Submission to GMP 2018

Directional Density Measure To Intrinsically Estimate And Counteract Non-Uniformity In Point Clouds

Submission number: 1040

Abstract

With the emergence of affordable 3D scanning and printing devices, processing of large point clouds has to be performed in many applications. Several algorithms are available for surface reconstruction, smoothing, and parametrization. However, many of these require the sampling of the point cloud to be uniform or at least to be within certain controlled parameters. For nonuniformly sampled point clouds, some methods have been proposed that deal with the nonuniformity by adding additional information such as topological or hierarchical data. In this paper, we focus on point clouds sampling surfaces in $\mathbb{R}^3$. We present the notion of local directional density measure that can be intrinsically computed within the point cloud, that is without further knowledge of the geometry despite the given point samples. Specifically, we build on the work of [1] to derive a local, directed, and discrete measure for density. Furthermore, we derive another discrete and a smooth density measure and compare these three experimentally. Each of the three considered measures gives density weights to use in discretizations of operators such that these become independent of sampling uniformity. We demonstrate the effectiveness of our method on both synthetic and real world data.

1 Introduction

Point clouds arise naturally in almost all kinds of 3D acquisition processes. As early as 1985 points were recognized as both fundamental and elementary means for graphic processing, see [9]. A guiding principle of the point-based approach is the direct processing of raw scanning data without prior meshing. Based on this idea, a multitude of algorithms has been devised to work on point clouds. However, because of missing connectivity data, point clouds pose new challenges on the corresponding processing methods. For example, when sampling a 3D object by a point cloud, reconstructing the object's topology from the point cloud is in general a hard problem.

When considering point clouds sampling surfaces in $\mathbb{R}^3$, many algorithms make additional assumptions on the sampling quality. For example, Amenta et al. assume in their power crust algorithm [1] that the distance of the sample to the sampled object is bounded by the local feature size. Results that are independent of the sampling, like the *kd tree* data structure of Friedmann et al. [4] are somewhat rare. In particular, when discretizing differential geometry operators like the gradient [8], the Laplace operator [2], or the shape operator [15], implicit or explicit assumptions are made on the uniformity of the point cloud.

In this paper, we present an approach to handle nonuniform densities in point clouds. Our method works intrinsically without using additional information beside the raw point cloud. We define a local density measure from a point of the point cloud into any tangential direction. The measure can be transformed to give local weights on the neighborhood of each point. These can then be used e.g. in the operator discretizations mentioned above in order to make them robust even on nonuniform samples. Computational experiments show the effectiveness of our approach on both synthetical and real world data.

Contributions of this paper are the following:

- definition of a discrete directional density measure for point clouds
- intrinsic computation of local density without additional information
- weights for discrete operators to overcome density effects

2 Related Work

Several methods have been devised to handle nonuniformly sampled point clouds within different applied scenarios.

In the setting of surface reconstruction, the authors of [6] propose an unsigned distance function which enables them to process input data consisting solely of 3D sample positions without any normal information. Since the surface extraction does not depend on a change of sign of the implicit representation, the method is immune to noisy and nonuniformly distributed samples. A different approach is taken in [10]. Here, the authors add volumetric or prior information to global implicit surface reconstruction to eliminate the ill-posedness of nonuniformly sampled point clouds. Yet another proposal is made in [17], where a local hierarchical clustering method is used to improve the consistency of the point cloud distribution. Within a two-step process, the computational complexity of the point cloud is reduced and the remaining points are transformed into a uniform sampling. Similarly, the authors of [11] use a combination of octree data partitioning, local Delaunay tetrahedralization, and graph cut optimization for their surface reconstruction approach. They achieve processing of point density variations of more than four orders of magnitude.

For registration of point clouds with nonuniform density, the authors of [5] propose to extend registration algorithms by including topological information of the sampled surface. Thereby, they aim to compensate effects that the nonuniformity has on 3D neighborhood searches. One of the main applications of [3] is also registration. The authors use nonuniform density in order to reduce cost in storing, processing, and visualizing a large-scale point cloud. They consider a randomized resampling strategy to select a representative subset of points while preserving features depending on the application.

Considering simplification methods for point-sampled surfaces, in [13], it is described that nonuniform samplings call for more sophisticated approaches yielding higher computational overhead. Furthermore, the authors state that finding suitable global factors can be difficult for nonuniformly sampled point clouds.

The general problem of nonuniformity in point clouds is approached in [16]. Finding that analyzing methods for uniform samples cannot be easily extended to nonuniform settings, the authors present an extension of Fourier analysis to measure spectral and spatial properties of various nonuniform sample distribution. This includes in particular adaptive, anisotropic, and non-Euclidean domains. A different approach is taken in [12]. The authors use spectral analysis of manifolds to derive optimal sampling conditions for a given surface representation. However, they also find that if their method is directly used on a point cloud with a nonuniform distribution limitations on convergence and stability may arise.

In conclusion, the state-of-the-art methods either propose solutions tailored to a special application, try to deal with nonuniformity by adding additional data, or do not work reliably on nonuniform point samples at all. In the following, we present an approach that is general, does no need further information beside the point cloud itself, and can handle nonuniform samples.

3 Three Approaches to Directional Density Measures

We will now introduce three different density measures on point clouds. The first is a continuation of the work presented in [8] and builds on the covariance matrix. The second works with a unit circle on the tangent plane and the corresponding segments induced by projections of neighboring points. Finally, the third measure utilizes smooth basis functions.

For the remainder of the paper, we consider a point cloud $P = \{p_i \in \mathbb{R}^3 \mid i = 1, \dots, n\}$. Denote for each point $p_i \in P$ its neighborhood by $\mathcal{N}_i \subseteq P$. We will assume $p_i \notin \mathcal{N}_i$. Note that the following considerations are independent of the actual definition of neighborhood. Therefore, they can be applied for a combinatorial k-nearest approach as well as for Euclidean neighborhoods.

In general, it is desirable to have a neighborhood that is as symmetric as possible with regard to all directions. When using combinatorial neighborhoods with k nearest neighbors, points lying almost as close to the query point as others might be excluded simply because k neighbors have already been found. Furthermore, to prevent numerical errors, especially when dividing by the distance, we drop those points that lie at most ε_m away from the query point, where ε_m denotes the machine accuracy. We define the k-nearest neighborhood of $p_i \in P$ with distance at least ε_m by

$$\tilde{\mathcal{N}}_i = \{p_j \in P \mid j = 1, \dots, k,\ p_j \text{ the } k\text{-nearest neighbors to } p_i \text{ with } \varepsilon_m < \|p_i - p_j\|\}.$$

This neighborhood is then relaxed to

$$\mathcal{N}_i = \tilde{\mathcal{N}}_i \cup \{p_j \in P \mid \operatorname{abs}(\|p_i - p_j\| - \max_{p_\ell \in \tilde{\mathcal{N}}_i} \|p_i - p_\ell\|) \leq \varepsilon_m\}, \tag{1}$$

where abs denotes the absolute value. That is, we include all those neighbors, that did not fall into the k-nearest neighborhood by a small account. This definition of neighborhood was proposed in [7].

We denote by $C_i \in \mathbb{R}^{3\times 3}$ the covariance matrix of the set $\mathcal{N}_i$, given by

$$C_i = \sum_{q \in \mathcal{N}_i \cup \{p_i\}} (q - \bar{b})(q - \bar{b})^T, \qquad \bar{b} = \frac{1}{|\mathcal{N}_i| + 1} \sum_{q \in \mathcal{N}_i \cup \{p_i\}} q$$

with $\bar{b}$ the barycenter. Furthermore, we identify its eigenvalues by $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$. Note that the covariance matrix is positive semidefinite, therefore all eigenvalues are nonnegative. We assume $\alpha_i \geq \beta_i \geq \gamma_i$ and call the corresponding eigenvectors $v_i, w_i, n_i \in \mathbb{R}^3$, with $\|v_i\| = \|w_i\| = \|n_i\| = 1$.

The covariance matrix induces a tangent plane T_i at the neighborhood $\mathcal{N}_i$ with distance r to a center point b and normal vector n by minimizing the least squares energy

$$E(n, r) = \sum_{q \in \mathcal{N}_i \cup \{p_i\}} (\langle q - b, n\rangle - r)^2.$$

The minimum is obtained for $b = \bar{b}$, n the eigenvector n_i of C_i corresponding to the smallest eigenvalue γ_i, and $r = 0$, see [14, Section 2.1.2]. As v_i and w_i are of norm 1 and orthogonal, they form an orthonormal basis for T_i.

The final goal is to define weights on all neighboring samples $p_j \in \mathcal{N}_i$ of a point $p_i \in P$. These weights can then be used in any discretization scheme, e.g. for the gradient [8], the Laplace operator [2], or the shape operator [15]. A first expectation on the weights is nonnegativity. Secondly, as a test case with known ground truth for our density measures, we consider the neighborhood $\mathcal{N}_i$ to be the vertices of a regular $|\mathcal{N}_i|$-gon, see Figure 1. In this regularly sampled case, each weight should be roughly 1. Therefore, we normalize all measures to fit these two expectations.

3.1 Covariance Matrix Densities

In [8], the authors present an approach to estimate the density of a point cloud P in a given direction from a point $p_i \in P$. They propose to use the eigenvalues and the eigenvectors of the covariance matrix built on a neighborhood $\mathcal{N}_i$ of p_i. Recall that the two eigenvectors v_i, w_i of the covariance matrix corresponding to the larger two eigenvalues α_i, β_i form an orthonormal basis of T_i. Therefore, any unit direction on T_i can be parametrized by $\varphi \in [0, 2\pi)$ as

$$e_\varphi = \cos(\varphi) v_i + \sin(\varphi) w_i \in T_i \tag{2}$$

with $\|e_\varphi\| = 1$. The density $\delta(e_\varphi)$ of P at p_i in direction e_φ is approximated using the quadratic form

$$\delta(e_\varphi) = \delta_i^1 \cos^2(\varphi) + \delta_i^2 \sin^2(\varphi) \tag{3}$$

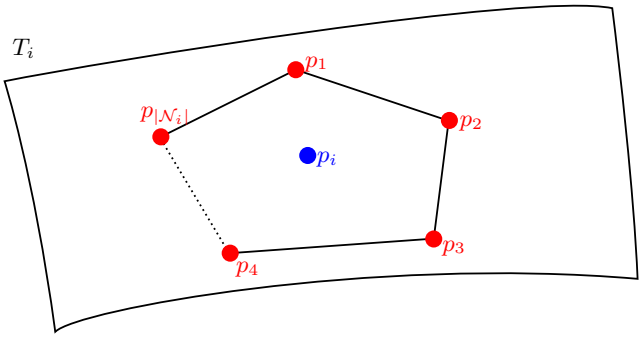


Figure 1: Neighborhood of a point p_i on T_i given as a regular $|\mathcal{N}_i|$-gon.

and the integral form of the tangential part of the diagonalized covariance matrix

$$\begin{pmatrix} \alpha_i & 0 \\ 0 & \beta_i \end{pmatrix} = \frac{1}{2\pi} \int_0^{2\pi} \delta(e_\varphi) e_\varphi e_\varphi^t \, d\varphi. \tag{4}$$

In particular, we obtain $\delta(v_i) = \delta_i^1$ for direction $e_0 = v_i$ and $\delta(w_i) = \delta_i^2$ for direction $e_{\pi/2} = w_i$. Inserting the quadratic form (3) into the integral (4) and computing the integral component-by-component, δ_i^1 and δ_i^2 can be expressed in terms of the eigenvalues α_i and β_i by

$$\delta_i^1 = 3\alpha_i - \beta_i, \qquad \delta_i^2 = 3\beta_i - \alpha_i. \tag{5}$$

The corresponding calculations are given in the appendix. Since the eigenvectors v_i, w_i form an orthonormal basis of the tangent space, we have $\langle e_\varphi, v_i \rangle = \cos(\varphi)$ and $\langle e_\varphi, w_i \rangle = \sin(\varphi)$. Plugging these expressions and the equalities (5) into (3), we obtain the directed density measure

$$\delta(e_\varphi) = (3\alpha_i - \beta_i)\langle e_\varphi, v_i \rangle^2 + (3\beta_i - \alpha_i)\langle e_\varphi, w_i \rangle^2.$$

For a point $p_i \in P$, the density in direction of a neighbor $p_j \in \mathcal{N}_i$ can then be computed as

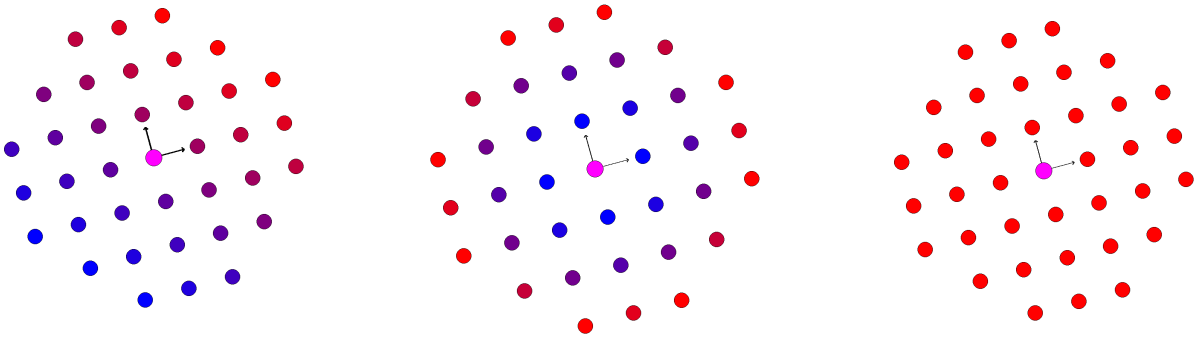
$$\delta(e_{ij}^{\tan}) = (3\alpha_i - \beta_i)\langle e_{ij}^{\tan}, v_i \rangle^2 + (3\beta_i - \alpha_i)\langle e_{ij}^{\tan}, w_i \rangle^2, \tag{6}$$

where $e_{ij}^{\tan}$ denotes the tangential part of the vector $e_{ij} = p_j - p_i$. Note that this expression is not strictly dependent on the direction, as the scalar products obtain different values with varying length of $e_{ij}^{\tan}$. Therefore, we normalize the argument and obtain

$$\bar{\delta}(e_{ij}^{\tan}) = (3\alpha_i - \beta_i)\langle e_{ij}^{\tan} / \left\| e_{ij}^{\tan} \right\|, v_i \rangle^2 + (3\beta_i - \alpha_i)\langle e_{ij}^{\tan} / \left\| e_{ij}^{\tan} \right\|, w_i \rangle^2. \tag{7}$$

In [8], the authors give the density measure slightly differently, namely as

$$\delta^{LP}(e_{ij}^{\tan}) = \frac{3\alpha_i - \beta_i}{2} \langle e_{ij}^{\tan} / \left\| e_{ij}^{\tan} \right\|, v_i \rangle + \frac{3\beta_i - \alpha_i}{2} \langle e_{ij}^{\tan} / \left\| e_{ij}^{\tan} \right\|, w_i \rangle. \tag{8}$$



(a) Density δ^{LP} given in (8) evaluated around point p_i on a uniform sampling. The density does not reflect the uniform sampling but assigns different weights to the neighbors.

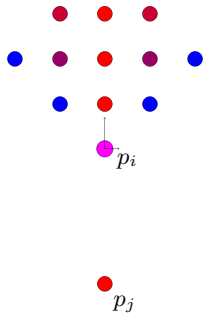
(b) Density δ given in (6) evaluated around point p_i on a uniform sampling. Note how the different distances affect the density measure.

(c) Density $\bar{\delta}$ given in (7) evaluated around point p_i on a uniform sampling. Note how all points are assigned similar values.

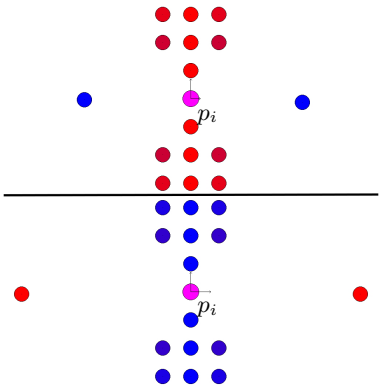
Figure 2: Density measures δ^{LP} (8) and δ with both regular (6) and normalized input (7) on the same, uniformly sampled point cloud. Values of δ^{LP} and δ ranging from low (blue) to high (red). Note how δ^{LP} and δ as in (6) assign varying density values although the neighborhood is very uniformly sampled, while δ as in (7) does assign equal density values.

This measure obtains both positive and negative values, which is not suitable to be used as weights. In particular, even for quite uniformly sampled neighborhoods, the points in the neighborhood are not attributed equal weights, see Figure 2a. Therefore, we assume that the authors of [8] meant to give (6) as density measure, which results from the above computations and which assigns equal weights to uniformly sampled neighborhoods, see Figure 2b for regular and Figure 2c for normalized input.

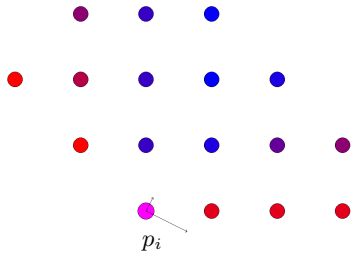
Note that expression (6) is symmetric with respect to its argument, that is, $\delta(e_{ij}^{\tan}) = \delta(-e_{ij}^{\tan})$. Therefore, a point $p_j \in \mathcal{N}_p$ in direction $e_{ij}^{\tan}$ that lies in a very sparsely sampled area might still get assigned high density weight if points on the opposite side $-e_{ij}^{\tan}$ of p_i are sampled densely, see Figure 3a. Furthermore, the density weight is sensitive to variations in the neighborhood. Moving one neighbor $p_j \in \mathcal{N}_i$ far away from p_i, the direction of the first principal component changes and thereby also the density measures assigned to the neighborhood. Consider the situation shown in Figure 3b, where one outlier drastically changes the density values. Finally, if the point p_i is not located close to the barycenter of $\mathcal{N}_i$, the assigned weights are not necessarily accurate, as given in Figure 3c.



(a) The point $p_j \in \mathcal{N}_i$ lies in a sparsely sampled area. However, the density measure δ still assigns a high value to p_j as there is a densely sampled area on the other side of p_i, in direction $-e_{ij}$.



(b) Density δ evaluated around point p_i on two slightly varying neighborhoods. While the upper neighborhood is assigned expected values, the assignment in the lower neighborhood is off because of slight variations.



(c) Density δ given in (6) evaluated around point p_i not being close to the barycenter of the given neighborhood. Assigned density values are large on the sparse points to the ends of the neighborhood and small in the sole densely sampled area.

Figure 3: Problems of the density δ given in (6) because of symmetry, outliers, and placement of the neighborhood. Densities computed from δ ranging from high (red) to low (blue).

Despite the shortcomings of δ mentioned above, it could also evaluate to a negative value. Namely, if $\alpha_i > 3\beta_i$ and $e_{ij}^{\tan}$ close to v_i, i.e. $\langle e_{ij}^{\tan}, v_i \rangle^2 \approx 1$, we get $\delta(e_{ij}^{\tan}) < 0$. As the weights are required to be positive, the values of δ have to be shifted to a strictly positive interval. We set

$$\delta^{\mathrm{Cov}}(e_{ij}^{\tan}) := \bar{\delta}(e_{ij}^{\tan}) + \max_{p_j \in \mathcal{N}_i} \{-\bar{\delta}(e_{ij}^{\tan}), 0\} \tag{9}$$

and obtain the first discrete density measure δ^{Cov}. In order to have a measure which gives large values for sparse directions and low values for dense directions, we reflect the values at their arithmetic mean

$$m_i^{\mathrm{Cov}} = \sum_{p_k \in \mathcal{N}_i} \delta^{\mathrm{Cov}}(e_{ik}^{\tan})/|\mathcal{N}_i|$$

and normalize to obtain a value between 0 and $|\mathcal{N}_i|$:

$$\sigma^{\mathrm{Cov}}(e_{ij}^{\tan}) = \frac{|\mathcal{N}_i| \left(2m_i^{\mathrm{Cov}} - \delta^{\mathrm{Cov}}(e_{ij}^{\tan})\right)}{\sum_{p_\ell \in \mathcal{N}_i} \left(2m_i^{\mathrm{Cov}} - \delta^{\mathrm{Cov}}(e_{i\ell}^{\tan})\right)} = 2 - \frac{\delta^{\mathrm{Cov}}(e_{ij}^{\tan})}{m_i^{\mathrm{Cov}}}. \tag{10}$$

In particular, if all weights are approximately equal, i.e. $\delta^{\mathrm{Cov}}(e_{ij}^{\tan}) \sim m_i^{\mathrm{Cov}}$ on a regular $|\mathcal{N}_i|$-gon, we have $\sigma^{\mathrm{Cov}}(e_{ij}^{\tan}) \sim 1$ for all $p_j \in \mathcal{N}_i$.

3.2 Arc Length Density

In this section, we will derive a discrete density measure that does not suffer from the disadvantages of δ^{Cov} as listed above. It was proposed in [7]. Given a point $p_i \in P$ and its neighborhood $\mathcal{N}_i$, we first project all neighbors $p_j \in \mathcal{N}_i$ as well as p_i to the tangent plane T_i and obtain their projections $\bar{p}_j, \bar{p}_i \in T_i$. The projected neighbors are then projected once more onto the circle of radius $r = 1$ around $\bar{p}_i$ on T_i, creating $\tilde{p}_j \in T_i$ with $\|\tilde{p}_j - \bar{p}_i\| = 1$ for all $p_j \in \mathcal{N}_i$, see Figures 4a and 4b.

Given an orientation of the tangent plane T_i by the normal at p_i, the points $\tilde{p}_j$ can be ordered along the unit circle around p_i by their angle φ_j. If $\varphi_j = \varphi_\ell$ for two points p_j, p_ℓ, we order by their indices j, ℓ. We set $\varphi(v_1) = 0$. Denote the order by $\tilde{p}_{j_1}, \ldots, \tilde{p}_{j_{|\mathcal{N}_i|}}$. For any point $\tilde{p}_{j_\ell}$, $\ell \in \{1, \ldots, |\mathcal{N}_i|\}$, consider the arc length on the unit circle from $\tilde{p}_{j_{\ell-1}}$ to $\tilde{p}_{j_\ell}$ and from $\tilde{p}_{j_\ell}$ to $\tilde{p}_{j_{\ell+1}}$, see Figure 4b.

The main idea for the density measure is now to assume a direction $e_{ij}^{\tan}$ to point into a dense area, if the arcs to the two neighbors of $\tilde{p}_j$ are short compared to the longest possible arc length 2π. The second discrete measure is then given by the sum of half the lengths of the adjacent arcs. For a point $p_j \in \mathcal{N}_i$ with projection $\tilde{p}_{j_\ell}$ we define

$$\delta^{\text{Arc}}(e_{ij}^{tan}) := \frac{\sphericalangle(\tilde{p}_{j_{\ell-1}} - \bar{p}_i, \tilde{p}_{j_\ell} - \bar{p}_i)}{2} + \frac{\sphericalangle(\tilde{p}_{j_\ell} - \bar{p}_i, \tilde{p}_{j_{\ell+1}} - \bar{p}_i)}{2} \tag{11}$$

with the angle $\sphericalangle$ between two vectors given in radians. Recall that the length a of an arc of angle $\phi \in [0, 2\pi]$ is given by $a = r \cdot \phi$, but since we project to the unit circle, this reduces to $a = \phi$.

Although this definition does not suffer from the problematic symmetry as δ^{Cov}, there is still a slight inconvenience. Consider a densely sampled region neighboring a sparsely sampled region as shown in Figure 4c. The circle arc separating the sparse point and the dense region is solely contributing to the density measure of the border point of the dense region, assigning it a significantly higher value than the other points of the dense region.

As in Section 3.1, we will now normalize the measure (11) in order to obtain positive weights that are about 1 on a regular $|\mathcal{N}_i|$-gon. Therefore, we set

$$\sigma^{\text{Arc}}(e_{ij}^{\tan}) = \frac{\delta^{\text{Arc}}(e_{ij}^{\tan})|\mathcal{N}_i|}{\sum_{p_k \in \mathcal{N}_i} \delta^{\text{Arc}}(e_{ij}^{\tan})}. \tag{12}$$

As in (10), for equal weights in our test case of the regular $|\mathcal{N}_i|$-gon, we have $\sigma^{\text{Arc}}(e_{ij}^{\tan}) \sim 1$ for all $p_j \in \mathcal{N}_i$.

3.3 Smooth Basis Density

Having presented two discrete density measures δ^{Cov} and δ^{Arc} above, we will now present a smooth density measure, based on the discrete data given. The first steps are the same as in Section 3.2. That is, we consider the projection $\tilde{p}_j$ of all points $p_j \in \mathcal{N}_i$ to a unit circle on T_i around p_i, see Figure 4. As before, each point $\tilde{p}_j$ is assigned an angle φ_j with $\varphi(v_1) = 0$. The main idea is now to define a basis function on each $\tilde{p}_j$ and thereby obtain a density measure at any angle $\varphi \in [0, 2\pi)$ by

$$\delta^{\text{RBF}} : [0, 2\pi) \to \mathbb{R}_{\geq 0}, \qquad \delta^{\text{RBF}}(\varphi) = \frac{1}{|\mathcal{N}_i|} \sum_{p_j \in \mathcal{N}_i} \psi_j(\varphi). \tag{13}$$

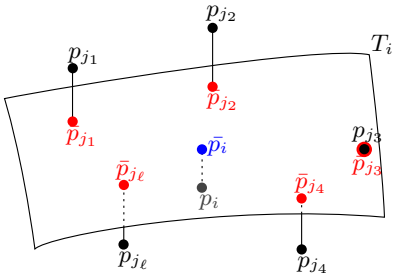
The basis function ψ_j should be of finite local support to only influence a small neighborhood around the sample point p_j. It should furthermore be smooth to obtain a smooth density measure δ^{RBF}. Finally, the size of the local support should be chosen such that uniformly distributed samples p_j lead to a preferably uniform density. Therefore, we propose the bump function as local basis

$$\psi_j : [0, 2\pi) \to \mathbb{R}_{\geq 0}, \qquad \psi_j(\varphi) = \begin{cases} \exp\left(\frac{1}{r_i^{-2}(\varphi - \varphi_j)^2 - 1}\right) & \varphi \in (-r_i, r_i), \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

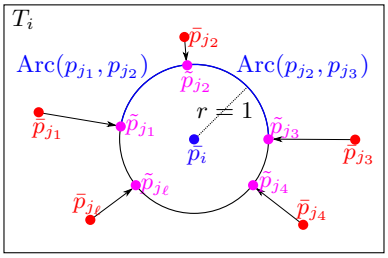
with $r_i = \frac{2\pi}{|\mathcal{N}_i|}$. Note that ψ_j is $\mathcal{C}^\infty$ and of finite local support. See Figure 5a for an corresponding illustration of equation (13) and see Figure 5b for a plot of the resulting density measure on a uniform point sample.

Once more, as in (10) and (12), we will normalize the values of δ^{RBF} to have density weights of approximately 1 on the regular $|\mathcal{N}_i|$-gon. Therefore, we set

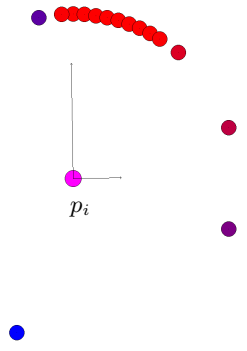
$$\sigma^{\text{RBF}}(e_{ij}^{\tan}) = \frac{\delta^{\text{RBF}}(e_{ij}^{\tan})|\mathcal{N}_i|}{\sum_{p_k \in \mathcal{N}_i} \delta^{\text{RBF}}(e_{ij}^{\tan})}. \tag{15}$$



(a) Points $p_j \in N_i$ and the center point p_i are projected to the tangent plane T_i.



(b) Points $\bar{p}_j$ are projected onto the unit circle around $\bar{p}_i$ on T_i. For each point p_j, its density is computed as half of the sum of the arc lengths on the circle starting at its projection $\tilde{p}_j$.



(c) Note how the leftmost point in the upper region gets a different weight than the other points. Values ranging from high density (red) to low density (blue).

Figure 4: Illustration of the arc length density. Points $p_j \in N_i$ are projected to the tangent plane T_i. There, they are projected to the circle of radius 1 around p_i. For each point p_j, its density is computed as half of the lengths of the arcs on the circle starting at its projection $\tilde{p}_j$. When applied to nonuniform dense samples, the borderpoints of dense regions get ranked significantly different from the inner points of the region, as the arc to the neighboring sparse region is only distributed on the end points.

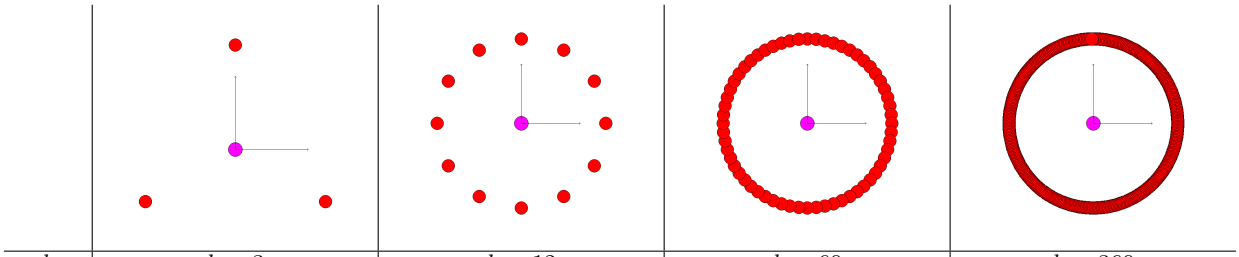
3.4 Concluding Overview

In the Sections 3.1 to 3.3 we presented three different directional density measures on point clouds. Namely, we built on a method by [8] to define a measure δ^{Cov}, (9), based on the covariance matrix. Furthermore, we projected points to an estimated tangent plane and distributed the arc lengths of the unit circle around the center point as density measure δ^{Arc}, (11). Finally, we defined smooth radial basis functions for each sample p_j and summed these up to obtain a smooth density measure δ^{RBF}, (15).

All three measures are to be used as weights in discretizations of differential geometry operators as described in the beginning of Section 3. Therefore, we normalized them to our test case: the regular $|\mathcal{N}_i|$-gon. We obtained three corresponding weights σ^{Cov}, (10), σ^{Arc}, (12), and σ^{RBF}, (15). In the next section, these different weights will be experimentally evaluated.

4 Experimental Results

In this Section, we summarize some experimental results proving the effectiveness of our approach. As a first preliminary test, we compute the different normalized density measures σ^{Cov} (10), σ^{Arc} (12), and σ^{RBF} (15) on k-neighborhoods consisting of regular k-gons. Thereby, we justify the definitions of the respective measures by showing that they obtain weights around 1 as desired. In Table 1, we presented the computed the maximal deviation from weight 1 for the three measures and different k. Note that all weights lie well within a range around 1, given a machine precision. We find that the values of δ^{Arc} seem to deviate most. However, this comes with growing values of n, where in any real application typically small values of k are used.



k	$k=3$	$k=12$	$k=60$	$k=360$
σ^{Cov}	$2 \cdot 10^{-16}$	$4 \cdot 10^{-16}$	$7 \cdot 10^{-16}$	$13 \cdot 10^{-16}$
σ^{Arc}	$1 \cdot 10^{-16}$	$7 \cdot 10^{-16}$	$197 \cdot 10^{-16}$	$7736 \cdot 10^{-16}$
σ^{RBF}	0.0	$1 \cdot 10^{-16}$	$4 \cdot 10^{-16}, -4 \cdot 10^{-16}$	$82 \cdot 10^{-16}$

Table 1: Results of density measures σ^{Cov} (10), σ^{Arc} (12), and σ^{RBF} (15) on a regular k-gon neighborhood for varying k. The numbers indicate the maximum deviation of the weights from 1.0.

Having passed this first preliminary experiment, we will benchmark our weights on a discretization of the shape operator originally proposed by Taubin in [15]. The discretization has been used on point clouds with some alterations discussed in [8, 14, 7]. The main idea is to discretize a matrix M_i at point $p_i \in P$ as

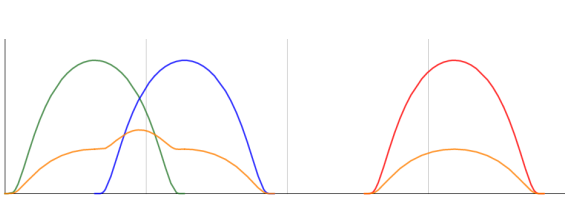
$$M_i = \frac{1}{2\pi} \sum_{p_j \in \mathcal{N}_i} \omega_{ij} \kappa_{ij} e_{ij}^{\tan} e_{ij}^{\tan t}, \tag{16}$$

where $e_{ij}^{\tan}$ is the same vector as in Section 3, ω_{ij} is a weight assigned to each point in order to have a faithful approximation, and κ_{ij} is the directional curvature for the direction $e_{ij}^{\tan}$. It is discretized as

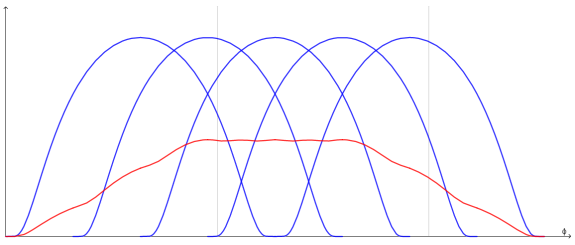
$$\kappa_{ij} = \frac{2\langle n_i, e_{ij} \rangle}{\|e_{ij}\|}.$$

With this setup, the principal curvatures κ_i^1, κ_i^2 at point $p_i \in P$ can be computed from the eigenvalues m_i^1, m_i^2 of M_i by

$$\kappa_i^1 = 3m_i^1 - m_i^2, \qquad \kappa_i^2 = 3m_i^2 - m_i^1.$$



(a) Three basis functions (14) shown in green, blue, and red, as well as the corresponding density δ^{RBF}, (13), shown in orange.



(b) Several basis functions on a set of uniform samples shown in blue. Note how the corresponding density (shown in red) becomes almost uniform in the center.

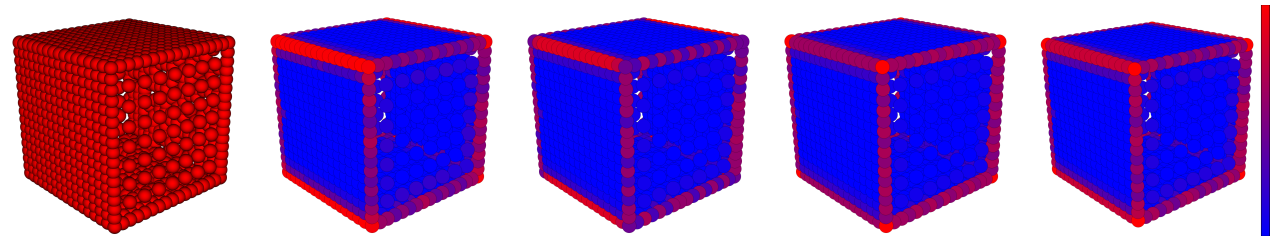
Figure 5: Plotting several basis functions (14) and the corresponding densities (13).

For the weights ω_{ij}, Taubin suggests in [15] to use

$$\omega_{ij}^{\text{Tau}} = \frac{2\pi \, \|e_{ij}\|}{\sum_{p_j \in \mathcal{N}_i} \|e_{ij}\|}, \tag{17}$$

which we will compare to our, properly adjusted weights:

$$\omega_{ij}^{\text{Cov}} = \frac{2\pi\sigma_{ij}^{\text{Cov}} \, \|e_{ij}\|}{\sum_{p_j \in \mathcal{N}_i} \|e_{ij}\|}, \qquad \omega_{ij}^{\text{Arc}} = \frac{2\pi\sigma_{ij}^{\text{Arc}} \, \|e_{ij}\|}{\sum_{p_j \in \mathcal{N}_i} \|e_{ij}\|}, \qquad \omega_{ij}^{\text{RBF}} = \frac{2\pi\sigma_{ij}^{\text{RBF}} \, \|e_{ij}\|}{\sum_{p_j \in \mathcal{N}_i} \|e_{ij}\|}. \tag{18}$$



(a) extracted point cloud (1186 samples)

(b) max. principal curvature with ω_{ij}^{Tau}

(c) max. principal curvature with ω_{ij}^{Cov}

(d) max. principal curvature with ω_{ij}^{Arc}

(e) max. principal curvature with ω_{ij}^{RBF}

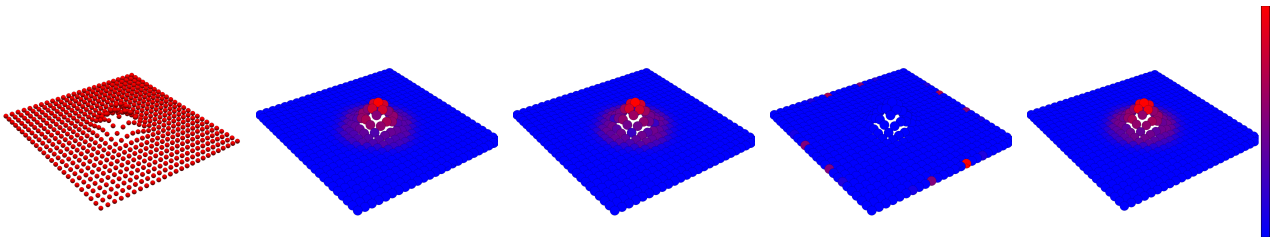
Figure 6: Plotting maximum principal curvature on a cube with differently sampled sides for weights ω_{ij} as indicated ranging from low (blue) to high (red).

4.1 Synthetic Models

We will first run some test on synthetical models before testing the weights on real world models. A first test case is a cube on which two opposing sides are sampled with only a quarter of the points on the each of the other sides. It is given as a mesh, from which we extract a point cloud (Figure 6a). On this, we test the different weights and plot the maximum principal curvature $\max(\kappa_i^1, \kappa_i^2)$ for each point $p_i \in P$. Images are created with $k = 13$ and relaxed neighborhoods as defined in Equation (1).

Note how the weights ω_{ij}^{Tau} and ω_{ij}^{Cov} assign higher values to the edge between two equally dense sampled sides than to the edges bridging two differently sampled sides of the cube. Additionally, ω_{ij}^{Cov} is not able to recover the corners of the cube as points of high curvature. The assigned curvature values on the edges are more regular for weights ω_{ij}^{Arc} and ω_{ij}^{RBF}, also the corners are marked as points of high principal curvature.

In Section 3.1 we saw that the weights ω_{ij}^{Cov} exhibit a problematic symmetrical behavior. This effect leads to an overestimate of those edges of the cube neighboring two dense regions. Furthermore, it prevents the weights ω_{ij}^{Cov} to determine the corners of the cube as points of highest curvature.



(a) extracted point cloud (1186 samples)

(b) max. principal curvature with ω_{ij}^{Tau}

(c) max. principal curvature with ω_{ij}^{Cov}

(d) max. principal curvature with ω_{ij}^{Arc}

(e) max. principal curvature with ω_{ij}^{RBF}

Figure 7: Plotting maximum principal curvature on a Gaussian bump on the plane for weights ω_{ij} as indicated ranging from low (blue) to high (red).

A second test case is a Gaussian bump on an otherwise flat plane given as a mesh. From that, we extract a point cloud (Figure 7a) on which we test the different weights and plot the maximum principal curvature $\max(\kappa_i^1, \kappa_i^2)$ for each point $p_i \in P$. Images are created with $k = 13$ and relaxed neighborhoods as defined in Equation (1). Note that in this case, weights ω_{ij}^{Tau}, ω_{ij}^{Cov}, and ω_{ij}^{RBF} perform very similar. However, the weights ω_{ij}^{Arc} are not able to recover the curvature of the bump. That is, because weights for some points on the border of the geometry become very large, as the arcs reach almost a length of π there. These throw off the curvature estimates for the whole geometry.

A third and final test on synthetic models is run on an octahedron with tangential noise added to it. The test is run only on the so far supreme weights ω_{ij}^{tau} and ω_{ij}^{RBF}. Figure 8a shows how despite the noise, weights ω_{ij}^{RBF} emphasize the edges and corners of the octahedron better than weights ω_{ij}^{tau}.

Model	#Points	Max. Princ. Curv. ω_{ij}^{Tau}	Max. Princ. Curv. ω_{ij}^{RBF}
Bearing	3,475	59.25111533960161	62.39209885818981
Dragon	50,000	2.55179467645741	2.36348617774811
Fandisk	6,475	14.14216652836357	15.01480643981332
noisy Rocker Arm	40,177	2.49300719256904	2.40278912989522
noisy Venus	17,018	22.53991528077439	23.56054013807663

Table 2: Models and values of the maximum principal curvature detected for the two weights ω_{ij}^{Tau} and ω_{ij}^{RBF}.

4.2 Real World Models

Due to the shortcomings found for ω_{ij}^{Cov} and ω_{ij}^{Arc} in Section 4.1, in the following we will concentrate on weights ω_{ij}^{Tau} and ω_{ij}^{RBF}. We will test the two weights on several real world models. For each, we evaluate the range of maximum principal curvatures detected and also give a visual evaluation of certain interesting features.

Given that the distribution of points on the curvature range is comparable, a larger detected range means for higher sensitivity to the curvature of the geometry. The detected highest maximum principal curvature values for the tested models are given in Table 2. Note that all models include points of maximum principal curvature $\max\{\kappa_i^1, \kappa_i^2\} = 0$. Therefore, the values in Table 2 give the range of maximum principal curvature for the models. Note that the curvature range of ω_{ij}^{RBF} is higher or comparable to that of ω_{ij}^{Tau}.

In Figures 8 and 9 we visually compare the results of principal maximum curvature visualization on several real world geometries. All experiments were performed with $k = 12$ neighbors and relaxed neighborhoods as defined in (1). Note that the curvature assignments with weights ω_{ij}^{RBF} emphasize features better, as highlighted in Figures 8b and 8c. Also, the values of curvature are more regular in particular along sharp features, even if curved, as highlighted in Figure 9a. The effect of noise on the curvature computation is shown in Figures 9b and 9c. In both cases, weights ω_{ij}^{RBF} are slightly more resilient in presence of noise and emphasize features better. In particular, all examples show that utilizing the additional weights, no artifacts are introduced and no curvature estimates become flawed.

5 Conclusion

In this paper, we presented three possible local density measures. Namely, we find that the density measure of [8] suffers from several shortcomings, therefore we introduce a different discrete measure and an additional smooth density measure. After deriving the measures theoretically, we prove their effectiveness on both synthetic and real world data in the context of maximum principal curvature computation. Different other applications are left for further research.

References

[1] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266. ACM, 2001.

[2] M. Belkin, J. Sun, and Y. Wang. Constructing laplace operator from point clouds in $\mathbb{R}^d$. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1031–1040. Society for Industrial and Applied Mathematics, 2009.

[3] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević. Fast resampling of 3d point clouds via graphs. *arXiv preprint arXiv:1702.06397*, 2017.

[4] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[5] D. Holz and S. Behnke. Registration of non-uniform density 3d point clouds using approximate surface reconstruction. In *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*, pages 1–7. VDE, 2014.

[6] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *Symposium on geometry processing*, pages 41–50, 2006.

[7] J. Jansen. Anisotropic smoothing and feature detection of large point clouds using principal curvatures. Master's thesis, Freie Universität Berlin, Berlin, Germany, 2017. In Cooperation with Carmeq GmbH, Berlin, Germany.

[8] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Computer Aided Design*, 22:680–692, 2005.

[9] M. Levoy and T. Whitted. The use of points as a display primitive. *Technical report, University of North Carolina, Department of Computer Science*, 1985.

[10] J. Liang, F. Park, and H. Zhao. Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation. *Journal of Scientific Computing*, 54(2-3):577–602, 2013.

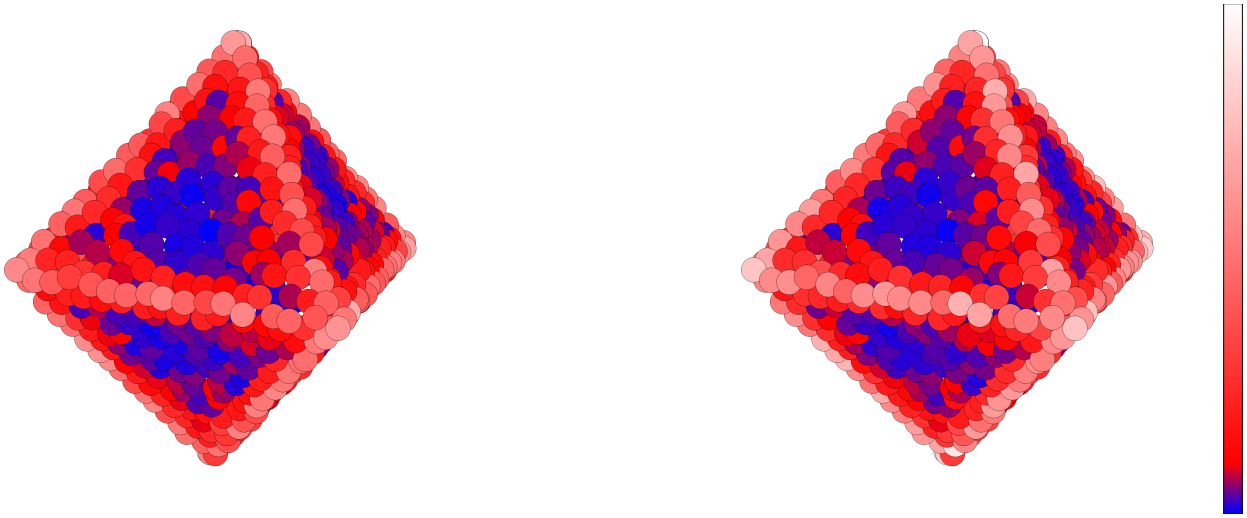
[11] C. Mostegel, R. Prettenthaler, F. Fraundorfer, and H. Bischof. Scalable surface reconstruction from point clouds with extreme scale and density diversity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 904–913, July 2017.

[12] A. C. Öztireli, M. Alexa, and M. Gross. Spectral sampling of manifolds. *ACM Transactions on Graphics (TOG)*, 29(6):168, 2010.

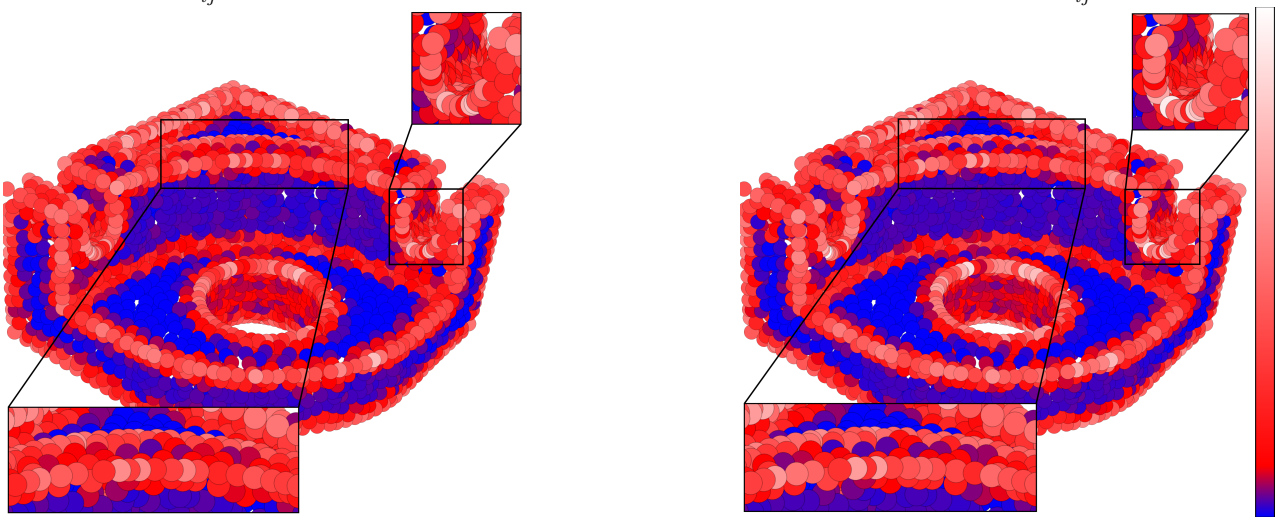
[13] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proceedings of the conference on Visualization'02*, pages 163–170. IEEE Computer Society, 2002.

[14] M. Skrodzki. Neighborhood computation of point set surfaces. Master's thesis, Freie Universität Berlin, Berlin, Germany, 2014.

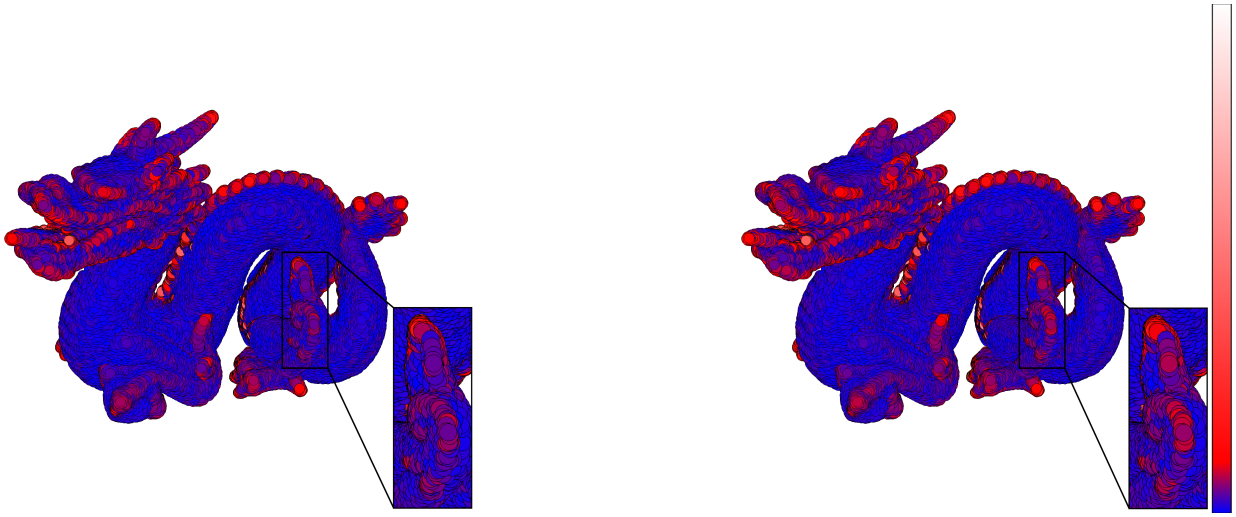
[15] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 902–907. IEEE, 1995.



(a) Maximum principal curvature on a noisy octahedron with weights ω_{ij}^{tau} on the left and ω_{ij}^{RBF} on the right. Note how despite the noise, weights ω_{ij}^{RBF} still capture the edges and corners of the octahedron more precisely than weights ω_{ij}^{tau}.

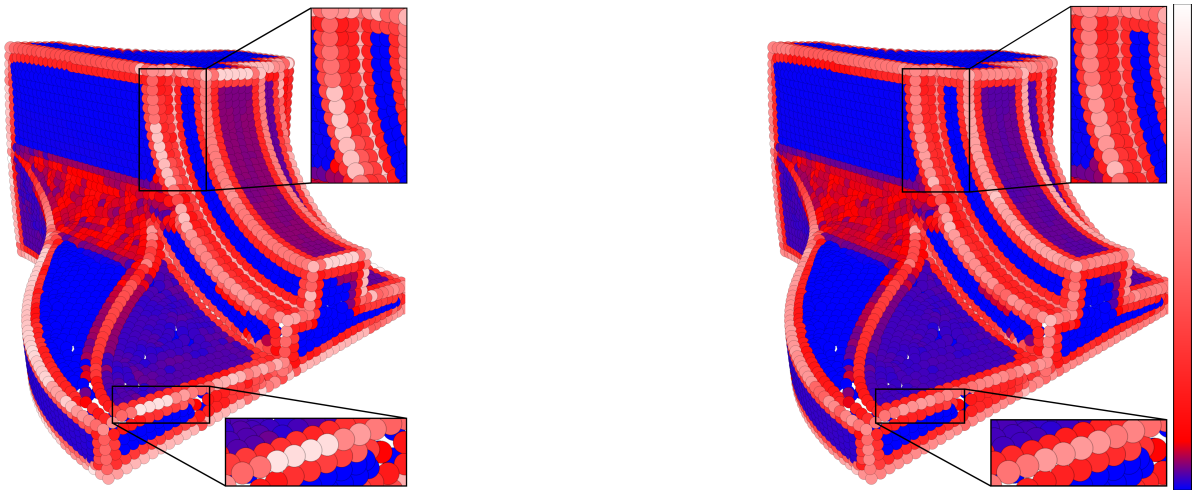


(b) Maximum principal curvature of the bearing model with weights ω_{ij}^{tau} on the left and ω_{ij}^{RBF} on the right. Note how the weights ω_{ij}^{RBF} recover more points of low curvature in the area highlighted in the lower left and assign higher curvature to the tightly curved area highlighted in the upper right.

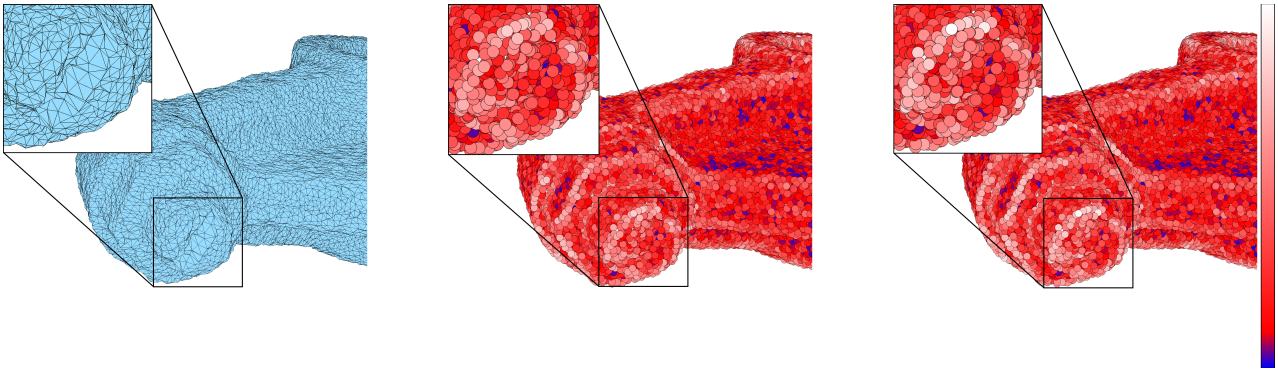


(c) Maximum principal curvature of the dragon model with weights ω_{ij}^{tau} on the left and ω_{ij}^{RBF} on the right. Note how the weights ω_{ij}^{RBF} better emphasize the curvature on the highlighted part.

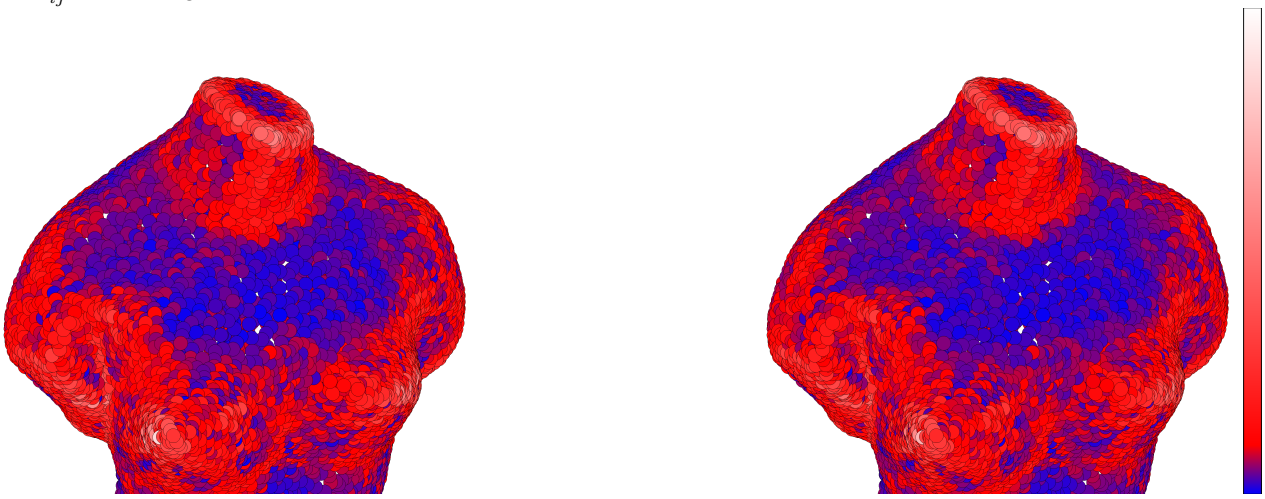
Figure 8: Comparison of weights ω_{ij}^{tau} and ω_{ij}^{RBF} on a noisy octahedron and two real world models.



(a) Maximum principal curvature of the dragon model with weights ω_{ij}^{tau} on the left and ω_{ij}^{RBF} on the right. Note how the weights ω_{ij}^{tau} create a jump in the curvature assignments on both straight and bend edges. This jump is less present when using weights ω_{ij}^{RBF}.



(b) A noisy mesh of the rocker arm model on the left colored by maximum principal curvature with weights ω_{ij}^{tau} in the center and ω_{ij}^{RBF} on the right.



(c) Maximum principal curvature of the venus model with weights ω_{ij}^{tau} on the left and ω_{ij}^{RBF} on the right.

Figure 9: Further comparison of weights ω_{ij}^{tau} and ω_{ij}^{RBF} on three more real world models. Note how the results are similiar due to mostly good sampling of the models. Despite the noise, weights ω_{ij}^{RBF} are still able to capture points of high curvature on the model (shown in white in the highlight).

[16] L.-Y. Wei and R. Wang. Differential domain analysis for non-uniform sampling. *ACM Transactions on Graphics (TOG)*, 30(4):50, 2011.

[17] J. Yang, R. Li, Y. Xiao, and Z. Cao. 3d reconstruction from non-uniform point clouds via local hierarchical clustering. In *Ninth International Conference on Digital Image Processing (ICDIP 2017)*, volume 10420, page 1042038. International Society for Optics and Photonics, 2017.

Appendix

We now have a closer look on the expression of δ_i^1, δ_i^2, given by the two largest eigenvalues α_i, β_i from (4). A unit direction on T_i can be expressed in terms of the orthonormal basis of the eigenvectors of the covariance matrix v_i, w_i. Hence,

$$e_\varphi = \begin{pmatrix}\cos(\varphi)\\ \sin(\varphi)\end{pmatrix}, \qquad \text{with} \qquad e_\varphi e_\varphi^t = \begin{pmatrix}\cos^2(\varphi) & \cos(\varphi)\sin(\varphi)\\ \cos(\varphi)\sin(\varphi) & \sin^2(\varphi)\end{pmatrix}. \tag{19}$$

for some $\varphi \in [0, 2\pi)$. We now compute the integral form of the tangential part of the diagonalized covariance matrix by using the quadratic form (3) and the matrix from (19)

$$\begin{aligned}
\begin{pmatrix}\alpha_i & 0\\ 0 & \beta_i\end{pmatrix} =& \frac{1}{2\pi}\int_0^{2\pi} \delta(e_\varphi) e_\varphi e_\varphi^t \, d\varphi \\
=& \frac{1}{2\pi}\int_0^{2\pi} \delta(e_\varphi) \begin{pmatrix}\cos^2(\varphi) & \cos(\varphi)\sin(\varphi)\\ \cos(\varphi)\sin(\varphi) & \sin^2(\varphi)\end{pmatrix} d\varphi \\
=& \frac{1}{2\pi}\delta_i^1 \int_0^{2\pi} \begin{pmatrix}\cos^4(\varphi) & \cos^3(\varphi)\sin(\varphi)\\ \cos^3(\varphi)\sin(\varphi) & \cos^2(\varphi)\sin^2(\varphi)\end{pmatrix} d\varphi + \frac{1}{2\pi}\delta_i^2 \int_0^{2\pi} \begin{pmatrix}\cos^2(\varphi)\sin^2(\varphi) & \cos(\varphi)\sin^3(\varphi)\\ \cos(\varphi)\sin^3(\varphi) & \sin^4(\varphi)\end{pmatrix} d\varphi \\
=& \frac{1}{2\pi}\delta_i^1 \begin{pmatrix}\frac{3\pi}{4} & 0\\ 0 & \frac{\pi}{4}\end{pmatrix} + \frac{1}{2\pi}\delta_i^2 \begin{pmatrix}\frac{\pi}{4} & 0\\ 0 & \frac{3\pi}{4}\end{pmatrix} \\
=& \delta_i^1 \begin{pmatrix}\frac{3}{8} & 0\\ 0 & \frac{1}{8}\end{pmatrix} + \delta_i^2 \begin{pmatrix}\frac{1}{8} & 0\\ 0 & \frac{3}{8}\end{pmatrix}.
\end{aligned}$$

This leaves us with the two equations

$$\alpha_i = \frac{3}{8}\delta_i^1 + \frac{1}{8}\delta_i^2 \qquad\qquad \beta_i = \frac{1}{8}\delta_i^1 + \frac{3}{8}\delta_i^2,$$

which are rearranged to express δ_i^1, δ_i^2 in terms of the eigenvalues

$$\delta_i^1 = 3\alpha_i - \beta_i \qquad\qquad \delta_i^2 = -\alpha_i + 3\beta_i$$

to finally obtain equation (5).