Eurographics Symposium on Geometry Processing 2023
P. Memari and J. Solomon
(Guest Editors)

*Volume 42 (2023), Number 5*

# Lightweight Curvature Estimation on Point Clouds with Randomized Corrected Curvature Measures

J.-O. Lachaud[1] and D. Coeurjolly[2] and C. Labart[1] and P. Romon[3] and B. Thibert[4]

[1]Université Savoie Mont Blanc, France
[2]Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS UMR5205, France
[3]Université Gustave Eiffel, France
[4]Université Grenoble Alpes, France

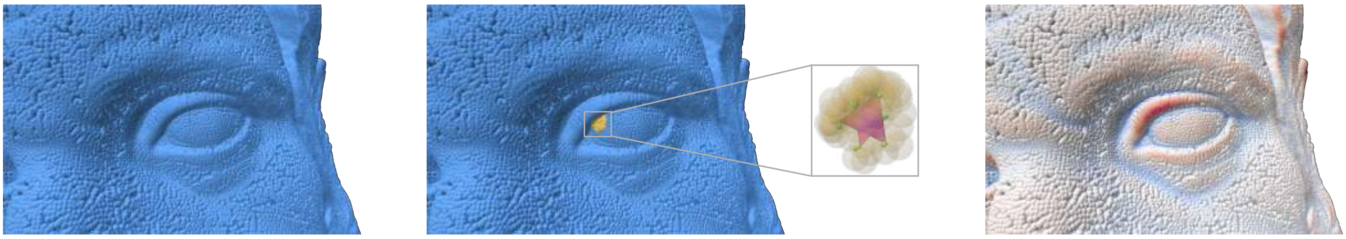

**Figure 1:** *Our technique uses corrected curvature measures on (quasi-)random triangles to estimate differential quantities on oriented point clouds: stable and accurate estimations (mean curvature here) are achieved with few neighbors (50) and triangles (2).*

**Abstract**

*The estimation of differential quantities on oriented point cloud is a classical step for many geometry processing tasks in computer graphics and vision. Even if many solutions exist to estimate such quantities, they usually fail at satisfying both a stable estimation with theoretical guarantee, and the efficiency of the associated algorithm. Relying on the notion of* corrected curvature measures *[LRT22, LRTC20] designed for surfaces, the method introduced in this paper meets both requirements. Given a point of interest and a few nearest neighbours, our method estimates the whole curvature tensor information by generating random triangles within these neighbours and normalising the corrected curvature measures by the corrected area measure. We provide a stability theorem showing that our pointwise curvatures are accurate and convergent, provided the noise in position and normal information has a variance smaller than the radius of neighbourhood. Experiments and comparisons with the state-of-the-art confirm that our approach is more accurate and much faster than alternatives. The method is fully parallelizable, requires only one nearest neighbour request per point of computation, and is trivial to implement.*

**CCS Concepts**

• ***Computing methodologies*** → ***Shape analysis; Point-based models;*** • ***Theory of computation*** → ***Computational geometry;***

## 1. Introduction

Point clouds data representing 3D shapes are now extremely common due to the developments of LiDAR technology or photogrammetry, which offer now affordable devices to capture quite accurately the 3D geometry of real shapes, from small objects to buildings or even reliefs. As a shape representation, raw point clouds are not reliable enough nor structured enough for most targeted applications, hence they are usually first processed and then transformed into other representations. It is worthy to note that point clouds can be massive with up to billions of points, present generally uneven sampling and their lack of structure prevent simple parameterization, hence signal processing techniques are difficult to apply.

Point cloud processing often relies on the extraction of local invariant features within point clouds, and is used in many applications like segmentation [KSNS07, KHS10, KLM*13, TGDM18, PCDKC20], registration [GMGP05, MDS15, YZLB21], static [Lia21] or dynamic compression [YSY*22], to quote a few. These features are most often related to recognizable geometric features like edges, ridges, valleys, peaks, creases, depressions, etc. Since all of them can be computed from curvature information, this objective is met by inferring the second order differential geometry of the real-world shape that was acquired by the scanning device. Ideally, the inferred curvature information should offer convergence guarantees according to the density of the sampling, while being robust to both perturbations in position and normal data. Finally, from a practical point of view, computing robust and precise curvature information should be fast and parallelizable to accommodate massive data, while keeping a small memory footprint.

This paper proposes a new method to infer the local curvature information from oriented point clouds (collection of points in $\mathbb{R}^3$ equipped with normal vectors). It relies on the recent notion of *Corrected Normal Current (CNC)* [LRT22], which is a theoretical setting that induces curvature measures on very general *surfaces*, and for instance simple formulae to approximate area and curvature measures within a triangle [LRTC20].

**Contributions.** Our key idea is to notice that measures do not require a surface topology to provide consistent results. It suffices then to generate arbitrary triangles in the neighborhood of the point of interest to collect area and curvature measures. When summed consistently, these measures reinforce their estimations. Dividing any one of the curvature measures (mean, Gaussian, anisotropic) by the area measure over the same triangles provides accurate pointwise estimation of the corresponding curvature.

We provide a theoretical result that bounds the pointwise mean curvature error even in presence of position and normal perturbations. The error is essentially constituted by a constant times the neighborhood radius, plus a stochastic term with zero expectancy and variance going to zero as the number of triangles increases. Otherwise said, when perturbations are small, a small radius with few points induces precise estimations, while when perturbations increase, a bigger radius authorizes more triangles and stabilizes the estimations, with a small loss in precision.

We then propose several strategies to generate triangles within the neighborhood of the point of interest. As a comparison basis, we have tested a local Delaunay triangulation reconstruction, which induces accurate results at the highest computational cost. We show after that the simplest uniform random triangle generation leads to the same accuracy with faster computations. We finally exhibit two variants that only generate two triangles, one is extremely fast yet accurate when data is not too noisy, the other is also fast and yet remains robust to noise. Finally, we present extensive experiments and comparisons with other state-of-the-art methods for computing curvatures of point clouds. It appears that our method is both accurate and robust to noise in practice, requires smaller neighborhoods for the same accuracy compared to the current best method, and is way faster in most practical cases.

## 2. Related works

There is an abundant literature on the problem of curvature estimation from point clouds. We discuss here the main approaches and especially the ones that come with some theoretical guarantees. A more complete review can be found in the recent paper [LCBM21], which seems to be the state-of-the-art for now.

**Polynomial fitting and learning methods.** Osculating jets (or Jet-Fitting) [CP05] fits a bi-variate polynomial matching neighbors over a tangent plane. Wave jets proposed by Béarzi *et al.* [BDC18] does a similar job but with radial polynomials. Results are precise for perfect data, with some guarantees, but these methods collapse in the presence of noise in practice, and are computationally costly. Besides, as noted by Khameneifar and Ghorbani [KG19], the choice of the neighborhood is important when fitting surfaces. Learning methods fits non-linear simple functions to the data. Guerrero *et al.* [GKOM18] introduced PCPNet for curvature estimations. Based on PointNet [QSMG17], these architectures are not designed to process massive point clouds. They compare favorably to JetFitting in presence of noise, but with no theoretical bounds.

**Point Set Surfaces and extensions.** Point Set Surfaces [ABCO*01] use Moving Least Squares regression to locally approach scattered data with a smooth manifold. Many variants have been proposed over the years, e.g. to improve efficiency, genericity, principal curvature computation, robustness to uneven sampling, etc. Algebraic Point Set Surfaces [GG07] directly fits algebraic spheres in order to avoid the estimation of a reference plane, but remains limited to mean curvature estimation. Its multiscale formulation makes it useful for multiscale analysis of points clouds [MGB*12], and it was shown recently by Lejemble *et al.* that the whole curvature tensor information can be estimated with scale differentiation [LCBM21]. This latter work not only is accurate in practice, but it also provides theoretical guarantees of accuracy in presence of perturbation in noise position.

**Integral invariants.** Since Weyl's tube formula, we know that surface geometry is related to integral quantities. Integral invariants are classical formula relating curvatures at a point of interest with a local volume or tensor [YLHP06, PWY*07, PWHY09], and are shown to provide more accurate results than PCA. Although more designed for mesh analysis, they can be adapted to point cloud analysis using weights or kernels [DM14]. However, there is no stability theorem for noisy data.

**Distance function and covariance measures.** A generic approach to analyze the local geometry of compact sets was proposed by Mérigot *et al.* [MOG10], with the elegant idea of analyzing the distance function to the set. They show that the Voronoi Covariance Measure is stable in the presence of Hausdorff noise. Its diagonalization provides stable characterization of edges or ridges, but does not provide pointwise curvature convergence. This work was extended by Cuel *et al.* [CLMT15] to be robust even to outliers, using a distance to a measure formulation.

**Curvature measures.** Generalizing curvatures to non-smooth geometry dates to the works of Federer [Fed59]. They have been generalized progressively to include triangulations [Win82, Zäh87, Fu94], and stability results have been established for meshes when vertex positions and normals are close to the smooth manifold cite [Fu93, CSM03, CSM06]. These works were recently extended to piecewise smooth surfaces whose normal vector field is not convergent [LRT22]. Curvature measures were also extended to cloud of points using offsets [CCSLT09], with some stability results, at the price of costly computations.

**Varifolds.** Introduced first for solving abstract shape optimization problems [Alm66], varifolds constitute another approach to define a geometry on a wide class of objects, e.g. it includes cloud of points with normals. Buet *et al.* have shown how to extract mean curvature information [BLM17] and recently a weak second fundamental form [BLM18, BLM19]. This general framework does not yet provide easy and explicit stability or convergent rates.

**Grassmannian embedding.** A few works in computer graphics have also extended the Euclidean space with the Gaussian sphere to carry out some specialized tasks. We may quote the works of [PSH*04, LZH*06, LB13], who use this trick to define a metric on surfaces that is dependent on features: indeed a feature implies a strong change in the normal vector, hence paths in this metric are much longer across features. This approach is quite useful to do feature-aware surface processing, but it cannot be used directly to estimate quantitatively curvatures.

## 3. Randomized corrected curvature measures for point clouds

**Summary.** Let us assume we have a point cloud of $N$ points with positions and normals $(\hat{\mathbf{x}}_m, \hat{\mathbf{u}}_m)_{m=1\dots N}$. Our curvature estimates are computed per point of interest independently of the others and the method is thus completely parallel. For a given point, we compute its nearest neighbors at distance less than some user parameter $\delta/2$. Then $L$ triangles are built randomly by picking points within these neighbors. Curvature measures are evaluated for every triangle and summed up. The same is done with area measures, also summed up. Pointwise curvatures (mean, Gaussian, principal curvatures and directions) are estimated by dividing the corresponding curvature measure with the area measure. This procedure is detailed below.

**Interpolated corrected curvature measures on a triangle.** We start by recalling the simple closed-form expressions, detailed in [LRTC20], that associate different measures of the area and curvatures to a single triangle $\tau_{ijk}$, with vertices $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ and unit normals $(\mathbf{u}_i, \mathbf{u}_j, \mathbf{u}_k)$ at these vertices.

**Property 1** The *interpolated corrected curvature measures* take the following values on a triangle $\tau_{ijk}$, with vertices $i$, $j$, $k$:

$$
\begin{aligned}
\mu_{\mathbf{u}}^{(0)}(\tau_{ijk}) &= \frac{1}{2}\left\langle \bar{\mathbf{u}} \,|\, (\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i) \right\rangle, \\
\mu_{\mathbf{u}}^{(1)}(\tau_{ijk}) &= \frac{1}{2}\left\langle \bar{\mathbf{u}} \,|\, (\mathbf{u}_k - \mathbf{u}_j) \times \mathbf{x}_i + (\mathbf{u}_i - \mathbf{u}_k) \times \mathbf{x}_j + (\mathbf{u}_j - \mathbf{u}_i) \times \mathbf{x}_k \right\rangle, \\
\mu_{\mathbf{u}}^{(2)}(\tau_{ijk}) &= \frac{1}{2}\left\langle \mathbf{u}_i \,|\, \mathbf{u}_j \times \mathbf{u}_k \right\rangle, \\
\mu_{\mathbf{u}}^{(\mathbf{X},\mathbf{Y})}(\tau_{ijk}) &= \frac{1}{2}\left\langle \bar{\mathbf{u}} \,|\, \left\langle \mathbf{Y} \,|\, \mathbf{u}_k - \mathbf{u}_i \right\rangle \mathbf{X} \times (\mathbf{x}_j - \mathbf{x}_i) \right\rangle \\
&\quad - \frac{1}{2}\left\langle \bar{\mathbf{u}} \,|\, \left\langle \mathbf{Y} \,|\, \mathbf{u}_j - \mathbf{u}_i \right\rangle \mathbf{X} \times (\mathbf{x}_k - \mathbf{x}_i) \right\rangle,
\end{aligned}
$$

where $\langle \cdot \mid \cdot \rangle$ denotes the usual scalar product, $\bar{\mathbf{u}} = \frac{1}{3}(\mathbf{u}_i + \mathbf{u}_j + \mathbf{u}_k)$.

All these scalar values associated to a triangle $(\tau_{ijk})$ are in some sense the integration of well-known geometric quantities over the triangle. Measure $\mu_{\mathbf{u}}^{(0)}$, called *corrected area density*, is easily spotted as the area of the triangle, however, weighted by the alignment between the geometric normals and the provided normals $\mathbf{u}$. This correction comes from the computation of this measure in the Grassmannian (more details will be given in Section 4).

The other measures are all related to integrals of curvatures, and mix positions and normal information. Measure $\mu_{\mathbf{u}}^{(1)}$ is the *corrected mean curvature density* and measure $\mu_{\mathbf{u}}^{(2)}$ the *corrected Gaussian curvature density* of this triangle. The (anisotropic) measure $\mu_{\mathbf{u}}^{(\mathbf{X},\mathbf{Y})}$ is the trace of the *corrected second fundamental form* along directions $\mathbf{X}$ and $\mathbf{Y}$. While the smooth second fundamental form is naturally a symmetric 2-tensor, there is no easy way to define tangent directions at a vertex, so the anisotropic measure depends on two 3D vectors; when $\mathbf{X}$ and $\mathbf{Y}$ are tangent, $\mu_{\mathbf{u}}^{(\mathbf{X},\mathbf{Y})}$ is close to the second fundamental form applied to these vectors, while its value along normal direction tends to zero asymptotically.

**Randomized local measures of curvatures.** Of course, our data is constituted of unrelated points, and does not form a triangle mesh. The method is parameterized by a length $\delta/2$ (or equivalently by an integer $K$) such that the ball of radius $\delta/2$ centered on the point of interest $\hat{\mathbf{x}}$ contains approximately $K$ points. Let $(\hat{\mathbf{x}}_i)_{i\in I}$ be the points near $\hat{\mathbf{x}}$, with $I \subset \{1,\dots,N\}$ and $\#(I) \approx K$.

We randomly build $L$ triangles $(\hat{\tau}_l)_{l=1\dots L}$ from these $K$ points among the possible $\binom{K}{3}$ triangles. This random generation can be made uniformly, or with smarter sampling strategy (see Section 5). Each triangle $\hat{\tau}_l$ with vertices $(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j, \hat{\mathbf{x}}_k)$ *must be oriented consistently* with its average normal vector $\bar{\hat{\mathbf{u}}}$, i.e. $\mu_{\hat{\mathbf{u}}}^{(0)}(\hat{\tau}_{ijk})$ *must be non-negative*. If not, we simply switch the role of $\hat{\mathbf{x}}_j$ and $\hat{\mathbf{x}}_k$.

We then add measures on every triangle for each kind of measure. We also build a $3 \times 3$ tensor from the anisotropic measure, by choosing for $\mathbf{X}$ and $\mathbf{Y}$ the basis vectors $\mathbf{e}_i$:

$$
\hat{A}^{(0)} := \sum_{l=1}^{L} \mu_{\hat{\mathbf{u}}}^{(0)}(\hat{\tau}_l) \quad \hat{A}^{(1)} := \sum_{l=1}^{L} \mu_{\hat{\mathbf{u}}}^{(1)}(\hat{\tau}_l) \quad \hat{A}^{(2)} := \sum_{l=1}^{L} \mu_{\hat{\mathbf{u}}}^{(2)}(\hat{\tau}_l)
$$

$$
\hat{A}^{(\mathbf{M})} := \sum_{l=1}^{L} \begin{bmatrix}
\mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_1,\mathbf{e}_1)}(\hat{\tau}_l) & \mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_1,\mathbf{e}_2)}(\hat{\tau}_l) & \mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_1,\mathbf{e}_3)}(\hat{\tau}_l) \\
\mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_2,\mathbf{e}_1)}(\hat{\tau}_l) & \mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_2,\mathbf{e}_2)}(\hat{\tau}_l) & \mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_2,\mathbf{e}_3)}(\hat{\tau}_l) \\
\mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_3,\mathbf{e}_1)}(\hat{\tau}_l) & \mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_3,\mathbf{e}_2)}(\hat{\tau}_l) & \mu_{\hat{\mathbf{u}}}^{(\mathbf{e}_3,\mathbf{e}_3)}(\hat{\tau}_l)
\end{bmatrix}
$$

**Estimations of curvatures for the point of interest.** We divide each curvature measure by the area measure to get a pointwise estimate for each kind of curvature. The *mean curvature estimate* $\hat{H}(\hat{\mathbf{x}})$ and the *Gaussian curvature estimate* $\hat{G}(\hat{\mathbf{x}})$ at point $\hat{\mathbf{x}}$ are then:

$$
\hat{H}(\hat{\mathbf{x}}) := \hat{A}^{(1)}/\hat{A}^{(0)} \qquad \hat{G}(\hat{\mathbf{x}}) := \hat{A}^{(2)}/\hat{A}^{(0)}.
$$

To get principal curvatures and directions, we proceed similarly as in [LRTC20]. The anisotropic measure tensor is not symmetric in general, although it tends asymptotically to the second fundamental form. We symmetrizes it and forces the orthogonality of tangent vectors ($C$ is a big constant, chosen as $1000\delta$ in all experiments):

$$
\hat{\mathbf{P}} := \frac{1}{\hat{A}^{(0)}}\left(\frac{1}{2}\left(\hat{A}^{(\mathbf{M})} + \hat{A}^{(\mathbf{M})\mathsf{T}}\right) + C\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}\right).
$$

Then the *first and second principal curvature estimates,* $\hat{\kappa}_1$ *and* $\hat{\kappa}_2$ at $\hat{\mathbf{x}}$, and their associated *principal direction estimates* $\hat{\mathbf{v}}_1$ *and* $\hat{\mathbf{v}}_2$ at point $\hat{\mathbf{x}}$ are respectively defined as:

$$
\hat{\kappa}_1(\hat{\mathbf{x}}) := -\lambda_2(\hat{\mathbf{P}}), \qquad \hat{\mathbf{v}}_1(\hat{\mathbf{x}}) := \mathbf{z}_2(\hat{\mathbf{P}}), \tag{1}
$$

$$
\hat{\kappa}_2(\hat{\mathbf{x}}) := -\lambda_1(\hat{\mathbf{P}}), \qquad \hat{\mathbf{v}}_2(\hat{\mathbf{x}}) := \mathbf{z}_1(\hat{\mathbf{P}}), \tag{2}
$$

where $\lambda_1(\hat{\mathbf{P}}) \leq \lambda_2(\hat{\mathbf{P}}) \leq \lambda_3(\hat{\mathbf{P}})$ are the eigenvalues of $\hat{\mathbf{P}}$ and $(\mathbf{z}_1(\hat{\mathbf{P}}), \mathbf{z}_2(\hat{\mathbf{P}}), \mathbf{z}_3(\hat{\mathbf{P}}))$ their associated eigenvectors.

## 4. Stability of curvature estimates

This section is devoted to the proof of the accuracy of our curvature estimators. We focus on the mean curvature estimator $\hat{H}$, but similar arguments lead to the convergence of the Gaussian curvature estimator $\hat{G}$ as well as principal curvatures $\hat{\kappa}_1$ and $\hat{\kappa}_2$. Principal directions $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ are also convergent except around umbilic points, using results from matrix perturbation theory.

### 4.1. Background

**Invariant and anisotropic forms.** Curvature measures $\mu^{(k)}$ are obtained by integrating a normal current against canonical invariant 2-forms $\omega^{(k)}$ defined on the Grassmannian $\mathbb{R}^3 \times \mathbb{S}^2$ (for $k \in \{0,1,2\}$). For any point $(\mathbf{x},\mathbf{u}) \in \mathbb{R}^3 \times \mathbb{S}^2$ and tangent vectors $\boldsymbol{\zeta}, \boldsymbol{\nu} \in T_{(\mathbf{x},\mathbf{u})}(\mathbb{R}^3 \times \mathbb{S}^2)$, we write $\boldsymbol{\zeta} = (\boldsymbol{\zeta}_p, \boldsymbol{\zeta}_n)$ and $\boldsymbol{\nu} = (\boldsymbol{\nu}_p, \boldsymbol{\nu}_n)$ in $\mathbb{R}^3 \times \mathbb{R}^3$ (separating position and velocity). Then the invariant forms are given by:

$$\omega^{(0)}_{(\mathbf{x},\mathbf{u})}(\boldsymbol{\zeta},\boldsymbol{\nu}) = \det(\mathbf{u},\boldsymbol{\zeta}_p,\boldsymbol{\nu}_p),\ \omega^{(2)}_{(\mathbf{x},\mathbf{u})}(\boldsymbol{\zeta},\boldsymbol{\nu}) = \det(\mathbf{u},\boldsymbol{\zeta}_n,\boldsymbol{\nu}_n),$$
$$\omega^{(1)}_{(\mathbf{x},\mathbf{u})}(\boldsymbol{\zeta},\boldsymbol{\nu}) = \frac{1}{2}\left(\det(\mathbf{u},\boldsymbol{\zeta}_p,\boldsymbol{\nu}_n) + \det(\mathbf{u},\boldsymbol{\zeta}_n,\boldsymbol{\nu}_p)\right).$$

Similarly, the anisotropic curvature form (for the directions $\mathbf{X},\mathbf{Y}$) is given by [CSM06]

$$\omega^{(\mathbf{X},\mathbf{Y})}_{(\mathbf{x},\mathbf{u})} = (\mathbf{u}\times\mathbf{X},0)^\flat \wedge (0,\mathbf{Y})^\flat,$$

where $\mathbf{w}^\flat$ denotes the linear map $\langle \mathbf{w} \mid \cdot \rangle$ dual to vector $\mathbf{w}$.

**Curvatures and curvature measures on a surface.** Let $S$ be some oriented connected smooth enough ($C^3$) surface in $\mathbb{R}^3$, and for any point $\mathbf{q}$ in $S$, let $\mathbf{n}(\mathbf{q})$ be the unit normal vector at $\mathbf{q}$. The reach of $S$ is the distance of $S$ to its medial axis, and we assume that it is strictly positive. For any vector field $\mathbf{u}$ defined on some subset $X$ of $\mathbb{R}^3$, let $\Gamma_{\mathbf{u}}$ be the map $\mathbf{x} \mapsto (\mathbf{x},\mathbf{u}(\mathbf{x}))$. For instance, if we choose the normal vector field $\mathbf{n}$ for $\mathbf{u}$, $\Gamma_{\mathbf{n}}(S)$ embeds the surface $S$ into the Grassmannian. For any Borel set $B$ and any vector field $\mathbf{u}$ of support $X$, let:

$$\forall k \in \{0,1,2\}, \quad \mu^{(k)}_{\mathbf{u}}(B\cap X) := \int_{B\cap X} \Gamma^*_{\mathbf{u}}\omega^{(k)}. \tag{3}$$

In the smooth case and when choosing $\mathbf{u} = \mathbf{n}$, the current is exactly the normal bundle, so we derive the Lipschitz–Killing curvature forms, which measure the area, the mean curvature density and the Gaussian curvature density (for a consistent orientation):

$$\mu^{(0)}_{\mathbf{n}}(B\cap S) = \int_{B\cap S} \mathrm{d}\mathbf{a} = \mathrm{Area}(B\cap S), \tag{4}$$
$$\mu^{(1)}_{\mathbf{n}}(B\cap S) = \int_{B\cap S} H\mathrm{d}\mathbf{a}, \qquad \mu^{(2)}_{\mathbf{n}}(B\cap S) = \int_{B\cap S} G\mathrm{d}\mathbf{a}, \tag{5}$$

where $H$ and $G$ denote the mean and Gaussian curvature, and $\mathrm{d}\mathbf{a}$ is the area form.

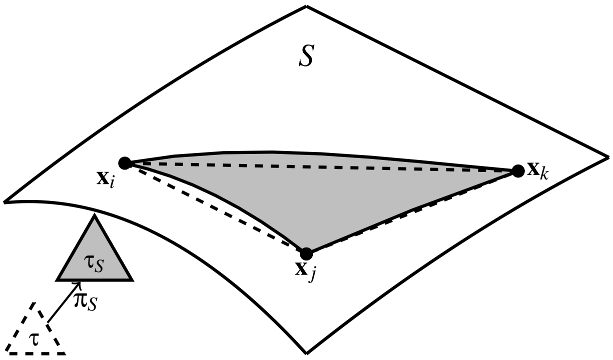

**Figure 2:** *Main notations: $S$ is the underlying smooth surface; $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ are three points of $S$; $\tau$ is their convex hull, i.e. the planar triangle joining them; $\tau_S := \pi_S(\tau)$ is the projection of $\tau$ onto $S$.*

### 4.2. Relating measures on a triangle and on its projection on $S$

Figure 2 illustrates the notations we will use throughout the proof. The cloud of points is for now assumed to sample a $C^3$-smooth surface $S$. Let $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ be three points of the cloud on $S$ such that the triangle $\tau$ spanned by these vertices stays within the reach of $S$. Let $\pi_S$ be the orthogonal projection on $S$ (which exists within the reach and which maps any point within the reach to its closest point on $S$), and let $\tau_S := \pi_S(\tau)$. The order of vertices is chosen such that it is consistent with the orientation of the normal vector. Let us now choose 3 unit vectors $\mathbf{u}_i$, $\mathbf{u}_j$ and $\mathbf{u}_k$ at vertices $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$ respectively. We linearly interpolate them within the triangle to build a vector field $\mathbf{u}$ on $\tau$. Missing proofs can be found in Appendix A.

**Area measure.** If $\delta$ bounds the distance between points $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ and if $\alpha$ bounds the difference between true normals $\mathbf{n}$ and interpolated normals $\mathbf{u}$, i.e. $\sup_\tau \|\mathbf{n}\circ\pi_S - \mathbf{u}\| \leq \alpha$, then the area measures of $\tau$ and $\tau_S$ are close:

$$|\mu^{(0)}_{\mathbf{n}}(\tau_S) - \mu^{(0)}_{\mathbf{u}}(\tau)| \leq \mathrm{Area}(\tau)\left(O(\delta^2)+\alpha\right). \tag{6}$$

**Mean curvature measure.** Now assume $\beta$ bounds the difference between the differentials of $\mathbf{u}$ and $\mathbf{n}$, i.e. $\sup_\tau \|\mathrm{D}(\mathbf{n}\circ\pi_S) - \mathrm{D}\mathbf{u}\| \leq \beta$. Then mean curvature measures of $\tau$ and $\tau_S$ are close:

$$|\mu^{(1)}_{\mathbf{n}}(\tau_S) - \mu^{(1)}_{\mathbf{u}}(\tau)| \leq O(\mathrm{L}_{\mathbf{n}}\delta^4) + O(\beta\delta^2) + O(\mathrm{L}_{\mathbf{u}}\alpha\delta^2), \tag{7}$$

if $\mathrm{L}_{\mathbf{n}}$ and $\mathrm{L}_{\mathbf{u}}$ are respectively the Lipschitz constants of $\mathbf{n}$ and $\mathbf{u}$.

**Relation with pointwise mean curvature.** If $\mathbf{q}$ is a point of $S$ at distance less than $\delta$ to any point of $\tau_S$, and let $\mathrm{L}_H$ be the Lipschitz constant of mean curvature $H$, then it holds:

$$|\mu^{(1)}_{\mathbf{n}}(\tau_S) - \mathrm{Area}(\tau_S)H(\mathbf{q})| \leq \mathrm{L}_H\mathrm{Area}(\tau_S)\delta. \tag{8}$$

**Bounds induced by equality of vector fields at sample points.** If we assume that $\mathbf{u} = \mathbf{n}$ on each vertex of the triangle $\tau$, then these quantities can be bounded above as follows:

$$\alpha \leq \beta\delta + O(\delta^2), \qquad \beta \leq O(\delta), \qquad \mathrm{L}_{\mathbf{u}} \leq \mathrm{L}_{\mathbf{n}}. \tag{9}$$

**Local relations.** Using equations (6) and (9) for the area measure, and (7), (8) and (9) for the mean curvature measure, it holds:

$$|\mu_{\mathbf{u}}^{(0)}(\tau) - \mathrm{Area}(\tau_S)| \leq \mathrm{Area}(\tau) O(\delta^2), \tag{10}$$

$$|\mu_{\mathbf{u}}^{(1)}(\tau) - \mathrm{Area}(\tau_S) H(\mathbf{q})| \leq O(\delta^3). \tag{11}$$

### 4.3. Introducing perturbations on positions and normals

Our input data is a collection $X$ of points $(\hat{\mathbf{x}}_i)_{i=1\dots N}$ in the space, and a collection $U$ of associated oriented unit normal $\hat{\mathbf{u}}_i$ with each point. They represent a perturbated sampling of a smooth surface $S$ with some normal information. We can write $\hat{\mathbf{x}}_i := \mathbf{x}_i + \varepsilon_i$ and $\hat{\mathbf{u}}_i := \mathbf{u}_i + \xi_i$, where the $\varepsilon_i$ form a sequence of independent and identically distributed (i.i.d.) random variables drawn from a distribution of expected value $\mathbf{0}$ and variance $\sigma_\varepsilon^2 \mathrm{Id}$, and where the $\xi_i$ form a sequence of i.i.d. random variables drawn from a distribution of expected value $\mathbf{0}$ and variance $\sigma_\xi^2 \mathrm{Id}$.

**Summing up measures on multiple triangles.** Let $\hat{\mathbf{q}}$ be a point of interest (e.g. some $\hat{\mathbf{x}} \in X$) and let $\mathbf{q}$ be its projection on $S$. Let $(\hat{\mathbf{x}}_i)_{i \in I}$ be the points in the ball centered on $\hat{\mathbf{q}}$ and of radius $\delta/2$, with $I \subset \{1, \dots, N\}$. Let $(\hat{\tau}_l)_{l=1\dots L}$ be $L$ triangles with vertices in $(\hat{\mathbf{x}}_i)_{i \in I}$. Order of vertices within triangle $\hat{\tau}_l$ is chosen so as $\mu_{\hat{\mathbf{u}}}^{(0)}(\hat{\tau}_l)$ is a non-negative number. To each triangle $\hat{\tau}_l$ corresponds an unperturbated triangle $\tau_l$ with vertices on the surface $S$.

Let $\hat{A}^{(0)} := \sum_{l=1}^{L} \mu_{\hat{\mathbf{u}}}^{(0)}(\hat{\tau}_l)$, $A^{(0)} := \sum_{l=1}^{L} \mu_{\mathbf{u}}^{(0)}(\tau_l)$, $\hat{A}^{(1)} := \sum_{l=1}^{L} \mu_{\hat{\mathbf{u}}}^{(1)}(\hat{\tau}_l)$, $A^{(1)} := \sum_{l=1}^{L} \mu_{\mathbf{u}}^{(1)}(\tau_l)$. Let also $\mathrm{A}$ be the total area of triangles $\tau_l$ and $\mathrm{A}_S$ be the total area of projected triangles $\pi_S(\tau_l)$.

Summing relation (10) and triangular inequality gives the first relation below, while summing relations (8) and (11) and applying triangular inequality gives the second one:

$$|A^{(0)} - \mathrm{A}_S| \leq \mathrm{A} O(\delta^2), \tag{12}$$

$$|A^{(1)} - \mathrm{A}_S H(\mathbf{q})| \leq \mathrm{L}_H \mathrm{A}_S \delta + O(L\delta^3). \tag{13}$$

Finally, dividing both summed measures provide the mean curvature estimation, i.e. $\hat{H}(\hat{\mathbf{q}}) := \hat{A}^{(1)}/\hat{A}^{(0)}$. We achieve the following stability result for this curvature estimator.

**Theorem 1** If we assume that we have generated enough triangles within the ball around $\hat{\mathbf{q}}$ so that $\mathrm{A}_S/L = \Theta(\delta^2)$, then

$$\begin{aligned} &|\hat{H}(\hat{\mathbf{q}}) - H(\mathbf{q})| \\ &\quad \leq \left| O(\delta) + \Theta(\delta^{-2}) \left( \bar{Z}_L^{(1)} - \bar{Z}_L^{(0)} H(\mathbf{q}) \right) \right| \Big/ \left| 1 + \Theta(\delta^{-2}) \bar{Z}_L^{(0)} \right|, \end{aligned}$$

where $\bar{Z}_L^{(0)}$ is the error law $\frac{1}{L}(\hat{A}^{(0)} - A^{(0)})$ and $\bar{Z}_L^{(1)}$ is the error law $\frac{1}{L}(\hat{A}^{(1)} - A^{(1)})$.

*Proof* It is essentially a computation where we use previous stability results, like (12), (13):

$$|\hat{H}(\hat{\mathbf{q}}) - H(\mathbf{q})| = \left| \frac{\hat{A}^{(1)} - \hat{A}^{(0)} H(\mathbf{q})}{\hat{A}^{(0)}} \right|$$

$$= \left| \frac{(A^{(1)} - A^{(0)} H(\mathbf{q}))/L + \bar{Z}_L^{(1)} - \bar{Z}_L^{(0)} H(\mathbf{q})}{A^{(0)}/L + \bar{Z}_L^{(0)}} \right|$$

$$\leq \left| \frac{(\mathrm{L}_H \mathrm{A}_S \delta / L + O(\delta^3)) + \mathrm{A} O(\delta^2) H(\mathbf{q})/L + \bar{Z}_L^{(1)} - \bar{Z}_L^{(0)} H(\mathbf{q})}{\mathrm{A}_S/L \left( 1 + O(\delta^2) + L/\mathrm{A}_S \bar{Z}_L^{(0)} \right)} \right|$$

(using (12) and (13))

$$\leq \left| \mathrm{L}_H \delta + \frac{L}{\mathrm{A}_S} O(\delta^3) + \frac{\mathrm{A}}{\mathrm{A}_S} H(\mathbf{q}) O(\delta^2) + \frac{L}{\mathrm{A}_S} \left( \bar{Z}_L^{(1)} - \bar{Z}_L^{(0)} H(\mathbf{q}) \right) \right| \Big/ \left| 1 + L/\mathrm{A}_S \bar{Z}_L^{(0)} \right|$$

Now the sum of areas $\mathrm{A}$ is almost $\mathrm{A}_S$ with a precision $O(\delta^2)$, and $H(\mathbf{q})$ is bounded. It follows that the third term is negligible with respect to the first term. Secondly the quantity $\mathrm{A}_S/L$ tends to the average size of a random triangle on $S$ within the ball centered on $\hat{\mathbf{q}}$ with radius $\delta/2$. If $L$ is big enough or if triangles are chosen within the ball to cover a reasonable area, then $\mathrm{A}_S/L = \Theta(\delta^2)$. Gathering remaining terms together gives the result. □

**Studying error laws.** We provide the expectations and variances of both error laws, which comes from the linearity of the scalar product and cross product (see supplementary for the proof).

**Property 2** The error laws $\bar{Z}_L^{(0)}$ and $\bar{Z}_L^{(1)}$ have both null expectations. Their variance follows, for $C$ and $C'$ some constants:

$$\mathbb{V}\left[\bar{Z}_L^{(0)}\right] \leq \frac{C}{L} \left( (\sigma_\xi^2 \delta^2 + \sigma_\varepsilon^2) \delta^2 + \sigma_\varepsilon^2 \sigma_\xi^2 \delta^2 + \sigma_\varepsilon^4 (1 + \sigma_\xi^2) \right)$$

$$\mathbb{V}\left[\bar{Z}_L^{(1)}\right] \leq \frac{C'}{L} \left( \sigma_\xi^2 \delta^2 + \sigma_\varepsilon^2 + \sigma_\varepsilon^2 \sigma_\xi^2 + \sigma_\xi^4 \delta^2 + \sigma_\varepsilon^2 \sigma_\xi^4 \right).$$

Theorem 1 and Property 2 explain the balance between several kinds of errors. When perturbations are small, it is advantageous to take the smallest possible $\delta$ (or $K$), with the sole constraint that the average triangle size must not be too small. Picking $K$ in 15–50 leads to very accurate results. When perturbations increase, augmenting the number $L$ of triangles reduces the variance of error laws, but it requires a larger neighborhood $\delta$ (or larger $K$) to get more independent samples. Summing up, there is an optimal balance between decreasing $\delta$ and increasing $L$, which gets higher as the noise variance gets larger.

## 5. Strategies for generating triangles

As shown before, we have to generate triangles around the point of interest with a few constraints in mind. First the average size $\mathrm{A}/L$ of triangles must be of the order $\Theta(\delta^2)$. This guarantees the accuracy of the curvature estimates in expectation. Second the more independent triangles we generate, the smaller is the variance of the error. There are at most $\lfloor \frac{K}{3} \rfloor$ independent triangles within the $K$-neighborhood of the point of interest. To improve the accuracy and efficiency in practice, we present and compare here four different strategies to generate triangles within the $K$-neighborhood around a sample $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i)$. Every one of them fit reasonably in the theoretical setting, while giving better results in practice. These four strategies are illustrated on Figure 3, while comparative accuracy and timings can be found in Figure 4 and Figure 5. Let $\rho$ be the average radius to get $K$ neighbors.

**CNC-Uniform** This strategy picks randomly $L$ triangles among the possible $\binom{K}{3}$ triangles. These triangles are not perfectly independent, but this elementary variant is very close to the theoretical setting, produces good results, while its accuracy increases

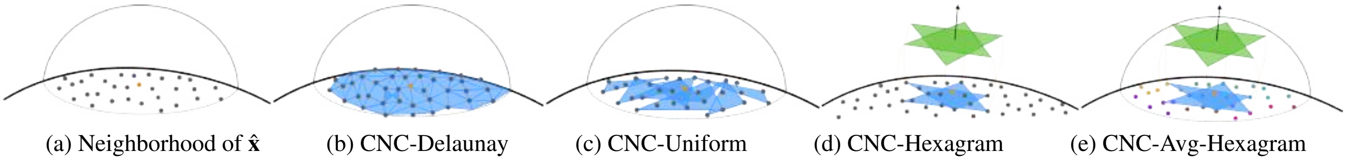
(a) Neighborhood of $\hat{\mathbf{x}}$ (b) CNC-Delaunay (c) CNC-Uniform (d) CNC-Hexagram (e) CNC-Avg-Hexagram

**Figure 3:** *This figure illustrates the different strategies for generating the random triangles around the point of interest (see text).*

when augmenting $L$. Its drawback is that its complexity per point is essentially proportional with the number $L$ of triangles.

**CNC-Delaunay** This strategy builds first around each point its local Delaunay triangulation umbrella as proposed in [SC20], and then, for the point $\hat{\mathbf{x}}$, assembles all such triangles that have vertices in the $K$-neighborhood of $\hat{\mathbf{x}}$. This variant is more alike a non-manifold local triangulated reconstruction of the cloud of points, and its accuracy is more related to the stability theorem in [LRTC20], since triangles are not really independent. It is given as a comparison basis for best possible accuracy, and its drawback is certainly its computation time.

**CNC-Hexagram** This strategy builds only two triangles forming an approximate hexagram. It builds six points $(\mathbf{p}_j)_{j=1\ldots6}$ forming a regular hexagon on the plane orthogonal to $\hat{\mathbf{u}}_i$ at distance $\rho/2$ from $\hat{\mathbf{x}}_i$. Then the closest point $\hat{\mathbf{x}}_{i_j}$ to each $\mathbf{p}_j$ is computed, and the two generated triangles are $(\hat{\mathbf{x}}_{i_1}, \hat{\mathbf{x}}_{i_3}, \hat{\mathbf{x}}_{i_5})$ and $(\hat{\mathbf{x}}_{i_2}, \hat{\mathbf{x}}_{i_4}, \hat{\mathbf{x}}_{i_6})$. Although this variant indeed satisfies that the average area is some $\Theta(\delta^2)$, it does not allow decreasing the error variance. It is the fastest variant and is quite accurate when the data present little perturbations.

**CNC-Avg-Hexagram** This strategy also builds only two triangles. However it partitioned the $K$ neighbors of $\hat{\mathbf{x}}$ according to the Voronoi regions of $(\mathbf{p}_j)_{j=1\ldots6}$ (obtained using simple nearest neighbors queries in ambient space). Within each region, positions and normal vectors are averaged to build six new (virtual) sample points $(\hat{\mathbf{y}}_j)$ with six associated normal vectors $(\hat{\mathbf{v}}_j)$. Thanks to the linearity of all curvature measure, averaging normal vectors and positions yields to better estimates than the static CNC-Hexagram approach (see supplementary). The two generated triangles are $(\hat{\mathbf{y}}_{i_1}, \hat{\mathbf{y}}_{i_3}, \hat{\mathbf{y}}_{i_5})$ and $(\hat{\mathbf{y}}_{i_2}, \hat{\mathbf{y}}_{i_4}, \hat{\mathbf{y}}_{i_6})$. This variant has a complexity proportional to $K$ but remains the fastest variant except for CNC-Hexagram. It is also the most accurate variant.

## 6. Experimental evaluation

As a first experiment, we consider point clouds sampling implicit surfaces, namely `goursat` and `torus`, on which ground-truth curvatures can be obtained (`goursat`: $0.03x^4 + 0.03y^4 + 0.03z^4 - 2x^2 - 2y^2 - 2z^2 - 8 = 0$, which fits in $[-9,9]^3$, `torus`: $(x^2 + y^2 + z^2 + 36 - 4)^2 - 144(x^2 + y^2) = 0$, with big radius 6 and small radius 2). The initial samples are obtained using a Lloyd relaxation algorithm restricted to the zero level set of the implicit functions, for various sample counts $N \in \{10000, 25000, 50000, 75000, 100000\}$. Perturbations on the samples position (resp. normal vector) are performed using a centered Gaussian perturbation parameterized by a global standard deviation $\sigma_\varepsilon \in \{0.0, 0.1, 0.2\}$ (resp. $\sigma_\xi \in \{0.0, 0.1, 0.2\}$). As a baseline for comparisons with prior works, we have considered the Monge form from jet fitting approach (named JetFitting for short) [CP05] as a representative of polynomial based fitting approach, and the recent algebraic shape operator of Lejemble *et al.* (ASO for short) [LCBM21]. Note that we use a similar experimental setting as Lejemble *et al.*, who have already shown that ASO outperforms many existing approaches from the related works (such as algebraic point set surfaces [GG07], wavejets [BDC18]). As the JetFitting approach returns absolute curvature values, intervals are clamped to zero for this estimator.

For performance evaluations, timings have been measured after the neighbors have been retrieved for the NN data structure, *i.e.* after the $K$-NN search for JetFitting, ASO, CNC-Delaunay (skipping the local Delaunay construction as well), CNU-Uniform, and CNC-Avg-Hexagram, and the six NN requests for CNC-Hexagram. Note that for all methods, we have used the same number of neighbors $K$. For the JetFitting approach, a polynomial surface of degree 4 has been considered. Timings in $\mu s$ have been measured on a single core, Apple M1. Code is available at `https://github.com/JacquesOlivierLachaud/PointCloudCurvCNC`.

### 6.1. Convergence and Stability

Figures 4 and 5 detail estimation errors for the mean and Gaussian curvatures, as the number of neighbors $K$ increases, averaged for all $N$, $\sigma_\varepsilon$ and $\sigma_\xi$. When comparing the CNC variants, the best results are obtained with CNC-Avg-Hexagram. Even if CNC-Hexagram upper bounds the estimation errors of all CNC variants (quality decreases as the noise parameters increase), this approach provides visually good results (see Figures 6 and 8) using only six NN requests, and is several order of magnitude faster than existing approaches. Compared to ASO and JetFitting, the proposed approach outperforms prior work both in terms of estimation errors and timings.

### 6.2. Additional results

Figures 7 and 8 illustrate the stability and the efficiency of our approach on point clouds from scans of historical pieces. Figure 7 displays results computed with CNC-Avg-Hexagram from 50 to 300 neighbors only ($K = 50$ for the first two objects of 500k points, $K = 300$ for the third one with 1.7M points). Overall timings (NN requests and curvature measure estimation) for these three point clouds are respectively 10.2s, 4.6s and 85.5s with a non-optimized NN data structure. Focusing on the curvature estimation only, timings are respectively 252ms, 258ms and 3991ms (2.5% to 5.6% of the total time). Figure 8 shows results computed by CNC-Hexagram, in order to evaluate the quality of the estimation using only six NN requests on a real dataset. As we increase $K$, which

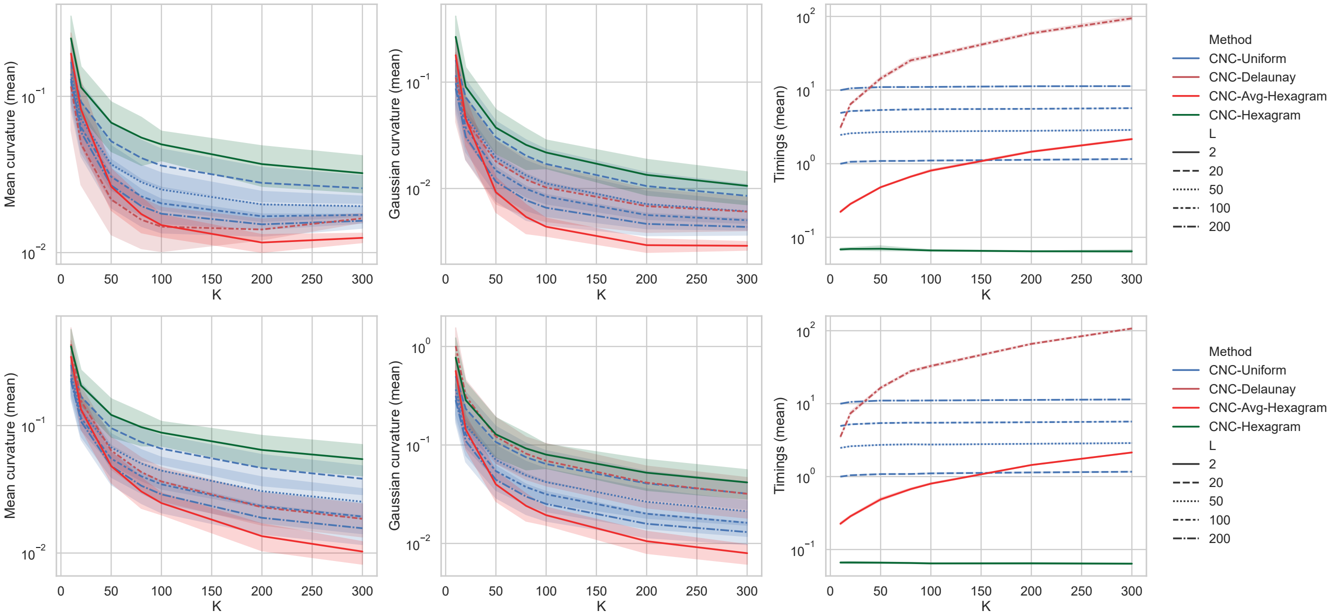

**Figure 4:** *Accuracy and timings of mean and Gaussian curvature estimation: we evaluate the accuracy and timings of CNC variants as $K$ increases for various values $L$ for the* `goursat`*, first row, and the* `torus` *shapes, second row (average and standard deviation of the error are computed from all $N$, $\varepsilon$ and $\xi$ values). Errors and timings are given in logscale and timings represent the average computation time per point in* microseconds *to estimate* all differential quantities.

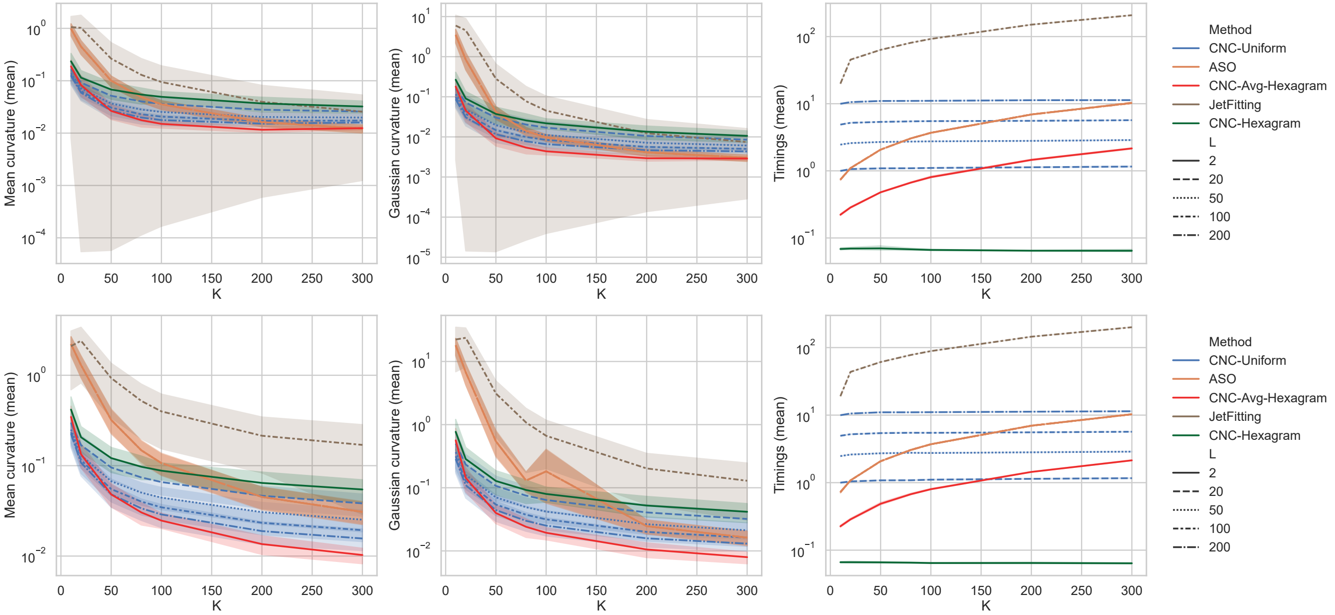

**Figure 5:** *Accuracy and timings of mean and Gaussian curvature estimation: : we compare the CNC-Uniform, the CNC-Hexagram and the CNC-Avg-Hexagram variants with two existing approaches, JetFitting [CP05] and ASO [LCBM21] (for respectively the* `goursat` *and the* `torus` *shapes). Errors and timings are given in logscale and timings represent the average computation time per point in* microseconds *to estimate* all differential quantities.

only scales the hexagram, this extremely fast variant still provides highly relevant differential quantities.

In supplementary material, we provide more results and com-

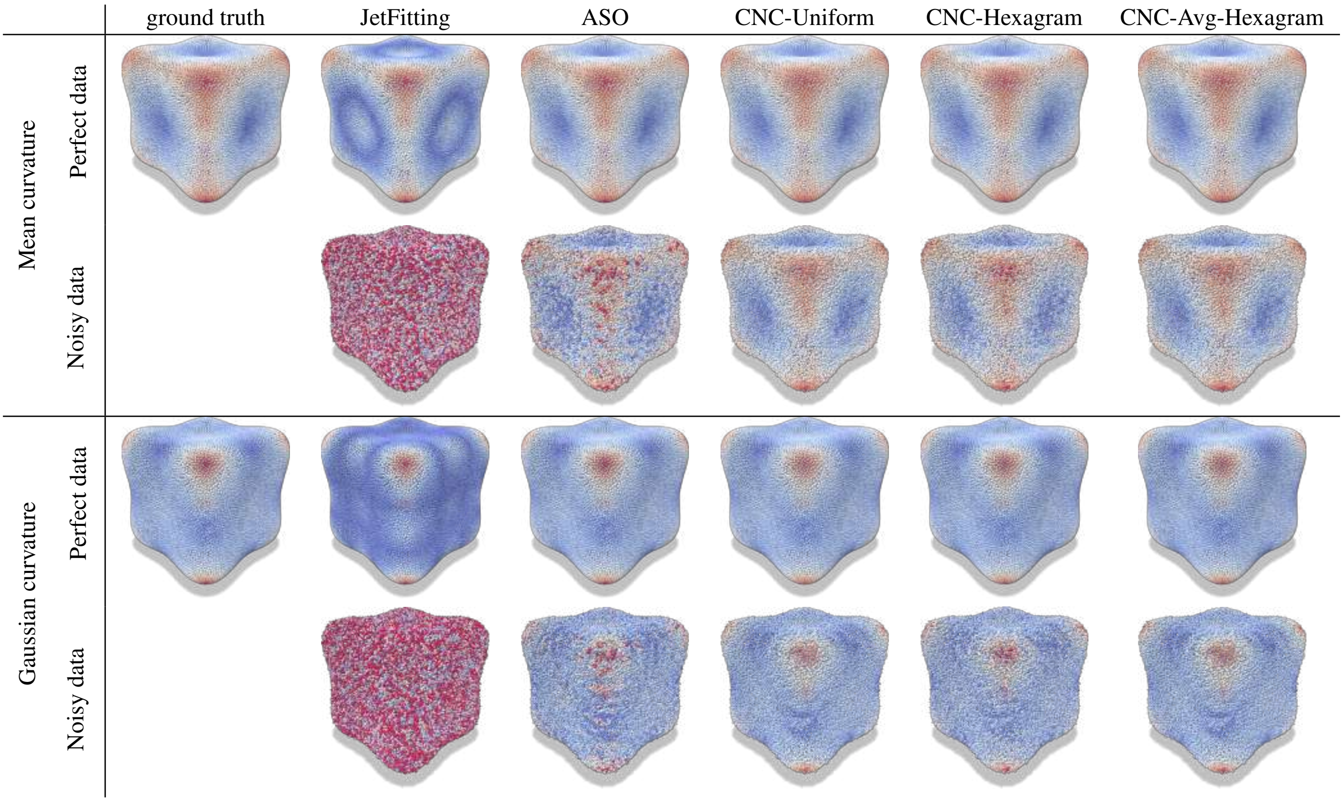

**Figure 6:** *Visual comparisons with $K = 50$ on point clouds sampling the zero level set surface of* `goursat` *($N = 25000$), without or with noise ($\sigma_\varepsilon = \sigma_\xi = 0.1$). We compare JetFitting [CP05], ASO [LCBM21], CNC-Uniform ($L = 100$), CNC-Hexagram and CNC-Avg-Hexagram. The colormap range for mean curvature (resp. Gaussian curvature) values is $[-0.107, 0.345]$ (resp. $[-0.034, 0.119]$).*

parisons (for `goursat` and `torus` with different neighborhood sizes). We also show results when the point cloud does not sample the underlying surface with uniform density (a stochastic rejection test and a highly anisotropic sampling mimicking LIDAR devices).

## 7. Discussion

We have presented a new method to estimate locally the whole curvature tensor information on oriented point clouds. We have provided quantitative theoretical guarantees of pointwise accuracy of curvature estimates, even in the presence of noise in point position *and* normal vector. In practice, our method is the most accurate and requires the smallest computation window, it is also faster and fully parallelizable, and is elementary to implement. An open-source implementation of the method is publicly available.

To sum up our experiments, JetFitting collapses as soon as tiny noise is added. Hence JetFitting is pertinent only if your data is perfect, yet remains costly to compute. ASO is experimentally more robust to a stronger noise than CNC but you must choose a big window of computation with many samples. For small to moderate noise, CNC clearly outperforms all other methods with much fewer samples (or equivalently a smaller computation window).

Currently, the main limitation of our method is that it requires oriented normals. The method itself allows reorienting normals locally with respect to the normal of the point of interest, but it may lead to output the opposite of curvatures. If there are unreliable or no normal information, our method can take as input the output of recent techniques that reorient normal vectors or even compute good oriented normal vector field for point clouds. As future works, we wish to estimate *a priori* the best radius $\delta/2$ or the best $K$ as a function of the sampling and of the noise amplitude. We would also like to relax the independence hypothesis in our theoretical guarantees, so that more triangle generation strategies would fit our setting. This would help to find better random generation techniques than CNC-Hexagram and CNC-Avg-Hexagram. Finally, the same idea could be extended to compute higher-order differential quantities. For instance, for point clouds approaching a 3D curve, one could complete the Grassmannian with another $\mathbb{R}^3$ vector space representing the tangent direction. Then differential forms approaching curvature and torsion of the curve are definable and a method similar to the one presented here would induce stable measures of 3D torsions.

## Acknowledgments

This work is supported by the French National Research Agency in the framework of the « France 2030 » program (ANR-15-IDEX-0002), by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01)

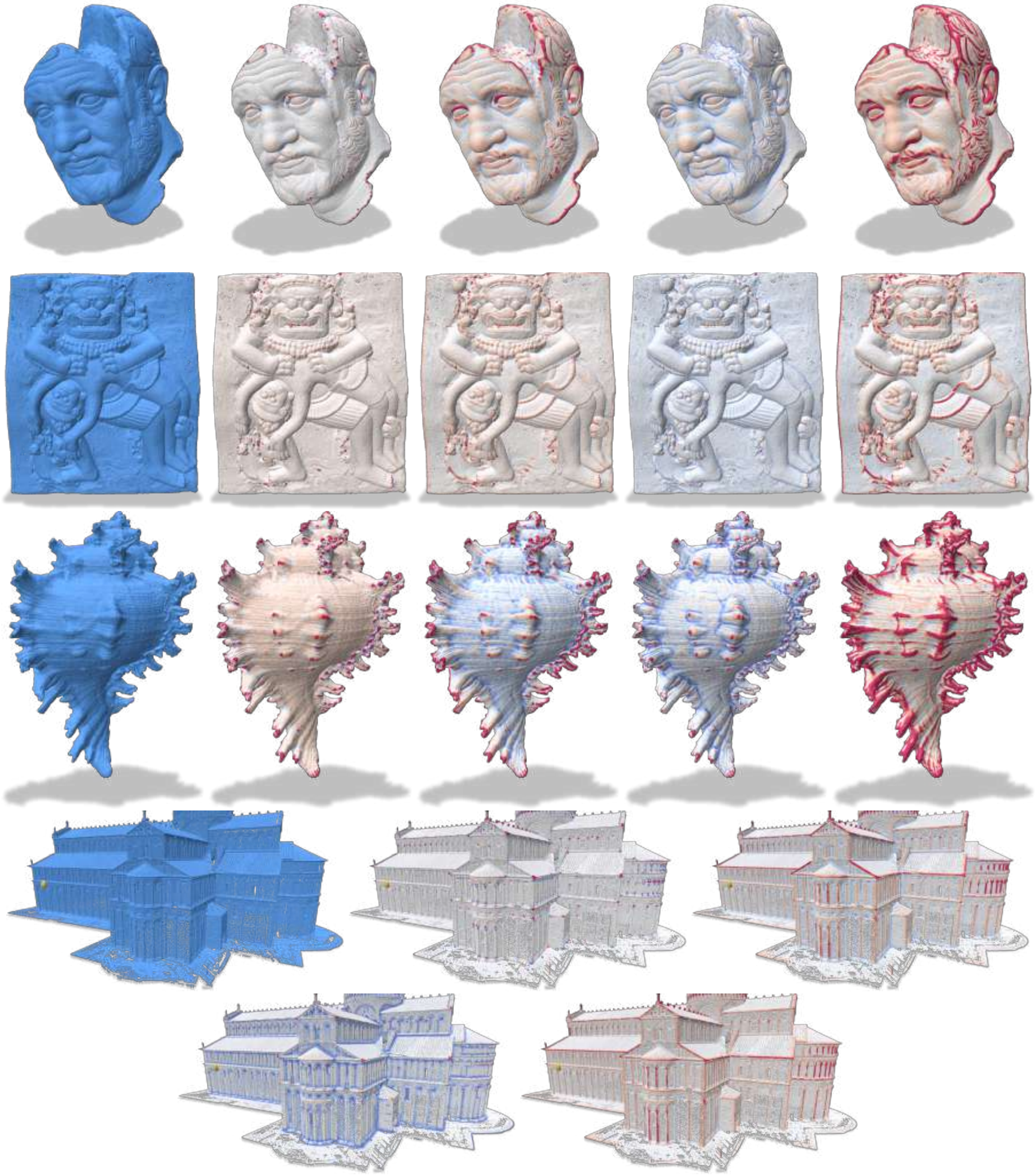

**Figure 7:** *Curvature estimations using the CNC-Avg-Hexagram approach on scanned data (from* `https://threedscans.com`*, points and normal vectors from the dense reconstructed surfaces, and filtered LIDAR data from [LCBM21]): from left to right, initial point cloud, Gaussian curvature, mean curvature, first and second principal curvatures ($K = 50$ for the first two objects —499500 and 494999 points respectively—, then $K = 300$ —1759781 points— and $K = 300$ —2506407 points— for the last two rows ).*

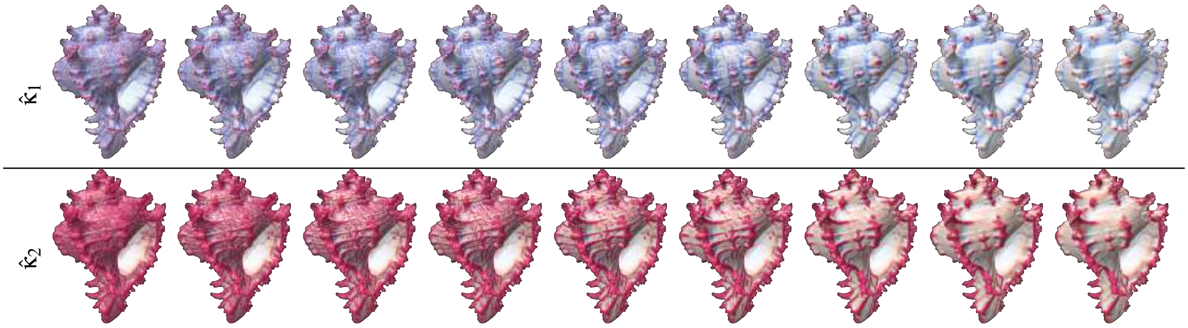

**Figure 8:** *First and second principal curvature values obtained using CNC-Hexagram for K values in* $\{20,50,80,100,200,300,1000,2000,3000\}$ *(same colormap for all figures). As the scale of the hexagram increases, large scale differential features emerge.*

and by the StableProxies project (ANR-22-CE46-0006) and gifts from Adobe Inc.

## **Appendix A:** Proofs of some properties

**Proof of area measure relation** (6).

$$|\mu^{(0)}_{\mathbf{n}}(\tau_S)-\mu^{(0)}_{\mathbf{u}}(\tau)| \leq \mathrm{Area}(\tau)\left(O(\delta^2)+\alpha\right).$$

*Proof* We recall that the Jacobian of $\pi_S$ satisfies [Fed59], for $\mathbf{y}$ in the reach of $S$:

$$\begin{aligned} J_{\pi_S}(\mathbf{y}) &= |\langle \mathbf{n}(\pi_S(\mathbf{y})) \mid \mathbf{n}_\tau\rangle|(1+O(\|\mathbf{y}-\pi_S(\mathbf{y})\|)) \\ &= \langle \mathbf{n}(\pi_S(\mathbf{y})) \mid \mathbf{n}_\tau\rangle\ (1+O(\delta^2)), \end{aligned}$$

where $\mathbf{n}_\tau$ is the unit normal to $\tau$. The last line comes from the fact that $\|\mathbf{y}-\pi_S(\mathbf{y})\| = O(\delta^2)$ and $\langle \mathbf{n}(\pi_S(\mathbf{y})) \mid \mathbf{n}_\tau\rangle > 0$. Since $\mu^{(0)}_{\mathbf{n}}$ is the area (cf. (4)), we also get by the change of variable formula

$$\mu^{(0)}_{\mathbf{n}}(\tau_S) = \int_{\tau_S} \mathrm{d}\mathbf{a} = \int_\tau J_{\pi_S}\mathrm{d}\mathbf{a} = \int_\tau \langle \mathbf{n}(\pi_S(\mathbf{y})) \mid \mathbf{n}_\tau\rangle\,(1+O(\delta^2))\mathrm{d}\mathbf{a}.$$

From the relation $\Gamma^*_{\mathbf{u}}\omega^{(0)} = \langle \mathbf{u} \mid \mathbf{n}_\tau\rangle\,\mathrm{d}\mathbf{a}$ which is proved in [LRT22], we also have

$$\mu^{(0)}_{\mathbf{u}}(\tau) = \int_\tau \Gamma^*_{\mathbf{u}}\omega^{(0)} = \int_\tau \langle \mathbf{u}(\mathbf{y}) \mid \mathbf{n}_\tau\rangle\,\mathrm{d}\mathbf{a},$$

which implies

$$\begin{aligned} &\left|\mu^{(0)}_{\mathbf{n}}(\tau_S)-\mu^{(0)}_{\mathbf{u}}(\tau)\right| \\ &\qquad =\left|\int_\tau \left\langle \mathbf{n}_\tau \mid \left[\mathbf{u}(\mathbf{y})-\mathbf{n}(\pi_S(\mathbf{y}))(1+O(\delta^2))\right]\right\rangle \mathrm{d}\mathbf{a}\right| \\ &\qquad \leq \mathrm{Area}(\tau)\sup_{\mathbf{y}\in\tau}\left(\|\mathbf{u}(\mathbf{y})-\mathbf{n}(\pi_S(\mathbf{y}))\|+O(\delta^2)\right) \\ &\qquad \leq \mathrm{Area}(\tau)\ (\alpha\ +O(\delta^2)). \end{aligned}$$

□

**Proof of mean curvature measure relation** (7)

$$|\mu^{(1)}_{\mathbf{n}}(\tau_S)-\mu^{(1)}_{\mathbf{u}}(\tau)| \leq O(\mathrm{L}_{\mathbf{n}}\delta^4)+O(\beta\delta^2)+O(\mathrm{L}_{\mathbf{u}}\alpha\delta^2).$$

*Proof* Let $\Delta$ be the standard 2-simplex and, for $(s,t)\in\Delta$, let $\mathbf{x}(s,t) := \mathbf{x}_i + s(\mathbf{x}_j-\mathbf{x}_i)+t(\mathbf{x}_k-\mathbf{x}_i)$, which spans the triangle $\tau$. The vector field $\mathbf{u}$ is built by linear interpolation over $\tau$, so can be written as $\mathbf{u}\circ\mathbf{x}(s,t) := \mathbf{u}_i + s(\mathbf{u}_j-\mathbf{u}_i)+t(\mathbf{u}_k-\mathbf{u}_i)$. Let finally $\mathbf{p}(s,t) := \pi_S(\mathbf{x}(s,t))$. We then proceed to express the two terms of the difference as determinants, with a parameterization $(s,t)\in\Delta$:

$$\begin{aligned} \mu^{(1)}_{\mathbf{n}}(\tau_S) &= \int_{\tau_S}\Gamma^*_{\mathbf{n}}\omega^{(1)} = \int_\tau \pi^*_S\Gamma^*_{\mathbf{n}}\omega^{(1)} = \int_\Delta \mathbf{x}^*\pi^*_S\Gamma^*_{\mathbf{n}}\omega^{(1)} \\ &= \frac{1}{2}\int_\Delta \det\left(\mathbf{n}\circ\mathbf{p}, \frac{\partial\mathbf{p}}{\partial s}, \frac{\partial\mathbf{n}\circ\mathbf{p}}{\partial t}\right) \\ &\qquad + \det\left(\mathbf{n}\circ\mathbf{p}, \frac{\partial\mathbf{n}\circ\mathbf{p}}{\partial s}, \frac{\partial\mathbf{p}}{\partial t}\right)\mathrm{d}s\,\mathrm{d}t \\ &= \frac{1}{2}\int_\Delta \left(\det\left(\mathbf{n}\circ\mathbf{p}, \mathbf{x}_j-\mathbf{x}_i, \frac{\partial\mathbf{n}}{\partial\mathbf{x}}(\mathbf{x}_k-\mathbf{x}_i)\right)\right. \\ &\qquad \left. + \det\left(\mathbf{n}\circ\mathbf{p}, \frac{\partial\mathbf{n}}{\partial\mathbf{x}}(\mathbf{x}_j-\mathbf{x}_i), \mathbf{x}_k-\mathbf{x}_i\right)\right)(1+O(\delta^2))\mathrm{d}s\,\mathrm{d}t. \\ \mu^{(1)}_{\mathbf{u}}(\tau) &= \int_\tau \Gamma^*_{\mathbf{u}}\omega^{(1)} \\ &= \frac{1}{2}\int_\Delta \det\left(\mathbf{u}, \frac{\partial\mathbf{x}}{\partial s}, \frac{\partial\mathbf{u}}{\partial t}\right) + \det\left(\mathbf{u}, \frac{\partial\mathbf{u}}{\partial s}, \frac{\partial\mathbf{x}}{\partial t}\right)\mathrm{d}s\,\mathrm{d}t \\ &= \frac{1}{2}\int_\Delta \det\left(\mathbf{u}, \mathbf{x}_j-\mathbf{x}_i, \mathbf{u}_k-\mathbf{u}_i\right) \\ &\qquad + \det\left(\mathbf{u}, \mathbf{u}_j-\mathbf{u}_i, \mathbf{x}_k-\mathbf{x}_i\right)\mathrm{d}s\,\mathrm{d}t. \end{aligned}$$

Let us bound the difference of the two measures as we did above, bounding $\|\frac{\partial\mathbf{n}}{\partial\mathbf{x}}(\mathbf{x}_j-\mathbf{x}_i)\|$ by $\mathrm{L}_{\mathbf{n}}\delta$ :

$$\begin{aligned} &|\mu^{(1)}_{\mathbf{n}}(\tau_S)-\mu^{(1)}_{\mathbf{u}}(\tau)| \\ &\quad \leq O(\mathrm{L}_{\mathbf{n}}\delta^4) \\ &\qquad + \int_\Delta \left|\det\left(\mathbf{n}\circ\mathbf{p}, \mathbf{x}_j-\mathbf{x}_i, \frac{\partial\mathbf{n}}{\partial\mathbf{x}}(\mathbf{x}_k-\mathbf{x}_i)-(\mathbf{u}_k-\mathbf{u}_i)\right)\right. \\ &\qquad\quad + \det\left(\mathbf{n}\circ\mathbf{p}-\mathbf{u}, \mathbf{x}_j-\mathbf{x}_i, \mathbf{u}_k-\mathbf{u}_i\right) \\ &\qquad\quad + \det\left(\mathbf{n}\circ\mathbf{p}, \frac{\partial\mathbf{n}}{\partial\mathbf{x}}(\mathbf{x}_j-\mathbf{x}_i)-(\mathbf{u}_j-\mathbf{u}_i), \mathbf{x}_k-\mathbf{x}_i\right) \\ &\qquad\quad \left. + \det\left(\mathbf{n}\circ\mathbf{p}-\mathbf{u}, \mathbf{u}_j-\mathbf{u}_i, \mathbf{x}_k-\mathbf{x}_i\right)\right|\mathrm{d}s\,\mathrm{d}t. \end{aligned}$$

Denoting $\mathrm{L}_{\mathbf{u}}$ the Lipschitz constant for $\mathbf{u}$, bounding $\|\frac{\partial \mathbf{n}}{\partial \mathbf{x}}(\mathbf{x}_j-\mathbf{x}_i)-(\mathbf{u}_j-\mathbf{u}_i)\|$ by $\beta\delta$, and using triangular inequality, it follows that

$$|\mu^{(1)}_{\mathbf{n}}(\tau_S)-\mu^{(1)}_{\mathbf{u}}(\tau)| \leq O(\mathrm{L}_{\mathbf{n}}\delta^4)+O(\beta\delta^2)+O(\mathrm{L}_{\mathbf{u}}\alpha\delta^2).$$

□

**Proof of relation with pointwise mean curvature** (8)

$$|\mu^{(1)}_{\mathbf{n}}(\tau_S)-\mathrm{Area}(\tau_S)H(\mathbf{q})| \leq \mathrm{L}_H\mathrm{Area}(\tau_S)\delta.$$

*Proof* If $\mathbf{q}$ and $\mathbf{r}$ are two points of $S$, and assume $S$ is $C^3$-smooth. Let $\mathrm{L}_H$ be the Lipschitz constant of $H$, so $|H(\mathbf{r})-H(\mathbf{q})| \leq \mathrm{L}_H\|\mathbf{r}-\mathbf{q}\|$. We integrate the mean curvature on a triangle $\tau_S \subset S \cap B(\mathbf{q},\delta/2)$, using first relation (5) then (4):

$$\begin{aligned}|\mu^{(1)}_{\mathbf{n}}(\tau_S)-\mathrm{Area}(\tau_S)H(\mathbf{q})| &= \left|\int_{\tau_S} H(\mathbf{r})-H(\mathbf{q})\mathrm{d}\mathbf{r}\right|\\ &\leq \int_{\tau_S} \mathrm{L}_H\|\mathbf{r}-\mathbf{q}\|\mathrm{d}\mathbf{r}\\ &\leq \mathrm{L}_H\mathrm{Area}(\tau_S)\delta.\end{aligned}$$

□

**Proof of bounds induced by equality of vector fields at sample points** (9)

$$\alpha \leq \beta\delta+O(\delta^2), \qquad \beta \leq O(\delta), \qquad \mathrm{L}_{\mathbf{u}} \leq \mathrm{L}_{\mathbf{n}}.$$

*Proof* Let $\mathbf{y}$ be any point of triangle $\tau$, with vertices $\mathbf{x}_0,\mathbf{x}_1,\mathbf{x}_2$ (say). Since $\mathbf{n}$ and $\pi_S$ are $C^2$ and $\mathbf{u}$ is linear, we can write using Taylor expansion:

$$\begin{aligned}\mathbf{n}\circ\pi_S(\mathbf{y}) &= \mathbf{n}_0+\mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{x}_0}(\mathbf{y}-\mathbf{x}_0)+O(\|\mathbf{y}-\mathbf{x}_0\|^2)\\ \mathbf{u}(\mathbf{y}) &= \mathbf{u}_0+\mathrm{D}\mathbf{u}_{|\mathbf{x}_0}(\mathbf{y}-\mathbf{x}_0).\end{aligned}$$

Noticing that $\mathbf{u}_0=\mathbf{n}_0$, and using the linearity of differentials, it holds:

$$\mathbf{n}\circ\pi_S(\mathbf{y})-\mathbf{u}(\mathbf{y}) = \mathrm{D}(\mathbf{n}\circ\pi_S-\mathbf{u})_{|\mathbf{x}_0}(\mathbf{y}-\mathbf{x}_0)+O(\|\mathbf{y}-\mathbf{x}_0\|^2).$$

Using the definition of $\beta$ and the fact that $\|\mathbf{y}-\mathbf{x}_0\| \leq \delta$ within the triangle concludes for the first relation.

For the second relation, we proceed also by Taylor expansion at $\mathbf{x}_0$ of $\mathbf{u}$ and $\mathbf{n}\circ\pi_S$, but we apply this expansion to determine values at $\mathbf{x}_1$ and $\mathbf{x}_2$:

$$\begin{aligned}\mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{x}_0}(\mathbf{x}_1-\mathbf{x}_0) &= \mathbf{n}_1-\mathbf{n}_0+O(\delta^2),\\ \mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{x}_0}(\mathbf{x}_2-\mathbf{x}_0) &= \mathbf{n}_2-\mathbf{n}_0+O(\delta^2),\\ \mathrm{D}\mathbf{u}_{|\mathbf{x}_0}(\mathbf{x}_1-\mathbf{x}_0) &= \mathbf{u}_1-\mathbf{u}_0,\\ \mathrm{D}\mathbf{u}_{|\mathbf{x}_0}(\mathbf{x}_2-\mathbf{x}_0) &= \mathbf{u}_2-\mathbf{u}_0.\end{aligned}$$

Again using the hypothesis that $\mathbf{u}=\mathbf{n}$ at sample points, we get:

$$\begin{aligned}\mathrm{D}(\mathbf{n}\circ\pi_S-\mathbf{u})_{|\mathbf{x}_0}(\mathbf{x}_1-\mathbf{x}_0) &= O(\delta^2),\\ \mathrm{D}(\mathbf{n}\circ\pi_S-\mathbf{u})_{|\mathbf{x}_0}(\mathbf{x}_2-\mathbf{x}_0) &= O(\delta^2).\end{aligned}$$

Since the two vectors $\mathbf{x}_1-\mathbf{x}_0$ and $\mathbf{x}_2-\mathbf{x}_0$ are not colinear and are both bounded by $\delta$, the linearity of the differential induces $\|\mathrm{D}(\mathbf{n}\circ\pi_S-\mathbf{u})_{|\mathbf{x}_0}\|=O(\delta)$. Since we assume that the boundary of $S$ is $C^3$, the differential $\mathrm{D}(\mathbf{n}\circ\pi_S)$ is $C^1$ over the triangle $\tau$. The differential $\mathrm{D}\mathbf{u}$ is constant. Hence for any $\mathbf{y}\in\tau$, Taylor expansion gives:

$$\begin{aligned}\mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{y}} &= \mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{x}_0}+O(\|\mathbf{y}-\mathbf{x}_0\|)\\ \mathrm{D}(\mathbf{u})_{|\mathbf{y}} &= \mathrm{D}(\mathbf{u})_{|\mathbf{x}_0}.\end{aligned}$$

It follows that, using also triangular inequality:

$$\begin{aligned}\|\mathrm{D}(\mathbf{n}\circ\pi_S-\mathbf{u})_{|\mathbf{y}}\| \leq& \|\mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{y}}-\mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{x}_0}\|\\ &+\|\mathrm{D}(\mathbf{n}\circ\pi_S)_{|\mathbf{x}_0}-\mathrm{D}\mathbf{u}_{|\mathbf{x}_0}\|+\|\mathrm{D}\mathbf{u}_{|\mathbf{x}_0}-\mathrm{D}(\mathbf{u})_{|\mathbf{y}}\|\\ \leq& O(\delta)+O(\delta)+0,\end{aligned}$$

which proves that $\beta=O(\delta)$.

For the last relation $\mathrm{L}_{\mathbf{u}} \leq \mathrm{L}_{\mathbf{n}}$, it suffices to establish this relation within a triangle $\tau$. We have:

$$\begin{aligned}\mathrm{L}_{\mathbf{u}} &= \sup_{\mathbf{y},\mathbf{z}\in\tau}\frac{\|\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{z})\|}{\|\mathbf{y}-\mathbf{z}\|}\\ &= \max\left(\frac{\|\mathbf{u}_1-\mathbf{u}_0\|}{\|\mathbf{x}_1-\mathbf{x}_0\|},\frac{\|\mathbf{u}_2-\mathbf{u}_1\|}{\|\mathbf{x}_2-\mathbf{x}_1\|},\frac{\|\mathbf{u}_0-\mathbf{u}_2\|}{\|\mathbf{x}_0-\mathbf{x}_2\|}\right)\\ &= \max\left(\frac{\|\mathbf{n}_1-\mathbf{n}_0\|}{\|\mathbf{x}_1-\mathbf{x}_0\|},\frac{\|\mathbf{n}_2-\mathbf{n}_1\|}{\|\mathbf{x}_2-\mathbf{x}_1\|},\frac{\|\mathbf{n}_0-\mathbf{n}_2\|}{\|\mathbf{x}_0-\mathbf{x}_2\|}\right) \leq \mathrm{L}_{\mathbf{n}}.\end{aligned}$$

□

## References

[ABCO*01] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T.: Point set surfaces. In *Visualization Conference* (2001), IEEE, pp. 21–29. 2

[Alm66] Almgren F. J.: *Plateau's problem: an invitation to varifold geometry*, vol. 13. American Mathematical Soc., 1966. 2

[BDC18] Béarzi Y., Digne J., Chaine R.: Wavejets: A local frequency framework for shape details amplification. *Computer Graphics Forum 37*, 2 (2018), 13–24. doi:10.1111/cgf.13338. 2, 6

[BLM17] Buet B., Leonardi G. P., Masnou S.: A varifold approach to surface approximation. *Archive for Rational Mechanics and Analysis 226*, 2 (2017), 639–694. 3

[BLM18] Buet B., Leonardi G. P., Masnou S.: Discretization and approximation of surfaces using varifolds. *Geometric Flows 3*, 1 (2018), 28–56. 3

[BLM19] Buet B., Leonardi G. P., Masnou S.: Weak and approximate curvatures of a measure: a varifold perspective. *arXiv preprint arXiv:1904.05930* (2019). 3

[CCSLT09] Chazal F., Cohen-Steiner D., Lieutier A., Thibert B.: Stability of curvature measures. *Computer Graphics Forum 28*, 5 (2009), 1485–1496. 2

[CLMT15] Cuel L., Lachaud J.-O., Mérigot Q., Thibert B.: Robust geometry estimation using the generalized voronoi covariance measure. *SIAM Journal on Imaging Sciences 8*, 2 (2015), 1293–1314. 2

[CP05] Cazals F., Pouget M.: Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design 22*, 2 (2005), 121 – 146. doi:https://doi.org/10.1016/j.cagd.2004.09.004. 2, 6, 7, 8

[CSM03] Cohen-Steiner D., Morvan J.-M.: Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry* (2003), pp. 312–321. 2

[CSM06] Cohen-Steiner D., Morvan J.-M.: Second fundamental measure of geometric sets and local approximation of curvatures. *Journal of Differential Geometry 74*, 3 (2006), 363–394. 2, 4

[DM14] Digne J., Morel J.-M.: Numerical analysis of differential operators on raw point clouds. *Numerische Mathematik 127*, 2 (2014), 255–289. 2

[Fed59] Federer H.: Curvature measures. *Transactions of the American Mathematical Society 93*, 3 (1959), 418–491. 2, 10

[Fu93] Fu J. H. G.: Convergence of curvatures in secant approximations. *Journal of Differential Geometry 37*, 1 (1993), 177–190. 2

[Fu94] Fu J. H.: Curvature measures of subanalytic sets. *American Journal of Mathematics* (1994), 819–880. 2

[GG07] Guennebaud G., Gross M.: Algebraic point set surfaces. *ACM Trans. Graph. 26*, 3 (July 2007), 23–es. doi:10.1145/1276377.1276406. 2, 6

[GKOM18] Guerrero P., Kleiman Y., Ovsjanikov M., Mitra N. J.: Pcpnet learning localshape properties from raw point clouds. *Computer Graphics Forum 37*, 2 (2018), 75–85. doi:10.1111/cgf.13343. 2

[GMGP05] Gelfand N., Mitra N. J., Guibas L. J., Pottmann H.: Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2005), SGP '05, The Eurographics Association, p. 197–es. 1

[KG19] Khameneifar F., Ghorbani H.: On the curvature estimation for noisy point cloud data via local quadric surface fitting. *Comput.-Aided Des. Appl. 16*, 1 (2019), 140–149. 2

[KHS10] Kalogerakis E., Hertzmann A., Singh K.: Learning 3d mesh segmentation and labeling. *ACM Trans. Graph. 29*, 4 (July 2010). doi:10.1145/1778765.1778839. 1

[KLM*13] Kim V. G., Li W., Mitra N. J., Chaudhuri S., DiVerdi S., Funkhouser T.: Learning part-based templates from large collections of 3d shapes. *ACM Trans. Graph. 32*, 4 (July 2013). doi:10.1145/2461912.2461933. 1

[KSNS07] Kalogerakis E., Simari P., Nowrouzezahrai D., Singh K.: Robust statistical estimation of curvature on discretized surfaces. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2007), SGP '07, The Eurographics Association, p. 13–22. 1

[LB13] Lévy B., Bonneel N.: Variational anisotropic surface meshing with voronoi parallel linear enumeration. In *Proceedings of the 21st international meshing roundtable* (2013), Springer, pp. 349–366. 3

[LCBM21] Lejemble T., Coeurjolly D., Barthe L., Mellado N.: Stable and efficient differential estimators on oriented point clouds. *Computer Graphics Forum (Proceedings of Symposium on Geometry Processing) 40*, 5 (July 2021). doi:10.1111/cgf.14368. 2, 6, 7, 8, 9

[Lia21] Liang J.: Research and implementation of compression algorithm for large-scale point cloud data. *Academic Journal of Computing & Information Science 4*, 7 (2021), 67–72. 1

[LRT22] Lachaud J.-O., Romon P., Thibert B.: Corrected curvature measures. *Discret. Comput. Geom. 68*, 2 (2022), 477–524. URL: https://doi.org/10.1007/s00454-022-00399-4, doi:10.1007/s00454-022-00399-4. 1, 2, 10

[LRTC20] Lachaud J.-O., Romon P., Thibert B., Coeurjolly D.: Interpolated corrected curvature measures for polygonal surfaces. *Comput. Graph. Forum 39*, 5 (2020), 41–54. URL: https://doi.org/10.1111/cgf.14067, doi:10.1111/cgf.14067. 1, 2, 3, 6

[LZH*06] Lai Y.-K., Zhou Q.-Y., Hu S.-M., Wallner J., Pottmann H.: Robust feature classification and editing. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2006), 34–45. 3

[MDS15] Mellado N., Dellepiane M., Scopigno R.: Relative scale estimation and 3d registration of multi-modal geometry using growing least squares. *IEEE Transactions on Visualization and Computer Graphics 22*, 9 (2015), 2160–2173. 1

[MGB*12] Mellado N., Guennebaud G., Barla P., Reuter P., Schlick C.: Growing least squares for the analysis of manifolds in scale-space. *Computer Graphics Forum 31*, 5 (2012), 1691–1701. doi:10.1111/j.1467-8659.2012.03174.x. 2

[MOG10] Mérigot Q., Ovsjanikov M., Guibas L. J.: Voronoi-based curvature and feature estimation from point clouds. *IEEE Transactions on Visualization and Computer Graphics 17*, 6 (2010), 743–756. 2

[PCDKC20] Paiva P. V., Cogima C. K., Dezen-Kempter E., Carvalho M. A.: Historical building point cloud segmentation combining hierarchical watershed transform and curvature analysis. *Pattern Recognition Letters 135* (2020), 114–121. URL: https://www.sciencedirect.com/science/article/pii/S0167865520301306, doi:https://doi.org/10.1016/j.patrec.2020.04.010. 1

[PSH*04] Pottmann H., Steiner T., Hofer M., Haider C., Hanbury A.: The isophotic metric and its application to feature sensitive morphology on surfaces. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV 8* (2004), Springer, pp. 560–572. 3

[PWHY09] Pottmann H., Wallner J., Huang Q.-X., Yang Y.-L.: Integral invariants for robust geometry processing. *Computer Aided Geometric Design 26*, 1 (2009), 37 – 60. doi:https://doi.org/10.1016/j.cagd.2008.01.002. 2

[PWY*07] Pottmann H., Wallner J., Yang Y.-L., Lai Y.-K., Hu S.-M.: Principal curvatures from the integral invariant viewpoint. *Computer Aided Geometric Design 24*, 8 (2007), 428 – 442. Discrete Differential Geometry. doi:https://doi.org/10.1016/j.cagd.2007.07.004. 2

[QSMG17] Qi C. R., Su H., Mo K., Guibas L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 2

[SC20] Sharp N., Crane K.: A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP) 39*, 5 (2020). 6

[TGDM18] Thomas H., Goulette F., Deschaud J.-E., Marcotegui B.: Semantic classification of 3d point clouds with multiscale spherical neighborhoods. In *2018 International Conference on 3D Vision (3DV)* (2018), IEEE, pp. 390–398. 1

[Win82] Wintgen P.: Normal cycle and integral curvature for polyhedra in riemannian manifolds. *Journal of Differential geometry 21* (1982). 2

[YLHP06] Yang Y.-L., Lai Y.-K., Hu S.-M., Pottmann H.: Robust principal curvatures on multiple scales. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2006), SGP '06, The Eurographics Association, p. 223–226. 2

[YSY*22] Yu S., Sun S., Yan W., Liu G., Li X.: A method based on curvature and hierarchical strategy for dynamic point cloud compression in augmented and virtual reality system. *Sensors 22*, 3 (2022), 1262. 1

[YZLB21] Yao Z., Zhao Q., Li X., Bi Q.: Point cloud registration algorithm based on curvature feature similarity. *Measurement 177* (2021), 109274. URL: https://www.sciencedirect.com/science/article/pii/S0263224121002797, doi:https://doi.org/10.1016/j.measurement.2021.109274. 1

[Zäh87] Zähle M.: Curvatures and currents for unions of sets with positive reach. *Geometriae Dedicata 23*, 2 (1987), 155–171. 2