Patch-Collaborative Spectral Surface Denoising

Guy Rosman, Anastasia Dubrovina and Ron Kimmel
Department of Computer Science
Technion, Haifa 32000, Israel

May 16, 2012

Abstract

We present a new framework for denoising of point clouds by patch-collaborative spectral analysis. A collaborative generalization of each surface patch is defined, combining similar patches from the 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.

The resulting denoising algorithm competes favorably with state-of-the-art approaches, and extends patch-matching 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.

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. This has lead to a multitude of denoising and reconstruction algorithms for surfaces, triangulated or sampled as point sets (see [20, 14, 13, 48, 2] for a few examples).

There are various approaches for surface denoising and smoothing. Several algorithms for surface smoothing use the moving-least-squares approach adopted from signal approximation theory [20]. Since the estimation process is assumed in most cases to be within a two-dimensional domain, robust estimation of the tangent plane is required, leading to the robust moving-least-squares technique [13].
Other methods for surface smoothing are based on diffusion processes on surfaces. These include several algorithms for Laplacian-based mesh fairing [46, 10], where the Laplacian of the coordinate functions is used to define smoothing iterations. Several papers further generalize this approach using higher-order differential operators [8, 41, 12, 47], normal diffusion [45], or different types of curvatures [50].

Many techniques for surface denoising originated from denoising methods in image processing, continuing the extension of signal processing approach for surfaces by [46]. Peng et al. [30] used Gaussian scale mixtures (GSM) on the multiscale coefficients of the surface. Fleishman et al. extended the bilateral filtering to surface smoothing [14]. Yoshizawa et al. [48] adapted the non-local means algorithm [1] for surface smoothing, while using radial basis function (RBF) approximation to overcome the problems associated with matching sampled surface patches. The local patch similarity was further used in urban scenes consolidation algorithm of [51], exploiting the additional structure available in a human-made scene.

A different approach is to denoise the volumetric indicator function, or levelset function of the surface instead of the points themselves. Thus it is possible to apply a variety of image-domain denoising method directly. This was suggested, for example, for the non-local means algorithm by Dong et al. [11], 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 [28], 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. [25]).

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 non-local means algorithm [40, 17], as well as a sparsity-based approach [23]. 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 analyzing the resulting signal group. This approach is also known as nonlocal multipoint modelling [19]. The analysis can be based on spectral [9] or sparsity [24] principles. In a sense, the non-local means uses the mean estimator and can be considered a single-point approach for col-
laborative image denoising [19]. As a similar example, similarity detection has also been strongly related into nonlocal denoising by Berner et al. [4].

The algorithm we describe in this paper builds upon the non-local, multipoint denoising framework, extending the signal processing approach to surface denoising into a collaborative spectral one. Each surface patch is grouped along with similar patches, and these are analyzed together in order to obtain a joint smoothing operator. Denoising over a multitude of such groups results in our final denoised version of the point cloud. The proposed algorithm defines, in a sense, a parallel of the Block-Matching 3D Denoising (BM3D, [9]) 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 denoised signal, based on the above estimate. The result is a high quality regularization, typical of non-local multipoint 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 artifacts.

We describe our model in Section 2, and explain its steps in detail. We demonstrate in Section 3 the results of the proposed method in various noise levels and models. Section 4 concludes the paper.
2 Collaborative Spectral Denoising of Point Clouds

We now turn to describe the notion of collaborative spectral denoising and related concepts. 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 for image denoising, for instance in [1, 9]. Two examples of 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 the 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 the two patches with respect to rigid transformation \((R, t)\),

\[
d(P_i, P_j) = \min_k \|RP_j + t - P_i\|_2, \tag{1}\]

where \(R \in SO(3)\) and \(t \in \mathbb{R}^3\) represent rotation and translation. In our case a robustified point-to-point distance between surfaces was used. Moreover, the patches used for matching are first denoised by a moving least squares in order to improve the similarity measure, as was suggested by Dabov et al. [9] in the context of image denoising.

In order to define a more precise notion of similarity, and to reduce the sensitivity to sampling artifacts in low noise settings, the quadratic distance
approximation [27] may be used instead. Yet another option, suggested by Yoshizawa et al. [48], 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. One should also make note of the type of functions used to describe the surface. In general, we want to perform spectral filtering on relatively regular functions. This is achieved by choosing a suitable support for which we employ the spectral shrinkage, described in the next section, and the local coordinate frame for the patch. The support and the coordinate frame are calculated using the forward-iterative support estimation suggested by Fleishman et al. [13]. An illustration of the support choice appears in Figure 3, at the top-left subfigure. We denote the local coordinates obtained from it by \((\tilde{x}, \tilde{y}, \tilde{z})\).

Once defined, this patch similarity can be exploited to obtain a meaningful transform that can be used to denoise the surface. In order to obtain this transform we look at the spectral decomposition of our patches, calculated using the Laplace-Beltrami operator (LBO) [34]. Specifically, in the same way that wavelets transforms and other dictionaries are used for image denoising and compression, the eigenfunctions of the LBO have found numerous applications. They are used for surface compression [18], analysis [21, 33, 29], recognition [6, 42], invariant representations [37], segmentation [43], surface deformation [5], and so forth [44, 49].

In contrast to previous methods that utilize the 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 \(P_i\) from the sampled surface we construct its collaborative patch, \(\hat{P}_i\), by registering onto it similar patches. In order to use only relevant patches to construct \(\hat{P}_i\), we keep patches that their distance to the patch \(P_i\) after the rigid alignment is smaller than some threshold. This defines the collaborative group for patch \(P_i\),

\[
G_i = \left\{ P_j \quad s.t. \quad \min_{R,t} \|RP_j + t - P_i\|_2 < \tau \right\}.
\]

These point patches are combined into the collaborative patch \(\hat{P}_i\) using the transformation 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 as 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. The following denoising procedure is performed on the collaborative patch in its local coordinate frame ($\tilde{x}, \tilde{y}, \tilde{z}$), as mentioned above.

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, but using fewer groups already gave good results.

Phase I requires defining the discrete Laplacian approximation on the points participating in the collaborative patch. In our experiments, both nearest-neighbor graph-Laplacian approximation and the method of Belkin et al. [3] gave comparable results in noisy point clouds. Furthermore, the spectral analysis can be done instead using wavelet bases constructed on a given point cloud, for example using one of the techniques [32, 7, 38]. Other algorithms can also be used for denoising the collaborative patch. A full comparison of different methods for spectral analysis and regularization of the collaborative point sets is beyond the scope of this paper.

### 2.1 Collaborative Point-cloud Shrinkage

The shrinkage is performed using the eigenfunctions of the Laplace-Beltrami operator of the collaborative patch, forming a spectral domain for functions defined on it. Different eigenfunctions may be thought of as corresponding to different frequencies in this spectral domain [21]. Figure 3 demonstrates the frequency nature of the LBO eigenfunctions.

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

$$f_k = \langle f, \phi_k \rangle,$$

where $\phi_k$ are the eigenfunctions of the Laplace-Beltrami operator, 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

$$S_{\mathcal{P},r}(f) = \sum_{k} \hat{f}_k \phi_k,$$

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

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

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 average these estimators with specifically defined weights. The weights $w_{ji}$ of point $j$ based on the collaborative patch $\mathcal{P}_i$ is

$$w_{ji} = \exp\{-||x_i - x_j||^2/\sigma_D^2\}w_{Q,ji},$$

where, besides the consideration of the distance from the patch center, we add a measure for the quality of the estimation of the patch based on the collaborative group it belongs to. Specifically, 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. We weigh our estimated points inversely proportional to the point density in the collaborative patch in order to avoid the resulting artifacts. In addition, we lower the weight of the estimators in patches that belong to boundary points, using a simple linear estimator for such points. For patches at the boundary, the estimation of local LBO basis is inaccurate, and a simplified model is expected to avoid overfitting. This is similar to the approach taken in the data-dependent moving-least-squares algorithm [22]. Detecting boundaries in point clouds have been reviewed and discussed in the context of manifold learning. We refer the readers to [35] for several approaches and an overview of the topic.

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 3 local coordinate functions. 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 linear coefficients 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 i.i.d. Gaussian noise in the spectral domain (and hence of a uniform PSD). This gives us the theoretical motivation for the shrinkage operator on the collaborative patch.

2.2 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 [9] or in the WienerShrink [15] 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 \( f_{\text{orig}}^k \) denote the spectral coefficients of the original noisy signal, and let \( f_{\text{den}}^k \) denote the coefficients of the denoised estimate obtained from phase I. The Wiener-filtered spectral coefficients are defined as

\[
f_{\text{wien}}^k = \left( \frac{(f_{\text{den}}^k)^2}{(f_{\text{den}}^k)^2 + (f_{\text{orig}}^k - f_{\text{den}}^k)^2} \right) f_{\text{orig}}^k.
\]  

(7)

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 [15]. As in the BM3D algorithm, the different choice of patches at phase II suffice to produce a different transform and ensure the effectiveness of the Wiener-filtering approach to shrinkage.

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

We now demonstrate the results of our approach on several examples. In Figures 6, 7 we show the denoising results on the Fandisk and Bust models. The point clouds were added a Gaussian noise of standard deviations of 0.05 and 0.1 for the Fandisk model, and 0.005 for the Bust model examples, or 1%, 2% and 0.43% of the objects’ diameters, respectively.

Besides visual comparison, we measure the mean-squared-error (MSE), and least-median-of-squares (LMedSq) estimate. We compare our method to the moving-least-squares implementation available in the Point Cloud Library (PCL) [39], as well as an implementation of non-local-means for point clouds. The MSE for point cloud denoising is given by the ICP cost measure

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

(8)

where $S^{GT}$ and $p_i^{den}$ denote the ground-truth (original) surface and the
Table 1: Mean squared error (MSE) and least-median of squares (LMedSq) of the point cloud after denoising for the Fandisk model in Figure 6 for the noise levels shown in the figure.

<table>
<thead>
<tr>
<th>Noise Level</th>
<th>( \sigma = 0.05 )</th>
<th>( \sigma = 0.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>LMedSq</td>
</tr>
<tr>
<td>Noisy surface</td>
<td>2.31 \cdot 10^{-3}</td>
<td>9.14 \cdot 10^{-3}</td>
</tr>
<tr>
<td>NL Means</td>
<td>6.29 \cdot 10^{-4}</td>
<td>2.45 \cdot 10^{-4}</td>
</tr>
<tr>
<td>MLS</td>
<td>4.83 \cdot 10^{-4}</td>
<td>1.52 \cdot 10^{-4}</td>
</tr>
<tr>
<td>Collaborative (phase I)</td>
<td>4.73 \cdot 10^{-4}</td>
<td>1.58 \cdot 10^{-4}</td>
</tr>
<tr>
<td>Collaborative (phase II)</td>
<td>3.83 \cdot 10^{-4}</td>
<td>1.24 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

denoised cloud points, respectively, and \( d(\cdot, \cdot) \) denotes a point-to-surface distance. In our case we used a point-to-point distance, given a dense enough ground truth model, but the quadratic distance approximation [31] is a good candidate if higher accuracy is desired, and the surface is sparsely sampled. Similarly, the LMedSq is the median of the terms summed in Equation 8. While MSE is the classical choice for denoising performance, LMedSq is quite relevant when the denoising is a preprocessing step for robust algorithms which ignore outlier points, as is often the case in ICP algorithms [36], or robust fitting algorithms. These measures for the Fandisk model are shown in Table 1, and demonstrate this proposed algorithm obtains and even surpasses the state of the art in terms or denoising accuracy of point clouds with strong noise. Parameters of all the methods were taken so as to minimize the resulting MSE.

Figure 8 demonstrates the results of our algorithm on data scanned from a structured light scanner. The resulting surface is clearly smoothed in a plausible manner, removing most of the scanning artifacts. 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.

4 Conclusions

In this paper we demonstrated a method for patch-collaborative spectral denoising of surfaces, generalizing the Block Matching 3D Denoising algorithm for image denoising. The suggested method reaches state-of-the-art results in denoising and smoothing of point cloud surfaces, and suggests other possibilities for point cloud denoising that we intend to explore in
<table>
<thead>
<tr>
<th>Noise Level</th>
<th>Model</th>
<th>Bust ( \sigma = 0.01 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>LMedSq</td>
</tr>
<tr>
<td>Noisy surface</td>
<td>1 \cdot 10^{-4}</td>
<td>4.63 \cdot 10^{-3}</td>
</tr>
<tr>
<td>NL Means</td>
<td>1.55 \cdot 10^{-5}</td>
<td>5.02 \cdot 10^{-6}</td>
</tr>
<tr>
<td>MLS</td>
<td>1.46 \cdot 10^{-5}</td>
<td>4.81 \cdot 10^{-6}</td>
</tr>
<tr>
<td>Collaborative (phase I)</td>
<td>1.43 \cdot 10^{-5}</td>
<td>4.21 \cdot 10^{-6}</td>
</tr>
<tr>
<td>Collaborative (phase II)</td>
<td>1.54 \cdot 10^{-5}</td>
<td>4.30 \cdot 10^{-6}</td>
</tr>
</tbody>
</table>

Table 2: Mean squared error (MSE) and least-median of squares (LMedSq) of the point cloud after denoising for the Bust model in Figure 7, for the noise levels shown in the figure.

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

Acknowledgements

The Fandisk models is courtesy of Christian Rössl and Jens Vorsatz of Max-Planck-Institut für Informatik. The authors thank Miki Elad for fruitful and interesting conversations. This research was supported by European Communitys FP7- ERC program, grant agreement no. 267414.
Figure 6: The denoising results for the Fandisk model. Top-to-bottom: results for additive component-wise Gaussian noise with standard deviation $\sigma = 0.05$ and $\sigma = 0.1$. Left-to-right: the noisy model, moving-least-squares result using PCL [39], 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 [16].

References


Figure 7: Left to right: a reconstruction of the Bust model, with added noise of $\sigma = 0.01$, with moving-least-squares results using PCL [39], denoising results by phase I and phase II of the proposed algorithm.


Figure 8: Left-to-right: raw range-scanned data triangulated into a surface, moving-least-squares results using PCL [39], 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.


