Eurographics Symposium on Geometry Processing (2018)
T. Ju and A. Vaxman (Editors)

Denoising of point-clouds based on structured dictionary learning

K. Sarkar[1,2] and F. Bernard[3,4] and K. Varanasi[2] and C. Theobalt[3,4] and D. Stricker[1,2]

[1]TU Kaiserslautern [2]DFKI Kaiserslautern [3]Max-Planck-Institute for Informatics [4]Saarland Informatics Campus

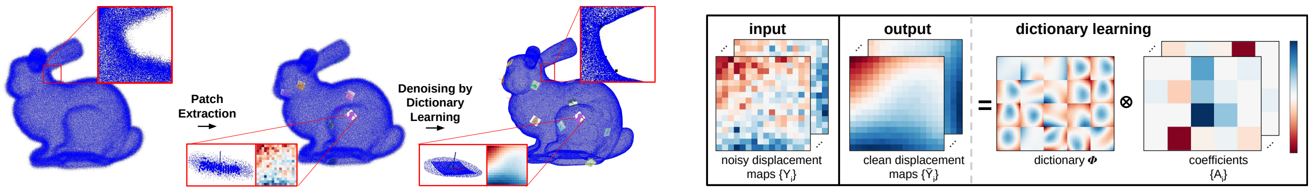

Figure 1: Given a noisy point-cloud, we first extract displacement maps defined over local patches that cover the entire point-cloud, which are then denoised based on dictionary learning in order to obtain the clean displacement maps.

Abstract

We formulate the problem of point-cloud denoising in terms of a dictionary learning framework over square surface patches. Assuming that many of the local patches (in the unknown noise-free point-cloud) contain redundancies due to surface smoothness and repetition, we estimate a low-dimensional affine subspace that (approximately) explains the extracted noisy patches. This is achieved via a structured low-rank matrix factorization that imposes smoothness on the patch dictionary and sparsity on the coefficients. We show experimentally that our method outperforms existing denoising approaches in various noise scenarios.

CCS Concepts

• ***Computing methodologies*** → *Shape analysis;*

1. Introduction

There are various sensors and capturing methods to acquire real-world objects as 3D point-clouds, but typically they are affected by noise. Hence, point-cloud denoising is a necessary step before using such unstructured data in applications. A common (but often oversimplified) assumption is that the noise is Gaussian. Furthermore, point-clouds generally lack a regular structure or topology, which makes their denoising more challenging compared to structured data that is defined over a regular grid, such as images.

In this work we propose a novel method for point-cloud denoising based on a factorization of local square-shaped point-cloud patches. Since the patches are *fixed-length and regular*, we can employ machine learning methods to discover regular features amongst them (Fig. 1). The technical contributions of our method are as follows: (i) We present a method to *robustly* extract a collection of patches covering the entire (given) point-cloud. Unlike previous works, we neither require the patches to be exceedingly small [DCV14], nor do we require an explicit meshing and quadrangulation step [SVS17]. Such restrictive assumptions may be suitable for other contexts like point-cloud compression [DCV14], but are not applicable for denoising. (ii) We propose a formulation of *structured matrix factorization* for 3D patches that handles missing data. (iii) We handle a wide range of noise characteristics, including real-world 3D scanner noise, and demonstrate superior denoising results compared to other methods.

2. Patch-based Point-Cloud Representation

Robust patch extraction: Given a 3D point-cloud $X \subset \mathbb{R}^3$ comprising n points in 3D, we first select a subset of s seed points $S_1, \ldots, S_s \in \mathbb{R}^3$ using a voxel-based downsampling method. For each seed point S_i we extract the patch $P_i = (t_i, R_i, Y_i)$, represented as a triplet, that describes the local structure of the point-cloud within the r-neighborhood $\mathcal{N}_r(S_i) \subseteq X \subset \mathbb{R}^3$ of S_i. Given $\mathcal{N}_r(S_i)$, the location of the center of the patch t_i and its orientation R_i are computed by a RANSAC-like sampling strategy to obtain robust patch locations. In each RANSAC iteration we fit a patch through 3 sample points and consider the *point-to-patch* distance (in contrast to the *point-to-plane* distance) for finding inliers. $Y_i \in \mathbb{R}^{m \times m}$ is the *displacement map* that we explain next.

Displacement map computation: We first represent the neighborhood points $\mathcal{N}_r(S_i)$ of the seed point S_i in a local reference frame that is defined by (t_i, R_i), then we project all points onto the

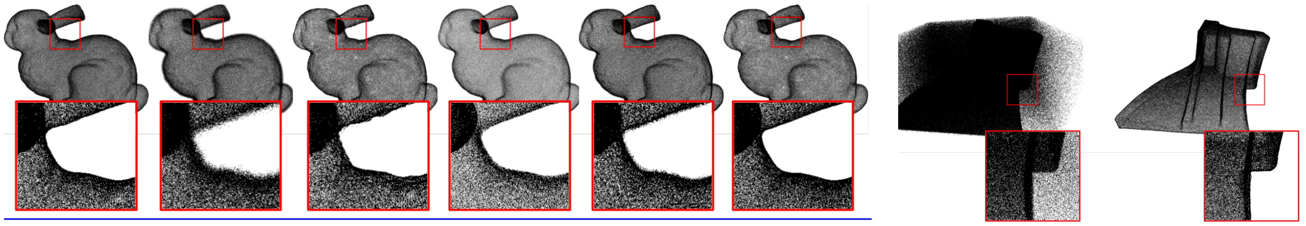

Figure 2: *Left*: Result of our method and a comparison with other denoising methods under Gaussian noise of $\sigma = 0.005$. From left to right: ground truth, noisy, denoised by plane-based method, denoised by [CL17], denoised by [ABCO*03], and denoised by our method. RMS error (relative to bounding box) of the methods are noisy: 4.98e-03, plane-fitting: 3.35e-03, [CL17]: 4.25e-03, [ABCO*03]: 2.58e-03, and ours: **1.15e-03**. *Right*: Result of our denoising algorithm with uniform noise (50% of input points) in addition to Gaussian noise.

square patch, and eventually we sample the patch on an $m \times m$ grid. The displacement value for each bin is then computed as the median of all "z-coordinates" of the points that fall in that particular bin. In addition, for each point, we keep track of the information to which patch and to which bin of each displacement map it falls during the patch computation. This correspondence information is used for the reconstruction of the shapes from the patches.

3. Denoising by Dictionary Learning

Once we have extracted the local patches, we use dictionary learning to denoise them, which we describe next.

Matrix factorization: Let $y_i := \text{vec}(Y_i) \in \mathbb{R}^q$, $q{=}m^2$, be the vectorization of Y_i, and let $\mathbf{Y} = [y_1, \ldots, y_s] \in \mathbb{R}^{q \times s}$ be the matrix that contains the mean-centered vectorized displacement maps. With that, we find a low-dimensional subspace that approximates the noisy $\mathbf{Y}$, which we phrase as the matrix factorization problem

$$\min_{\Phi, A} \ \ell(\mathbf{Y}, \Phi A) + \Omega(\Phi, A)\,, \tag{1}$$

where $\Phi \in \mathbb{R}^{q \times r}$ is the *factor matrix* that contains r dictionary atoms in its columns, and $A \in \mathbb{R}^{r \times s}$ contains the *coefficients* in order to reconstruct $\mathbf{Y}$ using the dictionary Φ. Here, the function $\ell(\cdot)$ is the loss function that measures how well the factorization ΦA approximates the given $\mathbf{Y}$, and $\Omega(\cdot)$ is a regularizer that has the purpose to impose desirable properties upon Φ and A.

Denoising: Due to appealing theoretical properties regarding optimality, as well as a promising performance in various applications (e.g. [BGH*16]), for tackling Problem (1) we build upon the *structured low-rank matrix factorization* framework [HYV14]. We use a binary matrix $M \in \{0,1\}^{q \times s}$ that masks out unobserved data in the displacement map matrix $\mathbf{Y}$, and define the loss function as

$$\ell(\mathbf{Y}, \Phi A) := \|M \odot (\mathbf{Y} - \Phi A)\|_F^2\,, \tag{2}$$

such that the error when approximating $\mathbf{Y}$ using the factorization ΦA is measured in a *weighted* least-squares sense. We choose the regularization term $\Omega(\cdot)$ to impose sparsity on the coefficients A, and to impose spatial smoothness and ℓ_2-regularization upon the dictionary. Our regularizer, motivated by [BGH*16], is given by

$$\Omega(\Phi, A) = \lambda \sum_{i=1}^{r} \|\Phi_{:,i}\|_\phi \|A_{i,:}^T\|_a\,, \tag{3}$$

where for $y \in \mathbb{R}^q$ and $z \in \mathbb{R}^s$ we define the norms as $\|y\|_\phi = \lambda_2 \|y\|_2 + \lambda_E \|Ey\|_2$ and $\|z\|_a = \lambda_1 \|z\|_1$. The matrix E is the incidence matrix of the 4-neighbourhood graph of the $m \times m$ grid of the patch, such that $\|E \cdot\|_2$ accounts for spatial smoothness. Once we have found Φ and A for a given $\mathbf{Y}$, we obtain the estimated matrix of clean displacement maps $\bar{\mathbf{Y}} = \Phi A \in \mathbb{R}^{q \times s}$, which are then used to reconstruct the denoised point-cloud.

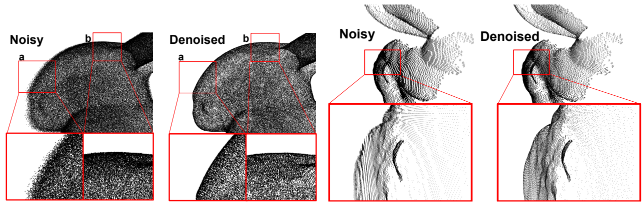

Figure 3: *Left*: Result of our denoising algorithm when the noise is added in the direction of camera. *Right*: Denoising of a point-cloud with noise obtained by a Kinect simulator.

4. Results

Fig. 2 summarizes our denoising results with i.i.d. Gaussian noise and uniform noise. Fig. 3 shows results on directional noise and noise generated by a Kinect simulator.

References

[ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *TVCG* (2003). 2

[BGH*16] BERNARD F., GEMMAR P., HERTEL F., GONCALVES J., THUNBERG J.: Linear shape deformation models with local support using graph-based structured matrix factorisation. In *CVPR* (2016). 2

[CL17] CHENG S., LAU M.: Denoising a point cloud for surface reconstruction. *CoRR abs/1704.04038* (2017). 2

[DCV14] DIGNE J., CHAINE R., VALETTE S.: Self-similarity for accurate compression of point sampled surfaces. *CGF* (2014). 1

[HYV14] HAEFFELE B., YOUNG E., VIDAL R.: Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *ICML* (2014). 2

[SVS17] SARKAR K., VARANASI K., STRICKER D.: Learning quadrangulated patches for 3d shape parameterization and completion. In *3DV* (2017). 1