Voronoi-based Curvature and Feature Estimation from Point Clouds

Quentin Mérigot, Maks Ovsjanikov, and Leonidas Guibas

Abstract—We present an efficient and robust method for extracting curvature information, sharp features and normal directions of a piecewise smooth surface from its point cloud sampling in a unified framework. Our method is integral in nature and uses convolved covariance matrices of Voronoi cells of the point cloud which makes it provably robust in the presence of noise. We show that these matrices contain information related to curvature in the smooth parts of the surface, and information about the directions and angles of sharp edges around the features of a piecewise-smooth surface. Our method is applicable in both two and three dimensions, and can be easily parallelized, making it possible to process arbitrarily large point clouds, which was a challenge for Voronoi-based methods. In addition, we describe a Monte-Carlo version of our method, which is applicable in any dimension. We illustrate the correctness of both principal curvature information and feature extraction in the presence of varying levels of noise and sampling density on a variety of models. As a sample application, we use our feature detection method to segment point cloud samplings of piecewise-smooth surfaces.

Index Terms—Computational Geometry, Object Modeling

1 Introduction

Estimating surface normals, principal curvatures and sharp edges from a noisy point cloud sampling has many applications in computer graphics, geometry processing and reverse engineering. Principal curvatures are rotation-invariant local descriptors, which together with principal curvature directions have proven useful in detecting structural regularity [1], global matching [2], modeling and rendering of point-based surfaces [3], and anisotropic smoothing [4] to name just a few. In these applications, various notions of curvature serve as local descriptors that encode second order variations of the surface. The location of sharp edges and highly curved areas of the surface is a precious piece of information in settings that include feature-aware reconstruction [5], non photorealistic rendering [6], and industrial metrology.

In practice, it is often interesting to recover this information when the input is an unstructured collection of point coordinates, obtained by a range scanner, before attempting surface reconstruction. These point clouds can be noisy, and can exhibit strong sampling bias. The ability to reliably estimate surface normals, principal curvatures, and curvature directions as well as sharp features directly on such point clouds can be used in both geometry processing algorithms and in surface reconstruction to improve the quality of the resulting mesh.

Devising robust local descriptors, which can handle both non-uniform noise, sampling bias and sharp edges is a challenging task. This is mainly because we are trying to estimate differential quantities which by nature are very sensitive to local perturbations. The lack of a natural parametrization of point cloud data introduces another challenge by making it difficult to estimate angles and areas on the surface. Finally, devising a method with theoretical guarantees on the approximation quality is not easy in the absence of a definition of curvature and sharp features that that would incorporate both point clouds and piecewise-smooth surfaces.

In this paper, we address some of these challenges by presenting a method for robustly estimating the location and direction of sharp edges as well as highly curved area from a possibly noisy point cloud sampling. We also show that the same method can be used to recover curvature information of the underlying surface, when the sampling is dense enough. The sharp edge estimation technique comes with theoretical guarantees on the quality of the results as a function of the Hausdorff distance between the point cloud and the underlying surface. We also address a certain class of outliers.

Prior work on feature and curvature estimation

The questions of curvature estimation and sharp feature detection are tightly related: sharp edges and corners can be thought as parts of the surface with infinite concentration of curvature (mean and Gaussian, respectively). Surprisingly, however, these research topics have known very different developments; we review the existing results separately.

Curvature estimation on meshes

Estimating the curvature of a smooth surface from a mesh is an important question in discrete differential geometry has been studied for many years and is now

- *Q. Mérigot is with Laboratoire Jean Kuntzmann, CNRS/Université Grenoble I, 51 rue des Mathématiques Campus de Saint Martin d'Hères BP 53, 38041 Grenoble Cedex 09, France*
- *M. Ovsjanikov and L. Guibas are with the Computer Science Department, Stanford University, Stanford CA, CA 94305.*

Manuscript received February 18, 2010; revised August 30, 2010.

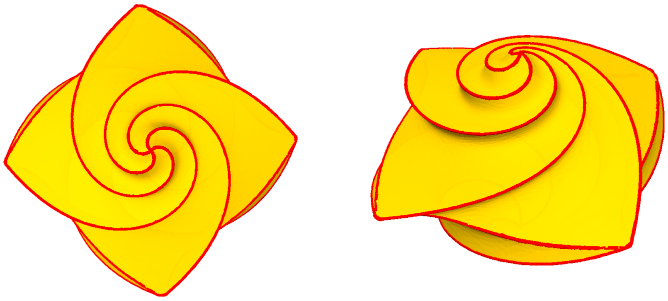


Fig. 1. Features (in red) computed by our algorithm from a point cloud sampling of the surface in yellow.

a well understood topic (see [7] for an extensive comparison of methods to estimate Gaussian and mean curvatures). The approach commonly used is to define a notion of curvature for meshes, and to study how well it approximates the real curvature of the underlying surface. The usual requirement is that (i) the Hausdorff distance between the mesh and the surface is low, and (ii) the normals to the mesh do not deviate from those of the surface too much.

Normal cycle. Cohen-Steiner and Morvan [8] used the normal cycle from geometric measure theory in order to define a closed-form second fundamental measure of a mesh. They also explicitly bound the error between the second fundamental measure of the mesh and the underlying surface, supposing (i) and (ii). This typically applies to the case of a mesh obtained by a Delaunay-based reconstruction of the surface from an ε-sample. (It should also be noted that the second fundamental measure can be used to compute reliably approximate the curvature of *offsets* of the underlying surface from a point cloud approximation [9].

Integral methods. A very different approach estimates the curvature and sharp features location from a mesh approximation of a surface S that bounds a domain D, by considering the volume and covariance matrices of intersections of small balls with the domain D [10], [11]. This approach can be used both for curvature estimation and sharp features detection. Another interesting aspect using an approximation D' of the domain D yields an error that depends on the volume of the symmetric difference $D \Delta D'$ only. This makes these integral methods resilient to noise in the normal directions. However, they require a knowledge of both the surface and the interior domain, and cannot be applied to point clouds straightforwardly.

Curvature estimation from points clouds

Only recently several methods have been proposed for computing curvature and feature information on the point cloud directly. One class of methods for computing curvature information from a point cloud sampling work by estimating directional curvatures in the neighborhood of a point and using them to estimate the principal curvatures and curvature directions: [4], [12]. Robust methods such as least squares [12] and local re-weighting [4] help these methods cope with noise and irregular sampling. Berkmann et al. [13] propose a method that rely on computing the covariance matrices of normal vectors to approximate the shape operator. More recently, Kalogerakis et al. [14] use robust statistical methods to fit the curvature tensor of a point cloud by considering normal variation in a neighborhood of a given point. Yang and Qian [15] derive analytic expressions for computing principal curvatures based on the implicit definition of the moving least squares surface by Amenta and Kil [16].

These algorithms start by estimating normals to the surface, or assume that they are given. Any error made in this estimation is therefore only aggravated in the computation of principal curvatures. A possible way to compute the normal and the second fundamental form at the same time is to fit local jets (see Cazals and Pouget [17]). Least-square jet-fitting is very fast, and its convergence properties are well understood. Unfortunately, a necessary condition for this method to give correct estimations is that the point cloud should be uniformly sampled. This can be a problem for some point clouds such as laser scans which are virtually never free of noise and often exhibit oversampled clusters of points along horizontal lines.

Sharp features estimation from point clouds

Although extracting sharp edges and corners is closely related to curvature estimation, research in these areas has been relatively independent. Fleishman et al. [18] detect sharp edges and corners by segmenting neighborhoods of points into regions corresponding to the same part of the surface. They achieve robustness by using a forward search technique which finds reference planes, corresponding to each segment. This work is extended by Daniels et al. [19] to extract feature-curves, by locally projecting onto feature points, and growing smooth polylines through the projected cloud. Lipman and colleagues [20] extract sharp edges within the MLS projection framework by examining the error of the MLS surface approximation. Jenke et al. [5] detect sharp features by robustly fitting local surface patches and computing intersections of nearby patches with dissimilar normals. In a similar spirit, Oztireli et al. [21] define a feature-preserving MLS projection operator by noting that angles between normal vectors, rather than point coordinates can be used to discard points from unrelated parts of a piecewise-smooth surface.

To sum things up, there is currently no provably-correct algorithm for estimating the sharp features of a piecewise smooth surface from an unoriented point cloud (i.e. without normals).

Contributions

In this paper, we associate to every point cloud C (or surface S) its *Voronoi covariance measure* $\mathcal{V}_{C,R}$ (resp. $\mathcal{V}_{S,R}$), from which one can extract information on the feature and curvature of the underlying surface. This measure is defined, following the Voronoi-based normal estimation techique of Alliez et al [22], by considering the covariance matrices of Voronoi cells. However, instead of intersecting them with a large bounding box of the point cloud C, we intersect them with an offset C^R, thus obtaining more *local* information about the variation in shape and size of the Voronoi cell, which is crucial for feature estimation. We present two algorithms for computing the Voronoi covariance measure: a Monte Carlo method, and a method based on tessellation (§3).

In order to gain robustness to Hausdorff noise, we have to consider convolved versions of the Voronoi covariance measure. For any positive convolution radius r, this defines a function $\mathcal{V}_{C,R} * \chi_r$ that maps points of $\mathbb{R}^d$ to symmetric matrices. The Stability Theorem (Theorem 5.1), asserts that replacing the convolved VCM of a surface S by the convolved VCM of a discrete approximation C yields an error of $\mathrm{O}(\mathrm{d}_H(C,S)^{1/2})$. We show in §6 that the eigenvectors of the convolved VCM of a smooth surface are the normal to the surface followed by the principal curvature directions; and experimentally, this information remains in the VCM of point clouds that approximate S densely and without noise. Much more interestingly, it appears that the principal curvature directions computed this way at *highly curved* parts of the surface remain good even when the sampling is very anisotropic or when Hausdorff noise is added to the point cloud.

Our main contribution is an algorithm that estimates the location and direction of sharp edges of a piecewise-smooth surface S from a point cloud approximation (§7) from the eigendecompotision of the convolved VCM. The algorithm requires as input a (rough) lower bound on the one-sided reach of the surface, a notion defined in §7.1. The existence of such a lower bound is similar to the positive local feature size (lfs) assumption needed for most provable surface reconstruction techniques such as [23]. The algorithms uses a threshold parameter, which can be specified by the user and is explicitely related to the minimum expected dihedral angle between smooth parts of the surface.

The Stability Theorem ensures that the sharp feature detection algorithm recovers the correct location of edges and is robust to Hausdorff noise. We experimentally check this fact on different models, under various levels of noise. Finally, as a sample application, we use this feature detection method to segment point cloud samplings of piecewise-smooth surfaces.

2 Overview of the Method

2.1 Geometric intuition

The method we propose is motivated by the same intuition as Voronoi-based normal estimation techniques such as the work by Amenta and Bern [23] and Alliez et al. [22]. Recall that the Voronoi cell of a point p in a point cloud C is the set of points of the space which are closer to p than to any other point in C. Denote this Voronoi cell by $\mathrm{Vor}_C(p)$.

Amenta and Bern remarked in [23] that in the absence of noise, if a point cloud samples a smooth surface densely enough, its Voronoi cells will be elongated in the direction of the normal to the surface 2(a). This allows to approximate the normal direction at a sample point p_0 by considering the shape of its Voronoi cell, by either computing its extremal points (poles) [23] or by performing principal component analysis [22]. Note, however, that the shapes of Voronoi cells provide more information than just the normal direction. Intuitively, the Voronoi cells at points sampled from highly curved areas will deviate from straight pencil shapes (which is well-approximated by a skinny cylinder) , and will be V-shaped with the angle related to the curvature of the underlying surface. Fig. 2(a) illustrates this effect on points sampled from a smooth planar curve. The lengths of the principal directions of each Voronoi cell, depicted in orange, are drawn proportional to the ratio $\lambda_{\min}/\lambda_{\max}$, of the two eigenvalues of the covariance matrix of the corresponding Voronoi cell, intersected with a large bounding box. Note that this ratio becomes larger for points sampled from highly curved areas of the shape, and therefore, it should not be surprising that the shape of the Voronoi cells, which can be captured by its covariance matrix, is related to the curvature of the underlying surface.

The deviation of Voronoi cells from being pencil shaped is most pronounced at points sampled from sharp corners and sharp edges of the surface, as shown in Fig. 2(b) and Fig. 6. Note that when the underlying surface is not smooth, some of its points will have normal cones rather than single normal directions. Nevertheless, even in this case, the shapes of Voronoi cells accurately reflect the shapes of the underlying normal cones, that can be characterized by the principal components of their covariance matrices. This observation is the key to allowing us to study both the smooth and piecewise-smooth surfaces in a unified framework.

One of the principal drawbacks of using Voronoi-based methods directly, is their sensitivity to noise. As shown in Fig. 2(c), even moderate noise can change the shapes of Voronoi cells dramatically. A beautiful insight made by Alliez et al. [22] is that although individual Voronoi cells can become ill shaped, by considering the union of Voronoi cells of a few neighbors, one can recover the correct normal directions. In their work, Alliez et al. used covariance matrices of the union of Voronoi cells to combat noisy sampling. One drawback of this approach is that taking the union unnecessarily produces more isotropic (fat) cells. A way to overcome this effect is to take a weighted average of covariance matrices of a few neighboring Voronoi cells instead of taking the covariance matrix of their union. In the continuous case, this average directly translates to convolution. In this paper we show that these averaged covariance matrices capture information about the shapes of normal cones of

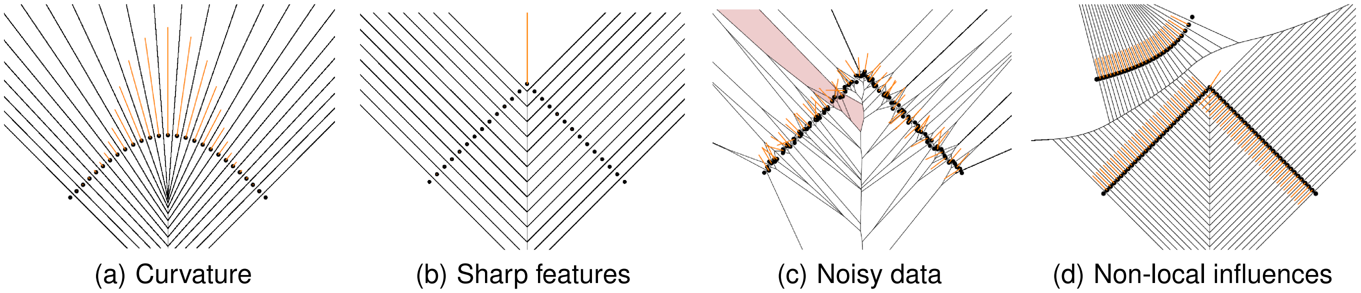


Fig. 2. The ratio of the two eigenvalues of the covariance matrix of a Voronoi cell (length of the direction in orange) contains curvature information (a). In the limit, Voronoi cells of points sampled from sharp corners are close to being isotropic (b). Noisy data (c) and non-local parts of the surface (d) can perturb the Voronoi cells, thus preventing correct curvature and sharp features estimations. Our method recovers from these effects using convolution and intersection with an offset.

the underlying surface.

Because of the global nature of the Voronoi diagram, points sampled from remote parts of the shape can influence the shapes of Voronoi cells of points even in absence of noise, thus leading to erroneous estimations of the underlying normal cones. Fig. 2(d) illustrates this effect. Note that the Voronoi cell of the point at the tip of the sharp corner is "invaded" by Voronoi cells of points from a different curve, which adversely influences the computation of the normal direction at the point. Note that this effect can occur even due to remote points of the same curve, is very global in nature and therefore difficult to rule out locally. To overcome this problem, we compute the covariance matrix of the Voronoi cell of each point p *intersected* with a small ball around p. This allows us to obtain purely local information about the variation of the surface around p, which is crucial for curvature and feature estimation.

2.2 Voronoi covariance of a point cloud

As described in the previous section, our algorithm aims to extract information about curvature and sharp features of the underlying surface from its point cloud sampling C, by analyzing the shapes of Voronoi cells of C intersected with small balls around each point. A way to get information about the shape of a domain E of $\mathbb{R}^d$ with finite volume is through its covariance matrix with respect to a base point p, denoted by $\mathrm{cov}(E,p)$. Specifically, the eigenvectors of $\mathrm{cov}(E,p)$ capture the principal axes of E while the ratio of the eigenvalues gives information about the anisotropy of E. The covariance matrix of a domain E can be computed through the following formula: $\mathrm{cov}(E,p) := \int_E (x-p)(x-p)^\intercal \mathrm{d}x$.

2.2.1 Definition

Given a point cloud C, and an offset radius R, we define the Voronoi covariance measure (VCM) of a point $p \in C$ at scale R to be:

$$\mathcal{V}_{C,R}(\{p\}) := \mathrm{cov}(\mathrm{Vor}_C(p) \cap \mathrm{B}(p,R), p) \qquad (2.1)$$

The VCM $\mathcal{V}_{C,R}(\{p\})$ of any point p in C is a symmetric matrix that captures information about the shape of the Voronoi cell of p. As mentioned earlier, however, Voronoi cells and as a result VCM can be unstable under noisy sampling. Thus, we define the *convolved Voronoi covariance measure* at a point x (not necessarily in C) by summing the covariance matrices defined above among all points $p \in C$ in a neighborhood of x:

$$\mathcal{V}_{C,R} * \chi_r(x) := \sum_{p \in \mathrm{B}(x,r) \cap C} \mathcal{V}_{C,R}(\{p\}) \qquad (2.2)$$

Intuitively, the convolved VCM of a point near the underlying surface S provides information about the normals and curvature of S.

2.3 Outline

In this paper, we demonstrate the practical implications of the Voronoi covariance measure. First, we indicate how to compute it efficiently from three-dimensional point clouds (§3). In §4 we give a general definition of the VCM suitable for any compact set, and prove a stability theorem of VCM under arbitrary Hausdorff noise (§5). Using the stability of the VCM, we devise robust curvature estimation and feature extraction algorithms (§ 6,7). Finally, we apply our feature extraction method to segment samplings of piece-wise smooth surfaces. (§8).

3 Computing VCM

In this section, we describe two algorithms for computing the VCM of a point cloud in practice. The first method is easy to implement and is applicable in any ambient dimension. The second algorithm is much faster, and is the one we used for all of our results. We then describe a straightforward way to convolve the VCM.

Given a point cloud $C = \{p_q, \dots, p_n\}$, computing the VCM of any point p_i at scale R amounts to evaluating the integral $\mathrm{cov}(B_i, p_i) = \int_{B_i} (x-p)(x-p)^\intercal \mathrm{d}x$ where B_i is the intersection $\mathrm{B}(p_i,R) \cap \mathrm{Vor}(p_i)$ between a ball of radius R centered at p_i and the Voronoi cell of p_i. No simple closed-form expression seems to exist for the covariance matrix of the intersection B_i, and we have to resort to approximation to compute it.

3.1 Monte-Carlo approximation of the VCM

The Voronoi covariance measure is a modification of the *boundary measure* introduced in [24]. In this work, instead of considering the covariance matrix $\mathrm{cov}(B_i)$, the authors only considered its volume. The Monte-Carlo algorithm for computing the boundary measures can be easily adapted to compute the VCM.

The input of Algorithm 3 is a point cloud C, an offset radius R and a number of steps N. For every point p in C it maintains a matrix $C(p)$ which approximates the covariance matrix $\mathrm{cov}(\mathrm{Vor}(p) \cap \mathrm{B}(p,R), p)$ as the number of steps N grows to infinity. The arguments presented in [24, p.6] show that the output of this algorithm converge to the real VCM with high probability: computing an ε-approximation of the VCM with high probability (e.g. 99%) can be done in $N = \mathrm{O}(|C| \ln(1/\varepsilon)/\varepsilon^2)$ steps.

This algorithm only uses nearest-neighbor queries and is easy to implement in any ambient dimension. However it is too slow in practice for computing the VCM of point clouds with hundreds of thousands of points. In the following, we describe a deterministic approach that can be used to improve the computation speed in low dimensions.

3.2 Approximating the VCM in 3D by tessellation

We base our second method on the fact that the covariance matrix of a tetrahedron can be computed analytically [22]. Therefore, using the additivity of the integral, in order to compute the covariance matrix of the intersection of a Voronoi cell with a ball, it is sufficient to approximate it with a union of tetrahedra. For this, we triangulate the boundary of this intersection and build tetrahedra by connecting each of these triangles to the center of the Voronoi cell — this can be done because the intersection is convex.

We start with an arbitrary triangulation of the boundary of the Voronoi cell $\partial(\mathrm{Vor}(p_i))$. Our goal is to subdivide each triangle so that its projection onto the ball $\mathrm{B}(p_i, R)$ is a sufficiently good approximation of the corresponding spherical triangle. For this, we process each triangle t in the original triangulation according to three simple rules (Algorithm 4).

The output of this algorithm yields a tetrahedrization of the intersection of the Voronoi cell with a ball of a given radius, centered at the same point. We then use the closed-form formulas of [22] to compute the covariance matrix of this intersection.

Fig. 3. Monte-Carlo algorithm for VCM

```
1: For all $p \in C$, set $C(p)$ to the zero matrix, set $M \leftarrow 0$.
2: Repeat $N$ times :
    2a: Select a random point $p$ in $C$ and generate a random point $x$ in the ball $\mathrm{B}(p, R)$.
    2b: $k \leftarrow$ number of points of $C$ in the ball $\mathrm{B}(x, R)$.
    2c: Find the nearest neighbor $p'$ of $x$ in $C$, and set $C(p') \leftarrow C(p') + \frac{1}{k}(x - p')(x - p')^{\intercal}$, $M \leftarrow M + \frac{1}{k}$.
3: For all $p \in C$, set $C(p) \leftarrow \frac{C(p)}{M}$.
```

TABLE 1
Computation times for VCM (in sec., on a 3GHz CPU)

Model	Delaunay	Tessellation	Total
Blade (195k)	23.73	90.82	114.55
Bimba (502k)	79.04	305.42	384.46
Nicolò (947k)	95.08	465.53	560.61

Convolution of VCM

Computing the convolved VCM of a point x simply amounts to summing the VCM matrices of all points p of C lying in the ball $\mathrm{B}(x, r)$. The points belonging to this intersection $\mathrm{B}(x, r) \cap C$ are the k nearest neighbors of x in C for a suitable value of k. The value of k and the corresponding neighbors can be determined by a binary search, using a structure adapted to k-NN queries (such as a kD-tree).

3.3 Implementation

We implemented the two algorithms described above. The tessellation of the Voronoi cells was done using the 3D Delaunay Triangulation package of CGAL [25]. Its running times of this algorithm on some standard point clouds is reported in Table 1. The convolution step is implemented using the ANN library [26], which includes a query for finding the set of points of a point cloud contained in a given ball. The time of the convolution step depends on the radius of convolution, but stays within 10 seconds for most models.

Parallelization

Note that both of the algorithms above can be easily parallelized. The Monte Carlo method relies on the ability to answer nearest neighbor queries, and therefore each of the N iterations of the method can be performed on a separate processor once a single shared nearest neighbor data structure is computed. This data structure can also

Fig. 4. Tetrahedrization of $\mathrm{Vor}_C(p_i) \cap \mathrm{B}(p_i, R)$

```
1: $\mathcal{T} \leftarrow$ triangulation of the boundary of $\mathrm{Vor}_C(p_i)$.
   $\mathcal{T}' \leftarrow \emptyset$
2: For all triangle $t \in \mathcal{T}$:
    A: If $t$ is completely outside the ball $\mathrm{B}(p_i, R)$:
        A1: Recursively subdivide $t$ into a family of smaller triangles $\{t_k\}$.
        A2: $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{t_k\}$
    B: If $t$ is completely inside the ball: $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{t\}$
    C: If $t$ crosses the sphere $\partial\mathrm{B}(p_i, R)$:
        C1: Subdivide $t$ by adding points along the circular arc of intersection $t \cap \partial\mathrm{B}(p_i, R)$
        C2: Apply A or B to the constructed triangles.
3: Return the tetrahedra joining $p_i$ and triangles $t \in \mathcal{T}'$
```

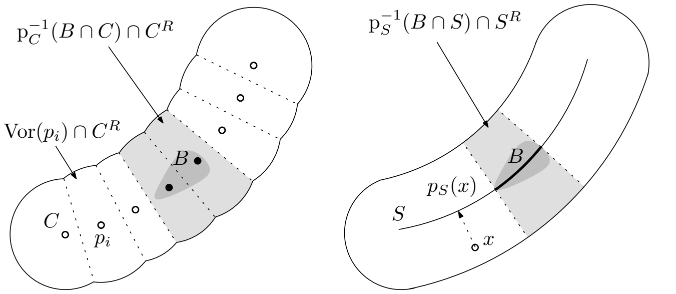

$\mathrm{p}_C^{-1}(B \cap C) \cap C^R$
$\mathrm{Vor}(p_i) \cap C^R$
B
C
p_i
$\mathrm{p}_S^{-1}(B \cap S) \cap S^R$
B
$p_S(x)$
S
x

Fig. 5. Voronoi covariance measures $\mathcal{V}_{C,R}(B)$ of a 2D point cloud C and of the underlying curve S, with respect to the same probing set B.

be used for convolution, which, again relies on nothing but nearest neighbor computations.

Note also that although the Voronoi diagram is a global object, whose computation is known to be difficult to parallelize, our tessellation-based method only requires *local* information, and can be parallelized as follows: when computing the VCM of a point p, rather than building a global Voronoi diagram, compute the Voronoi diagram of all points $q \in C$, s.t. $d(p,q) \leq 2R$. Since we are only interested in the intersection of the Voronoi cell of p with a ball of radius R, this set of points is sufficient to compute the VCM of p at scale R (the correctness of this observation follows from the proof of Lemma 5.2 below.) This means that we can limit the amount of memory necessary for computation by only processing batches of the point cloud that sample contiguous regions of the surface.

4 Voronoi covariance measure

In this section, we review the mathematical background necessary for the analysis of our method. We then define the Voronoi covariance measure of a general compact set, which will allow us to analyze both the robustness of the VCM with respect to noise, and its convergence properties in a single Stability Theorem (Th. 5.1).

4.1 General definition

In order to generalize the definition of VCM to any compact subset of $\mathbb{R}^d$, we make use of the notion of *projection function*. Given a compact set K of $\mathbb{R}^d$, the projection function maps any point x of $\mathbb{R}^d$ to its only[1] closest point in K. Denote this function by $\mathrm{p}_K : \mathbb{R}^d \to \mathbb{R}^d$. If C is a finite set of points $\{p_1, \dots, p_n\}$, the projection on C maps any $x \in \mathbb{R}^d$ to one of the points p_i in C whose Voronoi cells contain x. In particular, notice that $\mathrm{p}_C^{-1}(p_i)$ is exactly the Voronoi cell of p_i. By analogy, if K is any compact set, we will refer to the inverse image $\mathrm{p}_K^{-1}(B)$ of a subset B of K by the projection function p_K as the *Voronoi cell of* B (see Fig. 5). Likewise, we will call $\mathrm{p}_K^{-1}(\{p\})$ the *infinitesimal Voronoi cell* of the point p in K.

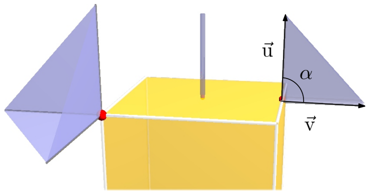


Fig. 6. The infinitesimal Voronoi cell $\mathrm{p}_C^{-1}(x)$ of a point x on a cube is pencil, triangle or cone-shaped depending on the dimension of the normal cone.

4.1.1 Definition

The Voronoi covariance measure of K with respect to K^R is a tensor-valued measure[2] denoted by $\mathcal{V}_{K,R}$. It maps every subset $B \subseteq \mathbb{R}^d$ to a positive semi-definite symmetric matrix $\mathcal{V}_{K,R}(B)$ defined by:

$$\mathcal{V}_{K,R}(B) = \int_{K^R \cap \mathrm{p}_K^{-1}(B \cap K)} (x - \mathrm{p}_K(x))(x - \mathrm{p}_K(x))^{\mathrm{T}} \mathrm{d}x \tag{4.3}$$

The domain of integration in the definition of $\mathcal{V}_{K,R}(B)$ is the Voronoi cell $\mathrm{p}_K^{-1}(B \cap K)$ of the subset B intersected with the chosen offset of K (see Fig. 5). In general, the matrix defined by (4.3) can be thought of as the covariance matrix of the domain of integration with *a varying base point*. Note that for a point cloud, this definition coincides with the definition given in §2.2.

4.1.2 Normal Cones

There is a strong relation between the notion of infinitesimal Voronoi cells and normal cones as defined in many fields of geometry, that is captured by the Voronoi covariance measure. For any p in K, let $\mathcal{N}_{p,R}K$ be the set of vectors defined by $\mathcal{N}_{p,R}K = \{x - p;\ x \in \mathrm{p}_K^{-1}(\{p\}) \cap K^R\}$. This set consists of all vectors that are normal to K at p, at a scale defined by the offset parameter R.

Remark that if S is a smooth hypersurface, and R is small enough, then $\mathcal{N}_{p,R}S$ is the segment $[-R\mathbf{n}(p), R\mathbf{n}(p)]$ (cf Fig. 5). Let B be a small neighborhood of the point p in S such as the ball $\mathrm{B}(p,r) \cap S$. Then, by definition, the matrix $\mathcal{V}_{K,R}(B)$ is the integral of all the covariance matrices of the normal cones $\mathcal{N}_{p,R}K$, with p in B. Therefore, this matrix captures the variation of the normals to the surface S around p, which is related to the curvature at p.

If, on the other hand, K is a convex polyhedron of $\mathbb{R}^3$, the set $\mathcal{N}_{p,R}K$ coincides with the normal cone at p as defined in convex geometry, intersected with the ball $\mathrm{B}(p,R)$. In particular, this set is two-dimensional if the point p lies on a edge of K and three-dimensional when p is a vertex of K (cf Fig. 6). The Voronoi covariance measure of K will reflect this fact, leading to the feature detection algorithm of presented in §7.

4.2 Convolved Voronoi covariance measure

The Voronoi covariance measure can be convolved by any *convolution kernel* $\chi : \mathbb{R}^d \to \mathbb{R}^+$, which turns it

[1] In reality, the projection function is uniquely defined everywhere but on the medial axis. Since the d-volume of the medial axis is zero, this has no consequences on our definition.

[2] Measure has to be understood as in Lebesgue measure theory, as a mass distribution on the space.

into a function denoted by $\mathcal{V}_{K,R} * \chi$. This operation smoothes the information contained in the covariance matrices, which is particularly useful when dealing with point clouds that can be noisy or that are anisotropically sampled. The value of the convolved VCM at a point p of $\mathbb{R}^d$ is the symmetric matrix defined by the formula:

$$\mathcal{V}_{K,R}*\chi(p) := \int_{K^R} (x-\mathrm{p}_K(x))(x-\mathrm{p}_K(x))^\intercal \chi(\mathrm{p}_K(x)-p)\mathrm{d}x \tag{4.4}$$

If χ_r is the *indicator function* of the ball $\mathrm{B}(0,r)$ i.e. $\chi_r(p) = 1$ if x belongs to $\mathrm{B}(0,r)$ and $\chi_r(p) = 0$ otherwise, then:

$$\mathcal{V}_{K,R} * \chi_r(p) = \mathcal{V}_{K,R}(\mathrm{B}(p,r)) \tag{4.5}$$

In this work, the convolution kernel is always chosen to be a Lipschitz approximation of such an indicator function, like the "hat function" $\chi(p) = \max(0, r - \|p\|^2)$. We also often use the indicator function itself for which we get good results even though the theoretical guarantees of convergence do not apply directly.

5 Robustness of the Convolved VCM

Throughout our analysis, we will be using the notion of Hausdorff distance, which can be defined for any two compact sets, K and K':

$$\mathrm{d}_H(K,K') = \inf_{\varepsilon} \text{ s.t. } K \subseteq K'^{\varepsilon} \text{ and } K' \subseteq K^{\varepsilon}$$

where $K^\varepsilon = \{x \in \mathbb{R}, d(x,K) < \varepsilon\}$.

For instance, if K is a point cloud sampling of a smooth surface K', then $\mathrm{d}_H(K,K')$ will be small provided that the sampling is dense enough and that there are no outliers. Note that this notion of distance is purely geometric and does not make any assumptions on the uniformity of the sampling, or on the sample points lying on the surface.

This section is devoted to the proof of robustness of the VCM. By robustness we mean that if two compact sets K and K' are very close in the Hausdorff sense, then their convolved VCM are also close. In applications, K will be a sampled surface, and K' will be a point cloud; however this result is very general and does not depend on the nature of these compact sets.

We will suppose that the convolution kernel χ is k-Lipschitz and bounded by a constant M. This means that $|\chi(x) - \chi(y)| \leqslant k\,\|x-y\|$ for every pair of points x, y of $\mathbb{R}^d$, and $|\chi(x)| \leqslant M$ for any point x. Under this assumption, it is possible to show that the convolved Voronoi covariance measure $\mathcal{V}_{K,R} * \chi$ depends continuously on the compact set K.

Theorem 5.1 (Stability Theorem for VCM). *Let χ be a bounded k-Lipschitz kernel. For every compact subset K of $\mathbb{R}^d$ and $R > 0$, there exists a constant $\mathrm{cst}(K,R)$ such that for every point x of $\mathbb{R}^d$,*

$$\|\mathcal{V}_{K,R} * \chi(p) - \mathcal{V}_{K',R} * \chi(p)\|_{\mathrm{op}} \leqslant \mathrm{cst}(K,R)\mathrm{d}_H(K,K')^{1/2}$$

where the operator norm $\|\cdot\|_{\mathrm{op}}$ is the standard matrix norm induced by the Euclidean metric.

Proof: Let E be the intersection between the offsets K^R and K'^R. Thanks to [24, Corollary 4.4], we know that the volume of $K^R \setminus E$ behaves as $O(\mathrm{d}_H(K,K'))$. Hence, letting $\delta(x) = x - \mathrm{p}_K(x)$, it is safe to change the integration domain in the definition of VCM and make the approximation,

$$\mathcal{V}_{K,R} * \chi(p) \simeq \int_E \delta(x)\delta(x)^\intercal \chi(\mathrm{p}_K(x) - p)\mathrm{d}x$$

The goal is then to bound the operator norm of the difference: $M = \int_E P(x) - P'(x)\mathrm{d}x$, where P (resp. P') is defined by $P(x) = \delta(x)\delta(x)^\intercal\chi(\mathrm{p}_K(x) - p)$. We have

$$\begin{aligned} &P(x) - P'(x) \\ &\quad = \chi(\mathrm{p}_K(x) - p)\left(\delta(x)\delta(x)^\intercal - \delta'(x)\delta'(x)^\intercal\right) \\ &\quad + \left(\chi(\mathrm{p}_K(x) - p) - \chi(\mathrm{p}_{K'}(x) - p)\right)\delta'(x)\delta'(x)^\intercal \end{aligned} \tag{5.6}$$

Using the definition of δ, one can factor the first term of this sum:

$$\begin{aligned} \chi(\mathrm{p}_K(x) - p)\,\big((\mathrm{p}_K(x) - x)\,&(\mathrm{p}_K(x) - \mathrm{p}_{K'}(x))^\intercal \\ &+ (\mathrm{p}_{K'}(x) - \mathrm{p}_K(x))\,(x - \mathrm{p}_{K'}(x))^\intercal\big) \end{aligned} \tag{5.7}$$

Recall that the kernel χ is bounded by M and that $\|\mathrm{p}_K(x) - x\| \leqslant R$ for every $x \in E$. Using the triangle inequality and these facts, the operator norm of (5.7) is bounded by $2RM\,\|\mathrm{p}_{K'}(x) - \mathrm{p}_K(x)\|$.

Thanks to the k-Lipschitz property for the kernel χ, the norm of the second term of the sum in (5.6) can be bounded by

$$\begin{aligned} |\chi(\mathrm{p}_K(x) - p) - \chi(\mathrm{p}_{K'}(x) - p)|\,&\|\delta'(x)\delta'(x)^\intercal\| \\ &\leqslant k\,\|\mathrm{p}_K(x) - \mathrm{p}_{K'}(x)\|\,R^2 \end{aligned}$$

Combining these two bounds in (5.6),

$$\|P(x) - P'(x)\|_{\mathrm{op}} \leqslant (2RM + kR^2)\,\|\mathrm{p}_K(x) - \mathrm{p}_{K'}(x)\|$$

Integrating this inequality on the set E yields

$$\|M\|_{\mathrm{op}} \leqslant (2RM + kR^2)\int_E \|\mathrm{p}_K(x) - \mathrm{p}_{K'}(x)\|\,\mathrm{d}x$$

The Projection Stability Theorem [24, Th. 3.2] asserts that $\int_E \|\mathrm{p}_K(x) - \mathrm{p}_{K'}(x)\|\,\mathrm{d}x \leqslant C(d,K,E)\mathrm{d}_H(K,K')^{1/2}$. This is enough to conclude. □

5.1 Consequences of the Stability Theorem

We now let χ_r be a Lipschitz approximation of the characteristic function of the ball $\mathrm{B}(0,r)$. If S is a piecewise smooth surface, p a point of S and C_n a sequence of point clouds converging to S in the Hausdorff sense, the Stability Theorem asserts that the convolved VCM $\mathcal{V}_{C_n,R} * \chi_r(p)$ of the point cloud C_n converges to the convolved VCM $\mathcal{V}_{S,R} * \chi_r(p)$ of S with respect to the the operator norm. It also quantifies the speed of convergence.

Using the standard results of matrix perturbation theory (see e.g. [27]), one then obtains the convergence of the eigenvalues and eigenvectors of $\mathcal{V}_{C_n,R} * \chi_r(p)$ to those of $\mathcal{V}_{S,R} * \chi_r(p)$, provided that the eigenvalues have multiplicity one. The speed of this convergence depends on the eigengap. As we will see in §6, at a smooth point p of the surface S the eigenvalues of $\mathcal{V}_{S,R} * \chi_r$ are proportional to 1, $\kappa_1^2(p)\frac{r^2}{4}$ and $\kappa_2^2(p)\frac{r^2}{4}$ where $\kappa_i(p)$ is the ith principal curvature of S at p. One can then expect a faster convergence rate for the estimated normal, and a faster convergence of the estimated principal curvature directions at points where the principal curvatures are very different (i.e. very non-umbilical points).

5.2 Robustness to some outliers

One limitation of the Hausdorff distance in the bound above, is its sensitivity to outliers. Indeed, even under controlled noise, outliers can contaminate the point cloud, and influence the shapes of the Voronoi cells. Nevertheless, as pointed out earlier, intersecting the Voronoi cells with an offset allows us to obtain local information which is unaffected by a certain class of outliers. The following Lemma shows that if the outliers are sufficiently far away from the point cloud, the convolved VCM will remain the same.

Lemma 5.2. *Let C be a point cloud and O a set of outliers with $\mathrm{d}(o, C) \geqslant 2R$ for any o in O. Then the convolved VCM $\mathcal{V}_{C,R} * \chi(p)$ and $\mathcal{V}_{C',R} * \chi(p)$ (where $C' = C \cup O$) agree on C provided that the support of χ is contained in a ball of radius smaller than R.*

Proof: See [28, Lemma 3.3]. □

6 Voronoi Covariance and Curvature

Let S be a compact smooth surface in $\mathbb{R}^3$ and $\mathbf{n}$ an oriented normal vector field on the surface. As before, let S^R denotes the R-offset of S. The relation between the Voronoi covariance measure and curvature relies on the following two facts:

(i) For any point x in the ambient space, $x = \mathrm{p}_S(x) \pm \mathrm{d}_S(x)\mathbf{n}(x)$, where $\mathrm{d}_S(x)$ is the distance from x to S.

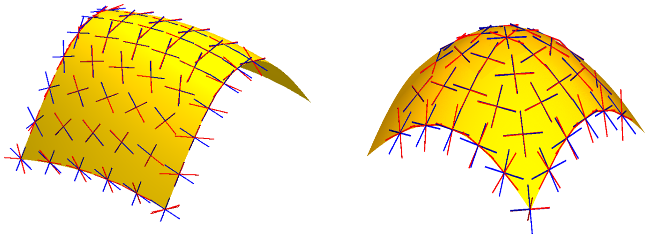
(ii) For any point p on the surface, let $\kappa_i(p)$ and $P_i(p)$, $i = 1, 2$, denote the two principal curvatures and principal curvature directions at p. Then the two eigenpairs of the derivative of the normal map $\mathrm{d}\mathbf{n}(p)$ are $(\kappa_i(p), P_i(p))$ for $i = 1, 2$.

Combining (i) and the change-of-variable formula, one can prove the following estimation:

$$\begin{aligned}
&\mathcal{V}_{S,R}(\mathrm{B}(p,r)) \\
&\quad = \int_{S^R \cap \mathrm{p}_S^{-1}(\mathrm{B}(p,r))} [\mathrm{d}_S(x)\mathbf{n}(\mathrm{p}_S(x))][\mathrm{d}_S(x)\mathbf{n}(\mathrm{p}_S(x))]^{\mathsf{T}}\mathrm{d}x \\
&\quad \simeq \int_{q \in \mathrm{B}(p,r) \cap S} \int_{-R}^{R} [t\mathbf{n}(q)][t\mathbf{n}(q)]^{\mathsf{T}}\mathrm{d}t\mathrm{d}q \\
&\quad \simeq \frac{2}{3}R^3 \int_{q \in \mathrm{B}(p,r) \cap S} \mathbf{n}(q)\mathbf{n}(q)^{\mathsf{T}}\mathrm{d}q
\end{aligned} \tag{6.8}$$

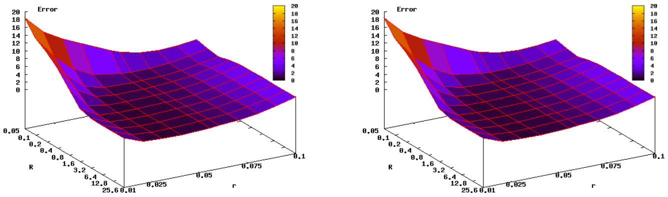
Putting (ii) in the first-order Taylor expansion of $\mathbf{n}$ gives

$$\begin{aligned}
\mathbf{n}(q) &\simeq \mathbf{n}(p) + \mathrm{d}\mathbf{n}(p)[q-p] \\
&\simeq \mathbf{n}(p) + \sum_{i=1}^{2} \kappa_i(p)\langle P_i(p) | q-p\rangle P_i(p)
\end{aligned} \tag{6.9}$$



(a) $z = \sin(3x) + \cos(x)$ (b) $z = \exp(-x^2) + \exp(-y^2)$

Fig. 7. Parametric surfaces with exact (in blue) and computed (in red) principal curvature directions. At the boundary computed directions follow the edges of the surface.



(a) $z = \sin(3x) + \cos(x)$ (b) $z = \exp(-x^2) + \exp(-y^2)$

Fig. 8. Average deviation (in degrees) of computed principal curvature directions from exact ones for different values of parameters R and r.

Then, using (6.8) and (6.9) one obtains the expansion

$$\begin{aligned}
&\mathcal{V}_{S,R}(\mathrm{B}(p,r)) \\
&\quad \simeq \frac{2\pi}{3}R^3r^2\left[\mathbf{n}(p)\mathbf{n}(p)^{\mathsf{T}} + \frac{r^2}{4}\sum_{i=1}^{2}\kappa_i^2(p)P_i(p)P_i(p)^{\mathsf{T}}\right]
\end{aligned} \tag{6.10}$$

More details on these computations are available in [28]. It follows from equation (6.10) that if the offset and convolution radii R and r are chosen small enough, the eigenvectors of the VCM near a smooth point are close to the principal curvature directions at that point.

6.1 Parametric Surfaces

To validate the theoretical guarantees presented above, we tested our method on parametric surfaces for which principal curvatures and principal curvature directions can be computed exactly.

We sampled two functions $z = \sin(3x) + \cos(y)$, with (x, y) in $[0, 1]^2$ and $z = \exp(-x^2) + \exp(-y^2)$, with (x, y) in $[-\frac{1}{2}, \frac{1}{2}]^2$. In both examples we used $100,000$ points that were chosen uniformly at random within the domain. Fig. 7 shows these two surfaces with the exact and computed principal curvature directions, using $R = 1$ and $r = 0.055$. As expected, away from the boundary the computed and the exact directions match very closely, except possibly at umbilical points (tip of the second surface). Near the boundary of the domain, the principal directions computed using our method follow the edges of the surface, which forms the basis of our feature detection method.

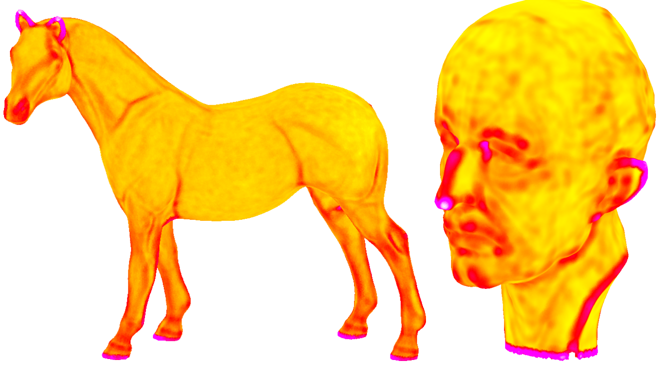


Fig. 9. Mean absolute curvature plot for the Horse and Max Planck models.

To measure the dependence of our method on the offset and convolution radii, we computed the average deviation (in degrees) of principal directions from exact ones. To minimize the effect of the points close to the boundary, where the directions are not expected to match, we only considered 90 percent of the data-set that is farthest from the boundary of the domain. Fig. 8 shows the average deviation for these two parametric surfaces. As can be seen, the results are quite stable for different choices of the parameters.

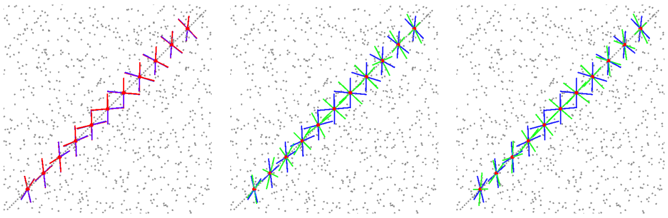
To illustrate the performance of our curvature estimation algorithm on complex shapes, we also plotted the mean absolute curvature: $|\kappa_1|+|\kappa_2|$ for the standard Max Planck and horse models (Figure 9), where the colors range from yellow to red and pink to signify low to high mean absolute curvature.

6.2 Comparison with Polynomial Fitting

6.2.1 Sampling Conditions

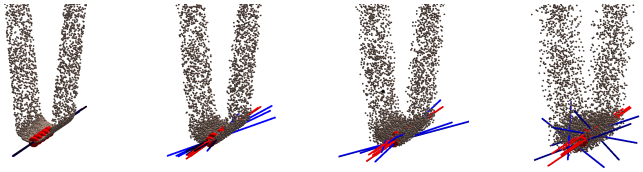
As mentioned in the introduction, the most common method of estimating principal curvatures and principal curvature directions on point clouds is to locally fit a polynomial of a given degree, and to analytically compute principal curvatures of this fitted polynomial [17].

Although these methods work well on noiseless and regularly sampled surfaces, polynomial fitting performs poorly if the data has strong noise and sampling bias. Our method, however, is oblivious to the sampling density, as a consequence of Hausdorff-robustness proved in section 5. To illustrate this, we added 50,000 points in a small band along the diagonal (in the parameter space) to the sampling of the surface shown in Fig. 7(a). The results obtained with our method and with the state of the art polynomial jet-fitting algorithm implemented in CGAL [17] are reported in Fig. 10. We used second order polynomial fitting with different neighborhood sizes k, which gave satisfactory results for the original sampling. As can be seen, the extra points do not affect the accuracy of our method. The results obtained with jet-fitting, however, become unreliable and strongly biased in the direction of the oversampling.



(a) Our method (b) Fit with k=100 (c) Fit with k=200

Fig. 10. Principal curvature directions on a biased dataset. Jet fitting *(b-c)* produces unreliable directions (in green) following oversampled areas.



(a) $r=0$ (b) $r=0.01$ (c) $r=0.04$ (d) $r=0.08$

Fig. 11. Principal curvature directions computed with our method (in red) are stable under noise. Directions computed by jet-fitting (in blue) are unreliable, especially when points from separate parts of the shape begin to mix.

6.2.2 Robustness

Other areas that are challenging for polynomial fitting algorithms include parts of the shape with high curvature, and regions where separate parts of the shape come in close contact, thus adversely influencing the quality of the fit. While the first problem can be addressed by fitting higher order polynomials, both of these settings are severely aggravated in the presence of noise. To illustrate this, we sampled a surface made by smoothly joining a small cylinder lying parallel to the z axis, with two planes on either side of the x axis. We used 0.1 as the radius of the cylinder, so the curvature at points along the z axis equals 10. Fig. 11 shows the principal curvature directions obtained on this model with varying levels of noise, using our method and using jet-fitting with 200 nearest neighbors and second order polynomial fitting. Note that when points from different parts of the shape mix, even the robust low order polynomial fitting fails, while our method preserves robustness

7 Detecting sharp edges using VCM

As remarked by Dey et al in [29], the shape of the Voronoi cell of a point p can be used to estimate the dimension of the part of the underlying shape this point belongs to. For instance, consider a point p lying on a sharp edge of a piecewise smooth surface S. As Fig. 6 shows, the infinitesimal Voronoi cell of p is triangle shaped, and the normal to the triangle is aligned with the direction of the sharp edge. Under good sampling conditions, the corresponding Voronoi cell will be very thin, and almost two-dimensional. Here, we show how to use the Voronoi covariance measure to translate this remark into a sharp feature detection algorithm, whose sampling conditions only involves the Hausdorff distance between the point cloud and the underlying surface.

Our main ingredient for this is to estimate the dimension of the infinitesimal Voronoi cell at a point p of the input point cloud by the number of “small” eigenvalues s of the convolved VCM of the cloud. This number can then be used to distinguish between a point lying on a face ($s = 2$), on a sharp edge ($s = 1$) and on a corner ($s = 0$). The actual meaning of “small” is made more precise in Proposition 7.1.

7.1 Theoretical derivation

We assume that the piecewise-smooth surface S we consider has a one-sided reach at least R, which we denote by $\mathrm{reach}_1(S) \geqslant R$. This means the following: for any point p on the surface S, and any unit vector v in the normal cone[3] to p, the projection of either $p + Rv$ or $p - Rv$ on the surface S should lie at p.

When a point p belongs to an edge of the surface S, denote by $\mathbf{u}_\pm(p)$ the leftmost and rightmost normals to the surface S at P. Also let $\mathbf{e}_1(p)$ be the mean of these two extreme normals, $e_1(p) = \frac{1}{2}(\mathbf{u}_+(p) + \mathbf{u}_-(p))$, and $\mathbf{e}_2(p)$ be orthogonal to $e_1(p)$ in the plane spanned by $\mathbf{u}_\pm(p)$, i.e $e_2(p) = \frac{1}{2}(\mathbf{u}_+(p) - \mathbf{u}_-(p))$. Finally, let $\alpha(p)$ be the positive angle between the two normals $\mathbf{u}_\pm(p)$ in the range $[0, \pi]$, and $\mathbf{t}(p)$ be the tangent direction to the edge.

Proposition 7.1. *Let S be a polyhedron of $\mathbb{R}^3$ whose 1-reach is lower bounded by R, and let p lie on a sharp edge of S. Then, the three eigenvalues of $\mathcal{V}_{S,R}(\mathrm{B}(p,r))$ are*

$$\begin{aligned}
\lambda_1(p) &= \frac{R^4 r}{8}\left[\alpha(p) + \sin\alpha(p) + \mathrm{O}(r/R)\right];\\
\lambda_2(p) &= \frac{R^4 r}{8}\left[\alpha(p) - \sin\alpha(p) + \mathrm{O}(r/R)\right];\\
\lambda_3(p) &= 0.
\end{aligned}$$

The corresponding eigenvectors are $\mathbf{e}_1(p)$, $\mathbf{e}_2(p)$ and $\mathbf{t}(p)$.

Proof: Suppose that r is small enough so that the only parts of the surface intersected by the ball $\mathrm{B}(p, r)$ are the edge E containing p and the two adjacent faces F_+ and F_-. Thanks to the additivity property of the VCM, one can write:

$$\begin{aligned}
\mathcal{V}_{S,R}(\mathrm{B}(p,r)) = \mathcal{V}_{S,R}(\mathrm{B}(p,r) \cap E) &+ \mathcal{V}_{S,R}(\mathrm{B}(p,r) \cap F_+)\\
&+ \mathcal{V}_{S,R}(\mathrm{B}(p,r) \cap F_-)
\end{aligned} \tag{7.11}$$

Using Lemma 7.2 below, we show that the term of (7.11) involving the edge, $\mathcal{V}_{S,R}(\mathrm{B}(p,r) \cap E)$, yields the constant parts of the eigenvalues $\lambda_1(p)$ and $\lambda_2(p)$. In order to get the $\mathrm{O}(R/r)$ approximation error, one just needs to bound the two terms of equation (7.11) involving the faces $F_\pm$:

$$\begin{aligned}
\|\mathcal{V}_{S,R}(\mathrm{B}(p,r) \cap F_\pm)\| &\leqslant \int_{F_\pm \cap \mathrm{B}(p,r)} \int_{-R}^{R} \|(t\mathbf{n}_\pm)(t\mathbf{n}_\pm)^{\mathsf{T}}\| \,\mathrm{d}t\mathrm{d}x\\
&= \pi\frac{r^2 R^3}{3} = \frac{R^4 r}{8}\,\mathrm{O}(r/R)
\end{aligned}$$

where $\mathbf{n}_\pm = \mathbf{u}_\pm(p)$ is the unit normal to the face $F_\pm$. □

Lemma 7.2. *Let $\mathbf{u}_\pm$ be two unit vectors in the plane, and Q denote the positive cone spanned by them intersected with the ball $\mathrm{B}(0, R)$, i.e $Q = \{a\vec{\mathrm{u}} + b\vec{\mathrm{v}}; a, b > 0\} \cap \mathrm{B}(0, R)$. The eigenpairs of the covariance matrix $\mathrm{cov}(Q, 0)$ are:*

$$\begin{aligned}
\lambda_1 &= \frac{R^4}{8}\left(\alpha + \sin(\alpha)\right),\ \mathbf{e}_1 = \mathbf{u}_+ + \mathbf{u}_- \quad \textit{and}\\
\lambda_2 &= \frac{R^4}{8}\left(\alpha - \sin(\alpha)\right),\ \mathbf{e}_2 = \mathbf{u}_+ - \mathbf{u}_-
\end{aligned}$$

where α is the angle between $\mathbf{u}_-$ and $\mathbf{u}_+$ in the range $[0, \pi]$.

Proof: Let $M = \mathrm{cov}(Q, 0)$, and consider the unit vector $\mathbf{d}(\beta)$ making an angle β with the vector $\mathbf{u}_-$. Then,

$$\begin{aligned}
\mathbf{d}^{\mathsf{T}}(\beta) M \mathbf{d}(\beta) &= \int_{\theta=0}^{\alpha} \int_{r=0}^{R} r^2 \cos^2(\theta - \beta) r \mathrm{d}r\mathrm{d}\theta\\
&= \frac{R^4}{8} \int_0^{\alpha} \cos(2(\theta - \beta)) + 1\mathrm{d}\theta\\
&= \frac{R^4}{8}\left(\sin(2(\alpha - \beta)) + \sin(2\beta) + \alpha\right)
\end{aligned} \tag{7.12}$$

By symmetry of Q, the vector $\mathbf{d}(\alpha/2)$ is necessarily an eigenvector of M. The second eigenvector is then $\mathbf{d}(\pi/2 + \alpha/2)$. To compute the eigenvalues associated with these eigenvectors, we plug in the values into (7.12). □

From Proposition 7.1, we deduce that the eigenvector corresponding to the smallest eigenvalue is the tangent direction to the feature, with an eigengap of $O(R/r)$.

7.1.1 Sharp corner of a surface

Similar calculations can be done if p is a vertex of the surface whose normal cone is isotropic. In this case, the two smallest eigenvalues will be equal, and contained in the plane orthogonal to the direction of the cone. The case where the normal cone isn't isotropic is more difficult to analyze precisely because the computations will depend on the sharp edge graph at p.

7.2 Description of our edge-detection method

The method needs three parameters: the offset radius R, the convolution radius r and a threshold T. It can be summarized as follows:

1: Compute the Voronoi covariance measure $\mathcal{V}_{C,R}$.
2: For every point $p \in C$:

2a: Compute $\mathcal{V}_{C,R}(\mathrm{B}(p, r))$ and diagonalize it.
2b: Sort the eigenpairs $(\lambda_i(p), \mathbf{e}_i(p))_{i \in \{1,2,3\}}$ by decreasing order of eigenvalues.
2c: Compute the ratio $r(p) = \lambda_2(p)/(\sum_i \lambda_i(p))$.
→ The point p is on a sharp edge if $r(p) \geqslant T$
→ The edge direction is given by $e_3(p)$.

[3] Recall that a vector v belongs to the normal cone to a point p of the surface S iff for some $\lambda > 0$, the projection of $p + \lambda v$ onto S is p.

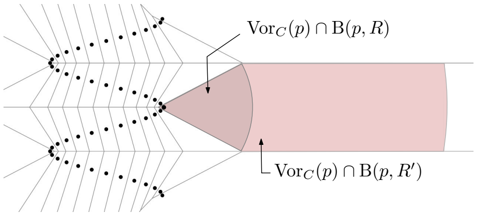

$\mathrm{Vor}_C(p) \cap \mathrm{B}(p, R)$
$\mathrm{Vor}_C(p) \cap \mathrm{B}(p, R')$

Fig. 12. Effect of the choice of offset radius on the anisotropy of the Voronoi of a point lying on a sharp feature.

7.2.1 Choice of parameters

The choice of the **offset radius** R should be done by considering the expected geometry of the underlying set. As Proposition 7.1 indicates, if the offset radius is less than the one-sided reach of the object, all of the sharp edges will be recovered by the method above. A too large offset parameter R can decrease the anisotropy of the Voronoi cell of a point that lies on a sharp feature (see Fig. 12), thus making it invisible to the method. From Theorem 5.1, one also sees that a higher value of the offset radius R will make the computations more robust. In conclusion, the offset radius should be chosen as the largest possible lower bound on the one-sided reach of the underlying surface.

The **convolution radius** r will be chosen depending on the expected density and amount of noise in the point cloud: it should be large enough so that every ball $\mathrm{B}(p, r)$ with p in C contains at least a few tens of points (eg. 50), and should definitely be larger than the amount of noise in the normal direction to the surface. However, a compromise should be made: the larger r, the less localized the detected sharp features are. We explore the dependence of the results on the choice of the convolution parameter in section 7.3.

We consider a point as a feature if the ratio of the second smallest eigenvalue to the sum of the three eigenvalues of the convolved Voronoi covariance measure is greater than some **threshold parameter**. This ratio provide a way to measures the thickness of the infinitesimal Voronoi cell. If p belongs to a sharp edge, Proposition 7.1 shows that

$$r(p) \simeq \frac{1}{2}\left(\frac{\alpha - \sin(\alpha)}{\alpha}\right) \simeq \frac{\alpha^2}{12}$$

Thus, given a threshold parameter T, a sharp edge with external angle α will be selected if $\alpha \geqslant 2\sqrt{3}T^{1/2}$.

Note that sometimes it may be necessary to detect only sharp edges filtering out corners as well as smooth parts of the shape. In this case, we use the ratio of the second smallest to the smallest eigenvalue. This ratio will be large for points around sharp edges, and close to one for points in the smooth areas as well as around the corners of the surface.

7.3 Experimental results

In order to assess the quality of the set of sharp edges detected by the method, we compare it to the set of sharp edges of the underlying surface.

TABLE 2
Distances between the detected features (D) and real features (R) of an icosahedron, with varying noise radius and convolution radius. The distances δ_∞, δ_1 and α_1 are between D and R while δ'_∞ and δ'_1 are between R and D

Noise	r	δ_∞	δ_1	α_1	δ'_∞	δ_1
0.0	0.05	0.35	0.037	3.25	0.076	0.011
0.0	0.1	0.118	0.051	0.33	0.124	0.016
0.02	0.1	0.226	0.049	1.65	0.139	0.020
0.05	0.1	0.220	0.050	2.82	0.155	0.025
0.1	0.15	0.271	0.069	3.12	0.178	0.036

7.3.1 Resilience to noise

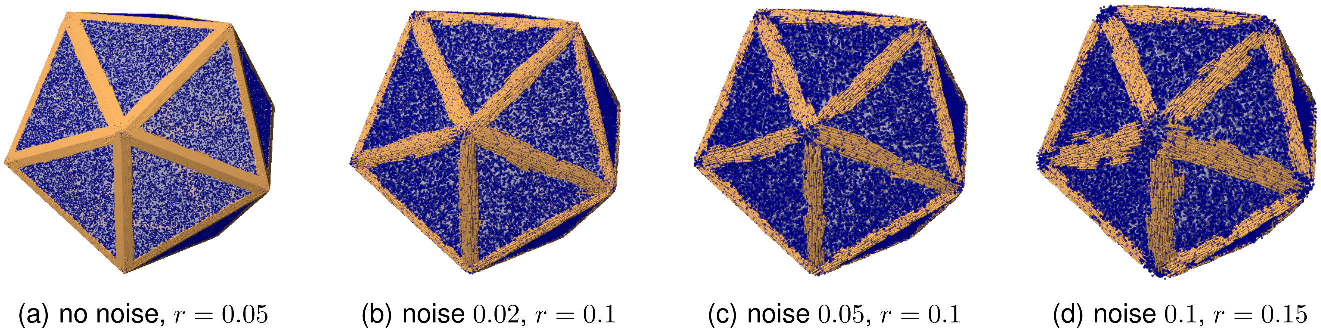
We evaluated the sharp edge estimation method and its resilience to noise on a unit icosahedron. We sampled 100k points randomly on it, ran the computations with $R = 20$ and various convolution parameters r. In order to test the resilience of our method to noise, the original 100k point cloud on the icosahedron is perturbed by adding to each point a random vector uniformly chosen in a ball of given radius (the radius of the noise). Fig. 13 shows the estimated feature directions on the icosahedron, for different noise and convolution radii.

We call *oriented set* a finite subset C of $\mathbb{R}^d$, each of whose points is endowed with a tangent direction $\mathbf{t}(p)$. We assess the similarity of two such sets C and C' using the following metrics. The *Half-Hausdorff distance* $\delta_\infty(C, C')$ (resp. average distance $\delta_1(C, C')$) is the maximum (resp. average) distance between a point in C and its nearest neighbor in C'. Finally, the *average angular deviation:* $\alpha_1(C, C')$ is the average angle (in degrees) between the direction $\mathbf{t}(p)$ of a point p in C and the direction $\mathbf{t}(p')$ of its nearest neighbor p' in C'.

Table 2 summarizes the values of the distances between the oriented set D of detected features, and the set R of real features, for various noise and convolution radii. The distance $\delta_\infty := \delta_\infty(D, R)$ measures the presence of isolated outliers, while $\delta_1 := \delta_1(D, R)$ measures the spread of the estimated features. One can see that increasing the convolution radius removes isolated outliers, but increases the spread of feature. The distances with primes (δ'_∞ and δ'_1) correspond to the same metrics when swapping the role of D and R. Both δ'_∞ and δ'_1 measure how "complete" the set of features is, i.e. how far points lying on actual sharp features are to the set of detected feature D. Most of the error here comes from the corners, that are discarded based on the ratio between the second and third eigenvalues.

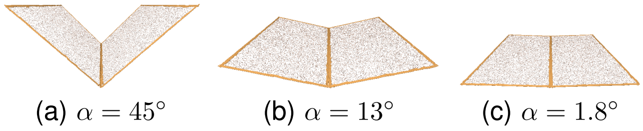
7.3.2 Sharp edges with low external angle

In order to understand the effect of sharpness of the edge on the quality of the feature detection, we sampled a surface made of two planar rectangular patches joined by a common edge and whose normals differ by an



(a) no noise, $r = 0.05$ (b) noise 0.02, $r = 0.1$ (c) noise 0.05, $r = 0.1$ (d) noise 0.1, $r = 0.15$

Fig. 13. Estimated feature directions on a unit icosahedron, with various convolution radii r and noise values.



(a) $\alpha = 45°$ (b) $\alpha = 13°$ (c) $\alpha = 1.8°$

Fig. 15. Estimated feature directions on a folded rectangle, for different values of the external angle α.

angle of 2α. As shown in Fig. 15, the feature estimation method described above is able to reliably detect sharp edges whose external dihedral angle is as small as $2°$; all the results were produced using the same convolution radius and threshold.

7.3.3 *Results on more complex point clouds*

On larger range scan point clouds, we were able to decrease both the offset and the convolution radii while keeping the detected features almost noiseless. This enables the algorithm to capture very small and non-sharp features, like the hair of Caesar or the braiding of Bimba, as shown in Fig. 14(a),14(b),14(c).

8 Surface Segmentation

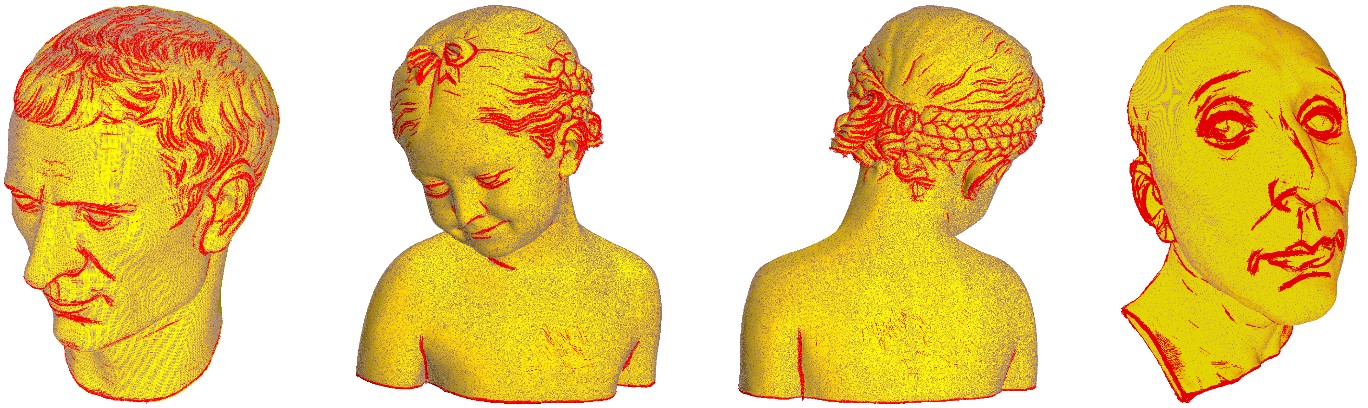
As a sample application to our feature detection method, we use it to segment point cloud samplings of piecewise smooth surfaces. Here, given a point cloud, our goal is to segment it into components sampled from contiguous regions of the surface separated by sharp edges and highly curved areas (see Fig. 16). The use of our feature detection method for point cloud segmentation was suggested by Frederic Chazal and Steve Oudot, who also contributed to the work presented in this section.

Shape segmentation has applications in reverse engineering and CAD model generation as well as surface reconstruction [5], rendering of point clouds [18], [20], and object recognition and matching [30]. As in the case of curvature and feature estimation, most existing methods rely on meshes (see [31] for a survey). Nevertheless, a rich body of work also exists in the context of point cloud segmentation with various targeted applications (e.g. [32], [33]). Unfortunately most existing methods lack a sound theoretical foundation. In this section, we demonstrate that our provably correct feature detection method can be used in shape segmentation. Note that rather than opposing existing techniques, our objective is to illustrate the performance of our feature detection method for shape segmentation and to make a step towards a provably correct segmentation technique.

A recent method for segmenting point clouds with theoretical guarantees on the number of components was proposed by Chazal and colleagues [34], using ideas from topological persistence. Given a scalar function f defined over a point cloud C, and a neighborhood parameter μ, their method starts by building a Rips graph of C by introducing an edge between any two points of C whose Euclidean distance is less than μ. Then, each point is associated to a local minima of the function f through a simple gradient descent on the Rips graph. The corresponding attraction basins are merged according to a persistence threshold parameter α. Using methods from the theory of topological persistence, the authors are able to prove that the number of components thus obtained is stable under perturbations of the function and irregular point cloud sampling of the underlying space.

We apply the method of Chazal et al. [34] to segment point clouds into smooth regions. To define a function on the point cloud, we use a two step process: first we identify sharp edges and corners using the method described above. Then, we build a Rips graph on the point cloud using the same μ parameter as defined earlier. For each point p in the cloud, define $g(p)$ as the geodesic distance in the Rips graph from p to the closest feature point. Since the method of Chazal et al. identifies local minima rather than maxima, we apply their algorithm to the function $f(p) = -g(p)$.

This process is illustrated in Fig. 16. The input point cloud is a 200k sampling of the Fandisk model 16(a). The feature points identified by our method are shown in Fig. 16(b); the negative distance function to the features in Fig. 16(c), and the final point cloud segmentation in Fig. 16(d). In Fig. 17, we show the results of this segmentation method on point cloud samplings of several other models. Note that one of the advantages of persistence-based clustering [34] is that this method is very global in nature. Therefore, even if spurious features are detected, e.g. due to a very noisy sampling, introducing additional



(a) Caesar

(b) Bimba

(c) Bimba

(d) Nicolo

Fig. 14. Results of the feature detection algorithm on the Julius Caesar *(a)*, Bimba *(b-c)* and Nicolò *(d)* point clouds.

local minima of the negative distance function, the segments associated with these minima will be merged with more persistent "true" local minima.

Conclusion and Future Work

In this paper, we have described a method for detecting sharp features and feature directions on point cloud data in a unified fashion. We have provided theoretical guarantees on its robustness, and implemented it and tested it on various point cloud data with varying amounts of sampling bias and noise.

In the future, it would be interesting to combine the theoretical stability properties of persistence-based clustering [34] with our feature detection method to give explicit guarantees on the quality of the segmentation method for point cloud samplings of piecewise-smooth surfaces described in §8.

One drawback of the Voronoi covariance measure is its sensitivity to outliers that are at intermediate distance from the point cloud. A possible way to overcome this limitation would be to use the recently introduced notion of distance-to-measure [35].

Acknowledgments

This work has been supported by INRIA/Stanford Associated Team TGDA, NSF grants FRG 0354543, TF 0634803 and FODAVA 808515, as well as by DARPA grant HR0011-05-1-0007. Some of the models are issued from the AIM@Shape Shape Repository. We would also like to thank the authors of [34] for sharing the implementation of their clustering algorithm.

References

[1] M. Pauly, N. Mitra, J. Wallner, H. Pottmann, and L. Guibas, "Discovering structural regularity in 3d geometry," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[2] D. Aiger, N. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[3] J. P. Gois, E. Tejada, T. Etiene, L. G. Nonato, A. Castelo, and T. Ertl, "Curvature-driven modeling and rendering of point-based surfaces," in *Braz. Symp. Comp. Graph. Imag. Proc*, 2006, pp. 27–36.

[4] C. Lange and K. Polthier, "Anisotropic smoothing of point sets," *Comp. Aided Geom. D.*, vol. 22, no. 7, pp. 680–692, 2005.

[5] P. Jenke, M. Wand, and W. Straßer, "Patch-graph reconstruction for piecewise smooth surfaces," in *Proc. of Vis., Mod., Vis.*, 2008.

[6] M. Pauly, R. Keiser, and M. Gross, "Multi-scale Feature Extraction on Point-Sampled Surfaces," in *Comput. Graph. Forum*, 2003.

[7] E. Magid, O. Soldea, and E. Rivlin, "A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data," *Comput. Vis. Image Und.*, vol. 107, no. 3, pp. 139–159, 2007.

[8] D. Cohen-Steiner and J. Morvan, "Restricted delaunay triangulations and normal cycle," in *Proc. ACM Symp. Comput. Geom*, 2003, pp. 312–321.

[9] F. Chazal, D. Cohen-Steiner, A. Lieutier, and B. Thibert, "Stability of Curvature Measures," in *Proc. Eurographics Symp. Geom. Proc.*, 2009.

[10] M. Connolly, "Measurement of protein surface shape by solid angles," *J. Mol. Graph.*, vol. 4, no. 1, pp. 3–6, 1986.

[11] H. Pottmann, J. Wallner, Q. Huang, and Y. Yang, "Integral invariants for robust geometry processing," *Comp. Aided Geom. D.*, vol. 26, pp. 37–60, 2009.

[12] J. Huang and C. H. Menq, "Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology," *Comp. Aided Geom. D.*, vol. 34, no. 2, pp. 149–165, 2002.

[13] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 11, pp. 1114–1116, 1994.

[14] E. Kalogerakis, D. Nowrouzezahrai, P. Simari, and K. Singh, "Extracting lines of curvature from noisy point clouds," *Special Issue of the Elsevier Computer-Aided Design journal on Point-Based Computational Techniques*, vol. 41, no. 4, pp. 282–292, 2009.

[15] P. Yang and X. Qian, "Direct computing of surface curvatures for point-set surfaces," in *Proc. Eurographics Symp. Point-based Graph.*, 2007.

[16] N. Amenta and Y. Kil, "Defining point-set surfaces," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 264–270, 2004.

[17] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," in *Comp. Aided Geom. D.*, 2005, pp. 121–146.

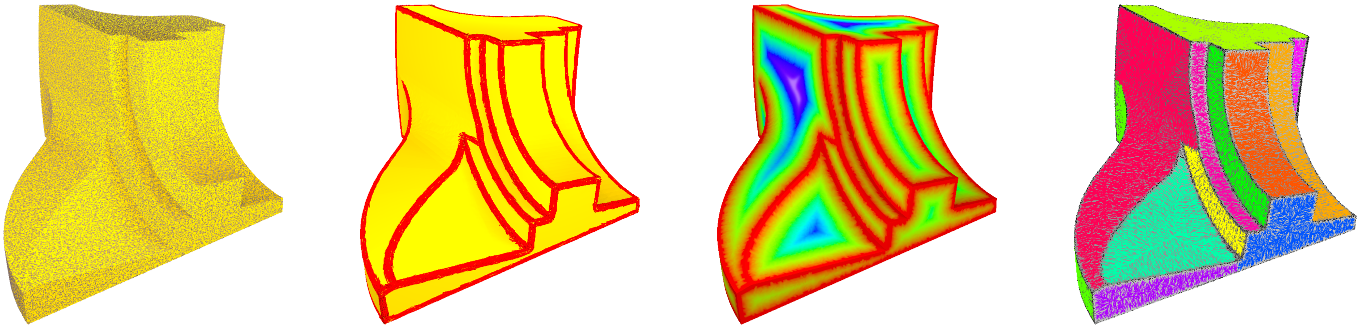
[18] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 544–552, 2005.

[19] J. Daniels Ii, T. Ochotta, L. Ha, and C. Silva, "Spline-based feature curves from point-sampled geometry," *The Visual Computer*, vol. 24, no. 6, pp. 449–462, 2008.

[20] Y. Lipman, D. Cohen-Or, and D. Levin, "Data-dependent MLS for faithful surface approximation," in *Proc. Eurographics Symp. Geom. Proc.*, 2007, pp. 59–67.

[21] C. Oztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Eurographics*, 2009.

[22] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun, "Voronoi-based variational reconstruction of unoriented point sets," in *Proc. Eurographics Symp. Geom. Proc.*, 2007, pp. 39–48.



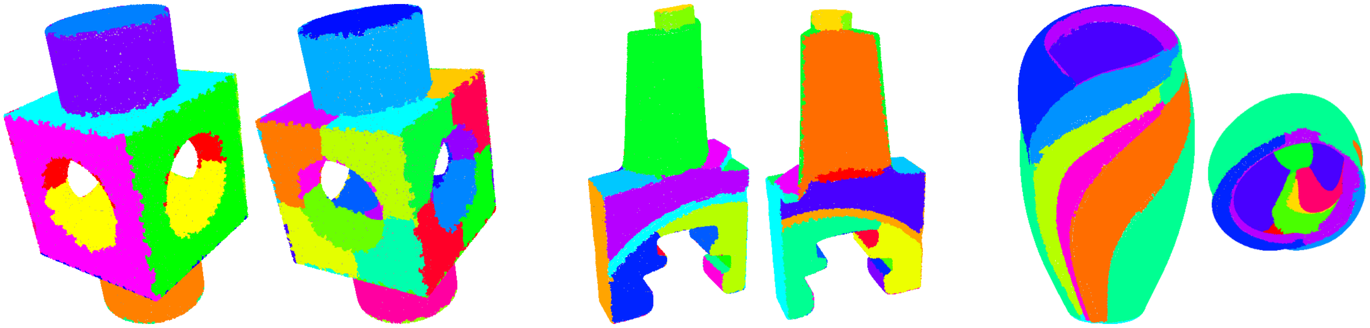
(a) Input point cloud rendered on the mesh

(b) Identified feature points

(c) Distance function to the features

(d) Output segmentation

Fig. 16. A segmentation of a point cloud sampling of the fandisk model using our feature detection method and the clustering algorithm of Chazal et al [34].



(a) Block

(b) Blades

(c) Vase

Fig. 17. Segmentations of point cloud samplings of the different models using our method.

[23] N. Amenta and M. Bern, "Surface reconstruction by voronoi filtering," *Discrete Comput. Geom*, vol. 22, pp. 481–504, 1999.

[24] F. Chazal, D. Cohen-Steiner, and Q. Mérigot, "Boundary measures for geometric inference," *Found. Comput. Math.*, vol. 10, no. 2, pp. 221–240, 2010.

[25] "CGAL, Computational Geometry Algorithms Library," http://www.cgal.org.

[26] D. Mount and S. Arya, "ANN: A library for approximate nearest neighbor searching."

[27] G. Stewart and J. Sun, *Matrix perturbation theory*. Computer Science and Scientific Computing, 1990.

[28] Q. Mérigot, M. Ovsjanikov, and L. Guibas, "Robust Voronoi-based curvature and feature estimation," in *Proc. ACM Symp. Solid Phys. Modeling*, 2009, pp. 1–12.

[29] T. Dey, J. Giesen, S. Goswami, and W. Zhao, "Shape dimension and approximation from samples," *Discrete and Computational Geometry*, vol. 29, no. 3, pp. 419–434, 2003.

[30] T. K. Dey, J. Giesen, and S. Goswami, "Shape segmentation and matching from noisy point clouds," in *Proc. Eurographics Symp. Point-based Graph.*, 2004, pp. 193–199.

[31] A. Shamir, "A survey on mesh segmentation techniques," *Comput. Graph. Forum*, vol. 27, no. 6, pp. 1539–1556, 2008.

[32] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," *IEEE Int. Conf. on Comp. Vis.*, 2009.

[33] N. Gelfand and L. J. Guibas, "Shape segmentation using local slippage analysis," in *Proc. Eurographics Symp. Geom. Proc.*, 2004, pp. 214–223.

[34] F. Chazal, L. Guibas, S. Y. Oudot, and P. Skraba, "Analysis of scalar fields over point cloud data," in *Proc. ACM/SIAM Symp. Disc. Algo.*, 2009, pp. 1021–1030.

[35] F. Chazal, D. Cohen-Steiner, and Q. Mérigot, "Geometric Inference for Measures based on Distance Functions," INRIA, Research Report RR-6930, 2009.

Quentin Mérigot received a Master degree in Mathematics from Université Grenoble I and École normale supérieure de Lyon in 2006, and the PhD degree in Computer Science from Université de Nice Sophia-Antipolis in 2009. He is now a researcher at CNRS and Université de Grenoble I. His research interests are related to approximation and discretization questions in geometry.

Maks Ovsjanikov is a PhD student at Stanford University in the Geometric Computing lab headed by professor Leonidas Guibas. His research interests include acquisition, manipulation and comparison of non-rigid 3d shapes. He is particularly interested in understanding structure-revealing relations between shapes.

Leonidas Guibas obtained his Ph.D. from Stanford in 1976, under the supervision of Donald Knuth. His main subsequent employers were Xerox PARC, MIT, and DEC/SRC. He has been at Stanford since 1984 as Professor of Computer Science. His interests span computational geometry, geometric modeling, computer graphics, computer vision, sensor networks, robotics, and discrete algorithms. More details about his research can be found on http://geometry.stanford.edu/member/guibas/.