

Article

Fast Feature-Preserving Approach to Carpal Bone Surface Denoising

Ibrahim Salim and A. Ben Hamza *

Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada; ibrahim_salim1976@yahoo.com

* Correspondence: hamza@ciise.concordia.ca

Received: 13 June 2018; Accepted: 19 July 2018; Published: 21 July 2018



Abstract: We present a geometric framework for surface denoising using graph signal processing, which is an emerging field that aims to develop new tools for processing and analyzing graph-structured data. The proposed approach is formulated as a constrained optimization problem whose objective function consists of a fidelity term specified by a noise model and a regularization term associated with prior data. Both terms are weighted by a normalized mesh Laplacian, which is defined in terms of a data-adaptive kernel similarity matrix in conjunction with matrix balancing. Minimizing the objective function reduces it to iteratively solve a sparse system of linear equations via the conjugate gradient method. Extensive experiments on noisy carpal bone surfaces demonstrate the effectiveness of our approach in comparison with existing methods. We perform both qualitative and quantitative comparisons using various evaluation metrics.

Keywords: mesh denoising; kernel similarity; matrix balancing; carpal bones

1. Introduction

Recent advances in 3D scanning technology have led to the increasing use of 3D models in many fields, including the entertainment industry, archaeology, computer vision, and medical imaging. These models are usually captured in the form of point clouds or polygonal meshes [1], but they are often corrupted by noise during the data acquisition stage. The main problem with 3D shape denoising is how we can distinguish between noise and features, especially sharp surface features. To ensure the development of high-quality 3D shapes for use in downstream applications, it is important to develop effective surface denoising techniques to remove inevitable noise in the measurements [2–8].

In recent years, a plethora of techniques have been proposed to tackle the 3D surface denoising problem. Generally, surface denoising methods can be classified into two major categories: isotropic and anisotropic. The former techniques filter the noisy data independently of direction, while the latter methods modify the diffusion equation to make it nonlinear or anisotropic in order to preserve the sharp features of a 3D mesh surface. The simplest surface denoising method is the Laplacian flow which repeatedly and simultaneously adjusts the location of each mesh vertex to the geometric center of its neighboring vertices [2].

Most surface denoising methods are adopted from the image processing literature [9–12], including the use of mean, median, and bilateral filters. In particular, bilateral filtering has been used extensively in image processing applications, due, in large part, to its good performance in smoothing noisy images while preserving edges. The bilateral filter takes into account the variation in image intensity by replacing the intensity value at a pixel by a weighted average of the intensity values from neighboring pixels. Although these filters have been successfully applied to image denoising, it is, however, not straightforward to apply them directly to graph-structured data. Fleishman et al. [5] proposed a bilateral mesh denoising approach that filters each mesh vertex in the normal direction

using local neighborhoods. Zheng et al. [8] applied the bilateral normal filter in a local iterative and a global non-iterative scheme for anisotropic denoising. Sun et al. [13] introduced a two-step mesh denoising framework. In the first step, the noisy face normals are filtered iteratively by weighted averaging of neighboring face normals. In the second step, the mesh vertex positions are iteratively updated based on the denoised face normals. Huang and Uscher proposed a multiscale anisotropic Laplacian (MSAL) model [14], which employs the anisotropic Laplacian operator combined with a roughness scale and yields significantly better results than the anisotropic Laplacian model and the bilateral filter. Ouafdi et al. [15] introduced a probabilistic mesh denoising method by performing anisotropic averaging of neighboring vertices weighted by a Riemannian metric. Zhang et al. [16] presented a guided mesh normal filtering framework by constructing the guidance for joint bilateral filtering of geometry signals using a two-step process. Joint bilateral filtering is applied to the face normals, followed by updating the mesh vertices to agree with the denoised face normals. More recently, Yadav et al. [17] proposed a two-stage mesh denoising approach using robust statistics. In the first stage, the face normals are filtered via bilateral normal filtering using Tukey's bi-weight as a similarity function. In the second stage, the mesh vertex positions are updated using edge-to-face normal orthogonality constraints along with differential coordinates.

On the other hand, image/surface denoising via graph signal processing techniques has received considerable attention in recent years [12,18,19]. A graph-based approach to image denoising and deblurring was introduced in [12] using a data-adaptive objective function derived from a normalized graph Laplacian. Chung et al. [19] used the graph Laplacian to construct the discrete version of heat kernel smoothing on graph-structured data obtained by binary segmentation of the computed tomography of human lung data. Also, Chung et al. [20] introduced a heat kernel regression approach to surface smoothing using the Laplace–Beltrami eigenfunctions which are obtained by solving a generalized eigenvalue problem. Such an approach can, however, be prohibitively expensive, especially when the problem size is large (i.e., large matrices). Another issue with spectral approaches is how to select the appropriate number of eigenvalues and associated eigenfunctions to be retained.

Motivated by the good performance of the similarity-based image denoising framework proposed in reference [12], we introduce a simple, yet effective, feature-preserving approach to 3D mesh denoising. The proposed method employs a normalized mesh Laplacian, which is defined in terms of a data-adaptive kernel similarity matrix in conjunction with matrix balancing. We formulate our surface denoising framework as a constrained minimization problem, which can be solved efficiently using the conjugate gradient (CG) method. Our approach can remove noise effectively while preserving the nonlinear features of surfaces, such as curved surface regions, sharp edges, and fine details. While our proposed framework is general enough to be applied to any problem involving surface denoising, the primary focus of this work is on noise removal from carpal bone surfaces. Further, recovering high quality surfaces from noisy carpal bone surfaces is a fundamental problem in computational anatomy and biomechanics and is of paramount importance to the diagnosis of wrist pathologies, such as arthritis. Our main contributions may be summarized as follows:

- We introduce a mesh denoising approach using a data-adaptive kernel similarity matrix in conjunction with matrix balancing.
- We formulate the proposed framework as a constrained minimization problem and solve it iteratively using the conjugate gradient method.
- Our experimental results show superior performance of the proposed framework over existing mesh denoising methods.

The rest of this paper is organized as follows. In Section 2, we briefly recall some basic concepts of geometry processing, followed by a general formulation of the surface denoising problem in the graph signal processing setting. In Section 3, we present the main building blocks of our method, and discuss, in detail, the algorithmic steps. In Section 4, we present experimental results to demonstrate the competitive performance of our denoising approach on carpal bone surfaces. Finally, Section 5 concludes the paper and points out future work directions.

2. Problem Formulation

Triangular mesh representation: A 3D shape is usually modeled as a triangle mesh, $\mathbb{M}$, whose vertices are sampled from a Riemannian manifold. A triangle mesh, $\mathbb{M}$, may be defined as a graph, $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ or $\mathbb{G} = (\mathcal{V}, \mathcal{T})$, where $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ is the set of vertices, $\mathcal{E} = \{\mathbf{e}_{ij}\}$ is the set of edges, and $\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_m\}$ is the set of triangles. Each edge, $\mathbf{e}_{ij} = [\mathbf{v}_i, \mathbf{v}_j]$, connects a pair of vertices, $\{\mathbf{v}_i, \mathbf{v}_j\}$. Two distinct vertices, $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}$, are adjacent (denoted by $\mathbf{v}_i \sim \mathbf{v}_j$ or simply $i \sim j$) if they are connected by an edge, i.e., $\mathbf{e}_{ij} \in \mathcal{E}$. The neighborhood of a vertex, $\mathbf{v}_i$, is the set $\hat{\mathbf{v}}_i = \{\mathbf{v}_j \in \mathcal{V} : \mathbf{v}_j \sim \mathbf{v}_i\}$.

Laplacian matrix of a weighted graph: The graph, $\mathbb{G}$, may be equipped with a nonnegative weight function, $\omega : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$, such that

$$\omega_{ij} = \begin{cases} \omega_{ii} & \text{if } i = j \\ \omega_{ij} & \text{if } i \sim j \\ 0 & \text{o.w.} \end{cases} \quad (1)$$

The Laplacian matrix, $\mathbf{L} = (\ell_{ij})$, of a weighted graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, whose elements are given by

$$\ell_{ij} = \begin{cases} d_i - \omega_{ii} & \text{if } i = j \\ -\omega_{ij} & \text{if } i \sim j \\ 0 & \text{o.w.} \end{cases} \quad (2)$$

where $\mathbf{A} = (w_{ij})$ is the weighted adjacency matrix, and $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$ is the degree matrix with $d_i = \sum_{j \sim i} \omega_{ij}$ being the degree of vertex i . The normalized weighted Laplacian matrix, $\mathcal{L}$, is defined as

$$\mathcal{L} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}. \quad (3)$$

Figure 1 displays a 3D hand model and its weighted Laplacian matrix, with weights $\omega_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|$, where $\|\cdot\|$ denotes the Euclidean norm. The sparsity pattern (or support) of $\mathbf{L} = (\ell_{ij})$ is the set of indices, ij , with $\ell_{ij} \neq 0$.

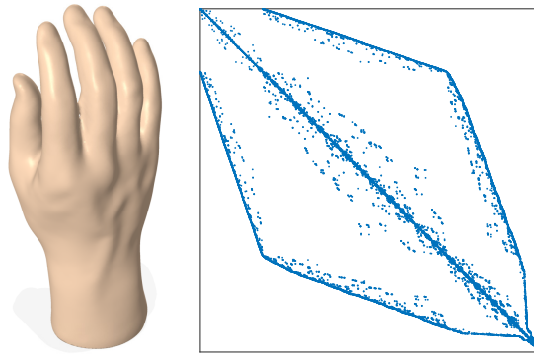


Figure 1. Hand model (left) and sparsity pattern plot of its weighted Laplacian matrix (right).

The Laplacian matrix may be viewed as an operator defined on the space of graph signals, $u : \mathcal{V} \rightarrow \mathbb{R}$, as follows:

$$\mathbf{L}u(i) = \sum_{i \sim j} \omega_{ij}(u(i) - u(j)), \quad \text{for all } i \in \mathcal{V}. \quad (4)$$

In other words, $\mathbf{L}u(i)$ is the sum of the weighted differences between the value of the graph signal, u , at vertex i and the values at the neighboring vertices.

Since $|\mathcal{V}| = n$, we may represent any graph signal, $u : \mathcal{V} \rightarrow \mathbb{R}$, as a column vector, $\mathbf{u} = (u(i)) \in \mathbb{R}^n$, with the i th element, $u(i)$. Thus, the quadratic form of the signal, $\mathbf{u}$, with respect to the Laplacian matrix can be expressed as

$$\mathbf{u}^T \mathbf{L} \mathbf{u} = \sum_{i \sim j} \omega_{ij} (u(i) - u(j))^2, \quad (5)$$

which shows that if the weights are symmetric, then the Laplacian matrix is symmetric positive semi-definite. So the action of the Laplacian on a signal may be viewed as measuring the smoothness of that signal across the edges in the mesh.

Mesh Denoising Model

In all real applications, measurements are usually perturbed by noise. In the course of acquiring, transmitting or processing a 3D model, for example, the noise-induced degradation often yields a resulting graph signal observation model, and the most commonly used is the additive one,

$$\mathbf{v} = \mathbf{u} + \boldsymbol{\eta}, \quad (6)$$

where the observed graph signal, $\mathbf{v}$, includes the original graph signal, $\mathbf{u}$, and the random noise process, $\boldsymbol{\eta}$, which is usually assumed to be Gaussian with zero mean and standard deviation σ .

Surface denoising refers to the process of recovering a 3D model contaminated by noise. The challenge of the problem of interest lies in recovering the graph signal, $\mathbf{u}$, from the observed signal $\mathbf{v}$, and furthering the estimation by making use of any prior knowledge/assumptions about the noise process $\boldsymbol{\eta}$.

When considering the noise model (6), our goal may be succinctly stated as one of estimating the underlying graph signal, $\mathbf{u}$, based on an observed signal, $\mathbf{v}$, and/or any potential knowledge of the noise statistics to further regularize the solution. This yields the following fidelity-constrained optimization problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & \mathcal{R}(\mathbf{u}) \\ \text{s.t.} \quad & \|\mathbf{v} - \mathbf{u}\|^2 \leq \sigma^2 \end{aligned} \quad (7)$$

where $\mathcal{R}$ is a given regularization functional, which often defines the particular emphasis on the features of the achievable solution. In other words, we want to find an optimal solution that yields the smallest value of the objective function among all solutions that satisfy the constraints. Using Lagrange's theorem, the minimizer of (7) is given by

$$\hat{\mathbf{u}} = \arg \min_{\mathbf{u}} \left\{ \|\mathbf{v} - \mathbf{u}\|^2 + \beta \mathcal{R}(\mathbf{u}) \right\}, \quad (8)$$

where β is a non-negative regularization parameter, which is often estimated or chosen *a priori*. A critical issue, however, is the choice of the regularization functional, $\mathcal{R}$, which is often driven by geometric arguments. A commonly used functional is the mesh Laplacian quadratic form defined as a (squared) weighted vector norm:

$$\mathcal{R}(\mathbf{u}) = \|\mathbf{u}\|_{\mathbf{L}}^2 = \mathbf{u}^T \mathbf{L} \mathbf{u}. \quad (9)$$

3. Methods

In this section, we present the main components of the proposed surface denoising framework and describe, in detail, its algorithmic steps. The flowchart of our approach is illustrated in Figure 2.

Kernel similarity: Using the Gaussian kernel, we define the kernel weight matrix, $\mathbf{S} = (s_{ij})$, as

$$s_{ij} = \exp \left(-\frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2}{2h^2} \right), \quad (10)$$

where $\mathbf{v}_i$ is the i th vertex of the noisy mesh, $\mathbf{v}_j$ are the neighboring vertices of $\mathbf{v}_i$, and h is the bandwidth

parameter of the Gaussian kernel. Each edge weight, s_{ij} , is a similarity measure whose value is large when i is closer to j . We define the kernel similarity weight matrix as follows:

$$\mathbf{K} = \frac{\mathbf{S} + \mathbf{S}^T}{2}, \quad (11)$$

which is a symmetric, non-negative matrix. Further, all of its off-diagonal elements are positive.



Figure 2. Flowchart of our proposed surface denoising method, where $\mathbf{v}$ is the noisy graph signal, and $\mathbf{u}^*$ is the estimated signal.

Sinkhorn matrix balancing: Applying the Sinkhorn matrix balancing procedure [21] to the kernel similarity weight matrix, $\mathbf{K}$, yields a symmetric non-negative doubly stochastic filtering matrix, $\mathbf{W}$, given by

$$\mathbf{W} = \mathbf{C}^{-1/2} \mathbf{K} \mathbf{C}^{-1/2}, \quad (12)$$

where $\mathbf{C}$ is a diagonal scaling matrix [22]. It should be noted that since the filtering matrix, $\mathbf{W}$, is doubly stochastic, its largest eigenvalue is equal to 1 with the associated eigenvector, $\mathbf{e} = \mathbf{1}/\sqrt{n}$, where $\mathbf{1}$ is a vector of all ones. In other words, the filtering matrix preserves the DC component of a graph signal (i.e., $\mathbf{W}\mathbf{e} = \mathbf{e}$).

Normalized mesh Laplacian: We define the normalized mesh Laplacian matrix as

$$\mathbf{L} = \mathbf{I} - \mathbf{W} = \mathbf{I} - \mathbf{C}^{-1/2} \mathbf{K} \mathbf{C}^{-1/2}, \quad (13)$$

which is symmetric positive semi-definite. The Laplacian matrix, $\mathbf{L}$, can be interpreted as a data-adaptive high-pass filter, enabling us to incorporate a variety of filters in the data term as well the regularization term.

From (13), it is easy to see that if λ is an eigenvalue of $\mathbf{W}$, then $1 - \lambda$ is an eigenvalue of $\mathbf{L}$. In particular, 0 is an eigenvalue of $\mathbf{L}$ with the associated eigenvector, $\mathbf{e}$. The eigenvalues of $\mathbf{L}$ may be viewed as graph frequencies. Moreover, the eigenvectors associated with the smallest eigenvalues have smooth oscillations and capture the large-scale properties of a shape well. As shown in Figure 3, the (non-trivial) eigenvectors of $\mathbf{L}$ encode important information about the global geometry of a shape. Notice that the eigenvectors associated with larger eigenvalues oscillate more rapidly. Blue regions indicate small eigenvector values and red regions indicate large values, while green and yellow regions are in between.

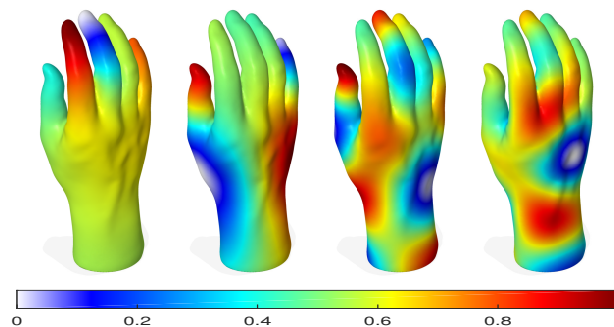


Figure 3. Visualization of four eigenvectors of the normalized mesh Laplacian matrix. From **left to right**: a 3D hand model Gouraud shaded and color-coded by the values of the second, eighth, fifteenth and twentieth eigenvectors.

3.1. Surface Denoising Approach

We formulated our surface denoising framework as a constrained optimization problem by minimizing the following cost function

$$\begin{aligned}\mathcal{C}(\mathbf{u}) &= \|\mathbf{v} - \mathbf{u}\|_{\mathbf{I} + \alpha\mathbf{L}}^2 + \beta\|\mathbf{u}\|_{\mathbf{L}}^2 \\ &= (\mathbf{v} - \mathbf{u})^\top (\mathbf{I} + \alpha\mathbf{L})(\mathbf{v} - \mathbf{u}) + \beta\mathbf{u}^\top \mathbf{L}\mathbf{u},\end{aligned}\quad (14)$$

where $\mathbf{v}$ is the noisy graph signal and $\mathbf{u}$ is the estimated signal. The non-negative parameters, α and β , are often estimated or chosen *a priori*. Note that the first term is a weighted error between the input and its estimate, and minimizing such an error yields a solution as close as possible to the input. Minimizing the second term, on the other hand, yields a smooth solution. Further, $\mathbf{I} + \alpha\mathbf{L}$ is a symmetric, positive-definite matrix.

The cost function, $\mathcal{C}(\mathbf{u})$, can be minimized by finding its gradient and setting it to zero

$$\nabla \mathcal{C}(\mathbf{u}) = -2(\mathbf{I} + \alpha\mathbf{L})(\mathbf{v} - \mathbf{u}) + 2\beta\mathbf{L}\mathbf{u} = \mathbf{0}, \quad (15)$$

resulting in the following system of linear equations:

$$(\mathbf{I} + (\alpha + \beta)\mathbf{L})\mathbf{u} = (\mathbf{I} + \alpha\mathbf{L})\mathbf{v}. \quad (16)$$

Since $\mathbf{I} + (\alpha + \beta)\mathbf{L}$ is a symmetric, positive-definite matrix, system (16) can be efficiently solved using iterative methods such as the CG method, which is a commonly used iterative algorithm for solving sparse systems of linear equations.

3.2. Algorithm

The objective of 3D mesh denoising is to remove noise while preserving features. Our proposed surface denoising approach consists of two major steps, as illustrated in Figure 2. In the first step, the normalized mesh Laplacian is computed using kernel similarity and matrix balancing. In the second step, a sparse system of linear equations is iteratively solved using the CG method. It should be noted that the proposed algorithm consists of both outer and inner iterations. The outer iterative process is used to compute the normalized mesh Laplacian, while the inner iterative process is employed to solve the constrained minimization problem. Algorithm 1 summarizes the main algorithmic steps of our approach.

Algorithm 1 Feature-Preserving Mesh Denoising

Input Noisy graph signal $\mathbf{v}$

- 1: $\hat{\mathbf{u}}^{(0)} = \mathbf{v}$
- 2: $k = 0$.
- 3: **while** not converged **do**
- 4: Compute the kernel similarity weight matrix $\mathbf{K}$ from $\hat{\mathbf{u}}^{(k)}$ using (10)–(11)
- 5: Apply Sinkhorn matrix balancing to $\mathbf{K}$ to get the diagonal matrix $\mathbf{C}$
- 6: Compute the Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{C}^{-1/2}\mathbf{K}\mathbf{C}^{-1/2}$
- 7: Solve the linear system in (16) using conjugate gradient to estimate $\hat{\mathbf{u}}^{(k+1)}$.
- 8: Set $\mathbf{u}^* = \hat{\mathbf{u}}^{(k+1)}$
- 9: $k = k + 1$
- 10: **end while**

return $\mathbf{u}^*$

Output Estimated signal $\mathbf{u}^*$

4. Experiments

In this section, through extensive experiments, we evaluate the performance of our proposed mesh denoising approach on carpal bone surfaces [23]. As shown in Figure 4, the carpal bones of the right wrist in a healthy male are the capitate, hamate, lunate, pisiform, scaphoid, trapezium, trapezoid, and triquetrum. Since the trapeziometacarpal joint of the thumb is a common site for osteoarthritis, the first metacarpal bone is also considered in our analysis. The forearm's radius and ulna bones, which support the many muscles that manipulate the bones of the hand and wrist, are also depicted in Figure 4.

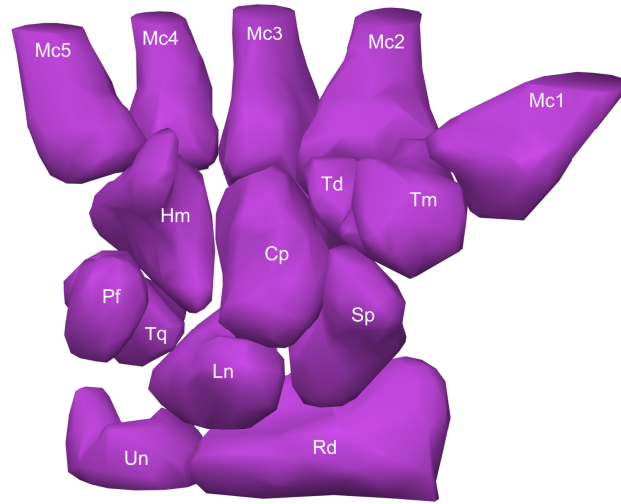


Figure 4. Carpal bone anatomy of a healthy male from a palmar view. The carpus consists of eight carpal bones which are arranged in proximal and distal rows. The proximal row contains the scaphoid (Sp), lunate (Ln), triquetrum (Tq), and pisiform (Pf), while the distal row contains the trapezium (Tm), trapezoid (Td), capitate (Cp), and hamate (Hm). The distal row adjoins the five metacarpals (Mc1-5) of the wrist. The radius (Rd) and ulna (Un) are also shown.

Implementation details: All experiments were performed on a desktop computer with an Intel Core 2 Duo running at 3.40 GHz and 16 GB RAM, and the proposed mesh denoising algorithm was implemented in MATLAB. The parameters, α and β , were chosen as the inverse of the minimum and maximum of the mesh degree values, respectively (i.e., $\alpha = 1/d_{min}$ and $\beta = 1/d_{max}$). The kernel bandwidth parameter, h , was estimated using the median absolute deviation (MAD) as follows:

$$h = 1.4826 \sum_{i=1}^n \text{MAD}(\mathbf{v}_i - \mathbf{v}_j), \quad (17)$$

where $\text{MAD}(\mathbf{x}) = \text{median}(\|\mathbf{x} - \text{median}(\|\mathbf{x}\|)\|)$ is a measure of spread that represents the expected absolute-error loss, and is robust to outliers.

Baseline methods: We compared the effectiveness of our proposed technique with several state-of-the-art approaches, including bilateral mesh denoising (BMD) [5], the multiscale anisotropic Laplacian (MSAL) method [14], guided mesh normal denoising (GMD) [16], and robust and high fidelity mesh denoising (RMD) [17].

4.1. Results

We performed extensive experiments on various carpal bone surfaces, including the right metacarpal, scaphoid, left metacarpal, left hamate, lunate, and pisiform, as shown in Figure 5.

We generated the noisy carpal bone models by setting the standard deviations of the noise to $0.5\bar{\ell}$ and $0.7\bar{\ell}$ of the mean edge length $\bar{\ell}$, as given by

$$\bar{\ell} = \frac{1}{|\mathcal{E}|} \sum_{\mathbf{e}_{ij} \in \mathcal{E}} \|\mathbf{e}_{ij}\|, \quad (18)$$

where $\|\mathbf{e}_{ij}\| = \|\mathbf{v}_i - \mathbf{v}_j\|$ if $i \sim j$, and $\|\mathbf{e}_{ij}\| = 0$ otherwise. More precisely, a vertex, $\mathbf{v}_i$, of a noisy mesh is given by the additive random noise model:

$$\mathbf{v}_i = \mathbf{u}_i + \sigma(\boldsymbol{\eta}_i \odot \mathbf{n}_i), \quad (19)$$

where $\boldsymbol{\eta}_i$ are i.i.d. Gaussian random vectors (i.e., $\boldsymbol{\eta}_i$ is a 3-dimensional vector containing pseudorandom values drawn from the standard normal distribution, $N(0, 1)$), $\mathbf{n}_i$ is the unit normal vector at the noise-free vertex, $\mathbf{u}_i$, and $\odot$ denotes the Hadamard product between two vectors (i.e., the elements of vector $\boldsymbol{\eta}_i \odot \mathbf{n}_i$ are obtained via element-by-element multiplication of vectors $\boldsymbol{\eta}_i$ and $\mathbf{n}_i$).

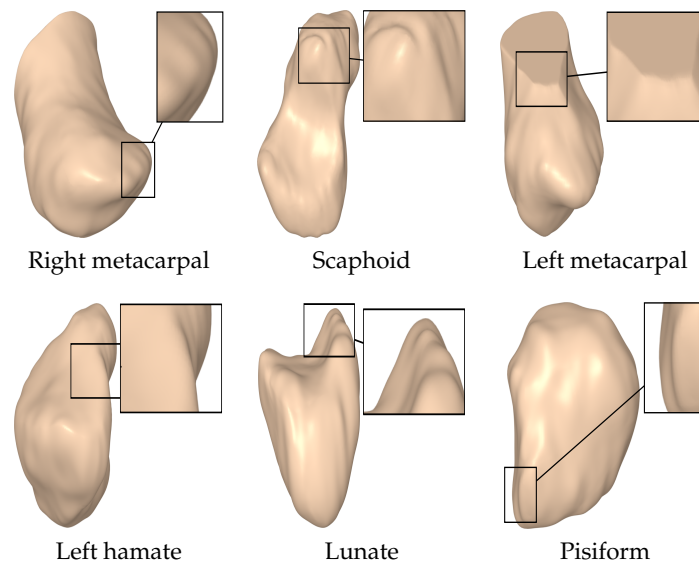


Figure 5. Carpal bone models.

4.1.1. Qualitative Comparison

The visual comparison was performed with the most prevalent methods of 3D mesh denoising, including BMD [5], MSAL [14], GMD [16] and RMD [17]. As shown in Figure 6, the noisy right metacarpal model was generated by adding a Gaussian noise with a standard deviation of $\sigma = 0.5$ to the vertices of the ground truth mesh along the vertex normals. As can be seen, the output results of BMD, MSAL, GMD and RMD still contained a considerable amount of noise in some regions of the denoised model, while the proposed approach removed the noise well and, at the same time, preserved the surface detail. Figure 7 displays the denoising results on the noisy scaphoid, left metacarpal, and left hamate models with a noise standard deviation of $\sigma = 0.5$, proportional to the mean edge length of the mesh. Notice again that the proposed approach preserved the edges well, while RMD tended to over-smooth the features. Further, the noise was mostly eliminated using our approach without affecting flat regions. Further, the sharp features were well preserved, as depicted in the enlarged views, which shows that the geometric structures and the fine details of the denoised carpal bone models were very well preserved.

Figure 8 shows the denoising results of the noisy scaphoid, lunate, and pisiform models with a higher noise standard deviation, $\sigma = 0.7$, proportional to the mean edge length of the mesh. As can be seen, RMD removed the noise relatively well but did not preserve the sharp features. The other

baseline methods did not remove the noise well and also tended to over-smooth the sharp regions, while our approach effectively removed the noise without creating any edge flips. While RMD yielded comparable results to our approach, it did not, however, preserve edges with the same effectiveness.

In all the experiments, we observed that our approach was able to suppress noise while preserving important geometric features of the carpal bone surfaces in a fast and efficient manner. This better performance is, in fact, consistent with a large number of 3D models used for experimentation.

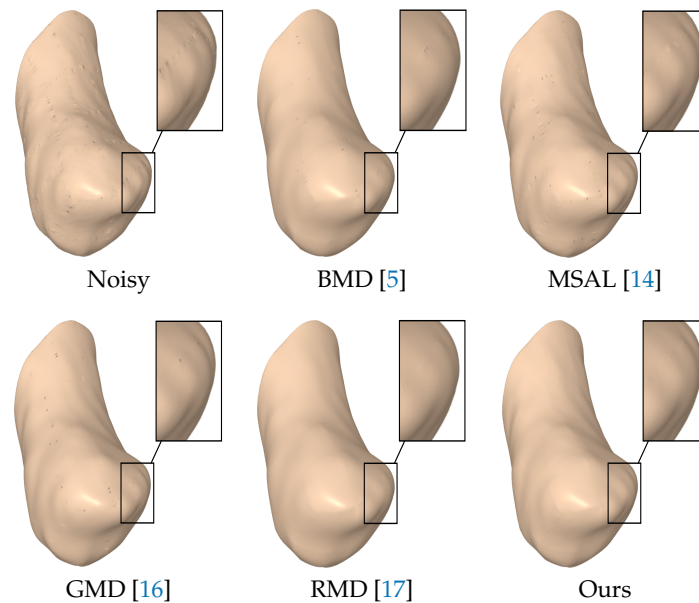


Figure 6. Surface denoising results of the noisy right metacarpal model corrupted by Gaussian noise with $\sigma = 0.5$. The magnified views of denoised models show that our method outperformed the baselines in preserving the surface features.

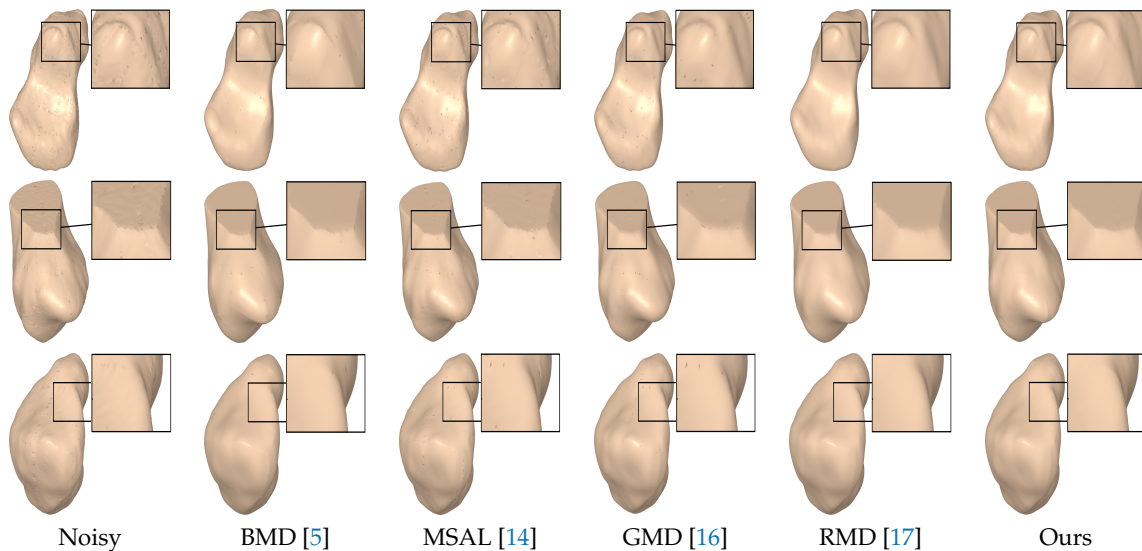


Figure 7. Surface denoising results for the noisy scaphoid, left metacarpal, and left hamate models. The noise standard deviation was set to $\sigma = 0.5$.

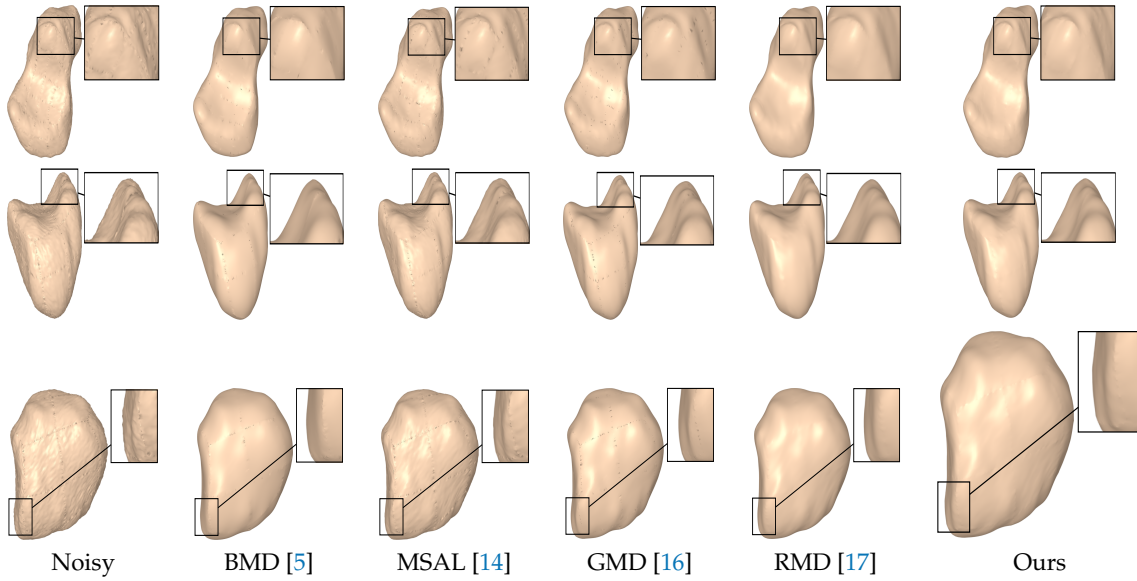


Figure 8. Surface denoising results for the noisy scaphoid, lunate, and pisiform models. The noise standard deviation was set to $\sigma = 0.7$.

4.1.2. Quantitative Comparison

To quantify the difference between the ground truth and estimated model, we used three different measures, namely, the mean orientation error metric, the face-normal error metric, and the face quality metric [17].

Let $\mathbb{M} = (\mathcal{V}, \mathcal{T})$ and $\hat{\mathbb{M}} = (\hat{\mathcal{V}}, \hat{\mathcal{T}})$ be the original and denoised models with vertex sets $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^n$ and $\hat{\mathcal{V}} = \{\hat{\mathbf{v}}_i\}_{i=1}^n$, and triangle sets $\mathcal{T} = \{\mathbf{t}_j\}_{j=1}^m$ and $\hat{\mathcal{T}} = \{\hat{\mathbf{t}}_j\}_{j=1}^m$, respectively.

Mean orientation error metric: The orientation error between the original model and the denoised one can be measured using the mean orientation error metric given by

$$E_o = \frac{1}{m} \sum_{j=1}^m \angle(\mathbf{n}(\mathbf{t}_j), \mathbf{n}(\hat{\mathbf{t}}_j)), \quad (20)$$

where $\mathbf{n}(\mathbf{t}_j)$ and $\mathbf{n}(\hat{\mathbf{t}}_j)$ are the unit face normals of $\mathbf{t}_j$ and $\hat{\mathbf{t}}_j$, respectively. The symbol, $\angle$, denotes the angle between two unit vectors and is defined as the inverse cosine of their dot product.

Face-normal error metric: To quantify the performance of the proposed approach, we computed the L^2 face-normal error metric given by

$$E_f(\mathbb{M}, \hat{\mathbb{M}}) = \frac{1}{\text{area}(\hat{\mathbb{M}})} \sum_{\hat{\mathbf{t}}_j \in \hat{\mathcal{T}}} \text{area}(\hat{\mathbf{t}}_j) \|\mathbf{n}(\mathbf{t}_j) - \mathbf{n}(\hat{\mathbf{t}}_j)\|, \quad (21)$$

where $\text{area}(\hat{\mathbf{t}}_j)$ is the area of $\hat{\mathbf{t}}_j$, and $\text{area}(\hat{\mathbb{M}})$ is the total area of the denoised mesh.

Face quality metric: The quality of mesh faces can be measured using the ratio of the circumradius-to-minimum edge length given by

$$Q = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{r_t}{\ell_t}, \quad (22)$$

where r_t and ℓ_t are the circumradius and minimum edge length of the associated triangle, respectively. In an ideal case, every face of the mesh should be an equilateral triangle with a quality index equal to $Q = 1/\sqrt{3}$.

The values of these metrics for our approach and the baseline methods are reported in Table 1. For fair comparison, we set the number of iterations to five for all the methods. Our approach yielded better or competitive results in terms of E_o and E_f for all models. Moreover, the values of Q for our method were lower than those of the baseline methods. The L^2 face-normal errors for the left metacarpal, scaphoid, lunate, right metacarpal, and left hamate are shown graphically in Figures 9–13. As can be seen in these figures, our method yielded the best overall results, indicating consistency with the subjective comparison.

Table 1. Quantitative comparison results using bilateral mesh denoising (BMD), multiscale anisotropic Laplacian (MSAL), guided mesh normal denoising (GMD), robust and high fidelity mesh denoising (RMD), and our approach.

Model	Method	E_o	$E_f \times 10^{-3}$	Q
Right metacarpal $ F = 27,912$ $ V = 13,958$	BMD	1.503	2.679	2.868
	MSAL	1.506	2.495	6.852
	GMD	1.503	2.183	7.973
	RMD	1.503	2.686	4.700
	Ours	1.470	1.011	1.623
Scaphoid $ F = 29,408$ $ V = 14,706$	BMD	1.506	7.770	5.226
	MSAL	1.530	16.838	2.247
	GMD	1.457	13.332	7.940
	RMD	1.453	3.976	1.888
	Ours	1.465	2.966	1.678
Left metacarpal $ F = 26,858$ $ V = 13,431$	BMD	1.510	5.652	2.449
	MSAL	1.512	3.275	1.592
	GMD	1.503	2.845	1.669
	RMD	1.485	2.554	6.404
	Ours	1.462	1.716	1.452
Left hamate $ F = 28,792$ $ V = 14,398$	BMD	1.494	3.855	13.354
	MSAL	1.506	2.445	5.597
	GMD	1.386	1.811	4.663
	RMD	1.418	1.847	2.826
	Ours	1.422	1.461	1.702

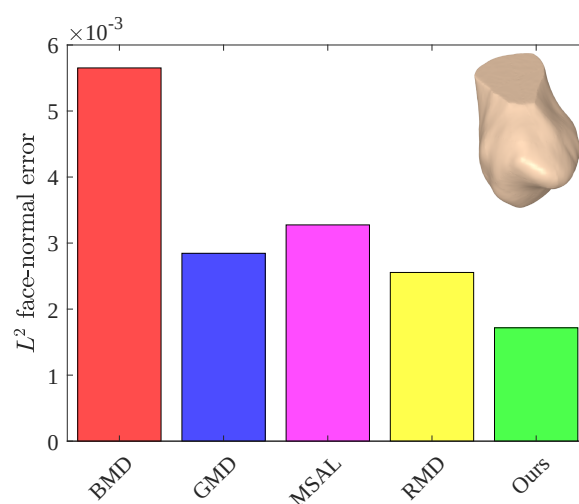


Figure 9. L^2 face-normal errors for the left metacarpal model.

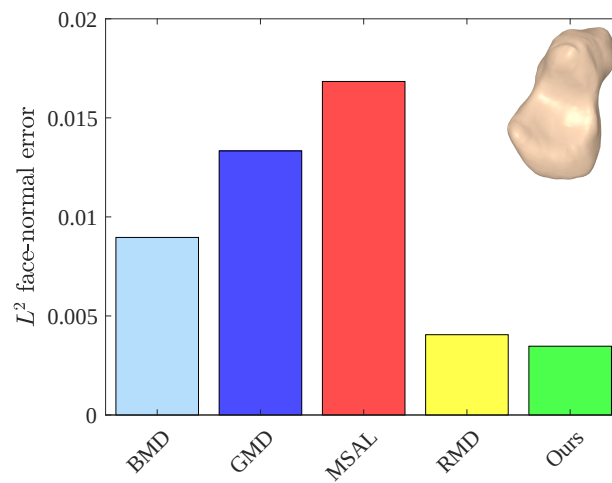


Figure 10. L^2 face-normal position errors for the scaphoid model.

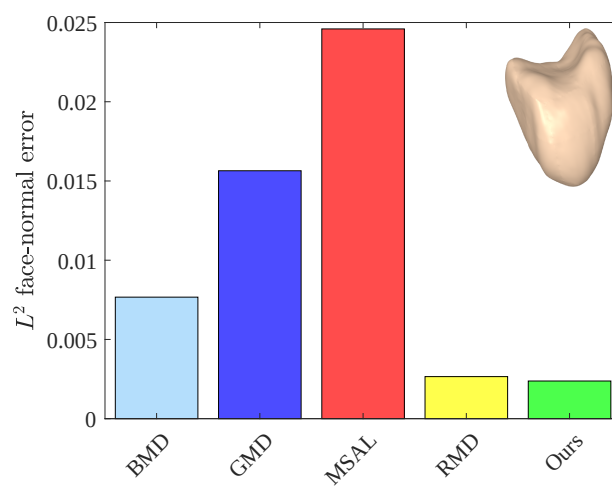


Figure 11. L^2 face-normal errors for the lunate model.

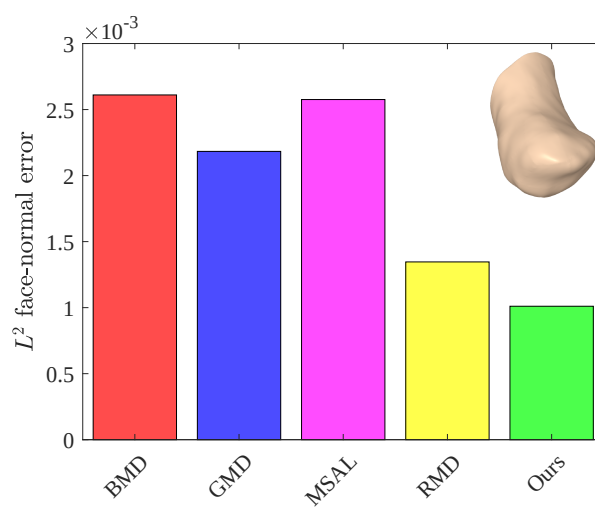


Figure 12. L^2 face-normal errors for the right metacarpal model.

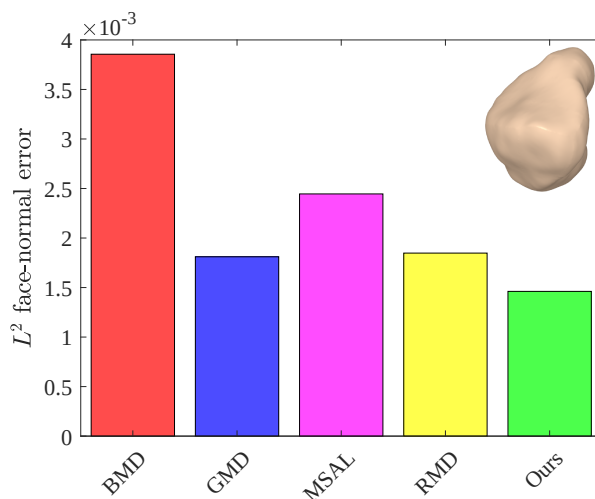


Figure 13. L^2 face-normal errors for the left hamate model.

4.1.3. Runtime Analysis

Most mesh denoising techniques perform filtering using a two-stage process by first filtering the face normals and then updating the vertex positions to match the filtered face normals, resulting in a computationally expensive process, particularly for large 3D meshes. Our method is, however, fast and simple to implement. Table 2 shows the runtime of our algorithm for different carpal bone models. In comparison, the runtimes (in seconds) per iteration for RMD, which is the best performing baseline method, were 2.555, 2.3004, 2.292 and 2.167 for the right metacarpal, scaphoid, left metacarpal, and left hamate, respectively. This strongly indicates that our algorithm not only performs well in terms of removing undesirable noise from bone surfaces, but is also computationally efficient.

Table 2. Runtime (in seconds) per iteration and number of iterations used for denoising different models.

Model	Vertices	Faces	Time (s)	N_{iter}
Right metacarpal	13,958	27,912	0.284	5
Scaphoid	14,706	29,408	0.286	10
Left metacarpal	13,431	26,858	0.487	5
Left hamate	14,398	28,792	0.213	10

5. Conclusions

In this paper, we presented a feature-preserving approach to surface denoising using a data-adaptive similarity in conjunction with matrix balancing. The proposed framework was formulated as a constrained minimization problem. The solution to this problem was estimated iteratively using the conjugate gradient method in an effort to recover sharp features from noisy surfaces. The qualitative and quantitative evaluation results demonstrate that our approach offers superior performance over existing mesh denoising techniques. For the future work, we plan to incorporate edge-aware filters into our framework to tackle data-driven geometry processing problems.

Author Contributions: Software, I.S.; Validation, I.S.; Investigation, I.S.; Original Draft Preparation, I.S.; Review & Editing, A.B.H.; Supervision, A.B.H.

Funding: This work was supported in part by NSERC Discovery Grant N00929.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Levy, B. *Polygon Mesh Processing*; AK Peters/CRC Press: Boca Raton, FL, USA, 2010.
2. Taubin, G. A signal processing approach to fair surface design. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 6–11 August 1995; pp. 351–358.
3. Yagou, H.; Ohtake, Y.; Belyaev, A. Mesh smoothing via mean and median filtering applied to face normals. In Proceedings of the Geometric Modeling and Processing, Wako, Japan, 10–12 July 2002; pp. 124–131.
4. Tasdizen, T.; Whitaker, R.; Burchard, P.; Osher, S. Geometric surface smoothing via anisotropic diffusion of normals. In Proceedings of the IEEE Visualization, Boston, MA, USA, 27 October–1 November 2002; pp. 125–132.
5. Fleishman, S.; Drori, I.; Cohen-Or, D. Bilateral mesh denoising. In Proceedings of the 2003 Symposium on Interactive 3D Graphics, Monterey, CA, USA, 28–30 April 2003; pp. 950–953.
6. Clarenz, U.; Diewald, U.; Rumpf, M. Processing textured surfaces via anisotropic geometric diffusion. *IEEE Trans. Image Process.* **2004**, *13*, 248–261.
7. Zhang, Y.; Ben Hamza, A. Vertex-based diffusion for 3D mesh denoising. *IEEE Trans. Image Process.* **2007**, *16*, 1036–1045.
8. Zheng, Y.; Fu, H.; Au, O.K.C.; Tai, C.L. Bilateral normal filtering for mesh denoising. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 1521–1530.
9. Weickert, J. *Anisotropic Diffusion in Image Processing*; Teubner-Verlag: Stuttgart, Germany, 1998.
10. Barash, D. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 844–847.
11. Paris, S.; Kornprobst, P.; Tumblin, J.; Durand, F. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH Courses*; Association for Computing Machinery: New York, NY, USA, 2007; pp. 1–50.
12. Kheradmand, A.; Milanfar, P. A general framework for regularized, similarity-based image restoration. *IEEE Trans. Image Process.* **2014**, *23*, 5136–5151.
13. Sun, X.; Rosin, P.L.; Martin, R.R.; Langbein, F.C. Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 925–938.
14. Huang, H.; Ascher, U. Fast denoising of surface meshes with intrinsic texture. *Inverse Probl.* **2008**, *24*, 034003.
15. El Ouafdi, A.F.; Ziou, D.; Krim, H. A smart stochastic approach for manifolds smoothing. *Comput. Graph. Forum* **2008**, *27*, 1357–1364.
16. Zhang, W.; Deng, B.; Zhang, J.; Bouaziz, S.; Liu, L. Guided mesh normal filtering. *Comput. Graph. Forum* **2015**, *34*, 23–34.
17. Yadav, S.K.; Reitebuch, U.; Polthier, K. Robust and High Fidelity Mesh Denoising. *IEEE Trans. Vis. Comput. Graph.* **2018**, doi:10.1109/TVCG.2018.2828818.
18. Elmoatz, A.; Lezoray, O.; Bogleux, S. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Trans. Image Process.* **2008**, *17*, 1047–1060.
19. Chung, M.K.; Wang, Y.; Wu, G. Heat Kernel Smoothing in Irregular Image Domains. *arXiv* **2017**, arXiv:1710.07849.
20. Chung, M.; Qiu, A.; Seo, S.; Vorperian, H. Unified heat kernel regression for diffusion, kernel smoothing and wavelets on manifolds and its application to mandible growth modeling in CT images. *Med. Image Anal.* **2015**, *2*, 63–76.
21. Sinkhorn, R.; Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pac. J. Math.* **1967**, *21*, 343–348.
22. Knight, P.A.; Ruiz, D. A fast algorithm for matrix balancing. *IMA J. Numer. Anal.* **2013**, *33*, 1029–1047.
23. Masoumi, M.; Rezaei, M.; Ben Hamza, A. Global spectral graph wavelet signature for surface analysis of carpal bones. *Phys. Med. Biol.* **2018**, *63*, 035034.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).