Shape from sensors: curve networks on surfaces from 3D orientations

Tibor Stanko[a,b], Stefanie Hahmann[b], Georges-Pierre Bonneau[b], Nathalie Saguin-Sprynski[a]

[a]*CEA, LETI, MINATEC Campus, Université Grenoble Alpes*
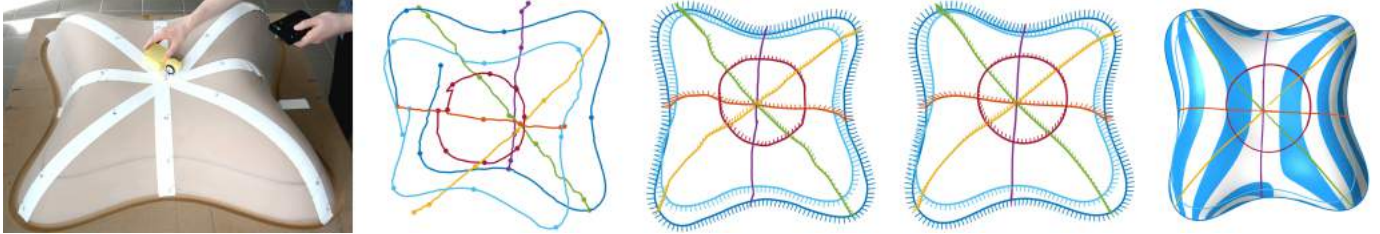[b]*Université Grenoble Alpes, CNRS (Laboratoire Jean Kuntzmann), INRIA*

Figure 1: Surface of the LILIUM is scanned by acquiring orientations along curves using a sensor-instrumented device (left). Naive integration of the scanned data fails to close the network (middle left). Reconstructing the network by solving a Poisson system resolves topological problems but yields noisy and inconsistent normals (middle). Filtering orientations prior to Poisson reconstruction gives a consistent network (middle right) available for surface fitting (right). See the accompanying video for the acquisition setup.

Abstract

We present a novel framework for acquisition and reconstruction of 3D curves using orientations provided by inertial sensors. While the idea of sensor shape reconstruction is not new, we present the first method for creating well-connected networks with cell complex topology using only orientation and distance measurements and a set of user-defined constraints. By working directly with orientations, our method robustly resolves problems arising from data inconsistency and sensor noise. Although originally designed for reconstruction of physical shapes, the framework can be used for "sketching" new shapes directly in 3D space. We test the performance of the method using two types of acquisition devices: a standard smartphone, and a custom-made prototype called the Morphorider.

Keywords: curve networks, shape acquisition, surface reconstruction, orientation sensors

1. Introduction

Curve networks play an important role in CAD and often serve for conveying designs [Coons, 1967; Moreton & Séquin, 1991]. At the same time, well-defined curve networks are used for automatic inference of underlying surfaces by minimizing fairing energy or using methods with prescribed degree of continuity.

Recent research in sketch-based modeling and virtual reality produced tools and devices for intuitive design of 3D curves (Fig. 3). These tools and devices attempt to overcome some of the limitations of the traditional CAD approach – designers can sketch curves directly, either on a flat screen or via a 3D interface. Although intuitive, these methods are targeted towards design of new shapes and are not suitable for reconstruction of existing real-world shapes.

We present a novel framework for acquisition of 3D curves using orientations provided by inertial sensors. While the idea of sensor shape reconstruction is not new, we present the first method for creating well-connected networks of curves on surfaces using only orientation and distance measurements and a small set of user-defined

constraints. We address three main challenges which arise when working with sensor data.

- **Unknown positions**. Sensors measure local orientations of the surface – no absolute positions in the world space nor relative positions of two adjacent sensors are known.

- **Inconsistent data**. Intersecting curves often provide conflicting data, for instance two different normals for the same point in the world space.

- **Sensor noise**. Raw data from inertial sensors is noisy and needs to be pre-processed prior to reconstruction.

Our approach differs from standard curve acquisition and reconstruction methods in the fact, that we formulate all algorithms in terms of orientations. This way, we can leverage consistency of reconstructed curves with respect to user-defined topological constraints. By working directly with orientations, our framework robustly resolves all above challenges in two steps. We first filter the acquisition noise by combining pre-filtering of the raw orienta-

tions in the quaternion space with a spline-based smoothing in the group of rotations (Sec. 4). We then introduce a curve-based Poisson reconstruction method, which transforms the pre-filtered orientation samples into a smooth and consistent curve network by satisfying in particular the user annotated topological constraints (Sec. 5). To enable a thorough evaluation and comparison with respect to ground truth data (Sec. 6), we use two physical objects fabricated from digital models: the LILIUM (Fig. 1) and the CONE (Fig. 15).

To demonstrate our framework, we use a *dynamic acquisition* setup, where we suppose the data is measured by a single moving node of sensors. We test and compare two types of devices: a standard smartphone, and a custom-made prototype for measuring orientations and distances (Fig. 10). Nevertheless, the algorithms presented in this paper are not limited to these devices; they are also applicable in a *static acquisition* setup, such as a grid/mesh of sensors [Saguin-Sprynski et al., 2014; Sprynski, 2007] or instrumented materials [Hermanis et al., 2016].

The primary application of the presented framework is reconstruction of existing shapes. Our setup provides a valuable alternative to traditional shape acquisition methods (3D scanners, depth images). Unlike most 3D scanners, which often use cameras and lasers, inertial sensors are independent from light conditions, material's optical properties and scale of the scanned object. They render the acquisition possible outdoors or underground, and can also be used for large and/or moving objects; imagine acquisition of underground pipes [Saguin-Sprynski et al., 2016], or an instrumented sail moving in the wind (Fig. 20). Possible applications include smart materials and structural health monitoring.

A major advantage of our approach is its generality. No special device is required for data acquisition; in fact, any sensor-instrumented device could be used for the task, thus making 3D acquisition accessible to everyone with an ordinary smartphone. Even though originally designed for reconstruction, our framework can also be used for design by "sketching" new shapes from scratch (Fig. 2).
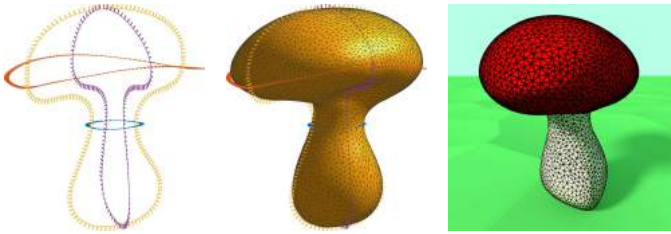

Figure 2: The MUSHROOM network created from scratch entirely with a smartphone. Curve lengths were estimated from acquisition time. The surfacing was computed using the method of Stanko et al. [2016] with soft positional constraints.

Figure 3: Recent curve acquisition devices which make use of inertial sensors. Left to right: Tilt Brush [Google], SmartPen [Milosevic et al., 2016] and 01 [InstruMMents, 2017].

2. Related work

Shape from curves. The problem of generating shapes from collections of curves has been well studied in CAGD (Computer Aided Geometric Design) and all standard techniques can be found in Farin [2002]. In the recent years, the interest in this problem was re-ignited due to its applications in sketch-based modeling [Andrews et al., 2011; Bae et al., 2008; Igarashi et al., 1999; Nealen et al., 2007; Xu et al., 2014] and in virtual reality (VR) systems [Google]. Inspired by years of research in CAGD, the *surface from curves* paradigm is invaluable for shape design in modern sketching and VR systems. Recently, Arora et al. [2017] have studied issues with accurate design of shapes in the existing VR systems. Their study suggests that users struggle even with simple tasks (drawing a closed circle) when sketching in three dimensions. A system like ours can help in solving such accuracy and consistency issues.

Although equally important, the problem of *reconstruction* of existing physical shapes from a collection of curves received less attention. 3D scanners usually provide large amounts of data and tend to ignore intrinsic structure of the scanned shape. Alternatively, objects can be defined by their characteristic curves as often done in perception and sketch-based modeling. Cao et al. [2016] detect characteristic curves in noisy point clouds, then use these curves for surface reconstruction. The use of inertial measurement units (IMUs) for shape acquisition might provide a good alternative in situations for which the optical methods do not yield proper results, positioning sensors along object's characteristic curves. Milosevic et al. [2016] introduced SmartPen, a low-cost system for capturing 3D curves which combines an IMU with a stereo camera (Fig. 3 middle). By combining a stereo camera with a sensor unit, their system is a mixture of traditional 3D scanners and our shape-from-sensors setup. However, SmartPen's sensors only serve for determining device's orientation needed for estimating relative position of the tip of the pen. Much like traditional point-cloud scanners, the system relies on visual input to get 3D position of the device in world space; this limits size of the scanned object. A recent example of curve acquisition device is the commercially available 01 [InstruMMents, 2017], a dimensioning tool with an IMU and a laser (Fig. 3 right). Allegedly, the device can be used to reconstruct 3D curves; this claim rests to be verified experimentally.
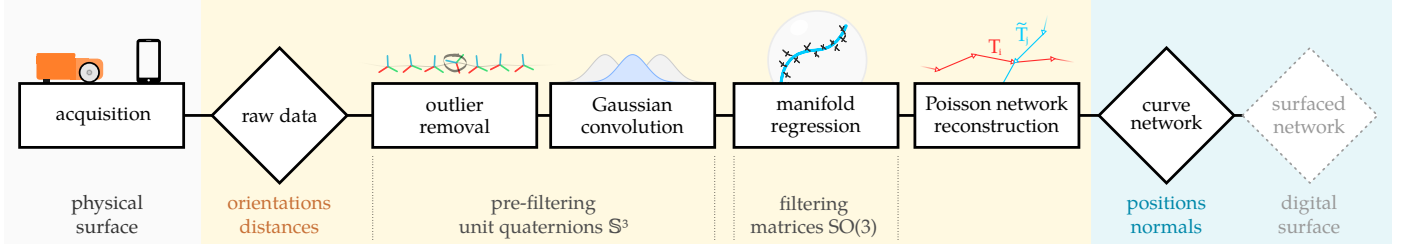
Figure 4: A schematic overview of our *shape from sensors* framework. First, the orientations and distances acquired from the physical surface are pre-filtered (Sec. 4.2) using efficient averaging schemes in the quaternion space (Sec. 4.1). Second, the pre-filtered data are smoothed using regression on the manifold of orientation matrices (Sec. 4.3) to obtain smoothly varying frames with consistent normals. Finally, the positions are reconstructed by solving a sparse linear Poisson system for the whole network (Sec. 5).

Shape from sensors. The use of sensors for shape acquisition was first explored by Sprynski [2007]. Instead of measuring absolute position of surface points of the scanned object in the world space, the reconstruction algorithms need to be formulated in terms of orientations provided by sensors, and geodesic distances between points of measurement. Curves are represented using natural parametrization and reconstructed via numerical integration. Surfaces are defined via geodesic interpolation [Sprynski et al., 2008] or using parallel ribbons of sensors [Saguin-Sprynski et al., 2014]. Huard et al. [2013] introduced method for computing smooth patches from a given piecewise geodesic boundary curve. Hoshi & Shinoda [2008] reconstruct the target surface using two families of sensors placed in orthogonal directions. Hermanis et al. [2016] construct sensor-instrumented fabric and compare the reconstruction results with Kinect data. Antonya et al. [2016] use an array of sensors for real-time tracking of human spine.

Attitude estimation and filtering. Attitude control has been studied extensively in aviation and spacecraft industry where accurate algorithms are indispensable for correct estimation of vehicle's orientation with respect to celestial objects [Markley & Mortari, 2000]. Noise in data is usually reduced using a Kalman filter – specific approaches depend on the representation used for orientations [Crassidis et al., 2007], such as unit quaternions, a representation well-known in the graphics community [Shoemake, 1985]. Markley et al. [2007] describe a classical algorithm for computing means in the group SO(3). Average rotation is defined as the minimizer of weighted penalty function, and the corresponding unit quaternion is computed efficiently via eigendecomposition of a 4-by-4 matrix; see Sec. 4.1 for more details.

Apart from statistical approaches such as Kalman filters, *geometric methods* can be used to denoise orientations. Shoemake [1985] introduced quaternion splines using spherical Bézier curves; Nielson [2004] later extended this algorithm and introduced ν–quaternion splines. Although spline methods work well for interpolation of rotations in a sequence of keyframes, the Bézier representation is not suitable for filtering noisy orientations. Energy-minimizing splines for data in Euclidean spaces have been well-suited. Reinsch [1967] introduced smoothing splines which minimize stretching and bending while approximating given data in a Euclidean space. Hofer & Pottmann [2004] describe a method for computing such splines for data on a Riemannian manifold $\mathcal{M}$. Data are first optimized in the ambient (Euclidean) space, then projected onto $\mathcal{M}$; these two steps are repeated until convergence. An equivalent result can be found by optimizing data directly on the manifold [Boumal & Absil, 2011]. In this sense, classical splines are obtained for $\mathcal{M} = \mathbb{E}^d$. We use smoothing splines on the manifold SO(3) to filter raw orientation data from sensors (Sec. 4.3).

Poisson reconstruction. Poisson's equation $\Delta \phi = \nabla \cdot \mathbf{V}$ often arises in the context of reconstruction of shapes (curves and surfaces). Kazhdan et al. [2006] resolve a Poisson problem $\Delta \chi = \nabla \cdot \mathbf{N}$ to find the indicator function ϕ, which implicitly defines the surface, from a sample of the surface's oriented normal field $\mathbf{N}$. This approach is a state-of-the-art method for surface reconstruction from unorganized points clouds. Our reconstruction algorithm is inspired by the work of Crane et al. [2013] who use the Poisson equation $\Delta f = \nabla \cdot \mathbf{T}$ to retrieve positions f of a closed planar curve from its tangent field $\mathbf{T}$ during isometric curvature flow. We extend this approach to closed curve networks on surfaces, and use it to reconstruct positions of the scanned curves (Sec. 5).
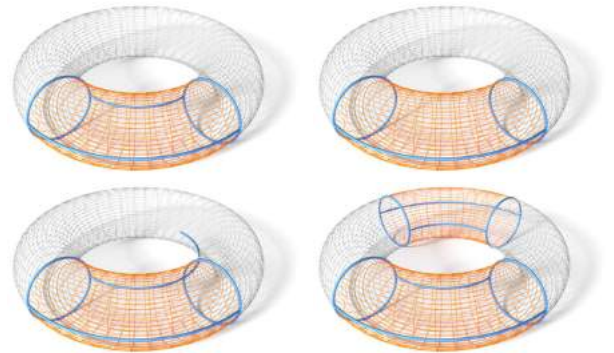

Figure 5: In this example, only the top left cell complex is a valid input to our algorithm. The other three networks have non-contractible cycles (top right), violate the definition of a cell complex (bottom left) or they are disconnected (bottom right).

3

3. Definitions and input data

Given a smooth connected 2-manifold surface $S \subset \mathbb{E}^3$, we consider a collection Γ of G^1 smooth curves embedded on S (Γ is therefore a topological subspace of S). Curves in Γ divide the surface into a finite number of cells forming a two-dimensional *cell complex* $C = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ on S, consisting of nodes $\mathcal{V}$ (0-cells), segments $\mathcal{E}$ (1-cells) and cycles $\mathcal{F}$ (2-cells) [Berberich et al., 2010; Hatcher, 2002]. Moreover we suppose the cell complex is connected and the cycles are contractible (Fig. 5). Each curve $\gamma \in \Gamma$ can be thought of as a *path* in C. We say that two curves intersect at a node $\eta \in \mathcal{V}$ if their corresponding paths contain η.

Denote by $\mathbf{x} : [0, L] \to \mathbb{E}^3$ the natural parametrization of $\gamma \in \Gamma$ where L is the length of γ. For a fixed point $\mathbf{x}$ on the curve, the orthonormal Darboux frame $\mathcal{D} = (\mathbf{T}, \mathbf{B}, \mathbf{N})$ is defined as

$$\mathbf{T} = \mathbf{x}', \quad \mathbf{N} \perp T_\mathbf{x}S, \quad \mathbf{B} = \mathbf{N} \times \mathbf{T}.$$

Here, $T_\mathbf{x}S$ denotes the two-dimensional tangent space of S at the point $\mathbf{x} \in S$ and $\mathbf{N}$ is chosen as the outward surface normal. We represent the Darboux frame $\mathcal{D}$ as an orientation matrix $\mathbf{A} = [\mathbf{T} \ \mathbf{B} \ \mathbf{N}] \in SO(3)$, which is a member of the special orthogonal group

$$SO(3) = \left\{ \mathbf{A} \in \mathbb{R}^{3 \times 3} \ : \ \mathbf{A}^\top \mathbf{A} = \mathbf{I} \text{ and } \det(\mathbf{A}) = 1 \right\}.$$

We denote by $\mathbf{A}|_\mathbf{T}$, $\mathbf{A}|_\mathbf{B}$, $\mathbf{A}|_\mathbf{N}$ the projections on the first, second and third column of the matrix $\mathbf{A}$, respectively. Throughout this paper, the terms "orientation" and "frame" refer to the same concept – the Darboux frame of a curve point with respect to the underlying surface.
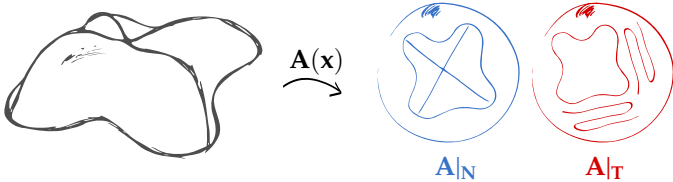

Figure 6: For a closed network, the orientation function $\mathbf{x} \mapsto \mathbf{A}(\mathbf{x})$ is *not* continuous – tangent vector is different for each curve passing through an intersection. Note that the projection on the normal component $\mathbf{A}|_\mathbf{N}$: $\Gamma \to \mathbb{S}^2$ (Gauss map of S restricted to curves) is closed as the surface normal varies smoothly.

The function $\mathbf{A} : \Gamma \to SO(3)$ which associates each curve point to its Darboux frame is not continuous. Indeed, at intersections the tangent vector $\mathbf{T}$ differs for each adjacent curve (Fig. 6). Note however that the projection on the normal component $\mathbf{A}|_\mathbf{N}$ is continuous and is identical to the Gauss map of S restricted to the curves in Γ.

Our goal is to retrieve the unknown positions $\mathbf{x}$ provided a sample of orientations $\mathbf{A}_i = \mathbf{A}(\mathbf{x}(d_i))$ for each curve γ at known distances $d_i \in [0, L]$. We suppose the topology of the underlying cell complex C is known. The main steps of our algorithm are shown in Fig. 4.

4. Filtering orientations

In this section, we describe our approach for filtering raw orientation data acquired using inertial sensors. The main problems which prevent direct use of raw orientations in the reconstruction of positions are sensor noise and limited scanning precision. By consequence, the acquired network of orientations is often *inconsistent*: at intersections, the normals are different for each passing curve. To eliminate acquisition problems, we use a two-step filtering process Fig. 7. First, raw data are pre-filtered using distance-weighted Gaussian convolution with fixed edge length. This way, we obtain uniform sampling of orientations with respect to curve lengths. Second, the uniform samples are smoothed and made consistent using regression on the manifold SO(3).
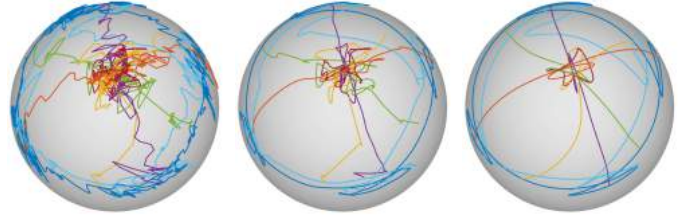

Figure 7: Two-step filtering of acquired orientations from Fig. 1, here visualized using the Gauss map $\mathbf{A}|_\mathbf{N}$ (surface normals extracted from orientations mapped to the unit sphere). Left to right: raw data, pre-filtered data (Sec. 4.2), filtered data (Sec. 4.3).

4.1. Averaging orientations

The compute (weighted) means in the group of rotations, it is convenient to use the quaternion representation [Markley & Mortari, 2000]. The rotation around the (unit) Euler axis $\mathbf{e}$ by the angle 2θ is given by the unit quaternion

$$\mathbf{q} = [\cos \theta; \mathbf{e} \sin \theta] \in \mathbb{S}^3 \subset \mathbb{H},$$

where $\mathbb{S}^3$ denotes the unit 3-sphere in the quaternion space $\mathbb{H} \approx \mathbb{R}^4$. We will denote by $\mathbf{A}(\mathbf{q})$ the conversion between rotation matrix $\mathbf{A}$ and the unit quaternion $\mathbf{q}$ (see Shoemake [1985] for details).

Given a set of unit quaternions $\mathbf{q}_i \in \mathbb{S}^3$ with weights w_i for $i = 1, \dots, N$, the average quaternion $\bar{\mathbf{q}}$ is defined via the following minimization problem [Markley et al., 2007]:

$$\bar{\mathbf{q}} = \arg \min_{\mathbf{q} \in \mathbb{S}^3} \sum_{i=1}^{N} w_i \left\| \mathbf{A}(\mathbf{q}) - \mathbf{A}(\mathbf{q}_i) \right\|_\mathrm{F}^2,$$

where $\| \cdot \|_\mathrm{F}$ is the Frobenius norm. This is equivalent to solving

$$\bar{\mathbf{q}} = \arg \max_{\mathbf{q} \in \mathbb{S}^3} \mathbf{q}^\top \mathbf{M} \mathbf{q},$$

where the 4×4 matrix $\mathbf{M}$ is the *structure tensor*

$$\mathbf{M} = \sum_{i=1}^{N} w_i \, \mathbf{q}_i \, \mathbf{q}_i^\top.$$

The solution to this optimization problem is known to be the (unit) eigenvector $\bar{\mathbf{q}}$ of $\mathbf{M}$ corresponding to the maximum eigenvalue. This procedure is known as the q method. The q method is computationally efficient and naturally overcomes the fact that unit quaternions $\mathbb{S}^3$ are a 2-to-1 covering of the rotation group: antipodal quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. This is why spherical linear interpolation [SLERP, Shoemake, 1985] is not suitable for averaging quaternions – changing the sign of a quaternion should not change the average, which is not the case for SLERP.

4.2. Pre-filtering via Gaussian convolution

Working with orientations which are sampled uniformly with respect to geodesic distance facilitates both the filtering (Sec. 4.3) and the reconstruction of positions (Sec. 5) – the uniform sampling is convenient for discretization of differential operators and intergrals. In this section we describe the computation of such uniform sampling from raw orientation measures in the quaternion space.

Before computing the uniform sampling, we first remove outliers and process duplicate measures. Outliers are removed by looking at the dot products of quaternion $\mathbf{q}_i$ with its immediate neighbors $\mathbf{q}_{i-1}$ and $\mathbf{q}_{i+1}$. If both dot products are smaller than the specified threshold α, the angles between $\mathbf{q}_i$ and its neighbors are too big and $\mathbf{q}_i$ is removed. For the examples in this paper, we use the threshold $\alpha = 0.9$; this is consistent with the assumption that the acquisition frequency is high enough and each two adjacent data points are close in the space of orientations. For sparsely sampled networks, a smaller threshold or an adaptive scheme might be more suitable. Duplicate measures, i.e. different orientations which correspond to the same curve point are averaged using the q method described in the previous section (with $w \equiv 1$).

To compute uniform sampling from unique sampling (no outliers or duplicates), we fix the edge length h_{fixed} and compute uniform distance parameters t_k for each curve γ:

$$t_k = \frac{kL}{N}, \quad k = 0, \ldots, N, \qquad t_k - t_{k-1} = h \qquad (1)$$

where L is the length of γ and $N = \text{round}(L/h_{\text{fixed}})$. For each t_k, we compute the corresponding orientation $\bar{\mathbf{q}}_k$ using convolution with distance weighting (Fig. 8):

$$\bar{\mathbf{q}}_k = \underset{\mathbf{q}_i \in \gamma}{\text{mean}} \left(w_{k,i} \, \mathbf{q}_i \right).$$

Here, mean() refers to the q method described in Sec. 4.1. The Gaussian weights are computed as

$$w_{k,i} = \exp\left[-\frac{1}{2} \left(\frac{t_k - d_i}{\sigma h_{\text{fixed}}} \right)^2 \right].$$

The parameter σ controls radius of convolution; in general we use values between 0.2 and 0.5.
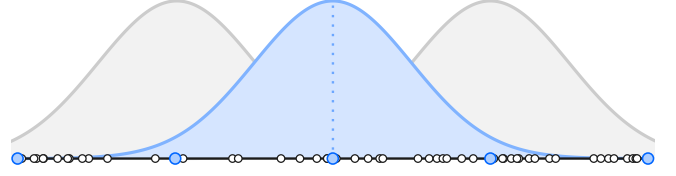

Figure 8: Schema of pre-filtering with $\sigma = 0.2$. Raw orientations (white) are convoluted with Gaussian kernel to obtain uniform sampling (blue).

4.3. Regression on SO(3) with normal constraints

To some extent, the pre-filtering smooths out acquisition artifacts in the scanned data. By varying the parameter σ, it might be possible to sufficiently filter the data while preserving the acquired geometry. However, this approach has a major drawback: since the Gaussian convolution is applied to each curve individually, there is no guarantee that the filtered orientations will be consistent at intersections (see Sec. 3). This type of coherence is however essential, not only for the final reconstruction step of our method (orientations transformed into positional surface data) which assumes consistent input, but also to ensure a correct subsequent processing of the curve network, for instance for surface fitting.

Our goal is therefore to compute a function which is as smooth as possible while fitting the set of frames and satisfying a set of consistency constraints. To have a filtering framework with unified representation, our initial idea was to use quaternion splines [Nielson, 2004; Park & Ravani, 1997; Shoemake, 1985]. However, expressing the consistency constraints directly in the quaternion space gives nonlinear equations; on the other hand, these constraints are linear in SO(3). Inspired by spline functions, which provide smooth curves that reasonably fit the input data by minimizing some energy functional, we propose in the following to formulate filtering of frames (orientations) as a customized energy minimization problem on SO(3).

Recall that a spline in tension $\mathbf{x}(t)$ in the d-dimensional Euclidean space $\mathbb{E}^d$ is the minimizer of the energy

$$\lambda \int_{t_0}^{t_N} \|\dot{\mathbf{x}}\|^2 \, dt + \mu \int_{t_0}^{t_N} \|\ddot{\mathbf{x}}\|^2 \, dt$$

under the interpolation constraints $\mathbf{x}(t_i) = \mathbf{p}_i \in \mathbb{E}^d$. The weights λ and μ control stretching and bending of the spline, respectively. Incorporating positional constraints directly in the energy

$$\sum_{i=0}^{N} \|\mathbf{x}(t_i) - \mathbf{p}_i\|^2 + \lambda \int_{t_0}^{t_N} \|\dot{\mathbf{x}}\|^2 \, dt + \mu \int_{t_0}^{t_N} \|\ddot{\mathbf{x}}\|^2 \, dt \qquad (2)$$

yields a smoothing spline, useful for non-parametric regression and data filtering in Euclidean spaces.

Smoothing splines have been generalized for data on Riemannian manifolds by many authors [Boumal & Absil, 2011; Brunnett & Crouch, 1994; Hofer & Pottmann, 2004].

We will now use this manifold formulation to smooth data in the three-dimensional group of rotations which is a Lie group with structure of a differentiable manifold.

Analogically to Euclidean spaces, a smoothing spline $\mathbf{X}(t)$ for orientation data $\mathbf{A}_i \in SO(3)$ is defined by minimizing the cost function

$$\sum_{i=0}^{N} \text{dist}^2\left(\mathbf{X}(t_i), \mathbf{A}_i\right) + \lambda \int_{t_0}^{t_N} \left\langle \dot{\mathbf{X}}, \dot{\mathbf{X}} \right\rangle dt + \mu \int_{t_0}^{t_N} \left\langle \frac{D^2\mathbf{X}}{dt^2}, \frac{D^2\mathbf{X}}{dt^2} \right\rangle dt,$$

where $\text{dist}(\cdot, \cdot)$ is the Riemannian (geodesic) distance on $SO(3)$, $\langle \cdot, \cdot \rangle$ is the Riemannian metric, $\dot{\mathbf{X}}$ is the first derivative and $D^2\mathbf{X}/dt^2$ is the second covariant derivative of $\mathbf{X}$. We will write the above cost function as

$$E(\mathbf{X}) = E_0(\mathbf{X}) + \lambda E_1(\mathbf{X}) + \mu E_2(\mathbf{X}).$$

To discretize this problem, the continuous curve $\mathbf{X}(t) : [0, L] \to SO(3)$ is replaced by the vector

$$\gamma = (\mathbf{X}(t_0), \ldots, \mathbf{X}(t_N)) = (\mathbf{X}_0, \ldots, \mathbf{X}_N) \in SO(3)^{N+1}$$

where t_k is the uniform parametrization defined in (1).

We use the method for spline fitting on manifold of Boumal & Absil [2011]. For the sake of completeness, we briefly review the main ideas here. The differential and integral operators in (2) are approximated via *geometric* finite differences. Geometrically, the difference $\mathbf{B} - \mathbf{A}$ of two points $\mathbf{A}, \mathbf{B} \in \mathbb{R}^n$ in the Euclidean space is the vector starting at $\mathbf{A}$ and pointing towards $\mathbf{B}$ whose length is the distance between $\mathbf{A}$ and $\mathbf{B}$. On manifolds, the same notion is expressed using the Riemannian logarithmic map as $\log(\mathbf{A}^\top\mathbf{B})$. Note that the logarithmic map is the inverse of the exponential map which generates geodesics. For orientation data $\mathbf{A}_i \in SO(3)$, the cost function (2) has the following form:

$$\sum_{i=0}^{N} \left\| \log\left(\mathbf{X}_i^\top \mathbf{A}_i\right) \right\|_F^2 \quad + \quad \frac{1}{h} \sum_{i=0}^{N-1} \left\| \log\left(\mathbf{X}_i^\top \mathbf{X}_{i+1}\right) \right\|_F^2$$
$$+ \quad \frac{1}{h^3} \sum_{i=1}^{N-1} \left\| \log\left(\mathbf{X}_i^\top \mathbf{X}_{i+1}\right) + \log\left(\mathbf{X}_i^\top \mathbf{X}_{i-1}\right) \right\|_F^2.$$

We use the following simplified costs [Boumal, 2013]:

$$E_0(\gamma) = \sum_{i=0}^{N} \|\mathbf{X}_i - \mathbf{A}_i\|_F^2, \quad E_1(\gamma) = \frac{1}{h} \sum_{i=0}^{N-1} \|\mathbf{X}_i - \mathbf{X}_{i+1}\|_F^2,$$

$$E_2(\gamma) = \frac{1}{h^3} \sum_{i=1}^{N-1} \left\| \text{skew}\left(\mathbf{X}_i^\top\left(\mathbf{X}_{i+1} + \mathbf{X}_{i-1}\right)\right) \right\|_F^2, \quad (3)$$

where γ is the vector of unknown orientations (discrete curve), $h = t_i - t_{i-1}$ is the uniform discretization step, and $\text{skew}(\mathbf{M}) = (\mathbf{M} - \mathbf{M}^\top)/2$. For closed curves, all sums run from 0 to N due to periodicity.

Let us recall how the simplified energy is obtained. The terms E_0, E_1 result from the approximation of the

geodesic distance on $SO(3)$ by the chordal distance in the ambient space $\mathbb{R}^{3\times3}$:

$$\text{dist}^2(\mathbf{A}, \mathbf{B}) = \left\| \log\left(\mathbf{A}^\top\mathbf{B}\right) \right\|_F^2 \approx \|\mathbf{A} - \mathbf{B}\|_F^2.$$

To obtain the simplified term E_2, consider the map $\mathbf{B} \mapsto \mathbf{A} \log(\mathbf{A}^\top\mathbf{B})$ which associates to $\mathbf{B} \in SO(3)$ a tangent vector at $\mathbf{A} \in SO(3)$. A similar vector is obtained by projecting the vector $\mathbf{B} - \mathbf{A}$ from the ambient space of $\mathbb{R}^{3\times3}$ to the tangent space $T_\mathbf{A}SO(3)$. This projection is characterized by the operator $\mathbf{A} \, \text{skew}(\mathbf{A}^\top\mathbf{B})$. It is therefore convenient to approximate the log operator by the skew operator:

$$\log(\mathbf{A}) \approx \text{skew}(\mathbf{A}).$$

See the paper of Boumal [2013] for an analysis and the error bounds of the simplified cost.

In our setup, the raw data $\mathbf{A}_i$ is associated to a network Γ and the above energy terms are summed over all curves $\gamma \in \Gamma$. Simultaneously with regression on $SO(3)$, we solve the consistency constraints as follows. Let $\mathbf{X}_i, \tilde{\mathbf{X}}_j$ be the local frames of two intersecting curves at their common node $\eta \in \mathcal{V}$. Then the two frames are called *consistent*, if the frame $\mathbf{X}_i = [\mathbf{T}_i \, \mathbf{B}_i \, \mathbf{N}]$ is obtained by rotating the frame $\tilde{\mathbf{X}}_j = \left[\tilde{\mathbf{T}}_j \, \tilde{\mathbf{B}}_j \, \mathbf{N}\right]$ around the surface normal $\mathbf{N}$ at the node η. Denote by $\mathcal{N}$ the set of all such pairs $\left(\mathbf{X}_i, \tilde{\mathbf{X}}_j\right)$. To enforce consistency, we add a term penalizing the difference in projection on the normal component

$$E_N(\Gamma) = \sum_{\left(\mathbf{X}_i, \tilde{\mathbf{X}}_j\right) \in \mathcal{N}} \left\| \mathbf{X}_i|_\mathbf{N} - \tilde{\mathbf{X}}_j|_\mathbf{N} \right\|^2. \quad (4)$$

The final energy has the form

$$\boxed{E(\Gamma) = \xi E_N(\Gamma) + \sum_{\gamma \in \Gamma} E_0(\gamma) + \lambda E_1(\gamma) + \mu E_2(\gamma)} \quad (5)$$

where $\xi, \lambda, \mu \geq 0$ are the weights which control consistency of surface normals, stretching, and bending, respectively (Fig. 16).

The energy E is minimized using the Riemannian trust-region (RTR), a generalization of the classical trust-region in $\mathbb{R}^n$ to differentiable manifolds; see Absil et al. [2007] for the description of the algorithm and a detailed convergence analysis. We use the RTR implemented in the ROPTLIB library [Huang et al., 2016].

5. Poisson network reconstruction

In the previous section, we described a filtering method of raw orientation data measured by sensors, which yields uniform, smooth and consistent sampling. We now address the problem of curve network reconstruction under topological constraints: these include intersections of two or more curves, and closure of a single curve.
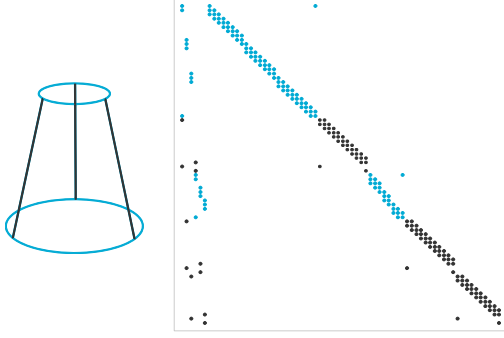
Figure 9: Matrix of the discrete Poisson system (6) for a cone network with two closed curves (blue) and three open curves (black). Dots indicate non-zero coefficients. First six columns correspond to network nodes.

Previous approaches treat each curve individually and employ various heuristics to glue the curves together to obtain the correct topology [Hermanis et al., 2016; Sprynski, 2007]. This process is limited to specific (grid) topologies and often requires manual corrections to obtain proper intersections. Our approach, on the other hand, naturally satisfies topological constraints by construction. In this section, we extend the discretization of the Poisson equation for a single closed curve described by Crane et al. [2013] to curve networks with the topology of a cell complex as defined in Sec. 3.

If $\mathbf{x}$ is the natural parametrization of the curve γ (one-dimensional Riemannian manifold), the unit tangent field $\mathbf{T}$ along γ is equal to the gradient of positions

$$\nabla \mathbf{x} = \mathbf{T}$$

The vector field $\mathbf{T}$ is generally not integrable [Crane et al., 2013; Kazhdan et al., 2006] and the solution needs to be found in the least-squares sense. To that end, the divergence operator is applied to both sides and $\mathbf{x}$ is found by solving the Poisson equation

$$\Delta \mathbf{x} = \nabla \cdot \mathbf{T} \tag{6}$$

Even though the curve network Γ is not a Riemannian manifold, it can be viewed as a collection of one-dimensional Riemannian manifolds constrained at intersections. This allows us to apply the above Poisson equation to retrieve the network from local orientations. With that in mind, we discretize the gradient ∇ and the Laplacian Δ individually for each curve and retrieve the positions by solving a single global linear system. The topological constraints are enforced by representing each node as a unique point in the system (Fig. 9).

After the filtering, each curve is represented as a discrete collection of unknown points $\mathbf{x}_i = \mathbf{x}(t_i) \in \mathbb{E}^3, i = 0, \ldots, N$, with known Darboux frames $\mathbf{X}_i$. We will denote by $\mathbf{T}_i = \mathbf{X}_i|_\mathbf{T}$ the (unit) tangent extracted from $\mathbf{X}_i$. With uniform parametrization $h = t_i - t_{i-1}$, the differential operators are discretized via finite differences by

$$\Delta \mathbf{x}_i = 1/h^2 \left(\mathbf{x}_{i-1} - 2\mathbf{x}_i + \mathbf{x}_{i+1} \right), \quad \nabla \mathbf{T}_i = 1/h \left(\mathbf{T}_i - \mathbf{T}_{i-1} \right).$$

The above discretization is valid for all interior vertices. At endpoints of open curves, we directly impose the following boundary conditions

$$1/h \left(\mathbf{x}_1 - \mathbf{x}_0 \right) = \mathbf{T}_0,$$
$$1/h \left(\mathbf{x}_N - \mathbf{x}_{N-1} \right) = \mathbf{T}_N.$$

The resulting linear system is sparse with at most three non-zero coefficients per row (Fig. 9). Thanks to the regression step described in Sec. 4.3, the tangent vectors of all curves meeting at network node are coplanar at that node. Therefore the above boundary conditions ensure the reconstruction of a G^1 curve network. Note that the discretization does not take into account the fact that $h = t_i - t_{i-1}$ is the *geodesic* distance along the reconstructed curve, which differs from the (Euclidean) norm $\tau_i = \|\mathbf{x}_{i-1} - \mathbf{x}_i\|$. In general, this is not an issue, as the least-squares minimization of the Poisson energy distributes the error evenly.

The euclidean distance τ_i could be estimated from the geodesic distance h, for instance by approximating τ_i by the chordal length of the circular arc defined by the length h and angle between $\mathbf{T}_{i-1}$ and $\mathbf{T}_i$ (see inset). This estimation comes at a price – the modified parametrization is no longer uniform. Our experiments have shown that the error $|\tau_i - h|$ is generally too small to influence the reconstruction.
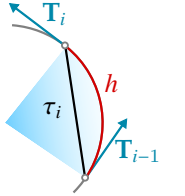
6. Evaluation

6.1. Data acquisition

To acquire data on physical surfaces, we use dynamic acquisition setup – the orientations are measured using a single IMU which is moved around on the surface by the user. In our experiments we used two types of devices: a standard smartphone, and the Morphorider. The *Morphorider* (Fig. 10) is a wireless mouse-like device for measuring local orientation, containing a single 3A3M sensor (a tri-axial accelerometer and a tri-axial magnetometer) with an incremental wheel for tracking distance. It is a custom-made prototype specifically designed for acquisition of curves on surfaces.
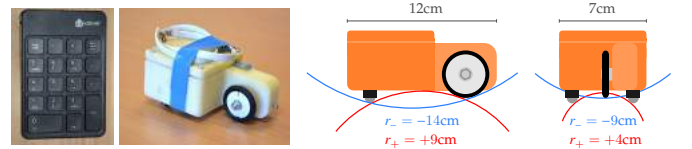

Figure 10: Left: the Morphorider and the wireless keypad that we use for marking nodes during acquisition. Right: the schema shows Morphorider's limitations with respect to radius of curvature at locally convex (red) and concave (blue) surface points.

We use a MATLAB interface to acquire data with the Morphorider (Fig. 11 left). Prior to acquisition, we assign

a unique index between 0 and N-1 to each of the N network nodes. The acquisition is controlled remotely using a wireless numpad (Fig. 10 left). When starting a new curve, we first indicate if the curve is at the boundary or in the interior. Then, every time the device passes through a node, we record its index. See the accompanying video for the acquisition setup.

Figure 11: The devices used for data acquisition. Left: the Morphorider and a screenshot of the MATLAB acquisition interface. Right: a standard smartphone and the application with a simple acquisition interface.

Similarly, orientation data can be measured with a standard *smartphone*. In addition to accelerometers and magnetometers, smartphones also contain gyroscopes. This enables orientation acquisition which is more robust to magnetic perturbations. We use the state-of-the-art sensor fusion algorithms to robustly estimate orientation from sensor measurements [Madgwick et al., 2011; Pacha, 2013]. However, measuring distances with a smartphone is difficult. The GPS is not an option for mid-sized objects such as the LILIUM (Fig. 11) whose bounding box has dimensions $1.00\text{m} \times 1.00\text{m} \times 0.25\text{m}$. To overcome this limitation, we assume the speed of acquisition is constant, then we parametrize each curve using acquisition timestamps. Optionally, manual distance measurements can be used to increase the precision of reconstruction. We measure the length L_s for each segment s which is then used to transform the time parameter $t \in [t_{\text{start}}, t_{\text{end}}]$ into the natural parameter $d \in [0, L_s]$. In this case, we suppose the speed of acquisition is constant for each segment, but it does not need to be constant through the whole acquisition process. In practice, the constant-speed-per-segment constraint is much easier to handle and the manual distance measurement improves the reconstruction (Fig. 12). To acquire data with a smartphone, we use a lightweight Java application (Fig. 11 right).

6.2. Convergence and error of reconstruction

We demonstrate the performance of our reconstruction framework on synthetic and acquired data.

- We use synthetic data extracted from known ground truth networks for convergence analysis of the Poisson network reconstruction (Fig. 13). The synthetic data are not corrupted by noise and therefore do not need to be filtered.

- Data acquired from known surfaces serve for evaluating the overall performance of the framework on real-world examples (Fig. 14 and 15). Moreover,
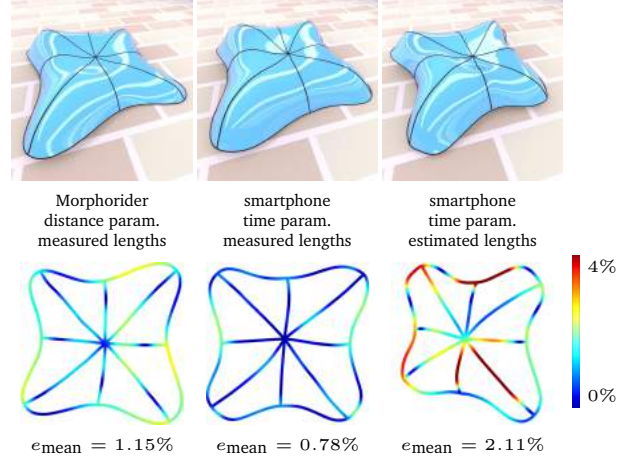

Figure 12: Comparison of three acquisition setups for the LILIUM. Top pictures show surfaced networks rendered with reflection lines; bottom pictures show the error of Poisson reconstruction for each network. In the left network, the orientations were acquired using the Morphorider and parametrized by distances. In middle and right networks, the orientations were acquired with the smartphone and parametrized by acquisition time – middle network uses measured segment lengths, right network uses time-estimated segment lengths. All lengths are relative to the diameter (length of AABB diagonal) of the ground truth surface.
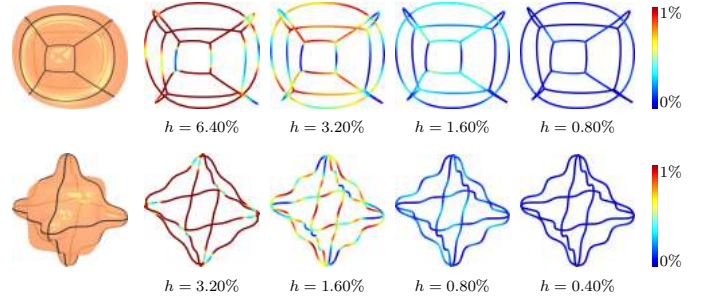

Figure 13: As $h \rightarrow 0$, synthetic networks reconstructed by solving the Poisson equation (6) converge to the ground truth network (black). All lengths are relative to the diameter (length of AABB diagonal) of the ground truth surface (orange).

they enable quantitative comparison of the two acquisition methods.

The error metric associated with each point **x** in the reconstructed network is defined as the distance between **x** and its closest point on the ground truth network or surface. For synthetic data, the closest point is found on the ground truth network. For data acquired from a physical surface for which we have a digital model (LILIUM, CONE), closest points are instead found on the underlying surface as the ground truth network is not available. Note that the dimensions of the fabricated surfaces are $1.00\text{m} \times 1.00\text{m} \times 0.25\text{m}$ (LILIUM) and $1.00\text{m} \times 1.00\text{m} \times 2.00\text{m}$ (CONE). Prior to error computation we need to register the acquired network to the surface, since the two objects do not live in the same coordinate systems. We use the ICP algorithm [Besl & McKay, 1992] for the registration.

The convergence plot for decreasing h is shown in Fig. 17 and the corresponding numerical data is given
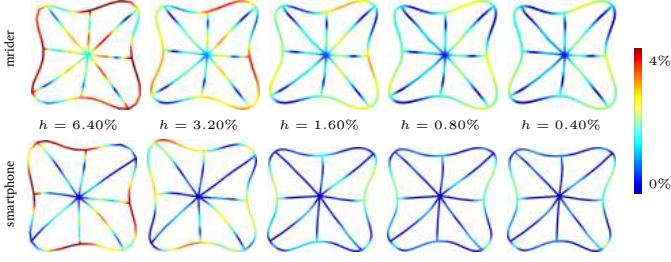
8

Figure 14: Reconstruction error for the LILIUM scanned with the Morphorider (top) and a smartphone (bottom) for decreasing h. For filtering, we used the weights $\lambda = \mu = 1$. All lengths are relative to the diameter (length of AABB diagonal) of the ground truth surface.

Figure 15: Reconstruction error for the CONE scanned with the Morphorider for decreasing h. For filtering, we used the weights $\lambda = \mu = 1$. All lengths are relative to the diameter (length of AABB diagonal) of the ground truth surface.
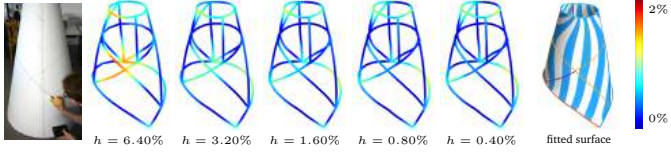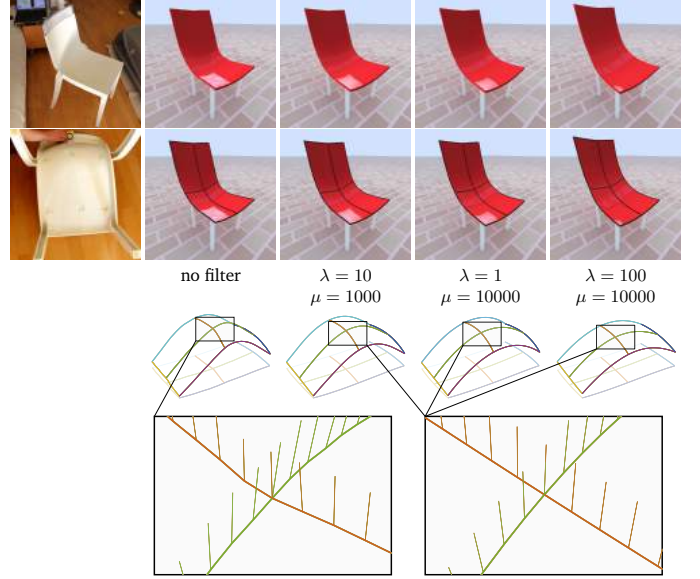

Figure 16: A CHAIR reconstructed with different sets of weights. After the pre-filtering, normals at nodes are not compatible (bottom left). The normal penalty term in the filtering energy ensures the consistency of the reconstructed network (bottom right).

in Table 1. All errors are relative to the diameter (the length of AABB diagonal) of the corresponding ground truth surface. For synthetic data (BOWL, BUMPYCUBE), the mean error approaches zero for decreasing h (Fig. 13). For acquired networks on the LILIUM, the mean error is about 1% using measured distances (both devices) and around 2% using time-estimated distances (smartphone). It is not surprising that when using the measured distances, the smartphone error decreases faster than the Morphorider error. One can further observe that even though the smartphone error (both mean and max) is bigger than the Morphorider error at coarser resolutions, it becomes smaller at finer scales. We explain this behavior by the fact that the distance data are not the same for the two devices. For the Morphorider data, we have exact correspondences between distance and orientation measurements; for the smartphone data, only the total length of each segment is known and the correspondences are estimated from acquisition time. It seems like at finer scales, the error of this estimation is better-distributed which results in a reconstruction with higher precision.

6.3. Filtering

The main goal of the filtering algorithm described in Sec. 4 is to obtain clean and consistent orientations from raw sensor measures. The weights λ, μ in the filtering energy (5) influence the shape of the reconstructed indirectly by controlling stretching and bending of splines on the manifold SO(3) used for smoothing the orientations. In all examples, we use the weight $\xi = $ 1e4 to enforce the consistency of normals.

The influence of λ and μ on the Poisson-reconstructed positions is compared in Fig. 18. Interestingly, even though λ and μ control stretching and bending of *orientations* $\mathbf{X}_i$, they also seem to control stretching and bending of reconstructed *positions* $\mathbf{x}_i$. The weights need to be chosen carefully, otherwise the network might end up over-filtered; see Fig. 18 where $\lambda = 100$.

6.4. Additional results

To further illustrate the benefits of our method, we compute smooth surfaces which fit not only the resulting network of curves but also the corresponding filtered surface normals using the algorithm of Stanko et al. [2016]. All other fitting techniques that take into account the normal vectors available along the curve network [Botsch & Kobbelt, 2004; Jacobson et al., 2010] could also be used. Fig. 20 shows some of the reconstructed surfaces.

Fig. 16 shows surfaces interpolating the CHAIR network filtered with various sets of weights. It also illustrates the phenomenon of inconsistent orientations: pre-filtered normals do not agree at intersections, and naive averaging of each pair of conflicting normals results in a shape with poor fidelity to the scanned object (Fig. 16 bottom left). Our regression filtering technique resolves this problem while preserving the underlying geometry (Fig. 16 bottom right). Note however that surface reconstruction is not in the scope of this paper.

During acquisition, we often faced the question of which curves to acquire in order to minimize the error of reconstruction – different curve networks on the same surface often produce different results. Fig. 19 shows the ROOF BOX reconstructed with seven different sets of curves. The optimal choice of curves to be scanned merits to be investigated futher, but goes beyond the scope of the paper.
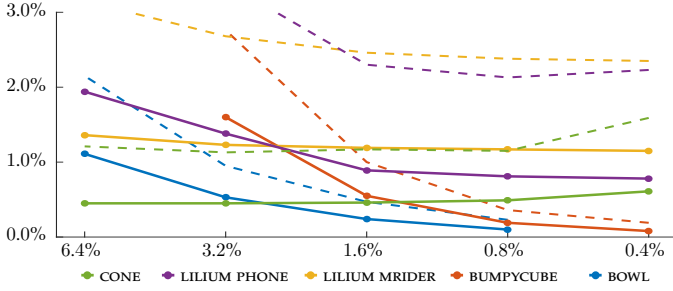
Figure 17: Reconstruction error with respect to ground truth. Abscissae is step size relative to object size. Solid lines represent mean error (Table 1), while dashed lines represent maximal error. All lengths are relative to the diameter (length of AABB diagonal) of the ground truth surface.

data	source	lengths	mean error (%)				
		h (%)	6.4	3.2	1.6	0.8	0.4
BOWL	synth.	exact	1.11	0.53	0.24	0.10	-
BUMPY	synth.	exact	-	1.60	0.55	0.19	0.08
CONE	mrider	measured	0.61	0.49	0.46	0.45	0.45
LILIUM	mrider	measured	1.36	1.23	1.19	1.17	1.15
LILIUM	phone	measured	1.94	1.38	0.89	0.81	0.78
LILIUM	phone	time-est.	-	1.83	2.01	2.03	2.11

Table 1: Mean distance error with respect to ground truth. All lengths are in % and relative to the diameter (length of AABB diagonal) of the corresponding ground truth surface.

7. Limitations and future work

Most limitations of our method are device-related. The current Morphorider is a proof of concept and suffers from construction drawbacks. Its relatively big size inhibits the acquisition; we often found it hard to manipulate, and some regions with high curvature (in absolute value) could not have been scanned (see Fig. 10). Another problem is that the distance-measuring wheel is not aligned with the sensor unit – in practice this means that there is an offset in the orientation measurement. Moreover, the device relies on magnetometers and cannot be used around ferromagnetic objects. A next generation prototype currently in development might help in resolving some of these problems. The biggest limitation of the smartphone setup is the inability to measure distances – these need to be estimated or obtained manually.

The current approach requires the user to have a specific network topology in mind prior to the acquisition. Nodes cannot be added during acquisition – adding a new node means that all previously scanned curves passing through this node need to be re-scanned, which is impractical and slows down the scanning process. Moreover, the system needs user feedback to know when the acquisition device passes through a node. In future, we plan to implement a more intuitive interface by using a heuristic for snapping curves to existing nodes to ease the acquisition.

Curvature plots in Fig. 18 imply that the reconstructed curves are G^2, but there are issues with overall G^2-continuity at nodes. We believe these issues might be resolved by a slight modification of the normal penalty term in the filtering energy, for instance by taking into account normals of node's neighboring vertices. We also plan to extend the method to surfaces with sharp features, i.e. curves with two sets of frames which agree up to rotation around curve's tangent vector.

References

Absil, P.-A., Baker, C., & Gallivan, K. (2007). Trust-region methods on riemannian manifolds. *Foundations of Computational Mathematics*, 7(3), 303–330.

Andrews, J., Joshi, P., & Carr, N. (2011). A linear variational system for modelling from curves. *Computer Graphics Forum*, 30(6), 1850–1861.

Antonya, C., Butnariu, S., & Pozna, C. (2016). Real-time representation of the human spine with absolute orientation sensors. In *14th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1–6).

Arora, R., Kazi, R. H., Anderson, F., Grossman, T., Singh, K., & Fitzmaurice, G. (2017). Experimental evaluation of sketching on surfaces in vr. In *Proc. Conf. Human Factors in Computing Systems (CHI)*. ACM.

Bae, S.-H., Balakrishnan, R., & Singh, K. (2008). ILoveSketch: As-Natural-As-Possible Sketching System for Creating 3D Curve Models. In *UIST '08* (pp. 151–160). ACM.

Berberich, E., Fogel, E., Halperin, D., Mehlhorn, K., & Wein, R. (2010). Arrangements on Parametric Surfaces I: General Framework and Infrastructure. *Mathematics in Computer Science*, 4, 45–66.

Besl, P., & McKay, N. (1992). A method for registration of 3-D shapes. *Trans. Pattern Analysis and Machine Intelligence*, 14, 239–256.

Botsch, M., & Kobbelt, L. (2004). An intuitive framework for real-time freeform modeling. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3), 630–634.

Boumal, N. (2013). Interpolation and regression of rotation matrices. In F. Nielsen, & F. Barbaresco (Eds.), *Geometric Science of Information* (pp. 345–352). Springer Berlin Heidelberg volume 8085 of *Lecture Notes in Computer Science*.

Boumal, N., & Absil, P.-A. (2011). A discrete regression method on manifolds and its application to data on SO(n). In *Proceedings of the 18th IFAC World Congress (Milan)* (pp. 2284–2289). volume 18.

Brunnett, G., & Crouch, P. E. (1994). Elastic curves on the sphere. *Advances in Computational Mathematics*, 2(1), 23–40.

Cao, Y., Nan, L., & Wonka, P. (2016). Curve networks for surface reconstruction. *preprint*, . http://arxiv.org/abs/1603.08753.

Coons, S. A. (1967). *Surfaces for computer-aided design of space forms*. Technical Report DTIC Document.

Crane, K., Pinkall, U., & Schröder, P. (2013). Robust fairing via conformal curvature flow. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32.

Crassidis, J. L., Markley, F. L., & Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of guidance, control, and dynamics*, 30(1), 12–28.

Farin, G. E. (2002). *Curves and surfaces for CAGD: a practical guide* . (5th ed.). Morgan Kaufmann.

Google (2017). Tilt Brush. https://www.tiltbrush.com/.

Hatcher, A. (2002). *Algebraic Topology*. Cambridge University Press.

Hermanis, A., Cacurs, R., & Greitans, M. (2016). Acceleration and magnetic sensor network for shape sensing. *IEEE Sensors Journal*, 16(5), 1271–1280.

Hofer, M., & Pottmann, H. (2004). Energy-minimizing splines in manifolds. *ACM Transactions on Graphics (TOG)*, 23(3), 284–293.

Hoshi, T., & Shinoda, H. (2008). 3D Shape Measuring Sheet Utilizing Gravitational and Geomagnetic Fields. In *SICE Annual Conference* (pp. 915–920).

Huang, W., Absil, P.-A., Gallivan, K. A., & Hand, P. (2016). *ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds*. Technical Report FSU16-14.v2 Florida State University. http://www.
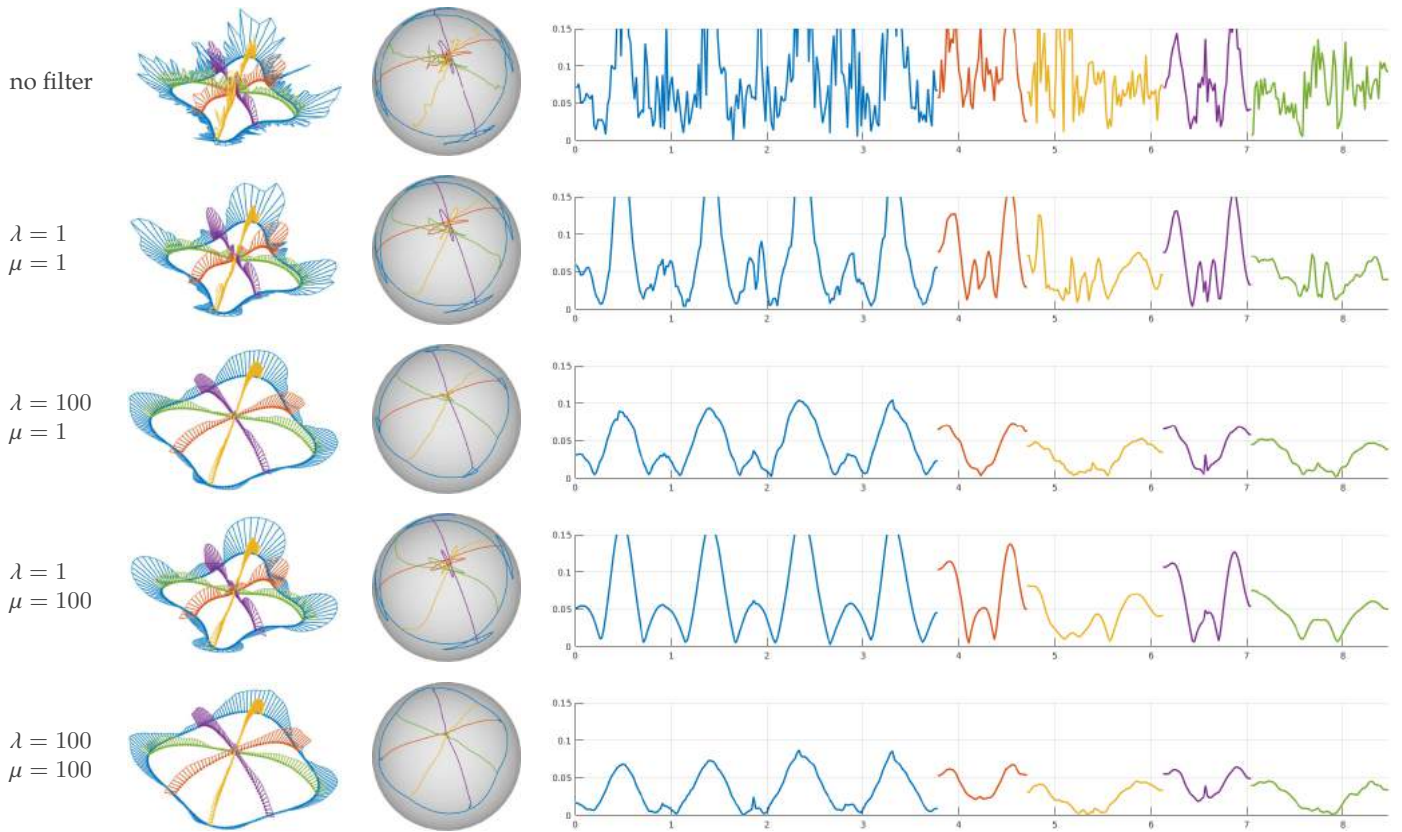
Figure 18: Reconstruction of LILIUM with various filtering weights. Left to right in each row: reconstructed network with porcupine plot (surface normal scaled by curvature); Gauss map; curvature plot.
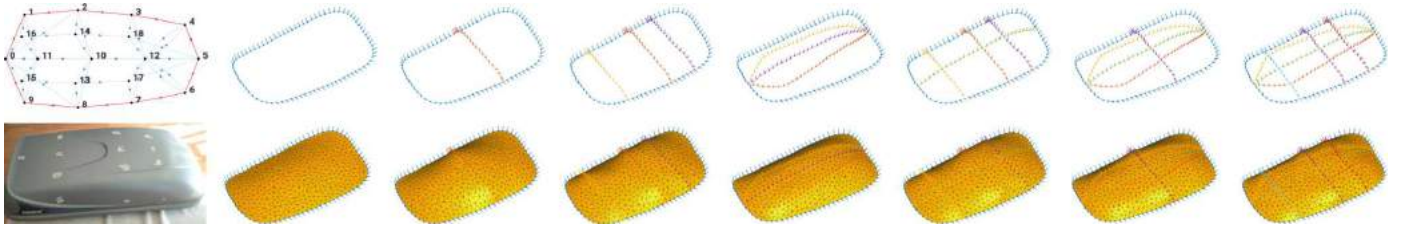

Figure 19: Surface of a ROOF BOX reconstructed with different sets of curves.

math.fsu.edu/~whuang2/papers/ROPTLIB.htm.

Huard, M., Sprynski, N., Szafran, N., & Biard, L. (2013). Reconstruction of quasi developable surfaces from ribbon curves. *Numerical Algorithms*, *63*(3), 483–506.

Igarashi, T., Matsuoka, S., & Tanaka, H. (1999). Teddy: A Sketching Interface for 3D Freeform Design. In *Proc. SIGGRAPH 99. Annual Conference Series* (pp. 409–416).

InstruMMents (2017). 01 dimensioning instrument. http://instrumments.com/.

Jacobson, A., Tosun, E., Sorkine, O., & Zorin, D. (2010). Mixed Finite Elements for Variational Surface Modeling. *Computer Graphics Forum (Proc. SGP)*, *29*(5), 1565–1574.

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson Surface Reconstruction. In *Proc. SGP* (pp. 61–70).

Madgwick, S., Harrison, A., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm . In *IEEE Int. Conf. Rehabilitation Robotics* (pp. 1–7).

Markley, F. L., Cheng, Y., Crassidis, J. L., & Oshman, Y. (2007). Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, *30*(4), 1193–1197.

Markley, F. L., & Mortari, D. (2000). Quaternion attitude estimation using vector observations. *Journal of the Astronautical Sciences*, *48*(2), 359–380.

Milosevic, B., Bertini, F., Farella, E., & Morigi, S. (2016). A SmartPen for 3D interaction and sketch-based surface modeling. *The International Journal of Advanced Manufacturing Technology*, *84*(5), 1625–1645.

Moreton, H. P., & Séquin, C. H. (1991). Surface Design with Minimum Energy Networks. In *Proc. ACM/SIGGRAPH Symposium on Solid Modeling and CAD/CAM Applications* (pp. 291–301).

Nealen, A., Igarashi, T., Sorkine, O., & Alexa, M. (2007). FiberMesh: Designing Freeform Surfaces with 3D Curves. *ACM Trans. Graph. (Proc. SIGGRAPH)*, *26*(3).

Nielson, G. M. (2004). ν-quaternion splines for the smooth interpolation of orientations. *IEEE Trans. Vis. Comput. Graphics*, *10*, 224–229.

Pacha, A. (2013). Sensor Fusion for Robust Outdoor Augmented Reality Tracking on Mobile Devices. Master's thesis.

Park, F. C., & Ravani, B. (1997). Smooth invariant interpolation of rotations. *ACM Trans. Graph.*, *16*(3), 277–295.

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische Mathematik*, *10*(3), 177–183.

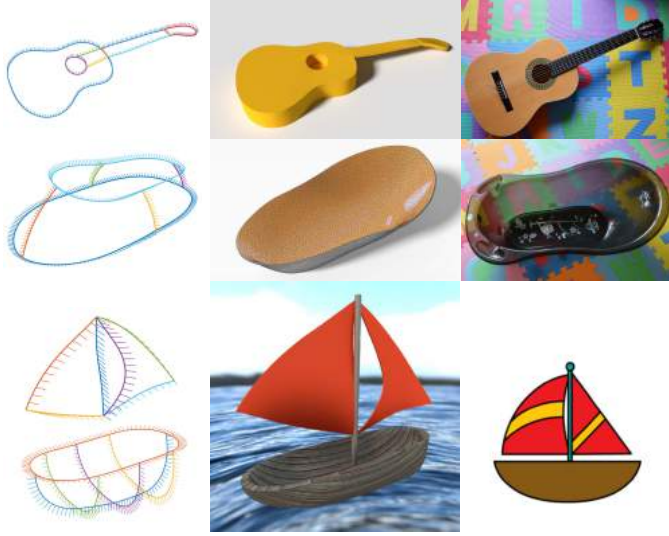Saguin-Sprynski, N., Carmona, M., Jouanet, L., & Delcroix, O. (2016).

Figure 20: Top: GUITAR and BATHTUB scanned using the Morphorider. Left to right, reconstructed networks, surfaces computed from the networks, the scanned objects. The GUITAR shape was created by extruding the surfaced network. Scanning the translucent BATHTUB was not an issue.
Bottom: inspired by the image on the right, we created a 3D model of the SAILBOAT by sketching the network on the left with a smartphone. The lengths were estimated from time of acquisition.

New generation of flexible risers equipped with motion capture – Morphopipe system. In *EWSHM – 9th European Workshop on Structural Health Monitoring*.

Saguin-Sprynski, N., Jouanet, L., Lacolle, B., & Biard, L. (2014). Surfaces Reconstruction Via Inertial Sensors for Monitoring. In *EWSHM – 7th European Workshop on Structural Health Monitoring*.

Shoemake, K. (1985). Animating rotation with quaternion curves. *Computer Graphics*, *19*(3), 245–254. Proc. SIGGRAPH.

Sprynski, N. (2007). *Reconstruction de courbes et de surfaces à partir de données tangentielles*. PhD thesis, Université de Grenoble.

Sprynski, N., Szafran, N., Lacolle, B., & Biard, L. (2008). Surface reconstruction via geodesic interpolation. *Computer Aided Design*, *40*(4), 480–492.

Stanko, T., Hahmann, S., Bonneau, G.-P., & Saguin-Sprynski, N. (2016). Surfacing curve networks with normal control. *Computers & Graphics*,

60, 1–8.

Xu, B., Chang, W., Sheffer, A., Bousseau, A., McCrae, J., & Singh, K. (2014). True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Trans. Graph. (Proc. SIGGRAPH)*, *33*(4), 131:1–131:13.

data	samples		time (ms)		
	raw	uni	conv.	regress.	Poisson
MUSHROOM Fig. 2	831	508	10	202	8
LILIUM Fig. 14 top	3243	96	3	51	1
		196	6	113	3
		382	6	234	7
		742	9	439	30
		1416	15	788	170
LILIUM Fig. 14 bottom	531	99	3	28	2
		186	6	55	3
		362	8	83	10
		713	11	192	31
		1420	16	428	172
CONE Fig. 15	949	205	5	176	3
		398	3	328	8
		783	8	618	28
		1553	14	1169	136
		3094	22	2779	904
CHAIR Fig. 16	584	247	3	592	6
				1315	
				1461	
LILIUM Fig. 18	3243	428	7	125	10
				319	
				517	
				522	

Table 2: Performance of the algorithm implemented in C++ using ROPTLIB on a standard quad-core CPU computer. The stats include number of raw samples, number of uniform samples, Gaussian convolution time, SO(3) regression time, Poisson network reconstruction time. All timings are in milliseconds.