Global Mesh Denoising with Fairness

Sk. Mohammadul Haque, Venu Madhav Govindu
Indian Institute of Science
Bengaluru, India
{smhaque, venu}@ee.iisc.ernet.in

Abstract

We propose a novel global 3D mesh denoising method that is carried out in two steps, i.e. mollification of normals followed by vertex correction. Both steps involve minimising sparse, quadratic cost functions that yield efficient non-iterative solutions. In the mollification step, we use adaptive weights that allows for appropriate diffusion while preserving features. While many existing methods only correct for the vertex position along the normal direction, we argue that this is inadequate in many scenarios. Instead, we allow for vertex correction in all directions while enforcing a novel face fairness penalty that preserves face shapes in the denoised mesh. We present a number of tests and examples that demonstrate the efficacy of our method in denoising while preserving face fairness. We demonstrate the superiority of our approach over some relevant methods in the literature.

1. Introduction

In recent years it has become easier to acquire 3D data using either depth cameras or by solving the dense multiview stereo problem using RGB images. Since such depth data contains significant amounts of noise, all 3D reconstruction or application pipelines need to carry out an important denoising step. In the case of depth maps, the 3D representations are dense with a depth value for every pixel. For such a pixel grid parametrisation, it is implicitly assumed that the observation noise in a depth map representation is only along the direction of the ray emanating from a given depth pixel. In other acquisition modalities which generate point clouds, we may assume the noise to be present in all directions about a 3D point and not only along the ray representing the homogeneous parametrisation of a pixel. This is the case for methods that carry out stereo triangulation.

There are a large number of methods for mesh denoising proposed in the literature. We may classify most of these approaches into a) local, or b) global methods. In *local* methods the correction for the noisy mesh is applied locally, resulting in approaches that are iterative in nature [5, 6, 15, 16, 18, 21]. An important drawback of these approaches is the difficulty in defining the number of iterations required for optimal denoising. Often local methods also result in artefacts like geometric distortion and surface shrinkage [12, 18]. In *global* methods, a global cost function is optimised [2, 9, 10, 12, 13, 14, 21]. This typically involved solving a sparse system of equations that are usually linear in nature [12]. The method proposed in this paper is a global one.

Many mesh denoising algorithms ignore the true distribution of observation noise in depth representations and assume the noise to be restricted along the direction of the surface normal. As a result, they correct for the position of a mesh vertex (point) by moving it along this normal direction. The noise component in the tangent plane about a surface point is completely ignored. This works well for low noise scenarios and is reasonable from a mesh fairing perspective [6]. But at higher noise levels, the noise component in the tangent plane leads to a severe distortion of the face shapes in the mesh, including face flipping with the surface normals being forced to point into a surface rather than out of it.

Apart from carefully accounting for the presence of noise in different directions, mesh denoising methods should also avoid typical problems such as volume shrinkage and smoothing over surface features such as edges and corners. Although the classical Laplacian mesh smoothing method [5] does account for noise in all directions, methods based on such Laplacian smoothing do not preserve surface features and are also affected by the problem of volume shrinkage [15, 18]. More recent works that present methods designed to preserve features include [1, 4, 8, 10, 11, 15, 16, 19, 20, 21]. The methods by He *et al.* [8] and Cheng *et al.* [1] are good for piecewise flat surfaces. However, they result in artificial edges in

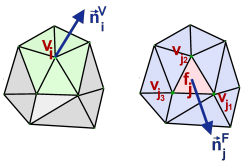

Figure 2. Conventions used in the paper: The left image shows a portion of a mesh where $\mathbf{v}_i$ is a vertex with a normal $\mathbf{n}_i^V$. The green shaded area around $\mathbf{v}_i$ is the face neighbourhood of $\mathbf{v}_i$ used in the paper. The right image represents the same portion of a mesh where $\mathbf{f}_j$ is a face with vertices $\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3}$. This face has a normal $\mathbf{n}_j^F$ defined at its centroid. The blue shaded area consisting of all the faces around $\mathbf{f}_j$ is the face neighbourhood used in the paper.

smooth regions and hence work poorly on natural data. Moreover the method of He *et al.* [8] solves a sequence of minimisations which is expensive. Most importantly, all these methods consider denoising only along the normals and hence end up retaining a significant amount of noise in the tangential directions in their solutions.

In addition to reducing noise, an important attribute of any denoising method is its ability to preserve the quality of the mesh triangles, *i.e.* mesh fairness. As pointed out in [15], in most of the literature, 3D mesh denoising, smoothing and fairing are used interchangeably which is incorrect.

In this paper, apart from being robust to depth discontinuities, we seek to account for the presence of noise in all directions by using a scheme which explicitly induces fairing of face shapes in the denoised output mesh. Additionally, while a large number of denoising methods are iterative in nature, our approach solves a global cost function that is quadratic in nature, resulting in a method that is in principle non-iterative.

In Section 2, we briefly describe the notation and definitions used in this paper. A brief discussion of a class of approaches to vertex correction sets the stage for our method. Section 3 describes our approach for mesh denoising that is based on minimising a global cost function that is quadratic and sparse in nature and explicitly incorporates a measure of face fairness. While the two steps of mollification and vertex correction are described in Sections 3.1 and 3.2 respectively, in Section 3.3 we present the salient properties of our approach and also compare them with other relevant methods in the literature. Section 4 presents extensive results of our method on a variety of datasets and also compares our performance with that of other related techniques in the literature.

2. Preliminaries

Consider a clean oriented surface $\mathbf{S}_0$ and let $\mathbf{M}_0 = (\mathbf{V}_0, \mathbf{E}_0, \mathbf{F}_0)$ be a oriented mesh sampled from $\mathbf{S}_0$. Here $\mathbf{V}_0 = \{\mathbf{v}_{0i}\}_{i=1}^{N_V}$ is the noise-free set of 3D points in $\mathbf{S}_0$, $\mathbf{E}_0 = \{\mathbf{e}_i | \mathbf{e}_i = (\mathbf{v}_p, \mathbf{v}_q) \in \mathbf{V}_0 \times \mathbf{V}_0\}_{i=1}^{N_E}$ is the set of edges and $\mathbf{F}_0 = \{\ \mathbf{f}_i | \mathbf{f}_i = (\mathbf{v}_p, \mathbf{v}_q, \mathbf{v}_r) \in \mathbf{V}_0 \times \mathbf{V}_0 \times \mathbf{V}_0$ and $(\mathbf{v}_p, \mathbf{v}_q), (\mathbf{v}_q, \mathbf{v}_r), (\mathbf{v}_r, \mathbf{v}_p) \in \mathbf{E}_0\ \}_{i=1}^{N_F}$ is the set of triplets defining the faces of the mesh.

The observation model for any vertex $\mathbf{v}_i$ is $\mathbf{v}_i = \mathbf{v}_{i0} + \mathbf{s}_i$ where $\mathbf{v}_{i0} \in \mathbf{V}_0$ is the true noise-free vertex and $\mathbf{s}_i$ is *i.i.d.* Gaussian noise. Hence, the noisy mesh can be represented as $\mathbf{M} = (\mathbf{V}, \mathbf{E}_0, \mathbf{F}_0)$ where $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{N_V}$ is the noisy set of vertex position measurements. For any face $\mathbf{f}_j$, the normal vector at its centroid is denoted as $\mathbf{n}_j^F$ and for any vertex $\mathbf{v}_i$, the normal vector is denoted as $\mathbf{n}_i^V$. This distinction between two types of normals is also schematically illustrated in Fig. 2. The boundary of a mesh $\mathbf{M}$ is denoted $\partial\mathbf{M} = (\mathbf{V}^B, \mathbf{E}^B)$ where $\mathbf{V}^B \subset \mathbf{V}$, $\mathbf{E}^B = \{(\mathbf{v}_p, \mathbf{v}_q)\} \subset \mathbf{E}$ and $\mathbf{v}_p, \mathbf{v}_q \in \mathbf{V}^B$. Since we consider only triangle meshes, we will use the terms triangle and face interchangeably. By $\mathcal{N}(\mathbf{x})$, we denote some neighbourhood operator on the entity $\mathbf{x}$ which is either a vertex or a face, depending on the context.

Before describing our approach, it is instructive to consider a commonly used vertex modification step:

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \sum_j w_{ij} \mathbf{A}_{ij} \mathbf{v}_j \tag{1}$$

where $\hat{\mathbf{v}}_i$ is the estimated vertex, $\mathbf{A}_{ij}$ is a linear operator defined locally around $\mathbf{v}_i$ and w_{ij} are the weights on the corresponding neighbouring vertices $\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)$. Such a vertex modification is often iteratively applied [6, 3, 17]. The weights w_{ij} and the local operator $\mathbf{A}_{ij}$ vary depending on the algorithm. In Laplacian smoothing [5], $\mathbf{A}_{ij} = \mathbf{I}$, *i.e.* the identity operator which is isotropic in nature, weights w_{ij} can be either constant or depend on the corresponding face areas or cotangents. Such Laplacian smoothing does not preserve features and also results in high volume shrinkage. These problems are mitigated to an extent by bilateral mesh filtering [6] where w_{ij} is set to the bilateral weights and $\mathbf{A}_{ij}$ is the orthogonal projection onto the normal direction which is anisotropic in nature.

The operator $\mathbf{A}_{ij}$ depends on the predicted local geometry which is more sensitive to noise than the vertex positions themselves [11]. Consequently, a mesh denoising algorithm always requires a *mollification* step for smoothing normals before actually performing the *vertex correction* step. Mollification is often implemented in two different ways [15], *i.e.* a) methods that iteratively

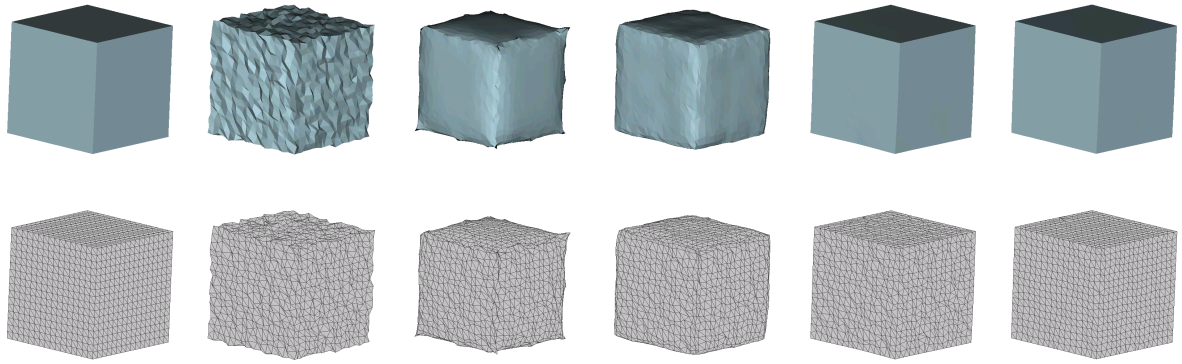

Figure 1. Denoised mesh quality of different methods on a cube ($N_v = 1538, N_F = 3072$) corrupted with isotropic Gaussian noise with standard deviation $\sigma = 0.15 \times$ mean edge length. The first row shows the surface quality and the second row depicts the mesh quality. The columns correspond to the ground truth mesh, noisy mesh, bilateral filtered [6], filtered output of Jones *et al.* [11], output of Sun *et al.* [15] and finally output of our method respectively.

improve the normals and vertex positions in an interleaved fashion, or b) methods that first mollify the normals and then correct the vertex positions. In the latter case, we observe that when the noise is moderately high, due to the large discrepancy between the mollified normals with respect to the noisy input vertices, the output mesh ends up with faces that are folded. This is due to the fact that denoising methods such as [15] move the noisy vertices $\mathbf{v}_i$ only along the normal directions, resulting in an inversion of the face normals and irregular face shapes.

3. Our Denoising Method

In most of the normal mollification methods, either the amount of normal smoothing is inadequate or it is non-robust leading to the loss of surface features [11]. In the case of iterative methods, selecting the number of iterations is not intuitive, *e.g.* [15] where two different numbers of iterations are used. In our method, while we use the two steps of mollification and vertex correction, each step is posed as the minimisation of a global cost function. Being quadratic and sparse, our cost functions have closed-form (non-iterative) solutions which can be efficiently computed. By explicitly incorporating a face fairness penalty term, our vertex correction method removes noise both along the normal as well as in the tangent plane about a vertex. Such an approach, when coupled with a careful design of the data-adaptive weights of our cost function leads to a high quality denoising method which also ensures face fairness.

3.1. Mollification

The first step of our denoising approach is a mollification of the normals. As explained in the previous section, the local operators $\mathbf{A}_{ij}$ and weights w_{ij} depend on local surface properties and need to be estimated from the noisy mesh $\mathbf{M}$. Hence, while the neighbourhoodness or the topology of $\mathbf{M}$ is assumed to be unaltered, in our method we only require to mollify the set of face normals $\left\{\mathbf{n}_i^F\right\}_{i=1}^{N_F}$. However unlike previous methods like [11, 15] we minimise a global cost function defined on the set of face normals, to obtain $\left\{\hat{\mathbf{n}}_i^F\right\}_{i=1}^{N_F}$ which is a smoothened version of the noisy face normals.

For each face normal $\mathbf{n}_i^F$, our cost function contains two terms, *i.e.* i) a data term $d_o^N\left(\hat{\mathbf{n}}_i^F, \mathbf{n}_i^F\right)$ which applies a quadratic penalty to the difference between the observed and estimated normals and ii) a weighted quadratic smoothness term over a local neighbourhood $\sum_{j \in \mathcal{N}_F(i)} w_{ij}^2 d_s^N\left(\hat{\mathbf{n}}_j^F, \hat{\mathbf{n}}_i^F\right)$ which induces local anisotropic smoothness. By adding up all the terms for each face normal, *i.e.* summing over index i, our solution for global mollification becomes one of minimising

$$\sum_{i=1}^{N_F} d_o^N\left(\hat{\mathbf{n}}_i^F, \mathbf{n}_i^F\right) + \lambda_N \sum_{i=1}^{N_F} \sum_{j \in \mathcal{N}_F(i)} w_{ij}^2 d_s^N\left(\hat{\mathbf{n}}_j^F, \hat{\mathbf{n}}_i^F\right)$$
$$\text{subject to } ||\hat{\mathbf{n}}_i^F||^2 = 1,\ i = 1, 2, \cdots, N_F \tag{2}$$

where λ_N is a regularising parameter depending on the noise variance and the face neighbourhood operator $\mathcal{N}_F(i)$ is defined as the set of faces which share a common vertex with the face $\mathbf{f}_i$ which is depicted in Fig. 2. As we wish to be robust in preserving edges, our weighting function is the

same as that of [15], *i.e.*

$$w_{ij}\left(\hat{\mathbf{n}}_j^F, \hat{\mathbf{n}}_i^F\right) = \begin{cases} \left(\hat{\mathbf{n}}_j^{F,T}\hat{\mathbf{n}}_i^F - t\right) & \text{if } \hat{\mathbf{n}}_j^{F,T}\hat{\mathbf{n}}_i^F > t \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where t is a threshold parameter.

While ideally we should use the geodesic distance of the normals on the unit sphere $\mathcal{S}^2$ in d_o^N and d_s^N in Eqn. 2, for the sake of efficiency we use the quadratic error metric which does not affect the quality of the mollified normals in any significant manner. The cost function in Eqn. 2 is minimised using gradient descent. In the case where the noise level is moderate or high, we recompute weights w_{ij} at every iteration of our gradient descent approach.

3.2. Vertex Correction

Once we obtain an estimate of the face normals $\left\{\hat{\mathbf{n}}_i^F\right\}_{i=1}^{N_F}$, we can apply our vertex correction step that is described in this subsection. Most of the previous denoising methods update the vertex positions only in the normal direction. As we demonstrate in Sec. 4.1, for many existing methods this leads to irregular and folded faces in the denoised mesh which in turn results in shadow artefacts when we use smooth shading rendering in graphics pipelines.

Notationally, the set of N_V vertices $\{\hat{\mathbf{v}}_i\}_{i=1}^{N_V}$ can be collected into a single concatenated vector denoted as $\mathbf{V}$. In the following, $\mathbf{V}$ denotes the set of observed vertices and the estimated set of vertices after vertex correction as $\hat{\mathbf{V}}$. To ameliorate the problems discussed above, in our vertex correction step, we minimise a global cost function $C_V(\hat{\mathbf{V}})$ which has three terms, *i.e.*, a data term $d_o^V(\hat{\mathbf{V}}, \mathbf{V})$, a smoothness term $d_s^V(\hat{\mathbf{V}})$ and a face fairness term $d_f^V(\hat{\mathbf{V}}, \mathbf{V})$. The resulting cost function to be minimised is

$$C_V(\hat{\mathbf{V}}) = d_o^V(\hat{\mathbf{V}}, \mathbf{V}) + \lambda_V d_s^V(\hat{\mathbf{V}}) + \eta d_f^V(\hat{\mathbf{V}}, \mathbf{V}) \quad (4)$$

where λ_V are η are parameters depending only on the type and amount of noise. Here, the data term is a simple quadratic penalty $d_o^V\left(\hat{\mathbf{V}}, \mathbf{V}\right) = \left\|\hat{\mathbf{V}} - \mathbf{V}\right\|_2^2$ and the smoothness term $d_s^V(\cdot)$ is a global function of a weighted Laplacian on the estimated mesh defined as $\|\mathbf{L}\mathbf{V}\|_2^2$. Our weighting in the global Laplacian operator is both novel and important as it leads to significantly improved denoising performance. In addition to such weighting, we also introduce a novel face fairness term $d_f^V(\cdot)$ that automatically results in fairness of the triangle shapes. Such an approach to fairness significantly reduces the possibility of folding artifacts in the denoised mesh which occurs in many other methods.

3.2.1 Construction of our global Laplacian

Our Laplacian operator is anisotropic in nature [18] which is defined only along the normal directions at the vertices. However the novelty of our approach lies in the careful selection of the weights. Specifically we use the following operator $\mathbf{L}_i(\cdot)$ to define the Laplacian form for each vertex $\mathbf{v}_i$ *i.e.*

$$\mathbf{L}_i(\mathbf{v}_i) = \sum_{j \in \mathcal{N}_V(i)} \left(\frac{a_j b_j}{(1+b_j)\sum_{j \in \mathcal{N}_V(i)} a_j} \right) \left(\mathbf{n}_j^F \mathbf{n}_j^{F,T}\right) \left(\mathbf{v}_i - \frac{(\mathbf{v}_{j_1} + \mathbf{v}_{j_2} + \mathbf{v}_{j_3})}{3}\right) \quad (5)$$

where a_j and b_j form the bilateral weighting functions and $\mathbf{n}_j^F$ is the normal of the neighbouring face $\mathbf{f}_j$ corresponding to the vertex $\mathbf{v}_i$ *i.e.* $\mathcal{N}_V(i)$ is the set of 1-ring neighbouring faces to the vertex $\mathbf{v}_i$ and $\mathbf{v}_{j_1}$, $\mathbf{v}_{j_2}$, $\mathbf{v}_{j_3}$ are vertices of $\mathbf{f}_j$. We denote $\Delta\mathbf{v}_{ij}$ as the difference between the vertex $\mathbf{v}_i$ and the centroid of face $\mathbf{f}_j$, *i.e.* $\Delta\mathbf{v}_{ij} = \frac{\mathbf{v}_{j_1}+\mathbf{v}_{j_2}+\mathbf{v}_{j_3}}{3} - \mathbf{v}_i$. Consequently, for our approach we define

$$a_j = \exp\left(-\frac{\left(\mathbf{n}_j^T \Delta\mathbf{v}_{ij}\right)^2}{2\sigma_1^2}\right) \quad (6)$$

which is the weighting along the normal direction of the considered neighbouring face corresponding to the vertex $\mathbf{v}_j$. Similarly, we have

$$b_j = \exp\left(-\frac{\|\Delta\mathbf{v}_{ij}\|_2^2}{2\sigma_2^2}\right). \quad (7)$$

Finally, each Laplacian $\mathbf{L}_i(\mathbf{v}_i)$ is a function pertaining to a single vertex and all such terms are concatenated to form our global weighted Laplacian $\mathbf{L}(\mathbf{V})$ used in Eqn. 4. We note here that unlike many previous methods where the bilateral scheme is used, we have decoupled the two weights a_j and b_j and we have also used the face normals instead of vertex normals. We use the face normals since the vertex normals are undefined at edges and corners. As a result, our method provides better feature preservation at the edges and corners.

3.2.2 Face fairness penalty

As discussed in previous sections, while most denoising method apply vertex correction only along the surface normal, the actual noise present in a mesh also has a component that lies in the tangent plane about a vertex. Neglecting this

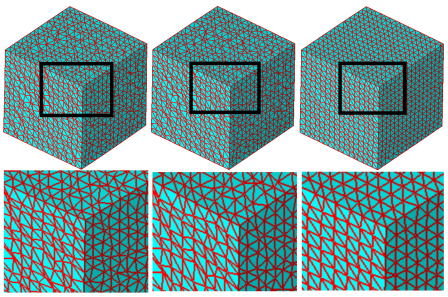

Figure 3. Face quality of denoised mesh of a cube ($N_V = 1538, N_F = 3072$) corrupted with isotropic Gaussian noise with standard deviation $\sigma = 0.15 \times$ mean edge length. Left image: Sun *et al.* [15]. Middle image: Our method without fairness penalty. Right image: Our method with fairness penalty. Our method that incorporates the fairness penalty is able to ensure face fairness whereas the other two methods fail to do so.

Error metric	Sun [15]	Ours (w/o fairness term d_f^V)	Ours (with fairness term d_f^V)
Mean NE (°)	0.6427	0.5704	0.4633
Mean VPE	0.0259	0.0258	0.0129

Table 1. Comparison of errors in the denoised output for different methods on the cube example given in Fig. 3 NE denotes 'Normal angle error' and VPE denotes 'Vertex position Euclidean distance error'.

fact under high noise levels leads to an undesirable folding of faces. To mitigate this problem, in our approach we introduce a face fairness term d_f^V that ensures that triangular faces do not become skinny or folded. The face fairness penalty for a single denoised vertex $\hat{\mathbf{v}}_i$, is

$$d_f^V(\hat{\mathbf{v}}_i) = \left\| r_i(\mathbf{I} - \mathbf{n}_i^V \mathbf{n}_i^{V,T})(\hat{\mathbf{v}}_i - \mathbf{v}_{c,i}) \right\|_2^2 \quad (8)$$

where $\mathbf{v}_{c,i}$ is the centroid of the 1-ring face neighbourhood $\mathcal{N}_V(i)$ around the vertex $\mathbf{v}_i$ and the weight r_i is given as

$$r_i = \begin{cases} 0 & \text{if } \mathbf{v}_i \in \mathbf{V}^B \\ 0 & \text{else if } \beta < 0 \\ \beta & \text{otherwise} \end{cases} \quad (9)$$

where $\beta = \min_{p,q \in \mathcal{N}_V(i)} (\mathbf{n}_p^{F,T} \mathbf{n}_q^F - \delta)$ and δ is a small positive value ($\delta = 0.2$ throughout this paper). Note the first condition in the weight defined in Eqn. 9 ignores the boundary in open meshes whereas the second condition carefully ignores the edges and corners. Also, the fairness penalty constrains the solution only in the tangential plane about a vertex without affecting the smoothness penalty term. Since, this term is explicitly defined with respect to the solution, our fairness penalty avoids tangential shift. Hence, the denoised mesh becomes free of irregular faces.

We note here that since our cost function is quadratic, we have a closed form solution for the vertex correction step given as

$$\hat{\mathbf{V}} = \left(\mathbf{I} + \lambda_V \mathbf{L}^T \mathbf{L} + \eta \mathbf{K}^T \mathbf{K}\right)^{-1} \left(\mathbf{V} + \eta \mathbf{K}^T \mathbf{K} \mathbf{V}_c\right) \quad (10)$$

where $\mathbf{K}$ is formed from Eqn. 8 and $\mathbf{V}_c$ is the concatenated vector formed from $\mathbf{v}_{c,i}, i = 1, 2, \cdots, N_V$. Since the cost function of Eqn. 4 is sparse in nature, in practice we can solve for the denoised mesh using efficient sparse solvers.

Our Algorithm: We summarise our approach to 3D mesh denoising as normal mollification followed by vertex correction which involve the minimisation of Eqn. 2 and Eqn. 4 respectively.

3.3. Salient Properties of Our Method

We can now present some of the salient features of our denoising algorithm and also compare its properties with that of other methods in the literature. Firstly, the significance of our fairness penalty is illustrated in Fig. 3 where we compare the denoised mesh faces of the method of Sun *et al.* [15] with our method where we solve Eqn. 4 both with and without the fairness penalty term d_f^V. We can observe that the solutions of Sun *et al.* [15] as well as our global method without the fairness penalty do not preserve the regular nature of the mesh faces. However, as is clearly evident, incorporating our fairness penalty term rectifies this problem and results in an accurate recovery of the mesh while also preserving the fairness of faces. In Table 1 we quantify the relative performance of the different methods in terms of the corresponding mean absolute errors of denoised normals in degrees and mean Euclidean error distances of the denoised vertices with respect to the ground truth. As can be seen, not only does the face fairness penalty improve the shapes of the denoised faces, it also improves the estimation of the vertex positions and the face normals.

The second salient feature of our approach is its ability to preserve surface feature without causing any volume shrinkage. In our experiment we used a spherical mesh ($N_V = 962, N_F = 1920$) corrupted with isotropic Gaussian noise with standard deviation $\sigma = 0.2 \times$ mean edge length. To compare different methods, we computed the normalised ratio of the volume of the denoised mesh and the original one. While the optimal ratio should be 1, the normalised volume ratios of the bilateral filtered method [6], Jones *et al.* [11], Sun *et al.* [15] and our own method were 0.7770, 0.9425, 1.0001 and 1.0011 respectively. From these ratios, it will be noted that the outputs of our method and Sun *et al.* [15] have insignificant volume shrinkage while the outputs of bilateral filtering [6]

Method	Feature Preservation	Volume Preservation	Face Fairness	Optimal Convergence
Laplacian [5]	Poor	Poor	Good	No
Taubin [17]	Poor	Good	Good	No
Bilateral [6]	Moderate	Moderate	Poor	No
Jones *et al.* [11]	Good	Moderate	No	NA
Sun *et al.* [15]	Good	Good	Poor	No
Zheng *et al.* [21]	Good	Moderate	Poor	No
Ours	Good	Good	Good	Yes

Table 2. A qualitative comparison of different mesh denoising algorithms. 'NA' denotes 'Not Applicable'. See text for details.

and Jones *et al.* [11] have a considerable shrinkage artefact. Hence, our method is able to both preserve volume as well as denoise in the tangential directions.

Finally, in Table 2 we provide a comparative assessment of some relevant denoising methods in the literature and our approach with regard to a variety of desirable properties. As will be noted, since the Laplacian method [5] and that of [17] are isotropic methods, triangle fairness is implicitly incorporated, but they perform poorly in preserving surface features. On the other hand, although Jones *et al.* [11], Sun *et al.* [15] and the BNF method of [21] have good feature preservation, they lack the fairness of the triangle shapes in their outputs. In contrast, apart from its ability to preserve desirable 3D surface features, our method guarantees both triangle shape fairing as well as optimal convergence unlike other methods. Thus our method has all the necessary attributes of a good mesh denoising scheme.

4. Results

We first compare the denoising performance of our algorithm with some of the relevant methods in the literature. Specifically, we compared our results with the bilateral filter [6], the method described in [11][1] and the method in [15]. In our implementation, we fixed the threshold t in Eqn. 3 to $t = 0.5$ for denoising objects with piecewise flat surfaces and $t = -0.25$ for denoising Kinect scans. It should be noted that the value of $t = -0.25$ is chosen as it is suitable for Kinect scans that have high amounts of quantisation noise leading to a staircase effect. All the parameter settings for different methods are chosen to provide the best performance for the respective methods.

4.1. Quantitative Evaluation of Performance

We first demonstrate the efficacy of our method on meshes corrupted with additive Gaussian noise of different levels to the vertex positions as compared to some of the other relevant methods. For our evaluation, we use both the mean and median absolute deviation of the face normals measured in degrees and the mean and median Euclidean error metric on the vertex positions. We manually tune the parameters of each of the methods we used for best performance with respect to the mean face normal deviations. The comparisons are tabulated in Table 3[2]. As can be seen from Table 3, in all the cases, our method has the lowest vertex position errors. This success of our method is mainly due to the addition of the fairness penalty for the face shapes which actually denoises the vertex positions on the surface along the tangential directions. Additionally, we can easily see that in most of the cases, the normal estimation is superior whereas our approach provides the lowest vertex position error for all datasets. In Fig. 4 the first row shows the surface quality of the results obtained from different methods applied on the Bunny scan from the Stanford repository ($N_V = 15861, N_F = 31001$) corrupted with isotropic Gaussian noise with standard deviation $\sigma = 0.2 \times$ mean edge length. The second row shows the same results rendered in the smooth shading mode in OpenGL. This rendering mode reveals the presence of folded faces as regions of black spots. As can be seen, our result does not contain such artefacts, while the results of Jones *et al.* [11] and Sun *et al.* [15] have numerous such black regions. However, although, bilateral filtering does not have such artefacts, its output has a high error with respect to the ground truth as listed in Table 3. In summary, it is clear that adding our face fairness penalty not only results in a fair denoised mesh, but also leads to better recovery of the vertex positions.

4.2. Evaluation on Real Datasets

We now present some visual comparisons on real datasets. Fig. 5 shows the comparison of denoising results on a raw Microsoft Kinect scan ($N_V = 46815, N_F = 91392$) of a person. The first column shows the snapshots of the full denoised mesh. The second column shows the zoomed-in view near the nose of the person. It will be noted here that the denoised surface of the nose in our case is far better recovered than those of the bilateral method [6] and that of Sun *et al.* [15]. While a folded triangle is observed in the case of the bilateral filtered nose, the jaggedness of the side of the nose is prominent in the case of Sun *et. al.* Both of these observations are explained by our use of an additional triangle shape fairness penalty term. Since the side of the nose has normals almost orthogonal to the viewpoint, these two methods fail to denoise the actual noise component

[1] We used the implementation of Trimesh 2 available at http://gfx.cs.princeton.edu/proj/trimesh2/.

[2] We could not evaluate the errors on the vertex positions for BNF [21] since the code available at http://www.youyizheng.net/ performs an artificial scaling.

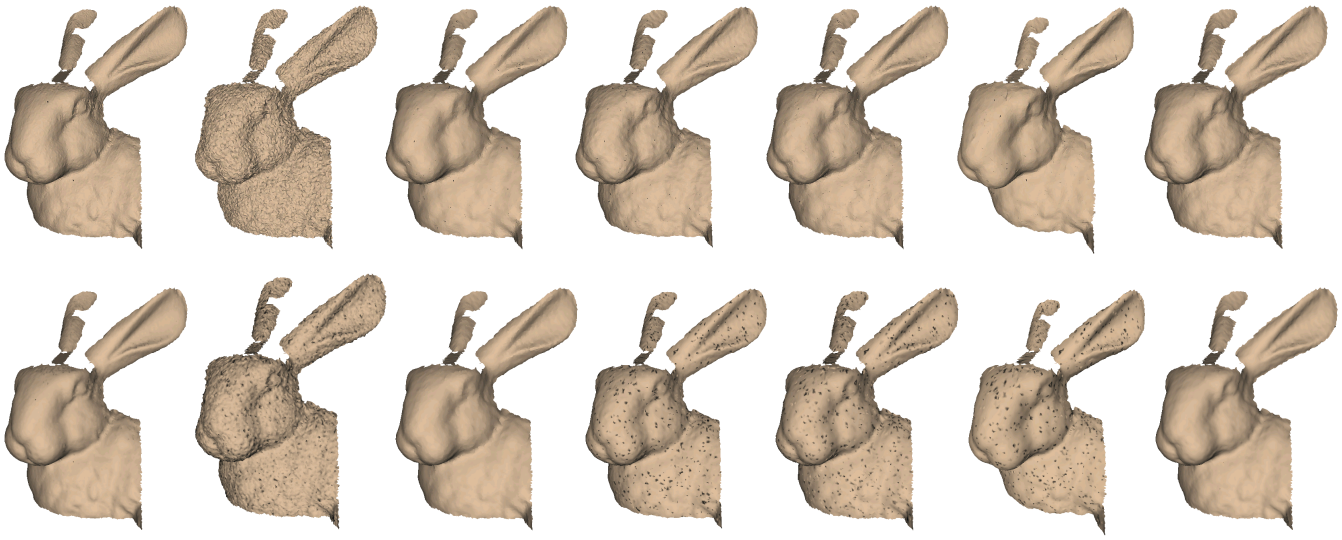

Figure 4. Denoised mesh quality of different methods on the bunny face ($N_V = 15861, N_F = 31001$) corrupted with isotropic Gaussian noise with standard deviation $\sigma = 0.2 \times$ mean edge length. The columns correspond to the ground truth, noisy mesh and solutions for bilateral filtered [6], Jones *et al.* [11], Sun *et al.* [15], BNF [21] and our method respectively. The first row shows the surface quality. The second row shows the same surface in a smooth shading rendering mode. In this mode, the folded face artefacts appear as black spots for many of the methods but not for our approach.

Object	Error metric	Noisy	Bilateral [6]	Jones [11]	Sun [15]	BNF [21]	Ours
Cube ($N_V = 1538, N_F = 3072$) $\sigma = 0.15 \times$ mean edge length	Mean NE (°)	17.8359	15.8377	8.0962	0.6427	1.0038	**0.4633**
	Median NE (°)	15.7464	9.9139	3.9450	0.4571	0.6924	**0.2496**
	Mean VPE	0.0339	0.0595	0.0359	0.0259	NA	**0.0129**
	Median VPE	0.0323	0.0467	0.0309	0.0236	NA	**0.0113**
Sphere ($N_V = 962, N_F = 1920$) $\sigma = 0.20 \times$ mean edge length	Mean NE (°)	25.4718	9.5720	6.7752	5.6170	4.5514	**3.0540**
	Median NE (°)	20.4477	4.0244	2.9805	2.1998	**0.9505**	1.8509
	Mean VPE	0.0404	0.0785	0.0383	0.0323	NA	**0.0169**
	Median VPE	0.0377	0.0802	0.0353	0.0290	NA	**0.0148**
Fandisk ($N_V = 6475, N_F = 12946$) $\sigma = 0.15 \times$ mean edge length	Mean NE (°)	17.4080	14.9835	16.9082	2.8616	**2.1682**	4.7996
	Median NE (°)	15.4042	8.8645	13.7399	**1.1175**	1.6097	1.5808
	Mean VPE	0.0098	0.0162	0.0382	0.0090	NA	**0.0086**
	Median VPE	0.0094	0.0130	0.0364	0.0080	NA	**0.0062**
Bunny face ($N_V = 15861, N_F = 31001$) $\sigma = 0.20 \times$ mean edge length	MeanNE (°)	29.8166	13.6589	14.0275	13.6516	13.3182	**8.0335**
	Median NE (°)	23.0867	6.0132	6.6298	6.4421	6.1740	**5.8553**
	Mean VPE	0.0340	0.0300	0.0304	0.0287	NA	**0.0252**
	Median VPE	0.0319	0.0277	0.0273	0.0263	NA	**0.0226**
Armadillo ($N_V = 67598, N_F = 134576$) $\sigma = 0.20 \times$ mean edge length	Mean NE (°)	24.4805	9.0491	**8.0203**	8.2836	8.2653	9.0906
	Median NE (°)	20.9570	6.3233	**5.7928**	5.8005	5.8508	7.4757
	MeanVPE	0.1669	0.1541	0.1435	0.1408	NA	**0.1235**
	MedianVPE	0.1598	0.1459	0.1344	0.1321	NA	**0.1147**

Table 3. Comparison of denoising performance of our approach with other methods in the literature. We compare both normal angle error (NE) and Vertex position Euclidean distance (VPE) error. 'NA' denotes 'Not Available'. The best performance for each dataset is indicated in bold. See text for details.

along the direction of the viewpoint which is tangential to the surface normal. However our method is able to handle this tangential noise. Moreover, the quality of the face is recovered very well as shown in the third column. Fig. 6 shows the denoising results on a raw Kinect scan of a chair ($N_V = 31159, N_F = 59979$). The bilateral filter and the method of Sun *et al.* [15] failed to denoise the chair. The reason for failure is the staircase artefact caused by the quantisation noise present in the Kinect scan. This makes most of the face normals almost tangential to the actual direction of noise which hinders the diffusion in these two methods. However since our method is a global one and has an implicit fairing in the tangential planes, it is not affected and successfully recovers the surface. Finally, Fig. 7 shows the denoising results on a mesh ($N_V = 185546, N_F = 360814$) of a sculptured pillar from the Vitthala temple at Hampi complex, a heritage cultural site. This mesh was generated using a standard

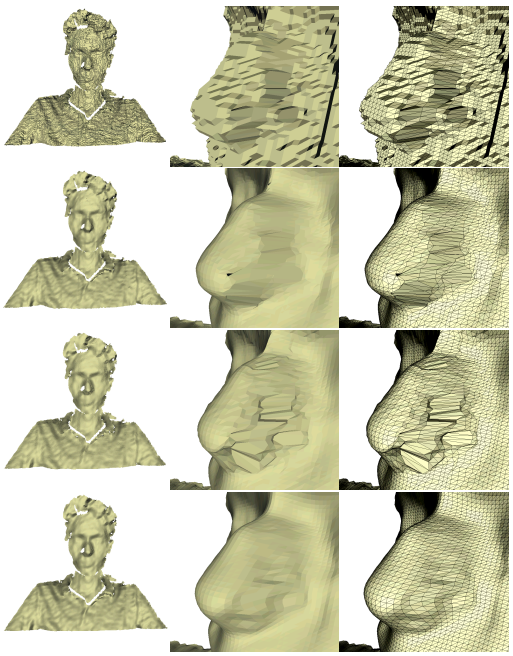

Figure 5. Denoising quality of different methods on a mesh ($N_V = 46815, N_F = 91392$) generated from a raw Kinect depth map of a person. The rows correspond to the raw noisy mesh, bilateral filtered [6], Sun *et al.* [15] and our method respectively. While the first column shows the surface quality, the second and third columns show the zoomed-in views on the nose of the person. These representations of the nose clearly show the improved face fairness quality of our method.

multiview stereo package on a set of RGB images of the pillar [7]. The second row shows the zoomed-in views (in smooth shading rendering mode) of the region marked as a black square. Since the noise level of the multiview stereo result is low, the smoothness of the outputs of the different methods are comparable. However, amongst the methods compared, ours has the lowest number of folded mesh faces which is due to our explicit incorporation of a fairness penalty term.

5. Conclusion

We have presented a two-step denoising method that globally solves for both normal mollification and vertex correction. Our vertex correction step accounts for noise in all directions and also incorporates a novel cost function for enforcing face fairness in the mesh. The ability of our method to provide good mesh denoising while preserving face fairness is demonstrated on a number of datasets. The superiority of our approach over other relevant methods in the literature is also established.

Acknowledgement We would like to thank the anonymous reviewers for useful feedback and Suvam Patra of Graphics and Computer Vision Lab, IIT Delhi for providing us the sculptural pillar dataset.

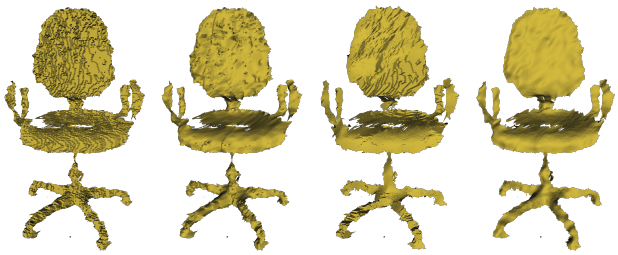

Figure 6. Denoising quality of different methods on a mesh ($N_V = 31159, N_F = 59979$) generated from a raw Kinect depth map of a chair. From left to right we show the noisy mesh, and results of bilateral filtered [6], Sun *et al.* [15] and our method respectively. As can be seen, the method of Sun *et al.* fails to denoise this dataset.

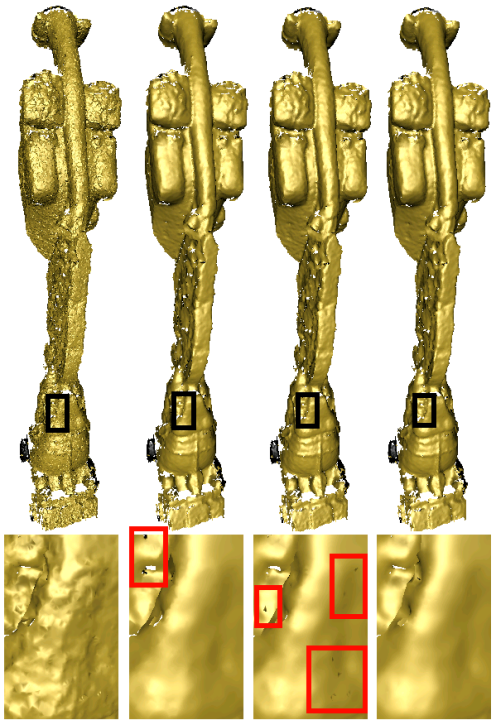

Figure 7. Denoising quality of different methods on a mesh ($N_V = 185546, N_F = 360814$) of a sculptural pillar generated using multiview stereo applied to a set of images. We show, from left to right, the noisy mesh, and results of bilateral filtered [6], Sun *et al.* [15] and our method respectively. The first row shows the surface quality. The second row shows the zoomed-in views with marked regions that show the presence of folded faces visible as black spots in smooth shading rendering mode. Such artefacts are not present in our denoised mesh.

References

[1] X. Cheng, M. Zeng, and X. Liu. Feature-preserving filtering with l0 gradient minimization. Computers & Graphics, 38:150–157, 2014. 1

[2] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999. 1

[3] A. El Ouafdi and D. Ziou. A global physical method for manifold smoothing. In Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on, pages 11–17, June 2008. 2

[4] H. Fan, Y. Yu, and Q. Peng. Robust feature-preserving mesh denoising based on consistent subneighborhoods. Visualization and Computer Graphics, IEEE Transactions on, 16(2):312–324, 2010. 1

[5] D. A. Field. Laplacian smoothing and delaunay triangulations. Communications in applied numerical methods, 4(6):709–712, 1988. 1, 2, 6

[6] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. ACM Transactions on Graphics (TOG), 22(3):950–953, 2003. 1, 2, 3, 5, 6, 7, 8

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. IEEE Trans. on Pattern Analysis and Machine Intelligence, 32(8):1362–1376, 2010. 8

[8] L. He and S. Schaefer. Mesh denoising via l0 minimization. ACM Transactions on Graphics (TOG), 32(4):64, 2013. 1, 2

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 19–26. ACM, 1993. 1

[10] Z. Ji, L. Liu, and G. Wang. A global laplacian smoothing approach with feature preservation. In Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics, pages 269–274. IEEE, IEEE Computer Society, 2005. 1

[11] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. ACM Transactions on Graphics, 22(3):943–949, July 2003. 1, 2, 3, 5, 6, 7

[12] L. Liu, C.-L. Tai, Z. Ji, and G. Wang. Non-iterative approach for global mesh optimization. Computer-Aided Design, 39(9):772–782, 2007. 1

[13] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian mesh optimization. In Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, pages 381–389. ACM, 2006. 1

[14] Y. Ohtake, A. Belyaev, and H. Seidel. Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals. In Vision Modeling and Visualization, 2002. 1

[15] X. Sun, P. Rosin, R. Martin, and F. Langbein. Fast and effective feature-preserving mesh denoising. Visualization and Computer Graphics, IEEE Transactions on, 13(5):925–938, Sept. 2007. 1, 2, 3, 4, 5, 6, 7, 8

[16] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein. Random walks for feature-preserving mesh denoising. Computer Aided Geometric Design, 25(7):437–456, 2008. Solid and Physical Modeling Selected papers from the Solid and Physical Modeling and Applications Symposium 2007 (SPM 2007) Solid and Physical Modeling and Applications Symposium 2007. 1

[17] G. Taubin. A signal processing approach to fair surface design. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95, pages 351–358, New York, NY, USA, 1995. ACM. 2, 6

[18] G. Taubin. Linear anisotropic mesh filtering. Res. Rep. RC2213 IBM, 2001. 1, 4

[19] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen. Decoupling noise and features via weighted l-1-analysis compressed sensing. ACM Trans. Graph., 33(2):18:1–18:12, Apr. 2014. 1

[20] H. Zhang, C. Wu, J. Zhang, and J. Deng. Variational mesh denoising using total variation and piecewise constant function space. Visualization and Computer Graphics, IEEE Transactions on, 21(7):873–886, July 2015. 1

[21] Y. Zheng, H. Fu, O.-C. Au, and C.-L. Tai. Bilateral normal filtering for mesh denoising. Visualization and Computer Graphics, IEEE Transactions on, 17(10):1521–1530, Oct. 2011. 1, 6, 7