Similarity based filtering of point clouds

Julie Digne
INRIA Sophia-Antipolis
julie.digne@inria.fr

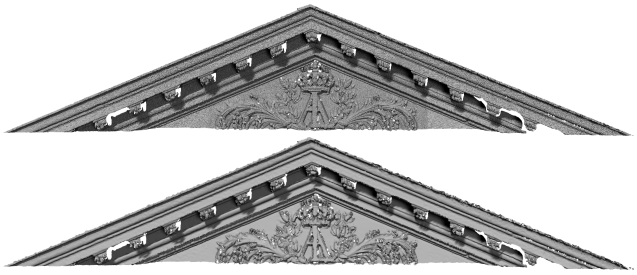


Figure 1. A point cloud and its denoising with our method (AIM@shape repository)

Abstract

Denoising surfaces is a a crucial step in the surface processing pipeline. This is even more challenging when no underlying structure of the surface is known, id est when the surface is represented as a set of unorganized points. In this paper, a denoising method based on local similarities *is introduced. The contributions are threefold: first, we do not denoise directly the point positions but use a low/high frequency decomposition and denoise only the high frequency. Second, we introduce a local surface parameterization which is proved stable. Finally, this method works directly on point clouds, thus avoiding building a mesh of a noisy surface which is a difficult problem. Our approach is based on denoising a height vector field by comparing the neighborhood of the point with neighborhoods of other points on the surface. It falls into the non-local denoising framework that has been extensively used in image processing, but extends it to unorganized point clouds.*

1. Introduction

When dealing with acquired data sampled on a real object surface, *denoising* is a crucial step of the processing pipeline. The idea behind denoising is to separate the input signal between a real shape corresponding to meaningful measures of the surface and a noisy component. This component is almost always considered additive because of the simplicity and relative good result this approximation provides. Yet a huge problem of the signal denoising methods is to distinguish between the shape and the noise: in many cases some signal features have very similar frequential properties to the noise (e.g. sharp features). The difficulty lies in robustly estimating the separation so as to remove noise without removing shape characteristics such as sharp features or geometric textures.

The remainder of this paper is divided as follows: section 2 gives relevant references on shape denoising. Section 3 gives an overview of the method. Sections 4, 5, 6 explain the three main steps of the algorithm. Section 7 explains implementation choices. Finally Section 8 presents the results on various point clouds.

2. Related work

Multiple isotropic filters have been proposed and studied. Those filters remove all details and sharp features since they denoise undistinctively noise and features: clearing the shape of all its high frequencies. To avoid that, different strategies have been proposed. The bilateral filter, for example, uses non isotropic neighborhoods to preserve features, the neighborhood becomes adaptive: not taking into account only distance between the points but also shape properties of the neighborhoods of the points ([11], [5]). Another type of approach for this problem is to find an equation of the locally underlying surface and project the point cloud on it. This approach was developped extensively through the MLS (Moving Least Squares) method ([18], [17]). This type of method performs a local polynomial regression and projects each point on it. A lot of variants were proposed to better preserve edges and sharp features with very good results ([12], [10], [23]). Our method can also be seen as a projection of the point on an underlying surface which is not explicitely modeled.

The denoising problem is common to many fields and in particular image processing. For example the bilateral filter for gray-scale and color images was introduced in [27]. Observing staircasing effects of this denoising filter and all types of neighborhood filters in [3], a non-local filter was proposed in [4] denoising an image pixel not only with its neighbors but with all the pixels in the image, each pixel be-

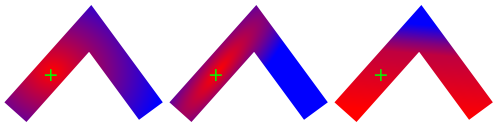


Figure 2. Different types of neighborhood used for filtering point clouds (from left to right: isotropic, bilateral, nonlocal). Red points contribute a lot to the denoising of the central point (green cross) while blue points barely contribute to the denoising.

ing weighted in the sum by a similarity factor: if the pixel is very similar to the point being denoised it will influence the denoising a lot, otherwise it will barely contribute to it.

The non-local filter can be seen as the limit of the neighborhood filters (see Fig. 2) since the neighborhood used for the denoising is no longer the euclidean local neighorhood but a neighborhood in the space of patches. This idea was used to make the computation faster ([1], [24]), by working directly in the space of image patches.

When trying to use the non-local framework for surface denoising, one of the difficulties lies in finding a good local descriptor. Such a descriptor should be intrinsic so that comparing descriptors is relevant. Building local descriptors for shapes is also a widely studied problem, though in a somewhat different context: local descriptors are usually devised for shape matching or shape retrieval purposes (e.g. [13],[20], [15] and [28]). In this paper we will also introduce an intrinsic descriptor that aims at providing a good local signature of the shape.

A non-local denoising for meshes has been proposed in [29] using as descriptor a local regression of the geometry by radial basis functions. But no precise way of parameterizing the local frames is given. As we will see local parameterization is crucial: if two similar patches are not parameterized in the same way then their comparison distance might be big: this actually restricts the set of similar points and makes the comparison less relevant. We address this problem in this paper. In [22], a variational framework is introduced for denoising meshes: it is based on denoising the mean curvature field and then solving for a minimization of a two-term functional: a data-closeness term and a term aiming at getting a curvature field close to the denoised curvature field. In a way, this method is not far from ours since we will also denoise a vector field.

For point clouds it is even more difficult since we lack any structuring information. Yet, in [6], such a denoising is proposed. It is based on considering the local covariance matrix of the points (as done in [25]) and parameterizing the local neighborhood with the two eigenvectors corresponding to the two largest eigenvalues. A MLS (Moving Least Squares) regression is performed and the vector of the coefficients is used as descriptor. Once again the dependency on the local frame parameterization is not analyzed. We will show that this parametrization choice is not stable enough.

This framework has been also used in recent works for surface analysis. In [30], a point cloud large scale redundancy is used to enhance details using local self-similarity. It is indeed a patch-based approach to the processing of surfaces but in a way that requires much larger patches (to be able to capture details more precisely) than in our method. Next section gives an overview of the method.

3. Overview

The input data of the algorithm is an unorganized point cloud. In comparison with most denoising algorithms we will not require any mesh or connectivity information. The point cloud will rather be processed by local neighborhoods analysis. The method introduced in this paper relies heavily on the definition of comparable local descriptors. The overall idea is the following:

1. Decompose the input surface into a smooth base and a height vector field $\vec{V}$ (Section 4)
2. For each point p compute a local descriptor $\mathcal{P}(p)$ encoding the variation of the height vector field around the point (Section 5)
3. Compute the denoised height vector for each point based on the similarity between the descriptors (Section 6)
4. Update all point positions by adding the denoised height field to the smooth surface.

The idea behind step 1, is that if the surface is decomposed into a smooth part and a vector field containing the sharp features, the details, but also the noise, then one needs only to denoise the vector field and not the smooth part. This decomposition can be achieved by a smoothing method filtering indiscriminately sharp edges and noise. This way, the vector field will contain all high frequencies. The most common isotropic filter, namely the mean curvature motion is used here, with the implementation described in [9] for point clouds. The next step (step 2) builds a local descriptor of the height field around each point p. The idea is to capture the variation of the height field. We encode these variations by sampling the height field on a 2D grid centered at point p. This 2d grid is aligned with stable vectors parameterizing the tangent space of the surface at p. Step 3 uses the local descriptors to denoise the height field: each point height vector is denoised as a weighted sum of the height vectors of other points of the shape. The weight of each point q in the sum for denoising point p is *similarity-based*: it depends on the distance between the descriptors of p and q. If the descriptors are very similar then the weight is high otherwise it is low. The idea behind this denoising is that points in areas similar to the neighborhood of p will contribute more to its denoising than points in areas very different from the neighborhood of p. Finally step 4 yields the denoised surface by combining the smooth surface found in step 1 with the denoised height field of step

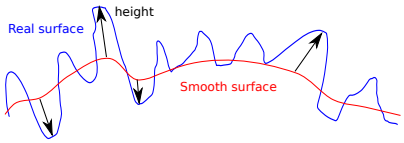


Figure 3. Example of a decomposition of an input surface (in blue) into a smooth surface (red) and a height vector field (black arrows)

3. The next sections will describe more precisely each of those steps.

4. Considering the image as a smooth surface + height

When dealing with a complex surface it is useful to consider it as a smooth surface and a high frequency term containing all details, features but also noise. This approach developped in [9] will be used here: we decompose the surface $\mathcal{S} = \mathcal{S}_{smooth} + \vec{V}$, where $\vec{V}$ is the high frequency vector field (alternatively one could also rely on [26]).

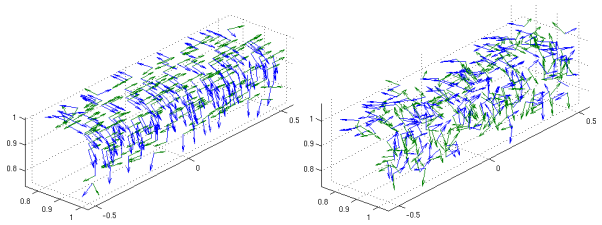
To achieve decomposition of the surface into a smooth basis and a high frequency vector field (Figure 3), the mean curvature motion filter is iterated over the shape $\mathcal{S}$ until all noise is removed, yielding a smoothed shape $\mathcal{S}_{smooth}$. At this point all sharp features are also removed, but those sharp features will be recovered when denoising the high frequency. This way, the high frequency is the residual of the smoothing: $\vec{V} = \mathcal{S} - \mathcal{S}_{smooth}$. Since there is a one to one correspondence between the points of the initial surface $\mathcal{S}$ and the points of the smoothed surface $\mathcal{S}_{smooth}$, one can compute for each point p of $\mathcal{S}$ the high frequency $\vec{V}(p)$. Indeed, $p = p_S + \vec{V}(p)$, where $p \in \mathcal{S}$, $p_S \in \mathcal{S}_{smooth}$ and, finally $\vec{V}(p) = p - p_S$.

In this paper, shape denoising will be achieved by denoising $\vec{V}(p)$. To denoise this high frequency vector we will use a *non-local strategy*: filtering high frequencies by similarity with other points regardless (or almost regardless) or where these points are.

5. Building local descriptors

When building a local descriptor, one needs either to make the descriptor parameterization-free or find a robust enough local parameterization of the surface. Most descriptors rely on the definition of at least a normal direction for all surface points, but try to avoid parameterization in the local tangent plane (which would involve computing principal directions for example, order 2 derivatives of the surface). For example, spin images introduced in [13] used a kind of occupancy 2d grid parameterized by the height $\langle q-p, \vec{n}(p)\rangle$ and the radial distance $(\|q-p\|^2 - \langle q-p, \vec{n}(p)\rangle^2)^{1/2}$. Other approaches have been proposed to make the descriptor invariant to rotations ([14]).

To get a parameterization of the tangent plane, a solution would be to compute a MLS surface and derive the principal directions from the analytical formulas. Here we will rather rely on recent results developped in [8].



(a) Ours (b) Deschaud et al.

Figure 4. Local frames in the neighborhood of a cube edge, this parameterization (4(a)) is far more stable than the one chosen by Deschaud et al. [6] (4(b))

5.1. Choice of a local coordinate system

In [8], it was proven that robust and stable equivalents of the principal directions on a surface could be found by simple covariance analysis. If an oriented normal is known at each point in the neighborhood of a point, then one can compute the centered covariance of these normals yielding a 3×3 matrix. Let $(p_i, \vec{n}_i)$ be a set of N points with known normal direction, then, on can compute the mean $\vec{n}_m$ and centered covariance matrix C of the normals.

$$\vec{n}_m = \frac{1}{N}\sum_{i=1}^{N} \vec{n}_i \text{ and } C = \frac{1}{N}\sum_{i=1}^{N} (\vec{n}_i - \vec{n}_m)^T \cdot (\vec{n}_i - \vec{n}_m).$$

The eigenvectors and eigenvalues of C are proven ([8]) to be respectively tangent to the principal directions and proportional to the squared principal curvatures of the smooth underlying surface. This formulation is practical because it only uses covariance analysis. Figure 4 shows such local coordinate frames around the edge of a noisy cube (the same cube as shown in Fig. 7(a)).

Notice that these directions due to numerical uncertainties might not be really orthogonal to the normal direction. We improve this by computing a stabler normal direction (by Principal Component Analysis of smooth surface, see [21] and [2]), and projecting the axes found by covariance analysis on this plane. At this point, we have three orthogonal axes for each point of the point cloud. One still needs to turn them into a stable orthonormal coordinate system: either $(\vec{t}_1, \vec{t}_2, \vec{n})$ or $(-\vec{t}_2, -\vec{t}_1, \vec{n})$. This last ambiguity will be removed after the descriptor is built, and the descriptor will then be adapted to fit the right frame denoted $(u_x(p), u_y(p), u_z(p))$ hereafter.

5.2. Describing the local neighborhood

The descriptor used in this work is very similar to the snapshots descriptors described in [20] and related to Spin

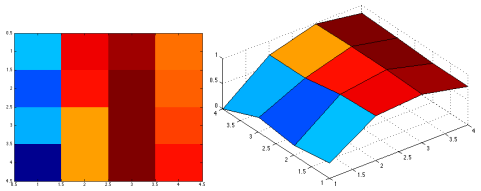


Figure 5. The 4×4 patch of a point located near an edge , color valued (left) and 3-dimensional (right). One sees distinctively the edge orthogonal to the first principal direction.

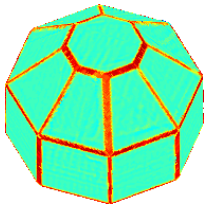


Figure 6. Similarity of all points of the shape to a point on an edge of the shape (red if the point is very similar to the edge point and blue otherwise).

Images described in [13]. Around each point, we build a small image describing the neighborhood, a local height image (the regular sampling on a grid of the local graph). By expressing the coordinates of each neighbor q in the local coordinate frame, we have a set of height values for irregularly sampled points on a plane. From this set of irregularly sampled heights, we interpolate for missing data on the grid using radial basis functions. The radius of the neighborhood and the size of the grid are very small so that the descriptors are very local. Figure 5 shows an example of a patch near an edge of a diamond shape (same shape as used in figures 6 and 9). The ambiguity mentioned in section 5.1 is finally removed by orientating the two tangent plane axes such that the standard deviation in the first quadrant of the tangent plane relatively to the two axes orientation choice is the highest, and the descriptor is adapted to this potential axes change.

5.3. Comparing local descriptors

Once the descriptor is built, one has to compare them in order to quantify their similarity. Since the descriptor we used is a pose-independent grid values, we will compare two descriptors by computing their L^2 distance. Computing distribution distances would probably yield better results but at the cost of additional computation. An example of the distance of all shape descriptors to a single point descriptor is shown on fig 6. This figure shows that, as expected, with the similarity measure introduced above points on edges are very self-similar, whereas a point on a more planar part is less similar to an edge point (and will therefore not contribute to the denoising of an edge point).

6. Denoising algorithm

The last part of the algorithm is to compute the denoised height field and use it to get the final denoised surface.

Let p be a point in $\mathcal{S}$, p_S its corresponding point on the smooth surface $\mathcal{S}_{smooth}$ and $\vec{V}(p) = p - p_S$. We denote by $\mathcal{P}(p)$ the local descriptor of p expressed in the local intrinsic coordinate system $u_x(p), u_y(p), u_z(p)$. The update equations for point p then writes:

$$p' = p_S + \delta p \text{ with: } \delta p_x = \frac{\sum_{q \in \mathcal{S}} w_{pq} \langle \vec{V}(q), u_x(q) \rangle}{\sum_{q \in \mathcal{S}} w_{pq}}$$

where w_{pq} is the *similarity* of points p and q.

$$w_{pq} = \exp - \frac{dist(\mathcal{P}(p), \mathcal{P}(q))^2}{\sigma^2}.$$

We have, similarly, $\delta p_y = \frac{\sum_{q \in \mathcal{S}} w_{pq} \langle \vec{V}(q), u_y(q) \rangle}{\sum_{q \in \mathcal{S}} w_{pq}}$ and $\delta p_z = \frac{\sum_{q \in \mathcal{S}} w_{pq} \langle \vec{V}(q), u_z(q) \rangle}{\sum_{q \in \mathcal{S}} w_{pq}}$. Notice that the obtained update $\delta p = (\delta p_x, \delta p_y, \delta p_z)$ is expressed in the local coordinate system $(u_x(p), u_y(p), u_z(p))$.

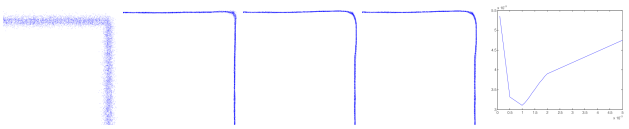
7. Implementation and parameters

7.1. Implementation choices

An octree is used to access quickly points and ball neighborhood of points. This choice might not be the best one for memory usage but it divides the space in subspaces which will be useful for parallelization: indeed one can process some of the octree cells at the same time without risking memory conflicts. Using the parallelization, on a 8-core processor, the denoising of a 5 million points suface, with an octree of depth 7 yields a computation time of 2 minutes, which is enough since we are not aiming at real-time applications. This could be made much faster using similar accelerations as in [1]. For computation time reason we do not compare a point to all points in the surface, but only to a certain subset of points: this subset is taken to be all points included in octree cells adjacent to the cell where the point is located at maximum depth (the maximum depth is set in accordance with the processing radius chosen), thus avoiding a quadratic complexity.

7.2. Tuning the parameters

The parameters of this method is the following: processing radius, number of iterations for determining the high frequency term, grid size and weight of the similarity weighting gaussians. The processing radius is set so that a ball centered on each point contains 30 points (average). The size of the local descriptor should remain small, the typical value used is 4×4 or 3×3. The real parameter



(a) Original (b) 0.01 (c) 0.1 (d) 1 (e) Error

Figure 7. Evolution of the denoising with the parameter σ, when σ is huge the denoising is brutal, whereas a small σ will have almost no effect on the shape. Fig 7(e) show the evolution of the measured error with the σ parameter

one should play with is the standard deviation of the gaussian for the similarity weight. If the value is big then, the similarity weight will be less discriminant so that the filter will behave more like an isotropic filter. On the contrary if the value of the deviation is small then the point cloud will barely be denoised. Figure 7 shows the results of the denoising of a cube with 3 different values of the standard variation σ, the rest of the parameters remaining unchanged. On the same shape, Figure 7(e) shows the measured error of the point cloud with respect to an increasing σ parameter: at first the error only decreases but when the modification due to the denoising (edge smoothing for higher values of σ) are greater than the benefits (noise reduction) the error increases again.

8. Results

Throughout this paper we are dealing with point clouds, but in order to better visualize the results we will generate meshes from the points using a non-smoothing, non-undersampling meshing method which allows for the exact visualization of the data ([9]) and render this mesh by ray tracing. We compare results given by our algorithm with other methods namely the bilateral filter described in [11], algebraic point set surface projector [12] (as a representative of the point set surfaces denoising method) and Deschaud et al. non-local means for point clouds ([6]). Though, originally, the bilateral filter of [11] was designed for meshes, it is straightforward to extend it to point clouds using ball neighborhoods instead of 1-ring neighborhood. This method updates a point by comparing it to its neighbors. Instead of considering a usual isotropic neighborhood, it uses an adapted neighbohood ensuring that points are denoised only with neighbors lying on the same side of a sharp crease. Such a simple extension is not possible for the non local methods for meshes described in [29] or [22], making the comparison impossible.

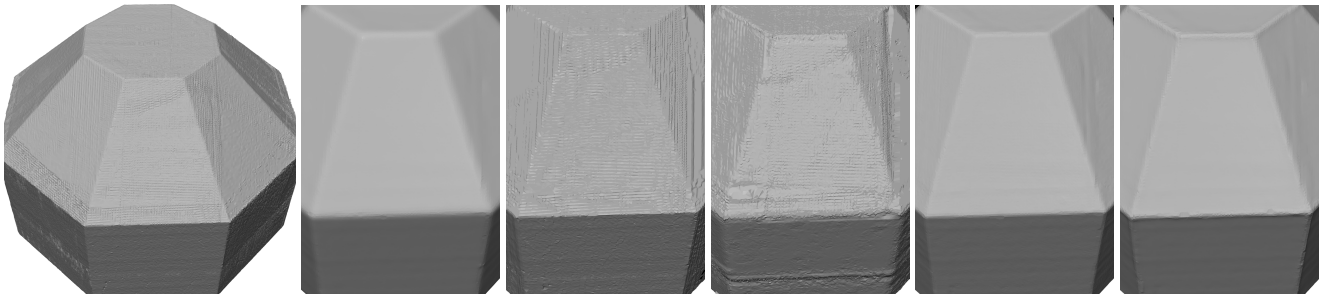
Quantatively on artificial shapes such as a cube one can estimate the RMSE (Root Mean Square Error) of the denoising methods compared. To have a meaningful comparison, the filters must be calibrated: first the same neighborhood size is used for all filters. All filters have a parameter (a gaussian standard deviation) which indicates either how similarity tolerant the filter (for Deschaud et al. and our method) or how tolerant to tangent plane deviation the bilateral filter is. This parameter is the key to the filter success. The RMSE curves exhibit all the same shape as in figure 7(e), we therefore get experimentally for all filters the RMSE minimum. The results are shown on figure 8.

Shape	Bilateral	Deschaud et al.	Ours
Noisy cube	0.0047	0.0048	0.0031
Noisy cylinder	0.002	0.0018	0.0017

Figure 8. Comparison of minimum RMSE found for three filters using the same neighborhood radius. When denoising a smooth shape all three methods perform equally well but for sharp edges, the similarity based method (ours) get a better denoising quantitative result.

Figure 9 shows a real shape with natural acquisition noise and sharp creases. On such simple geometric shape, the bilateral filter and our similarity based filter tend to work equally well. By construction of the bilateral filter this good behavior is not surprising: it is especially designed to preserve points around an edge. Mean curvature motion removes all sharp features and other methods fail to preserve sharp features as well as remove the noise. Next experiments use point sets from the stanford FUR database ([16]), the Farman Institute high precision dataset [7] and the AIM@SHAPE repository acquired using triangulation or range laser scanner. Fig 10 shows the performance of the filter in the case of a rather smooth surface with only some engravings. It shows clearly that the details are preserved by the filter while the rest of the shape is denoised. We also refer the reader to figure 1 and 11 for the denoising of a point cloud acquired with a range laser scanner showing how the denoising allows for detail preservation even for noisy shapes with boundaries, noise and highly variable sampling. On figure 12, we show the denoising of a raw scan and compare our filter with the bilateral filter. To make a meaningful comparison, we chose a parameter for the bilateral filter to have a visually similar result in the smooth areas, and compare the effect on the details and sharp areas. This experiment shows that the proposed non-local filter preserves much better the thin structures of the point cloud, while removing noise in the smoother areas. Finally, figure 13 shows the denoising of a point cloud with added gaussian noise using the non-local filter and the bilateral filter: while the bilateral filter allows for the recovery of the pyramid edges, the similarity filter does a far better job for recovering the engravings of the shape.

Limitations Our algorithm is expected to behave well when dealing with reasonable point densities. In case of low densities, the descriptors may fail to capture a local property of the shape. The algorithm can also generate artefacts near open surface boundaries. Another limitation for this algo-



(a) Initial shape (b) Mean Curvature (c) Algebraic PSS (d) Deschaud et al. (e) Bilateral Filter (f) Ours

Figure 9. From left to right: a shape with natural acquisition noise, its denoising via Mean Curvature Motion, projection on the algebraic point set surface [12], Deschaud et al. point cloud non local means ([6]), bilateral filter and similarity based denoising (our method) . In the case of a geometric shape, both bilateral and non-local perform well, though the non-local filter tends to preserve better the creases. The algebraic point set surface projection tends to create structures due to the irregular sampling. The method from Deschaud et al. ([6]) fails to remove the noise and still removes the sharp corners.

Figure 11. Denoising of a data set acquired with a range laser scanner (left: original, middle: our denoising, right: bilateral filter).

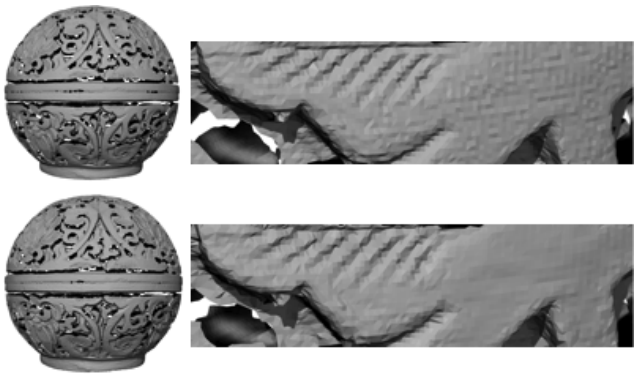


Figure 10. Denoising of a pointset with boundaries, details and holes (shape courtesy of Laurent Saboret, INRIA (AIM@shape repository), the pointset used in this experiment is the set of vertices of the raw mesh. Top: intial scan, bottom: our denoising.

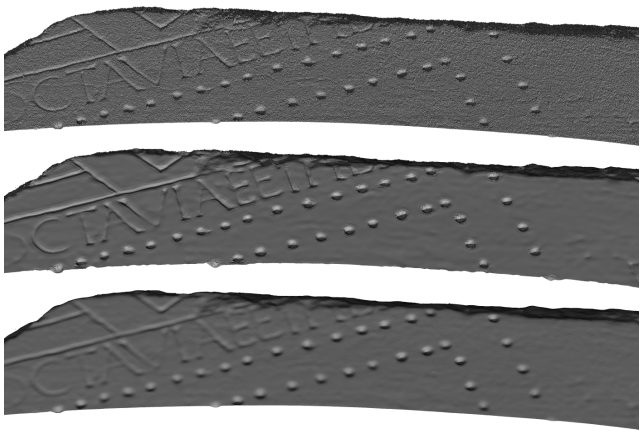


Figure 12. Denoising a raw scan of fragment 31u of the forma urbis romae project : initial fragment; denoising with our method; bilateral denoising

rithm lies in the case of a very structured noise: in that case, the algorithm can not distinguish between texture and noise and will likely enhance the noise pattern.

9. Conclusion

The algorithm introduced proposes a similarity-based approach to the analysis of surfaces. Thanks to a reliable intrinsic descriptor we are able to overcome the usual limitations of non-local filtering methods for surfaces. In addition, a key feature of our method is that it is independent of a surface mesh: it can work directly on point clouds, which is useful, since building a mesh of a noisy point cloud is never easy, whereas building a mesh of a properly denoised

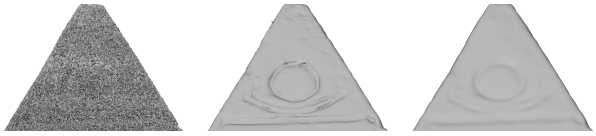


Figure 13. Denoising a pyramid (with added noise) from the farman institute 3D point sets : gaussian pyramid with added gaussian noise; denoising with our method; bilateral denoising

shape is well understood. A possible extension for this work would be to use the filter as a projector onto the surface, in a spirit similar to [19] for example.

Acknowledgments This work was partially funded by the European Research Council (ERC Starting Grant 'Robust Geometry Processing', Grant agreement 257474). The author thanks Prof. Marc Levoy for the data used in fig 12 (property of the Sovraintendenza of Rome and of Stanford University), and Délégation Générale de l'Armement for fig. 11.

References

[1] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph.*, 28:21:1–21:12, July 2009. 2, 4

[2] J. Berkmann and T. Caelli. Computation of surface geometry and segmentation using covariance techniques. *IEEE PAMI*, 16(11):1114–1116, 1994. 3

[3] A. Buades, B. Coll, and J. Morel. The staircasing effect in neighborhood filters and its solution. *IEE TIP*, 15(6):1499–1505, June 2006. 1

[4] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005. 1

[5] P. Choudhury and J. Tumblin. The trilateral filter for high contrast images and meshes. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 5, USA, 2005. ACM. 1

[6] J.-E. Deschaud and F. Goulette. Point cloud non local denoising using local surface descriptor similarity. IAPRS, 2010. 2, 3, 5, 6

[7] J. Digne, N. Audfray, C. Lartigue, C. Mehdi-Souzani, and J.-M. Morel. Farman Institute 3D Point Sets - High Precision 3D Data Sets. *Image Processing On Line*, 2011. 5

[8] J. Digne and J.-M. Morel. A numerical analysis of differential operators on raw point clouds. *preprint CMLA*, 2011-03, 2011. submitted to SINUM. 3

[9] J. Digne, J.-M. Morel, C. Mehdi-Souzani, and C. Lartigue. Scale space meshing of raw data point sets. *Computer Graphics Forum*, 6:1630–1642, September 2011. 2, 3, 5

[10] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, 2005. 1

[11] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graph.*, 22(3):950–953, 2003. 1, 5

[12] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM Trans. Graph.*, 26, 2007. 1, 5, 6

[13] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE PAMI*, 21:433–449, 1999. 2, 3, 4

[14] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proc. SGP 03*, pages 156–164, 2003. 3

[15] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Symmetry descriptors and 3d shape matching. In *SGP '04*, pages 115–123, NY, USA, 2004. ACM. 2

[16] D. Koller, J. Trimble, T. Najbjerg, N. Gelfand, and M. Levoy. Fragments of the city: Stanford's digital forma urbis romae project. In *Proc. Third Williams Symposium on Classical Architecture, Journal of Roman Archaeology Suppl. 61*, pages pp. 237–252, 2006. 5

[17] D. Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998. 1

[18] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–49. Springer-Verlag, 2003. 1

[19] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, 26, 2007. 7

[20] S. Malassiotis. Snapshots: A novel local surface descriptor and matching algorithm for robust 3d surface alignment. *IEEE PAMI*, 29(7):1285–1290, 2007. 2, 3

[21] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Symposium on Computational Geometry*, pages 322–328, 2003. 3

[22] S. Morigi, M. Rucci, and F. Sgallari. Nonlocal surface fairing. In *Scale Space and Variational Methods in Computer Vision*, volume 6667 of *LNCS*, pages 38–49. Springer, 2012. 2, 5

[23] A. C. Oztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. *CGF*, 28:493–501(9), 2009. 1

[24] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision*, 81(1):24–52, 2009. 2

[25] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proc. VIS '02*, pages 163–170, USA, 2002. IEEE. 2

[26] M. Pauly, L. P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Trans. Graph.*, 25(2):177–193, 2006. 3

[27] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. ICCV '98*, page 839, Washington, DC, USA, 1998. IEEE. 1

[28] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proc. ECCV 2010*, pages 356–369. Springer, 2010. 2

[29] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In *SMI '06*. IEEE, 2006. 2, 5

[30] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen. Non-local scan consolidation for 3d urban scenes. *ACM Transactions on Graphics*, 29(3), 2010. 2