POINT CLOUD NON LOCAL DENOISING USING LOCAL SURFACE DESCRIPTOR SIMILARITY

Jean-Emmanuel Deschaud, François Goulette

POINT CLOUD NON LOCAL DENOISING USING LOCAL SURFACE DESCRIPTOR SIMILARITY

Jean-Emmanuel Deschaud and François Goulette

Mines ParisTech, CAOR-Centre de Robotique, Mathématiques et Systèmes
60 Boulevard Saint-Michel 75272 Paris Cedex 06
jean-emmanuel.deschaud@mines-paristech.fr, francois.goulette@mines-paristech.fr

Commission III/3

KEY WORDS: Point Cloud, Denoising, LIDAR, Descriptor, Mesh

ABSTRACT:

This article addresses the problem of denoising 3D data from LIDAR. It is a step often required to allow a good reconstruction of surfaces represented by point clouds. In this paper, we present an original algorithm inspired by a recent method developed by (Buades and Morel, 2005) in the field of image processing, the Non Local Denoising (NLD). With a local geometric descriptor, we look for points that have similarities in order to reduce noise while preserving the surface details. We describe local geometry by MLS surfaces and we use a local reference frame invariant by rotation for denoising points. We present our results on synthetic and real data.

1 INTRODUCTION

LIDAR data (from fixed LIDAR Scanning or Mobile Mapping systems) are widely used for 3D reconstruction of objects or large indoor/outdoor environments. A common step in 3D recontruction is the creation of a triangulated mesh. However many methods are very sensitive to noise, like triangulation methods such as the Ball Pivoting Algorithm (BPA) introduced by (Bernardini et al., 1999). To use some triangulation algorithms, it is necessary to remove or filter the random acquisition noise present in the data, called "denoising". The denoising in point cloud is an essential step to enable a faithful recontruction of a surface scanned by a LIDAR. Many denoising methods of point cloud exist in the literature but only few of them manage to preserve the surface details while smoothing noise. Similarly, the sharp edges and corners are often too smooth in the denoising methods. Or to build a realistic model of an environment from LIDAR data, it is important to keep and recover the corners. In this paper, we present an innovative method for denoising, adapted from a so called "Non Local Denoising" designed for noise filtering in 2D images. Our algorithm is able to smooth the surface and preserve features due to geometric similarities.

2 PREVIOUS WORK

There are mainly two ways to use a denoising algorithm: as preprocessing on a point cloud before a reconstruction method or as a post-treatment directly on meshes. The methods in such a framework is often transferable to another context as bilateral denoising in (Fleishman et al., 2003) which was presented to denoise a mesh but which may equally well denoise a point cloud. However, the denoising operated directly on the point cloud may allow a better surface recontruction, especially for triangulation algorithms which are dependent on the noise as the Ball Pivoting Algorithm (BPA) in (Bernardini et al., 1999).

As described in (Wang et al., 2008) and (Schall et al., 2008), we can classify the denoising methods in several categories. The two main categories are neighborhood filtering and projection-based approaches.

For methods based on neighborhood filtering, many are extensions of 2D filters to 3D as the well-know Bilateral filter in (Tomasi and Manduchi, 1998) adapted to meshes by (Fleishman et al., 2003). (Choudhury and Tumblin, 2003) introduce the trilateral filtering for images and adapt it for meshes. (Yoshizawa et al., 2006) present an extension of the bilateral filter by adding a weight of similarity between points like in the Non Local Denoising. We use the same concept of geometric similarity between points to denoise the surface but we do not need to be near the point to denoise. Methods based on the bilateral filter also introduce a disruptive effect of shrinkage, ie the volume loss from the surface. (Schall et al., 2008) and (Huhle et al., 2008) are also based on non-local denoising but only for depth images. (Wang et al., 2008) use non local denoising by geometry intensity similarities to denoise point sample surfaces.

For methods based on projections, there is the well-known moving-least-squares (MLS) surfaces introduced by (Levin, 1998) for finding a local approximation of the denoised surface by least square. Mederos et. al. uses this approximation to the surface to denoise the point cloud in (Mederos et al., 2003). (Fleishman et al., 2005) and (Oztireli et al., 2009) extend the definition of MLS surfaces in terms of local kernel regression which provides a robust projection method on the MLS surface: this is the Robust Moving Least Square (RMLS) and Robust Implicit Moving Least Square (RIMLS).

3 NON LOCAL DENOISING FOR IMAGES

Non Local Denoising is an algorithm for image denoising introduced by (Buades and Morel, 2005). The objective is to denoise a pixel using other pixels whose neighborhoods are similar. An image is $V = \{v_i \mid i \in I\}$. The new value of the image pixel v_i is $v'_i = \sum_{j \in N(i)} w(i,j) v_j$. The weight $w(i,j)$ is a measure of similarity between the pixel i and the pixel j with $0 \leq w(i,j) \leq 1$ and $\sum_{j \in N(i)} w(i,j) = 1$. The similarity between pixels i and j will depend on the similarity of intensities of gray in the neighborhood of i and j. We have therefore:

$$\delta v(i,j,k,h) = \frac{\mid v(j+k) - v(i+k) \mid^2}{h^2} \qquad (1)$$

$$w(i,j) = \frac{1}{Z_i} e^{-\sum_{k\in N^l(i)} e^{-\frac{\|k\|^2}{a^2}} \delta v(i,j,k,h)} \qquad (2)$$

with normalization constant,

$$Z_i = \sum_{j\in N(i)} e^{-\sum_{k\in N^l(i)} e^{-\frac{\|k\|^2}{a^2}} \delta v(i,j,k,h)} \qquad (3)$$

where $N^l(i)$ is the local neighborhood of a pixel v_j in order to make the comparison of similarity with pixel v_i. This neighborhood should be large enough to be robust to noise and small enough to be representative of the local image details in pixel v_j. a is the parameter for the gaussian kernel around the pixel v_i, ie the local influence of points around v_i. h is a control parameter of the similarity influence. To avoid a similarity search on all image pixels, it is restricted to a large window $N(i)$. $N(i)$ must be large enough to find more similar weights and small enough to limit the time calculation.

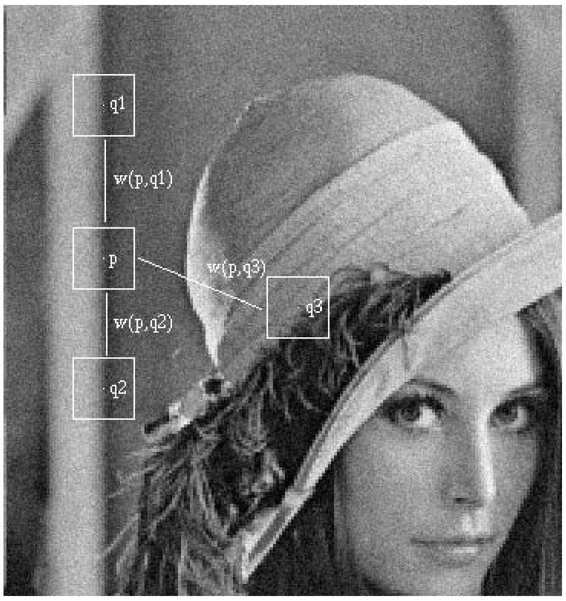

Figure 1: NL-means strategy

We see on Figure 1 (image taken from (Buades and Morel, 2005)) that for denoising the pixel p, we compare the neighborhoods of other image pixels with the neighborhood of p. We see here that the similarity weight $w(p,q1)$ between p and $q1$ and weight $w(p,q2)$ between p and $q2$ will be close to 1 while weight $w(p,q3)$ between p and $q3$ will be close to 0.

4 POINT CLOUD NON LOCAL DENOISING

4.1 From 2D to 3D

Buades et. al. compare pixels of neighborhood for similarity but this notion is not extensible in 3D. For that, we have to define an other neighborhood similarity notion in point clouds. For this, we do not restrict ourselves to points of the neighborhood as in (Yoshizawa et al., 2006) and not just take the local geometry intensity as in (Choudhury and Tumblin, 2003) but we want to consider points which have the same local geometric similarity and we use for that a local geometric description for neighborhood comparison. For this we use the concept of local geometric descriptor. Indeed, points that have the same geometry in their neighborhood can be used to reduce noise while preserving the surface details. We need these descriptors to be robust to noise and to represent the local geometry by keeping the details. An other important feature of the local descriptor is to be invariant by rotation. For example, if we want to denoise points on a sphere, we want to use geometric similarities of other points on the sphere and the local descriptor must be invariant by rotation.

4.2 Algorithm Overview

Consider a point cloud $P = \{p_i \mid i \in I\}$ where p_i is the ith vertex. Our method proceeds in three steps. First, we define at each point p_i a local coordinate system. Then in this coordinate system, we calculate a bivariate polynomial g_i that will be a descriptor of local geometric point p_i. In the last step, we modify a point v_i with points v_j in a large neighborhood $N(i)$ in setting a weight $w(i,j)$ between two points by the difference between the two approximation surfaces g_i and g_j.

4.3 Local Coordinate System

We note b_i the weigthed barycenter of points in the local neighborhood $N^l(i)$ of p_i :

$$b_i = \sum_{j\in N^l(i)} \frac{1}{Z_i} e^{-\frac{\|p_j - p_i\|^2}{a^2}} p_j \qquad (4)$$

with Z_i the normalization constant

We use the local plane $H_i = \{x \mid n_i \cdot x - d_i = 0\}$ which minimizes the weighted sum of the distances bewteen the points p_i and the plane H_i, ie minimizing the sum :

$$\sum_{j\in N^l(i)} (n_i \cdot p_j - d_i)^2 e^{-\frac{\|p_j - p_i\|^2}{a^2}} \qquad (5)$$

We know that n_i is the normal of the plane H_i and is the eigenvector of the smallest eigenvalue of the weighted covariance matrix C_i associated with H_i. For more details, see (Hoppe et al., 1992). We know that the covariance matrix of a local neighborhood can be used to estimate local properties. The eigenvectors (e_i^0, e_i^1, e_i^2) of the covariance matrix C_i form an orthogonal frame associated with eigenvalues $(\lambda_i^0, \lambda_i^1, \lambda_i^2)$ with $\lambda_i^0 \leq \lambda_i^1 \leq \lambda_i^2$. The eigenvector e_i^0 is equal to the normal n_i of the local plane H_i. We know that the barycenter b_i is on the plane H_i. We take the vectors $(e_i^2, -e_i^1, e_i^0)$ and the point b_i as a local frame of p_i because their are robust to noise and invariant by rotation. We call this local frame F_i.

4.4 MLS Surface

As local geometric descriptor, we use a variant of the MLS surfaces, like explained in (Levin, 1998) and in (Alexa et al., 2001). Once we have found a local frame for a point p_i, we can compute a local bivariate polynomial approximation $g_i(u,v)$ which minimizes the following weighted sum :

$$\sum_{j \in N^l(i)} (g_i(u_i^j, v_i^j) - w_i^j)^2 e^{-\frac{\|p_j - p_i\|^2}{a^2}} \tag{6}$$

Here, (u_i^j, v_i^j, w_i^j) are the local coordinates of point p_j in the frame F_i. g_i is thus a smoothed representation of the surface around the point p_i. We see that g_i depends on the local system chosen for H_i. The local coordinates of the point p_i is (u_i^i, v_i^i, w_i^i). w_i^i represents the signed distance from the point p_i to the plane H_i.

g_i represents the smoothed local surface of the point p_i. In order to compare these local descriptors in the second step, it requires that g_i are calculated in a similar local reference, ie that g_i must be invariant under rotation of the neighborhood of the point p_i. Therefore we have choosen as reference the local basis $(e_i^2, -e_i^1, e_i^0)$ with eigenvectors of the covariance matrix of C_i sorted in ascending eigenvalues and for the center the weighted barycenter. As explained in (Pauly et al., 2002), eigenvalues represent the configuration of the point around the plane H_i and if the local neighborhood distribution of two points p_i and p_j is the same with a certain rotation, we know that g_i and g_j will be identical in their local frame F_i and F_j.

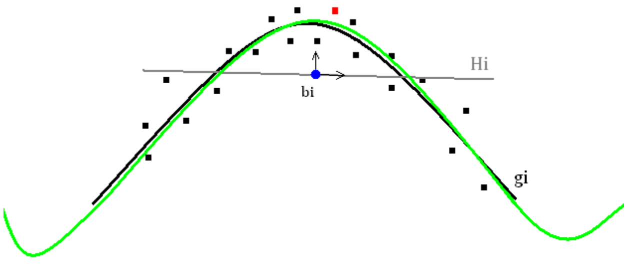

Figure 2: Local Descriptor g_i

The Figure 2 shows for a red point p_i the plane H_i in gray, the barycenter b_i in blue, the bilateral polynomial g_i in black and the denoised surface in green.

4.5 Denoising

For denoising a point p_i, we look for points that have the same local surface descriptor, that is why we use a weight of similarity between points p_i and p_j like :

$$w(i,j) = \frac{1}{Z_i} exp^{-\frac{\|g_i - g_j\|_P^2}{h^2}} \tag{7}$$

with Z_i the normalization constant
and $\| \|_P = \sum_{k=0}^{k=n} | a_k |$
(a_k are the coefficients of the bivariate polynomial g_i)

The coordinates of the new point p_i' is calculated as follows:

$$p_i' = b_i + \sum_{j \in N(i)} w(i,j)(u_j^j e_i^2 - v_j^j e_i^1 + w_j^j e_i^0) \tag{8}$$

where (u_j^j, v_j^j, w_j^j) are the local coordinates of points p_j in their frame F_j. If the weight $w(i,j)$ is close to one (ie p_i and p_j have the same local geometry) then we use the local coordinates of p_j in F_j to smooth the point p_i.

4.6 Other Local Descriptors

We have used MLS surfaces as local descriptor but we could imagine other local descriptors. For example, the covariance matrix represents the variation of distribution in local neighborhood, robust to the noise but not invariant by rotation. Eigenvalues of the covariance matrix describe also the variation of the sampling distribution in the tangent plane, are robust to noise and invariant by rotation. So it could be used as local descriptor. But only 3 values is poor to describe a local distribution of points. We have found best results with MLS surface descriptor but we can imagine other local descriptors like Radial basis functions (RBF) as in (Yoshizawa et al., 2006).

5 RESULTS AND DISCUSSION

5.1 Parameters

We give here the parameters for our non local denoising (NLD) algorithm. We use for the large neighborhood $N(i)$ the 200 nearest neighbors. The larger the neighborhood $N(i)$ is, the more points will be properly denoised because they will find more points with the same local geometry. However, the computing time will be larger. For the local neighborhood $N^l(i)$, we keep the 20 closest neighbors. The parameter a in the weight gaussian kernel represents the locality of the surface descriptor g_i. We have taken a equal to the distance from p_i to the farthest point in the local neighbor in $N^l(i)$. For the degree of the bivariate polynomial g_i, we have chosen polynomials of degree 3. We use these values for all tested point clouds.

The parameter h used in the weight similarity bewteen two local surface g_i and g_j is the most difficult to choose because it depends on the sampling distribution of the point cloud and the noise. It varies from 0.1 to 10.

5.2 Results on synthetic data

First, we have tested our algorithm on synthetic data : a corner point cloud and a ring point cloud. With synthetic data, we can control the noise and evaluate the denoising method. The corner point cloud is made of 2 planes with a 90 degree angle with 1000 points on each plane (2000 points in all) and with a distance bewteen points of 1 meter. We have add a gaussian noise on points with a standard deviation of 0.5 meter. For comparison, we have implemented the Bilateral algorithm from (Fleishman et al., 2003) and the Robust Implicit Moving Least Square (RIMLS) algorithm from (Oztireli et al., 2009). For Bilateral and RIMLS, we tried to choose the parameter settings that produce the best results. All methods presented here work on point cloud data but for better visualization purpose, we have constructed the surface mesh with a triangulation algorithm which is a variant of the Ball Pivoting Algorithm (Bernardini et al., 1999).

We can see the results of denoising algorithms on Figure 3. Our NLD algorithm is able to keep the corner due to geometric similarities, ie other points of the corner that has been used to remove the noise. To have a quantitative comparison, we use the Hausdorff distance bewteen meshes.

The Hausdorff distance bewteen two surfaces is :

$$d(X,Y) = max\{\underset{y\in Y}{sup}\ \underset{x\in X}{inf}\ \delta(x,y), \underset{x\in X}{sup}\ \underset{y\in Y}{inf}\ \delta(x,y)\} \quad (9)$$

We compute the distance between each denoised mesh and the original mesh. The results are in the Table 1. We see that our denoised mesh is more closer from the original mesh than other denoised meshes. We get a surface closer to the original surface than the bilateral filter or RIMLS filter. We can also see a difference in computation time. It is because other algorithms have to do multiple iterations (around 4-5 for bilateral and 2-3 for RIMLS) but our algorithm works in one pass.

	CORNER	CORNER	RING	RING
Mesh	H-distance	Time	H-distance	Time
Noise	0.159103 m	-	0.113102 m	-
Bilateral	0.126050 m	1.5 s	0.081701 m	1.4 s
RIMLS	0.095922 m	1.7 s	0.069357 m	1.8 s
NLD	0.073229 m	1.2 s	0.06316 m	1.1 s

Table 1: Comparison of denoising methods on corner point cloud and ring point cloud with Hausdorff distance (H-distance) from the original mesh.

We have also compared denoising algorithms on an other synthetic point cloud : a ring. This is a half sphere with 2000 points. The mean distance between points is 0.3 meter and we have add a gaussian noise with a standard deviation of 0.1 meter. We can see the results of denoising algorithms on Figure 4. We have also the Haussdorff distance comparison bewteen meshes on the table 1. We see that our algorithm has a good behavior on the border of the ring because we almost recover the border of the original ring but we have worst result on the little variation of the border. It is because we have no other point with the same local geometry. It is one limitation of our algorithm. If a point has a singular geometry in his large neighborhood $N(i)$, then we cannot denoise it. It is why we think we can have better results by adding just one pass of a projection denoising algorithm after the NLD algorithm like the RIMLS filter. This idea is on work.

5.3 Results on real data

We have tested our algorithm on real data from fixed scanner and data from mobile scanners as in Figures 5, 6 and 7. In these three cases, we see that the noise has greatly decreased while keeping details like edges. We can see that points in windows with singular geometry have not been denoised.

6 CONCLUSION

We have proposed an innovative method for 3D point cloud denoising that is an extension of the 2D "Non Local Denoising" by using MLS surfaces as local descriptors. We have tested on synthetic and real data sets and compared with state of the art in denoising algorithm. We have demonstrated its ability to smooth the surface in noisy areas and to keep edges and corners by local geometric similarities on the point cloud. We have seen one limitation of this algorithm in points with singular geometry, i.e. points with no neighborhood similarities in their neighborhood.

REFERENCES

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D. and Silva, C. T., 2001. Point set surfaces. Proceedings of the conference on Visualization.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. and Taubin, G., 1999. The ball-pivoting algorithm for surface reconstruction. IEEE Transactions on Visualization and Computer Graphics 5(4), pp. 349–359.

Buades, A. and Morel, J.-M., 2005. A non-local algorithm for image denoising. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Choudhury, P. and Tumblin, J., 2003. The trilateral filter for high contrast images and meshes. Eurographics Symposium on Rendering.

Fleishman, S., Cohen-Or, D. and Silva, C. T., 2005. Robust moving least-squares fitting with sharp features. SIGGRAPH.

Fleishman, S., Drori, I. and Cohen-Or, D., 2003. Bilateral mesh denoising. ACM Transactions on Graphics (TOG) 22(3), pp. 950–953.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., 1992. Surface reconstruction from unorganized points. SIGGRAPH.

Huhle, B., Schairer, T., Jenke, P. and Strasser, W., 2008. Robust non-local denoising of colored depth data. Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops.

Levin, D., 1998. The approximation power of moving least-squares. Mathematics of Computation 67, pp. 807–815.

Mederos, B., Velho, L. and de Figueiredo, L. H., 2003. Robust smoothing of noisy point clouds. Proceedings of the SIAM Conference on Geo-metric Design and Comuting.

Oztireli, A. C., Guennebaud, G. and Gross, M., 2009. Feature preserving point set surfaces based on non-linear kernel regression. Computer Graphics Forum.

Pauly, M., Gross, M. and Kobbelt, L. P., 2002. Efficient simplification of point-sampled surfaces. Proceedings of the conference on Visualization.

Schall, O., Belyaev, A. and H.-P.Seidel, 2008. Adaptive feature-preserving non-local denoising of static and time-varying range data. Computer-Aided Design 40(6), pp. 701–707.

Tomasi, C. and Manduchi, R., 1998. Bilateral filtering for gray and color images. Proceedings of the Sixth International Conference on Computer Vision.

Wang, R.-F., Chen, W.-Z., Zhang, S.-Y., Zhang, Y. and Ye, X.-Z., 2008. Similarity-based denoising of point-sampled surfaces. Journal of Zhejiang University - Science A 9(6), pp. 807–815.

Yoshizawa, S., Belyaev, A. and Seidel, H.-P., 2006. Smoothing by example: Mesh denoising by averaging with similarity-based weights. Proceedings of the IEEE International Conference on Shape Modeling and Applications.

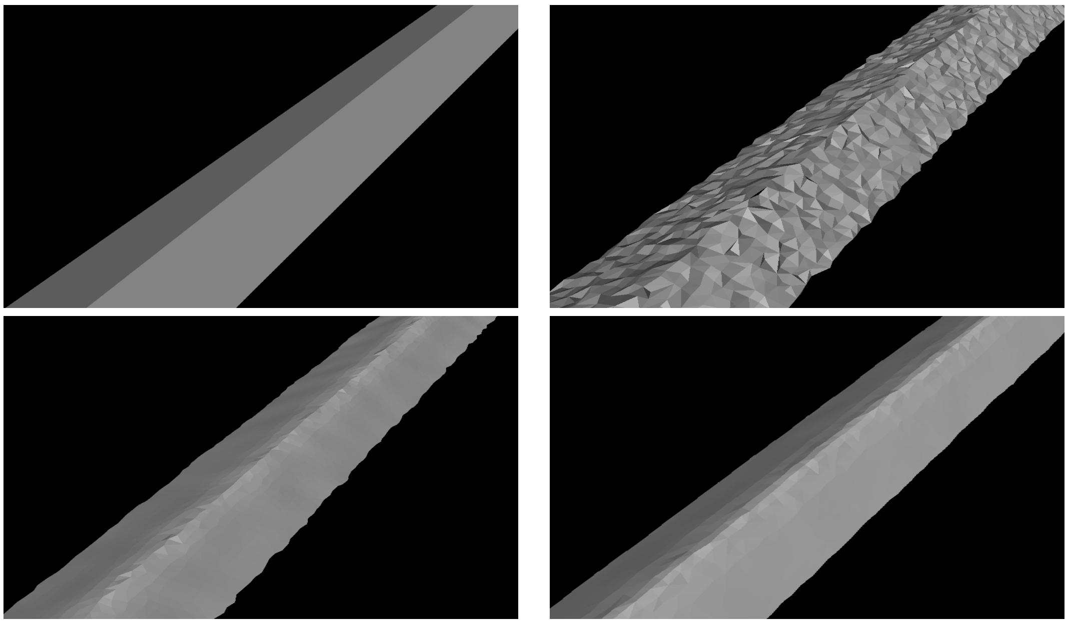

Figure 3: Results of denoising a corner point cloud (top left : original, top right : noise, bottom left : RIMLS, bottom right : NLD)

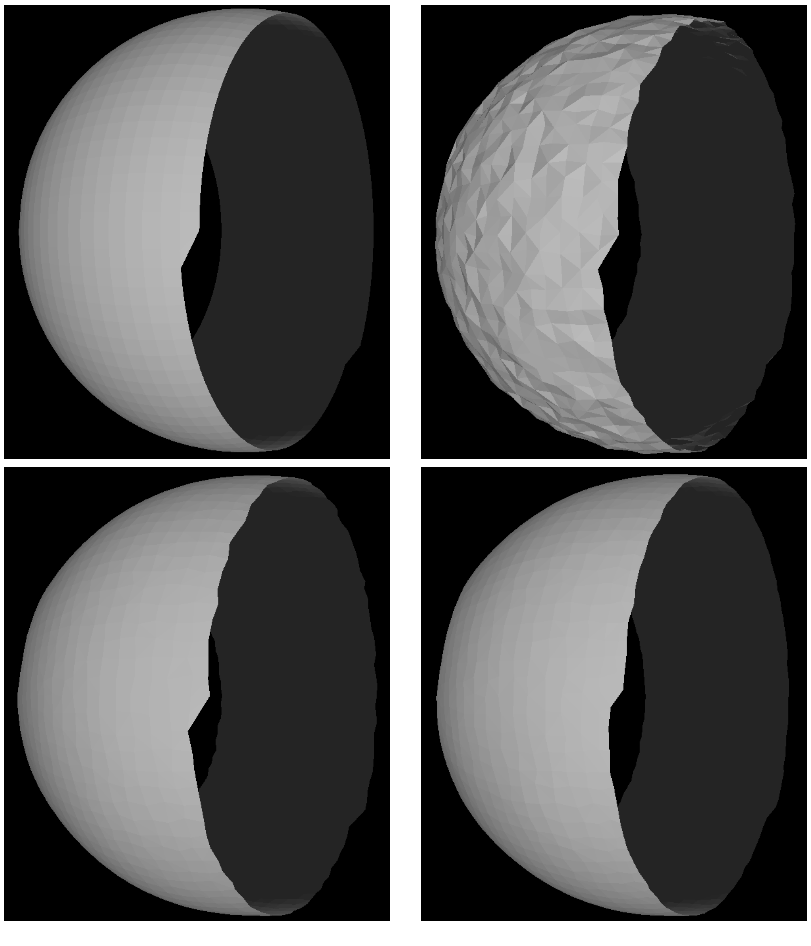

Figure 4: Results of denoising a ring point cloud (top left : original, top right : noise, bottom left : Bilateral, bottom right : NLD)

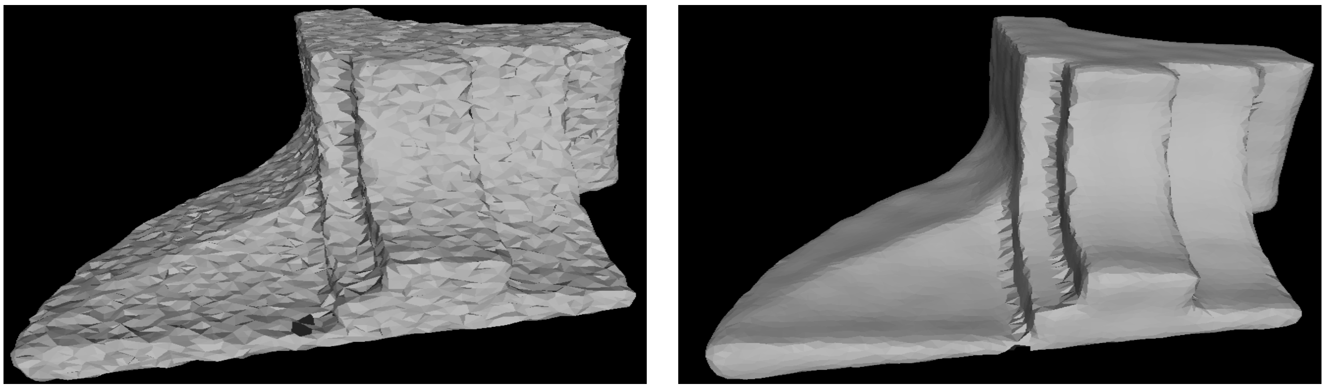

Figure 5: Results of denoising a fandisk point cloud (left : original, right : denoised by NLD)

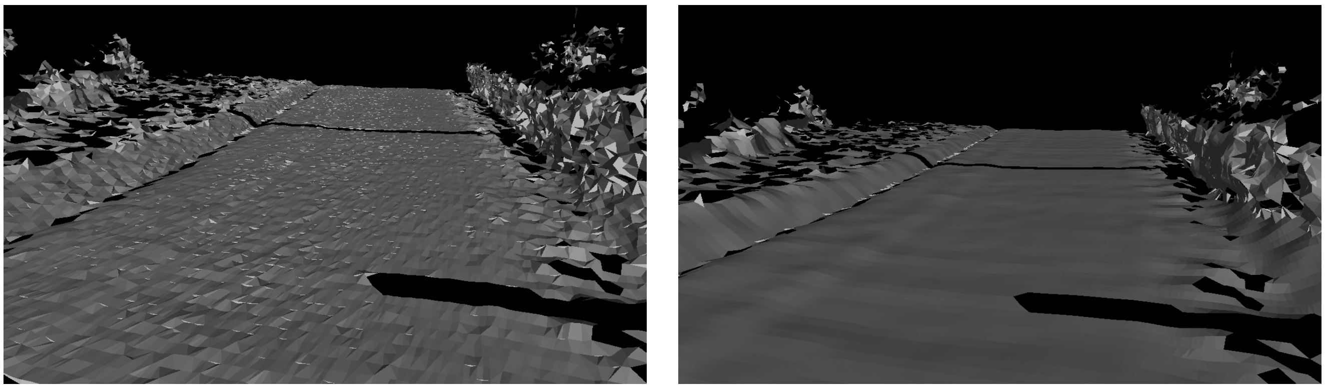

Figure 6: Results of denoising a point cloud from a mobile mapping system (left : original, right : denoised by NLD)

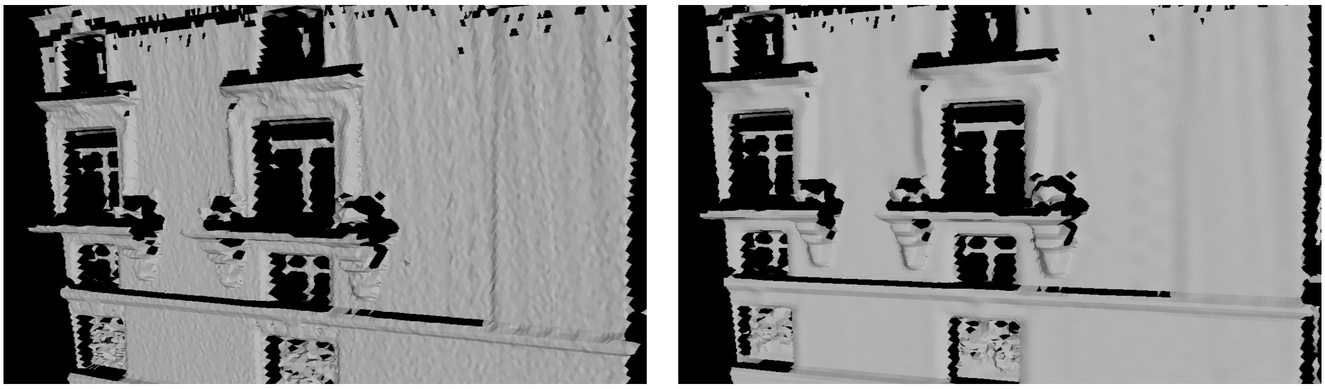

Figure 7: Results of denoising a point cloud from a mobile mapping system (left : original, right : denoised by NLD)