:1–127:10.

**[Saguin-Sprynski 2014]** Saguin-Sprynski N, Jouanet L, Lacolle B, Biard L. Surfaces Reconstruction Via Inertial Sensors for Monitoring . In: 7th European Workshop on Structural Health Monitoring. 2014.

**[Sullivan 2008]** Sullivan J. Curvatures of Smooth and Discrete Surfaces. In: Bobenko A, Schroder P, Sullivan J, Ziegler GM, eds. Discrete Differential Geometry, Oberwolfach Sem. Vol. 38. Birkhauser Basel; 2008, p. 175–88.