Robust Smoothing of Noisy Point Clouds

Boris Mederos, Luiz Velho and Luiz Henrique de Figueiredo

Abstract. This paper addresses the problem of removing the noise from noisy points clouds in the context of surface reconstruction. We introduce a new smoothing operator Q inspired by the moving least-squares method and robust statistics theory. Our method can be seen as an improvement of the moving least-squares method to preserve sharp features. We also present effective numerical optimization algorithms to compute Q and some theoretical results on their convergence.

§1. Introduction

Data acquired with 3D scanners is invariably noisy. Moreover, the increasing use of 3D scanners has implied a growth in the complexity of the scanned models. Therefore, it is crucial that we develop efficient and robust algorithms for surface reconstruction and denoising point clouds that preserve the fine features of the models for further processing.

A significant effort has been done in mesh smoothing in the last years, resulting in a variety of algorithms such as the Laplacian operator [20], anisotropic diffusion [3, 6, 8, 9], diffusion of the normal field [4, 17, 21], and locally adaptive Wiener filtering [1, 19].

Recent methods [11, 14] introduced feature-preserving mesh smoothing based on robust statistics. Those methods implement non-interactive w-estimators [13], where the new position p' of a vertex p is computed as a weighted sum of the predictors $\Pi_q(p)$ from its spatial neighborhood:

$$\theta = k^{-1} \sum_{q \in N(p)} a_q w_g(\Pi_q(p) - p) w_f(\|q - p\|) \Pi_q(p) \tag{1}$$

In [14], the predictor $\Pi_q(p)$ is the projection of p onto the tangent plane of the triangle q and the new position of the vertex p is $p' = \theta$. In [11], a local parameter space is determined using a tangent plane at the vertex p,

xxx and xxx (eds.), pp. 1–4.
Copyright © 200x by Nashboro Press, Brentwood, TN.
ISBN 0-9728482-x-x
All rights of reproduction in any form reserved.

where the neighborhood of p can be seen as a height field over the tangent plane. The predictor $\Pi_q(p) = I(q)$ is the height of the point q and the new position is computed as $p' = p + \theta n_p$, where n_p is the normal of the tangent plane.

Levin [15] developed a mesh-independent method for smooth surface approximation, the moving least-squares method (MLS), introducing a different paradigm based on a projection procedure. Given a manifold S and a set of points $\{r_i\}_{i\in I}$ on or near S, an approximating manifold $\widetilde{S}$ is defined as the set of fixed points of a projection operator Ψ, that is, $\widetilde{S} = \{\, x \in \mathbb{R}^3 : \Psi(x) = x \,\}$. The operator Ψ is defined by a two-step procedure. First, given a sample point r, a local reference frame around r is computed by fitting a hyperplane $H = \{\, x \in \mathbb{R}^3 : n \cdot x - D = 0 \,\}$ minimizing the following weighted least-squares error in the neighborhood of the sample

$$\sum_{i\in I} (n \cdot r_i - D)^2 \theta(\|r_i - q\|), \tag{2}$$

where q is the projection of r onto H. Next, a local polynomial approximation p of S is determined by taking the hyperplane H as reference domain and minimizing the following least-squares error:

$$\sum_{i\in I} (p(x_i) - f_i)^2 \theta(\|r_i - r - tn\|), \tag{3}$$

where x_i is the projection of r_i onto H and f_i is the height of r_i over H. The projection of r onto $\widetilde{S}$ is then defined as $\Psi(r) = q + p(0) \cdot n$. For a detailed implementation of this method for high-quality rendering of point set surfaces, see [2].

Based on the approaches above [11, 14, 15], our main goal here is to propose an algorithm for surface approximation and point cloud smoothing that detects outliers in the data and preserves the significant edges of the surface. We introduce an *M-estimator procedure* as an improvement of [15] to smooth point clouds while preserving salient features. Briefly, at each sample point r, our algorithm determines a normal vector n_r and shifts the sample along that direction obtaining a new position $Q(r) = r + t_r n_r$. The normal n_r and the displacement t_r are computed by the robust fitting of a hyperplane H in the neighborhood of the sample point r.

The organization of the paper is as follows: in Section 2 we explain the proposed method in detail; in Section 3 we discuss the numerical methods used and their convergence; in Section 4 we show the proposed method in action, with several examples illustrating how the algorithm preserve data features; finally, our conclusions are given in Section 5.

§2. Proposed Method

Starting from a noisy point cloud P near a surface S, our goal is determine a noise-free data set $P' = \{ r' = Q(r) : r \in P \}$ that retains the features of the model, using an operator $Q(r) = r + t_r n_r$. At each sample point $r \in P$, we estimate a new position $r' = r + t_r n_r$, where n_r is a projection direction and t_r is a displacement along n_r. As mentioned in the introduction, the optimal values for t_r and n_r are computed by the robust fitting of an hyperplane in a neighborhood $N(r)$ of the sample point r. The hyperplane H_r is obtained minimizing the following cost functional with respect to t and n, subject to the restriction $\|n\| = 1$:

$$\sum_{q \in N(r)} \rho(\widetilde{h}_q) w(\|q - r\|), \tag{4}$$

where $h_q = n^t(q - r - tn)$ is the height of the point q with respect to the hyperplane H passing through the point $r + tn$ and orthogonal to n.

The new position r' can be seen as the projection of r onto a local linear approximation H_r of the surface S. The estimation of this approximating hyperplane H_r is robust to gross deviation of the points $q \in N(r)$. The identification of outliers is controlled by a robust error norm ρ in (4), which penalizes points $q \in N(r)$ with large values of the height h_q. According to this, the points $q \in N(r)$ on different sides of a sharp feature are considered possible outliers. See Figure 1.

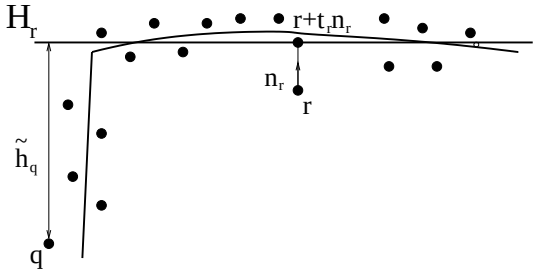


Fig. 1. The smoothing operator $Q(r) = r + t_r n_r$. The point r is projected onto the fitting hyperplane H_r, a linear approximation of the surface S, computed giving less influence to outliers.

Several examples of robust error norms can be found in the literature [5]. In general all these robust potential grow slower than the quadratic error norm ($\lim_{t \to \infty} \frac{\rho'(t)}{2t} = 0$). We use a Gaussian error norm $\rho(x) = 1 - e^{-\frac{x^2}{2\sigma_\rho}}$, where the parameter σ_ρ controls the sensitivity of (4) to outliers. We also use a Gaussian weight $w(x) = e^{-\frac{x^2}{2\sigma_w}}$, where the parameter σ_w controls the influence of points far away from r.

The first step to compute the smoothing operator Q at a sample point r is to determine a neighborhood $N(r)$. We do this through a growing process. We initially compute a subset $N = \{s : \|s - r\| < \sigma_w\}$ of the k-nearest neighbors of r with distance to r smaller than σ_w. We typically use $k = 8$. The set N is augmented $N \leftarrow N \cup N_s$ for each element $s \in N$ not processed, where N_s is a subset of the k-nearest neighbors of s that are not in N with distance to r smaller than the parameter σ_w. This process is repeated for each new point in N not processed until there are not more points to be added, then the final neighborhood is $N(r) = N$. Observe that determining the neighborhood in this way tends to eliminate points that are in different connected components or on different sides of a thin region. See Figure 2.

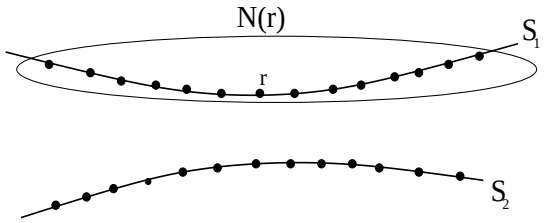


Fig. 2. Two close sheets S_1 and S_2 of a surface S corresponding to a thin region of S or two different connected components. The neighborhood $N(r)$ is computed with respect the sheet S_1.

Having found the neighborhood $N(r)$, we apply the smoothing operator $Q(r) = r + t_r n_r$, where t_r and n_r minimize (4) subject to $\|n\| = 1$:

$$\{n_r, t_r\} = \arg\min_{\{n,t\}} \sum_{q \in N(r)} \rho(n^t(q - r) - t) w(\|q - r\|) \tag{5}$$

The optimal values of n_r and t_r are computed by alternately minimizing with respect to n and t until we are close to the minimum value. This procedure is summarized in Algorithm 1.

Algorithm 1

$t_0 = 0$
repeat {for $k = 1, \ldots$}
(i) Compute n_k as the minimum of (5) with $t = t_{k-1}$ fixed
(ii) Compute t_k as the minimum of (5) with $n = n_k$ fixed
until we are close to the minimum.

Step (i) of Algorithm 1 is devoted to solving a constrained optimization problem. We solve problem (5) with respect to n having fixed t, subject to

the restriction $\|n\| = 1$. To solve this problem, we use Newton's method over Riemannian manifolds [22, 10]. Given a C^∞ manifold (in our case, the sphere S^2) with Riemannian structure g, Levi-Civita connection ∇, and a C^3 function f over S^2, we have the following algorithm:

Algorithm 2 Newton Method

Select a point $p_1 \in P$ such as $(\nabla^2 f)_{p_1}$ is nondegenerate
repeat {for $i = 1, \ldots$}
(1) Solve $H_i = -(\nabla^2 f)_{p_i}^{-1}(\nabla f)_{p_i}$
(2) Compute $p_{i+1} = \exp_{p_i} H_i$
until we are close to the minimum.

In (1) we solve the linear system $(\nabla^2 f)_{p_i} H_i = (\nabla f)_{p_i}$ where $\nabla^2 f$ is the Hessian of (5) with respect to n, $(\nabla f)_{p_i}$ is its differential, and H_i is the solution on the tangent plane $T_{p_i}S^2$ of the above linear system. In step (2), we make a displacement (with $t = 1$) along the unique geodesic $\gamma(t) = \exp_{p_i} H_i\, t$, with $\gamma(0) = p_i$ and direction $\gamma'(0) = H_i$.

As is widely known, Newton's method has quadratic convergence when started near the solution. For that reason, in the first iteration ($k = 1$) of the algorithm 1, we use as initial input vector an estimative of the solution of the equation (5) with $t = 0$ and in the next iterations ($k > 1$) we use as input the normal n_{k-1} computed at the previous step. To determine the initial normal for the first iteration ($k = 1$), we compose the Lagrange equation of (4) for $t = 0$ and its derivate with respect to n, obtaining:

$$L(n, \lambda) = \sum_{q \in N(r)} \rho(h_q) w(\|q - r\|) + \lambda(\|n\|^2 - 1) \tag{6}$$

$$L_n(n, \lambda) = \sum_{q \in N(r)} \theta_q (q - r)(q - r)^t n - \lambda n = 0, \tag{7}$$

with weight $\theta_q = \Psi_\rho(h_q)\Psi_w(\|q - r\|)$ and weight functions $\Psi_\rho(x) = e^{-\frac{x^2}{2\sigma_\rho}}$ and $\Psi_w(x) = e^{-\frac{x^2}{2\sigma_w}}$. As before, $h_q = n^t(q - r)$.

Equation (7) can be represented in matrix form with a symmetric and definite positive matrix $M(n)$ depending of n:

$$M(n)n = \lambda n \quad \text{where} \quad M(n) = \sum_{q \in N(r)} \theta_q (q - r)(q - r)^t \tag{8}$$

The weight θ_q determines the influence of the term $(q - r)(q - r)^t$. Large values of h_q imply small values of the weight θ_q and consequently less influence of $(q - r)(q - r)^t$ in the matrix $M(n)$.

To solve equation (8), we propose the following iterative scheme:

$$M(n_k)n_{k+1} = \lambda_{k+1} n_{k+1}, \tag{9}$$

where λ_{k+1} is the smallest eigenvalue of $M(n_k)$ and n_{k+1} is its associate orthonormal eigenvector. The traditional covariance analysis is obtained for the starting normal $n_0 = 0$ ($M(0)n_1 = \lambda_1 n_1$). A few iterations of this method produce good results and always decrease the objective function, as Lemma 2 in the next section shows.

In step (ii) of Algorithm 1, we minimize (5) with respect to t, keeping the normal n fixed; This is an unconstrained optimization problem. The necessary first-order condition for a minimum is then:

$$\sum_{q \in N(r)} \Psi_\rho(h_q - t)\Psi_w(\|q - r\|)(h_q - t) = 0 \tag{10}$$

Solving this equation for t yields the following recurrence equation

$$t_{i+1} = k_{t_i}^{-1} \sum_{q \in N(r)} \Psi_\rho(h_q - t_i)\Psi_w(\|q - r\|)h_q \tag{11}$$

where we have introduced a normalization factor

$$k_{t_i} = \sum_{q \in N(r)} \Psi_\rho(h_q - t_i)\Psi_w(\|q - r\|) \tag{12}$$

Lemma 1 in the next section shows that the sequence t_i always converges. Note that one iteration of equation (11) starting at $t_0 = 0$ is exactly the proposed method in [11].

§3. Numerical Methods

Convergence. We now analyze some theoretical aspects about the convergence of the iterative method proposed in the previous section. We shall prove that the sequence (11) $t_{i+1} = f(t_i)$ converges to a fixed point of the function $f(t) = k_t^{-1} \sum_{q \in N(r)} \Psi_\rho(h_q - t)\Psi_w(\|q - r\|)h_q$.

We need some results of Comaniciu and Meer [7] to state the convergence of (11). They introduce the Mean Shift (MS) to estimate the mode of the data kernel density estimator of a finite set of samples $\{x_i\}_{i=1}^n$ in $\mathbb{R}^d$:

$$d(x) = \frac{1}{NH^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

The Mean Shift is defined as the difference of two consecutive steps $MS(y_m) = y_{m+1} - y_m$ of the sequence (y_m)

$$y_{m+1} = \frac{\sum_{i=1}^n x_i l(\|\frac{y_m - x_i}{h}\|^2)}{\sum_{i=1}^n l(\|\frac{y_m - x_i}{h}\|^2)}, \tag{13}$$

where $l(t) = -k'(t)$ and $k(t)$ is the profile of the Kernel $K(x)$; in other words, $k(\|x\|^2) = K(x)$. The following two theorems from [7] discuss the convergence of the sequence (y_m):

Theorem 1. *Given three consecutive steps y_{m+1}, y_m, y_{m-1} of the sequence $\{y_m\}$ with Gaussian weight $l(x) = \exp(-x)$, we have $(y_{m+1} - y_m)(y_m - y_{m-1}) > 0$.* □

Theorem 2. *If the kernel $K(x)$ has a convex and monotonically decreasing profile, then the sequence $(d(y_m))$ is monotonically increasing.* □

We now applying Theorems 1 and 2 to our particular case:

Lemma 1. *The sequence $t_{i+1} = f(t_i)$ is strictly monotone and converges to a stationary point that is not a maximum.*

Proof: From Theorem 1, we get that $(t_i - t_{i-1})(t_{i+1} - t_i) > 0$. Thus, $t_i - t_{i-1}$ and $t_{i+1} - t_i$ have the same sign. Therefore, the sequence t_i is strictly monotone (increasing or decreasing).

Because the t_i are convex combinations of h_q and $|h_q| = |n^t(q-r)| < \|q-r\| < \sigma_w$, we have that t_i is inside the interval $[-\sigma_w, \sigma_w]$. Thus, the sequence (t_i) is monotone and limited, and so it converges.

Problem (5) can be transformed into a maximization problem

$$\{t\} = \arg\max_t \sum_{q \in N(r)} K(h_q - t) w(\|q - r\|), \tag{14}$$

where the function $K(t) = 1 - \rho(t)$ has a convex and monotone profile $\exp(-t)$. Applying Theorem 2 to equation (14), we obtain that the sequence $d(t_i) = \sum_{q \in N(r)} \rho(h_q - t_i) w(\|q - r\|)$ is decreasing and therefore $t_r = \lim t_i$ is not a maximum. □

Now we analyze the convergence of the iterative scheme (9) proposed in the previous section to minimize problem (5) with $t = 0$:

$$n_r = \arg\min\ F(n) \quad \text{with} \quad F(n) = \sum_{q \in N(r)} \rho(n^t(q-r)) w(\|q-r\|) \tag{15}$$

The next lemma shows that the sequence (n_k) generated by (9) always decreases the objective function $F(n)$.

Lemma 2. *With (n_k) defined by $M(n_k)n_{k+1} = \lambda_{k+1} n_{k+1}$, where λ_{k+1} is the minimum eigenvalue of $M(n_k)$ with associated orthonormal eigenvector n_{k+1}, the sequence $(F(n_k))$ is strictly decreasing.*

Proof: Denoting the height of the point q with respect to the hyperplane passing at r and orthogonal to n_k by $h_k = n_k^t(q-r)$, we have:

$$\begin{aligned} F(n_{k+1}) - F(n_k) &= \sum_{q\in N(r)} (\rho(h_{k+1}) - \rho(h_k)) w(\|q-r\|) \\ &= \sum_{q\in N(r)} (e^{-\frac{(h_k)^2}{2\sigma_\rho^2}} - e^{-\frac{(h_{k+1})^2}{2\sigma_\rho^2}}) w(\|q-r\|) \end{aligned}$$

Since $L(x) = e^{-x}$ is a convex function, the following inequality $L(x_1) - L(x_2) < L'(x_1)(x_1 - x_2)$ hold. Combining this inequality with equation (8), we obtain:

$$\begin{aligned} F(n_{k+1}) - F(n_k) &< \sum_{q\in N(r)} -e^{-\frac{h_k^2}{2\sigma_\rho^2}} w(\|q-r\|)(\frac{h_k^2}{2\sigma_\rho^2} - \frac{h_{k+1}^2}{2\sigma_\rho^2}) \\ &= \frac{1}{2\sigma_\rho^2} \sum_{q\in N(r)} \theta_q (h_{k+1}^2 - h_k^2) \\ &= \frac{1}{2\sigma_\rho^2} \sum_{q\in N(r)} \theta_q (h_{k+1} - h_k)^t (h_{k+1} + h_k) \\ &= \frac{1}{2\sigma_\rho^2} \sum_{q\in N(r)} (n_{k+1} - n_k)^t (\theta_q (q-r)(q-r)^t)(n_{k+1} + n_k) \\ &= \frac{1}{2\sigma_\rho^2} (n_{k+1} - n_k)^t M(n_k)(n_{k+1} + n_k) \\ &= \frac{1}{2\sigma_\rho^2} (\lambda_{k+1} - n_k^t M(n_k) n_k) \end{aligned}$$

Since $\lambda_{k+1} = \min_{\|x\|=1} x^t M(n_k) x^t$, we obtain $\lambda_{k+1} - n_k M(n_k) n_k < 0$, and the lemma follows. □

Newton's Method over S^2. We now present Newton's method on the sphere in more detail. As known, the geodesics in S^2 satisfy the following second order differential equation $\ddot{x}^k + x^k = 0$, for $k = 1, 2, 3$. Therefore, the Cristoffel symbols are given by $\Gamma_{i,j}^k = \delta_{i,j} x^k$, where $\delta_{i,j}$ is the Kronecker symbol. Hence, the ij-th component of the bilinear form

$$(\nabla^2 f)_n = \sum_{i,j} ((\frac{\partial f}{\partial x^i \partial x^j})_n - \sum_k \Gamma_{i,j}^k (\frac{\partial f}{\partial x})_n)) dx^i \otimes dx^j$$

is given by

$$((\nabla^2 f)_n)_{i,j} = (\frac{\partial f}{\partial x^i \partial x^j})_n - \delta_{i,j} (n^t \cdot (\frac{\partial f}{\partial x})_n)$$

Writing the equation above in matrix form we obtain:

$$(\nabla^2 f)_n = H_f(n) - \lambda H_h(n) = H_f(n) - \lambda I,$$

where $H_f(n)$ and $H_h(n)$ are the Hessian of $f(n)$ and of the restriction $h(n) = \|n\|^2 - 1$ respectively and the dot product $\lambda = n^t f_n$ is the Lagrangian multiplier.

In Algorithm 2 we need to solve $H_i = -(\nabla^2 f)^{-1}_{n_i}(\nabla f)_{n_i}$ with H_i on the tangent plane, where $(\nabla f)_{n_i} = \nabla f - \lambda n_i$ is the gradient of $f(n_i)$. In general, a linear operator $A : R^3 \to R^3$ defines a linear operator on the tangent plane $T_n S^2$ for each n in S^2 such that $A \cdot u = (I - nn^t)Au$. Therefore the solution of the linear system $A \cdot u = v$ with u and v in $T_n S^2$ has the form

$$u = A^{-1}v - \alpha^{-1}(A^{-1}n)(nA^{-1}v), \;\; \alpha = n^t A^{-1} n \tag{16}$$

Rewriting Algorithm 2 in this context with $L_{nn} = \nabla^2 f$ and $L_n = \nabla f$ to simplify the notation we obtain:

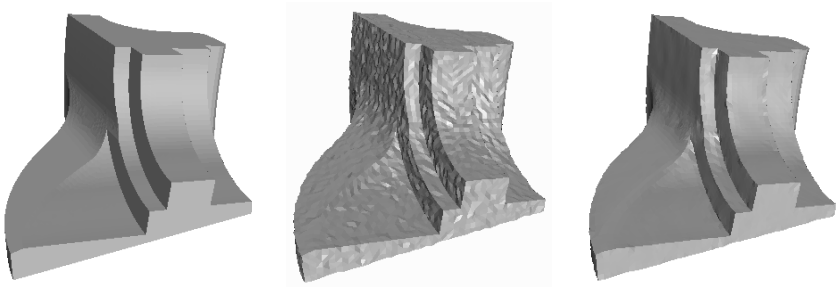
Algorithm 3 Newton's Method in S^2

Find an initial point $n_1 \in S^2$
repeat {for $i = 1, \ldots$}
(i) $v_i = L^{-1}_{n_i n_i} L_{n_i} - \alpha^{-1}(L^{-1}_{n_i n_i} n_i)(n_i L^{-1}_{n_i n_i} L_{n_i})$, with $\alpha = n_i^t L^{-1}_{n_i n_i} n_i$
(ii) $n_{i+1} = \cos(\|v_i\|)n_i + \sin(\|v_i\|)\frac{v_i}{\|v_i\|}$
until we are close to the minimum.

In Step (i) we solve a linear system $L_{n_i n_i} v_i = L_{n_i}$ with right and left side on the tangent space $T_{n_i} S^2$. Applying equation (16) we obtain the solution $v_i = L^{-1}_{n_i n_i} L_{n_i} - \alpha^{-1}(L^{-1}_{n_i n_i} n_i)(n_i L^{-1}_{n_i n_i} L_{n_i})$, where $\alpha = n_i^t L^{-1}_{n_i n_i} n_i$. The linear systems $x_0 = L^{-1}_{n_i n_i} L_{n_i}$ and $x_1 = L^{-1}_{n_i n_i} n_i$ are in general ill conditioned, so we used the DGSVD methods proposed in [12] to solve ill conditioned linear problems. Step (ii) corresponds to taking $t = 1$ in the geodesic $\gamma(t) = \exp_{n_i}(tv_i)$, taking the form $n_{i+1} = \cos(v_i)n_i + \sin(v_i)\frac{v_i}{\|v_i\|}$ on the sphere S^2.

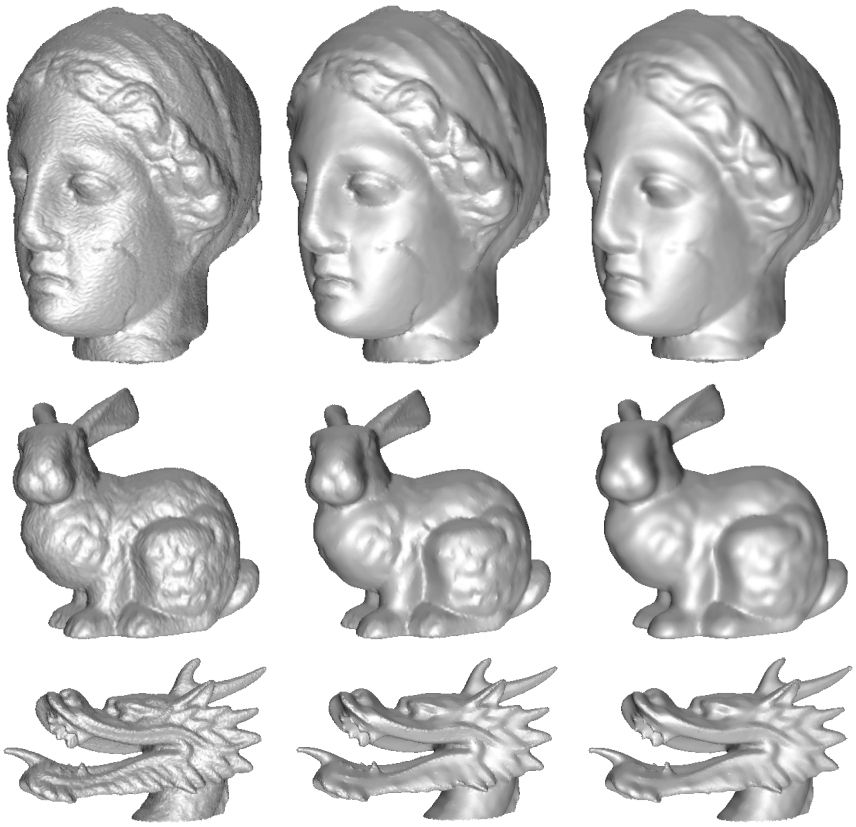
§4. Experiments

Our method has been tested on several data sets in combination with a surface reconstruction method that we are developing [16]. Figure 3 shows the ability of the algorithm to preserve fine features: note how the edges and corners are preserved. Figure 4 presents a comparison of our method with moving least-squares. Note how details are preserved: the eye, hair and scar of the igea; the eye, nose, feet and creases of the bunny; and the eye, nose and teeth of the dragon.



(a) original (b) noisy model (c) our method

Fig. 3. Preserving features in the fandisk model.



(a) noisy models (b) our method (c) MLS method

Fig. 4. Comparing our method with MLS.

All the models were contaminated with Gaussian noise along of the normal direction: the igea, bunny and dragon models with zero mean and variance of 15% of $\frac{1}{10}$ bounding box diagonal; the fandisk model with variance of 8%.

The parameters σ_ρ and σ_w were: $\sigma_\rho = 2.5h$ and $\sigma_w = 0.2h$ for the igea model; $\sigma_\rho = 4.5h$ and $\sigma_w = 1.0h$ for the dragon; $\sigma_\rho = 2.5h$ and $\sigma_w = 0.26h$ for the bunny; and $\sigma_\rho = 0.12h$ and $\sigma_w = 2.0h$ for the fandisk, where h is the mean spacing between the points. In all the models the diameter of the neighborhood was set to σ_w. As in methods [14, 11], a small value in the parameter σ_w leads to faster computation because the neighborhood $N(r)$ is small and large value may cross sharp features and over-smooth the results. The parameter σ_ρ controls the sensitivity to outliers; for small values of σ_ρ the small features of the models are preserved and for larges values only salient features are preserved.

§5. Conclusion

We have presented a new method for point cloud denoising which is inspired by robust statistics and can be seen as an extension of the moving least-squares method to be preserve features. The method makes a robust estimation of the surface normal at the point cloud and was implemented with an effective numerical optimization procedure. It has two empirical parameters σ_ρ and σ_w that can be set by the user. How to determine these parameters automatically is an open issue.

Acknowledgments. The authors are partially supported by CNPq research grants and are members of Visgraf, the Computer Graphics Laboratory at IMPA, which is sponsored by CNPq, FAPERJ, FINEP and IBM Brasil. This work is part of Boris Mederos's doctoral thesis at IMPA.

§6. References

1. Alexa, M., Wiener filtering of meshes. Proc. Shape Modeling International 2002, pp. 51–57.
2. Alexa, M., J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, Point set surfaces, Proc. IEEE Visualization 2001, pp. 21–28.
3. Bajaj, C., and G. Xu, Anisotropic diffusion on surface and function on surfaces. ACM Trans. on Graphics **22** (2003), pp. 4–32.
4. Balyaev. A., and Y. Ohtake, Nonlinear diffusion of normals for crease enhancement. Vision Geometric X, SPIER Annual Meeting, 2001, pp. 42–47.
5. Black, M., and A. Rangarajan, On the unification of the line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision **19** (1996), pp. 57–91.

6. Clarentz, U., U. Diewald, and M. Rumpf, Anisotropic geometric diffusion in surface processing. Proc. IEEE Visualization 2000, pp. 397–405.
7. Comaniciu, D., and P. Meier, Mean Shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. and Machine Intelligence**24** (2002), pp. 603–619.
8. Desbrun, M., M. Meyer, P. Schroder, and H. A. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow. Proc. SIGGRAPH'99, pp. 317–324.
9. Desbrun, M., M. Meyer, P. Schroder, and H. A. Barr, Anisotropic feature preserving denoising of height field and bivariate data. Proc. Graphics Interface 2000, pp. 145–152.
10. Edelman. A., T. A. Arias, and S. T. Smith, The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Analysis Applied **20** (1994), pp. 202–353.
11. Fleishman, S., I. Drori, and D. Cohen-Or, Bilateral mesh denoising. Proc. ACM SIGGRAPH 2003, pp. 950–953.
12. Hansen, P, C., Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems, Numerical Algorithms **6** (1994) pp. 1–35.
13. Huber, P, J., *Robust Statistics*. John Wiley and Sons, 1981.
14. Jones, T. R., F. Durand, and M. Desbrun Non-Iterative, feature-preserving mesh smoothing. Proc. ACM SIGGRAPH 2003, pp. 943–949.
15. Levin, D., Mesh independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, edited by Brunnett, Hamann and Mueller, Springer-Verlag, 2003, pp. 37–49.
16. Mederos, B., L. Velho, L. H. de Figueiredo, Moving least squares multiresolution surface approximation. Proc. SIBGRAPI 2003, pp. 19-26.
17. Ohtake, Y., A. Balyaev, H. P. Seidel, Mesh smoothing by adaptive and anisotropic Gaussian filter applied to the mesh normal. Proc. Vision, Modeling, and Visualization 2002, pp. 203–210.
18. Pauly, M., and M. Gross, Spectral processing of point-sampled geometry. Proc. ACM SIGGRAPH 2001, pp. 279–386.
19. Peng, J., V. Strela, and D. Zorin, Simple algorithm for surface denoising. Proc. IEEE Visualization 2001, pp. 107–112.
20. Taubin, G., A signal processing approach to fair surface design. Proc. SIGGRAPH 95, pp. 351–358.
21. Tasdizen, T., R. Whitaker, P. Burchard, and S. Asher, Geometric surface smoothing via anisotropic diffusion of the normals. Proc. IEEE Visualization 2002, pp. 125–132.

22. Smith, S. T., Optimization Techniques on Riemannian Manifolds. *Fields Institute Communications* **3**, AMS, Providence, RI, 1994, pp. 113–146

Boris Mederos, Luiz Velho and Luiz Henrique de Figueiredo
IMPA – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110
22460-320 Rio de Janeiro, RJ, Brazil
boris@impa.br, lvelho@impa.br and lhf@impa.br