

Point Cloud Denoising via Moving RPCA

E. Mattei¹ and A. Castrodad²

¹Contractor

enrico.mattei@gmail.com

²NGA, Springfield, VA, USA

Abstract

We present an algorithm for the restoration of noisy point cloud data, termed Moving Robust Principal Components Analysis (MRPCA). We model the point cloud as a collection of overlapping two-dimensional subspaces, and propose a model that encourages collaboration between overlapping neighbourhoods. Similar to state-of-the-art sparse modelling-based image denoising, the estimated point positions are computed by local averaging. In addition, the proposed approach models grossly corrupted observations explicitly, does not require oriented normals, and takes into account both local and global structure. Sharp features are preserved via a weighted ℓ_1 minimization, where the weights measure the similarity between normal vectors in a local neighbourhood. The proposed algorithm is compared against existing point cloud denoising methods, obtaining competitive results.

Keywords: point cloud, robust PCA, geometry processing, denoising, sparse modelling

ACM CCS: I.3.3 [Computer Graphics]: Computational Geometry and Object Modelling—Curve, surface, solid, and object representations

1. Introduction

Three-dimensional point cloud data (*3D-PCD*) are becoming ubiquitous as low-cost, high-resolution *3D* scanners begin to be commercially available. Additionally, advances in structure-from-motion (*SfM*) and multi-view stereo coupled with massive computational resources are enabling the generation of *3D* reconstructions from large, unstructured image datasets. This enables us to exploit online photo sharing services such as Flickr to create large-scale *3D* models of virtually any location (www.cs.cornell.edu/projects/bigsfm). The resulting *PCD* are used in a variety of fields such as computer graphics, robotics, geoscience, law enforcement and medicine, just to name a few.

As in the case of any type of measured data, *3D-PCD* are inherently noisy. The sources of noise are sensor- and medium-dependent. In the case of laser scanners, noise can be introduced due to imperfect optics, medium characteristics and erroneous range measurements [BLN*13]. In the case of *SfM PCD*, noise is introduced at different stages of the reconstruction algorithm such as incorrect feature correspondences, inaccuracies in the estimation of the camera parameters and errors introduced by bundle adjustments.

Therefore, denoising of the generated *3D-PCD* is of utmost importance to render visually appealing and geometrically accurate data, and to improve the performance in higher level computer vision tasks such as automatic object recognition.

1.1. Main contribution

In this work, we propose a model and a numerical algorithm for *3D-PCD* denoising. The model is intuitively simple and does not require oriented normals. We assume that the point cloud is a collection of overlapping two-dimensional subspaces which naturally collaborate in producing the denoised data. Our approach could be seen as an adaptation of the model proposed in [EA06] for image denoising using sparse modelling techniques, where it is assumed that every image patch admits a sparse representation over a learned dictionary. In the *3D-PCD* setting, we assume that local neighbourhoods in the point cloud live in two-dimensional subspaces.

The proposed algorithm has the following key properties:

1. The method is robust *vis-à-vis* outliers, resulting in accurate and stable estimation of the underlying two-dimensional subspaces.

2. In contrast to other recent approaches, the proposed algorithm does not require oriented normals as part of the input. Instead, the point normals are iteratively computed via feature-aware *RPCA* and re-oriented in a post-processing step.
3. The method is easy to implement and parallelize.

1.2. Previous work

Denoising of *3D-PCD* has been a subject of much attention during the past 20 years. Previous point cloud denoising approaches relevant to our work can be put into one of the following categories: moving least squares (*MLS*)-based [ABCO*01, AK04, GG07, OGG09], particle methods [LCOLTE07, HLZ*09, PMA14], non-local [DG10, Dig12, GAB12], collaborative [RDK13] and sparse modelling methods [ASGCO10, SSW15, XZZ*14].

MLS methods. The *MLS* projection approximates a smooth surface from noisy input samples. The idea of point set surfaces (*PSS*) was introduced in [ABCO*01], where a computational solution was given to project the points onto the *MLS* surface. Each point is projected by (i) finding a local reference domain (by solving a non-linear minimization problem), (ii) defining a function above the local reference domain (by fitting a bivariate polynomial to the neighbourhood data) and (iii) computing the projection of the point by evaluating the estimated polynomial at the origin. Several extensions and modifications have also been proposed, including an explicit surface model and connections with extremal surfaces [AK04], handling of sharp features [FCOS05], as well as models in which the surface is computed as the zero level set of an implicit function approximating a signed distance field from the input points to the surface [AA03, Kol08]. The above methods have several drawbacks. Firstly, they are not robust *vis-à-vis* outliers, and secondly, the projection operators can become unstable with decreasing sampling rate and in the presence of high curvature. Furthermore, convexity of the input shape might not be preserved [AA09].

To address the instabilities in the presence of low sampling rates and high curvature, algebraic *PSSs* (*APSSs*) were introduced in [GG07] by fitting an algebraic sphere to the local neighbourhood instead of the plane fit that is inherent in standard *MLS* procedures. To address the robustness issue, an implicit surface definition was proposed in [OGG09], where local kernel regression is coupled with robust statistics to arrive at the robust implicit *MLS* (*RIMLS*) method. This method, besides being robust to a small number of spatial outliers, is able to preserve sharp features.

Particle methods. This class of techniques aims at producing a set of points, called particles, that represent the underlying surface while enforcing a uniform distribution over the input point cloud. This is done by approximating the ℓ_1 median of local neighbourhoods. Well known approaches are the locally optimal projection (*LOP*) [LCOLTE07], the weighted LOP (*WLOP*) [HLZ*09] and more recently the continuous LOP (*CLOP*) [PMA14] and the anisotropic *WLOP* (*AWLOP*) [HWG*13]. The *CLOP* improves the computational efficiency of *WLOP* by representing the input point set more compactly using Gaussian mixture models, while *AWLOP* achieves sharp feature preservation by modifying *WLOP* to use an anisotropic weighting function.

Non-local similarity and collaborative methods. Approaches in this category are generalizations to *3D-PCD* of the non-local means [BCM05] and the *BM3D* [DFKE07] image denoising algorithms. These methods exploit the self-similarities that exist between small patches in natural images. Extending these approaches to *3D-PCD* is not straightforward, since the methods work in feature space (patches in the case of images), and obtaining neighbourhood descriptors for *3D-PCD* may become challenging due to arbitrary rotations, scaling and low sampling rates. An early attempt at extending the non-local means algorithm to handle *3D-PCD* was presented in [DG10]. In this approach, the authors proposed to use the polynomial coefficients of the local *MLS* surface as neighbourhood descriptors. More recent methods [Dig12, GAB12] employ a scale-space scheme where the point cloud is decomposed at two scales: a coarse scale representing the topological structure, and a fine scale which represents the high-frequency content. Non-local means denoising is applied to the fine-scale data, and then added back to the coarse-scale data to produce the final point cloud.

Another type of non-local approach relies on collaborative filtering of similar neighbourhoods. In [RDK13], the *BM3D* algorithm was extended to *3D-PCD*. For a single neighbourhood, similar neighbourhoods are found by iterative closest point (*ICP*) registration and then combined to form a collaborative group. Then, a spectral decomposition is computed in terms of the eigenfunctions of the Laplace–Beltrami operator (*LBO*), and a hard thresholding is applied to the spectral coefficients. The set of all estimators for each point are averaged to produce the final estimate. In the second stage of the algorithm, a Wiener filter is applied to refine the solution.

Sparsity-based methods. This class of methods is based on the theory of sparse representations. Significant attention is being devoted to extending sparsity-based algorithms to geometry processing problems over recent years, see [XWZ*15] and references therein. For the point cloud denoising task, these methods are based on a two-stage approach. In the first stage, a sparse reconstruction of the point normals is obtained by solving a global minimization problem with sparsity regularization. In [ASGCO10], ℓ_1 regularization is employed, whereas in [SSW15], more aggressive sparsity is sought by employing ℓ_0 regularization. The main assumption is that points that belong to the same smooth region will have similar normal vectors. Therefore, the gradient should be sparse. This is conceptually a total-variation minimization problem. In the second stage, based on the reconstructed normals, the point positions are updated by solving a second global ℓ_1 (or ℓ_0) minimization problem based on a local planarity assumption. Another method making use of sparse modelling techniques was presented in [XZZ*14] to directly estimate a surface mesh from the noisy samples. The approach approximates the input samples by a three-sparse representation over the vertices of the reconstructed mesh. The overall optimization problem involves a $l_{2,q}$ data fitting term ($q \in (0, 1)$) with edge length and possibly normal regularizers. This method is shown to achieve state-of-the-art results in surface reconstruction. However, point set denoising must still be used when the input point set is not dense or the desired output is a ‘cleaner’ representation of the input points.

Paper organization. The remaining of this paper is organized as follows: In Section 2, we briefly review the relevant theory of sparse and low-rank modelling, which constitutes the foundations for our work. We present the proposed approach, moving robust principal

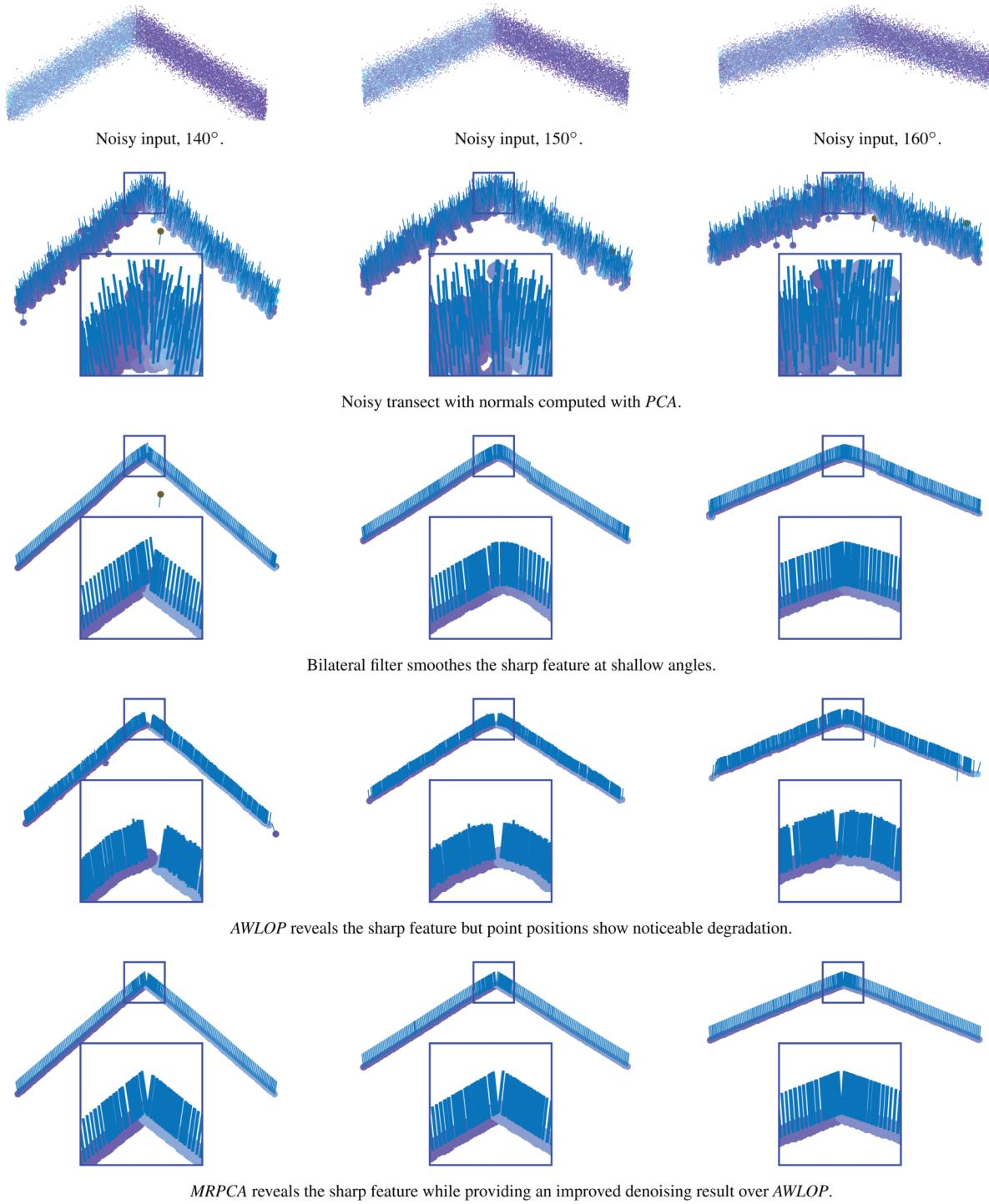


Figure 1: Denoising synthetic datasets of two planes meeting at increasingly shallow angles (20.4K points) with added Gaussian noise of standard deviation equal to 1% of the length of the bounding box diagonal. The two planes meet at an angle of 140° , 150° and 160° . The first and second rows show the noisy 3D data and 2D transects, respectively. Rows 3–5 show the results of the bilateral filter, AWLOP and MRPDA.

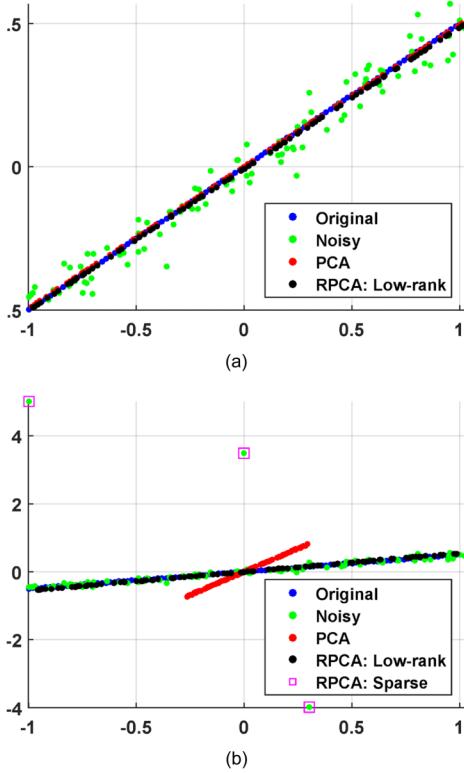


Figure 2: Comparison of classical PCA and RPCA in estimating a 1D subspace (a) without outliers and (b) with three outliers.

components analysis (MRPCA), and describe it in detail in Section 3. Results using publicly available synthetic and real point clouds are provided in Section 4. Finally, in Section 5, we conclude and discuss future research directions.

2. Sparse and Low-rank Models

Sparse models, i.e. models that seek to represent the data as a linear combination of a few elementary signals from a (possibly redundant) dictionary, have enjoyed great success over the last decade due to their simplicity, ease of interpretation and state-of-the-art results in many applications (see [Ela10] and references therein). The straightforward way to find the sparse coefficients of a signal over a dictionary is to minimize the ℓ_0 -pseudo-norm of the coefficient vector. It is well known that such problem is of combinatorial complexity, rendering the approach impractical unless solved via greedy methods. The practical solution to sparse modelling problems was enabled by convex optimization techniques for ℓ_1 -norm minimization, as the ℓ_1 -norm is commonly used as a convex surrogate to the ℓ_0 -pseudo-norm. A typical sparse modelling problem has the following general structure:

$$\underset{\mathbf{D} \in \mathcal{C}, \mathbf{A}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{Y} - \mathbf{DA}\|_F^2 + \lambda f(\mathbf{A}), \quad (\text{P1})$$

where $\mathbf{Y} \in \mathbb{R}^{n \times m}$ is the data matrix, whose columns are signals, $\mathbf{D} \in \mathbb{R}^{n \times k}$ is a dictionary which is possibly over-complete (redu-

dant) and belongs to the set \mathcal{C} , i.e. its columns $\mathbf{d}_i \forall i \in 1, 2, \dots, k$ are constrained to have $\|\mathbf{d}_i\|_2 \leq 1$, $\mathbf{A} \in \mathbb{R}^{k \times m}$ is the matrix of sparse coefficients, i.e. each column of \mathbf{A} contains the sparse representation of the corresponding column of \mathbf{Y} with respect to \mathbf{D} and $f : \mathbb{R}^{k \times m} \mapsto \mathbb{R}$ is a sparsity-promoting function. As it is customary in the literature, a popular function which promotes sparse solutions is $f(\mathbf{A}) = \sum_{i=1}^m \|\mathbf{a}_i\|_1$. The parameter λ controls the trade-off between sparsity and reconstruction error. With this choice for f , problem (P1) belongs to the class of biconvex optimization problems. Therefore, the usual technique to solve (P1) is to find a local minimum by alternating minimization. Note that when \mathbf{A} is fixed, solving for the dictionary can be done using, for instance, a block-coordinate descent algorithm [MBPS09]. Solving for the sparse coefficients while keeping the dictionary fixed amounts to solving m problems of the form

$$\underset{\mathbf{a}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \lambda \|\mathbf{a}_i\|_1. \quad (\text{P2})$$

The solution to (P2) can be obtained in terms of a scalar thresholding defined as $\mathcal{S}_\lambda(x) = \text{sgn}(x)(|x| - \lambda)_+$.

A related type of problem that has regained widespread attention recently is the low-rank modelling problem. Although this type of problem has been known for a long time, it has become a very active research area in recent years due to the work of Candès *et al.* [CR09, CT10, CLMW11]. Consider the problem of recovering a rank- q matrix, with $q < m$, from a noisy observation $\mathbf{Y} = \mathbf{L} + \mathbf{E}$, where $\mathbf{Y} \in \mathbb{R}^{n \times m}$, \mathbf{L} is the low-rank matrix that we wish to recover and \mathbf{E} is a perturbation matrix. It is well known that PCA gives the optimal least-squares solution to this problem. However, a single entry that largely deviates from the true distribution of the data will result in a solution which arbitrarily deviates from \mathbf{L} ; therefore, PCA is not a robust estimator. In [CLMW11], the *principal components pursuit (PCP)* algorithm was proposed to robustify PCA by explicitly accounting for the presence of grossly corrupted entries in the data matrix, i.e. $\mathbf{Y} = \mathbf{L} + \mathbf{O} + \mathbf{E}$, where \mathbf{O} is a matrix of outliers. PCP does not assume that the rank of \mathbf{L} is known *a priori*, and accounts for the outlying entries in \mathbf{Y} based on the assumption that \mathbf{O} has sparse support. The PCP algorithm solves the following optimization problem:

$$\underset{\mathbf{L}, \mathbf{O}}{\text{minimize}} \quad \|\mathbf{L}\|_* + \lambda \|\mathbf{O}\|_1 \\ \text{subject to} \quad \mathbf{L} + \mathbf{O} = \mathbf{Y}, \quad (\text{P3})$$

where $\|\cdot\|_*$ denotes the nuclear norm, which is a convex surrogate for the nonconvex rank functional.

In some applications, it is possible to know the dimension of the signal subspace; therefore, the rank of \mathbf{L} is known. For these cases, the following bilinear formulation of robust PCA (RPCA) was proposed in [MG12]:

$$\underset{\mathbf{U}, \mathbf{S}, \mathbf{m}, \mathbf{O}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{Y} - \mathbf{m}\mathbf{1}^T - \mathbf{US} - \mathbf{O}\|_F^2 + \lambda \|\mathbf{O}\|_{2,c} \\ \text{subject to} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}_{q \times q}, \quad (\text{P4})$$

where $\mathbf{U} \in \mathbb{R}^{n \times q}$ is an orthonormal matrix that spans the signal subspace, $\mathbf{S} \in \mathbb{R}^{q \times m}$ is a matrix whose columns are the principal components, $\mathbf{m} \in \mathbb{R}^n$ is the mean vector, $\mathbf{1} \in \mathbb{R}^m$ is a vector where

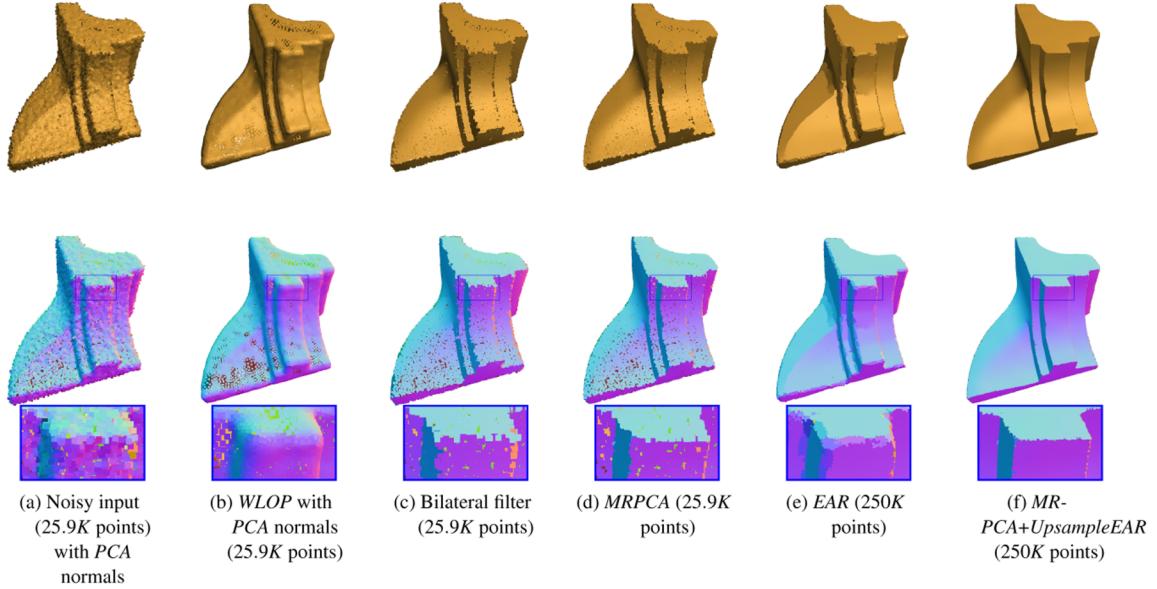


Figure 3: Denoising of the Fandisk model. Various methods are compared. Overall, MRPDA produces the more appealing results.

each element is equal to one and \mathbf{I} is the identity matrix. The desired rank- q matrix is $\mathbf{L} = \mathbf{U}\mathbf{S} + \mathbf{m}\mathbf{1}^T$. In this context, it is assumed that \mathbf{O} is a column-wise sparse matrix, and $\|\mathbf{O}\|_{2,c} = \sum_{i=1}^m \|\mathbf{o}_i\|_2$. The computational solution to problem (P4) is depicted in Algorithm 1, where $\mathcal{S}_{g,\lambda}(\mathbf{x}) = \frac{\mathbf{x}(\|\mathbf{x}\|_2 - \lambda)_+}{\|\mathbf{x}\|_2}$ is the group thresholding operator. $\mathcal{S}_{g,\lambda}$ operates column-wise on matrix-valued arguments. Figure 2 shows a synthetic 2D example comparing classical PCA and RPCA in estimating a line from noisy samples in Figure 2(a), and with noise plus three outliers in Figure 2(b). The non-zero columns of the sparse component identify the outlying samples.

With the above preparations, we now proceed to present the *MR-PCA* algorithm for 3D-PCD.

Algorithm 1 RPCA Solver

```

Input:  $\mathbf{Y} \in \mathbb{R}^{n \times m}$ ,  $\lambda$  (Outlier-sparsity parameter),  $q$  (Desired output rank).
Output:  $\mathbf{U}$ ,  $\mathbf{S}$ ,  $\mathbf{m}$ .
Let  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_q$  be the first  $q$  canonical basis vectors.
Initialization: Set  $\mathbf{U} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_q]$ ,  $\mathbf{O} = \mathbf{0}_{n \times m}$ .
repeat
    Update the mean vector:  $\mathbf{m} = (\mathbf{Y} - \mathbf{O}) \frac{\mathbf{1}_m}{m}$ .
    Centered and outlier compensated data:  $\mathbf{Y}_o = \mathbf{Y} - \mathbf{m}\mathbf{1}^T - \mathbf{O}$ .
    Update the principal components:  $\mathbf{S} = \mathbf{U}^T \mathbf{Y}_o$ .
    Update the orthonormal basis:  $[\mathbf{V}_l, \mathbf{D}, \mathbf{V}_r] = \text{svd}(\mathbf{S}\mathbf{Y}_o^T)$  and set  $\mathbf{U} = \mathbf{V}_r \mathbf{V}_l^T$ .
    Update the outlier matrix:  $\mathbf{O} = \mathcal{S}_{g,\lambda}(\mathbf{Y} - \mathbf{m}\mathbf{1}^T - \mathbf{US})$ .
until convergence or a maximum number of iterations

```

3. Moving RPCA

Our exposition is divided into four parts. First, we state our assumptions and justify the use of a factor analysis model to process local neighbourhoods. Second, we propose a global approach in

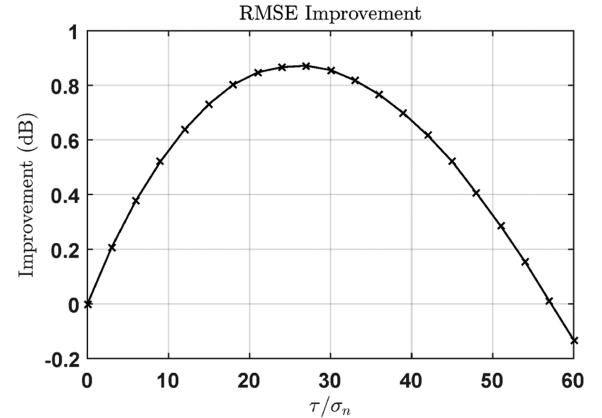


Figure 4: As τ increases, the RMSE improves. After τ reaches a critical value, the RMSE stops improving and deteriorates. Here, the reference anchor model was corrupted with Gaussian noise of standard deviation equal to 0.328.

which overlapping neighbourhoods collaborate to denoise points in the overlapped regions. Third, a numerical solution is presented and its implementation details are discussed. Finally, we provide an analysis of the computational complexity of the proposed algorithm.

3.1. Local, feature-aware modelling

We consider a set of N unstructured and noisy 3D points sampling an underlying piecewise smooth surface \mathcal{S} . These points are organized as columns of a matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N] \in \mathbb{R}^{3 \times N}$. Also, let $\mathbf{R}_i \in \mathbb{R}^{N \times K}$ be the operator that extracts the K -neighbourhood of \mathbf{p}_i , and $\mathbf{Y}_i = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K] \in \mathbb{R}^{3 \times K}$ be the local data matrix, i.e. its

columns are the neighbours of \mathbf{p}_i . We assume that \mathcal{S} can be locally approximated from \mathbf{Y}_i by projecting each \mathbf{y}_j onto the local reference plane. More precisely, let $\mathbf{U}_i = [\mathbf{u}_1, \mathbf{u}_2] \in \mathbb{R}^{3 \times 2}$ be an orthonormal basis for the local reference plane. The normal vector to the local plane is $\mathbf{n}_i = \mathbf{u}_1 \times \mathbf{u}_2$. Relative to \mathbf{U}_i , \mathbf{p}_i has coordinates $(s_1, s_2)^T$. To account for noise and the possibility of outliers in the data, the local neighbourhood is characterized by the following factor analysis model:

$$\mathbf{Y}_i = \mathbf{P}\mathbf{R}_i = \mathbf{U}_i\mathbf{S}_i + \mathbf{m}_i\mathbf{1}^T + \mathbf{O}_i + \mathbf{E}_i, \quad (1)$$

where $\mathbf{S}_i = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K] \in \mathbb{R}^{2 \times K}$ is a matrix whose j^{th} column is the representation of \mathbf{y}_j with respect to the local reference plane, $\mathbf{O}_i = [\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_K] \in \mathbb{R}^{3 \times K}$ is a matrix whose columns represent large deviations from the local distribution of points (outliers), assumed to be column-wise sparse, and \mathbf{E}_i is a Gaussian noise component with bounded energy, i.e. $\|\mathbf{E}_i\|_F^2 < \infty$.

To obtain the projection of \mathbf{Y}_i onto the local reference plane in an feature-aware manner, we solve

$$\begin{aligned} & \underset{\mathbf{U}_i, \mathbf{S}_i, \mathbf{m}_i, \mathbf{O}_i}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{Y}_i - \mathbf{m}_i\mathbf{1}^T - \mathbf{U}_i\mathbf{S}_i - \mathbf{O}_i\|_F^2 + \lambda \sum_{l=1}^K w_{il} \|\mathbf{o}_l\|_2 \\ & \text{subject to} \quad \mathbf{U}_i^T \mathbf{U}_i = \mathbf{I}_{2 \times 2}. \end{aligned} \quad (\text{P5})$$

The weights w_{il} are a function of the point normals in the neighbourhood of \mathbf{p}_i . The goal is to make points with similar normal vectors have a larger influence on the estimate of \mathbf{U}_i , thus similarly to [ASGCO10] we have that

$$w_{ij} = e^{-\theta_{ij}^2/\sigma_\theta^2}, \quad (2)$$

where θ_{ij} is the angle between \mathbf{n}_i and \mathbf{n}_j and σ_θ is a bandwidth parameter. Problem (P5) is solved using Algorithm 1, with a spatially-varying outlier sparsity parameter.

Notice that we could extract $\hat{\mathbf{p}}_i$, the denoised version of \mathbf{p}_i , from the solution of problem (P5) and move on to process the following point. However, the denoised point cloud would consist of a collection of independent local estimates, which will result in a non-smooth solution. Instead, notice that each point will be a member of multiple overlapping neighbourhoods. Then, for each point, we could compute the average of all its independent estimates. This way, we promote collaboration between neighbourhoods in denoising a single point. In the sequel, we formulate a more general model to attack the point cloud denoising problem which ‘globalizes’ the purely local processing described in this section.

3.2. From locally linear estimators to a global solution

We seek a method in which all the independent estimates for a single point are pooled to yield a collaborative estimate for that point. Similar to [EA06], we propose a model which imposes a rank-2 prior on all neighbourhoods of the denoised point cloud. The proposed model naturally leads to a local collaborative estimator which exploits the redundancies present in the overlapping neighbourhoods.

We obtain the denoised point cloud, $\hat{\mathbf{P}}$, as the solution of the following optimization problem:

$$\underset{\hat{\mathbf{P}}, \{\mathbf{L}_i\}_{i=1}^N}{\text{minimize}} \quad \sum_{i=1}^N \|(\hat{\mathbf{P}}\mathbf{R}_i - \mathbf{L}_i)\Omega_i\|_F^2 + \tau \|\hat{\mathbf{P}} - \mathbf{P}\|_F^2, \quad (\text{P6})$$

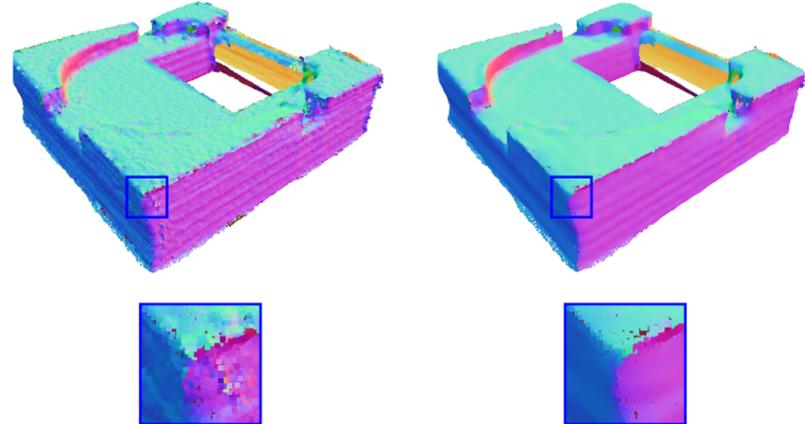
where \mathbf{L}_i is the rank-2 approximation of the local data matrix $\hat{\mathbf{P}}\mathbf{R}_i$, obtained from the solution of problem (P5), and τ is a regularization parameter that controls the trade-off between the denoising ‘strength’ and data fidelity. Each diagonal matrix $\Omega_i \in \mathbb{R}^{K \times K}$ has entries computed using Equation (2).

Algorithm 2 MRPCA Solver

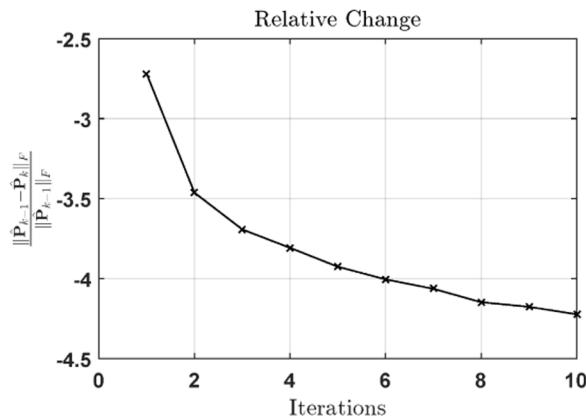
Input: $\mathbf{P} \in \mathbb{R}^{3 \times N}$ (Noisy point cloud), K (Size of local neighborhoods), τ (Data fitting parameter), λ_0 (Base RPCA outlier sparsity parameter), σ_θ (Weight function bandwidth parameter)
Output: $\hat{\mathbf{P}}$ (Denoised point cloud), $\hat{\mathbf{N}}$ (Estimated normals).
Initialization: $\hat{\mathbf{P}}_0 = \mathbf{P}$.
Compute d_{bb} .
 $\lambda = \lambda_0 \times d_{bb}/\sqrt{N}$.
for $k = 1, 2, \dots$ **do**
 Process points (in parallel):
 for $i = 1, 2, \dots, N$ **do**
 $\mathbf{Y}_i = \hat{\mathbf{P}}_{k-1}\mathbf{R}_i$.
 if $k == 1$ **then**
 $\mathbf{w} = \mathbf{1}^T$.
 else
 $\mathbf{N}_i = \hat{\mathbf{N}}_{k-1}\mathbf{R}_i$.
 Compute the neighborhood weights, \mathbf{w} , using \mathbf{N}_i according to Equation 2.
 end if
 $[\mathbf{U}_i, \mathbf{S}_i, \mathbf{m}_i] = \text{RPCA}(\mathbf{Y}_i, \lambda\mathbf{w}, 2)$.
 $\mathbf{L}_i = \mathbf{U}_i\mathbf{S}_i + \mathbf{m}_i\mathbf{1}^T$.
 $\hat{\mathbf{N}}(:, i) = \mathbf{U}_i(:, 1) \times \mathbf{U}_i(:, 2)$.
 $\Omega_i = \text{diag}(\mathbf{w})$.
 end if
 $\hat{\mathbf{P}}_k = (\tau\mathbf{P} + \sum_{i=1}^N \mathbf{L}_i\Omega_i\Omega_i^T\mathbf{R}_i^T)(\tau\mathbf{I}_{N \times N} + \sum_{i=1}^N \mathbf{R}_i\Omega_i\Omega_i^T\mathbf{R}_i^T)^{-1}$.
 end for

3.3. Discussion

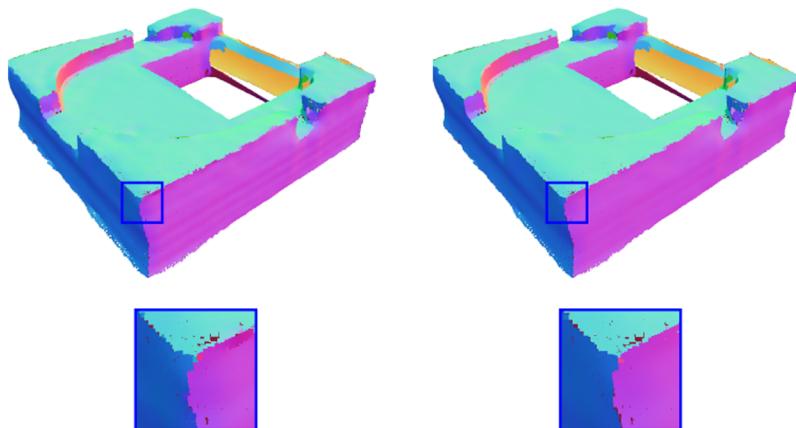
Implementation details. Solving problem (P6) implies that we would need to solve jointly for the denoised point cloud, $\hat{\mathbf{P}}$, and the rank-2 approximations, $\{\mathbf{L}_i\}_{i=1}^N$. We can initialize $\hat{\mathbf{P}} = \mathbf{P}$ and noting that the optimization for the \mathbf{L}_i s is decoupled, solving for the proposed model amounts to independently computing each \mathbf{L}_i , updating $\hat{\mathbf{P}}$ while the \mathbf{L}_i s are kept fixed, and iterating until a local minimum is reached. This iterative procedure is computationally expensive, specially when dealing with very large point clouds. However, since almost all of the computational effort is spent processing each point independently, the algorithm is amenable to parallelization, including implementations on graphics processing units. Additionally, to update $\hat{\mathbf{P}}$ all that needs to be done is to accumulate the \mathbf{L}_i s at the corresponding locations in $\hat{\mathbf{P}}$, and after all the points have been processed, the necessary normalization factor is applied. Therefore, updating $\hat{\mathbf{P}}$ amounts to an averaging operation. The proposed method is depicted in Algorithm 2. An example showing the progression of the algorithm while denoising the Box model is shown in

(a) Noisy input with PCA normals
(222.5K points).

(b) MRPCA after two iterations.



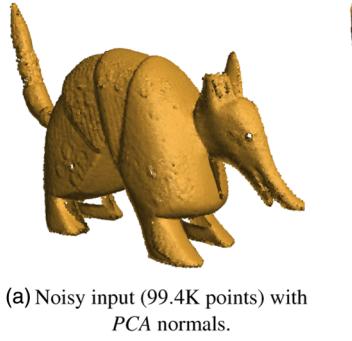
(c) Relative change (in a logarithmic scale).



(d) MRPCA after five iterations.

(e) MRPCA after 10 iterations.

Figure 5: Progression of MRPCA as problem (P6) is being solved. Here, we show normal mappings at 2, 5 and 10 iterations, as well as a plot showing the relative change between consecutive iterates.



(a) Noisy input (99.4K points) with PCA normals.

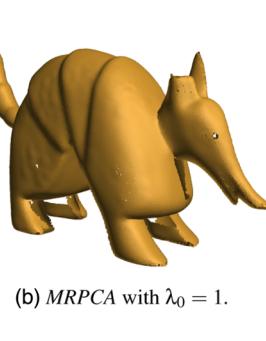
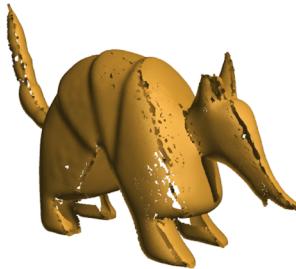
(b) MRPCA with $\lambda_0 = 1$.(c) MRPCA with $\lambda_0 = 0.01$.

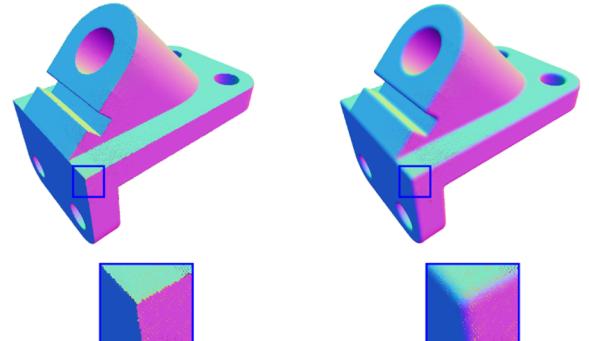
Figure 6: Effect of decreasing λ_0 on the denoising result. As λ_0 decreases, artefacts begin to appear. These artefacts are the result of the outliers component becoming dense, and samples having less influence on the estimation of the \mathbf{U}_i s.

Figure 5. As the algorithm progresses, a piecewise smooth solution is obtained, revealing the sharp features.

The proposed approach can also be understood from the point-of-view of locally linear manifold learning. Since \mathbf{P} are noisy samples from \mathcal{S} , we have that Equation (1) maps points on local 2D affine subspaces to the measured points in \mathbf{P} . The low-rank (rank-2) matrix $\mathbf{L}_i = \mathbf{U}_i\mathbf{S}_i + \mathbf{m}_i\mathbf{1}^T$ represents the local linear structure in \mathcal{S} . Denoising and outlier robustness are achieved as a by-product of minimizing the reconstruction error, and sparsity regularization. Note that this is essentially the first step of the local tangent space alignment (*LTSA*) algorithm [ZZ04]. However, since our ultimate goal is denoising rather than manifold learning, we do not need to compute the alignment matrix or the global 2D coordinates. We thus perform denoising directly on the sample points. The independent estimates are then ‘blended’ together on the overlapping regions during the global iterations.

Parameter selection and tuning. We now give insight on how to select and tune the parameters of the proposed algorithm. We also provide recommended settings that should give good results on a variety of data.

The sparsity parameter, λ , depends on the desired outlier sparsity level and will affect the reconstruction of sharp features. There exist data-driven methods to select this parameter, see [MG12] and references therein. Intuitively, we observe that for point clouds of higher point density, λ can be set to a (relatively) high value, i.e. we can make the outlier component of the *RPCA* model sparser. There-



(a) Clean input (263.3K points).

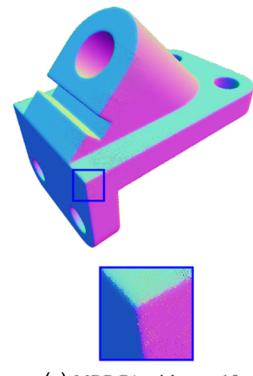
(b) MRPCA with $\tau = 0$.(c) MRPCA with $\tau = 10$.(d) MRPCA with $\tau = 100$.

Figure 7: Effect of increasing τ on the reference anchor model. As τ increases, the smoothing effect decreases.

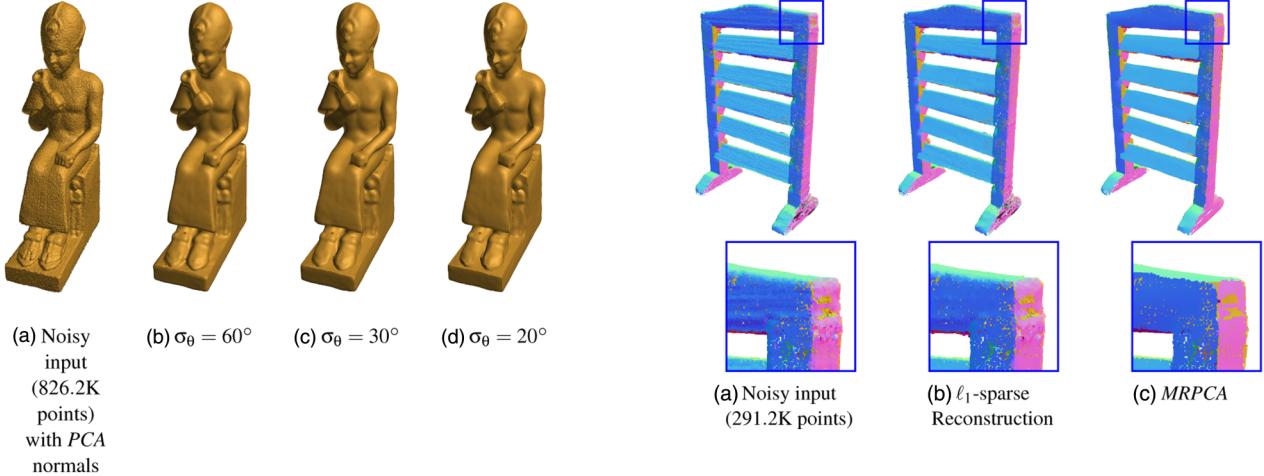
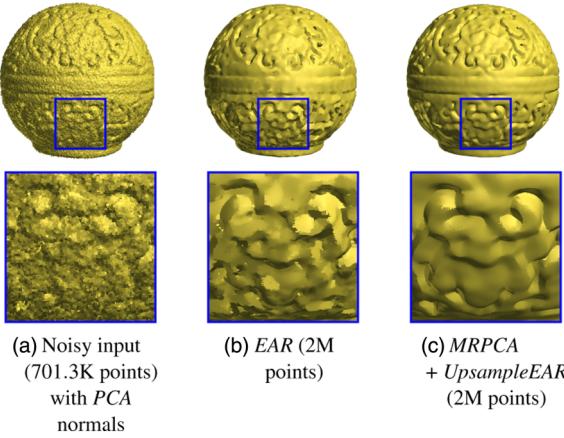
fore, more points in the neighbourhood influence the estimation of the local reference plane. For point clouds of lower point density, selecting λ too low will make the outlier component become dense and the *RPCA* solver can fail to produce an accurate result. Based on these observations we use the following rule, $\lambda = \lambda_0 \times d_{bb}/\sqrt{N}$, where λ_0 is a multiplier and d_{bb} is the length of the bounding box diagonal. This makes λ a function of a rough estimate of the point density [HWG*13]. The effect of varying the value of λ_0 is illustrated in Figure 6. In our experiments, we found that $\lambda_0 = 1$ usually leads to good results.

The number of iterations, t , and the convergence tolerance, tol , of the *RPCA* solver affect the quality of the estimated reference planes and the processing speed of *MRPCA*. We stop the *RPCA* iterations when the change in objective value falls below a user-specified tolerance, or a maximum number of iterations is reached. In general, $t \in \{50, 100\}$ and $tol \in \{1e^{-6}, 1e^{-12}\}$ work well.

The neighbourhood size, K , should be chosen large enough to achieve the desired smoothing effect but small enough to avoid over-smoothing. Alternatively, K can be chosen as a function of the local point density. In the results presented in this paper, we chose a constant K , tuned to achieve visually appealing results. A value of $K \in (30, 100)$ provides good results on a variety of data. However, this parameter should be adjusted per dataset.

Table 1: MRPDA parameters and processing times for the models used in this work.

Model	N	K	λ_0	r	t	RPCA tolerance	σ_θ (degrees)	τ	σ_n	Time (minutes)
Fandisk	25 894	30	1	20	100	$1e^{-12}$	15	$1e^{-6}$	see [ZCL*13]	4.9
Box	222 481	100	1	10	50	$1e^{-6}$	15	$1e^{-6}$	scanning noise	25.7
Armadillo	99 416	30	1	5	50	$1e^{-6}$	30	$1e^{-6}$	scanning noise	2
Anchor	263 286	30	1.0	5	50	$1e^{-6}$	45	100	–	8.5
Ramesses	826 266	100	1	10	100	$1e^{-12}$	20	$1e^{-6}$	0.7162	131.2
Ball	701 322	100	1	3	100	$1e^{-12}$	15	$1e^{-6}$	0.0092	79.4
Face	84 398	30	1.0	10	50	$1e^{-6}$	30	$1e^{-6}$	scanning noise	3.9
Fertility	241 607	30	0.5	3	50	$1e^{-6}$	30	$1e^{-6}$	scanning noise	3.8
Carter	534 512	300	0.35	5	100	$1e^{-12}$	15	$1e^{-6}$	0.0132	139.4
Iron Vise	161 004	100	0.1	20	100	$1e^{-12}$	20	$1e^{-4}$	scanning noise	77.4
Vienna Cathedral	502 156	150	1.0	5	50	$1e^{-6}$	20	0.1	SfM noise	54.1
Shutter	291 220	100	1.0	10	100	$1e^{-12}$	15	0.1	scanning noise	48.9
Octahedron	36 967	100	2	12	100	$1e^{-12}$	15	$1e^{-6}$	see [HWG*13]	6.4

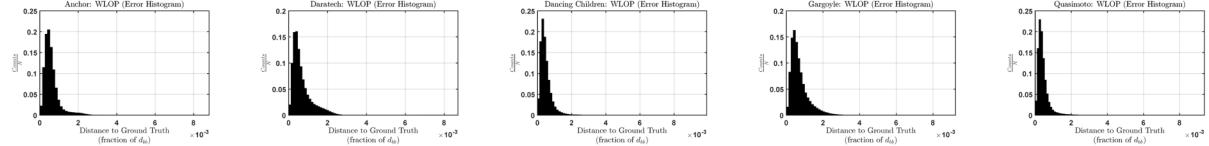
**Figure 8:** Sharpening effect of decreasing the parameter σ_θ . Over-sharpening occurs if σ_θ is set too low.**Figure 9:** MRPDA can improve the results of the EAR up-sampler vis-à-vis AWLOP.**Figure 10:** Denoising of the shutter model. MRPDA is compared to the ℓ_1 -sparse reconstruction method [ASGCO10] (the normal smoothing code is courtesy of Haim Avron).

The parameter of the normal weights, σ_θ , affects the reconstruction of sharp features. This parameter should be chosen such that points in regions with smoothly varying normals have a larger influence in the estimation of the local reference plane. Points with a large normal deviation from the local neighbourhood are promoted to enter the outlier model. Reasonable values are $\sigma_\theta \in (15^\circ, 30^\circ)$.

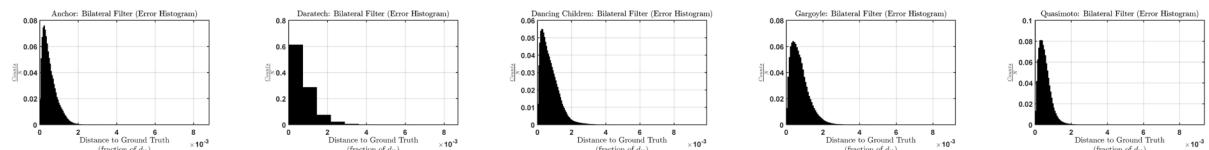
The data fitting parameter, τ , should ideally be chosen as a function of the noise level. Its effect can be better observed by applying the algorithm on a clean dataset, as depicted in Figure 7. Note how the result goes from over-smoothed to becoming almost identical to the input as τ increases. In most of the results presented in this paper, we chose $\tau = 1e^{-6}$ to obtain strong denoising. We found it unnecessary to tune this parameter further as the default value already provided visually appealing results. When the goal is to minimize a performance measure, for example, the RMSE, between a reference point cloud and the denoised result, tuning τ becomes important. As an example, we corrupted the reference anchor dataset



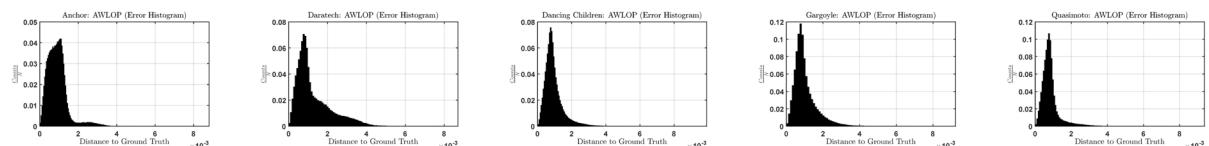
(a) Benchmark dataset corrupted with Gaussian noise of standard deviation equal to 0.2% of d_{bb} . From left to right, the Anchor, Daratech, Dancing Children, Gargoyle and Quasimoto models.



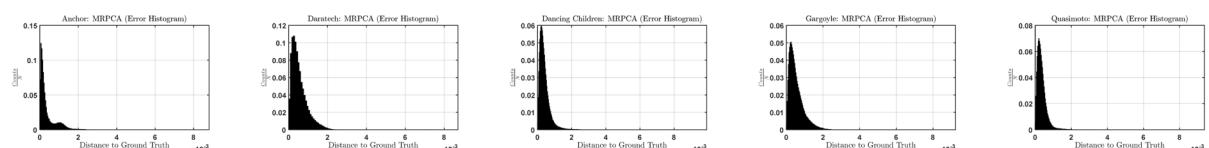
(b) WLOP



(c) Bilateral Filter



(d) AWLOP



(e) MRPCA

Figure 11: Quantitative results on the benchmark dataset of [BLN*13]. We show histograms of errors in point positions relative to the ground truth point cloud. Each method was run with several parameter configurations (following the advice given by the authors), and the results with the lowest RMSE were used to compute the error histograms.

shown in Figure 7 with Gaussian noise of standard deviation equal to 0.3% of d_{bb} . In Figure 4, we show the improvement and degradation of the RMSE as τ varies as a function of the noise level.

The number of MRPCA iterations, r , can be fixed, or automatically determined by monitoring the relative change between successive iterates. In all of the results presented in this paper, MRPCA was run for a fixed number of iterations (see Table 1 for the values used). The value of r should be as small as possible since this parameter will directly impact the running time of the algorithm.

Note that although the recommended settings should give good results for a variety of data, some fine-tuning could be done to get

the best possible results. For example, given a fixed value of σ_θ , λ_0 could be decreased to better reconstruct sharp features or increased to avoid over-sharpening.

3.4. Computational complexity

The computational complexity of the MRPCA algorithm (ignoring weight computation and the local averaging operation) can be deduced as follows. We use a *KD-tree* data structure for the neighbour searches. The cost of building a *KD-tree* is $\mathcal{O}(N \log N)$. The cost of the K -nearest neighbour search for each point is $\mathcal{O}(K \log N)$. For each point, we need to run the *RPCA* solver (Algorithm 1) using its K -neighbourhood. The cost of running the *RPCA* solver

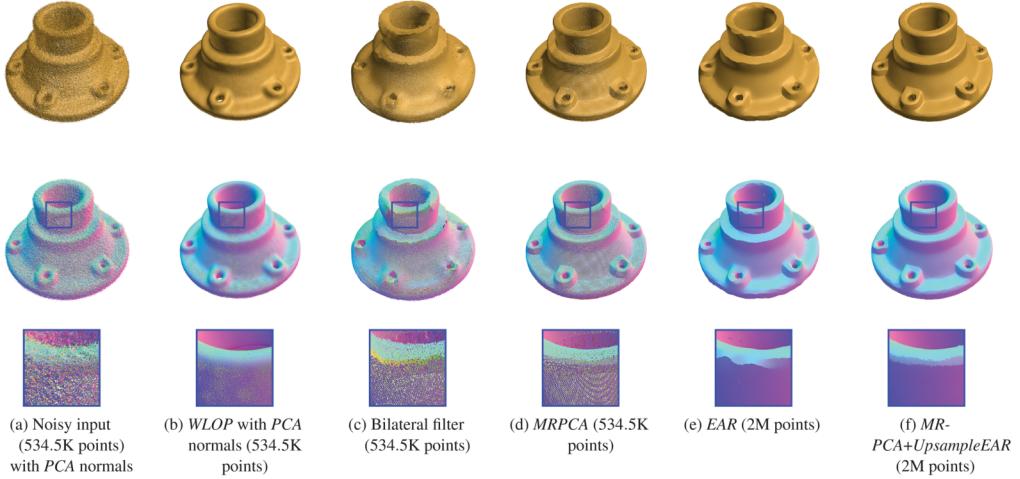


Figure 12: Denoising of the Carter model. *WLOP* produces an over-smoothed result. The bilateral filter has trouble handling errors in the orientation of the initial normals, while *EAR* over-sharpened the edges. *MRPCA* produces a more appealing result and improves the *EAR* up-sampler vis-à-vis *AWLOP*.

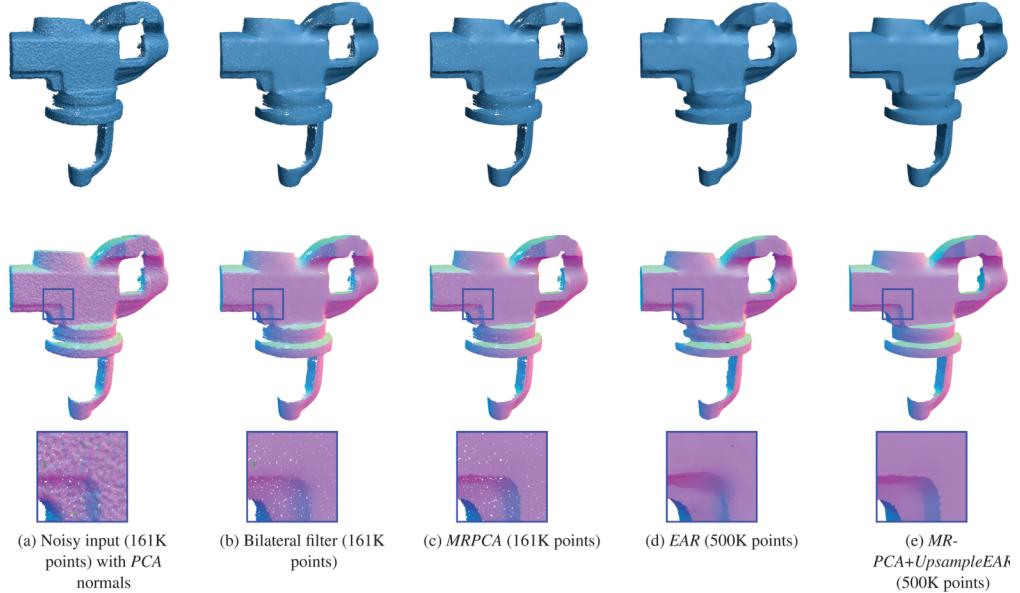


Figure 13: Denoising of the Iron Vise model. The scan shows the natural imperfections obtained by real laser scanners. *MRPCA* achieves better results in complicated regions than the bilateral filter and *EAR*.

for a maximum of t iterations is roughly $t\mathcal{O}(Kd + dq^2)$, where d is the dimension of the data points and q is the dimension of the signal subspace. Noting that $d = 3$ and $q = 2$ are constants, the cost of *RPCA* becomes $t\mathcal{O}(3K + 12) \approx t\mathcal{O}(K)$. Finally, if the *MRPCA* algorithm is run for a maximum of r iterations (a *KD-tree* is built on each iteration), its computational cost will be roughly $r(\mathcal{O}(N\log N) + N\mathcal{O}(K\log N) + Nt\mathcal{O}(K)) \approx r\mathcal{O}(N(\log N + K\log N + tK))$. The running time of the algorithm depends on three parameters (1) the size of the neighbourhoods, (2) the number of iterations specified to run Algorithm 1 and (3) the total number of iterations specified to run the *MRPCA* algorithm

(outer loop in Algorithm 2). We now proceed to present experimental results using *MRPCA*.

4. Results and Limitations

We evaluated *MRPCA* using synthetic and real datasets. We provide comparisons with *WLOP*, and the bilateral filter implemented in CGAL. We also provide comparisons with the original *AWLOP+EAR* re-sampling method. To provide fair comparisons with *AWLOP*, we also run the output of *MRPCA* through the *EAR* up-sampler and explicitly label the results as

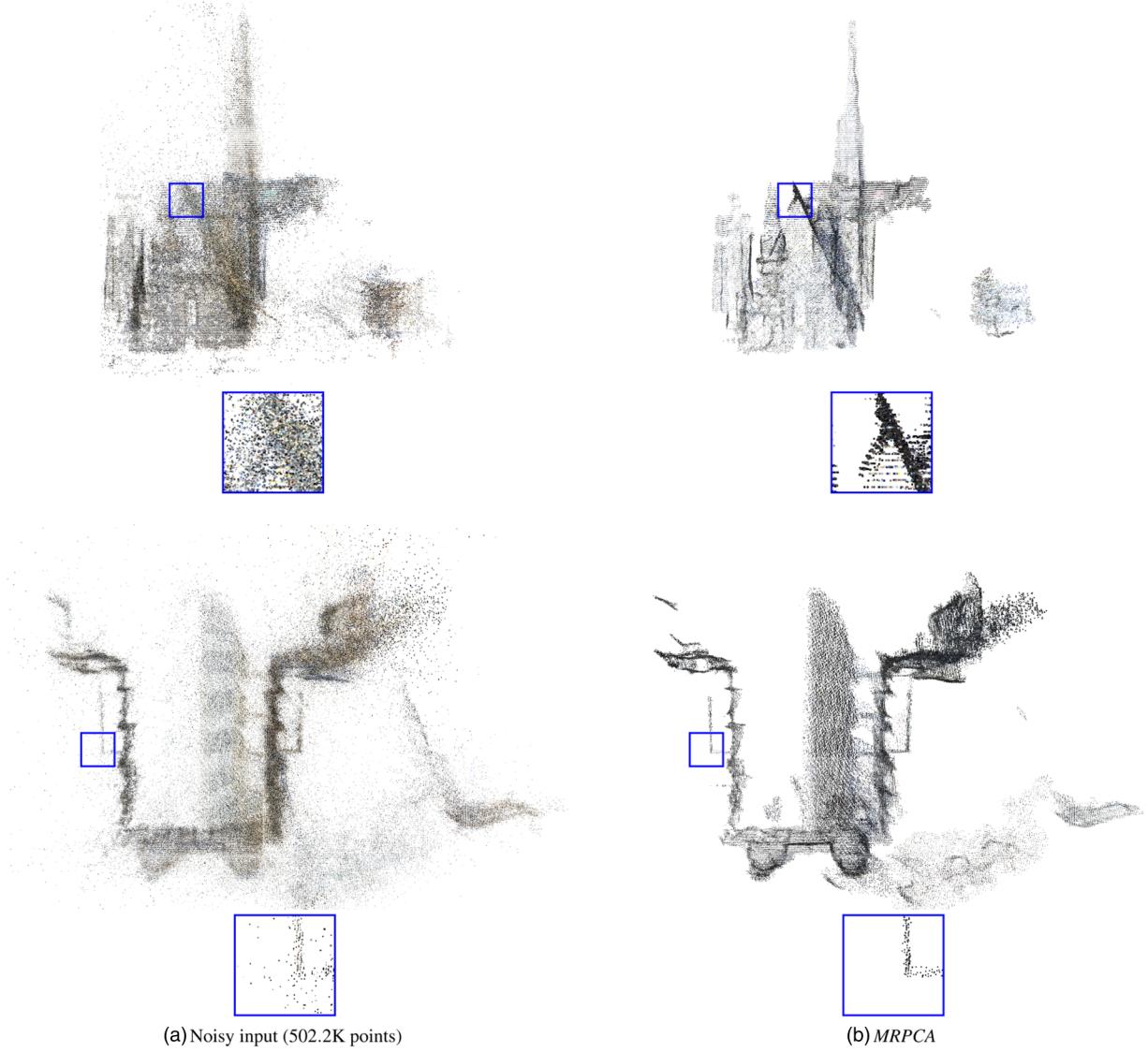


Figure 14: Denoising of the Vienna cathedral SfM model. The noisy input was processed with MRPCA followed by a simple outlier removal method using Meshlab.

MRPCA+UpsampleEAR. In this case, we used the software available at <http://vcc.szu.edu.cn/research/2013/EAR> to run *AWLOP* and used the *EAR* implementation in CGAL to run the up-sampler. For the methods that required initial normals as input, these were computed with a local *PCA*. Additionally, we provide a comparison with the global ℓ_1 -sparse method of [ASGCO10]. In this case, the code to run the normal smoothing step was provided by the corresponding author, and we implemented the point reconstruction step using *CVX*, a package for specifying and solving convex programs [GB14, GB08]. For all methods, we follow the recommendations in the corresponding publication and we show the best results that we were able to obtain. Algorithm 2 is implemented in MATLAB, taking advantage of the Parallel Processing Toolbox. We conducted all experiments on an MSI laptop computer with an Intel Core i7-4700HQ CPU running at 2.40 GHz and 16 GB of RAM.

Results. The parameters used to run MRPCA and timings on all datasets are given in Table 1. To some of the datasets, we added zero-mean *i.i.d.* Gaussian noise with standard deviation, σ_n , given in the table. The noisy data were generated as $\mathbf{P} = \{\mathbf{r}_i + \langle \mathbf{g}_i, \mathbf{n}_i \rangle \cdot \mathbf{n}_i\}_{i=1}^N$, where \mathbf{r}_i is the reference point, \mathbf{g}_i is the additive noise component and \mathbf{n}_i is the normal vector.

In Figure 1, we compare the proposed approach to existing methods in denoising synthetic datasets consisting of two planes meeting at shallow angles. This figure demonstrates that MRPCA can recover these features in the presence of considerable noise. Note that the bilateral filter smooths out the sharp feature as the angle between the two planes increases. *AWLOP* and *MRPCA* clearly preserve the sharp feature, with *MRPCA* producing better denoising performance. In Figure 3, we use the Fandisk model obtained

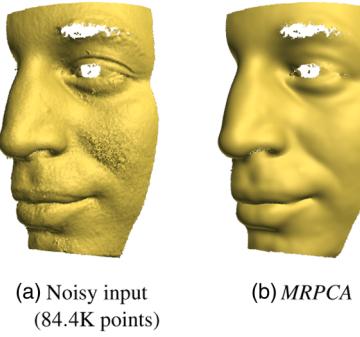


Figure 15: Denoising of the face model with natural imperfections.

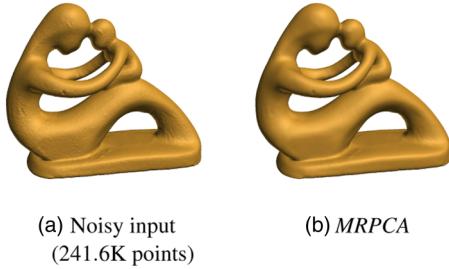


Figure 16: MRPDA handles smooth surfaces without over-sharpening.

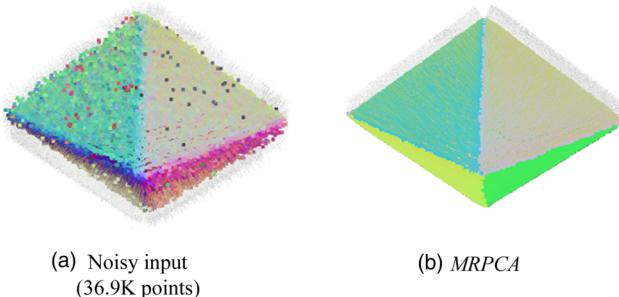


Figure 17: One limitation of the algorithm is that the estimated normals might remain inverted over large contiguous regions even after being consistently re-oriented. Note from (b) that the bottom pyramid has inward facing normals.

from <https://github.com/jjcao/sf-pcd>. This model contains Gaussian noise with standard deviation equal to 50% of the average point spacing. We compare against *WLOP*, bilateral filter and *EAR*. In this particular example, *EAR* preserves the sharp crease better than the other methods and *MRPCA* produces the best overall result. *WLOP* smoothed sharp edges, while the bilateral filter has trouble overcoming orientation errors in the input normals. In Figure 8, we demonstrate the sharpening effect of decreasing the parameter of the Gaussian weights in denoising the Ramesses model. As expected, decreasing σ_θ produces sharper results, but over-sharpening occurs when σ_θ is set too low. We highlight the positive effect that *MRPCA* has on the *EAR* up-sampler in Figure 9. This is a difficult dataset

with a high level of detail. Clearly, *MRPCA* is able to restore complicated structures with higher fidelity than *AWLOP*. A comparison of the proposed algorithm with the global ℓ_1 -sparse method of [AS-GCO10] is shown in Figure 10. The global ℓ_1 -sparse algorithm is not able to obtain an accurate solution for the point normals, which have a negative effect on the estimation of the point positions. In this case, *MRPCA* was able to obtain a better result.

An accuracy analysis of different methods is presented in Figure 11. Here, we compute position error histograms relative to the ground-truth point cloud for *WLOP*, the bilateral filter, *AWLOP* and *MRPCA*. In general, *MRPCA* produced thinner histograms closer to the origin than the other methods. More examples of denoising point clouds with sharp features are presented in Figures 12 and 13. Figure 12 compares *MRPCA* against *WLOP*, the bilateral filter and *EAR*, in denoising a mechanical carter model, while in Figure 13, *MRPCA* is compared against the bilateral filter and *EAR* in denoising the Iron Vise model. In both cases, *MRPCA* (and its corresponding up-sampled version) produces more appealing results. In Figure 14, we provide an example of *MRPCA* denoising of an SfM model. Note how the algorithm is able to perform well even on a very challenging dataset with a large number of outliers. Finally, Figures 15 and 16 are examples of *MRPCA* handling datasets without sharp features. The proposed algorithm is able to handle these cases gracefully, without over-sharpening.

Limitations. Point clouds with sharp features and high noise levels require *MRPCA* to use large neighbourhoods. This considerably increases the running time of the algorithm. We note, however, that this can be alleviated by running the *RPCA* solver with fewer iterations and a more relaxed convergence tolerance while maintaining acceptable accuracy.

Another limitation is that since the normals estimated by the algorithm are obtained via *RPCA*, they are not oriented consistently. This requires the output normals to be re-oriented in a post-processing step. We have observed that in some cases, even after this post-processing, the normals can remain inverted over large contiguous regions as illustrated in Figure 17.

5. Conclusions and Future Work

We have presented a method for denoising 3D-*PCD*, termed as *MRPCA*. This algorithm has been shown to be effective in denoising point clouds with and without sharp features, as well as those produced by an SfM pipeline. The proposed method is based on low-rank and sparse modelling techniques, which are the basis of an increasing body of research in the geometry processing community. We provided a complexity analysis of the proposed algorithm, presented a simple computational solution and showed comparisons against existing state-of-the-art methods. We have shown that *MRPCA* can outperform well-known approaches such as the bilateral filter, *WLOP* and *AWLOP*, having a positive impact on the *EAR* up-sampler. Although *MRPCA* is robust against outliers, this robustness is achieved only locally. One simple modification to achieve global outlier robustness is to use an ℓ_1 data fitting term in problem (P6). Although the ℓ_1 -norm will be able to handle the global outliers better than the Frobenius norm used in this work, the computational cost will increase significantly. We point out that the size of the

neighbourhoods is set globally. One improvement over the current method could be to make the neighbourhood size a function of the local point density. This could have a positive effect when handling datasets with spatially varying noise.

Acknowledgements

The authors would like to thank the anonymous reviewers. Their comments and critique helped improve the presentation and quality of the paper. We are grateful to Dr. Matthew Berger, from the Air Force Research Laboratory. He provided meaningful comments, insight and recommendations since we first communicated with him about this work. Professor Guillermo Sapiro, from Duke University, introduced us to the *MLS* method and provided insightful advice throughout. Professor Haim Avron, from Tel Aviv University, shared the normal smoothing code for the ℓ_1 -sparse reconstruction method. The Carter, Ramesses, Ball and Fertility models are made available courtesy of the AIM@SHAPE-VISIONAIR Shape Repository. Hui Huang provided the Box, Armadillo, Iron and Octohedron models. Matthew Berger, Jie Zhang and Reinhild Preiner provided the benchmark dataset, Fandisk and Face models, respectively. The Vienna Cathedral model was made available by Kyle Wilson.

References

- [AA03] ADAMSON A., ALEXA M.: Approximating and intersecting surfaces from points. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2003).
- [AA09] ALEXA M., ADAMSON A.: Interpolatory point set surfaces—Convexity and hermite data. *ACM Transactions on Graphics* 28, 2 (2009), 20:1–20:10.
- [ABCO] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point set surfaces. In *Proceedings of the Conference on Visualization '01* (2001), IEEE Computer Society.
- [AK04] AMENTA N., KIL Y. J.: Defining point-set surfaces. *ACM Transactions on Graphics* 23, 3 (2004), 264–270.
- [ASGCO10] AVRON H., SHARF A., GREIF C., COHEN-OR D.: ℓ_1 -sparse reconstruction of sharp point set surfaces. *ACM Transactions on Graphics* 29, 5 (2010), 135:1–135:12.
- [BCM05] BUADES A., COLL B., MOREL J.-M.: A non-local algorithm for image denoising. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)* (2005).
- [BLN*13] BERGER M., LEVINE J. A., NONATO L. G., TAUBIN G., SILVA C. T.: A benchmark for surface reconstruction. *ACM Transactions on Graphics* 32, 2 (2013), 20:1–20:17.
- [CLMW11] CANDÈS E., LI X., MA Y., WRIGHT J.: Robust principal component analysis? *Journal of the ACM* 58, 3 (2011), 11:1–11:37.
- [CR09] CANDÈS E. J., RECHT B.: Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9, 6 (2009), 717–772.
- [CT10] CANDÈS E. J., TAO T.: The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory* 56, 5 (2010), 2053–2080.
- [DFKE07] DABOV K., FOI A., KATKOVNIK V., EGIAZARIAN K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing* 16, 8 (2007), 2080–2095.
- [DG10] DESCHAUD J.-E., GOULETTE F.: Point cloud non local denoising using local surface descriptor similarity. In *Proceedings of PCV (Photogrammetric Computer Vision)* (2010).
- [Dig12] DIGNE J.: Similarity based filtering of point clouds. In *Proceedings of 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2012).
- [EA06] ELAD M., AHARON M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing* 15, 12 (2006), 3736–3745.
- [Ela10] ELAD M.: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 2010.
- [FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics* 24, 3 (2005), 544–552.
- [GAB12] GUILLEMOT T., ALMANSA A., BOUBEKEUR T.: Non local point set surfaces. In *Proceedings of 2012 Second International Conference on 3D Imaging, Modelling, Processing, Visualization and Transmission (3DIMPVT)* (2012).
- [GB08] GRANT M., BOYD S.: Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences. V. D. BLONDEL (Ed.), Springer-Verlag Limited, London (2008), pp. 95–110. http://stanford.edu/~boyd/graph_dcp.html.
- [GB14] GRANT M., BOYD S.: CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Transactions on Graphics* 26, 3 (2007), 23:1–23:9.
- [HLZ*09] HUANG H., LI D., ZHANG H., ASCHER U., COHEN-OR D.: Consolidation of unorganized point clouds for surface reconstruction. *ACM Transactions on Graphics* 28, 5 (2009), 176:1–176:7.
- [HWG*13] HUANG H., WU S., GONG M., COHEN-OR D., ASCHER U., ZHANG H. R.: Edge-aware point set resampling. *ACM Transactions on Graphics* 32, 1 (2013), 9:1–9:12.

- [Kol08] KOLLURI, R.: Provably good moving least squares. *ACM Transactions on Algorithms* 4, 2 (2008), 18:1–18:25.
- [LCOLTE07] LIPMAN, Y., COHEN-OR, D., LEVIN, D., TAL-EZER, H.: Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics* 26, 3 (2007), 22:1–22:6.
- [MBPS09] MAIRAL, J., BACH, F., PONCE, J., SAPIRO, G.: Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009).
- [MG12] MATEOS, G., GIANNAKIS, G.: Robust pca as bilinear decomposition with outlier sparsity regularization. *IEEE Transactions on Signal Processing* 60, 10 (2012), 5176–5190.
- [OGG09] ÖZTIRELI, A. C., GUENNEBAUD, G., GROSS, M.: Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum* 28, 2 (2009), 493–501.
- [PMA14] PREINER, R., MATTASCH, O., ARIKAN, M.: Continuous projection for fast l1 reconstruction. *ACM Transactions on Graphics* 33, 4 (2014), 47:1–47:17.
- [RDK13] ROSMAN, G., DUBROVINA, A., KIMMEL, R.: Patch-collaborative spectral point-cloud denoising. *Computer Graphics Forum* 32, 8 (2013), 1–12.
- [SSW15] SUN, Y., SCHAEFER, S., WANG, W.: Denoising point sets via l_0 minimization. *Computer Aided Geometric Design* 35 (2015), 22–15.
- [XWZ^{*}15] XU, L., WANG, R., ZHANG, J., YANG, Z., DENG, J., CHEN, F., LIU, L.: Survey on sparsity in geometric modelling and processing. *Graphical Models* 82 (2015), 160–180.
- [XZZ^{*}14] XIONG, S., ZHANG, J., ZHENG, J., CAI, J., LIU, L.: Robust surface reconstruction via dictionary learning. *ACM Transactions on Graphics* 33, 6 (2014), 201:1–201:12.
- [ZCL^{*}13] ZHANG, J., CAO, J., LIU, X., WANG, J., LIU, J., SHI, X.: Point cloud normal estimation via low-rank subspace clustering. *Computers & Graphics* 37, 6 (2013), 697–706.
- [ZZ04] ZHANG, Z., ZHA, H.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing* 26 (2004), 313–338.