DOI: 10.1111/cgf.12139

COMPUTER GRAPHICS *forum*
Volume 00 (2013), number 0 pp. 1–12

Patch-Collaborative Spectral Point-Cloud Denoising

G. Rosman, A. Dubrovina and R. Kimmel

Computer Science Department, Technion - IIT, Israel
{rosman, nastyad, ron}@cs.technion.ac.il

Abstract

We present a new framework for point cloud denoising by patch-collaborative spectral analysis. A collaborative generalization of each surface patch is defined, combining similar patches from the denoised surface. The Laplace–Beltrami operator of the collaborative patch is then used to selectively smooth the surface in a robust manner that can gracefully handle high levels of noise, yet preserves sharp surface features. The resulting denoising algorithm competes favourably with state-of-the-art approaches, and extends patch-based algorithms from the image processing domain to point clouds of arbitrary sampling. We demonstrate the accuracy and noise-robustness of the proposed algorithm on standard benchmark models as well as range scans, and compare it to existing methods for point cloud denoising.

Keywords: Laplace-Beltrami, point cloud, denoising

ACM CCS: G.1.2 [Mathematics of Computing]: Approximation—Approximation of surfaces and contours I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—Geometric algorithms; languages; and systems I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Surface fitting

1. Introduction

In recent years, significant effort has been devoted to noise removal and smoothing of surfaces. With the increasing availability of commodity range scanners, even more attention is required to denoising of point clouds obtained from such depth sensors, where the surface is often strongly corrupted. This has led to various denoising and reconstruction algorithms for surfaces, triangulated or sampled as point sets (see [Lev98, FDCO03, FCOS05, YBS06, LCOLTE07, HLZ*09, ASGCO10] for a few examples).

There are various approaches for surface denoising and smoothing. Several algorithms for surface smoothing use the moving-least-squares (MLS) approach adopted from signal approximation theory [Lev98]. Since the estimation process is defined in most cases on the tangent plane, robust estimation of the tangent plane is required. A principled approach for robust tangent plane and surface estimation can be found, for example, in the robust MLS technique [FCOS05].

Other methods for surface smoothing are based on diffusion processes on surfaces. These include several algorithms for Laplacian-based mesh fairing [Tau95, DMSB99], where the Laplacian of the coordinate functions is used to define smoothing iterations. Several papers further generalize this approach using higher order differential operators [CDR00, SKS01, DR04, Xu09], normal diffusion [TWBO02], different types of curvatures [ZX06], or anisotropic diffusion [HP04, LP05].

Many techniques for surface denoising originated from denoising methods in image processing, continuing the extension of signal processing approach for surfaces by [Tau95]. Peng et al. [PSZ01] used Gaussian scale mixtures (GSM) on the multi-scale coefficients of the surface. Fleishman et al. extended the bilateral filtering to surface smoothing [FDCO03]. Zheng et al. [ZFAT11] applied bilateral filter on mesh normal field in order to perform anisotropic mesh denoising. Yoshizawa et al. [YBS06], adapted the non-local means (NLM) algorithm [ABM05] for surface smoothing, while using radial basis function (RBF) approximation to overcome the problems associated with matching sampled surface patches.

Local patch similarity was further used by Sharf et al. [SACO04] for context-aware surface completion, and by Zheng et al. [ZSW*10]—in urban scenes consolidation algorithm, exploiting the additional structure available in a human-made scene. Kim et al. [KMYG12] used repeatability of man-made objects for 3D acquisition and recognition of indoor scenery. In another related appoach, patch similarity about symmetry axes has been used by Bokeloh et al. [BBW*09] for surface reconstruction and denoising.

Guillemot et al. [GAB12] extended the non-local approach by decomposing the surface in a multi-scale fashion.

A different approach is to denoise an implicit volumetric representation, or level-set function of the surface instead of the points themselves. Thereby it is possible to apply a variety of image-domain denoising method directly. This was suggested, for example, for the NLM algorithm by Dong et al. [DYOD08], as well as many other reconstruction techniques. Although the added dimension makes these methods memory intensive, the Cartesian coordinates allow fast memory access. Thus, such methods have found use in real-time algorithms for surface modelling via fast dual algorithms for total variation [NLD11], and techniques based on octree Haar wavelets have shown an impressive scalability in terms of the number of points and accuracy (see for example the paper by Manson et al. [MPS08]).

Yet another family of methods for surface denoising operates on range scans. For these, again, image-processing algorithms are suitable without major modifications. Among this group are variants of the NLM algorithm [SBS08, HSJS10], as well as a sparsity-based approach [MSMS09, RDK12]. These algorithms, however, assume a very specific input which often cannot be generalized if the data is already given in a different format, or if multiple viewpoints are involved.

Most of the current state-of-the-art denoising techniques in image processing are collaborative in nature, bringing together several patches from the image and analysing the resulting signal group. This approach is also known as non-local multi-point modelling [KFEA10], and has been shown to work with heavily corrupted images. The analysis can be based on spectral [DFKE07] or sparsity [MBP*09] principles. In a sense, the NLM uses the mean estimator and can be considered a single-point approach for collaborative image denoising [KFEA10]. The recently introduced NL-Bayesian denoising [LBM12] interprets the patch-collaborative approach in a probabilistic framework, resulting in a new denoising algorithm.

The algorithm we describe in this paper builds upon the non-local, multi-point denoising framework, extending the signal processing approach to surface denoising into a collaborative spectral one. However, functions on point clouds are not defined on a Cartesian domain for which the spectral decomposition has a separable form. Even the bijection that exists between patch pixels in images is no longer available in noisy point clouds with irregular sampling, introduced questions of interpolations and sampling errors. These differences, among others, requires several fundamental changes when we address collaborative spectral denoising of point clouds. In our algorithm, each surface patch is grouped along with similar patches, and these are analysed together in order to obtain a joint smoothing operator. Denoising over several such groups results in a denoised version of the point cloud. The proposed algorithm defines, in a sense, an analogue of the Block-Matching 3D Denoising (BM3D, [DFKE07]) algorithm in the context of point clouds. We now shortly describe the BM3D algorithm, and its two main phases.

In both phases of the BM3D algorithm, for each patch in the image, a set of similar patches is collected. The patches are then stacked into a 3D volume. This 3D volume undergoes spectral (via DCT or wavelets) decomposition, and is denoised in the spectral domain, before being aggregated along with the results of other patches and their collaborative sets, into a final estimate of the image. In the first phase, the patches are denoised by hard thresholding of the spectral coefficients. In the second phase, the result of phase I is taken as an estimate of the clean signal. A Wiener filter is created in the spectral domain, and used to obtain the clean signal, based on the denoised estimate. The result is a high-quality regularization, typical for non-local multi-point algorithms. As we demonstrate in this paper, the various steps of this algorithm all have their parallel in the context of surface processing, resulting in a robust and accurate surface smoothing algorithm.

We concentrate on dealing with arbitrary point clouds, without an underlying connectivity. This scenario is often important in more general cases of surface processing where data is obtained from several scanners, or where large changes in view angles and depths make scanner-based triangulation inadequate. We therefore describe our algorithm in this setting, working with the discrete point sets themselves whenever possible. This is consistent with the observation taken from image denoising algorithms, where incorrect interpolation in the presence of noise can lead to smoothing artefacts. Exploiting a known connectivity of a mesh is one possible extension of the proposed algorithm, which is left for future work.

We describe our model in Section 2, and explain its steps in detail. We demonstrate in Section 3 the results of the proposed method on several models and subject to various noise levels and types.

2. Collaborative Spectral Denoising of Point Clouds

We now turn to describe the notion of collaborative spectral denoising and related concepts. The flow diagram of the complete algorithm is shown in Figure 4. Given a surface, we assume that for each small surface patch we can find a set of similar patches. This assumption is consistent with the one used in image denoising, for instance, in [ABM05, DFKE07]. Two examples of a such similar patches are shown in Figure 1. Note that for better visualization in all figures showing point clouds, the points constructing them are marked by small spheres, as in Figure 1.

A definition of the distance between surface patches is required to obtain these similar patches. Unlike the case of image processing, where data sampled on a Cartesian grid can be considered part of a vector space (endowed with several metrics), in the case of point cloud patches a different approach must be taken. In general, our notion of distance, or dissimilarity, between surface patches should be rotation invariant to account for differences in the local coordinate frames of the different patches.

One such distance is based on the *iterative closest point* (ICP) cost function, that matches two given patches with respect to rigid transformation (R, t),

$$d(\mathcal{P}_i, \mathcal{P}_j) = \min_{(R,t)} \|R\mathcal{P}_j + t - \mathcal{P}_i\|_2, \tag{1}$$

where $R \in SO(3)$ and $t \in \mathbb{R}^3$ represent rotation and translation. In our case, we use $\|\cdot\|_2$ to denote a quadratic distance approximation [PH03, MGPG04] between point clouds. Moreover, the patches used for matching are first denoised by a MLS in order to improve the similarity measure, as was suggested by Dabov et al. [DFKE07] in the context of image denoising. The ICP algorithm is initialized

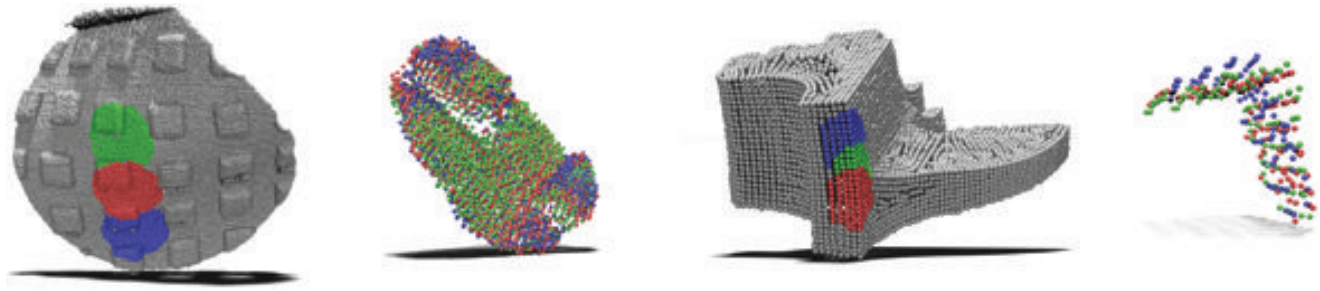

Figure 1: *Left-to-right: a patch in the point cloud along with two other similar patches, taken from [MBB10], followed by the collaborative point cloud obtained from these three patches, and a similar example from the Fandisk model of three similar patches, and the resulting collaborative patch.*

by aligning the means of the two patches. The resulting distance value over all computed patches is shown in Figure 11. Yet another option, suggested by Yoshizawa et al. [YBS06], is to approximate the surface by RBFs. Comparing the performance of such distance measures as well as other measures of local surface similarity is beyond the scope of this paper.

Once defined, this patch similarity also provides us with a transformation (R, t) that brings together similar patches into a single local coordinates frame. Once the patches are in the same coordinate frame, a transform can be defined simultaneously on points from the patches. This transform can then be used to denoise the surface. In order to obtain this transform we look at the spectral decomposition of our rotated and translated patches, calculated using the Laplace–Beltrami operator (LBO) [Ros97]. Specifically, in the same way that wavelets transforms and other dictionaries are used for image denoising and compression, the eigenfunctions of the LBO can be used for surface denoising. Other uses of LBO eigenfunctions include surface compression [KG00], analysis [Lév06, RWP06], [OSG08], recognition [BK10, SHCB11], invariant representations[Rus07], segmentation [SOCG10], surface deformation [BS08], and so forth [Sor06, ZVKD10].

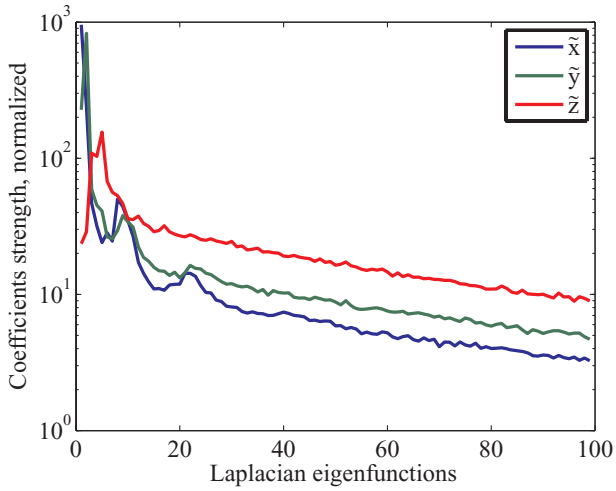

Figure 2: *The average normalized coefficient strength for the collaborative patch Laplacian eigenfunctions sampled on the Fandisk model. Red, green and blue represent the absolute magnitude of the normal (red) and two tangent coordinates in the local frame as estimated by our algorithm.*

2.1. Collaborative patch construction

In contrast to previous methods that utilize spectral properties of surfaces, we look at a set of similar patches as a local collaborative representation of the surface and perform the spectral decomposition of this combined representation. Specifically, for each patch $\mathcal{P}_i$ from the sampled surface we construct its *collaborative patch*, $\hat{\mathcal{P}}_i$, by registering onto it similar patches. In order to use only relevant patches to construct $\hat{\mathcal{P}}_i$, we keep patches whose distance to the patch $\mathcal{P}_i$ after the rigid alignment is smaller than some threshold, hereafter denoted by μ. This defines the *collaborative group* (of patches) for patch $\mathcal{P}_i$,

$$G_i = \left\{ \mathcal{P}_j \quad \text{s.t.} \quad \min_{R,t} \|R\mathcal{P}_j + t - \mathcal{P}_i\|_2 < \mu \right\}. \tag{2}$$

The above patches are combined into the collaborative patch $\hat{\mathcal{P}}_i$ using the transformations found in Equation (2) to form a single point cloud approximating the surface, but with many more data samples. This point cloud should give us a better approximating power in the presence of noise, assuming the surface has self-similarities.

The collaborative patch can be thought of as a new sampling of the surface (assuming the patches are sufficiently similar), or as a point cloud where each point is associated with a source point from our original sampled surface. We demonstrate two such point clouds in Figure 1. We note, however, that even in the lack of complete matching patch candidate, the algorithm still creates a thickly sampled patch, making spectral denoising more robust.

One should also make note of the type of (coordinate) functions used to describe surface smoothness. In general, we want to use relatively regular functions. This is achieved by choosing a suitable support region and a local coordinate frame for each collaborative patch, by applying the forward-support estimation algorithm introduced by Fleishman et al. [FCOS05]. We illustrate a collaborative patch and the smooth support region obtained for it in Figure 3, the top-left sub-figure. The spectral denoising procedure, described in the next sections, is then applied to the estimated support region, in the local coordinate frame of the patch, hereafter denoted by

$(\tilde{x}, \tilde{y}, \tilde{z})$. This also assures us, in the case of uniquely appearing patches, that the collaborative group only pertains to the common primitives inside each patch. Thus, such patches still receive ample support and can be denoised using a suitable collaborative group.

After construction of the collaborative patch, our algorithm proceeds in two main phases: employing a shrinkage operator based on the collaborative patch LBO eigenfunctions, followed by Wiener filtering based on the denoised estimate and the noisy point cloud. We denote these phases as phase I and II, respectively. In each phase we first gather for each patch a collaborative group, process it, and obtain a new estimator for the original patches that participated in the group. We need not create a collaborative patch for each vertex—all that is required is a sampling dense enough to cover each vertex with several estimators. As in the BM3D algorithm, we then average these estimators. For the surfaces we show in the paper, we used $P = 400$ collaborative groups, each using patches chosen from the patches surrounding the nearest 1000 vertices. Using fewer groups or smaller search windows, however, already gave satisfactory results.

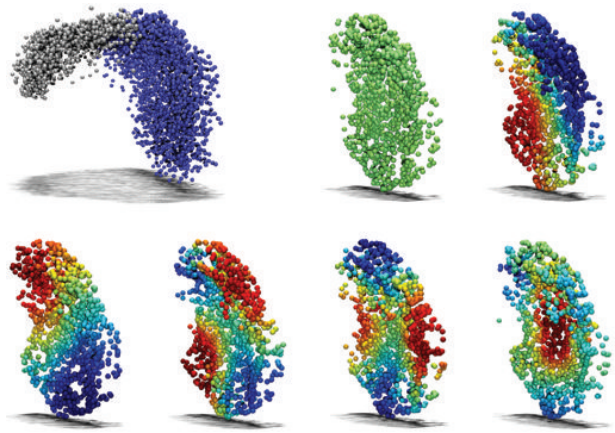

Figure 3: *Top left: a collaborative patch with the support, shown in blue, obtained as suggested in [FCOS05]. The rest of the images illustrate the first six eigenfunctions of the LBO defined over the chosen support.*

2.2. Collaborative point-cloud shrinkage

Both phase I and phase II of the algorithm require defining the discrete Laplacian approximation on the points participating in the collaborative patch. In our experiments, both the nearest-neighbour graph-Laplacian approximation and the method of Belkin et al. [BSW09] gave comparable results for noisy point clouds. We note that other algorithms can also be used for denoising the collaborative patch—for example, spectral analysis can be applied using wavelet bases constructed on a given point cloud, for example, using one of the techniques [REC11, CM11, Rus11].

The shrinkage is performed using the eigenfunctions of the LBO of the collaborative patch, forming a spectral domain for functions defined on it. Different eigenfuntions may be thought of as corresponding to different frequencies in this spectral domain [Lév06]. Figure 3 demonstrates the typical low-frequency nature of the first few LBO eigenfunctions.

Let f be a function defined on the point set of $\hat{\mathcal{P}}_i$. Let f_k denote the spectral coefficients of the collaborative patch,

$$f_k = \langle f, \phi_k \rangle , \tag{3}$$

where ϕ_k are the eigenfunctions of the LBO, indexed according to increasing order of the eigenvalues of the LBO they correspond to, and the inner product is the standard l_2 inner product defined on the discrete points set, although other choices are possible. The collaborative hard shrinkage of f is defined by

$$S_{\hat{\mathcal{P}}_i,\tau}(f) = \sum_k \hat{f}_k \phi_k, \tag{4}$$

where $\hat{f}_k$ denote the hard thresholding of each spectral coefficient f_k with threshold τ,

$$\hat{f}_k = \begin{cases} f_k, & |f_k| \geq \tau \\ 0, & |f_k| < \tau \end{cases} \tag{5}$$

We apply the shrinkage operator to each of the aligned local coordinate functions of the surface separately, in order to obtain the denoised version of the collaborative patch. This collaborative shrinkage process provides us with a set of estimators for each point in our point cloud, obtained for the collaborative groups the point belongs to.

We give an example of the estimated coefficients in Figure 2, averaged over 400 collaborative patches of the Fandisk model, shown in Figure 6, for the three local coordinate functions. In our examples, the first 100 eigenfunctions sufficed to describe the collaborative patch. As expected, the fast decay of these coefficients allows us to use only a few dozens of eigenfunctions without losing accuracy. It is interesting to note the strong spectral components in the tangent directions and small linear coefficient in the normal direction, as expected. The resulting graphs can be viewed as a power spectral density (PSD) estimation of the coordinate functions. Similarly, it is easy to show that independent, identically distributed (i.i.d.) Gaussian noise defined on the points is transformed into a uniform noise PSD (for non-overlapping patches). This gives us the theoretical motivation for the shrinkage operator on the collaborative patch. While the uniformity assumption is no longer true in the patch-collaborative case, the collaborative shrinkage operator works well in practice, as was shown in [DFKE07]. We now turn to describe several key practical aspects regarding the combination of estimators.

Averaging estimators: special care must be given to the way different patch estimators are combined together, as in the image-denoising case. We now describe the weights used to average different denoised estimates of each point. Let us denote by σ_D, a constant proportional to the patch radius. The weights w_{ij} of point i based on the collaborative patch $\mathcal{P}_j$ are obtained by multiplying three different aspect-related weights,

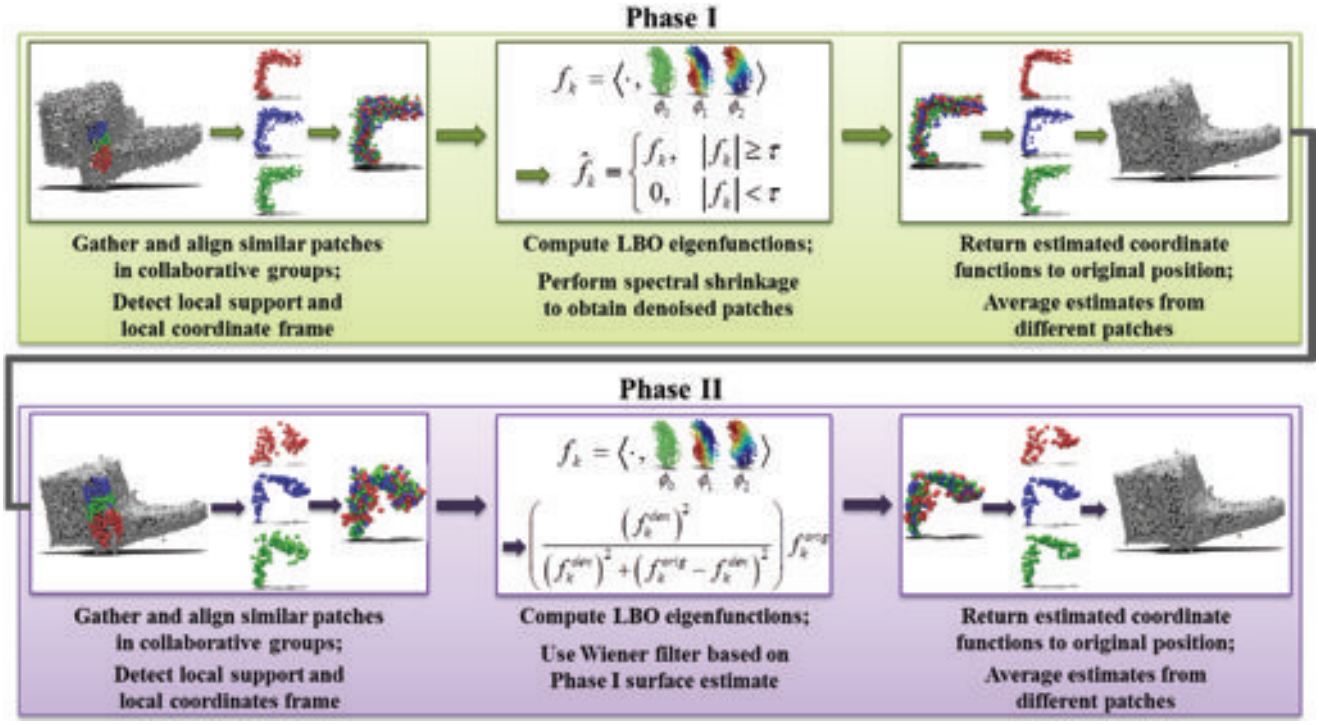

Figure 4: *An overview flowchart of the collaborative spectral denoising scheme.*

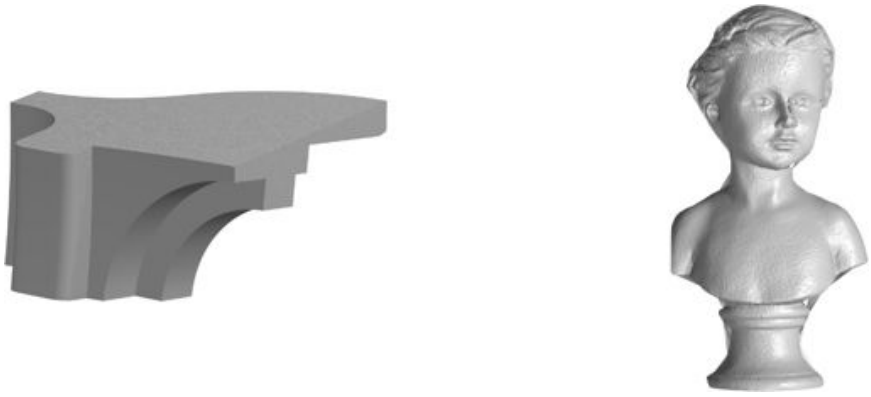

Figure 5: *The two standard models used in our experiments—Fandisk (left) and Bust (right).*

(1) A weight decreasing with the distance between the points i and j—in our case a Gaussian kernel,

$$\exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma_D^2\}, \tag{6}$$

where $\mathbf{x}_j$ is the centre of patch $\mathcal{P}_j$.

(2) The quality measure of the estimation of the patch $\mathcal{P}_j$ based on the collaborative group it belongs to, as defined by how much of the signal's (l_2) energy was removed by the shrinkage

$$\frac{\|S_{\hat{\mathcal{P}}_i,\tau}(f)\|}{\|f\|}. \tag{7}$$

(3) A coefficient inversely proportional to the density around point i in the collaborative patch $\mathcal{P}_j$. This is important since isolated points in the collaborative patch often have a poor approximation of the Laplace–Beltrami eigenfunctions, leading to a relatively poor estimation by the shrinkage operator at these points. This also helps prevents the ghosting artifacts that are often associated with examplar-based methods where partially matched patches are grouped together.

Handling boundaries: in addition, we replace the spectral estimators in patches that belong to boundary points with a simple linear estimator. For patches at the boundary, the estimation of local LBO basis is inaccurate, and a simplified model is expected to help avoid overfitting. This is similar to the approach taken in the data-dependent moving-least-squares algorithm [LCOL07]. Detecting

boundaries in point clouds have been reviewed and discussed in the context of manifold learning. We refer the readers to [RBBK10] for several approaches for boundary detection on point clouds and an overview of the topic.

Patch construction: the set of patches used for denoising is constructed as follows: we consider a subset of surface points, denoted by *seed points*. We then define a patch $\mathcal{P}_i$ centred at a seed point $\mathbf{x}_i$ as

$$\mathcal{P}_i = \{\mathbf{x}_j \mid \|\mathbf{x}_i - \mathbf{x}_j\| \leq r_0\}, \tag{8}$$

where r_0 is some predefined patch size.

Not all vertices can be used to seed a collaborative group due to the computational effort required. Special care must therefore be given to choosing a uniform covering of patches. In order to assure spatially uniform selection, the farthest point sampling strategy ([Gon85, HS85, ELPZ97]) can be used. This, however, does not guarantee that each point receives a sufficient number of estimators. In order to verify this, we look at each point's estimator weights *effective sample size*,

$$N_{\text{eff}}(\mathbf{x}_i) = \frac{\left(\sum_j w_{ij}\right)^2}{\sum_j \left(w_{ij}\right)^2}. \tag{9}$$

We use this measure, taken from the particle filtering literature [DGA00], by selecting the farthest point from the points whose effective sample size is above the median. This strategy gave us a reasonably uniform coverage while ensuring that all points are sufficiently represented in multiple collaborative groups. We therefore use it, for both the first and second phases of our algorithm.

2.3. Collaborative point-cloud Wiener filtering

The spectral decomposition of functions on the collaborative patch also allows us to define a Wiener filtering process, similar to the second stage of the BM3D algorithm. As in the BM3D [DFKE07] or in the WienerShrink [GSB97] algorithms, the denoised surface from phase I acts as our assumed clean signal, and the difference in the coordinate functions acts as the assumed additive noise.

Specifically, let f_k^{orig} denote the spectral coefficients of the original noisy signal, and let f_k^{den} denote the coefficients of the denoised estimate obtained from phase I. The Wiener-filtered spectral coefficients are defined as

$$f_k^{\text{wien}} = \left(\frac{\left(f_k^{\text{den}}\right)^2}{\left(f_k^{\text{den}}\right)^2 + \left(f_k^{\text{orig}} - f_k^{\text{den}}\right)^2}\right) f_k^{\text{orig}}. \tag{10}$$

Similar to phase I, we construct the collaborative groups and collaborative patches, now using the denoised estimate of the point cloud from phase I. It is important that the transform computed in phase II will differ significantly from the transform found in the first phase, as was discussed in the context of the WienerShrink algorithm [GSB97]. As in the BM3D algorithm, the different choice of patches at phase II suffices to produce a different transform and ensure the effectiveness of the Wiener-filtering approach to shrinkage. Another possibility is to use a different spectral transform such as one of [REC11, CM11, Rus11].

An overall algorithmic description of the proposed approach is given as Algorithm 1. Its flow diagram is shown in Figure 4.

Algorithm 1 Collaborative spectral denoising of point-cloud surfaces.

1: Obtain initial estimation via moving least-squares.
2: *Phase I*: Obtain denoised surface via collaborative spectral shrinkage.
3: **for** each patch $\mathcal{P}_i$, $i = 1, 2, \ldots, P$ **do**
4: Collect collaborative group G_i.
5: Build collaborative patch $\hat{\mathcal{P}}_i$.
6: Estimate a local coordinate frame and coordinate functions $(\tilde{x}, \tilde{y}, \tilde{z})$.
7: Apply shrinkage operator on collaborative patch's coordinate functions.
8: Return patch estimates to their original location, averaging overlapping estimates together.
9: **endfor**
10: *Phase II*: Obtain denoised surface via collaborative spectral Wiener filtering.
11: **for** each patch $\mathcal{P}_i$, $i = 1, 2, \ldots, P$ **do**
12: Collect collaborative group G_i.
13: Build collaborative patch $\hat{\mathcal{P}}_i$.
14: Estimate a local coordinate frame and coordinate functions $(\tilde{x}, \tilde{y}, \tilde{z})$.
15: Apply Wiener filter on collaborative patch's coordinate functions, using the denoised estimate from phase I.
16: Return patch estimates to their original location, averaging overlapping estimates together.
17: **end for**

The above algorithm was implemented using Matlab© and the Point Cloud Library (PCL) [RC11]. Currently, the current implementation of the algorithm is computationally demanding, taking a few hours to run on surfaces with $15K$ vertices, depending on parameters. We note, however, that the most computationally costly stages of the algorithm are the ICP optimization and spectral decomposition calculation, performed in both denoising phases. Since, however, collaborative groups are analysed independently, it is possible to significantly speed up the algorithm by parallelizing the calculations.

3. Results

We now demonstrate the results of our approach on several example models, with different levels of noise. In Figures 6 and 8, we show the denoising results on the Fandisk and Bust models. The point clouds were added a Gaussian noise with standard deviations of 0.005, 0.01 and 0.02 for the Fandisk model, and 0.01 for the Bust model examples, or 1%, 2% and 4% and 0.43% of the objects' diameters, respectively.

Figure 6: *The denoising results for the Fandisk model. Top-to-bottom: results for additive componentwise Gaussian noise with standard deviation* $\sigma = 0.005, 0.01$ *and* 0.02*. Left-to-right: the noisy model, non-local means result, bilateral filtering result, moving-least-squares result using PCL [RC11], the result after Phase I of the proposed method, the result after Phase II of the proposed method. The surface reconstruction from point clouds was performed using [Gia11].*

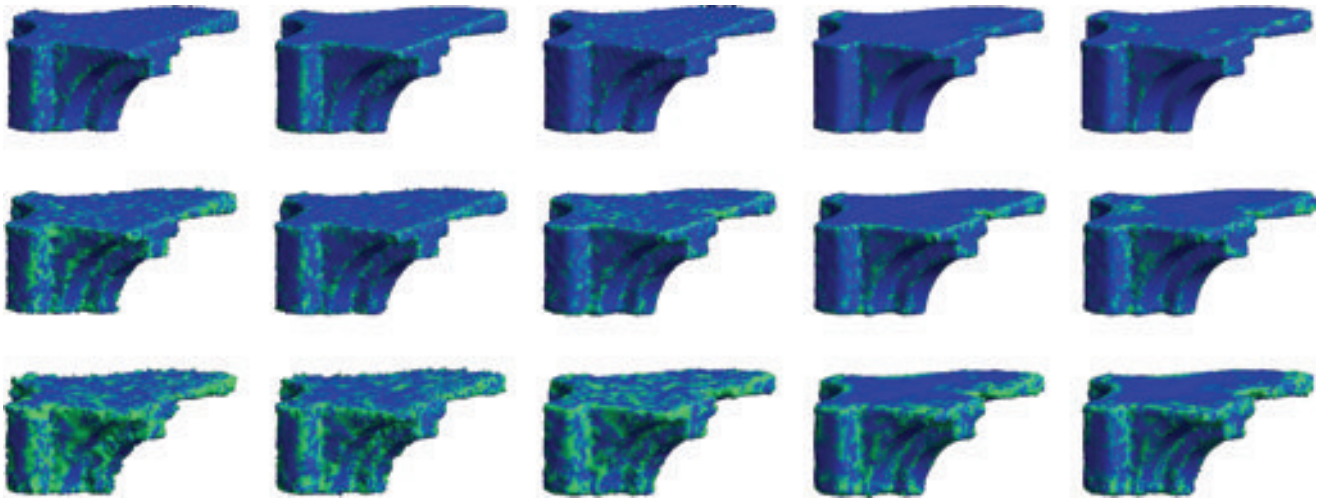

Figure 7: *A residual plot, showing the method noise for various methods. Left-to-right: NL-means, bilateral filtering, MLS, and phase I and phase II of our method, under various noise levels. Colours measure point-to-plane error. Top-to-bottom: standard deviations of* $\sigma = 0.005$*,* 0.01*,* 0.02*.*

Besides visual comparison, we measure the mean-squared-error (MSE), and median-of-squares (MedSq) estimate. We compare our method to the MLS implementation available in the Point Cloud Library (PCL) [RC11], as well as an implementation of NLM and bilateral filtering algorithms for point clouds.

The MSE for point cloud denoising is given by the ICP cost measure

$$\min_{R\in SO(3),t\in\mathbb{R}^3} \frac{1}{N}\sum_{i=1}^{N} d^2(Rp_i^{\text{den}} + t, \mathcal{S}^{\text{GT}}), \tag{11}$$

where $\mathcal{S}^{\text{GT}}$ and p_i^{den} denote the ground-truth (original) surface and the denoised cloud points, respectively, and $d(\cdot,\cdot)$ denotes a point-to-surface distance. In our case we used a quadratic point-to-surface distance approximation [PH03] due to its relatively high accuracy in the case of sparsely sampled surfaces. Similarly, we look at MedSq, the median of the summands in Equation (11). While MSE is the classical choice for denoising performance, MedSq is quite relevant when the denoising is a preprocessing step for robust algorithms which ignore outlier points, as is often the case in ICP [RL01], or robust fitting algorithms. The error measures for the Fandisk model are summarized in Table 1, and for the Bust model—in Table 2.

Figure 8: *Left to right: a reconstruction of the Bust model with added noise of $\sigma = 0.01$, with moving-least-squares results using PCL [RC11], denoising results by phase I and phase II of the proposed algorithm.*

Table 1: *Mean squared error (MSE) and median of squares (MedSq) of the point cloud after denoising for the Fandisk model in Figure 6, for the noise levels shown in the figures.*

Noise level	$\sigma = 0.005$		$\sigma = 0.01$		$\sigma = 0.02$	
	MSE	MedSq	MSE	MedSq	MSE	MedSq
Noisy surface	2.58×10^{-5}	1.24×10^{-5}	1.01×10^{-4}	4.97×10^{-5}	3.86×10^{-4}	1.82×10^{-4}
NL Means	1.42×10^{-5}	4.44×10^{-6}	3.87×10^{-5}	1.24×10^{-5}	1.08×10^{-4}	3.49×10^{-4}
Bilateral	1.27×10^{-5}	2.52×10^{-6}	2.60×10^{-5}	0.84×10^{-5}	8.04×10^{-5}	2.71×10^{-5}
MLS	1.23×10^{-5}	4.22×10^{-6}	2.88×10^{-5}	1.01×10^{-5}	7.89×10^{-5}	3.03×10^{-5}
Proposed approach (phase I)	1.06×10^{-5}	2.60×10^{-6}	1.88×10^{-5}	4.99×10^{-6}	6.36×10^{-5}	1.65×10^{-5}
Proposed approach (phase II)	0.96×10^{-5}	1.67×10^{-6}	1.54×10^{-5}	4.83×10^{-6}	5.26×10^{-5}	1.38×10^{-5}

Table 2: *Mean squared error (MSE) and median of squares (MedSq) of the point cloud after denoising for the Bust model in Figure 8, for the noise level shown in the figure.*

Noise level	$\sigma = 0.01$	
	MSE	MedSq
Noisy surface	9.52×10^{-5}	4.32×10^{-5}
NL Means	2.20×10^{-5}	6.85×10^{-6}
Bilateral	1.49×10^{-5}	5.76×10^{-6}
MLS	1.26×10^{-5}	4.08×10^{-6}
Proposed approach (phase I)	1.30×10^{-5}	3.82×10^{-6}
Proposed approach (phase II)	1.21×10^{-5}	3.90×10^{-6}

Both tables demonstrate that the proposed algorithm performs favourably compared to the state of the art in terms of denoising accuracy of point clouds with strong noise. Parameters of all methods were taken so as to minimize the resulting MSE. Note that the NLM implementation used in our experiments attempts to fit together nearby similar patches, and average them. It does so while referring to the point samples themselves, which are quite sparse, and while fitting together patches from the area which may not convey a lot of information. Thus, it is often less successful than the MLS algorithm, which approximates the surface and thus is expected to be more accurate and robust to sampling artifacts. The MLS, on the other hand, efficiently smoothes the point clouds, but then it also smoothes sharp surface features, which are preserved by the proposed algorithm. We also compare to an implementation of the bilateral filter for point clouds [FDCO03], with a robust estimation of the normal. Since most of the surfaces shown in this paper appear locally as a relatively simple signal, they agree with this filter, which surpasses the results of the NLM. In order to see the regions where the algorithms encountered problems, residual plots, or method-noise plots in Figure 7 demonstrate the error left after denoising by each method, again in terms of distance from the original surface, per surface point.

Figure 9 demonstrates the results of the proposed algorithm applied to data obtained from a structured light scanner. The resulting surface is clearly smoothed in a plausible manner, removing most of the scanning artefacts, while still managing to capture subtle details such as the lips and eyes areas where sampling was irregular and other techniques may smooth the area altogether. We note that this result is obtained despite the fact that the noise is far from a Gaussian i.i.d model, as is often assumed in shrinkage-based denoising.

Figure 10 illustrates denoising of a point cloud having both sharp features and smooth areas, taken from [RDK]. The search radius for detecting similar patches was taken to be sufficiently small, so that each protrusion is considered separate, and each collaborative patch

Figure 9: *Left-to-right: raw range-scanned data triangulated into a surface, moving-least-squares results using PCL [RC11], and the results of phase I and II of the proposed algorithm. Note the delicate structures such as the eye and lip areas, and the relative smoothness of the forehead.*

Figure 10: *Left to right: original shape, with added Gaussian noise, denoised result of the proposed method.*

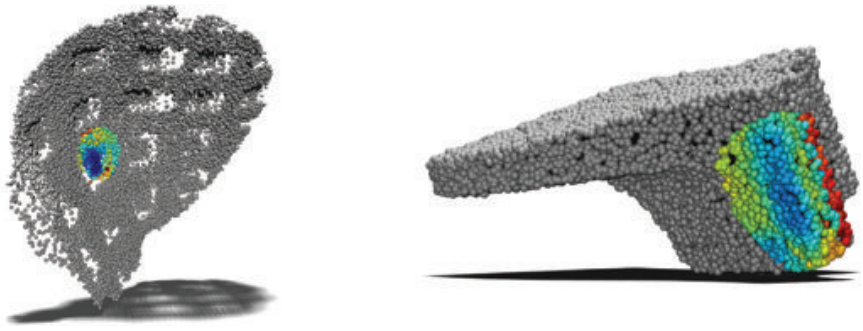

Figure 11: *The measured distances for a patch over from the half-smooth and Fandisk models. Blue signifies low distances, red denotes high distances. Grey points are outside the search area for the specific patch.*

contains merely different variations of the same patch taken along dominant features. While the incomplete set of similar patches limits the power of the collaborative filter, results are still convincing. This example also raises the question of non-repetitive patches in non-local denoising methods. In general, the implicit assumption made in non-local methods is that patches are small enough so that variations of the patch appear at least a few times. This assumption may break down if the surface sampling is not dense enough, the local feature size [AB99] is relatively small, and the surface does not have a repetitive structure. We note that such patches are less influential in the denoised results because of the choice of patch support set, and the weight given to such estimators, as described in Section 2.2. An attempt to address this issue and allow several scales to be used simultaneously in denoising has been made in the context of non-local image denoising [LFSB09]. In our paper we restrict ourselves to the usual assumption of a single scale of patches.

We illustrate the above limitation in the left sub-figure of Figure 11, by showing the measured distances between patches according to Equation (1), over the search area. Despite this intentionally posed limitation, each patch still successfully undergoes a robustified spectral shrinkage and Wiener filtering. Even though the collaborative patches are limited to perturbed replicas of the same patch, and each patch is limited in the number of sampled points, the protrusions are still preserved in the smoothed version while the noise is eliminated, as shown in Figure 10. A different case of selection for a collaborative group is shown for the Fandisk model, in the right sub-figure of Figure 11, where many low-distance patches can be found, limited mostly by the surface density.

4. Conclusions

In this paper we demonstrated a method for patch-collaborative spectral denoising of surfaces, generalizing the BM3D algorithm from images to surfaces. The suggested method reaches state-of-the-art results in denoising and smoothing of point cloud surfaces, and suggests other possibilities for surface reconstruction that we intend to explore in future work.

Acknowledgement

This research was supported by the European Community's FP7-ERC program, grant agreement no. 267414.

References

[AB99] Amenta N., Bern M.: Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry 22* (1999), 481–504.

[ABM05] Antoni Buades B. C., Morel J.-M. : A review of image denoising algorithms, with a new one. *SIAM Interdisciplinary Journal 4* (2005), 490–530.

[ASGCO10] Avron H., Sharf A., Greif C., Cohen-Or D.: l_1-sparse reconstruction of sharp point set surfaces. *Transactions on Graphics 29*, *5* (November 2010), 135:1–135:12.

[BBW*09] Bokeloh M., Berner A., Wand M., Seidel H. P., Schilling A.: Symmetry Detection Using Feature Lines. *Computer Graphic Forum 28*, *2* (2009), 697–706.

[BK10] Bronstein M. M., Kokkinos I. : Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proceedings of CVPR* (2010), pp. 1704–1711.

[BS08] Botsch M., Sorkine O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, *1* (2008), 213–230.

[BSW09] Belkin M., Sun J., Wang Y.: Constructing Laplace operator from point clouds in rd. In *Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2009), SIAM, pp. 1031–1040.

[CDR00] Clarenz U., Diewald U., Rumpf M.: Anisotropic geometric diffusion in surface processing. In *Proceedings of the 11th IEEE Visualization Conference* ((New York, NY, USA, 2000), ACM, pp. 397–405.

[CM11] Chen G., Maggioni M.: Multiscale geometric dictionaries for point-cloud data. In *Proceeding of the International Conference on Sampling Theory and Applications (SampTA)* (2011).

[DFKE07] Dabov K., Foi A., Katkovnik V., Egiazarian K. : Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing 16* (2007), 2007.

[DGA00] Doucet A., Godsill S., Andrieu C.: On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing 10*, *3* (July 2000), 197–208.

[DMSB99] Desbrun M., Meyer M., Schröder P., Barr A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH* (1999), pp. 317–324.

[DR04] Droske M., Rumpf M.: A level set formulation for Willmore flow. *Interfaces and Free Boundaries 6*, *3* (2004), 361–378.

[DYOD08] Dong B., Ye J., Osher S., Dinov I.: Level set based nonlocal surface restoration. *Multiscale Modeling & Simulation 7*, *2* (2008), 589–598.

[ELPZ97] Eldar Y., Lindenbaum M., Porat M., Zeevi Y.: The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing 6*, *9* (September 1997), 1305–1315.

[FCOS05] Fleishman S., Cohen-Or D., Silva C. T.: Robust moving least-squares fitting with sharp features. In *Proceedings of the SIGGRAPH* (New York, NY, USA, 2005), ACM, pp. 544–552.

[FDCO03] Fleishman S., Drori I., Cohen-Or D.: Bilateral mesh denoising. *Transactions on Graphics 22*, *3* (2003), 950–953.

[GAB12] Guillemot T., Almansa A., Boubekeur T.: Non local point set surfaces. In *Proceedings of 3DIMPVT* (2012), pp. 324–331.

[Gia11] Giaccari L.: Surface reconstruction toolbox 0.2, Jan 2011.

[Gon85] Gonzalez T. F.: Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science 38* (1985), 293–306.

[GSB97] Ghael S., Sayeed A. M., Baraniuk R. G.: Improved wavelet denoising via empirical Wiener filtering. In *Proceedings of SPIE* (1997), pp. 389–399.

[HLZ*09] Huang H., Li D., Zhang H., Ascher U., Cohen-Or D.: Consolidation of unorganized point clouds for surface reconstruction. In *Proceedings of TOG*(2009), vol. 28, p. 176.

[HP04] Hildebrandt K., Polthier K.: Anisotropic filtering of nonlinear surface features. In *Proceedings of CGF* (2004), vol. 23, Wiley-Blackwell, Hoboken, NJ, USA, pp. 391–400.

[HS85] Hochbaum D., Shmoys D.: A best possible approximation for the k-center problem. *Mathematics of Operations Research 10*, *2* (1985), 180–184.

[HSJS10] Huhle B., Schairer T., Jenke P., Strasser W.: Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Computer Vision and Image Understanding 114*, *12* (2010), 1336–1345. Special issue on Time-of-Flight Camera Based Computer Vision.

[KFEA10] Katkovnik V., Foi A., Egiazarian K., Astola J.: From local kernel to nonlocal multiple-model image denoising. *International Journal of Computer Vision 86*, *1* (2010), 1–32.

[KG00] Karni Z., Gotsman C.: Spectral compression of mesh geometry. In *Proceedings of SIGGRAPH* (New York, NY, USA, 2000), ACM Press, pp. 279–286.

[KMYG12] Kim Y. M., Mitra N. J., Yan D.-M., Guibas L.: Acquiring 3D indoor environments with variability and repetition. *TOG 31*, *6* (2012), 138.

[LBM12] Lebrun M., Buades A., Morel J.-M.: Implementation of the non-local bayes image denoising algorithm. *Image Processing On Line* (2012). preprint.

[LCOL07] Lipman Y., Cohen-Or D., Levin D.: Data-dependent MLS for faithful surface approximation. In *Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 59–67.

[LCOLTE07] Lipman Y., Cohen-Or D., Levin D., Tal-Ezer H.: Parameterization-free projection for geometry reconstruction. In *Proceedings of TOG* (2007), vol. 26, article no. 22.

[Lev98] Levin D.: The approximation power of moving least-squares. *Mathematics of Computation 67*, *224* (1998), 1517–1531.

[Lév06] Lévy B.: Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *Proceedings of SMI* (2006).

[LFSB09] Lou Y., Favaro P., Soatto S., Bertozzi A.: Nonlocal similarity image filtering. In *Proceedings of the 15th International Conference on Image Analysis and Processing* (Berlin, Heidelberg, 2009), ICIAP '09, Springer-Verlag, pp. 62–71.

[LP05] Lange C., Polthier K.: Anisotropic smoothing of point sets. *Computer Aided Geometric Design 22*, *7* (2005), 680–692.

[MBB10] Mitra N. J., Bronstein A. M., Bronstein M. M. : Intrinsic regularity detection in 3D geometry. In *Proceedings of ECCV (3)* (2010), pp. 398–410.

[MBP*09] Mairal J., Bach F., Ponce J., Sapiro G., Zisserman A.: Non-local sparse models for image restoration. In *ICCV* (2009).

[MGPG04] Mitra N. J., Gelfand N., Pottmann H., Guibas L.: Registration of point cloud data from a geometric optimization perspective. In *Eurographics Symposium on Geometry Processing* (2004), pp. 23–31.

[MPS08] Manson J., Petrova G., Schaefer S.: Streaming surface reconstruction using wavelets. *Computer Graphics Forum 27*, *5* (2008), 1411–1420.

[MSMS09] Mahmoudi M., Sapiro G., Mahmoudi M., Sapiro G. : *Sparse Representations For Three-Dimensional Range Data Restoration*. IMA Preprint 2280, University of Minnesota, 2009.

[NLD11] Newcombe R. A., Lovegrove S., Davison A. J. : DTAM: Dense tracking and mapping in real-time. In *Proceedings of ICCV* (2011), pp. 2320–2327.

[OSG08] Ovsjanikov M., Sun J., Guibas L.: Global intrinsic symmetries of shapes. In *Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2008), Eurographics Association, pp. 1341–1348.

[PH03] Pottmann H., Hofer M.: Geometry of the squared distance function to curves and surfaces. In *Visualization and Mathematics III*. H.C. Hege and K. Polthier (Eds.), Springer, the Netherlands, pp. 223–244.

[PSZ01] Peng J., Strela V., Zorin D.: A simple algorithm for surface denoising. In *Proceedings of the conference on Visualization '01* (Washington, DC, USA, 2001), IEEE Computer Society (2003), pp. 107–112.

[RBBK10] Rosman G., Bronstein M. M., Bronstein A. M., Kimmel R.: Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision 89*, *1* (2010), 56–68.

[RC11] Rusu R. B., Cousins S.: 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation* (Shanghai, China, May 9–13, 2011), IEEE.

[RDK] Rosman G., Dubrovina A., Kimmel R.: Scanned 3D denoising dataset. http://tx.technion.ac.il/rosman/dataset.html. Accessed 9 May 2013

[RDK12] Rosman G., Dubrovina A., Kimmel R.: Sparse modeling of shape from structured light. In *Proceedings of 3DIMPVT* (2012), pp. 456–463.

[REC11] Ram I., Elad M., Cohen I.: Redundant wavelets on graphs and high dimensional data clouds. *IEEE Signal Processing Letter 19*, *5* (2012), 291–294.

[RL01] Rusinkiewicz S., Levoy M.: Efficient variants of the ICP algorithm. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling (3DIM)* (June 2001), IEEE, pp. 145–152.

[Ros97] Rosenberg S.: *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. Cambridge University Press, Cambridge, UK, 1997.

[Rus07] Rustamov R. M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 225–233.

[Rus11] Rustamov R. M.: Average interpolating wavelets on point clouds and graphs. *CoRR abs/1110.2227* (2011).

[RWP06] Reuter M., Wolter F.-E., Peinecke N.: Laplace-Beltrami spectra as "shape-DNA" of surfaces and solids. *Computer-Aided Design 38*, *4* (2006), 342–366.

[SACO04] Sharf A., Alexa M., Cohen-Or D.: Context-based surface completion. In *Proceedings of TOG* (2004), vol. 23, ACM, pp. 878–887.

[SBS08] Schall O., Belyaev A., Seidel H.-P.: Adaptive feature-preserving non-local denoising of static and time-varying range data. *Computer-Aided Design 40*, *6* (2008), 701–707.

[SHCB11] Sharma A., Horaud R., Cech J., Boyer E. : Topologically-robust 3D shape matching based on diffusion geometry and seed growing. In *Proceedings of CVPR* (2011), pp. 2481–2488.

[SKS01] Schneider R., Kobbelt L., Seidel H.-P.: Mathematical methods for curves and surfaces. In *Improved Bi-Laplacian Mesh Fairing*. T. Lyche and L. L Schumaker (Eds.). Vanderbilt University, Nashville, TN, USA (2001), pp. 445–454.

[SOCG10] Skraba P., Ovsjanikov M., Chazal F., Guibas L.: Persistence-based segmentation of deformable shapes. In *Proceedings of the CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment* (June 2010), pp. 45–52.

[Sor06] Sorkine O.: Differential representations for mesh processing. *Computer Graphics Forum 25*, *4* (2006), 789–807.

[Tau95] Taubin G.: A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 351–358.

[TWBO02] Tasdizen T., Whitaker R., Burchard P., Osher S.: Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), VIS '02, IEEE Computer Society, pp. 125–132.

[Xu09] Xu G.: Mixed finite element methods for geometric modeling using general fourth order geometric flows. *Computer Aided Geometric Design 26*, *4* (May 2009), 378–395.

[YBS06] Yoshizawa S., Belyaev A., Seidel H. P.: Smoothing by example: mesh denoising by averaging with similarity-based weights. In *Proceedings of International Conference on Shape Modelling and Applications* (2006), pp. 38–44.

[ZFAT11] Zheng Y., Fu H., Au O., Tai C.: Bilateral normal filtering for mesh denoising. *IEEE Transactions on Visualization and Computer Graphics 17*, *10* (2011), 1521–1530.

[ZSW*10] Zheng Q., Sharf A., Wan G., Li Y., Mitra N. J., Cohen-Or D., Chen B.: Non-local scan consolidation for 3D urban scenes. *ACM Transactions on Graphics 29*, *4* (July 2010), 94:1–94:9.

[ZVKD10] Zhang H., Van Kaick O., Dyer R.: Spectral mesh processing. In *Proceedings of CGF* (2010), Wiley Online Library, vol. 29, pp. 1865–1894.

[ZX06] Zhao H., Xu G.: Triangular surface mesh fairing via Gaussian curvature flow. *Journal of Computational and Applied Mathematics 195*, *1* (October 2006), 300–311.