Smooth Interpolation of Curve Networks with Surface Normals

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

[1]CEA, LETI, MINATEC Campus, Grenoble [2]Univ. Grenoble Alpes [3]CNRS (LJK), Inria

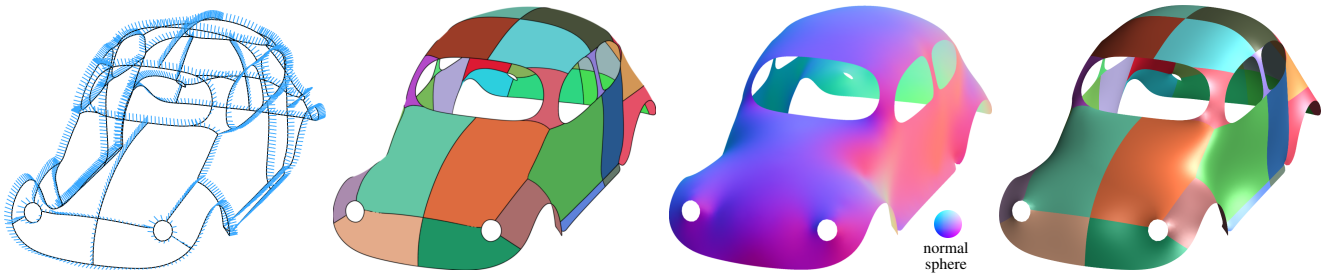

Figure 1: Smooth interpolation of the beetle network with surface normals. The cycles in the input network (left) are efficiently extracted and tessellated (middle left). The propagated normals (middle right) are used to compute the resulting globally smooth surface (right).

Abstract

Recent surface acquisition technologies based on microsensors produce three-space tangential curve data which can be transformed into a network of space curves with surface normals. This paper addresses the problem of surfacing an arbitrary closed 3D curve network with given surface normals. Thanks to the normal vector input, the patch finding problem can be solved unambiguously and an initial piecewise smooth triangle mesh is computed. The input normals are propagated throughout the mesh and used to compute mean curvature vectors. We then introduce a new variational optimization method in which the standard bi-Laplacian is penalized by a term based on the mean curvature vectors. The intuition behind this original approach is to guide the standard Laplacian-based variational methods by the curvature information extracted from the input normals. The normal input increases shape fidelity and allows to achieve globally smooth and visually pleasing shapes.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1 Introduction & Related Work

Traditionally, digital models of real-life shapes are acquired with 3D scanners, providing point clouds for surface reconstruction algorithms. However, there are situations when 3D scanners fall short, e.g. in hostile environments, for very large or deforming objects. In the last decade, alternative approaches to shape acquisition using data from microsensors have been developed [SDLB07, HS08]. Small size and cost of these sensors facilitate their integration in numerous manufacturing areas; the sensors are used to obtain information about the equipped material, such as spatial data or deformation behavior. Ribbon-like devices incorporated into soft materials [SSJLB14] or instrumented mobile devices moving on the surface of an object provide tangential and positional data along geodesic curves. In this context, we focus only on the resulting surface reconstruction problem and leave aside all problems related to acquisition and transformation of sensor signals into geometric data.

We address the problem of smooth surface reconstruction given discrete positional and normal data along a network of 3D curves. The goal is to obtain a fully automatic, efficient and robust method. We propose a new mesh-based variational approach which first searches for the topological patches, then computes a visually pleasing surface while maintaining a high fidelity with the underlying object. Our method is based on the insight that the combination of shape and normal optimization into a compact expression does not require the usual reformulation of normal constraints into layers of positional boundary constraints.

Surfacing curve networks. With the advent of sketch-based modeling tools, considerable effort has been dedicated to the design of methods for surfacing curve networks originating from sketching

DOI: 10.2312/egsh.20161005

tools [ZZCJ13, PLS*15, BWSS12, SS14]. The common assumption in these works is the underlying curve network was created with some design intent, and the input information is minimal. Our work shares the goal of shaping a surface from a network of curves. As we assume more input data, some parts of the process are easier, such as finding cycles in the network of curves, but we tackle the complementary problem of providing more control over the final shape.

Variational modeling with normal constraints. The minimum variation surfaces [MS92] enable direct prescription of normals and principal curvatures along a curve network; however, the resulting optimization is highly nonlinear. The boundary constraint modeling methods of Botsch and Kobbelt [BK04], Jacobson et al. [JTSZ10], prescribe C^k continuity indirectly by fixing $k+1$ rings of vertices. It is however not clear how to set additional rings of vertices consistently with C^k continuity at the intersection of constrained curves. This issue, referred to as twist compatibility problem or vertex consistency problem [Far82], arises when joining smooth patches around a common vertex of arbitrary valence with tangent plane continuity. In contrast, our goal is to deal with normal constraints even at the intersection of multiple curves.

Contributions. In this paper, we describe a framework for surface reconstruction from a sample of points and normals along a curve network. We introduce a new variational method in which the standard bi-Laplacian is penalized by a term based on the mean curvature vectors. By extracting the curvature information from the input normals propagated throughout the mesh, we achieve a better control over the resulting shape.

2 Framework

The surface $\mathcal{S}$ we aim to reconstruct is a connected 2-manifold, with or without boundary, parameterized by $\mathbf{p} : \Omega \subset \mathbb{R}^2 \to \mathcal{S} \subset \mathbb{R}^3$. Moreover, the tangent space $T_{\mathbf{p}}(\mathcal{S})$ varies continuously. Next, we consider a curve network $\mathcal{C} \subset \mathcal{S}$ which is connected and closed. The curves $c_i \in \mathcal{C}$ are C^1 smooth and without self-intersections; the intersection of two different curves is either empty or a discrete set of points.

Overview of the method. We use the following pipeline to generate a globally smooth surface from curve and normal vector input (Fig. 1):

1. Raw data are first interpolated with cubic splines and resampled uniformly. We efficiently detect the network cycles, then triangulate them in plane.
2. By solving two biharmonic systems with boundary constraints, we propagate the surface normals and obtain an initial guess for the vertex positions; this allows us to compute discrete mean curvature for the whole mesh.
3. Finally, we solve a constrained quadratic optimization, minimizing an energy functional which takes into account the estimated mean curvatures.

Definitions. A *node* n is the intersection between two or more curves c_i. A *segment* is a portion of curve bounded by two adjacent nodes. A *cycle* is a set of adjacent segments which constitute a boundary of some surface patch. In the following section we present an algorithm for detection of such cycles, exploiting our knowledge of surface normals.

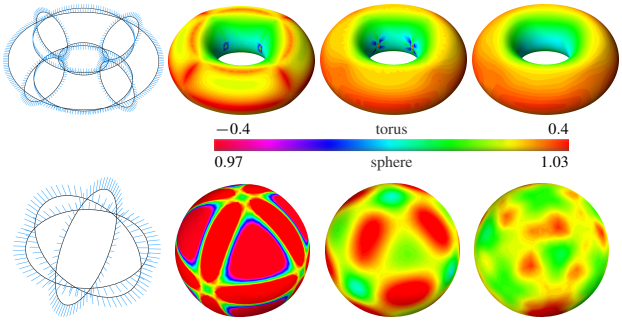

Figure 2: Comparison of our method with biharmonic and triharmonic surfaces. Left, middle: surfaces were computed by solving $\Delta^2\mathbf{v} = 0$ and $\Delta^3\mathbf{v} = 0$ with hard constraints and without fixing the normals. Right: result of our optimization using soft constraints. Both torus and sphere are colored by the discrete mean curvature.

2.1 Exploiting local tangent space to detect cycles

The detection of cycles in a general curve network is a complex and ambiguous problem, which often does not have a unique solution. In order to overcome this problem, methods for surfacing sketched networks adopt a variety of heuristics to mimic the human perception [ZZCJ13]. In our specific setting, due to the assumptions on surface smoothness and manifoldness and thanks to the availability of the oriented normals, any possible ambiguity can be efficiently resolved. The algorithm is inspired by face extraction in edge-based manifold data structures. First the segments adjacent to any node are cyclically sorted with respect to the orientation given by the input normal at that node. Then, starting from any (Node, Segment) pair, we trace a unique cycle by choosing the next node as the other endpoint of the current segment; the next segment is then picked from the ordered set. To handle surfaces with boundary, we require the user to tag the boundary segments.

2.2 Network tessellation

We represent the surface $\mathcal{S}$ as a triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ with vertices $\mathcal{V}$ and faces $\mathcal{F}$. Prior to the tessellation, the given positions and normals along the curve network $\mathcal{C}$ are interpolated with cubic splines and resampled with arc length parameterization, to obtain a curve network such as the one shown in Fig. 1. Each cycle defines a closed 3D curve Γ bounding an n-sided surface patch. We triangulate a planar projection of each cycle individually to obtain the topology $\mathcal{F}$ of the whole mesh. The planar triangulation is computed using the Triangle tool [She96]. The plane of projection is defined by the average position $\tilde{\mathbf{p}}$ and average unit normal $\tilde{\mathbf{n}}$ computed from resampled Γ. Notice that a more robust but time-consuming 3D curve tessellation method can be used [ZJC13].

2.3 Variational smoothing

At this point of the process we have computed the topology $\mathcal{F}$ of the mesh $\mathcal{M}$, and we have the input positions and normals for vertices along the curve network $\mathcal{C}$. In this section, we describe a variational method for computing the positions of the free vertices, based on the discretization of the Laplace-Beltrami operator and of the mean curvature vector for piecewise linear surfaces.

Discretization of Δ. Given a piecewise-linear function $f_i = f(\mathbf{v}_i)$ defined over the vertices $\mathbf{v}_i \in \mathcal{V}$ of $\mathcal{M}$, the discretization of the Laplace-Beltrami has the form [BS08]

$$\Delta f(\mathbf{v}_i) = w_i \sum_{j \in N_1(i)} w_{ij}(f_j - f_i)$$

where $N_1(i)$ is the index set of 1-ring neighborhood of $\mathbf{v}_i$. The vertex weights are stored in the diagonal mass matrix $\mathbf{M}_{ii} = 1/w_i$, while the edge weights w_{ij} are stored in a symmetric matrix $\mathbf{L}_s$

$$(\mathbf{L}_s)_{ij} = \begin{cases} -\sum_{k \in N_1(i)} w_{ik}, & i = j, \\ w_{ij}, & j \in N_1(i), \\ 0, & \text{otherwise.} \end{cases}$$

The discrete Laplacian operator is then characterized by the matrix $\mathbf{L} = \mathbf{M}^{-1}\mathbf{L}_s$. In the following, we use the cotangent Laplacian $w_i = 1/A_i, w_{ij} = \frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij})$ where α_{ij} and β_{ij} are the two angles opposite to the edge (i,j), and A_i is the Voronoi area of $\mathbf{v}_i$ [MDSB03].

Mean curvature vectors. To guide the optimization, we compute the mean curvature vectors for all vertices. Let $\mathcal{V}_c$ denote the set of vertices lying on the curve network $\mathcal{C}$, and $\mathcal{V}_f$ denote the remaining free vertices. We start by computing an initial position and an initial normal for all vertices by solving two biharmonic systems $\mathbf{L}^2\mathbf{V}^* = \mathbf{0}$ for positions and $\mathbf{L}^2\mathbf{N}^* = \mathbf{0}$ for normals. The *propagated normals* $\mathbf{N}^*$ are then normalized. We choose $\mathbf{L}$ as the cotangent Laplacian based on the planar triangulation computed in Section 2.2. The boundary conditions $\mathcal{V}_c$ are incorporated into the system as hard constraints by eliminating the corresponding rows of the matrix $\mathbf{L}^2$ as described in [BKP*10]. Following [Sul08], the discrete mean curvature vector at the vertex $\mathbf{v}_i$ is proportional to the integral of the co-normal η, i.e. the vector product of the normal and the unit tangent to the boundary

$$2\mathbf{h}(\mathbf{v}_i) = \oint_{\partial N_1(i)} \eta \, ds$$

computed along the boundary of the 1-neighborhood of $\mathbf{v}_i$. In order to take the input normals into account, we evaluate this integral using the propagated normals $\mathbf{N}^*$ rather than the triangle normals. More precisely, we compute the mean curvature vector for the initial surface as follows:

$$2\mathbf{h}(\mathbf{v}_i) = \sum_{(j,k) \in \mathrm{e}(i)} \frac{\mathbf{n}_j^* + \mathbf{n}_k^*}{\|\mathbf{n}_j^* + \mathbf{n}_k^*\|} \times (\mathbf{v}_k - \mathbf{v}_j),$$

where $\mathbf{n}_j^*$ denotes the propagated normal at the vertex $\mathbf{v}_j$, $\times$ denotes the vector product, and $\mathrm{e}(i)$ is the set of oriented edges (j,k) opposite to the vertex $\mathbf{v}_i$.

Optimization. The constrained vertices $\mathcal{V}_c$ are further partitioned into *hard constraints* $\mathcal{V}_b$ and *soft constraints* $\mathcal{V}_s$. The set of soft constraints might be empty if we require exact interpolation of all positions; the soft formulation becomes convenient if the input data are subject to noise. Without loss of generality, we assume the index sets $I_b = \{1,\ldots,b\}$, $I_s = \{b+1,\ldots,b+s\}$, $I_f = \{b+s+1,\ldots,b+s+f\}$ correspond to hard, soft and free vertices, respectively. We can now define the energy functional

$$E(\mathcal{V}) = \sum_{\mathbf{v} \in \mathcal{V}} \|\Delta^2\mathbf{v}\|^2 + \omega_1 \sum_{\mathbf{v} \in \mathcal{V}} \|\Delta\mathbf{v} + 2h(\mathbf{v})\mathbf{n}^*\|^2 + \omega_0 \sum_{\mathbf{v}_s \in \mathcal{V}_s} \|\mathbf{v}_s - \mathbf{v}_s^*\|^2$$

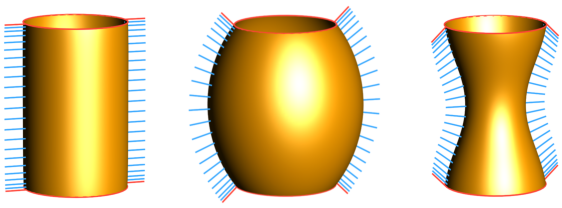

Figure 3: Influence of input normals: the two circular curves (red) are given as input together with normal vectors of different orientation (red normals). Propagation of the input normals over the surface (blue) guides the computation of three different shapes.

were $\omega_i \in \mathbb{R}^+$ are weights, $h(\mathbf{v}) = \|\mathbf{h}(\mathbf{v})\|$ is the scalar mean curvature at $\mathbf{v}$ and $\mathbf{n}^*$ is the propagated normal at $\mathbf{v}$. The first term forces the mesh to be as-biharmonic-as-possible; the last term pushes the result toward the soft constraints. The middle term, summed over all vertices, enables to match the mean curvature and the propagated normals and is derived from the well-known formula $\Delta\mathbf{v} = -2h\mathbf{n}$.

In order to exactly interpolate the hard constraints $\mathbf{v}_b^*$, we perform the following optimization:

$$\min E(\mathcal{V}) \quad \text{s.t.} \quad \mathbf{v}_b = \mathbf{v}_b^*,\ b \in I_b\,.$$

The energy E is written in matrix form as

$$E(\mathcal{V}) = \|\mathbf{L}^2\mathbf{V}\|^2 + \omega_1\|\mathbf{L}\mathbf{V} - 2\mathbf{H}\|^2 + \omega_0\|\tilde{\mathbf{I}}\mathbf{V} - \mathbf{V}_s^*\|^2$$

and minimized by solving

$$\begin{bmatrix} \mathbf{W}\mathbf{A}^\top\mathbf{A} & \mathbf{C}^\top \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{W}\mathbf{A}^\top\mathbf{B} \\ \mathbf{V}_b^* \end{bmatrix}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{L}^2 \\ \mathbf{L} \\ \tilde{\mathbf{I}} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{H} \\ \mathbf{V}_s^* \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{I}_b & \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_s & \mathbf{0} \end{bmatrix},$$

and Λ is the matrix of Lagrange multipliers, $\mathbf{W}$ is the matrix of weights ω_i, $\mathbf{I}_k$ is the $k \times k$ identity matrix, and $\mathbf{H}$ is the matrix of propagated normals $\mathbf{N}^*$ scaled by the mean curvature h.

3 Results

The sphere and torus curve networks on Fig. 2 allow us to compare our method with surfaces obtained by solving biharmonic (L2) and triharmonic (L3) systems $\Delta^k\mathbf{v} = 0$, $k = 2,3$, using hard positional constraints and no normal constraints. The discrete mean curvature plots reveal interesting details; notice the discontinuity on the L2 and L3 torus and on L2 sphere located at the intersections of input curves. On the other hand, our surface is smooth and contains no such singularities.

To demonstrate the control provided by the input normals we sample the positions along two circles from the same cylinder while changing the directions of the normals, see Fig. 3. With the original normals, the cylindrical surface is nicely reconstructed; using the two sets of rotated normals results in the barrel and bottleneck surfaces, as expected intuitively. The method works well even for challenging input data, such as the networks with large curvature variations on Fig. 4.

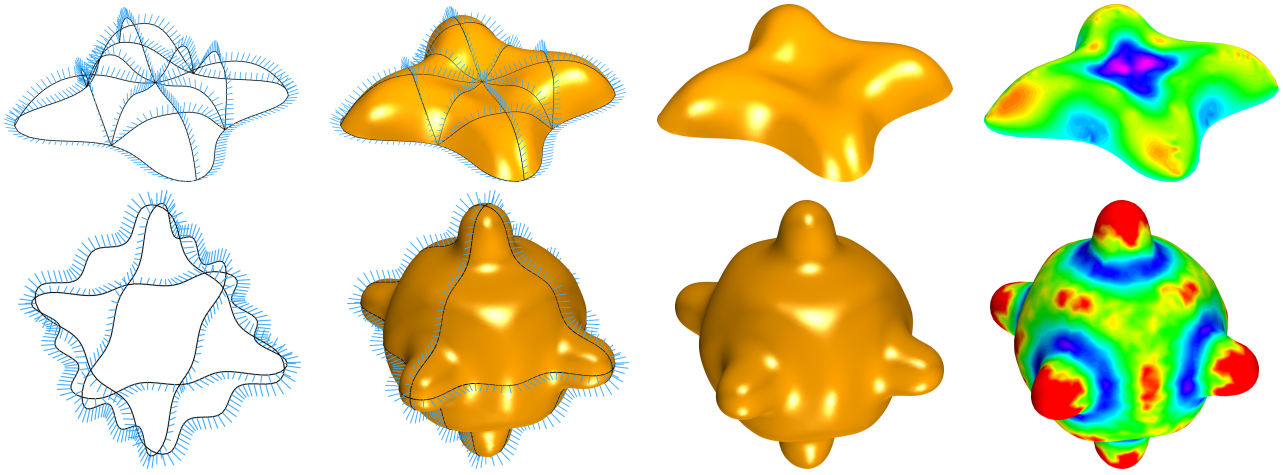

Figure 4: Reconstructing surfaces with strong variations in curvature and high valence intersections in the curve network. (Left) input network with normals, (middle left) reconstructed surface with visualised constraints, (middle right) without visualised constraints (right) mean curvature plot. In these examples, all positional constraints are hard and $\omega_1 = 15000$.

4 Discussion & Future Work

We have introduced a Laplacian-based surface reconstruction method from curve and normal input. After propagating the input normals smoothly over the surface and computing the corresponding mean curvature vectors, the normal constraints are integrated into the energy functional. Efficiency and robustness are achieved by using a linearized objective functional, such that the global optimization amounts to solving a sparse linear system of equations.

The presented framework is intended to serve for curve networks with normal vectors acquired by mobile devices equipped with microsensors. For this application we plan the following extensions. The method, currently requiring a closed curve network, could be modified to work with open curve networks. The initial tessellation could be improved by using a more advanced 3D patching algorithm. Since our current implementation runs at interactive time rates (order of 0.1s for the beetle mesh with 10k vertices and 1k constraints), we plan to allow the user to scan objects interactively by incrementally adding curves. We therefore want to investigate how to update the optimization when the input data changes locally.

Acknowledgements

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

References

[BK04] BOTSCH M., KOBBELT L.: An Intuitive Framework for Real-time Freeform Modeling. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3 (2004), 630–634.

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. CRC press, 2010.

[BS08] BOTSCH M., SORKINE O.: On Linear Variational Surface Deformation Methods. *IEEE Trans. Vis. Comput. Graphics 14*, 1 (2008), 213–230.

[BWSS12] BESSMELTSEV M., WANG C., SHEFFER A., SINGH K.: Design-driven Quadrangulation of Closed 3D Curves. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6 (2012), 178:1–178:11.

[Far82] FARIN G.: A construction for visual c1 continuity of polynomial surface patches. *Computer Graphics and Image Processing 20*, 7 (1982), 272–282.

[HS08] HOSHI T., SHINODA H.: 3D Shape Measuring Sheet Utilizing Gravitational and Geomagnetic Fields. In *SICE Annual Conference* (2008), pp. 915–920.

[JTSZ10] JACOBSON A., TOSUN E., SORKINE O., ZORIN D.: Mixed Finite Elements for Variational Surface Modeling. *Computer Graphics Forum (Proc. SGP) 29*, 5 (2010), 1565–1574.

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete Differential Geometry Operators for Triangulated 2-manifolds. In *Visualization and mathematics III*. Springer, 2003, pp. 35–57.

[MS92] MORETON H. P., SÉQUIN C. H.: Functional Optimization for Fair Surface Design. *Computer Graphics (Proc. SIGGRAPH) 26*, 2 (1992), 167–176.

[PLS*15] PAN H., LIU Y., SHEFFER A., VINING N., LI C.-J., WANG W.: Flow-Aligned Surfacing of Curve Networks. *ACM Trans. Graph. (Proc. SIGGRAPH) 34*, 4 (2015), 127:1–127:10.

[SDLB07] SPRYNSKI N., DAVID D., LACOLLE B., BIARD L.: Curve Reconstruction via a Ribbon of Sensors. In *14th IEEE Int. Conf. on Electronics, Circuits and Systems* (2007), pp. 407–410.

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied computational geometry towards geometric engineering*. Springer, 1996, pp. 203–222.

[SS14] SADRI B., SINGH K.: Flow-complex-based Shape Reconstruction from 3D Curves. *ACM Trans. Graph. 33*, 2 (2014).

[SSJLB14] 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).

[Sul08] SULLIVAN J. M.: Curvatures of smooth and discrete surfaces. In *Discrete Differential Geometry*, Bobenko A., Sullivan J., Schröder P., Ziegler G., (Eds.), vol. 38 of *Oberwolfach Seminars*. Birkhauser Basel, 2008, pp. 175–188.

[ZJC13] ZOU M., JU T., CARR N.: An Algorithm for Triangulating Multiple 3D Polygons. *Computer Graphics Forum (Proc. SGP) 32*, 5 (2013), 157–166.

[ZZCJ13] ZHUANG Y., ZOU M., CARR N., JU T.: A General and Efficient Method for Finding Cycles in 3D Curve Networks. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 32*, 6 (2013), 180:1–180:10.