Robust Pointset Denoising of Piecewise-Smooth Surfaces through Line Processes

Jiayi Wei, Jiong Chen, Damien Rohmer, Pooran Memari, Mathieu Desbrun

Published at EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner (Guest Editors)
Authors' copy

Robust Pointset Denoising of Piecewise-Smooth Surfaces through Line Processes

Jiayi Wei[1] Jiong Chen[1] Damien Rohmer[1] Pooran Memari[1,2] Mathieu Desbrun[2,1]

[1]LIX, Ecole Polytechnique/CNRS, IP Paris, France
[2]INRIA Saclay, France

Abstract

Denoising is a common, yet critical operation in geometry processing aiming at recovering high-fidelity models of piecewise-smooth objects from noise-corrupted pointsets. Despite a sizable literature on the topic, there is a dearth of approaches capable of processing very noisy and outlier-ridden input pointsets for which no normal estimates and no assumptions on the underlying geometric features or noise type are provided. In this paper, we propose a new robust-statistics approach to denoising pointsets based on line processes to offer robustness to noise and outliers while preserving sharp features possibly present in the data. While the use of robust statistics in denoising is hardly new, most approaches rely on prescribed filtering using data-independent blending expressions based on the spatial and normal closeness of samples. Instead, our approach deduces a geometric denoising strategy through robust and regularized tangent plane fitting of the initial pointset, obtained numerically via alternating minimizations for efficiency and reliability. Key to our variational approach is the use of line processes to identify inliers vs. outliers, as well as the presence of sharp features. We demonstrate that our method can denoise sampled piecewise-smooth surfaces for levels of noise and outliers at which previous works fall short.

CCS Concepts

• ***Computing methodologies*** → ***Point-based models;***

1. Introduction

Digital scanning devices irremediably suffer from various sources of noise and corrupted samples during 3D geometry acquisition, even more so due to the rapid development of low-cost sensors such as line of motion detectors and time of flight cameras. Yet, the need for detailed geometric models is prevalent in a variety of application domains, from archaeology, to retail, to biomedical fields. As a consequence, "denoising" scanned geometric data has spurred a wide range of research efforts that relied on tools from statistical inference and/or differential geometry.

By now, a number of geometric denoising algorithms are able to address specific cases particularly well. For instance, a simple implicit mean curvature flow [DMSB99] can easily and efficiently handle a small amount of isotropic noise, but would fail on anything more challenging. More generally, there exist many numerical methods to extract quite reliably smooth models (i.e., organic shapes) or piecewise-flat models (e.g., mechanical parts) from noise-laden data, even in the presence of (unstructured) outliers. In addition, prior knowledge on the scanning process may help dissociate shape samples from corrupted samples; but while one can identify inaccuracies for a specific sensor by learning its systematic flaws on a few known models [WLT16], the increasing number of 3D datasets constructed from multi-source data integration (including noise with different characteristics due to varying lighting conditions or heterogeneity of sensor technology) renders the task of geometry denoising far more difficult: no simple noise model can be leveraged to infer the original geometry.

When neither a scanning noise model nor a prior on the type of shapes being captured is available, discriminating geometric features from noise to reconstruct piecewise-smooth surfaces becomes exceedingly difficult: the low-pass filter nature of geometric denoising often smooths out sharp edges and corners, particularly if the noise level is significant or if we are dealing with pointsets where no connectivity and normal orientation is available. In this very challenging case, most existing typical tools used to detect noise or outliers simply fail to provide help. In this paper, we aim at providing a novel approach to help remedy this situation.

1.1. Previous Work

Because geometry denoising is both a common and critical operation in geometry processing aiming at recovering high-fidelity models of objects from noise-corrupted data, the related literature in this topic is sizable. We thus focus our review only on the most salient and/or most relevant approaches, be them for meshes, or more relevant to our case, for pointsets—the interested reader can check more thorough reviews, e.g., [HJW*17, ZSL*22].

Filtering-based methods. Early methods for mesh denoising relied on isotropic surface flows designed to remove high frequencies [Tau95], such as the Laplace-Beltrami flow [DMSB99].

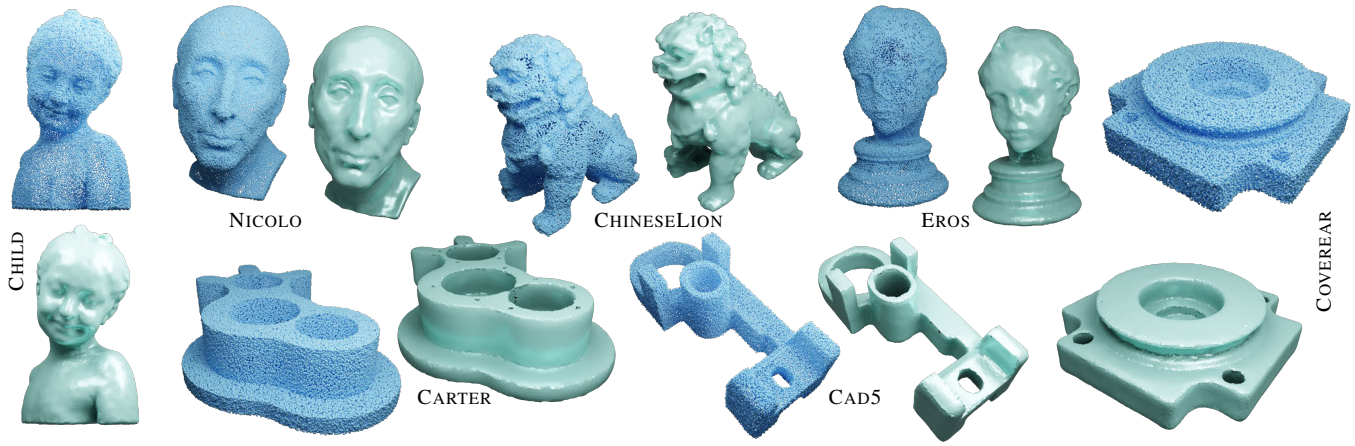


Figure 1: ***Denoising through line processes.*** *By globally optimizing a robust and regularized fitting energy using line processes associated with local pairs of tangent planes, our method can robustly denoise raw point clouds (blue dots) while leaving features (if any are present) well preserved (green). Denoised pointsets are then visualized via their interpolating reconstruction through advancing front [DCS22] to offer a clearer visual evaluation of their (piecewise) smoothness.*

This line of work was generalized to anisotropic flows and signal-processing based filtering [CRT04, LP05], and extended to pointsets [PG01, PKG02, LSK*10, LEL14]. Most of these methods are, however, unable to detect or preserve sharp features during denoising, which spurred the use of more sophisticated approaches such as bilateral filtering [JDD03, FDCO03, DDF17] or median filtering [SB04] for instance: these iterative methods take their own analytical choice of dependence on spatial closeness and normal similarity between points or facets of the input to smooth either the normal estimates [SRML07] (followed by vertex position updating), the positions, or both [ZDZ*15]. Spectral-based methods re-express filtering through an optimization problem based on a hypergraph of the input data obtained via patch selection or downsampling; denoising is then obtained via Laplacian regularization [ALMF19,ZCD20]. The results of these optimization-based approaches, however, depend heavily on the particular expression of spatial and normal dependence for bilateral filtering or on the patch selection for spectral methods, although they usually exceed the quality of the original isotropic methods in terms of feature preservation and robustness to noise — for a proper choice of *user-selected* parameters. Moreover, many of these approaches only apply to input meshes or pointsets equipped with normals.

Optimization-based methods. Another approach to denoising geometric samples of a surface is to formulate an optimization problem to find a deformation that best fits the input data given priors on the underlying surface and noise. Typical priors on the final surface include ℓ_0 or ℓ_1 minimization for CAD-like shapes [HS13], while Gaussian or i.i.d. noise is often assumed [DTB06, WYL*14]. Even if these approaches generate great results with little to no parameter tuning for input data containing specific geometric features or noise patterns, they rarely generalize to more complex cases. Improvement can be achieved by leveraging non-local self-similarity between patches, exploiting low-rank priors [CDY*20, LSL*22], or learning a dictionary [DVC18], at the cost of added computational complexity and possible failures for pointsets lacking self-similarity. Another form of optimization-based approach is the use of moving least squares (MLS [Lev98, ABCO*03]) which was extended to handle outliers and sharp features [FCOS05, OGG09, XF21] through robust statistics. A similar line of research tries to locally project the points on an underlying surface while enforcing uniform distribution (LOP [LCOLTE07, HLZ*09, LXJF13]), this time without relying on tangent plane fitting. These methods, like MLS-based techniques, tend to over-smooth the data when the noise level is high because of their use of fixed and isotropic spatial weighting functions, which can be partially remedied if the noise happens to be a Gaussian mixture [LWC*18].

Learning-based methods. Recent works have focused on learning how to denoise: from a training set containing known shapes sampled by a specific scanning device, supervised approaches train a single neural network layer [WLT16] or a deep network [BJH22] to identify the typical sensor flaws so as to clean future scans. However, many of these methods require a complete remeshing or a voxelization of the input, which significantly reduces their usefulness for pointset denoising. Based on PointNet or PCPNet, supervised approaches specifically targeting pointsets [YLF*18, RLBG*20, PFVM21, ZLQH21] do provide noticeably increased robustness to strong noise levels and outliers. Unsupervised methods (which do not need access to clean examples for training) have been proposed too, mostly based on autoencoders. They either infer the displacement of noisy points from the assumed surface [HRR19, CWS*20] or directly learn the underlying manifold [LH20]. While unsupervised learning has not yet demonstrated robustness to noise, supervised methods have exhibited some of the best results thus far. But supervised learning methods typically suffer from performance degradation when the input pointset deviates significantly from the training data; in particular, features not included in the training set cannot be recovered.

1.2. Contributions

Geometry denoising is a very mature field of research by now, with a number of solutions as we just reviewed. Yet, when dealing with very noisy and outlier-ridden input pointsets without given normals, for which no assumptions on the underlying geometric

features or noise type are known, there are currently very few numerical approaches that one can turn to in order to clean it up. Consolidating this type of geometry input is, however, necessary for even the most basic processing: a simple Poisson reconstruction on a raw pointset often fails because providing a consistent normal orientation is notoriously difficult in the presence of noise [MDGD*10, WLL*22]. We thus propose a new robust-statistics approach to the denoising of challenging pointsets based on a non-linear optimization of the tangent spaces. While the use of robust statistics in denoising is hardly new (Yadav et al. [YSZP20] even argued that most methods are based on some form of robust statistics), most approaches rely on prescribed filtering using closed-form blending expressions based on spatial and normal closeness. Instead, we learn how to derive piecewise-smooth local planar approximations through a non-linear optimization, handled numerically via alternating minimizations for robustness and efficiency. Key to our variational approach will be the use of line processes to detect, within the noise, outliers and sharp features.

2. Robust Statistics and Line Processes

Denoising geometry requires the identification of noise and outliers. Statistical inference can achieve this goal by relying on prior assumptions on the data (randomness, independence, etc). These assumptions are not meant to be precisely true, but just a convenient rationalization of experience. One then hopes that a minor error in the mathematical model should only result in a small error in the final inference. Unfortunately, this does not always hold, and the concept of robust statistics becomes very relevant: as described in [HRRS86], its main goals are "to describe the structure best fitting the bulk of the data, and to identify the deviating data points (outlier) for further treatment", with good performance for data drawn from a wide range of probability distributions. Before delving into our approach, we thus recall some key concepts of robust statistics [Hub81] upon which we will construct our approach.

2.1. M-estimators to generalize least-squares estimates

Suppose we are trying to reconstruct (in 1D for simplicity of explanation) a signal $f(u)$ from noisy samples $y_i = f(u_i) + \varepsilon$, where u_i are the locations where the signal is sampled and ε reflects uncertainty in measurement. From a parameterized set of functions $f_{\mathbf{p}}$ spanning a chosen functional space, the typical least-squares reconstruction $f_{\mathbf{p}^*}$ of the signal f is expressed as:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \sum_i \|y_i - f_{\mathbf{p}}(u_i)\|^2. \tag{1}$$

If the noise ε turns out to be a zero-mean Gaussian random variable, this least-squares fit is identical to the maximum likelihood estimator (MLE). However, noise is rarely normally distributed due, in part, to the unavoidable presence of outliers (i.e., samples that do not follow the noise distribution assumption), and the least-squares reconstruction can be massively affected in this case: outliers can significantly influence the overall solution. In order to remediate this sensitivity to outliers without degrading the quality of the fit, M-estimators propose to recast the variational definition of the best estimate by changing the ℓ^2 norm of the residual by another norm, called the ρ-function. M-estimators thus determine the optimal reconstruction $f_{\mathbf{p}^*_\rho}$ of the signal f in our example via:

$$\mathbf{p}^*_\rho = \arg\min_{\mathbf{p}} \sum_i \rho(y_i - f_{\mathbf{p}}(u_i)). \tag{2}$$

While the least-squares approach corresponding to $\rho(x) = x^2$ is notoriously sensitive to outliers, one can design a ρ function more robust to the breaking of the assumptions about the underlying noise distribution by noting that the derivative ρ', called the influence function [HRRS86], characterizes the bias that a particular sample has on the solution. A number of ρ-functions have been proposed, varying from the truncated L^1 norm to Tukey's function [BT74], for which the influence functions do not grow linearly as for the least-squares approach (thus assigning high weights to far-away samples), but instead offer an upper bound and localized support for the influence of a sample on the solution, see Fig. 2. Using these robust ρ-functions drastically improves the resilience of surface denoising to arbitrary noise; but it also requires non-linear minimizations to solve for the optimal fitting parameters, far more complex to solve than the least-squares approach. Yadav et al. [YRS*18] argue that most state-of-the-art methods for iterative normal filtering (from bilateral filtering to moving least-squares) can be interpreted as estimates of a denoised normal field based on the product of two robust ρ-functions – one based on positions, one based on normals – for various choices of ρ-functions. While such a robust-statistics denoising can handle higher amounts of noise and outliers, the local estimate of a normal is based on a hard-coded choice of spatial weighting and normal weighting of the neighboring samples, rather than a *variational approach* which best robust-fits the input while enforcing *(piecewise) smoothness* as a prior.

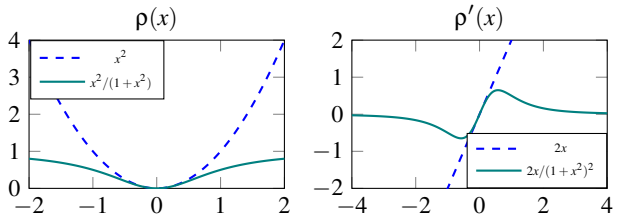


Figure 2: ρ-***function:*** *The dependence of the least-squares estimator (dashed blue) on the error value of a sample is quadratic (left), leading to an unbounded influence (right) of outliers. In contrast, M-estimators such as the one proposed in Geman and McClure [GM87] (solid green) follow a quadratic profile for small error values, but rapidly cut off the influence of outliers, bringing robustness to the M-estimate.*

2.2. Line processes to handle discontinuities

Another powerful tool in robust statistics framework is known as "line processes". Initially proposed by Geman and Geman [GG84] in the context of Markov random fields to handle discontinuities via *penalty terms*, line processes have been popular for segmentation, image reconstruction and optical flow in computer vision. The idea is to favor spatial smoothness as long as neighboring samples are similar enough, and to acknowledge the presence of a discontinuity otherwise. Obviously, the notion of local similarity is very much similar to the notion of inliers vs. outliers, when the prior is smoothness. In fact, line processes are directly related to ρ-functions — although this was realized a posteriori [BR96]. Indeed,

one can regard the penalty term $\Psi(\cdot)$ of a line process as related to a ρ-function if the following property holds:

$$\rho(x) = \inf_{0\leq z\leq 1} zx^2 + \Psi(z), \tag{3}$$

where z is the line process taking values between 0 and 1. When the minimum is reached for z close to 1, the line process identifies the data sample as an inlier, for which the typical least-squares weighting (x^2) is ideal. Inversely, when z is close to 0, the typical quadratic weighting is significantly dampened, identifying a data sample as an outlier to reject or indicating the presence of a discontinuity. In between, the weighting corresponds to an actual ρ-function reflected by the equality in Eq. (3). As a relevant example to illustrate this property, consider the Geman and McClure's robust function [GM87] (see Fig. 2) and its associated line process penalty with *selectivity* μ, expressed respectively as:

$$\rho_\mu(x) = \frac{x^2}{1+x^2/\mu}, \quad \Psi_\mu(z) = \mu(\sqrt{z}-1)^2. \tag{4}$$

Indeed, the necessary condition for minimization in z reads:

$$0 = \frac{\partial}{\partial z}\left[zx^2 + \Psi_\mu(z)\right] = x^2 + \Psi'_\mu(z) = x^2 + \mu\left(1 - \frac{1}{\sqrt{z}}\right).$$

So the minimum is reached for $z^{\min} = (1+\frac{x^2}{\mu})^{-2} \in [0,1]$, for which

$$\begin{aligned} x^2 z^{\min} + \Psi_\mu(z^{\min}) &= \frac{x^2}{(1+\frac{x^2}{\mu})^2} + \mu\left(\frac{1}{1+\frac{x^2}{\mu}} - 1\right)^2 \\ &= \frac{x^2 + \mu\frac{x^4}{\mu}}{\left(1+\frac{x^2}{\mu}\right)^2} = \frac{x^2}{1+\frac{x^2}{\mu}} = \rho_\mu(x). \end{aligned}$$

as expected. In other words, the use of a robust-statistics ρ-function to generalize least-squares estimates can be replaced by extra variables z_i to modulate the quadratic "least-squares" weighting of the various samples, and Ψ_μ can be understood of as a penalty for introducing a discontinuity with selectivity μ. While Eq. (3) proves that robust functions and line processes are equivalent, the latter brings a few significant advantages. First, they offer a straightforward approach to minimizing energies of the same type as Eq. (2) through alternating optimization of simpler problems (similar to Expectation Maximization). Instead of invoking an involved non-linear solver, one can jointly optimize over x and the z_i by alternating restricted minimizations over each individual variable while the others are considered fixed: fixing x leads to a closed-form expression of the optimal $z_i^{\min}$, while fixing all the z_i only requires a linear solve for x, amounting to a weighted least-squares solution. Line processes also have the advantage of making *explicit* which samples are identified to be inliers or outliers — or somewhere in between, as a line process can have any value between 0 and 1. In the context of piecewise smoothness, these line processes can be further leveraged by adding objective functions dependent on the line processes to enforce prior assumptions about the spatial organization of discontinuities, such as straightness or prescribed orientation [BR96]. Finally, since line-process penalty functions are typically convex, one can play with their selectivity μ to devise a continuation method called Graduated Non-Convexity (GNC [BZ03]) in which the continuation parameter μ can be adjusted to construct a convex approximation to the original objective function, before decreasing it to offer an efficient non-linear minimization: the initial approximation will assume only inliers, then outliers will begin to appear (and possibly interact) as minimization goes on, preventing the minimization to get trapped too early in a local minimum.

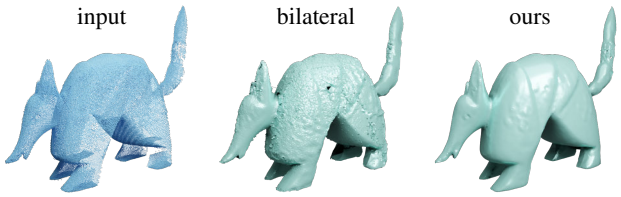


Figure 3: ***When normals are missing:*** *Approaches such as bilateral filtering require normal estimates along with the input pointset; deducing normals from a local neighborhood PCA or more advanced normal estimators [CCC*08] results in bumpy surfaces (left), while our variational approach manages to denoise the input reliably without normal guesses (right). The scanned Armadillo model is from [HWG*13].*

2.3. Discussion

Robust statistics have been highly successful in computer vision by offering tools for the detection of inliers vs. outliers, and similarly, of local smoothness vs. discontinuity. Surprisingly, line processes have not, to our knowledge, been exploited for geometric denoising. As we have seen earlier in Sec. 1.1, this is not to say that robust statistics has not, itself, been leveraged: most bilateral filtering and MLS techniques make heavy use of ρ-functions to offer robustness to spurious data [YSZP20]. While this is often satisfactory for noisy input meshes (where normals are already quite reliable), this strategy too often fails on raw pointsets where normals are not initially known due to the difficulty of finding a globally consistent normal orientation and the fact that initially-inaccurate normals can be falsely detected as features. In our context of denoising raw pointsets, we do not only need to identify outliers, i.e., point samples erroneously located, to prevent them from affecting the probable shape, but sharp geometric features should also be clearly identified and cleaned up to offer accurate reconstruction. Line processes are precisely "exposing" these spatial and normal outliers instead of leaving them implicit in the use of M-estimators — a bit like exposing a transport plan instead of a mere optimal transport cost provides more handles than the cost alone [PCS21].

In the rest of this paper, we present our approach to geometric denoising through robust and regularized fitting of the initial pointset. In order to best adapt previous works in vision to our 3D context, we change the typical two-term objective function accounting for both *data fitting* and *smoothness* by working on the normal field instead: we find local tangent planes that faithfully robust-fit the data while ensuring piecewise-smoothness of the resulting set of all tangent planes, where line processes are used both for robust fitting of the data and for piecewise-smoothness enforcement in order not to smooth sharp features. While previous robust methods for geometric denoising could reasonably well be explained as a form of tangent fitting as well (although not necessarily a variational one), none favors piecewise-smoothness through a robust Dirichlet energy: local smoothness is typically reached through repeated averages instead of through a global competition between data fitting and smoothness. This piecewise-smoothness term also

demarcates our approach compared to LASSO-like L^1-based reconstruction, which uses a piecewise-flat prior. Moreover, we show the advantage of exposing and exploiting the line processes for the final reconstruction of the denoised pointset if needed.

3. Method

We now present our variational approach to geometric denoising through robust and regularized tangent plane fitting of the initial pointset. Note that throughout our exposition, (column) vectors will be denoted with **bold** letters, while scalars will use *italicized* letters.

3.1. Problem formulation

From a pointset $\bar{\mathbf{P}} = \{\bar{\mathbf{p}}_i \in \mathbb{R}^3\}_{i=1}^n$ of n 3D samples, we denote by $\bar{\mathbf{q}}_i = \begin{pmatrix}\bar{\mathbf{p}}_i^t & 1\end{pmatrix}^t$ their corresponding homogeneous coordinates in $\mathbb{R}^4$. For *each* input point i located at $\bar{\mathbf{p}}_i$, we refer as $N(i)$ its k-nearest neighbor samples based on the Euclidean distance, where k is a user-defined constant. From these neighborhoods, we denote by M the pairs of indices (i,j) with $1 \leq i,j \leq n$ for which $i \in N(j)$ or $j \in N(i)$, representing all pairs of nearby input points. Furthermore, we define for each input point a number of associated variables: we denote by $\mathbf{t}_i$ and $\mathbf{h}_i$ two other vectors in $\mathbb{R}^4$ which will represent *two local tangent planes*, implicitly defined in homogeneous coordinates through $\{\mathbf{t}_i^t\,\mathbf{q} = 0 \,|\, \mathbf{q} \in \mathbb{R}^4\}$ and $\{\mathbf{h}_i^t\,\mathbf{q} = 0 \,|\, \mathbf{q} \in \mathbb{R}^4\}$ respectively — the coordinates of these two vectors thus encode both the plane *normals* and *distances* to the origin; we also define line processes through two sets of variables: $l_{ij}\ \forall i$ with $j \in N(i) \cup \{i\}$ (which will help robust-fitting the local plane h_i) and $m_{ij}\ \forall (i,j) \in M$ (which will help enforce piecewise-smoothness); at last, we consider a set of scalars $s_{ij}\ \forall (i,j) \in M$ which will help us deal with normal orientation. For convenience, we will denote by capital letters the sets of all these types of variable, i.e., $\bar{\mathbf{Q}} = \{\bar{\mathbf{q}}_i\}_{i=1}^n$, $\mathbf{T} = \{\mathbf{t}_i\}_{i=1}^n$, and $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^n$ for the vectors in $\mathbb{R}^4$, and similarly for the scalar variables, with $\mathbb{L} = \{l_{ij}\}_{i,j}$, $\mathbb{M} = \{m_{ij}\}_{i,j}$, and $\mathbb{S} = \{s_{ij}\}_{i,j}$.

Our approach boils down to a variational formulation involving the iterative minimization of an energy E dependent on $\mathbf{H}, \mathbf{T}, \mathbb{L}, \mathbb{M}$, and $\mathbb{S}$, formed as the sum of three coupled terms favoring respectively *local plane fitting* term, *piecewise smoothness*, and *stitching*, under simple constraints that avoid degeneracies. More precisely, the explicit formulation of our energy minimization reads:

$$\begin{aligned}\min_{\mathbf{H},\mathbf{T},\mathbb{L},\mathbb{M},\mathbb{S}} \quad & \underbrace{\frac{1}{2}\sum_{i=1}^{n}\sum_{j\in N(i)\cup\{i\}} \alpha_i \left[\mathbf{h}_i^t\,\bar{\mathbf{q}}_j\bar{\mathbf{q}}_j^t\,\mathbf{h}_i\, l_{ij} + \Psi_{\mu_l}(l_{ij})\right]}_{\text{local fitting}} \\ & + \underbrace{\frac{\lambda}{2}\sum_{(i,j)\in M} \beta_{ij}\left[||\mathbf{t}_i - s_{ij}\,\mathbf{t}_j||^2\, m_{ij} + \Psi_{\mu_m}(m_{ij})\right]}_{\text{piecewise smoothness}} \\ & + \underbrace{\frac{\eta}{2}\sum_{i=1}^{n}\alpha_i||\mathbf{h}_i - \mathbf{t}_i||^2}_{\text{stitching}} \\ \text{s.t.} \quad & \|\mathbf{h}_i\| = 1\ \forall i \in [1,n].\end{aligned} \tag{5}$$

The local weights α_i are used to weight the fitting energy by a rough estimation of the area that point i covers, in order to accommodate for varying density in the input pointset $\bar{\mathbf{P}}$ through:

$$\alpha_i = \frac{\sum_{j\in N(i)} ||\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j||^2}{|N(i)|},$$

where $||\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j||^2$ measures a small square area including points $\bar{\mathbf{p}}_i$ and $\bar{\mathbf{p}}_j$. Similarly, the pairwise smoothness term is weighted by β_{ij}, derived from the area covered by the pair (i,j) via:

$$\beta_{ij} = \frac{\dfrac{\alpha_i}{|N(i)|} + \dfrac{\alpha_j}{|N(j)|}}{||\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j||^2}.$$

3.2. Interpretation of our variational formulation

The three terms of the energy E are simple to analyze. The first (local fitting) term is designed to robust-fit a plane $\mathbf{h}_i$ for each point $\bar{\mathbf{p}}_i$ to its immediate neighborhood using line "outlier" processes l_{ij} and their associated penalty function Ψ_{μ_l}. The variables l_{ij} will naturally adapt to local inliers (resp., outliers) by going towards 1 (resp., 0), providing a reliable local estimate of the tangent plane. The added constraint on the magnitude $\mathbf{h}_i$ prevents the trivial (but meaningless) solution $\mathbf{h}_i = \mathbf{0}$. The second (piecewise-smoothness) term forces the robust alignment of neighboring planes $\mathbf{t}_i$ and $\mathbf{t}_j$ using line "feature" processes m_{ij}. One can regard this energy as a "robust" Dirichlet energy: if two nearby planes are too different, their associated m_{ij} will become 0 (preferring the small penalty $\Psi_{\mu_m}(m_{ij})$ rather than the large squared difference): sharp features will thus not be subject to alignment, creating piecewise smoothness. The scalar coefficients s_{ij} are also added to remove the ambiguity of orientation: while $\mathbf{t}_i$ defines a plane, $-\mathbf{t}_i$ (or any other scaling) defines the same plane. Minimizing over s_{ij} will automatically pick the correct scaling/sign, thus properly forming a Dirichlet-like energy for tangent planes (Fig. 4 shows how not addressing the possibly flipped signs severely affect denoising). Finally, the third term is simply forcing the two tangent-plane estimates $\mathbf{t}_i$ and $\mathbf{h}_i$ for any given point i to match via a straightforward quadratic penalty. The use of two variables to encode what should be the same tangent plane will become obvious when we describe our minimization procedure in the next section: it will make our iterative minimization procedure far simpler. Note finally that we use Geman and McClure's penalty functions [GM87] as described in Sec. 2.2, with selectivities μ_l for fitting and μ_m for smoothness. Finally the values λ and η are constants to tune the balance between enforcement of local fitting vs. global piecewise-smoothness. Note that the norm used for the term $||\mathbf{t}_i - s_{ij}\,\mathbf{t}_j||^2$ could be altered to weight the last homogenous coordinate differently from the first three, to tweak the balance between normal difference and distance-difference. In practice, we kept the usual ℓ^2 norm.

3.3. Optimization via alternating restricted minimizations

Our variational formulation seeking to balance local fitting and global piecewise smoothness has been designed to be particularly simple to optimize via alternating restricted minimizations. That is, solving for a given family of variables while keeping all the other fixed (akin to an expectation maximization or a coordinate-block descent) is particularly simple as we now review.

Optimizing the line processes $\mathbb{L}$ and $\mathbb{M}$. If all other variables

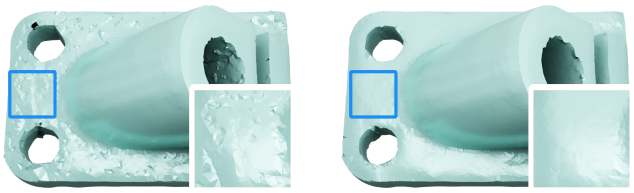


Figure 4: ***Normal orientations.*** *Because our tangent plane encoding is not scale invariant, one must introduce a scalar s_{ij} when comparing two nearby tangent planes through $||\mathbf{t}_i - s_{ij}\mathbf{t}_j||^2$: at any point during the optimization, assuming $s_{ij}=1$ introduce spurious artifacts (left) that an optimization over s_{ij} resolves (right).*

are held constant, then minimizing our energy with respect to $\mathbb{L}$ (resp., $\mathbb{M}$) is trivial as it has an explicit optimal solution:

$$l_{ij} = \left(\frac{\mu_l}{\mu_l + \mathbf{h}_i^t \bar{\mathbf{q}}_j \bar{\mathbf{q}}_j^t \mathbf{h}_i} \right)^2 \tag{6}$$

for outlier processes, while feature processes need to be set to:

$$m_{ij} = \left(\frac{\mu_m}{\mu_m + ||\mathbf{t}_i - s_{ij}\mathbf{t}_j||^2} \right)^2. \tag{7}$$

Optimizing orientations $\mathbb{S}$. Similarly, optimizing the variables s_{ij} when all other variables are held constant is done by setting explicitly the optimal solution through:

$$s_{ij} = \frac{\mathbf{t}_i^t \mathbf{t}_j}{\mathbf{t}_j^t \mathbf{t}_j}. \tag{8}$$

Optimizing tangent planes T. Because we used two separate tangent planes (through $\mathbf{T}$ and $\mathbf{H}$), optimizing for $\mathbf{T}$ is particularly simple too as it consists in solving a weighted least-squares problem. Assuming that $\mathbf{T}$ (resp., $\mathbf{H}$) is thought of as a $n\times 4$ matrix storing one tangent plane $\mathbf{t}_i$ (resp., $\mathbf{h}_i$) per row, then $\mathbf{T}$ is found efficiently by solving the following linear sparse linear system:

$$\mathbf{K}\mathbf{T} = \mathbf{W}, \tag{9}$$

with the left-hand and right-hand side matrices being

$$\begin{cases} \mathbf{K} = \eta\, \mathrm{diag}(\alpha_i) + \lambda \sum_{(i,j)\in M} \beta_{ij} m_{ij} (\mathbf{e}_i - s_{ij}\mathbf{e}_j)(\mathbf{e}_i - s_{ij}\mathbf{e}_j)^t, \\ \mathbf{W} = \eta\, \mathrm{diag}(\alpha_i)\, \mathbf{H}, \end{cases}$$

where $\mathbf{e}_i$ denotes the indicator vector of size n with the i-th element set to 1 and all the others to 0, and $\mathrm{diag}(\alpha_i)$ represents a $n\times n$ diagonal matrix containing the values $\alpha_1, \alpha_2, ..., \alpha_n$ on the diagonal.

Optimizing tangent planes H. Optimizing the local tangent planes $\{\mathbf{h}_i\}$ is the most complicated task, as it is akin to an eigenvalue problem due to the presence of a unit constraint — but this constraint is necessary to prevent a collapse to the trivial and undesirable solution $\mathbf{h}_i = \mathbf{0}$. Minimizing the (constrained) energy with respect to $\mathbf{H}$ amounts to solving, for each $i \in [1,n]$, the following local non-linear problem:

$$\min_{\mathbf{h}_i} \ \frac{1}{2}\mathbf{h}_i^t \mathbf{A}_i \mathbf{h}_i - \mathbf{b}_i^t \mathbf{h}_i \ \ \text{s.t.} \ \ ||\mathbf{h}_i|| = 1, \tag{10}$$

for $\mathbf{A}_i = \alpha_i(\eta\mathbf{I} + \sum_{j\in N(i)\cup\{i\}} \bar{\mathbf{q}}_j \bar{\mathbf{q}}_j^t l_{ij})$ ($\mathbf{I}$ denoting the 4×4 identity matrix), and $\mathbf{b}_i = \eta\alpha_i \mathbf{t}_i$. As detailed in [Hag01] (Lemma 2.2), the optimal solution can be expressed in terms of the eigenpairs of $\mathbf{A}_i$, computed through a combination of orthonormal eigenvectors after a diagonalization of $\mathbf{A}_i$ for non-degenerate cases of $\mathbf{b}_i$, which is our case —see App. A for computational details. Note that although using only a unit constraint on the *normal* of $\mathbf{h}_i$, the inverse matrix in Eq. (11) would no longer be diagonal, making Eq. (12) far more complex to solve [Hag01]. We thus prefer a constraint on the full norm of $\mathbf{h}_i$, as it also accounts for the distances to planes. Since each matrix involved is only of size 4×4 and that each local tangent plane can be computed individually and in parallel, evaluating all the optimal $\mathbf{h}_i$ is still simple and efficient computationally.

Optimization order. While one could adopt any order for the iterated solves required in the global minimization process, we first solve the local fitting stage by updating $\mathbb{L}$ and $\mathbf{H}$; next, from the resulting approximations of the local tangent planes, we update $\mathbb{M}$, $\mathbb{S}$, and $\mathbf{T}$. We also found systematically improved results if we iterate this latter step twice, see the pseudocode of the whole minimization process in Alg. 1. Note that we typically stop optimizing when we reach a maximum number of iterations, or when the total energy does not change by more than 1% over three iterations.

Algorithm 1 Robust denoising via line processes

Input: noisy point cloud $\{\bar{\mathbf{p}}_i\}_{i=1}^n$, k for k-nearest neighbors, line-process selectivities μ_m, μ_l, weight for smoothness λ

Initialize sets M and N using k-nearest neighbors search
$\mathbf{H} \leftarrow \mathbf{0}_{n\times 4}$
$\mathbf{T} \leftarrow \mathbf{0}_{n\times 4}$
$l_{ij} \leftarrow 1, \forall i, j \in N(i)\cup\{i\}$
$m_{ij}, s_{ij} \leftarrow 1, \forall (i,j) \in M$
$iter \leftarrow 0$
while $iter <$ `maxIter` **do**
  /* local tangent plane fitting step */
  update $\mathbf{H}$ by solving Eq.(10)
  update $\mathbb{L}$ using Eq.(6)
  /* global piecewise-smoothness step */
  $innerIter \leftarrow 0$
  while $innerIter < 2$ **do**
    update $\mathbf{T}$ following Eq.(9)
    update $\mathbb{M}$ using Eq.(7)
    update $\mathbb{S}$ using Eq.(8)
    $innerIter \leftarrow innerIter + 1$
  end while
  $iter \leftarrow iter + 1$
end while
/* final denoising (Sec. 3.4) */
for $\bar{\mathbf{p}}_i \in \mathcal{V}$ **do**
  project $\bar{\mathbf{p}}_i$ onto plane $\mathbf{t}_i$ to obtain $\mathbf{p}_i$
end for

Output: denoised point cloud $\{\mathbf{p}_i\}_{i=1}^n$

3.4. Final denoising

The result of our minimization provides all the necessary tools to finalize pointset denoising — in particular, having the line processes readily available lends crucial help to cleaning up and handling sharp features.

Outlier indicator. The line processes l_{ij} were used to sort out inliers from outliers. We can then use their final values as a robust indicator of the presence of outliers as seen in Fig. 6. As l_{ij} is a pairwise indication between neighboring points, we declare an input point i to be an outlier of the pointsets if a majority of its neighbors

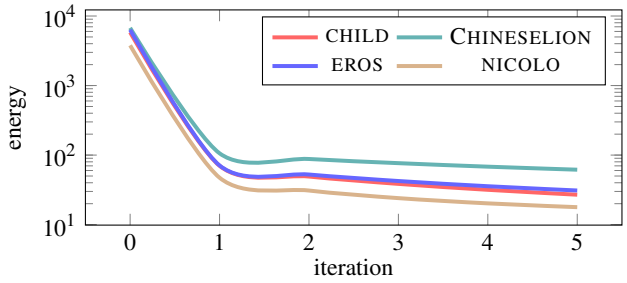


Figure 5: ***Energy minimization.*** *Our alternating restricted minimizations reliably decrease the regularized fitting energy in very few iterations, as demonstrated here on four of the models of Fig. 1.*

$j \in N(i)$ are associated to small l_{ji} values. In our implementation we consider the threshold where 90% of the l_{ji} values are below 0.5. We note that the precise value of the percentage and threshold is not particularly important: changing them to, e.g., 80% of values below 0.3 typically changes the number of identified outliers by a negligible number of points in our examples.

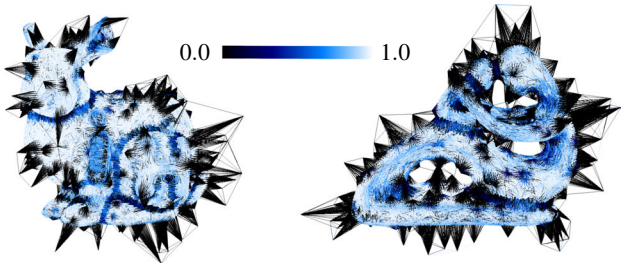


Figure 6: ***Outlier indicator:*** *Color-encoded segments between i and j are associated to the value* $\min(l_{ij}, l_{ji})$*, after minimization of our regularized fitting energy. This visualization reveals that outliers are clearly attached to small l_{ij} (or l_{ji}), while inliers values range between* 0.75 *and* 1 *depending on the local curvature.*

Feature indicator. Similarly, the line processes m_{ij} were used to sort out smooth vs. sharp transitions between tangent planes during the optimization. So we declare a point i as being close to a feature if more than 70% of the values of the m_{ji}'s $\forall j$ are below 0.5. Here again, the exact values hardly matter, see Fig. 7.

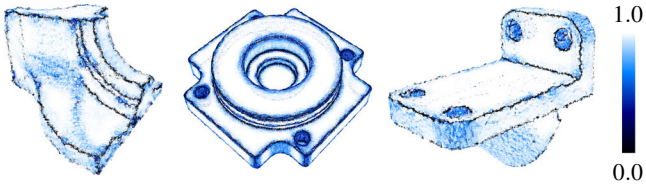
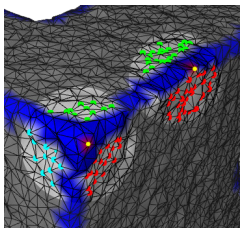


Figure 7: ***Feature indicator:*** *After minimization of our regularized fitting energy, a visualization of the line processes m_{ij} by segments between i and j with a color corresponding to its magnitude reveals that small values all correspond to the local presence of sharp features, while values close to* 1 *can be seen everywhere else, with small variations due to local mean curvature.*

Denoising and cleaning. The final denoising step is then easily achieved: an input point i located at $\bar{\mathbf{p}}_i$ is simply projected onto its associated tangent plane $\mathbf{t}_i$, thus forming a denoised sample: $\mathbf{p}_i = \bar{\mathbf{p}}_i - \mathbf{n}_i(\mathbf{t}_i^t \bar{\mathbf{q}}_i)/\|\mathbf{n}_i\|^2$ where $\mathbf{n}_i = (\mathbf{t}_i[1]\ \mathbf{t}_i[2]\ \mathbf{t}_i[3])^t$ is the 3D vector equal to the first three coordinates of $\mathbf{t}_i$, — except if it was recognized as an outlier, in which case we simply remove it as its projection could be arbitrarily bad.

Optional feature improvement. Points on or around the features can be further processed, in particular if the denoised pointset is to be fed to a surface reconstruction algorithm afterwards so as to better capture sharp features. Currently, a point i near a feature is simply projected to its own optimized tangent plane $\mathbf{t}_i$. Consequently, sharp features may contain very few or no points on the features themselves: this may reconstruct jagged features when fed to a reconstruction algorithm based on interpolating meshes. Therefore, we found more productive to use our feature indicator based on the feature processes: if a point is declared to be near a feature, then we slightly modify the projection so that it is projected on the most likely position of the nearby feature. To achieve this, we perform clustering on the local $\mathbf{t}_j$ for all j such that $(i,j) \in M$ of its neighborhood using the DBSCAN clustering method [EKS*96] where we ask for a maximum of three clusters. If only two clusters are found, we are likely to be in a typical feature line or curve (see inset, yellow sample on the right as part of the anchor shape); we thus compute the intersection line between the two planes that the centroid of these two clusters define: we then project $\bar{\mathbf{p}}_i$ to the resulting line. If three clusters are found, we are more likely to be near a cube or polyhedral corner (see inset, yellow sample on the left); we then pick the intersection line of the two centroid-based planes to which $\bar{\mathbf{p}}_i$ is closest in terms of orthogonal distance, and project the sample to this line as above. We found this simple post-processing to be the simplest reliable way to provide better shape reconstruction. Approaches using more involved local feature reconstructions are possible as well, but we kept the simplest approach to focus on the "raw" power of line processes.



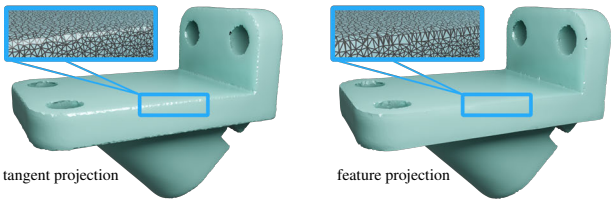


Figure 8: ***Feature improvement:*** *While our final projection to the local tangent plane provides a proper denoising even near features (left) due to the use of feature processes, one can improve the quality of downstream* interpolating *reconstructions by, instead, projecting near-feature points to guessed sharp features (right).*

3.5. Discussion

In this section, we have adapted the traditional robust approach of image denoising [BR96] to the case of 3D pointsets via a robust and regularized fitting of local tangent planes. The precise form of the energy we minimize has been selected to offer a simple iterative minimization through the use of line processes and soft enforcement. While many other formulations could have been simpler, we settled on this choice for its *efficiency* and *reliability*. Convexity

could have even been offered by first computing a robust fitting of local tangent planes $\hat{\mathbf{t}}_i$, then only using the smoothness term to better align these planes; however, the level of noise and outliers that we are seeking leads irremediably to initial guesses of tangent planes that are significantly off, so relying too much of these static fitting planes fails to provide good results: large errors are very often ending up selected as features instead of being properly regularized. Similarly, keeping one single variable (for instance, $\mathbf{t}_i$) instead of two in order to encode a local tangent plane turns the entire minimization into a large constrained and non-linear optimization — our soft stitching term breaks this large and difficult minimization into a series of simple and/or local steps, eventually requiring much less computational time and a far-simpler implementation.

One of the biggest benefits of our formulation is its ability to "learn", as iterations proceed, which samples are outliers and where sharp features are likely to be by regularizing the local tangent planes: it thus requires no input normals (although if they are available, they can be used in the initialization of the tangent planes), and one can tweak the amount of regularization while existing methods like bilateral filtering can only add further iterations — which does not allow for local adaptivity of the amount of smoothing performed. Next, we discuss how our approach behaves on a variety of geometric objects and for diverse types of noise, noise levels, and outlier density.

4. Results

We now present a series of results and comparisons to help understand how our approach behaves and differs from previous works. In order to provide a fair assessment, we use both CAD models (containing sharp features) and organic shapes with synthetic noise and outliers, as well as real-world scanned data; we also provide off-the-shelf reconstructions derived from our results so as to better evaluate the quality of our denoising strategy. *Note we did not perform any post-processing on our results*: even if straightening of features or local smoothing could be easily added to drastically improve the results, we refrain from such practices. We will release our code upon publication of this paper.

4.1. Setup

Starting from a raw input pointset rescaled to fit a unit cube (to ensure scale invariance), implementing our approach is rather straightforward, and selecting its few parameters is quite intuitive.

Implementation details. We implemented the denoising approach in C++, and run all our tests on a desktop equipped with a 6-core Intel Xeon® W-2133 CPU at 3.60GHz and 64G RAM. We follow the approach of [Hag01] to solve Eq. (10) as described in App. A in parallel with 6 threads, while the positive-definite linear system from Eq. (9) is solved using a Conjugate Gradient solver with an algebraic multigrid preconditioner [Dem20], which scales well with the problem size.

Parameter settings. Our variational approach relies on four main parameters: the two selectivities for feature and outlier processes μ_m and μ_l, and the two coefficients λ and η that scale, respectively, the smoothness term and the stitching term of the energy we minimize. Selectivity μ_l can be set to 5×10^{-9} as it roughly defines the squared distance to a fitting plane that would be considered as too large to be an inlier. Selectivity μ_m, instead, influences the minimum normal angle change that one must witness to decide that a sharp feature is happening: as Fig. 9(top) demonstrates, a small value identifies the edges and corners of a noisy sampled cube, while higher values will assume that there is no features present. The effect of λ is also clear, as its value directly controls the strength of the piecewise-smoothness term: as Fig. 9(bottom) shows, values around 0.5 may leave oscillating faces on the noisy cube, while 3.0 enforces flatter sides. This last parameter was found to be the most valuable to tweak depending on the data: based on the result using a default $\lambda=1.0$ value, one can relaunch the optimization from the current optimal tangent planes after changing the weight λ to lower or increase the smoothness of the final denoising very efficiently. Finally, we set η, the stitching strength, to 5000 for all results in order to reliably force the agreement of $\mathbf{t}_i$ and $\mathbf{h}_i$. Incidentally, one can also adjust the number k of neighbors used to perform the local fitting neighborhood and to define the line processes. In practice, we found that selecting $k=20$ leads to systematically good results; however, the noisier the input data is, the larger this value could be set to as it offers marginal improvements by drawing more neighbors into local estimates — in Fig. 13 for instance, we obtained better results by increasing the number of nearest neighbors to 150 for very large amount of noise. If noise is not particularly prominent, using 20 offers a sufficiently large neighborhood to help recognizing outliers and features reliably.

Reconstruction test. One of the common downstream applications of pointset denoising is mesh reconstruction. In order to both test the quality of denoising and offer visually-clear representations of our results, we sometimes use reconstructions in our figures. We chose two existing interpolating methods as "neutral" reconstruction methods: the *Advancing Front* [DCS22] (typically, for low noise or CAD models), and *Scale Space* techniques [vL22], both from the CGAL [CGA22] library. We display the best of these two reconstructions to highlight the potential flaws of the results.

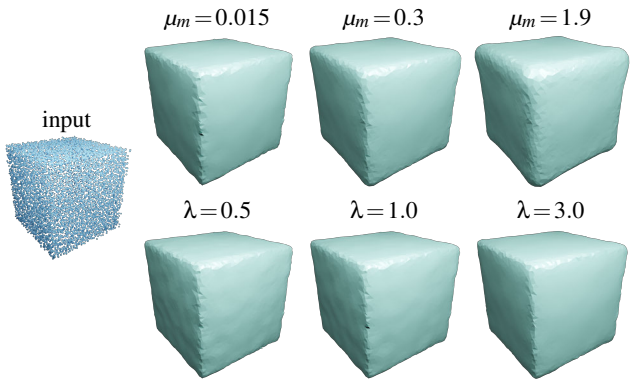


Figure 9: ***Effects of parameters:*** *On a simple noisy cube example (Gaussian noise with a standard deviation equal to 1% of the model diameter), running our minimization with varying values of the selectivity μ_m captures (left) or do not capture (right) the sharp edges (top) ; if the smoothness strength λ is changed, the result ends up with more or less flat faces (bottom). We used $\lambda=3.0$ for the top row, and $\mu_m=0.13$ for the bottom row.*

4.2. Performance

Our iterated restricted minimization strategy always resulted in drastic loss in the energy to minimize, as shown on four examples in Fig. 5; note that even if the energy is not technically convex (due to the norm constraint), we witness a mostly monotone decrease. In Tab. 3, we show that the global linear solve is the slowest part of an iteration of the minimization, but even this stage scales roughly linearly with the number of samples, see Figs. 10 and 11 for examples with over a million samples. As expected, we also see that the execution is linear in the number of neighbors used. In practice, our minimization-based method can be slower than methods using fixed blending of sample values to provide denoising, but surprisingly, the efficiency of our approach makes the difference only marginal for the examples we tried. As a representative example, a 50K-sample noisy version of the accessory model (as used in Fig. 14) takes three iterations of Bilateral Filtering [AGJ*22] for a total of 13.5*s*, while our approach takes 10 iterations to converge, for a total execution time of 16*s*. On the same example, APSS and RIMLS [CCC*08] take 19*s* and 35.5*s* respectively; in comparison, GLR [ZCN*19], which only runs on MatLab, took 400*s*, while PointCleanNet [RLBG*20] took 438*s* on a NVidia GPU Tesla K80 using CUDA 10.1.

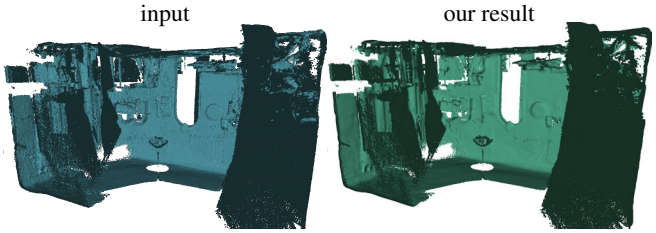


Figure 10: ***Large-scale denoising I:*** *Our method exhibits good scalability in problem size: for an indoor scan with 1.15 million points, our denoising method takes 115s (convergence is reached in 5 iterations), with about 3s for local fitting and 20s for the global linear solve per iteration.*

4.3. Robustness to outliers

Eliminating outliers in raw poinsets has been a long-standing issue which received a large amount of attention. Robust approaches exist, often based on eigenanalysis, machine learning [RLBG*20], various thresholding strategies [WKZ*16], or optimal transport [CCSM11, DCSA*14], and there are often parameters to tune in order to correctly identify outliers. In our denoising method, the fact that we know that we are looking for a surface embedded in $\mathbb{R}^3$ makes the task particularly simple and robust: the outlier indicator from Sec. 3.4 obtained after optimization can reliably identify outliers as the outlier processes will jointly declare these samples as too unfit to serve for local fitting. And indeed, Fig. 12 demonstrates that the amount of outliers added to a noisy pointset does not noticeably affect the denoised results, which we fed to a neutral reconstruction algorithm [vL22] to better illustrate the resulting shape.

4.4. Robustness to Gaussian noise

Gaussian noise is the usual go-to noise type to test denoising algorithms. However, the amount of noise is often quite small compared

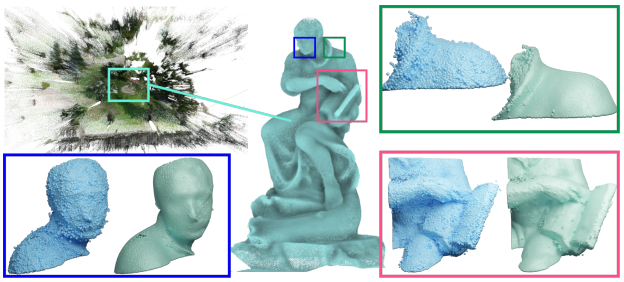


Figure 11: ***Large-scale denoising II:*** *After applying our method on a large-scale real scanning-scene from [KPZK17] (∼7M points) where noise type and level vary in space, we visualize the original vs. denoised pointset on the statue at the center of the scene in closeup insets: face, collar, hand and book are well reconstructed, with clear sharp features, e.g., on the collar and the open book.*

to the features of the shape, and the noise is often only added in the normal direction, which renders denoising simpler. Moreover, most denoising methods take a mesh as an input, which still provides a reliable way to deduce normal orientation. On raw pointsets, normals and their orientation are simply not available — and particularly difficult to infer when the noise level is large enough. While many approaches can handle small noise (and ours too, as demonstrated previously in Figs. 1 and 12), we focus our evaluations here on larger noise magnitudes to render the visual inspection of the results much more obvious. Once again, we also provide the visualization of a neutral reconstruction (Sec. 4.1) run on the denoised results to make the flaws more visually salient. Fig. 13 shows how badly bilateral filtering can fail to properly denoise if the amount of isotropic Gaussian noise (not just along the normal) is too large — in this example, for a standard deviation of 1%, 2%, and 3% of the model diameter estimated by FPS (farthest point sampling technique [ZCN*19]). Our approach, instead, degrades gracefully as the noise amplifies. Similarly, we compare other previous methods (namely, APSS [GG07], RIMLS [OGG09] (both included in MeshLab [CCC*08]), Bilateral Filtering [HWG*13] (implemented in [AGJ*22]), GLR [ZCN*19], and PointCleanNet [RLBG*20]) to

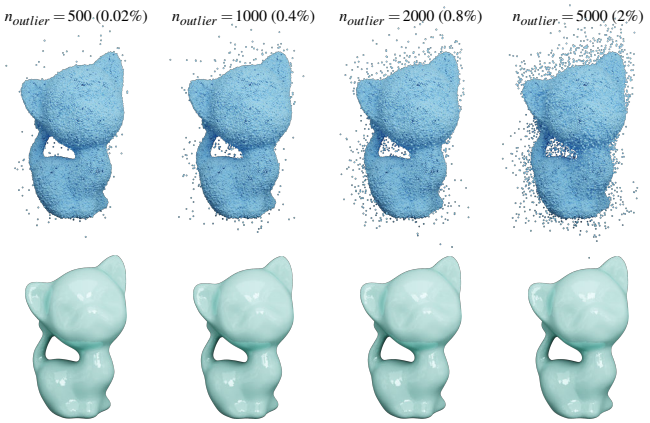


Figure 12: ***Robustness to outliers:*** *For a moderately noisy 25K-sample kitten (taken from [WLT16]), the addition of 500, 1K, 2K or 5K outliers barely changes the final denoising as outlier processes quickly identify these spurious points from all the samples.*

ours in Fig. 14, exemplifying the brittleness of denoising methods in the case of large noise (note that for the methods needing a normal per point, we use Meshlab's normal estimator to offer good approximations). These methods were selected to represent a variety of prior art and due to the existence of available implementation (either from the authors' repository, or from CGAL [CGA22] or MeshLab [CCC*08]) for transparency. We also provide various quantitative comparisons in Tab. 1: in order to measure the geometric error of our denoising technique and compare it to state-of-the-art methods, we use three classical error metrics as proposed in [ZCN*19], namely mean-square-error (MSE, defined as the average between the symmetrized squared Euclidean distances between ground-truth points and their closest denoised points), signal-to-noise ratio (SNR, a function of the log of the inverse of MSE), and mean city-block errors (MCD, a variant of MSE in which the ℓ^2 norm is replaced by ℓ^1 norm). While our approach seems to outperform the others on most pointsets, one should notice that these errors are not representative of the "visual" quality: often, a lower value of MSE or MCD or a larger value of SNR does not imply a visually better result. Nevertheless, we found that our approach is the only one to be always among the best ones, both visually and in terms of error metrics — sometimes by a large margin. Note that all other methods required parameter tuning to provide optimal results: we selected what we found to be the best values — for instance, we varied the neighborhood size from 50 to 150 for the GLR and bilateral methods, and the filter scale from 5 to 15 for the APSS and RIMLS methods depending on the models. For our method, we vary the neighborhood size k from 50 to 150 based on the level noise and the smoothness weight λ from 1.0 to 2.0, see Tab. 2.

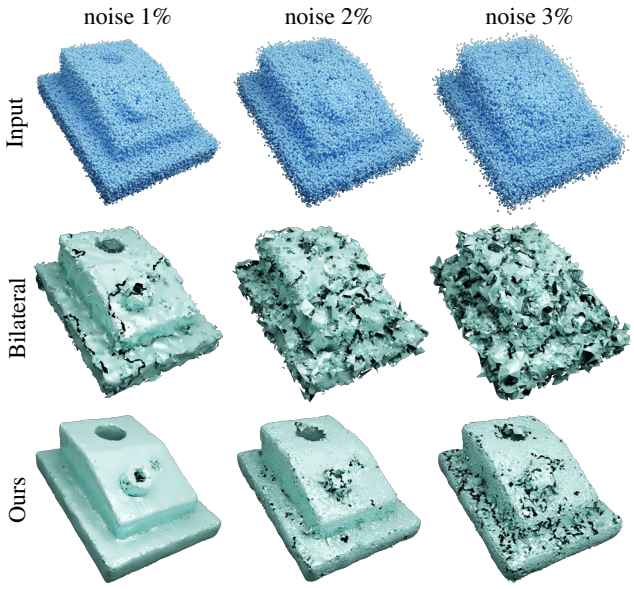


Figure 13: ***Robustness to noise levels****: On this* ACCESSORY *CAD model, adding a Gaussian noise with a standard deviation equal to 1% of the model diameter (left column) already exceeds the ability of bilateral filtering, while our approach still returns a nearly clean denoised model. Doubling (middle), then tripling (right) the standard deviation only slowly degrade our results, but render the bilateral result unusable.*

4.5. Robustness to raw pointsets

For certain denoising methods, a (possibly noisy) normal must be provided with each sample of the input pointset. We found out that many such methods are quite sensitive to errors in the normal; for instance, providing a PCA estimate of the normal (which, in the absence of outliers, should be about the best guess one can derive) can result in poor result for a bilateral filtering approach [HWG*13], whereas our denoising involves the variational establishment of its own normal field, thus resulting in significantly better results, as demonstrated in Fig. 3.

4.6. Robustness to real data

We also compared our approach to existing methods on real scan pointsets, issued from Microsoft Kinect® v1.0 and provided in [WLT16]. Here, the noise is neither close to Gaussian, nor spatially constant, so it is a real-life denoising example rather than a mere academic exercise. As Fig. 16 demonstrates, our approach holds its own on this type of data too. Indeed, GLR [ZCN*19] code systematically fails on these examples as their implementation assumes the absence of collinear points among neighbors, which happens not to be true in practice; and surprisingly, even machine-learning approaches which use some of these examples as training do not provide very good results. Note that here also, methods needing a normal per point are provided with the normal estimate from MeshLab [CCC*08].

4.7. Visualization

Finally, we provide a few examples of visualizations of the results of our energy minimization, as they provide useful insights on our approach. It is instructive to see in Figs. 6 and 7 for instance that the line processes are able to clearly find outliers and sharp features, while the range of the processes indicates clearly the presence or absence of local curvatures. Similarly, the error maps displayed in Fig. 15 compare the local denoising errors generated by our approach vs. PointCleanNet [RLBG*20]. Here again, our energy minimization using line processes is able to identify features and outliers quite well, despite the amount of noise present in the input data.

5. Limitations and future works

Denoising 3D data with little to no prior knowledge on the noise type and amount is a difficult task, for which no method can argue to provide a definite, general solution. What we presented in this paper is a solution based on a rather classical variational approach balancing local fitting and global smoothness, to which we incorporated robust statistics and line processes that were initially proposed in image processing. We were able to outperform prior art in the challenging case of raw pointsets where no normals are known, for which very few existing techniques apply. However, this first use of line processes for denoising has a number of remaining limitations. First, thin structures are still impossible to extract for very noisy data: our algorithm does not always reconstruct the tip of a cone perfectly for example, see Fig. 17. Fixing this issue would most likely require dedicated RANSAC-like approaches to

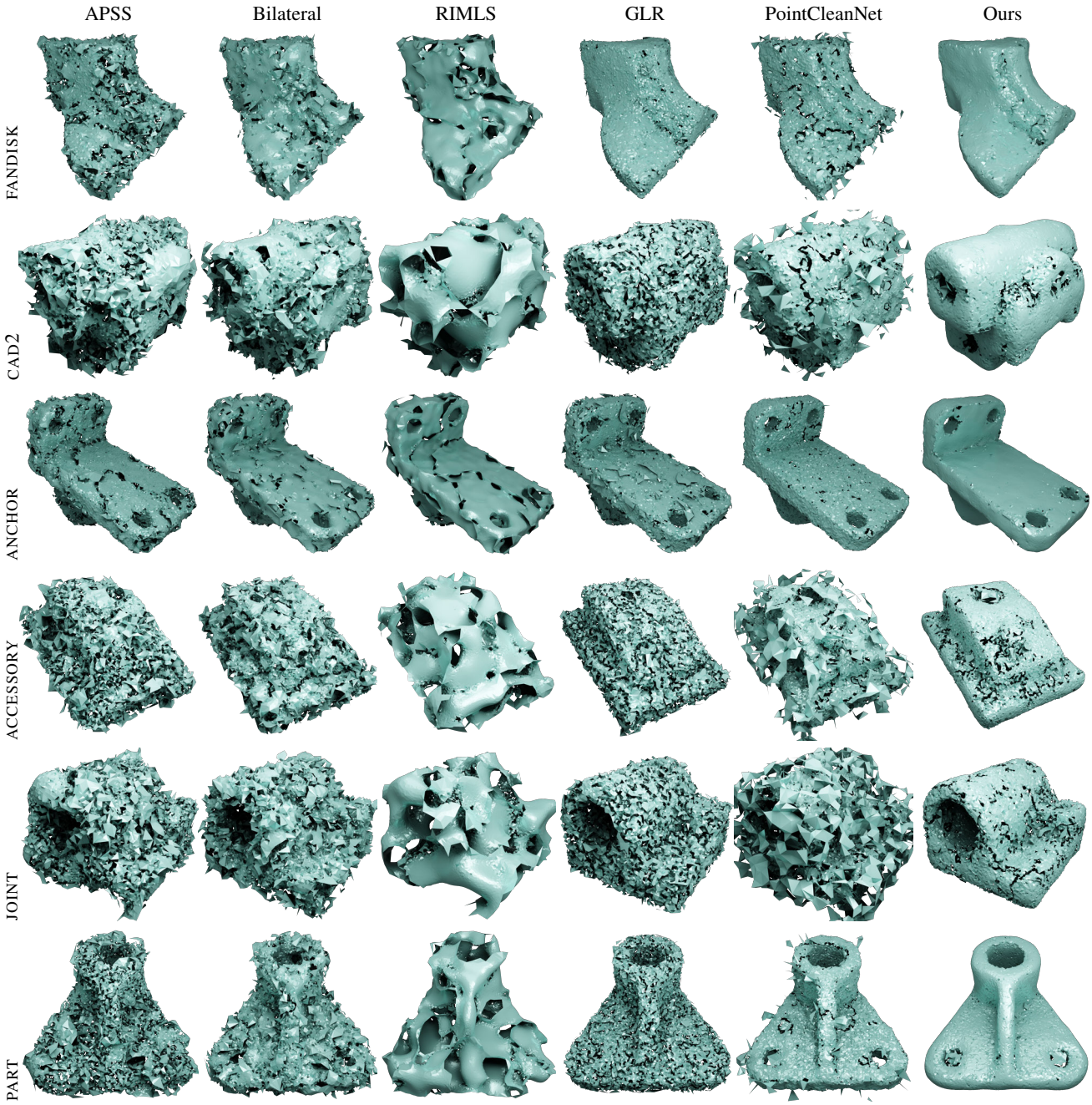


Figure 14: ***Stress tests.*** *For various CAD models sampled with a very large noise level (Gaussian noise for a standard deviation of 3% of the model diameter), we test interpolating reconstructions of the denoised results of APSS [GG07], Bilateral Filtering [HWG*13], RIMLS [OGG09], Graph Laplacian Regularization [ZCN*19], PointCleanNet [RLBG*20], and our approach.*

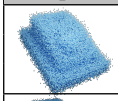
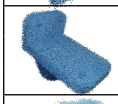
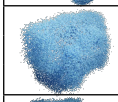
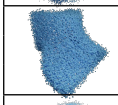
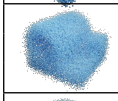
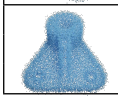
Input	APSS	Bilateral	RIMLS	GLR	PCNet	Ours
	1.15 3.51 2.74	0.74 2.94 2.93	1.57 4.33 2.60	0.559 2.60 3.05	0.484 2.01 3.12	**0.328** **1.91** **3.29**
	261 64.7 3.57	245 63.8 3.60	250 64.9 3.59	181 54.5 3.73	**121** **48.2** **3.91**	195 59.5 3.70
	1.39 4.12 2.98	0.866 3.42 3.18	2.23 5.36 2.77	0.545 2.74 3.38	0.544 2.49 3.38	**0.385** **2.42** **3.53**
	5.37 8.43 4.65	4.39 7.89 4.73	5.94 9.09 4.60	31.9 19.0 3.87	2.88 5.91 4.92	**1.71** **5.00** **5.14**
	0.788 2.96 2.53	0.481 2.44 2.74	1.14 3.81 2.36	0.347 2.10 2.88	0.471 1.95 2.75	**0.174** **1.48** **3.18**
	2.08 5.01 3.22	1.11 3.85 3.50	3.54 6.81 2.99	0.731 3.19 3.68	**0.397** 2.36 **3.94**	0.418 **2.34** 3.92

Table 1: ***Quantitative comparisons:*** *We measure various error metrics (namely, MSE ($\times 10^{-3}$), MCD ($\times 10^{-2}$), and SNR ($\times 10$)) w.r.t. the clean shape for denoising performed with, respectively, APSS [GG07], Bilateral Filtering [HWG*13], RIMLS [OGG09], Graph Laplacian Regularization [ZCN*19], PointCleanNet [RLBG*20], and our approach. While PointCleanNet outperforms the variational approach presented here on a few examples, we systematically exceed (lower MSE and MCD, higher SNR) all other methods.*

be incorporated within our approach. Second, we only perform first order fitting (tangent planes), but higher-order fitting like quadrics could further improve denoising by detecting round parts. Third, our energy formulation could also be improved, for instance by making λ (the smoothness strength) adapted to the detected noise level so as to add spatially just the right amount of normal smoothness — right now, this coefficient can of course be tuned spatially by the user, but an automatic treatment would be highly preferable — or by using the GNC continuation method [BZ03] to accelerate convergence. Other piecewise smoothness energies, such as Willmore's, could also be explored. We have also not explored more advanced treatment of detected sharp features: incorporating a piecewise-smoothness term for feature curves for instance could improve the results tremendously. Having direct access to the line processes (unlike current robust statistics approaches) should render these ideas easier. Finally, we note that the issue of structured noise or outliers (e.g., artificial but structured and repeated clumps of points) is also not addressed by our new approach.

Acknowledgment

We thank the reviewers for their constructive comments. JW was partially funded by IP Paris AI program, JC acknowledges support from the Uber chair in integrated urban mobility, while MD gratefully acknowledges support from Ansys and an Inria Chair.

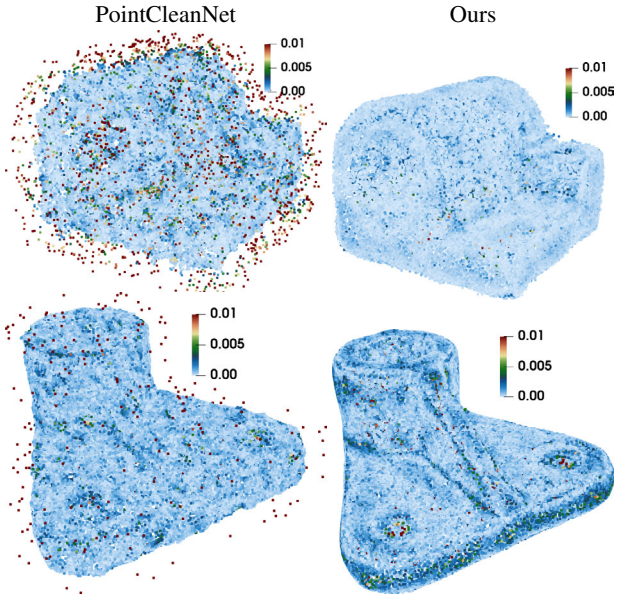


Figure 15: ***Error map:*** *For the* JOINT *and the* PART *models from Fig. 14, we show the "error map" of PointCleanNet ([RLBG*20], left) vs. our results (right), i.e., each point of the denoised results is colored based on its shortest distance to the ground-truth surface. At this large level of noise, many noisy samples are left in the machine-learning based results, which is not the case for ours; but a small inflation of the surface can be seen due to the normal smoothness induced by our energy minimization.*

References

[ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., T. SILVA C.: Computing and rendering point set surfaces. *IEEE Trans. Visualization and Computer Graphics 9*, 1 (2003), 3–15. 2

[AGJ*22] ALLIEZ P., GIRAUDOT S., JAMIN C., LAFARGE F., MÉRIGOT Q., MEYRON J., SABORET L., SALMAN N., WU S., YILDIRAN N. F.: Point set processing. In *CGAL User and Reference Manual*, 5.5 ed. CGAL Editorial Board, 2022. 9

[ALMF19] ARVANITIS G., LALOS A. S., MOUSTAKAS K., FAKOTAKIS N.: Feature preserving mesh denoising based on graph spectral processing. *IEEE Trans. Vis. Comp. Graph. 25*, 3 (2019), 1513–1527. 2

Method Parameter(s)	APSS s	Bilateral k	RIMLS s	GLR k	Ours (k, λ)
FANDISK	5	50	5	50	50, 1
CAD2	15	150	15	150	150, 2
ANCHOR	5	50	5	50	30, 1
ACCESSORY	15	150	15	150	150, 2
JOINT	15	150	15	150	150, 2
PART	15	150	15	150	150, 2

Table 2: *Parameter table for the results shown in Fig. 14. We list the parameters varying through the experiments, and used default parameters if not listed. Note that s stands for the filter scale, and k is the number of neighboring points. The default parameters of our method are $\eta = 5000$, $\mu_l = 5\times 10^{-9}$, $\mu_m = 0.13$. As pointed out in the original article, PointCleanNet is parameter-free, so we iterate their algorithm three times to generate the final results.*

[BJH22] BOTSCH J., JAIN H., HELLWICH O.: Imd-net: A deep learning-based icosahedral mesh denoising network. *IEEE Access 10* (2022), 38635–38649. 2

[BR96] BLACK M. J., RANGARAJAN A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision 19*, 1 (1996), 57–91. 3, 4, 7

[BT74] BEATON A. E., TUKEY J. W.: The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics 16*, 2 (1974), 147–185. 3

[BZ03] BLAKE A., ZISSERMAN A.: The Graduated Non-Convexity Algorithm. In *Visual Reconstruction*. MIT Press, 2003, pp. 131–166. 4, 12

[CCC*08] CIGNONI P., CALLIERI M., CORSINI M., DELLEPIANE M., GANOVELLI F., RANZUGLIA G.: MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference* (2008), Scarano V., Chiara R. D., Erra U., (Eds.). 4, 9, 10

[CCSM11] CHAZAL F., COHEN-STEINER D., MÉRIGOT Q.: Geometric inference for probability measures. *Found. Comput. Math.*, 11 (2011), 733–751. 9

[CDY*20] CHEN S., DUAN C., YANG Y., LI D., FENG C., TIAN D.: Deep unsupervised learning of 3d point clouds via graph topology inference and filtering. *IEEE Trans. Image Processing 29* (2020), 3183–3198. 2

[CGA22] CGAL: *CGAL User and Reference Manual*, 5.5 ed. CGAL Editorial Board, 2022. 8, 10

[CRT04] CLARENZ U., RUMPF M., TELEA A.: Fairing of point based surfaces. In *Computer Graphics International, 2004.* (2004), pp. 600–603. 2

[CWS*20] CHEN H., WEI M., SUN Y., XIE X., WANG J.: Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint. *IEEE Trans. Vis. Comp. Graph. 26*, 11 (2020), 3255–3270. 2

[DCS22] DA T. K. F., COHEN-STEINER D.: Advancing front surface reconstruction. In *CGAL User and Reference Manual*, 5.5 ed. CGAL Editorial Board, 2022. 2, 8

[DCSA*14] DIGNE J., COHEN-STEINER D., ALLIEZ P., DE GOES F., DESBRUN M.: Feature-preserving surface reconstruction and simplification from defect-laden point sets. *Journal of Mathematical Imaging and Vision*, 48(2) (2014), 369–382. 9

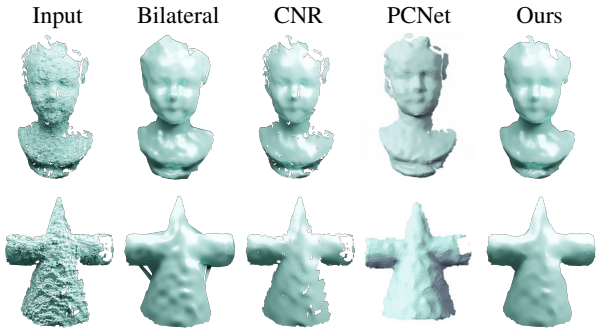


Figure 16: ***Real scans.*** *From scans using Microsoft Kinect® provided in [WLT16], we compare Bilateral Filtering [HWG*13], Cascaded Normal Regression [WLT16], PointCleanNet [RLBG*20], and our approach — the GLR approach [ZCN*19] failed because of the presence of collinear points. While we do not make use of the input mesh connectivity (which helps with normal estimation), our result is comparable to or better than mesh denoising methods, and outperforms the pointset results from bilateral filtering.*

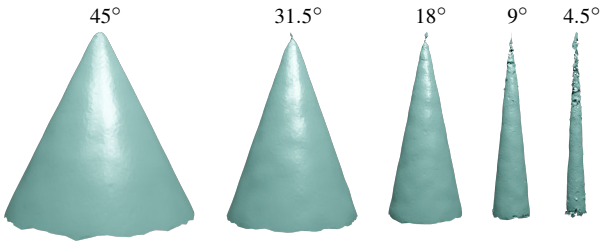


Figure 17: ***Limitations.*** *When applied to cones with added Gaussian noise for a standard derivation equal to 1% of the model diameter, our approach cannot perfectly recover the tip when the cone angle reaches a certain level.*

[DDF17] DIGNE J., DE FRANCHIS C.: The bilateral filter for point clouds. *Image Processing On Line 7* (2017), 278–287. 2

[Dem20] DEMIDOV D.: Amgcl – a c++ library for efficient solution of large sparse linear systems. *Software Impacts 6* (2020), 100037. 8

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), ACM Press/Addison-Wesley Publishing Co., p. 317–324. 1

[DTB06] DIEBEL J., THRUN S., BRUENING M.: A bayesian method for probable surface reconstruction and decimation. *ACM Trans. Graph. 25*, 1 (2006), 39–59. 2

[DVC18] DIGNE J., VALETTE S., CHAINE R.: Sparse geometric representation through local shape probing. *IEEE Trans. Vis. Comp. Graph. 24(7)*, 7 (2018), 2238–2250. 2

[EKS*96] ESTER M., KRIEGEL H.-P., SANDER J., XU X., ET AL.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 'Knowledge Discovery and Data Mining'* (1996), vol. 96(34), pp. 226–231. 7

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. In *ACM SIGGRAPH Proceedings* (2005), pp. 544–552. 2

[FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *ACM Trans. Graph. 22*, 3 (2003), 950–953. 2

[GG84] GEMAN S., GEMAN D.: Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-6*, 6 (1984), 721–741. 3

model	n	k	Eqs. (10)+(6)	Eqs. (9)+(7)+(8)
KITTEN	24956	20	0.192	0.508
		50	0.254	0.769
		100	0.452	1.062
ACCESSORY	50000	20	0.232	1.074
		50	0.375	2.002
		100	0.610	3.136
ARMADILLO	99416	20	0.399	1.857
		50	0.680	2.976
		100	1.132	5.065

Table 3: ***Timings for restricted minimization iterations.*** *We provide the timings of the two steps (local plane fitting, and global piecewise smoothness) involved in one iteration of our global minimization, as a function of the number of samples n and of the number of neighbors k. We observe a linear dependency in both variables. All timings are expressed in seconds.*

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Trans. Graph. 26*, 3 (2007). 9, 11, 12

[GM87] GEMAN S., MCCLURE D. E.: Statistical methods for tomographic image reconstruction. *Bull. Int. Stat. Inst 52*, 4 (1987), 5–21. 3, 4, 5

[Hag01] HAGER W. W.: Minimizing a quadratic over a sphere. *SIAM Journal on Optimization 12*, 1 (2001), 188–208. 6, 8, 15

[HJW*17] HAN X.-F., JIN J. S., WANG M.-J., JIANG W., GAO L., XIAO L.: A review of algorithms for filtering the 3d point cloud. *Signal Processing: Image Communication 57* (2017), 103–112. 1

[HLZ*09] HUANG H., LI D., ZHANG H., ASCHER U., COHEN-OR D.: Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. 28*, 5 (dec 2009), 1–7. 2

[HRR19] HERMOSILLA P., RITSCHEL T., ROPINSKI T.: Total denoising: Unsupervised learning of 3d point cloud cleaning. In *International Conference on Computer Vision* (2019), pp. 52–60. 2

[HRRS86] HAMPEL F. R., RONCHETTI E. M., ROUSSEEUW P. J., STAHEL W. A.: *Robust Statistics: The Approach Based on Influence Functions*. Wiley Series in Probability and Statistics. John Wiley & Sons, 1986. 3

[HS13] HE L., SCHAEFER S.: Mesh denoising via l_0 minimization. *ACM Trans. Graph. 32*, 4 (2013). 2

[Hub81] HUBER P. J.: *Robust Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, 1981. 3

[HWG*13] HUANG H., WU S., GONG M., COHEN-OR D., ASCHER U., ZHANG H.: Edge-aware point set resampling. *ACM Trans. Graph. 32*, 1 (2013), 1–12. 4, 9, 10, 11, 12, 13

[JDD03] JONES T. R., DURAND F., DESBRUN M.: Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph. 22*, 3 (2003), 943–949. 2

[KPZK17] KNAPITSCH A., PARK J., ZHOU Q.-Y., KOLTUN V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics 36*, 4 (2017). 9

[LCOLTE07] LIPMAN Y., COHEN-OR D., LEVIN D., TAL-EZER H.: Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph. 26*, 3 (jul 2007). 2

[LEL14] LOZES F., ELMOATAZ A., LÉZORAY O.: Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Trans. Image Processing 23*, 9 (2014), 3896–3909. 2

[Lev98] LEVIN D.: The approximation power of moving least-squares. *Math. Comput. 67*, 224 (oct 1998), 1517–1531. 2

[LH20] LUO S., HU W.: Differentiable manifold reconstruction for point cloud denoising. In *ACM International Conference on Multimedia* (2020), pp. 1330–1338. 2

[LP05] LANGE C., POLTHIER K.: Anisotropic smoothing of point sets. *Computer Aided Geometric Design 22*, 7 (2005), 680–692. 2

[LSK*10] LI B., SCHNABEL R., KLEIN R., CHENG Z., DANG G., JIN S.: Robust normal estimation for point clouds with sharp features. *Computers & Graphics 34*, 2 (2010), 94–106. 2

[LSL*22] LU X., SCHAEFER S., LUO J., MA L., HE Y.: Low rank matrix approximation for 3d geometry filtering. *IEEE Trans. Vis. Comp. Graph. 28*, 4 (2022), 1835–1847. 2

[LWC*18] LU X., WU S., CHEN H., YEUNG S.-K., CHEN W., ZWICKER M.: GPF: GMM-inspired feature-preserving point set filtering. *IEEE Trans. Vis. Comp. Graph. 24*, 8 (2018), 2315–2326. 2

[LXJF13] LIAO B., XIAO C., JIN L., FU H.: Efficient feature-preserving local projection operator for geometry reconstruction. *Computer-Aided Design 45*, 5 (2013), 861–874. 2

[MDGD*10] MULLEN P., DE GOES F., DESBRUN M., COHEN-STEINER D., ALLIEZ P.: Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum 29*, 5 (2010), 1733–1741. 3

[OGG09] OEZTIRELI A. C., GUENNEBAUD G., GROSS M.: Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum* (2009). 2, 9, 11, 12

[PCS21] PEYRÉ G., CUTURI M., SOLOMON J.: Computational optimal transport. In *SIGGRAPH Asia '21 Courses* (2021). 4

[PFVM21] PISTILLI F., FRACASTORO G., VALSESIA D., MAGLI E.: Learning robust graph-convolutional representations for point cloud denoising. *IEEE Journal of Selected Topics in Signal Processing 15*, 2 (2021), 402–414. 2

[PG01] PAULY M., GROSS M.: Spectral processing of point-sampled geometry. In *Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, pp. 379–386. 2

[PKG02] PAULY M., KOBBELT L., GROSS M.: *Multiresolution Modeling of Point-Sampled Geometry*. Tech. rep., ETH Zurich, 2002. 2

[RLBG*20] RAKOTOSAONA M.-J., LA BARBERA V., GUERRERO P., MITRA N. J., OVSJANIKOV M.: Pointcleannet: Learning to denoise and remove outliers from dense point clouds. *Computer Graphics Forum 39*, 1 (2020), 185–203. 2, 9, 10, 11, 12, 13

[SB04] SHEN Y., BARNER K.: Fuzzy vector median-based surface smoothing. *IEEE Trans. Vis. Comp. Graph. 10*, 3 (2004), 252–265. 2

[SRML07] SUN X., ROSIN P. L., MARTIN R., LANGBEIN F.: Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. Comp. Graph. 13*, 5 (2007), 925–938. 2

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, pp. 351–358. 1

[vL22] VAN LANKVELD T.: Scale-space surface reconstruction. In *CGAL User and Reference Manual*, 5.5 ed. CGAL Editorial Board, 2022. 8, 9

[WKZ*16] WOLFF K., KIM C., ZIMMER H., SCHROERS C., BOTSCH M., SORKINE-HORNUNG O., SORKINE-HORNUNG A.: Point cloud noise and outlier removal for image-based 3d reconstruction. In *International Conference on 3D Vision* (2016), pp. 118–127. 9

[WLL*22] WANG S., LIU X., LIU J., LI S., CAO J.: Deep patch-based global normal orientation. *Computer-Aided Design 150* (2022). 3

[WLT16] WANG P.-S., LIU Y., TONG X.: Mesh denoising via cascaded normal regression. *ACM Trans. Graph. 35*, 6 (nov 2016). 1, 2, 9, 10, 13

[WYL*14] WANG R., YANG Z., LIU L., DENG J., CHEN F.: Decoupling noise and features via weighted ℓ_1-analysis compressed sensing. *ACM Trans. Graph. 33*, 2 (2014). 2

[XF21] XU Z., FOI A.: Anisotropic denoising of 3d point clouds by aggregation of multiple surface-adaptive estimates. *IEEE Trans. Visualization and Computer Graphics 27*, 6 (2021), 2851–2868. 2

[YLF*18] YU L., LI X., FU C.-W., COHEN-OR D., HENG P.-A.: EC-Net: an edge-aware point set consolidation network. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 386–402. 2

[YRS*18] YADAV S. K., REITEBUCH U., SKRODZKI M., ZIMMERMANN E., POLTHIER K.: Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics. *Computers & Graphics 74* (2018), 234–243. 3

[YSZP20] YADAV S. K., SKRODZKI M., ZIMMERMANN E., POLTHIER K.: Surface denoising based on normal filtering in a robust statistics framework. *CoRR abs/2007.00842* (2020). 3, 4

[ZCD20] ZHANG S., CUI S., DING Z.: Hypergraph spectral analysis and processing in 3d point cloud. *IEEE Trans. Image Processing 30* (2020), 1193–1206. 2

[ZCN*19] ZENG J., CHEUNG G., NG M., PANG J., YANG C.: 3D point cloud denoising using graph Laplacian regularization of a low-dimensional manifold model. *IEEE Trans. Image Processing 29* (2019), 3474–3489. 9, 10, 11, 12, 13

[ZDZ*15] ZHANG W., DENG B., ZHANG J., BOUAZIZ S., LIU L.: Guided mesh normal filtering. *Computer Graphics Forum 34*, 7 (2015), 23–34. 2

[ZLQH21] ZHANG D., LU X., QIN H., HE Y.: PointFilter: Point cloud filtering via encoder-decoder modeling. *IEEE Trans. Vis. Comp. Graph. 27*, 3 (2021), 2015–2027. 2

[ZSL*22] ZHOU L., SUN G., LI Y., LI W., SU Z.: Point cloud denoising review: from classical to deep learning-based approaches. *Graphical Models 121* (2022), 101140. 1

Appendix A: Non-linear local plane fitting

Solving for Eq. (10) amounts to minimizing a quadratic function over a sphere. The approach of [Hag01] to solving this nonlinear problem is as follows. First, we introduce a Lagrangian multiplier γ for the norm constraint. The optimality conditions with respect to $\mathbf{h}$ and γ are thus expressed as:

$$\begin{cases} \mathbf{A}\mathbf{h} - \mathbf{b} + \gamma\mathbf{h} = \mathbf{0} \\ \|\mathbf{h}\| = 1 \end{cases}.$$

From the eigen-decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{U}^t$, we create a new variable $\mathbf{z}$ by substituting $\mathbf{h} = \mathbf{U}\mathbf{z}$. Since $\mathbf{U}$ is orthonormal, the constraint on $\mathbf{z}$ remains a unit norm; thus the above equations reduce to

$$\begin{cases} \Lambda\mathbf{z} - \mathbf{U}^t\mathbf{b} + \gamma\mathbf{z} = \mathbf{0} \\ \|\mathbf{z}\| = 1 \end{cases},$$

where Λ is a diagonal matrix containing $\mathbf{A}$'s eigenvalues. Denoting $\mathbf{g} = \mathbf{U}^t\mathbf{b}$ and solving for $\mathbf{z}$ from the first equation, we get

$$\mathbf{z} = (\Lambda + \gamma\mathbf{I})^{-1}\mathbf{g} = \left(\frac{g_1}{\lambda_1+\gamma} \quad \frac{g_2}{\lambda_2+\gamma} \quad \frac{g_3}{\lambda_3+\gamma} \quad \frac{g_4}{\lambda_4+\gamma}\right)^t. \tag{11}$$

Plugging this into the second equation results in a nonlinear equation in γ, namely:

$$\left(\frac{g_1}{\lambda_1+\gamma}\right)^2 + \left(\frac{g_2}{\lambda_2+\gamma}\right)^2 + \left(\frac{g_3}{\lambda_3+\gamma}\right)^2 + \left(\frac{g_4}{\lambda_4+\gamma}\right)^2 = 1. \tag{12}$$

Given g_i and λ_i, the solution can be degenerated as discussed by [Hag01], but trivial solutions are easy to find. For non-degenerated cases, finding a root of the above equation between the bounds given in [Hag01] yields a solution, from which $\mathbf{z}$, and then $\mathbf{h}$ are directly deduced.