Point Cloud Denoising

Boris Mederos Luiz Velho Luiz Henrique de Figueiredo
IMPA–Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina 110, 22461-320 Rio Me Janeiro, RJ, Brazil
boris@impa.br lvelho@impa.br lhf@impa.br

November 6, 2003

Abstract

We present a new method for point cloud denoising. We introduce a robust smoothing operator $Q(r) = r + t^* n^*$, inspired in moving least squares and M-estimators robust statistics theory. Our algorithm can be seen as a generalization and improvement of the Fleishman et al algorithm for mesh denoising.

1 Introduction

With the increasing use of 3D scanner and the growth in complexity of this models, the necessity of robust method for denoising points cloud preserving the fine features in the surface has increased.
A highly effort has been done in the last years, resulting in variety of mesh smoothing algorithms. Taubin [10] introduce the Laplacian operator for mesh denoising, Desbrun [14] used geometric flow, this initial techniques are isotropic and unable to separate noise from sharp features.
Recently Anisotropic diffusion has been introduced by Desbrun [15] et al, Clarentz et al [9], Bajaj and Xu [9]. All this methods produces good quality results, but they have high computational time and suffer of ill conditioning of the diffusion equation in sharp features.
Balyaev and Ohtake [18], Ohtake [12] and Tasdizen et al [12] have proposed diffusion-type smoothing on the normal field producing similar results to anisotropic diffusion, but still are computational expensive
Locally adaptive Wiener filtering has been used successfully for 3D meshes by Peng et al [13], Pauly and Gross et al [12] and Alexa [7] who give a different approach similar to anisotropic diffusion. This method is fast and overcome some of the limitation above but still is based on a connected mesh.
Recently Jones et al [2] and Fleishman et al [3] introduce featured-preserving mesh smoothing based in robust statistic theory in order to discriminated sharp featured from noise. Levin [21] introduce moving least square algorithm as a mesh independent approximation procedure. Our method is inspired in the above approaches, we introduce a new point cloud denoising algorithm that work successfully in front of high noisy data set and preserve salient features of the models.
Section 2 describes related works, Section 3 explain the proposed method in details, Section 4 is reserved to study the numerical methods used and its convergence and in Section 5 we discuss the proposed method in action with severals examples.

2 Related Works

In the field of surface approximation, Levin [21] developed a mesh independent method for smooth surface approximation, the Moving Least Square (MLS), introducing a different paradigm based on a projection procedure. The method construct a local reference and a local mapping for each point near the surface, given a Manifold S and a set $\{r_i\}_{i\in I}$ on S or near of S, the smoothing manifold $\tilde{S}$ approximating the $\{r_i\}_{i\in I}$ is defined by two step projection procedure:

I) Given a point r near of S find a local hyperplane $H_r = (n_r, D_r),\ D_r = n_r^t r + t_r$, such as the following function is minimize in n and t, where $\theta(.)$ is a decreasing weight function.

$$\{n_r, t_r\} = \sum_{i\in I} (n^t(r_i - r - t\cdot n))^2 \theta(\|r_i - r - t\cdot n\|)$$

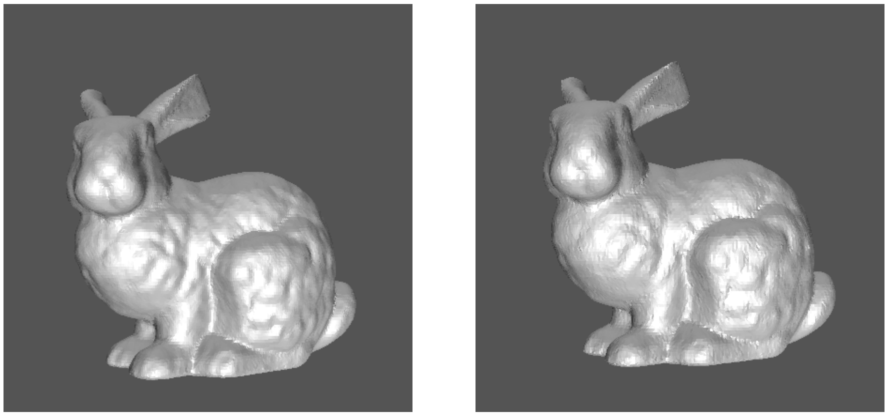
II) Find a local polynomial approximation $p(.)$ of S, defined over the hyperplane H_r such the following least square error is minimized, where the $x_i,\ i\in I$ are the projections of the $r_i,\ i\in I$ onto H_r and the $\{f_i\}$ are the heights of $\{r_i\}$ over H_r.

$$\sum_{i\in I} (p(x_i) - f_i)^2 \theta(\|r_i - r - t\cdot n\|)$$

The above procedure define a set operator $\Psi(F) = \{f + (t_f + p(0))\cdot n_f \mid \forall f\in F\}$, the approximation manifold $\tilde{S}$ is the fixed point of the operator $\Psi(.)$.

Recently works on mesh smoothing are based on robust statistics, this is concerned with the statistical estimators that are robust to the deviation from theoretical distribution and to the occurrence of gross errors. Tomasi and Menduchi [22] proposed Bilateral filters as a robust statistic technique for edge-preserving image filtering, that is a non-iterative robust estimator - w-estimator, Huber [16]. Inspired on the Ideas of Bilateral filters Jones [2] and Fleishman [3] developed a non-iterative - w-estimator for mesh denoising, the extension from images to mesh filter is far from trivial because the nature of the surface, where the signal and the domain are confusing. Both method are based on w-estimator $\theta^1 = \frac{\sum w(x_i-\theta)x_i}{\sum w(x_i-\theta)}$, where the new position $p^{'}$ of a vertex p is computed as a weighted sum of the predictor from its spatial neighborhood.

In Jones method the new position $p^{'}$ is computed as $p^{'} = \frac{\sum_{q\in N(p)} a_q w_g(\Pi_q(p)-p) w_f(\|q-p\|)\Pi_q(p)}{\sum_{q\in N(p)} a_q w_g(\Pi_q(p)-p) w_f(\|q-p\|)}$, where the predictor $\Pi_q(p)$ is the projection of q in the tangent plane of the surface at q. In Fleishman method a local parameter space is determined using a tangent plane at the vertex p and the neighborhood of p can be seen as a the height field over the tangent plane, the position of the new vertex is computed as $p^{'} = p + t_p\cdot n_p$, where $t_p = \frac{\sum_{q\in N(p)} w_g(I(q)) w_f(\|q-p\|) I(q)}{\sum_{q\in N(p)} w_g(I(q)) w_f(\|q-p\|)}$ is the weighted average of the heights $I(q)$ of the points q again the tangent plane, the $I(q)$ are used as the predictors. The two methods uses the parameters σ_g and σ_f on the gaussian weights $w_g()$ and $w_f()$ to control the level of smoothing , the parameter σ_f control the smoothing in the domain and σ_g control when the predictors can be considered outlier.



a) b)

Figure 1: Results of apply our method a) and moving least square method b) to a noisy bunny model . b) does not preserve sharp features

3 Proposed Method

Given a noise point cloud $\{q| \ q \in S\}$ from a surface S and a point $r \in S$, our goal is to approximate r to the original surface S using a projection procedure similar to MLS. Given $r \in S$ we estimate a new position $r^{'} = r + t_r \cdot n_r$, where n_r is a direction with $\|n_r\| = 1$ and t_r a displacement along the direction. The parameters t_r and n_r are estimated using the moving heights h_q of the points q in the neighborhood $N(r)$. The moving heights $h_q = n^t(q - r - tn)$ are determined as the height of the point q with respect to the plane with normal n passing at the point $r + tn$.

In order to calculate $r^{'}$ we propose the following m-estimator $\sum_{q \in N(r)} g(h_q) w(\|q - r\|)$ with $\|n\| = 1$. This can be seen as a way to smooth the point r and to estimate a normal at $r^{'}$ robust to gross deviation of the points q, where the influence of the outliers is controlled by the robust error norm $g(.)$ that give low weight to outlier(points $q \in N(r)$ with high values of the height h_q). Several examples of robust error norms can be found in the literature [24]: 1) Huber's Minimax, 2) Tukey's biweight, 3) Hampel's function, 4) Gaussian error norm, all these robust potential in general grows at a slower rate than the quadratic error norm ($\lim_{t \to \infty} \frac{g^{'}(t)}{2t} = 0$). In our particular case we use a Gaussian error norm $g_{\sigma_g}(x) = \sigma_g - \sigma_g e^{-\frac{x^2}{2\sigma_g}}$ and a weight function $w(x) = e^{-\frac{x^2}{2\sigma_w}}$ producing good results, see figure (1) above for a comparison between our proposed method and the moving least square method.

Our method start computing the neighborhood $N(r)$ by a growing neighborhood process. In order to determine $N(r)$ we initially compute a subset $S = \{s \mid \|s - r\| < \sigma_w\}$ of the k-nearest neighbors of r with distance to r smaller than σ_w, the set S is augmented $S = S \cup N_s$ for each element $s \in S$ not processed, where N_s is a subset of the k-nearest neighbors of s (a typical value $k = 8$) that are not in S with distance to r smaller than the parameter σ_w, This process is repeated for each new point in S not processed until there are not more points to be added, then the final neighborhood is $N(r) = S$. Observe that determine the neighborhood in this way tend to eliminate points that are in a different connected component and in different side of a thin region.

Once we have computed the neighborhood, we apply the smoothing operator $Q(r) = r + t^* n^*$, where n^* and t^* satisfy the following minimization problem, subject to $\|n\| = 1$.

$$\{n^*, t^*\} = \min_{\{n,t\}} \sum_{q \in N(r)} g(n^t(q - r - tn))w(\|q - r\|) \tag{1}$$

The optimal values n^*, t^* are computed interchanging minimization with respect to n and t until we are close to the minimum value, it can be summarized in the following algorithm:

Algorithm 3.1

$t_0 = 0 \;\; k = 0$
repeat
i) Compute n_{k+1} as the minimum of (1) with respect to n and $t = t_k$ fixed
ii) Compute t_{k+1} as the minimum of (1) with respect to t and $n = n_{k+1}$ fixed
$k = k + 1$
until we are close to the minimum.

The stage i) is devoted to solve a constrained optimization problem, we minimize the equation (1) with respect n with fixed t subject to the restriction $\|n\|^2 = 1$. In order to solve this problem we will used the newton method over Riemannian manifolds, see Smith [21] and Edelman [6]. Given a C^∞ manifold S in our case the sphere S^2 with Riemannian structure g, Levi-Civita conecction ∇ and a C^3 function f over S we have the following algorithm:

Algorithm 3.2 Newton Method

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

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

As is known the newton method has a fast convergence of order $O(n^2)$ starting near the solution, for that reason we use as initial input vector the computed normal one step before n_{k-1} for $k \neq 0$ and in the first iteration $k = 0$ we used as initial estimative the solution of equation (1) with $t = 0$, obtaining a robust method to compute the normals at the point cloud. In order to determine this initial normal we compose the Lagrange equations of (1) for $t = 0$ and its derivate with respect to n:

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

$$L_n(n, \lambda) = \sum_{q \in N(r)} \Psi_w(\|q - r\|)\Psi_g(n^t(q - r))(q - r)(q - r)^t n - \lambda n = 0 \tag{3}$$

where $\Psi_g(x) = \frac{g'(x)}{2x\sigma_g} = e^{-\frac{x^2}{2\sigma_g}}$ and $\Psi_w(x) = w(x) = e^{-\frac{x^2}{2\sigma_w}}$ are weight functions respectively. , the equation (3) can be represented in a matrix form $M(n)n = \lambda n$ where $M(n)$ is the symmetric matrix with elements depending of the vector n

$$M(n) = \sum_{q \in N(r)} \Psi_g(\|q-r\|)\Psi_w(n^t(q-r))(q-r)(q-r)^t \tag{4}$$

In the matrix $M(n)$ the influence of the factor $(q-r)(q-r)^t$ is weighted by the height $\tilde{h}_q = n^t(q-r)$ of the point q with respect to the hyperplane with normal n passing at r. A way to solve this equation is to use the following, iterative scheme

$$M(n_k)n_{k+1} = \lambda_{k+1}n_{k+1} \tag{5}$$

where λ_{k+1} is the smallest eigenvalue of $M(n_k)$ and n_{k+1} it's associate eigenvector. The traditional covariance analysis is obtained for the starting normal $n_0 = 0$ $(M(0)n_1 = \lambda_1 n_1)$. Few iteration of this method produce good results.

The stage ii) of algorithm 3.1 correspond to minimize (1) with respect to t keeping the normal n fixed which is an unconstrained optimization problem, looking for the necessary first-order condition for the minimum conduce to equation (6), where $\Psi_g(x) = \frac{g'(x)}{2x\sigma_g} = e^{-\frac{x^2}{2\sigma_g}}$ and $\Psi_w(x) = w(x) = e^{-\frac{x^2}{2\sigma_w}}$ are weight functions respectively.

$$\sum_{q \in N(r)} \Psi_w(\|q-r\|)\Psi_g(h_q)(n^t(q-r)-t) = 0 \tag{6}$$

Solving this equation in the variable t yield the recurrence equation (7) with $\tilde{h}_q = n^t(q-r)$ and the normalization factor $k_{t_i} = \sum_{q \in N(r)} \Psi_g(n^t(q-r-t_i n))\Psi_w(\|q-r\|)$,

$$t_{i+1} = k_{t_i}^{-1} \cdot \sum_{q \in N(r)} \Psi_g(n^t(q-r-t_i n))\Psi_w(\|q-r\|)\tilde{h}_q \tag{7}$$

note that the first iteration $k = 1$ of the algorithm 3.1 above with only one iteration $(i = 1)$ in the sequence t_i, $t_1 = k_{t_0}^{-1} \cdot \sum_{q \in N(r)} \Psi_g(\tilde{h}_q)\Psi_w(\|p-r\|)\tilde{h}_q$ is exactly the Fleishman et al [3] method.

4 Numerical Methods

4.1 Theory

In this section we will analyze theoretical aspects on the convergence of the iterative method proposed in above section. Now we proof that the sequence (7) $t_{i+1} = k_{t_i}^{-1} \cdot \sum_{q \in N(r)} \Psi_g(n^t(q-r-t_i n))\Psi_w(\|q-r\|)\tilde{h}_q$ corresponding to $t_{i+1} = f(t_i)$ with $f(t) = k_t^{-1} \cdot \sum_{q \in N(r)} \Psi_g(n^t(q-r-tn))\Psi_w(\|q-r\|)\tilde{h}_q$ is converging.

Lemma 1 *The sequence $t_{i+1} = f(t_i)$ converge and is strictly monotone increasing(decreasing) $t_i < t_{i+1}$ $(t_i > t_{i+1})$ for all i*

Proof. The sequence t_i is a particular case of a location M-estimator see [23], applying Theorem 2 from [23] we have that the sequence t_i satisfy the inequality below

$$(t_i - t_{i-1})(t_{i+1} - t_i) > 0$$

concluding that $t_i - t_{i-1}$ and $t_{i+1} - t_i$ have the same sign, therefore the sequence t_i is monotone increasing or decreasing.
Because the values t_i are convex combination of the $\tilde{h}_q$ and $|\tilde{h}_q| = |n^t(q-r)| < \|q-r\| < \sigma_w$ we have that t_i is inside the interval $[-\sigma_w, \sigma_w]$. The convergence of t_i follow from this fact and the monotonicity of the sequence.

□

Because the sequence t_i converge we have that $\tilde{t} = \lim t_i$ is a fixed point of the function $f(t), (f(\tilde{t}) = \tilde{t})$, hence it is a solution of the first order condition (6)

In the case that we want to compute a normal n at a point r, it is equivalent to solve (1) with $t = 0$ as we explain above, in other words find

$$n^* = argmin\ F(n) \tag{8}$$

$$F(n) = \sum_{q \in N(r)} g(n^t(q-r))w(\|q-r\|) \tag{9}$$

This conduce to the iterative problem $M(n_k)n_{k+1} = \lambda_{k+1}n_{k+1}$ proposed in the above section. The sequence $\{n_k\}_{k=1,2,...}$ alway decrease the objective function $F(n)$ converging to stationary point that is not a maximum, as the next lemma proof.

Lemma 2 *Given $M(n_k)n_{k+1} = \lambda_{k+1}n_{k+1}$ with λ_{k+1} the minimum eigenvalue of $M(n_k)$ with associated orthonormal eigenvector n_{k+1}, then the sequence $\{F(n_k)\}$ is decreasing when k increase.*

Proof.

$$\begin{aligned} F(n_{k+1}) - F(n_k) &= \sum_{q \in N(r)} (g(n_{k+1}^t(q-r)) - g(n_k^t(q-r)))w(\|q-r\|) \qquad (10) \\ &= \sigma_g \cdot \sum_{q \in N(r)} (e^{-\frac{(n_k^t(q-r))^2}{2\sigma_g}} - e^{-\frac{(n_{k+1}^t(q-r))^2}{2\sigma}})w(\|q-r\|) \qquad (11) \end{aligned}$$

Because $L(x) = e^{-x}$ is a convex function then follow that $L(x_1) - L(x_2) < L^{'}(x_1)(x_1 - x_2)$, applying this inequality to (11) we obtain

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

But $\lambda_{k+1} - n_k M(n_k) n_k < 0$ due to $\lambda_{k+1} = \min x^t M(n_k) x^t$ with $\|x\| = 1$ then the lemma follow. □

4.2 Newton Method over S^2

In this section we present the newton method on the sphere in more detail. Using that in the sphere S^2 the geodesics satisfy the following second order equation $\ddot{x}^k + x^k = 0,\ k = 1, ..., n$, we obtain that the Cristoffel symbols are given by $\Gamma^k_{i,j} = \delta_{i,j} x^k$, with $\delta_{i,j}$ the Kronecker symbol, then the $ijth$ component of the bilinear form $(\nabla^2 f)_n$

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

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 above equation in a matrix form we obtain:

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

where $H_f(n),\ H_h(n)$ are the Hessian of $f(x)$ and the restriction $h(x) = \|n\|^2 - 1$ respectively, the dot product $\lambda^{'} = n^t f_n$ is the Lagrangian multiplier and the expression for the gradient is $(\nabla f)_n = \nabla f - \lambda^{'} n$.

Observe that the well known necessary condition at the minimum $x(0) = n^*$ with $x(t)$ a geodesic are equations (12) and (13), where we denote $(\nabla^2 f)_n$ and $(\nabla f)_n$ by L_{nn} and L_n respectively,

$$\frac{d}{dt} f(x(t)) = f^t_{n^*} \cdot \dot{x} = (f^t_{n^*} - \lambda^{'} n^*)^t \cdot \dot{x} = L^t_{n^*} \cdot \dot{x} = 0 \qquad (12)$$

$$\frac{d^2}{dt^2} f(x(t)) = \dot{x} L_{n^* n^*} \dot{x} \geq 0 \qquad (13)$$

Now we explain what is the mining of the equation $H_i = -(\nabla^2 f)^{-1}_{n_i} (\nabla f)_{n_i}$, given a a linear operator $A : R^3 \rightarrow R^3$ it define an operator on the tangent plane $T_n S^2$ for each n in S^2 such that $A \cdot u = Au - (n^t A n)n = (I - nn^t)Au$. Then the solution of the system $A \cdot u = v$ with $u,\ v$ in $T_n S^2$ has the form

$$u = A^{-1} v - \alpha^{-1}(A^{-1} x)(x A^{-1} v),\ \ \alpha = x^t A^{-1} x \qquad (14)$$

Rewriting the algorithm 3.2 in this context we obtain:

Algorithm 4.1 Newton Method in S^2

$i = 1$, find an initial point $n_1 \in S^2$
repeat
  i) Compute $v_i = L^{-1}_{n_i n_i} L_{n_i} - \alpha^{-1}(L^{-1}_{n_i n_i} \cdot n_i)(n_i \cdot L^{-1}_{n_i n_i} \cdot L_{n_i}),\ \ \alpha = n^t_i \cdot L^{-1}_{n_i n_i} n_i$
  ii) Find $n_{i+1} = cos(\|v_i\|) n_i + sen(\|v_i\|) \frac{v_i}{\|v_i\|}$
  $i = i + 1$
until we are close to the minimum.

In step i) we solve a linear system $L_{kk} \cdot v_k = L_k$ with $v_k, L_k \in T_{n_k} S^2$, applying equation (14) we obtain $v_k = L^{-1}_{n_k n_k} L_{n_k} - \alpha(L^{-1}_{n_k n_k} \cdot n_k)(n_k \cdot L^{-1}_{n_k n_k} \cdot L_{n_k}),\ \ \alpha = n^t_k \cdot L^{-1}_{n_k n_k} n_k$, where the linear systems $L_{n_k n_k} x = L_{n_k}$ and $L_{n_k n_k} x = n_k$ are in general ill conditioned, so we used the L-curve and GSVD methods proposed in [5] to solve ill conditioned problems. The step ii) correspond to make $t = 1$ in the geodesic $\gamma(t) = exp_{n_k}(t v_k)$, that in the sphere S^2 take the form $n_{k+1} = cos(v_k) n_k + sen(v_k) \frac{v_k}{\|v_k\|}$.

5 Results

Our method has been tested on several data set, the results are shown in figure $(2-3)$. All the models were contaminated with gaussian noise of 0 mean and variance (15 of a $\frac{1}{10}\times$ bounding box diagonal) alone the normal direction.

The parameters $\sigma_g,\ \sigma_w$ for the dragon and igea model were $\sigma_g = h,\ \sigma_w = 2h$, where h is the mean spacing between the points, the parameter for the bunny model was $\sigma_g = 0.4h,\ \sigma_w = 1.5h$ and for the fandisk $\sigma_g = 0.12h$, $\sigma_w = 2.0h$. In all the models the diameter of the neighborhood was set to σ_w. As in the methods [2, 3] a small value in the parameter σ_w lead to faster computation because the neighborhood $N(r)$ is small. Large value may cross sharp features. The parameter σ_g control when a point q in the neighborhood $N(r)$ is an outliers, for small values of σ_g the small features of the model are kept and for larges values only salient features are preserved.

6 Conclusion

We present a new method for point cloud denoising without any connectivity information in the input data set, we give theoretical justification on the convergence of the numerical method used to solve this problem. The proposed method deals with irregular points cloud and does not perform any parametrization, It's robust again outliers and behave satisfactory again strong noise.

Acknowledgments. The authors are partially supported by CNPq research grants. The authors 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.

References

[1] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH'92*. pp. 71–78.

[2] Thouis R. Jones, Fredo. Durand, Mathieu Desbrun Non-Iterative, Feature-Preserving Mesh Smoothing. *SIGGRAPH'2003*.

[3] Shashar. Fleishman, Iddo. Dori, Daniel Cohen - Or Bilateral Mesh Denoising. *SIGGRAPH'2003*.

[4] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics* 9(1):3–15, 2003.

[5] P. C. Hansen. The L-curve and its used in the numerical treatment of inverse problems. *Computational Inverse Problems in Electrocardiology, WIT Press, Southampton, 2001; pp. 119-142.*

[6] A. Edelman, T. A. Arias and S. T Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis Applied* Vol 20, No 2, pp, 202-353.

[7] Alexa, M. 2002. Wiener Filtering of Meshes. *In Proceedings of Shape Modeling international, 51-57.*

[8] Bajaj, C., AND Xu, G. Anisotropic diffusion on surface and function on surfaces. *ACM Transaction on Graphics 22, 1, pp 4-32.*

[9] Clarentz, U., Diewald, U., and Rumpf, M. 2000. Anisotropic geometric diffusion in surface processing. *In IEEE Visualization 2000, 397-405.*

[10] Taubin, G. 1995. A signal processing approach to fair surface dising. *In Proceedings of SIGGRAPH 95, 351-358.*

[11] M. Pauly, M. Gross. Spectral processing of point-sampled geometry. *In Proceedings of ACM SIGGRAPH 2001, 279-386*

[12] Tadsdizen, T., Whitaker, R., Burchard, P., and Osher, S. Geometric surface smoothing via anisotropic diffusion of the normals. *In Proceeding, IEEE visualization 2002, 125-132.*

[13] Peng, J.., Strela, V., and Zorin, D. 2001. Simple algorithm for surface denoising. *Tech. rep.*

[14] Desbrun, M., Meyer, M., Schoder, P., and Barr, A. H. Implicit Fairing of irregular meshes using diffusion and curvature flow. *In Proceedings of SIGGRAPH 99, 317-324.*

[15] Desbrun, M., Meyer, M., Schoder, P., and Barr, A. H. Anisotropic feature preserving denoising of height field and bivariate data. *In Graphics Interface, 145-152.*

[16] Huber, P, J. 1981. *Robust Statistics. John Wiley and Sons.*

[17] Balyaev. A., and Ohtake, Y . 2001. Nonlinear Diffusion of Normals for Crease Enhancement. *In Vision Geometric X, SPIE Annual Meeting, 42-47.*

[18] Ohtake, Y, Balyaev. A., Seidel, H, P. 2002 Mesh smoothing by adaptive and anisotropic gaussian filter applied to the mesh normal. *In Vision Modeling and Visualization(Erlangen) Germany.*

[19] Levin, D. Mesh independent surface interpolation. *In advances in Computational Mathematics, in Press*

[20] Smith, S, T. Optimization Techniques on Riemmanian Manifolds. *In Fields Institute Communications, Vol. 3, AMS, Providence, RI, 1994, pp. 113-146*

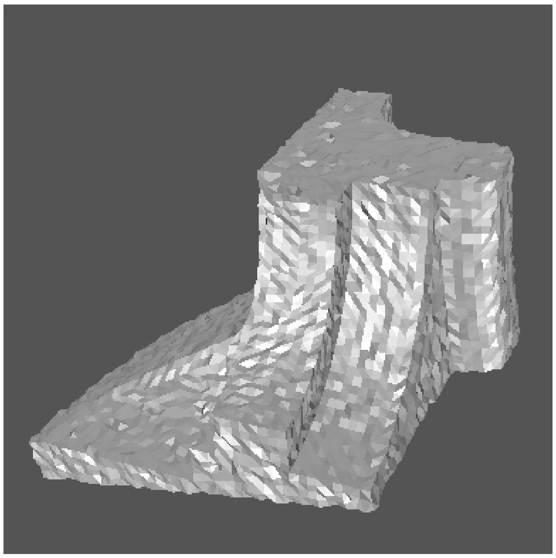
[21] Hampel, F. R., Ronchetti E. M S., Rousseuw, P. J., and Stahel, W. A. 1986. Robust Statistics: The Approach based on influence functions. *John Wiley and Sons. ISBN 0471-63238-4*

[22] Tomassi, C., and Mendushi, R. 1998 . Bilateral filtering for gray and color images. *In ICCV, 839-846*

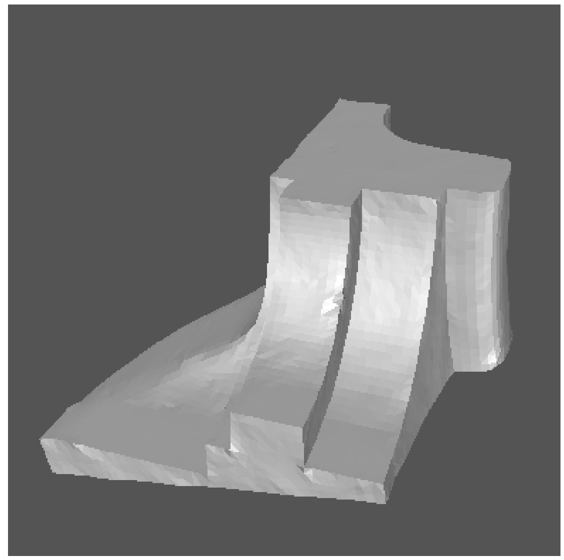
[23] Comaniciu, Dorin and Meer, Peter. 2002 . Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE transaction on pattern analysis and machine intelligence, Vol 24, No 5, May 2002*

[24] Black, M and Rangarajan, A: On the Unification of the line Processes, Outlier Rejection, and Robust Statistics with applications in Early Vision. *International Journal of Computer Vision 19(1), 57-91(1996)*

Figure 2: Fandisk and Bunny Model



Noisy Fandisk



Denoise Model

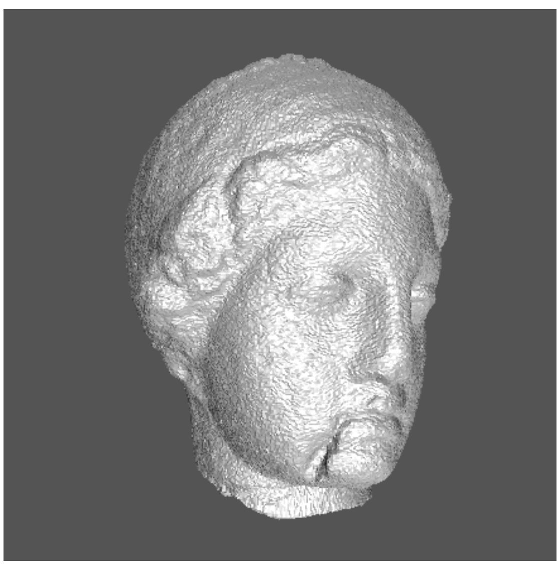
Noisy bunny

Denoise Model

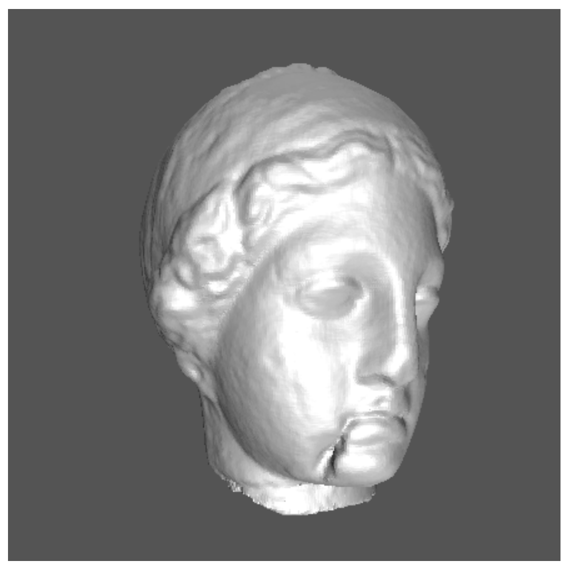
Figure 3: Dragon and Venus Model

Noisy Dragon

Denoise Dragon



Venus with Noise



Denoise Venus