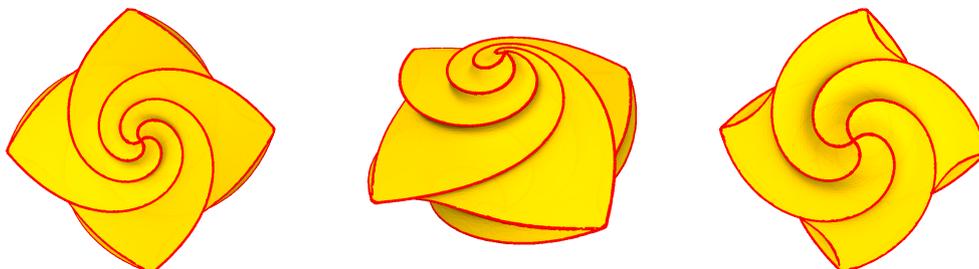# Robust Voronoi-based Curvature and Feature Estimation

Quentin Mérigot
INRIA Sophia-Antipolis
quentin.merigot@sophia.inria.fr

Maks Ovsjanikov
Stanford University
maks@stanford.edu

Leonidas Guibas
Stanford University
guibas@cs.stanford.edu

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

ABSTRACT

Many algorithms for shape analysis and shape processing rely on accurate estimates of differential information such as normals and curvature. In most settings, however, care must be taken around non-smooth areas of the shape where these quantities are not easily defined. This problem is particularly prominent with point-cloud data, which are discontinuous everywhere. In this paper we present an efficient and robust method for extracting principal curvatures, sharp features and normal directions of a piecewise smooth surface from its point cloud sampling, with theoretical guarantees. 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 analytically that our method recovers correct principal curvatures and principal curvature directions in smooth parts of the shape, and correct feature directions and feature angles at the sharp edges of a piecewise smooth surface, with the error bounded by the Hausdorff distance between the point cloud and the underlying surface. Using the same analysis we provide theoretical guarantees for a modification of a previously proposed normal estimation technique. 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.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and 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 [22], global matching [1], modeling and rendering of point-based surfaces [11], and anisotropic smoothing [15] to name just a few. 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 [14], non photorealistic rendering [21], and industrial metrology.

In practice, one is often interested in recovering curvature information of the underlying surface from point cloud data. The input in this setting is an unstructured collection of point coordinates, obtained by a range scanner. 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 a robust curvature estimation method, which can handle both non-uniform noise, sampling bias and sharp edges is a challenging task. This is mainly because curvature is a second order differential quantity which is by nature 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 on a unified framework that would incorporate both point clouds and piecewise-smooth surfaces.

In this article, we address some of these challenges by presenting a method for robustly estimating principal curvatures and curvature directions, as well as sharp edges of the underlying surface from point cloud data. We provide theoretical guarantees on the robustness of the results, by deriving a bound on the quality of the estimation based on

the Hausdorff distance between the point cloud and the underlying surface. We also address a certain class of outliers.

Prior work on curvature and feature estimation

Curvature estimation. Estimating curvature information of the underlying surface from a discrete approximation has been studied extensively over the past several decades (see e.g. [18] for a survey dating 2002 and [17] for an extensive comparison of methods to estimate Gaussian and mean curvatures). Nearly all existing methods for reliably estimating principal curvatures and curvature directions, however, rely on meshes. These methods are difficult to extend to the point cloud setting, because the mesh defines a discrete parametrization of the surface making angle and area computations easier.

Only recently several methods have been proposed for computing curvature information directly on point clouds. Yang and Qian [29] derive analytic expressions for computing principal curvatures based on the implicit definition of the Moving Least Squares (MLS) surface by Amenta and Kil [4]. In [26] the authors propose a sampling framework that allows sampling curves on the surface through a given point. This helps to reliably estimate normal curvatures in different directions, which are in turn used to determine the principal curvatures at the point. Lange and Polthier [15] adapt a framework by Taubin [27] to the point cloud setting by incorporating a discrete directional density measure to the classical discrete shape operator. This requires estimating the density of samples in different directions, which is challenging in the presence of noise, sampling bias and regions where separate parts of the shape come in close contact. Huang and Menq [12] propose a least-squares scheme based on estimating normal curvatures in different directions and using Euler's theorem to approximate principal curvatures. Tong and Tang [28] similarly estimate the principal curvatures by first estimating the normals and using them to compute directional curvatures. They achieve robustness to outliers by using a 3-step tensor voting procedure and correcting the positions of misaligned points. All of the above algorithms start by estimating normals to the surface, and any error made in this estimation is aggravated in the computation of principal curvatures. This is especially true around sharp edges of the surface and in the presence of noise. Our approach computes the normal direction and the principal curvatures simultaneously and handles both sharp edges and noise gracefully, by following the feature directions when they are present and being provably robust.

The most commonly used methods for computing curvatures on point clouds in practice, rely on polynomial fitting (see *e.g.* Cazals and Pouget [6] and references therein). By nature, these methods are very sensitive to the sampling conditions, requiring a rather uniform and noise-free sampling of the underlying surface whereas common real-world point clouds — like laser scans — often exhibit strong sampling bias and noise, *e.g.* clusters of points along horizontal lines. Resampling and denoising a point cloud while not oversmoothing is not an easy task. Our approach, on the other hand, makes no assumptions on the nature of the sampling of the surface, as long as the Hausdorff distance to the point cloud is small.

Feature Estimation. Although extracting sharp edges and corners is closely related to curvature estimation, research in these areas has been relatively independent. Fleishman et

al. [10] 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. [13] to extract feature-curves, by locally projecting onto feature points, and growing smooth polylines through the projected cloud. Lipman and colleagues [16] extract sharp edges within the Moving Least Squares (MLS) projection framework by defining a Singularity Indicator Field (SIF) based on the error of the MLS surface approximation, and performing a one dimensional projection of SIF to approximate feature curves. Jenke et al. [14] 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. [20] define a feature-preserving MLS projection operator by noting that normal vectors, rather than point coordinates can be used to discard points from unrelated parts of a piecewise-smooth surface.

Our method is based on similar intuition as [14] and [20] that features can be easily detected *in the offset* of the surface, but differs from all the previously proposed techniques in that we provide theoretical guarantees on the quality of the feature approximation. In addition, our method allows to recover not only the positions of feature points, but also the directions of sharp edges.

Voronoi-Based Normal Estimation. Our work is also related to Voronoi-based normal estimation techniques, pioneered by Amenta and Bern [3]. Their algorithm picks two Voronoi vertices called *poles* per sample point, which are then used to estimate the normal direction. Dey and Sun [9] extend this method to handle noisy data. As remarked in [2], the motivation behind Voronoi-based methods is that the shapes of the Voronoi cells of a point cloud provide meaningful information about the normal directions of the underlying surface. Alliez and colleagues [2] extend this intuition, by noting that noise can cause the Voronoi cells to become irregular. They rectify this effect by aggregating nearby cells, and computing the covariance matrix of a union of Voronoi cells.

We formalize this intuition by proving that a modification of the algorithm in [2] is guaranteed to return not only the correct normal directions, but also principal curvatures and principal curvature directions. In addition, we note that this method can be extended to points on sharp features, such as edges and corners of a piecewise-smooth surface.

Theoretical guarantees. We measure the robustness of our curvature estimation method by the difference between the estimated principal curvatures and principal directions of the underlying surface. Despite the multitude of methods for computing curvatures in the discrete setting, very few theoretical guarantees are known. In [8] Cohen-Steiner and Morvan prove the robustness of the normal cycle from geometric measure theory and show that it can be used to compute curvature information on Delaunay meshes. Pottmann et al. [24] introduce an integral-based curvature estimator on meshes, which uses principal component analysis of the intersection of balls with the interior of the domain D. The robustness here follows from the fact that integral quantities computed for two domains D and D' can remain close even though the boundary of these domains are far in the $\mathcal{C}^1$-sense.

Contributions

The main contribution of this article is a framework for estimating curvature and feature information of the underlying surface from a *point cloud C*, based on integral quantities. We modify the method of Alliez and colleagues [2], to compute covariance matrices of the intersection of Voronoi cells with the offset C^R of the point cloud, rather than its bounding box, as done in [2]. Intersecting with an offset allows us to obtain *local* information about the variation in shape and size of the Voronoi cell, which is crucial for curvature estimation. We present two algorithms for computing covariance matrices of this intersection: a Monte Carlo method, and a method based on tessellating it with tetrahedra.

The theoretical results are twofold: first, for any compact set K, we define its Voronoi Covariance Measure (VCM), through the *projection function* on K. This allows us to study the discrete case of point clouds and the underlying continuous piecewise-smooth surfaces in a single framework. With this notion at hand, we prove that if K is a piecewise-smooth submanifold of $\mathbb{R}^d$, then the eigenvalues and eigenvectors of its convolved VCM provide information on the normal directions, principal curvatures and directions, directions of sharp features, and dihedral angles between its smooth parts. In the second part, we prove that if the point cloud C is a good Hausdorff approximation of K, then the convolved VCM of C and K are also uniformly close.

Outline

The rest of the paper is organized as follows: in Section 2 we introduce the Voronoi Covariance Measure (VCM) and the convolved VCM for an arbitrary compact set K. We give an expression for the convolved VCM of a point cloud in Section 2.2. In Sections 2.3 and 2.4 we show that the convolved VCM of a piecewise smooth surface provides information about the normal and principal curvatures in the smooth parts and the feature direction around sharp edges. In Section 3 we show that the Earthmover distance between the convolved VCM for two compact sets can be bounded by a multiple of their Hausdorff distance, implying the convergence of the convolved VCM of the point cloud to that of the underlying surface. We also note that the convolved VCM of a point cloud is not influenced by a certain class of outliers. In Section 4 we describe two algorithms for computing the convolved VCM of a point cloud in practice, a Monte Carlo method, and a method based on tessellating the offset with tetrahedra. Finally, Section 5 is dedicated to the results obtained with our method on a variety of surfaces.

2. 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 and derive its properties for a point cloud and for a piecewise smooth surface. The relation between these two cases is given in Section 3 in the form of a stability theorem, which implies in particular that principal curvatures and curvature directions of the underlying piecewise-smooth surface can be estimated from its point cloud sampling.

2.1 Mathematical background

Projection on a compact set and Voronoi cells. If $K \subseteq \mathbb{R}^d$ is a compact set, we will denote by d_K the distance function to K, *i.e.* for any $x \in \mathbb{R}^d$, $\mathrm{d}_K(x) = \min\{\|x - y\| \, ; y \in$
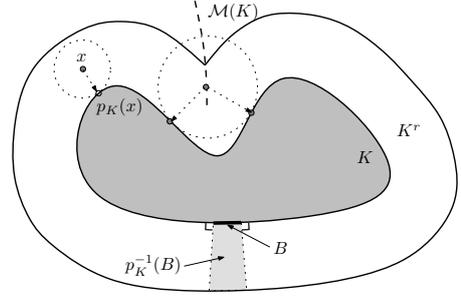

Figure 2.1: Projection function and medial axis of a compact set K with smooth boundary.

$K\}$. The nearest neighbors of x in K are the points y which realize this minimum: the set of points for which there exist more than one nearest neighbor in K is called the medial axis $\mathcal{M}(K)$ of K.

The projection function maps a point $x \in \mathbb{R}^d \setminus \mathcal{M}(K)$ to its only closest point $\mathrm{p}_K(x)$ in $K \subseteq \mathbb{R}^d$. It is well known that the medial axis of any compact set has zero d-volume, implying that the projection function is well defined almost everywhere. Since we are only interested in integral properties, the value of p_K on $\mathcal{M}(K)$ can be arbitrary, and we will consider p_K as being defined on the whole space $\mathbb{R}^d$.

If $K = \{p_1, \ldots, p_n\}$ is a point cloud, let $\mathrm{Vor}(p_i)$ denote the Voronoi cell of p_i : $\mathrm{Vor}(p_i) = \{x : \mathrm{d}(x, K) = \|x - p_i\|\}$ *i.e.* the set of points of $\mathbb{R}^d$ that are closer to p_i than to any other point of K. The projection on K maps any $x \in \mathbb{R}^d$ to p_i such that $x \in \mathrm{Vor}(p_i)$, except on the boundaries of the Voronoi cells – which constitute the medial axis of K. By analogy, we will refer to the set $\mathrm{p}_K^{-1}(B)$ as the Voronoi cell of the set $B \subseteq K$, and to $\mathrm{p}_K^{-1}(\{x\})$ as the infinitesimal Voronoi cell of the point $x \in K$, for any compact K.

R-offset and the Reach of a compact set $K \subseteq \mathbb{R}^d$. For any compact set K, and a positive scalar R, the R-offset of K is defined as $K^R = \{x \in \mathbb{R}^d : \mathrm{d}(x, K) < R\}$ – the set of points no more than R away from K.

If $K \subseteq \mathbb{R}^d$ the *reach* of K is defined as the maximum R such that for all $x \in K^R$, $\mathrm{p}_K(x)$ is well-defined. In other words, $\mathrm{reach}(K) = d(K, \mathcal{M}(K))$.

Symmetric Hausdorff distance. The symmetric Hausdorff distance $\mathrm{d}_H(K, K')$ between two compact sets K and K' is defined as the smallest number R such that $K' \subseteq K^R$ and $K \subseteq K'^R$.

Covariance matrix. If $E \subseteq \mathbb{R}^d$ has finite volume, its covariance matrix is a 2-tensor whose eigenvectors capture the principal axes of E with respect to a base point p.

$$\mathrm{cov}(E, p) = \int_E (x - p)(x - p)^{\mathbf{T}} \mathrm{d}x$$

2.2 Voronoi Covariance Measure

General definition. The Voronoi Covariance Measure (or VCM) is defined for *any* compact set K of $\mathbb{R}^d$. This can be a finite point cloud, a (piecewise) smooth manifold, or an even wilder object. We also need a scale parameter R, which will be used to define an offset K^R of K.

The Voronoi Covariance Measure of K with respect to K^R is a tensor-valued measure denoted by $\mathcal{V}_{K,R}$. Being tensor-valued means that unlike a usual measure μ in the sense of
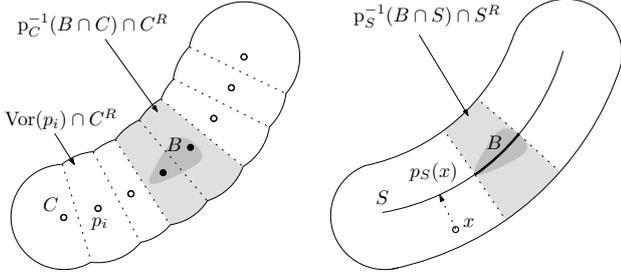
Figure 2.2: Voronoi Covariance Measures $\mathcal{V}_{C,R}(B)$ of a 2D point cloud C and of the underlying curve S, with respect to the same probing set B.

Lebesgue, which maps each subset B of $\mathbb{R}^d$ to a non-negative number $\mu(B)$, the Voronoi Covariance Measure maps every such $B \subseteq \mathbb{R}^d$ to a non-negative definite covariance matrix $\mathcal{V}_{K,R}(B)$. This covariance matrix is defined as follows:

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

Figure 2.2 illustrates the domain of integration in the definition of $\mathcal{V}_{K,R}(B)$ when K is a point cloud and when it is a curve. Intuitively, this definition corresponds to the covariance matrix of the set $\mathrm{p}_K^{-1}(B \cap K) \cap K^R$, but with a *varying base point*: one can think of it as the integral over all $p \in B \cap K$ of the covariance matrices of the infinitesimal Voronoi cell $\mathrm{p}_K^{-1}(\{p\}) \cap E$, with base point p.

The set of vectors $\mathcal{N}_{p,R}K = \{x - p; \ \mathrm{p}_K(x) = p \text{ and } x \in K^R\}$ corresponds in to the *normal cone* of K at p, at a scale defined by R. Indeed, if K is smooth and $R < \mathrm{reach}(K)$ then $\mathcal{N}_{p,R}K$ is the segment $[-R\vec{\mathrm{n}}(p), R\vec{\mathrm{n}}(p)]$ (Figure 2.2). If, on the other hand, $K \subseteq \mathbb{R}^3$ is a convex polyhedron, $\mathcal{N}_{p,R}K$ corresponds to the usual normal cone at p, intersected with K^R; it is a 2-dimensional subset of $\mathbb{R}^3$ when p lies on a edge and 3-dimensional when p is a vertex (*cf.* Figure 2.3).

If B is a small neighborhood of a point $p \in K$, the covariance matrix $\mathcal{V}_{K,R}(B)$ is the sum of all the infinitesimal covariance matrices of the normal cones $\mathcal{N}_{p,R}K$, with $p \in B$. Therefore, $\mathcal{V}_{K,R}(B)$ captures the variation of the normal cone of K around p, which is related to the curvature at p, when K is locally smooth.

In the following, we formalize this intuition by computing the Voronoi Covariance Measure for point clouds, smooth hypersurfaces and sharp edges of a piecewise smooth surface.

Offset parameter. Note that unlike [2], where the authors intersected the Voronoi cells with the bounding box, we integrate over an intersection with the R-offset K^R of K. Intersecting with the bounding box introduces *global* effects in the cells and can cause bias in different directions. This effect may be acceptable for estimating normal directions but can hurt the results significantly for feature detection as well as for curvature estimation. In our experiments we remarked that choosing an excessively large offset parameter prevented the detection of small features (such as the braiding in Figure 5.4). The same phenomena appear when the integration domain is a bounding box.

Convolved VCM. Convolving the tensor-valued measure $\mathcal{V}_{K,R}$ by a continuous and integrable function $\chi : \mathbb{R}^d \to \mathbb{R}$ turns it into a (tensor-valued) *density function* $\mathcal{V}_{K,R} * \chi :$

$\mathbb{R}^d \to \mathrm{Sym}(\mathbb{R}^d)$, defined by:

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

The indicator function of the ball $\mathrm{B}(0, r)$ is the function χ_r defined by $\chi_r(p) = 1$ if $x \in \mathrm{B}(0, r)$ and $\chi_r(p) = 0$ otherwise. Note that in this case one has:

$$\mathcal{V}_{K,R} * \chi_r(p) = \mathcal{V}_{K,R}(\mathrm{B}(p, r)) \tag{2.1}$$

In this work, the convolution kernel is always chosen to be a Lipschitz approximation of such an indicator function, like the "hat function" $\chi(p) = \max(0, r - \|p\|^2)$, for which we can prove that $\mathcal{V}_{K',R} * \chi$ converges to $\mathcal{V}_{K,R} * \chi$ as K' converges to K (see Section 3). 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.

VCM of a point cloud. Applying the definition of VCM to a point cloud $C = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d$ we obtain:

$$\mathcal{V}_{C,R}(\{p_i\}) = \int_{C^R \cap p_C^{-1}(p_i)} (x - p_i)(x - p_i)^{\mathbf{T}} \mathrm{d}x$$

Since the inverse image $p_C^{-1}(p_i)$ coincides with the Voronoi cell $\mathrm{Vor}(p_i)$, we get $\mathcal{V}_{C,R}(\{p_i\}) = \mathrm{cov}\,(\mathrm{Vor}(p_i) \cap \mathrm{B}(p_i, R), p_i)$. Using the additivity of the integral, we can describe $\mathcal{V}_{C,R}(B)$ for an arbitrary set $B \subseteq \mathbb{R}^d$ as $\mathcal{V}_{C,R}(B) = \sum_{p_i \in B \cap C} \mathcal{V}_{C,R}(\{p_i\})$ (*cf.* Figure 2.2), or:

$$\mathcal{V}_{C,R}(B) = \sum_{p_i \in B \cap C} \mathrm{cov}\,(\mathrm{Vor}(p_i) \cap \mathrm{B}(p_i, R), p_i)\,.$$

Convolved VCM of a point cloud. Using equation 2.1, and choosing the convolution kernel χ to be the indicator function of a ball with radius r, we obtain the following expression:

$$\mathcal{V}_{C,R} * \chi_r(p) = \sum_{p_i \in \mathrm{B}(p,r) \cap C} \mathrm{cov}\,(\mathrm{Vor}(p_i) \cap \mathrm{B}(p_i, R), p_i)$$

Intuitively, convolving the VCM by a function smoothes the information contained in the covariance matrices. This is particularly useful when dealing with point clouds that can be noisy or that are anisotropically sampled from an underlying smooth surface. Note that the effect of the convolution is not easily predicted, since, for example, the principal directions (eigenvectors) of the sum of two covariance matrices do not equal, in general, to the sum of the eigenvectors of the covariance matrices. Nevertheless, one of the main results of this this paper is Theorem 3.2, which establishes that the difference between *convolved* VCM of two compact sets (e.g. a point cloud and an smooth surface) can be bounded as a function of their Hausdorff distance.

We describe two practical algorithms to compute the convolved VCM of a point cloud in section 4.

2.3 VCM of a smooth hypersurface

VCM of a smooth hypersurface. Let S be a compact smooth hypersurface embedded in $\mathbb{R}^d$, $\vec{\mathrm{n}}$ an oriented normal vector field on S and R_S the reach of S. As before, S^R denotes the R-offset of S.

Let us recall a few facts about the curvature of embedded hypersurfaces. The map $x \in S \mapsto \vec{\mathrm{n}}(x)$ is called the *Gauss map* of S, while its derivative $\mathrm{d}\vec{\mathrm{n}}$ is called the *shape operator* of S. If $\kappa_1(x), \ldots, \kappa_{d-1}(x)$ denote the $(d-1)$ principal curvatures at x and $P_1(x), \ldots, P_{d-1}(x)$ is a set of vectors

in the tangent plane spanning the principal curvature directions, with $\|P_i(x)\| = 1$, one has: $\forall v \in T_x S$, $d\vec{n}(x)(v) = \sum_{i=1}^{d-1} \kappa_i(x)\langle P_i(x)|v\rangle P_i(x)$.

In the following, we show that the Voronoi Covariance Measure $\mathcal{V}_{S,R}$ evaluated on a subset $B \subseteq S$ can be described in terms of the covariance matrix of the normal vectors $\vec{n}(x)$, $x \in B$:

THEOREM 2.1. *If $R < R_S$, the reach of S, then for every $B \subseteq S$ one has*

$$\mathcal{V}_{S,R}(B) = \frac{2}{3}R^3\left[1 + O(R^2/R_S^2)\right]\int_{p \in B \cap S} \vec{n}(p)\vec{n}(p)^{\mathbf{T}}\,dp$$

If S is a curve in the 2-dimensional plane, the error term vanishes; if S is a hypersurface in 3-dimensions, it is bounded more explicitly by $\frac{3}{5}\frac{R^2}{R_S^2}$.

Proof. If R is smaller than the reach of S, the map $\varphi : S \times [-R, R] \to S^R, (p, t) \mapsto p + t\vec{n}(p)$, where S^R is the R-neighborhood of S, is a diffeomorphism. Thus, the change-of-variable formula yields:

$$\mathcal{V}_{S,R}(B) = \int_{S^R \cap p_S^{-1}(B \cap S)} (x - p_S(x))(x - p_S(x))^{\mathbf{T}}dx$$

$$= \int_{p \in B \cap S}\int_{-R}^{R} \vec{n}(p)\vec{n}(p)^{\mathbf{T}}\, t^2\, \mathrm{J}\varphi(p, t)\, dt\, dp$$

where $|\mathrm{J}\varphi(p, t)|$ is the Jacobian determinant of φ at $(p, t) \in S \times [-R, R]$. In the local frame of the tangent space given by the $d - 1$ principal curvature directions $P_1(p), \dots, P_{d-1}(p)$, the derivative of φ is the diagonal matrix with components $1 + t\kappa_1(p), \dots, 1 + t\kappa_{d-1}(p), 1$. From this, we are able to compute the Jacobian determinant:

$$|\mathrm{J}\varphi(p, t)| = \prod_{i=1}^{d-1}(1 + t\kappa_i(p)) = \sum_{i=0}^{d-1} t^i \sigma_i(p)$$

In this expression, $\sigma_i(p) = \sum_{j_1 < \dots < j_i} \kappa_{j_1}(p) \dots \kappa_{j_i}(p)$ is the ith symmetric polynomial of the principal curvatures at p. Since all terms with odd powers of t vanish when we integrate from $-R$ to R, we get the following polynomial expansion for the VCM:

$$\mathcal{V}_{S,R}(B) = \sum_{k=0}^{\lfloor(d-1)/2\rfloor} \frac{2R^{2k+3}}{2k+3}\int_{p \in B \cap S} \sigma_{2k}(p)\vec{n}(p)\vec{n}(p)^{\mathbf{T}}\, dp$$

This implies that for curves in 2-dimensional Euclidean space:

$$\mathcal{V}_{S,R}(B) = \frac{2R^3}{3}\int_{p \in B \cap S} \vec{n}(p)\vec{n}(p)^{\mathbf{T}}\, dp.$$

In general dimension, the principal curvature radii $|\kappa_i(p)|^{-1}$ are greater than $R_S = \mathrm{reach}(S)$ at any point $p \in S$. Using this to evaluate the error made by truncating this polynomial to its first term only yields the desired formula. $\qquad\square$

Note that although we use one offset parameter, since all of the operations are local, Theorem 2.1 would also hold if the offset parameter was chosen locally to be less than local feature size. Thus, if parts of the surface were known to be more flat, a higher offset could be chosen there.

Convolved VCM at a smooth point. Let us recall that if χ is the indicator function of the ball of radius r centered at the origin, then by equation 2.1 $\mathcal{V}_{S,R} * \chi(p) = \mathcal{V}_{S,R}(\mathrm{B}(p, r))$. The goal of the remainder of this section is to show that the biggest eigendirection of the matrix $\mathcal{V}_{S,R}(\mathrm{B}(p, r))$ is close to the normal direction at p, and the $(d-1)$ smallest eigendirections are close to the $(d-1)$ principal curvature directions.

Convolved VCM of a point on a curve in 2D. We first compute the convolved VCM of a smooth point on a curve S in 2-dimensional Euclidean space, when the convolution kernel is an indicator function of a ball. From the previous paragraph we have, for any $B \subseteq \mathbb{R}^2$:

$$\mathcal{V}_{S,R}(B) = \frac{2R^3}{3}\int_{p \in B \cap S} \vec{n}(p)\vec{n}(p)^{\mathbf{T}}\, dp.$$

By equation 2.1, to compute the convolved VCM of a point p_0 on the curve, we need to compute $\mathcal{V}_{S,R}(\mathrm{B}(p_0, r))$, where r is the kernel radius. Let $\tilde{B}(0, r)$ be the projection of $B(p, r)$ on the tangent line at $p_0 : \mathrm{T}_{p_0}S = s\vec{t}(p_0)$, where $s \in \mathbb{R}$ and $\vec{t}(p_0)$ is the unit vector in the tangent direction. Then for small r, we have:

$$\mathcal{V}_{S,R}(B(p_0, r))$$

$$\simeq \frac{2R^3}{3}\int_{\tilde{B}(0,r)} (\vec{n}(p_0) + d\vec{n}(p_0)(p))(\vec{n}(p_0) + d\vec{n}(p_0)(p))^{\mathbf{T}}dp$$

$$= \frac{2R^3}{3}\int_{-r}^{r} \left(\vec{n}(p_0) + s\kappa(p_0)\vec{t}(p_0)\right)\left(\vec{n}(p_0) + s\kappa(p_0)\vec{t}(p_0)\right)^{\mathbf{T}} ds$$

$$= \frac{4R^3r}{3}\left(\vec{n}(p_0)\vec{n}(p_0)^{\mathbf{T}} + \kappa^2(p_0)\frac{r^2}{3}\vec{t}(p_0)\vec{t}(p_0)^{\mathbf{T}}\right)$$

Since the normal and tangent vectors are orthogonal, this implies that $\vec{n}(p_0)$ and $\vec{t}(p_0)$ are the eigenvectors of $\mathcal{V}_{S,R}(\mathrm{B}(p_0, r))$ with the corresponding eigenvalues $\lambda_0 = \frac{4}{3}R^3r$ and $\lambda_1 = \frac{4}{9}R^3r^3\kappa^2(p_0)$. In particular, $\frac{\lambda_1}{\lambda_0} = \frac{r^2}{3}\kappa^2(p_0)$.

Convolved VCM of a point on a surface in 3D. A nearly identical calculation as above can be performed for a hypersurface in 3D. Now for a point p_0 on the surface and a ball $\mathrm{B}(p_0, r)$, we denote by $\tilde{B}(0, r)$ the projection of this ball on the tangent plane $\mathrm{T}_{p_0}S = \{\vec{n}(p_0)\}^{\perp}$. Then,

$$\mathcal{V}_{S,R}(B(p_0, r))$$

$$\simeq \frac{2}{3}R^3\int_{v \in \tilde{B}(0,r)} (\vec{n}(p_0) + d\vec{n}(p_0)(v))(\vec{n}(p_0) + d\vec{n}(p_0)(v))^{\mathbf{T}}\, dv$$

For a hypersurface in $\mathbb{R}^3$, $d\vec{n}(p_0)(v) = \kappa_1(p_0)\langle P_1(x)|v\rangle P_1 + \kappa_2(p_0)\langle P_2(x)|v\rangle P_2$, where $\kappa_i(p_0)$ and $P_i(p_0)$ are the principal curvatures and principal curvature directions at p_0 respectively. In polar coordinates: $d\vec{n}(p_0)(r, \theta) = \kappa_1(p_0)r\cos(\theta)P_1 + \kappa_2(p_0)r\sin(\theta)P_2$. Since the integration above is over a circle, all terms with $\sin(\theta)$ or $\cos(\theta)$ will vanish. Therefore, in polar coordinates:

$$\mathcal{V}_{S,R}(\mathrm{B}(p_0, r))$$

$$\simeq \frac{2}{3}R^3\int_{s=0}^{r}\int_{\theta=0}^{2\pi} \left[n(p_0)n(p_0)^{\mathbf{T}} + s^2\cos^2(\theta)\kappa_1^2(p_0)P_1P_1^{\mathbf{T}}\right.$$
$$\left. + s^2\sin^2(\theta)\kappa_2^2(p_0)P_2P_2^{\mathbf{T}}\right]\, s\, d\theta ds$$

$$= \frac{2\pi}{3}R^3r^2\left[\vec{n}(p_0)\vec{n}(p_0)^{\mathbf{T}} + \frac{r^2}{4}\sum_{i=1}^{2}\kappa_i^2(p_0)P_i(p_0)P_i(p_0)^{\mathbf{T}}\right]$$

It follows from this formula that under the assumptions on the offset radius R and convolution radius r, the eigenvectors of the VCM near a smooth point are close to the principal curvature directions at that point.

A similar formula with different constants can be derived for higher dimensional spaces as well.
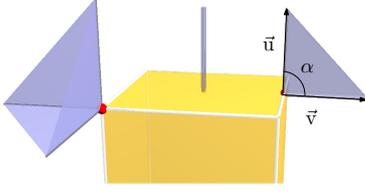
Figure 2.3: The infinitesimal Voronoi cell $p_C^{-1}(x)$ of a point x on a cube is pencil, triangle or cone-shaped depending on the dimension of the normal cone.

2.4 VCM at a point on a sharp feature

Let S be a piecewise smooth hyper-surface in R^3, and let p_0 be a point on an edge between two smooth parts of the surface. Intuitively, the the infinitesimal Voronoi cell of p_0 is triangle shaped (Figure 2.3)), and the triangle is orthogonal to the feature direction. In the Appendix we formalize this intuition and derive the eigenvalues and eigenvectors of the covariance matrix $\mathcal{V}_{S,R}(\mathrm{B}(p_0, r))$.

We work under the assumption that $p_S(p_0 + tv) = p_0$ for all unit vectors v in the normal cone at p_0 and any $0 \leqslant t < R$. This assumption can be seen as a one-sided lower bound on the local feature size. Then, if $\vec{u}$ and $\vec{v}$ denote the projections of the two outward normals to S at p_0 on the orthogonal plane to the feature direction (*cf.* Figure 2.3),

$$\mathcal{V}_{S,R}(\mathrm{B}(p_0, r)) = \frac{R^4 r}{8}[(\alpha - \sin(\alpha))\, e_1 e_1^{\mathbf{T}} + (\alpha + \sin(\alpha))\, e_2 e_2^{\mathbf{T}} + O(r/R)]$$

with $\vec{e_1} = \dfrac{\vec{u} - \vec{v}}{\|\vec{u} - \vec{v}\|}$, $\vec{e_2} = \dfrac{\vec{u} + \vec{v}}{\|\vec{u} + \vec{v}\|}$, and $\cos(\alpha) = \langle \vec{u} | \vec{v} \rangle$

In particular, the eigendirection corresponding to the smallest eigenvalue is the tangent direction to the feature, with an eigengap of $O(R/r)$.

Similar calculations can be done if p_0 is an isotropic corner of the surface, such as the tip of a cone. In this case, the two smallest eigenvalues will be equal, and contained in the plane orthogonal to the direction of the cone. Figure 2.3 shows how the dimension of Voronoi cells, and hence the number of small eigenvalues s of their covariance matrix, can help distinguish between a point on a face ($s = 2$), on a sharp edge ($s = 1$) and on a corner ($s = 0$).

3. ROBUSTNESS OF VCM

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 we stress that this result is very general and does not depend on the nature of these compact sets.

The convergence property is given in terms of the (symmetric) *Hausdorff distance* between the compact sets K and K', denoted by $\mathrm{d}_H(K, K')$. A point cloud is Hausdorff-close to the underlying surface provided that it is sampled densely 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.

We will prove a continuity result for the map $K \mapsto \mathcal{V}_{K,R} * \chi$, from compact subsets of $\mathbb{R}^d$ to tensor-valued functions, where the distance between two convolved VCM $\mathcal{V}_{K,K^R} * \chi$ and $\mathcal{V}_{K',K'^R} * \chi$ is measured by

$$\|\mathcal{V}_{K,K^R} * \chi - \mathcal{V}_{K',K'^R} * \chi\|_\infty$$
$$:= \sup_{p \in \mathbb{R}^d} \left\|\mathcal{V}_{K,K^R} * \chi(p) - \mathcal{V}_{K',K'^R} * \chi(p)\right\|_{\mathrm{op}}$$

Where the operator norm $\|\cdot\|_{\mathrm{op}}$ is the standard matrix norm induced from the vector norm $\|\cdot\|$. The main theoretical result we use is the following Projection Stability Theorem from [7]:

THEOREM 3.1. *Let K and K' be two compact sets of $\mathbb{R}^n$, and E an open set with rectifiable boundary. Then there is a constant $C(d, K, E)$ such that:*

$$\int_E \|p_K(x) - p_{K'}(x)\| \,\mathrm{d}x \leqslant C(d, K, E)\mathrm{d}_H(K, K')^{1/2}$$

As a consequence of this theorem, we will prove the Hausdorff-robustness of the convolved VCMs when the convolution kernel $\chi : \mathbb{R}^d \to \mathbb{R}$ is k-Lipschitz and bounded by 1 — this means that there exists a $k > 0$ such that:

$$\forall(x, y),\ |\chi(x) - \chi(y)| \leqslant k\,\|x - y\| \text{ and } \forall x,\ |\chi(x)| \leqslant 1$$

THEOREM 3.2. *If $\chi : \mathbb{R}^d \to \mathbb{R}$ is a bounded k-Lipschitz function, for every compact set $K \subseteq \mathbb{R}^d$ and $R > 0$, there is a constant $C'(d, K, R)$ such that for every other $K' \subseteq \mathbb{R}^d$,*

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

Proof. We let E be the intersection of K^R and K'^R. By Corollary II.5 in [7], we know that the volume $K^R \setminus E$ is $O(\mathrm{d}_H(K, K'))$. Hence, letting $\delta(x) = x - p_K(x)$, we can make the approximation,

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

The goal is now 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 $P(x) = \delta(x)\delta(x)^{\mathbf{T}}\chi(p_K(x) - p)$. Then,

$$P(x) - P'(x)$$
$$= \chi(p_K(x) - p)\left(\delta(x)\delta(x)^{\mathbf{T}} - \delta'(x)\delta'(x)^{\mathbf{T}}\right)$$
$$+ (\chi(p_K(x) - p) - \chi(p_{K'}(x) - p))\,\delta'(x)\delta'(x)^{\mathbf{T}}$$

The first term can be written as:

$$\chi(p_K(x) - p)\left((p_K(x) - x)(p_K(x) - p_{K'}(x))^{\mathbf{T}}\right.$$
$$\left. + (p_{K'}(x) - p_K(x))(x - p_{K'}(x))^{\mathbf{T}}\right)$$

Now since χ is bounded by 1, $\|p_K(x) - x\| < R$, and using the triangle inequality, the operator norm of this expression is bounded by $2R\,\|p_{K'}(x) - p_K(x)\|$.

By the k-Lipschitz property of χ, the norm of the second term can be bounded by

$$|\chi(p_K(x) - p) - \chi(p_{K'}(x) - p)|R^2$$
$$\leqslant kR^2\,\|p_K(x) - p_{K'}(x)\|$$

Hence,

$$\left\|P(x) - P'(x)\right\|_{\mathrm{op}} \leqslant (2R + kR^2)\,\|p_K(x) - p_{K'}(x)\|$$

Integrating and using theorem 3.1 yields the bound

$$\|M\|_{\mathrm{op}} \leqslant C(d, K, E)[2R + kR^2]\mathrm{d}_H(K, K')^{1/2} \qquad \square$$

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 says that $\mathcal{V}_{C_n, R} * \chi_r(p)$ converges to $\mathcal{V}_{S, R} * \chi_r(p)$ w.r.t the operator norm, and quantifies the speed of convergence.

Using the standard results of matrix perturbation theory (see e.g. [25]), 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.

Since at a smooth point $p \in S$, the eigenvalues of $\mathcal{V}_{S, R} * \chi_r$ are proportional to 1, $\kappa_1^2(p) \frac{r^2}{4}$ and $\kappa_2^2(p) \frac{r^2}{4}$, one can 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).

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 3.3. *Given a point cloud C and a set of outliers O, with $\min_{o \in 0} \mathrm{d}(o, C) \geqslant 2R > r$, and χ is the indicator function of $\mathrm{B}(0, r)$ then the convolved Voronoi Covariance Measures of C and $C' = C \cup O$ agree on C:*

$$\forall\, p \in C, \ \mathcal{V}_{C, R} * \chi(p) = \mathcal{V}_{C', R} * \chi(p)$$

Proof. We first prove that for any point $p \in C$, the intersection between its Voronoi cell and a ball of radius R remains unchanged after introducing points from O. For this, simply note that the bisecting plane formed between p and any point $p' \in O$ is at least R away from p, because by assumption $\mathrm{d}(p, p') > 2R$. This implies that:

$$\forall\, p_i \in C, \ \mathcal{V}_{C, R}(\{p_i\}) = \mathcal{V}_{C', R}(\{p_i\})$$

Finally, since

$$\mathcal{V}_{C', R} * \chi(p) = \sum_{p_i \in \mathrm{B}(p, r)} \mathrm{cov}\left(\mathrm{Vor}(p_i) \cap C'^R, p_i\right)$$

and because $\|p - p'\| > r$ there will be no outliers in the ball $\mathrm{B}(p, r)$, and the desired equality follows. $\qquad\square$

4. COMPUTATION OF THE 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.

As remarked earlier, the VCM at a scale R of a point cloud $C = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^3$ is the tensor-valued measure $\mathcal{V}_{C, R}$ concentrated on the point cloud C and such that $\mathcal{V}_{C, R}(\{p_i\})$ is the covariance matrix of the intersection $B_i = \mathrm{B}(p_i, R) \cap \mathrm{Vor}(p_i)$, where $\mathrm{Vor}(p_i)$ is the Voronoi cell of p_i.
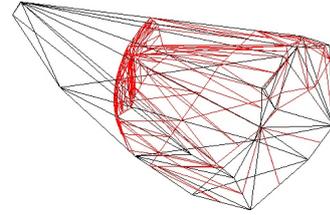

Figure 4.4: The tessellated intersection of a Voronoi cell with a ball.

No simple closed-form expression seems to exist for the covariance matrix of B_i. As remarked in [5], in order to compute the *volume* of the intersection of a Voronoi cell with a ball, one can use the inclusion-exclusion formula to reduce to the case of the intersection of two and three half-planes with the same ball. The same formula can be used to compute the covariance matrix of B_i, reducing it to integrals of the elementary quadratic polynomials $P_{i,j}(x_1, x_2, x_3) = x_i x_j$ over these two type of intersections. However, whereas the integral of the constant function $P(x_1, x_2, x_3) = 1$ admits a (pretty intricate) closed form, there seems to be no such form for the $P_{i,j}$.

Monte-Carlo approximation of the VCM. The Voronoi covariance measure is a modification of the *boundary measure* introduced in [7]. In this work, instead of considering the covariance matrix $\mathrm{cov}(B_i)$, one only considered its volume. The Monte-Carlo algorithm for computing the boundary measures can be easily adapted to compute the VCM:

Algorithm 4.1 Monte-Carlo algorithm for VCM

Input: a point cloud C, a scalar R, a number N
Output: an approximation of $\frac{\mathcal{V}_{C,R}}{\mathrm{vol}_d(C^R)} \simeq \frac{1}{N} \sum_{p_i \in C} C_i \delta_{p_i}$
loop {N times}
   [1.] Choose a random p_i uniformly in C ;
   [2.] Choose a random point X uniformly in $\mathrm{B}(p_i, R)$;
   [3.] Compute $k = \#(\mathrm{B}(X, R) \cap C)$;
   [4.] Find the closest point p_j of X in C:
      $C_j \leftarrow C_j + \frac{1}{k}(X - p_j)(X - p_j)^{\mathbf{T}}$

The convergence of the output of this algorithm to the real VCM with high probability follows from the arguments presented in [7, p.6], as well as the analysis of the speed of convergence. This algorithm has the advantage of being easy to implement in any ambient dimension, since it only uses nearest-neighbor queries. 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.

Approximating the VCM in 3D. We base our second method on the fact that the covariance matrix of a tetrahedron can be computed analytically [2]. 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 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 $\partial(\text{Vor}(p_i))$. Our goal is subdivide each triangle so that its projection onto the ball $\text{B}(p_i, R)$ is a sufficiently good approximation of the corresponding spherical triangle. For this, we process each triangle t in the original triangulation according to the following three rules:

1. t *is completely outside the ball* $\text{B}(p_i, R)$*:*
   Recursively subdivide t into a family of smaller triangles $\{t_k\}$, $t_k = (a_k, b_k, c_k)$, until the family of projected triangles $t'_k = (\text{p}(a_k), \text{p}(b_k), \text{p}(c_k))$ is a precise enough tessellation of the underlying spherical patch. Then add each t'_k to the final triangulation $\mathcal{T}'$.

2. t *is completely inside the ball* $\text{B}(p_i, R)$*:*
   Simply add t to the final triangulation $\mathcal{T}'$.

3. t *crosses the sphere* $\partial\text{B}(p_i, R)$*:*
   Subdivide t by adding points along the circular arc of intersection, and apply **1.** or **2.** to the constructed triangles depending on whether they are inside or outside of the ball.

The output of this algorithm yields a tetrahedralization 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 [2] to compute the covariance matrix of this intersection.

Convolution of VCM. Convolving the VCM of a point cloud using a kernel function χ supported on a ball of radius r can be done by computing for each point $p_i \in C$ the intersection $\text{B}(p_i, r) \cap C$, and then summing:

$$\mathcal{V}_{C,R} * \chi(p_i) = \sum_{p_j \in \text{B}(p_i, r) \cap C} \chi(p_j - p_i) \mathcal{V}_{C,R}(\{p_j\})$$

The points belonging to $\text{B}(p_i, r) \cap C$ are the k nearest neighbor to p_i in C for the suitable value of k, which can be determined by a binary search, using a structure adapted to k-NN queries (like a kD-tree). As mentioned in Section 2.2, in this work, we always used $\chi = \mathbf{1}_{B(0,r)}$.

Implementation. We implemented the two algorithms described above. The tessellation of the Voronoi cells was done using the 3D Delaunay Triangulation package of CGAL [23]. Running this algorithm for the Voronoi cell at the origin and 15 random points on the unit sphere, with $R = 1$, yields an average of 120 triangles for tessellating the boundary of $\text{Vor}(p_i) \cap \text{B}(p_i, R)$, for a target approximation error of 1%. The running times of this algorithm on more complex point clouds are reported in Table 1.

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

Table 1: Computation times for VCM of range-scan models (in sec., on a 3GHz Dual Core CPU)

The convolution step is implemented using the ANN library [19], which includes a query for finding the set of points of a point cloud contained in a given ball. The time of the

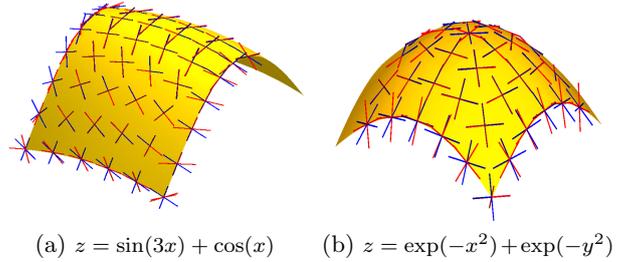(a) $z = \sin(3x) + \cos(x)$ (b) $z = \exp(-x^2) + \exp(-y^2)$

Figure 5.5: Parametric surfaces with exact (in blue) and computed (in red) principal curvature directions. At the boundary, the computed directions follow the edges of the surface.

convolution step depends on the radius of convolution, but stays within 10 seconds for most models.

Note also that both the tessellation and the convolution steps can be easily parallelized once the Delaunay triangulation and kD-tree have been constructed, since the computations for a given point of the cloud do not involve access to any shared data structures.

5. RESULTS

5.1 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 $z = \sin(3x) + \cos(y)$, where $0 \le x, y \le 1$ and $z = \exp(-x^2) + \exp(-y^2)$, where $-\frac{1}{2} \le x, y \le \frac{1}{2}$. In both examples we used 100,000 points that were chosen uniformly at random within the domain. Figure 5.5 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.

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. Figure 5.6 shows the average deviation for these two parametric surfaces. As can be seen, the results are quite stable for different choices of the parameters. Interestingly, although our analysis above applies only if the offset is smaller than the reach, we get good results even when the former is larger.

5.2 Comparison with Polynomial Fitting

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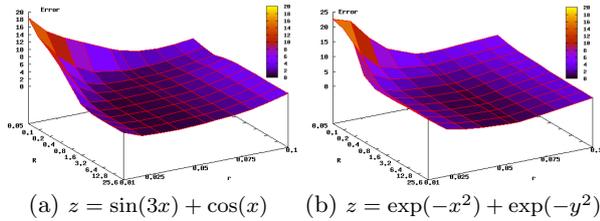
(a) $z = \sin(3x) + \cos(x)$ (b) $z = \exp(-x^2) + \exp(-y^2)$

Figure 5.6: Average deviation (in degrees) of computed principal curvature directions from exact ones for different values of parameters R and r.

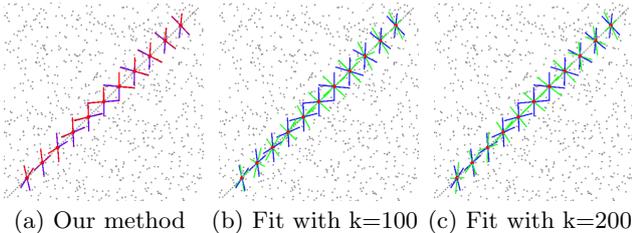(a) Our method (b) Fit with k=100 (c) Fit with k=200

Figure 5.7: Principal curvature directions on a biased dataset. Jet fitting (b-c) produces unreliable directions (in green) following oversampled areas.

compute principal curvatures of this fitted polynomial [6].

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, as a consequence of Hausdorff-robustness proved in Section 3. 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 Figure 5.5(a). The results obtained with our method and with the state of the art polynomial jet-fitting algorithm implemented in CGAL [6] are reported in Figure 5.7. 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.

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. Figure 5.8 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

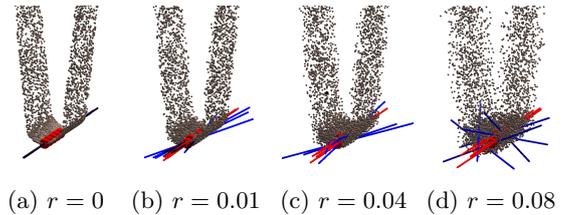(a) $r = 0$ (b) $r = 0.01$ (c) $r = 0.04$ (d) $r = 0.08$

Figure 5.8: 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 (d).

5.3 Detection of Sharp Edges

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. We consider a point as a feature if the ratio of the second smallest to the smallest eigenvalue is greater than some threshold parameter. This ratio measures the thickness of the infinitesimal Voronoi cell, as described in Section 2.4 and on Figure 2.3. The results do not seem to be sensitive to this threshold parameter, if it is within the range 10–40; for all the experiments below, we set it to 20.

Resilience to noise. In order to test the resilience of our method to noise, we perturbed the original 100k point cloud on the icosahedron by adding to each point a random vector uniformly chosen in a ball of given radius (the radius of the noise). Figure 5.9 shows the estimated feature directions on the icosahedron, for different noise and convolution radii.

We quantify the quality of the approximation using the three following distances between two oriented point clouds $C_1 = (p_i^1, d_i^1)$ and $C_2 = (p_i^2, d_i^2)$:

1. the maximum distance δ_∞ between a point p_i^2 in C_2 and its nearest neighbor in C_1 (this is the half-Hausdorff distance);

2. the average distance δ_{avg} between a point p_i^2 in C_2 and its nearest neighbor in C_1;

3. the average angular deviation (in degrees) α_{avg} between the direction d_i^2 of a point p_i^2 in C_2 and the direction of its nearest neighbor in C_1.

We also consider the symmetric quantities between C_2 and C_1. Table 2 shows the values of these quantities between the real oriented features C_1 of the icosahedron and the estimated ones C_2, for various noise and convolution radii. The first line of each experiment in the table corresponds to the distances from C_1 to C_2 (as above), while the second line corresponds to the distances from C_2 to C_1.

On the first line of each experiment, δ_∞ measures the presence of isolated outliers, while δ_{avg} measures the spread of the estimated features. One can see that increasing the convolution radius removes isolated outliers, but increases the spread of feature (in fact it seems that $\delta_{\mathrm{avg}} \simeq r/2$).

On the second line, δ_∞ and δ_{avg} both evaluate how far every point on an edge of the icosahedron is to an estimated

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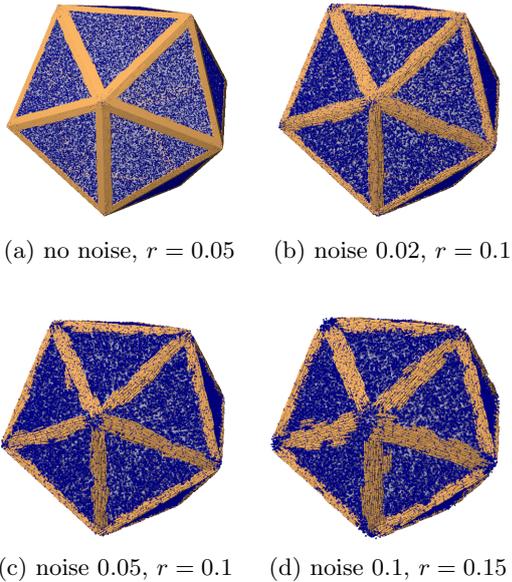

(c) noise 0.05, $r = 0.1$ (d) noise 0.1, $r = 0.15$

Figure 5.9: Estimated feature directions on an icosahedron of radius 1, with various convolution radii r and noise values.

Noise	r	δ_∞	δ_{avg}	α_{avg}
0.0	0.05	0.35	0.037	3.25
		0.076	0.011	1.42
0.0	0.1	0.118	0.051	0.33
		0.124	0.016	1.34
0.02	0.1	0.226	0.049	1.65
		0.139	0.020	3.46
0.05	0.1	0.220	0.050	2.82
		0.155	0.025	5.45
0.1	0.15	0.271	0.069	3.12
		0.178	0.036	7.94

Table 2: Distances between the estimated features and real features of an icosahedron, with varying noise radius and convolution radius

feature. Most of the error here comes from the corners: since we select feature points based on the ratio between the second and third eigenvalues, points nearby corner – at which these two eigenvalues are small – are discarded (see figures 2.3 and 5.9).

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 angle of 2α. As shown in Figure 5.10, the feature estimation method described above is able to reliably detect sharp edges whose external dihedral angle is as small; all the results were produced using the same convolution radius and threshold.

5.4 Results on more complex point clouds

In order to further illustrate the robustness of the feature estimation method, we tested it on a 300k point cloud sam-
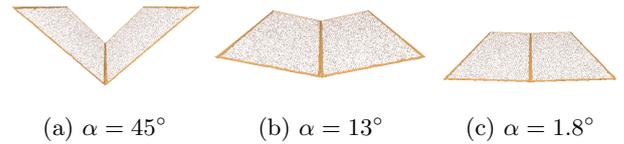

(a) $\alpha = 45°$ (b) $\alpha = 13°$ (c) $\alpha = 1.8°$

Figure 5.10: Estimated feature directions on a folded rectangle, for different values of the external angle α.

pled on the standard fandisk model. We then perturbed it by uniform noise of radius 0.01δ, where δ is the radius of the model. Figure 5.4.(d) shows the features and feature directions produced by our algorithm. While the features are diffused with very strong noise, the feature directions are quite stable and closely follow the edges of the model.

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 Figure 5.4.(a-c).

6. CONCLUSION AND FUTURE WORK

In this paper, we have described a method for computing principal curvatures and principal curvature directions and detecting sharp features and feature directions on point cloud data in a unified fashion. We have provided theoretical guarantees on the robustness of our method, giving a bound on the deviation of the estimated directions from the real ones in terms of the Hausdorff distance between the point cloud and the underlying surface. We implemented the method and tested it on multiple surfaces with varying amounts of sampling bias and noise.

One disadvantage of our method is that the principal directions that it produces are only defined up to a sign, and thus lack consistent orientation. One possible way of obtaining an orientation of the normal directions would be to intersect the Voronoi cells with *half* spheres on each side of the surface, and resolve ambiguities at the features, where one of the sides touches the medial axis.

In the future, we also intend to parallelize our algorithm and make the computations purely local. In addition, it would be interesting to apply our method to more challenging surface reconstruction problems.

7. REFERENCES

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graph.*, 27(3), 2008.
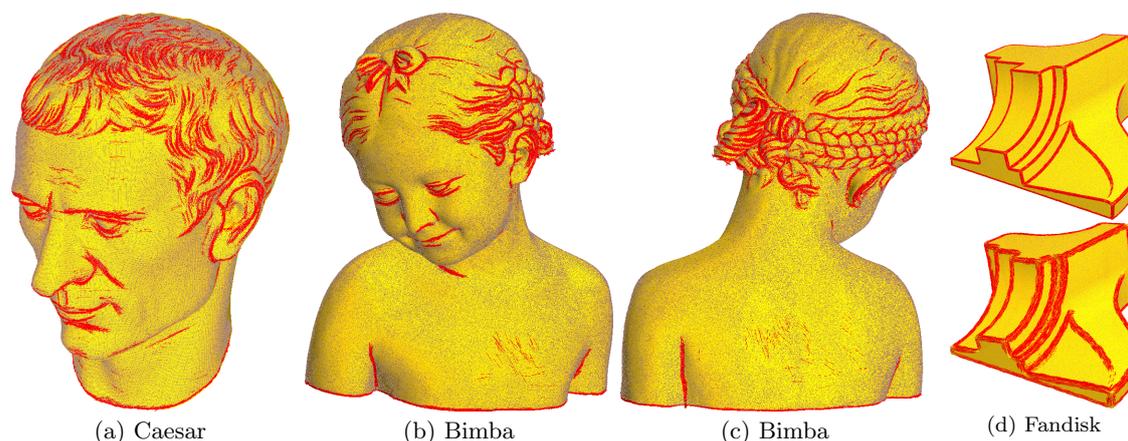
(a) Caesar   (b) Bimba   (c) Bimba   (d) Fandisk

Figure 5.11: Results of the feature detection algorithm on the Julius Caesar *(a)*, Bimba *(b-c)* and fandisk point clouds with no and 0.01δ noise, where δ is the radius of the model. *(d)*

[2] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Proc. of SGP*, pages 39–48, 2007.

[3] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.

[4] N. Amenta and Y.J. Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, 2004.

[5] D. Avis, B. K. Bhattacharya, and H. Imai. Computing the volume of the union of spheres. *The Visual Computer*, 3(6):323–328, 1988.

[6] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Comp. Aided Geom. Design*, pages 121–146, 2005.

[7] F. Chazal, D. Cohen-Steiner, and Q. Mérigot. Stability of boundary measures. *Arxiv Preprint arXiv:0706.2153*, 2007.

[8] D. Cohen-Steiner and J.M. Morvan. Restricted delaunay triangulations and normal cycle. *ACM Symp. Comp. Geom.*, pages 312–321, 2003.

[9] T. K. Dey and J. Sun. Normal and feature estimations from noisy point clouds. Technical Report OSU-CISRC-7/50-TR50, Ohio State University, 2005.

[10] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, 2005.

[11] 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*, pages 27–36, 2006.

[12] J. Huang and C. H. Menq. Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology. *Comp. Aided Geom. Design*, 34(2):149–165, 2002.

[13] J. Daniels II, T. Ochotta, Linh K. H., and Cláudio T. S. Spline-based feature curves from point-sampled geometry. *The Visual Computer*, 24(6):449–462, 2008.

[14] P. Jenke, M. Wand, and W. Straßer. Patch-graph reconstruction for piecewise smooth surfaces. In *Proc. of Vision, Modeling and Visualization*, 2008.

[15] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Comp. Aided Geom. Design*, 22(7):680–692, 2005.

[16] Y. Lipman, D. Cohen-Or, and D. Levin. Data - dependent mls for faithful surface approximation. In *Eurographics Symp. Geom. Proc.*, pages 59–67, 2007.

[17] E. Magid, O. Soldea, and E. Rivlin. A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data. *CVIU*, 107(3):139–159, 2007.

[18] J.-L. Maltret and M. Daniel. Discrete curvatures and applications : a survey. Technical report, Laboratoire des Sciences de l'Information et des Systemes, 2002.

[19] D. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching. `http://www.cs.umd.edu/~mount/ANN/`.

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

[21] M. Pauly, R. Keiser, and M. Gross. Multi-scale Feature Extraction on Point-Sampled Surfaces. In *Computer Graphics Forum*, 2003.

[22] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering structural regularity in 3d geometry. *ACM Trans. Graph.*, 27(3), 2008.

[23] S. Pion and M. Teillaud. 3d triangulations. In CGAL Editorial Board, editor, *CGAL-3.2 User and Reference Manual*. 2006.

[24] H. Pottmann, J. Wallner, Q. Huang, and Y.L. Yang. Integral invariants for robust geometry processing. *Comp. Aided Geom. Design*, 26:37–60, 2009.

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

[26] X. Tang. A sampling framework for accurate curvature estimation in discrete surfaces. *IEEE Trans. Vis. Comp. Graph.*, 11(5):573–583, 2005.

[27] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. *IEEE Int. Conf. on Comp. Vis.*, pages 902–907, 1995.

[28] W.-S. Tong and C.-K. Tang. Robust estimation of adaptive tensors of curvature by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):434–449, 2005.

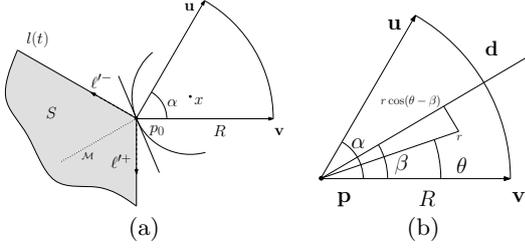[29] P. Yang and X. Qian. Direct computing of surface curvatures for point-set surfaces. In *Proc. of SBG*, 2007.

Figure A.12: Planar view of the Voronoi cell of a point on an edge of a piecewise smooth surface

APPENDIX

In this section, we assume that S is a piecewise smooth hyper-surface in R^3, and p_0 is lying on an edge separating two smooth parts of the surface. Specifically, we assume that the edge can be represented as a differentiable curve $\gamma(t)$ parameterized by arc-length, with $\gamma(0) = p_0$.

Convolved VCM for a feature point. Let S be an embedded piecewise smooth surface in $\mathbb{R}^3$. The *normal cone* at a point $p \in S$ is by definition

$$\mathcal{N}_p S = \{\lambda(x - p)\,;\, \lambda > 0 \text{ and } \mathrm{p}_S(x) = p\}$$

At a smooth point p of S, $\mathcal{N}_p S$ is simply the ray generated by the normal at p. In general, if the local feature size of S at a point p is greater than R, then $\mathrm{p}_K^{-1}(\{p\}) \cap \mathrm{B}(p, R) = p + (\mathcal{N}_p S \cap \mathrm{B}(0, R))$: this means that the VCM $\mathcal{V}_{S,S^R}$ at that point is fully described by the normal cone. Since the local feature size vanishes at any point on a sharp edge, we have to introduce a weaker notion of feature size. The 1-*sided local feature size*, or *lfs*[1], of a point p is defined as:

$$\mathrm{lfs}_S^1(p) = \inf\{R\,;\, \forall n \in \mathcal{N}_p S,\ \|n\| \leqslant 1 \Longrightarrow \mathrm{p}_S(p + Rn) = p\}$$

PROPOSITION A.1. *Let S be a piecewise smooth surface embedded in $\mathbb{R}^3$, and p_0 be a point on a sharp edge of S, with $\mathrm{lfs}_S^1(p_0) \geqslant R$. Let $\vec{t}$ be the tangent to this edge at p_0, and $\vec{u}$, $\vec{v}$ be the projections of the two outward normals to S at p_0 on the orthogonal plane to $\vec{t}$. Then,*

$$\mathcal{V}_{S,S^R}(\mathrm{B}(p_0, r)) = \frac{R^4 r}{8}[(\alpha - \sin(\alpha))\,e_1 e_1^{\mathbf{T}} + (\alpha + \sin(\alpha))\,e_2 e_2^{\mathbf{T}} + O(r/R)]$$

where $\vec{e_1} = \frac{\vec{u} - \vec{v}}{\|\vec{u} - \vec{v}\|}$, $\vec{e_2} = \frac{\vec{u} + \vec{v}}{\|\vec{u} + \vec{v}\|}$, and $\cos(\alpha) = \langle\vec{u}|\vec{v}\rangle$.

To describe the VCM of S near p_0, we first need to characterize the Voronoi cell $\mathrm{p}_S^{-1}(p_0) = \{x \in \mathbb{R}^3 | \mathrm{p}_S(x) = \gamma(t)\}$.

Lemma A.2. *The infinitesimal Voronoi cell $\mathrm{p}_S^{-1}(p_0)$ is contained in the positive cone Q generated by the vectors $\vec{u}$ and $\vec{v}$, i.e.*

$$\mathrm{p}_S^{-1}(\{p_0\}) \subseteq Q := \{p_0 + a\vec{u} + b\vec{v}\,;\, a, b \geqslant 0\}$$

Proof. Consider any point $x \in \mathbb{R}^d$ with $\mathrm{p}_S(x) = p_0$. By definition of the projection, the open ball $\mathrm{B}(x, \mathrm{d}(x, p_0))$ does not contain any points in S and in particular points of the curve γ. Hence, it must be tangent to γ at p_0; this can be rephrased as $\langle x - p_0 | \vec{t} \rangle = 0$. Let P denote the plane $p_0 + \{\vec{t}\}^\perp$. The point x can be written as $x = \gamma(t) + a\vec{u} + b\vec{u}$.
We let ℓ be a (local) parametrization of the intersection $P \cap S$, with $\ell(0) = p_0$, and $\ell'^-(0)$ and $\ell'^-(0)$ be the left and right derivatives of ℓ at 0. We suppose that ℓ is oriented so that $\langle\ell'^-(0)|\vec{u}\rangle > 0$ and $\langle\ell'^+(0)|\vec{v}\rangle < 0$. Since by hypothesis, x projects onto p_0, we have $\langle \ell(t) - p_0 | x - p_0 \rangle \leqslant 0$ for all values of t. For $t < 0$, we also have the Taylor expansion

$\ell(t) = p_0 + t\ell'^-(0) + o(t)$; using the inequality above and $\langle\ell'^-(0)|\vec{u}\rangle = 0$, we deduce $b\langle\ell'^-(0)|\vec{v}\rangle \geqslant 0$, and finally $b \geqslant 0$. The same argument proves that $a \geqslant 0$. $\square$

Because of the 1-lfs assumption, every point in Q within distance R of p_0 projects onto S at p_0. This means that if S^R denotes the R-neighborhood of S, one has $\mathrm{p}_S^{-1}(\{p_0\}) \cap S^R = Q \cap S^R$: this intersection is a circular slice with radius R and angle α which equals the dihedral angle between the two smooth parts of the shape (Figure A.12(a)).

Lemma A.3. *Let $\vec{u}, \vec{v}$ be two unit vectors in the plane, and $Q = \{a\vec{u} + b\vec{v}\} \cap \mathrm{B}(0, R)$. The eigenpairs of the covariance matrix $\mathrm{cov}(Q, 0)$ are:*

$$\lambda_1 = \frac{R^4}{8}\left(\alpha - \sin(\alpha)\right),\ e_1 = \vec{u} - \vec{v} \quad and$$

$$\lambda_2 = \frac{R^4}{8}\left(\alpha + \sin(\alpha)\right),\ e_2 = \vec{u} + \vec{v}$$

where $\cos(\alpha) = \langle\vec{u}|\vec{v}\rangle$.
Proof. Consider the covariance in some fixed unit direction $\vec{d}$ which is at angle β from $\mathbf{v}$ (Figure A.12(b)). Then, with $M = \mathrm{cov}(Q, 0)$,

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

To find the direction $\vec{d}$ corresponding to an eigenvector of M we need to find the minimizer/maximizer of the above quantity with respect to β. For this, we set the derivative with respect to β to 0, and get:

$$-\cos(2(\alpha - \beta)) + \cos(2\beta) = 0 \Longleftrightarrow \beta = \frac{\alpha}{2} \text{ or } \beta = \frac{\alpha - \pi}{2}$$

To compute the eigenvalues associated with these eigenvectors, we plug in the values into (A.2) to get $\lambda_1 = \frac{R^4}{8}\left(\alpha - \sin(\alpha)\right)$ and $\lambda_2 = \frac{R^4}{8}\left(\alpha + \sin(\alpha)\right)$. $\square$

Proof of proposition A.1. By definition, we have, with $B = \mathrm{B}(p_0, r)$:

$$\mathcal{V}_{S,E}(B(p_0, r)) = \int_{E \cap \mathrm{p}_K^{-1}(B \cap S)}(x - \mathrm{p}_S(x))(x - \mathrm{p}_S(x))^{\mathbf{T}}\mathrm{d}x$$

$$= \int_{E \cap \mathrm{p}_K^{-1}(B \cap e(t))}(x - \mathrm{p}_S(x))(x - \mathrm{p}_S(x))^{\mathbf{T}}\mathrm{d}x \quad \text{(A.3)}$$

$$+ \int_{E \cap \mathrm{p}_K^{-1}(B \cap (S_1 \cup S_2))}(x - \mathrm{p}_S(x))(x - \mathrm{p}_S(x))^{\mathbf{T}}\mathrm{d}x \quad \text{(A.4)}$$

where S_1 and S_2 are the two smooth parts of the surface on either side of the edge $\gamma(t)$. The computations for the smooth case prove that the order of A.4 is $O(R^3 r^2)$, whereas the integral along the curve A.3 has an order of $O(R^4 r)$.

$$\begin{aligned}\int_{S^R \cap \mathrm{p}_K^{-1}(B \cap \mathrm{Im}\gamma)}&(x - \mathrm{p}_S(x))(x - \mathrm{p}_S(x))^{\mathbf{T}}\mathrm{d}x \\ &\simeq \int_{-r}^{r}\int_{S^R \cap \mathrm{p}_K^{-1}(\gamma(t))}(x - \mathrm{p}_S(x))(x - \mathrm{p}_S(x))^{\mathbf{T}}\mathrm{d}x\mathrm{d}t \\ &\simeq 2r\int_{S^R \cap \mathrm{p}_K^{-1}(p_0)}(x - \mathrm{p}_S(x))(x - \mathrm{p}_S(x))^{\mathbf{T}}\mathrm{d}x\end{aligned}$$

We conclude using the two previous lemmas. $\square$