Quality Point Cloud Normal Estimation by Guided Least Squares Representation

Xiuping Liu[a], Jie Zhang[a], Junjie Cao[a,b], Bo Li[b], Ligang Liu[c]

[a]School of Mathematical Sciences, Dalian University of Technology, Dalian, China.
[b]School of Mathematical Sciences, Nanchang Hangkong University, Nanchang , China.
[c]School of Mathematical Sciences, University of Science and Technology of China, Anhui, China.

Abstract

In this paper, we present a quality point cloud normal estimation method via subspace segmentation based on guided least squares representation. A structure guided low-rank subspace segmentation model has been employed in normal estimation (LRRSGNE). In order to select a consistent sub-neighborhood for a point, the subspace segmentation model is adopted to analyze the underlying structure of its neighborhood. LRRSGNE generates more faithful normals than previous methods but at the price of a long runtime which may take hours. Following its framework, two improvements are proposed. We first devise a novel least squares representation based subspace segmentation model with structure guiding (LSRSG) and design a numerical algorithm which has a natural parallelism for solving it. It segments subspaces as quality as the low-rank model used in LRRSGNE but with less runtime. We prove that, no matter whether the subspaces are independent or disjoint, it generates a block-diagonal solution which leads to a quality subspace segmentation. To reduce the computational cost of the normal estimation framework further, we develop a subspace structure propagation algorithm. Only parts of the candidate feature points' neighborhoods are segmented by LSRSG and those of the rest candidate points are inferred via the propagation algorithm which is faster than LSRSG. The experiments exhibit that our method and LRRSGNE generate comparable normals and are more faithful than other state-of-the-art methods. Furthermore, hours of runtime of LRRSGNE is reduced to just minutes.

Keywords: Normal estimation, Feature preserving, Low-rank representation, Least squares representation, Subspace segmentation

1. Introduction

A tremendous amount of works on point clouds processing and analyzing, such as high quality point based rendering [3, 4], surface reconstruction [5, 6] and anisotropic smoothing [7], benefit from a qualify normal associated with each point. Although several kinds of 3D scanners output normals with point positions simultaneously, more of the ever-broadening range of general digitizing devices are not equipped with normals. Taking the most commonly used laser scanners as an example, points digitized by them are not intrinsically equipped with normals, which have to be estimated from acquired image or geometry data [8]. However, the acquired points are inevitably defect-ridden and normal estimation is sensitive to these detects including noise, non-uniformities, and so on. Hence the computation of quality normals is a challenge especially in the presence of sharp features, *e.g.*, see Fig. 1.

Regression-based normal estimation methods [9, 10, 11, 12] are most widely employed. They use all neighbors of a point to estimate its normal and tend to smooth sharp features. Some robust statistics approaches[13, 14, 1] estimate consistent sub-neighborhoods to compute normals for feature preserving. However the most recently proposed statistics-based method [1] generates unfaithful results for points with variational density near the sharp features, as shown in the top row of Fig. 1. To overcome the sampling anisotropy, Boulch *et al.* [2] design an uniform sampling strategy. However, in the vicinity of sharp features, some erroneous normals may still persist, as shown in Fig. 1. Moreover, the performance of this method drops when the dihedral angel is large. Utilizing the subspace structures of the underlying piecewise surfaces, LRRSGNE [15] selects a consistent sub-neighborhood to estimate qualify normals in the presence of noise and anisotropic samplings. It generates more faithful normals than previous methods but at the price of a long runtime which may take hours. Hence it is impractical to employ it in practice.

In this paper we present a fast and robust approach to estimate normals for point clouds with sharp features. It follows the framework of LRRSGNE with two improvements, which contribute to make it generate quality normals as faithful as LRRSGNE, but with far less runtime. First, the core of LRRSGNE is the neighborhood segmentation via subspace segmentation. It employs the structure guided low-rank representation model (LRRSG), which is a time-consuming non-smooth optimization problem. We formulate the neighborhood segmentation as a least squares representation with structure guiding (LSRSG). A rapid algorithm to solve it is devised and the algorithm has a natural parallelism. Large-scale dataset can be handled efficiently using the parallel implementation. We also prove that LSRSG generates a block-diagonal solution no matter whether the subspaces are independent or disjoint, which leads to a quality subspace segmentation[1]. Second, to reduce the runtime fur-

[1] N subspaces are called independent if and only if $dim(\bigoplus_{i=1}^{N} \mathcal{S}_i) = \sum_{i=1}^{N} dim(\mathcal{S}_i)$, where $\bigoplus$ is the direct sum. Two subspaces are said to be disjoint

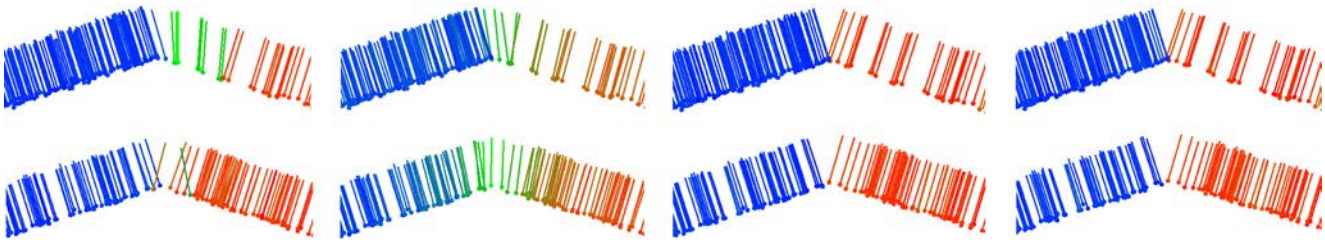


Figure 1: Estimated normals of two planes with a shallow angle. The results of Li *et al.*[1], Boulch *et al.* [2], LRRSGNE, and our method are shown from the first column to the last. The points are sampled non-uniformly in the top row and uniformly in the bottom row. The points and normals are colored according to the normals' direction. Normals consistent with normals of left and right plane are colored in blue and red respectively, and the rest are colored in green.

ther, a subspace structure propagation algorithm is proposed. After analyzing the subspace structures for a small percentage points near sharp features via LSRSG, the rest candidate feature points' sub-neighborhoods are inferred from the previous computed structures. This speeds up the normal estimation significantly and reduces the process from hours to minutes. The contributions of our work are summarized as follows:

- A novel linear subspace segmentation model, LSRSG, is proposed. Even if the subspaces are not independent, it can exactly recover the subspace structure as well as LRRSG with less runtime.
- We prove the effectiveness of LSRSG in theory, and design a rapid numerical algorithm for solving it. The algorithm has a natural parallelism which makes it more suitable for handling the large-scale dataset efficiently.
- Combining LSRSG and the subspace structure propagation algorithm, we devise a fast and robust feature preserving normal estimation method. Comparable normals are estimated in minutes instead of LRRSGNE's hours of runtime and they are more faithful than other state-of-the-art methods.

2. Related work

2.1. Normal estimation

Normals play an important role in surface reconstruction and point rendering. There has been a considerable mount of works on normal estimation. Hoppe *et al.* [9] (PCA) estimate a point's normal by fitting a local plane to all neighbors of it. The method is the pioneer of regression based normal estimation and many variants of it are proposed [16]. Some higher order algebraic surfaces are used to replace planes. The properties of the spherical fitting are exploited by Guennebaud *et al.*[10]. Cazals *et al.*[17] introduce the quadrics fiting to the normal estimation. Pauly *et al.*[18] propose a weighted version of PCA. They assign the Gaussian weights to the neighbors when estimating the local plane. By analyzing local information, such as curvature and noise, Niloy *et al.* [12] find the size of neighborhoods adaptively. For each point, Yoon *et al.*[14] obtain several different normals by generating random subsets of point cloud. Then a ensemble technique is used to combine the several different normals into a single. It is more robust to noise and outliers. However, all these methods fail to correctly estimate normals near sharp features.

Inspired by the feature preserving image filters, methods based on the improvement of preliminary normals are studied. Jones *et al.* [19] derive more faithful normals by 3D bilateral filter. Given a point, Calderon *et al.* [20] select the nearest neighbors belonging to the same plane with it by half-quadratic regularization which takes into account both positions and preliminary normals of the points. By fitting the points and their preliminary normals, [21, 22] define normals as the gradients of locally reconstructed implicit surfaces. Although these methods improve the preliminary normals, estimating the preliminary normals roughly respecting sharp features are necessary.

Another class of methods is based on Voronoi diagram or Delaunay triangulation. For each point, Amenta *et al.*[23] define the normal as the line through it and the furthest Voronoi vertex in its Voronoi cell. But it works only for the noise-free point clouds. By finding big Delaunay balls, Dey *et al.*[24] extend this technology to noisy point clouds. Alliez *et al.*[25] introduce a more stable normal estimation method which combines the advantages of PCA and Voronoi diagram. However, none of these methods are designed for the point clouds with sharp features.

More recently, various works on feature preserving normal estimation are proposed. Hang *et al.*[26] present an interesting combination of point cloud resampling and normal estimation. It is capable of producing accurate normals for the models with noise and outliers. However, the output of this method is a new consolidated point cloud, thus the normals corresponding to the original points are not computed. By maximizing the objective function based on kernel density estimation, Li *et al.*[1] reduce the influence of neighbors lying on different surfaces. It generates quality normals only for the point clouds which are sampled uniformly, since the kernel density estimation is sensitive to the sampling anisotropy. An uniform sampling strategy is proposed by Boulch *et al.*[2] to overcome the problem. How-

if they intersect only at the origin. $\{\mathcal{S}_i\}_{i=1}^N$ are said to be disjoint if every two subspaces are disjoint. Notice that if N subspaces are independent, they are disjoint as well. Hence disjointness is a more general assumption for the subspace set.

ever, this method still fails to correctly estimate the normals for the points extremely near sharp features. Moreover, it tends to smooth out the edges when the dihedral angles are large. Wang *et al.*[27] identify an anisotropic neighborhood via iterative reweighted plane fitting. Three kinds of weight functions related to point distance, fitted residual, and normal difference are considered. However, the estimated normal of a point with a close-by irrelevant surface may be inaccurate. Utilizing the structure of the underlying piecewise surfaces, Zhang *et al.* [15] (LRRSGNE) propose a robust normal estimation method which can recover the sharp features faithfully, even in the presence of noise and anisotropic samplings. However, it is too slow to employ it in practice, since it requires to solve a non-smooth optimization problem for each point near the sharp features. By only solving a linear system for a small percentage candidate feature points and propagating the structure information to the rest rapidly, we design a novel normal estimation method much faster than LRRSGNE.

2.2. subspace segmentation based on Low-rank representation and its variations

The low-rank representation (LRR) is pioneered by Liu *et al.* [28] for the subspace segmentation. Lu *et al.* [29] propose a generalized version of the LRR under the Enforced Block Diagonal conditions and design a least squares regression model for the subspace segmentation. These methods outperform the state-of-the-art algorithms especially when the data is corrupted by noise. Moreover, they prove that LRR and least squares regression model can exactly recover the subspace structures, if the data is drawn from a union of subspaces which are independent. However, they may fail when the assumption is violated. By incorporating a structure guiding item into the LRR, Zhang *et al.* [15] propose LRRSG which provides a practical way to handle more general subspace segmentation problem. It achieves excellent performance in normal estimation. Given a neighborhood of a point near sharp features, they segment it into several sub-neighborhoods by LRRSG. From all the sub-neighborhoods, a consistent one is picked to estimate the normal. The subspace segmentation method, LRRSG, is further introduced into 3D mesh segmentation and labeling by [30]. Tang *et al.* [31] analyze and discuss the effectiveness of this model in theory. In order to improve the efficiency of the algorithm, we relax it to a smooth optimization problem which generates comparable results but with far less runtime.

3. Overview

LRRSGNE [15] actually presents a framework for estimating normals in the presence of sharp features. Generally, we follow the framework. We will go over the general framework and then introduce two improvements which contribute to generate normals as quality as LRRSGNE but with far less runtime.

We assume that the point clouds are sampled from piecewise smooth surfaces and the continuity between these surfaces could be G^0. Sharp features can be considered as sharp edges or corners with G^0 continuity or round edges or corners with very small blending radii [32]. For a point far away from sharp features, its neighborhood may be approximated by a plane. But the neighbors of a point near sharp features are usually sampled from different surface patches across the sharp features, and each of them could be approximated by a plane. Our objective is to identify these planes by subspace segmentation and find a consistent sub-neighborhood enclosing neighbor points sampled from the same smooth surface patch as the point only. Neighbor points on other surface patches are discarded. Then quality normals can be estimated by the consistent sub-neighborhood.

Given a noisy point cloud $\mathcal{P} = \{p_i\}_{i=1}^N$ as input, we take three steps to estimate the normals respecting shape features. First, we detect the points close to sharp features and regard them as candidate feature points. Then the neighborhood of each candidate point may be segmented into several anisotropic sub-neighborhoods. Each sub-neighborhood encloses only the points located on the same surface patch. Finally, we estimate its normal by selecting a consistent sub-neighborhood for the point. The overall procedure is shown in Fig. 2.

The first step, the detection of candidate feature points, follows LRRSGNE. To make our paper self-contained, we give a brief introduction here and details are referred to [15]. For each point p_i, we select a neighborhood $\mathcal{N}_i$ of size S. A weight w_i and a normal n_i are computed by covariance analysis of the local neighborhood. The weight w_i is defined as:

$$w_i = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \tag{1}$$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are the singular values of the covariance matrix of $\mathcal{N}_i$ [18]. The weight w_i measures the confidence of point p_i close to a feature. If w_i is larger than the threshold w_t, p_i is regarded as a candidate feature point, *i.e.* p_i is close to a feature. The threshold w_t is automatically selected by the smoothed distribution of $\{w_i\}_{=1}^N$ [15].

In the second step, LRRSGNE segments the neighborhood of each candidate feature point using LRRSG. However we only segment the neighborhoods of partial candidate feature points. A propagation algorithm is devised to infer the sub-neighborhoods of the rest candidate feature points. The algorithm is described in section 6. Furthermore, the time-consuming LRRSG is replaced by our newly designed LSRSG (see section 4). The theoretical analysis and algorithm for solving LSRSG are introduced in section 5.

Finally, we follow the process of LRRSGNE to estimate normals for both candidate feature points and the rest points. For each non-candidate point, its neighborhood is consistent and the normal n_i is estimated by PCA. For each candidate feature point, utilizing the segmentation of its neighborhood, we select one consistent sub-neighborhood to estimate its normal, which is introduced in section 4.

4. Neighborhood segmentation by LSRSG

Generally, the neighborhoods of points near sharp features are sampled from several surfaces. Each surface can be approximated by a 2D plane of the 3D Euclidean space where

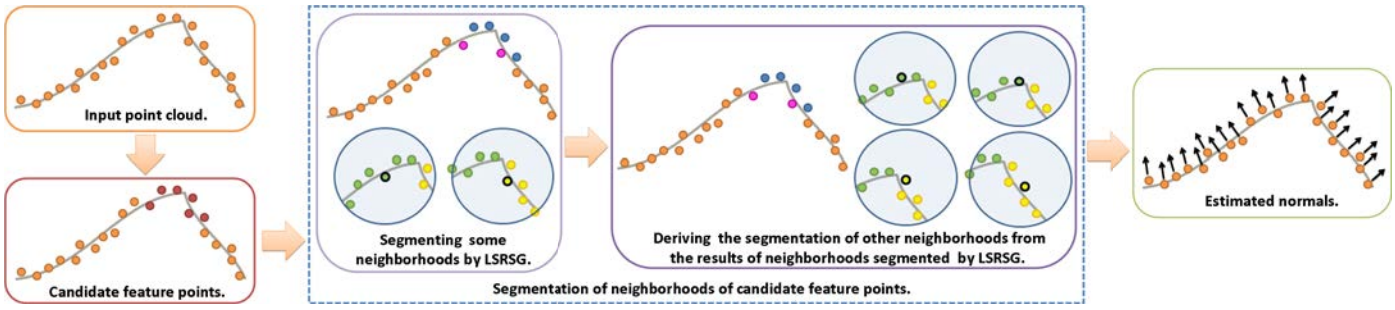


Figure 2: Overview of our method. First, we select the points near sharp features as candidate feature points. Then we classify the neighborhoods of candidate feature points into anisotropic sub-neighborhoods. In order to speed up this process, we segment some neighborhoods by LSRSG and derive the segmentation of other neighborhoods from their results. Finally, the accurate normal of each candidate point is estimated using a selected consistent sub-neighborhood.

the model is embedded. We formulate neighborhood segmentation as a subspace segmentation problem. Given a set of data drawn from a union of multiple subspaces, subspace segmentation aims to group data into segments and each segment corresponds to a subspace. To capture the underlying subspace structure of a neighborhood efficiently and effectively, we propose the least squares representation with structure guiding (LSRSG):

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_F^2 + \beta\|\mathbf{\Omega}\odot\mathbf{Z}\|_F^2 \ \ \mathbf{s.t.}\ \ \|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2 \le \delta, \tag{2}$$

where β, δ are parameters, $\|\cdot\|_F$ is the Frobenius norm and $\odot$ denotes the Hadamard product. $\mathbf{X}$ is the data matrix, each column of which is a sampling. $\mathbf{Z} \in \mathbb{R}^{N\times N}$ is a coefficients matrix, *i.e.* $\mathbf{X}(:,i) \approx \sum_{j=1}^{N} \mathbf{Z}(j,i)\mathbf{X}(:,j)$, where N is the sampling number. Now we will introduce how to segment the neighborhood by this model. The theoretical analysis will be given in the next section.

For a candidate feature point p_i, a larger neighborhood $\mathcal{N}_i^*$ of size S^* is selected. The j-th neighbor point p_i^j of p_i is represented as $\mathbf{x}_j = [x^j, y^j, z^j, n_x^j, n_y^j, n_z^j]'$, where $[n_x^j, n_y^j, n_z^j]$ is its normal computed by PCA and $[x^j, y^j, z^j]$ is local coordinate of p_i^j with p_i as the origin. The data matrix $\mathbf{X}$ is defined as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \cdots, \mathbf{x}_{S^*}]$.

$\mathbf{\Omega}$ is a prior matrix for guiding the segmentation of neighborhood $\mathcal{N}_i^*$ and $\mathbf{\Omega}(i,j) \geqslant 0$. The guiding matrix $\mathbf{\Omega}$ should have the property that the samples from intraclass have smaller weights, whereas the samples from interclass have larger weights. For any two neighbor points p_i^j and p_i^k of p_i, the distance between them is defined as

$$\mathbf{D}_i(j,k) = 1 - |<n^j, n^k>|, \tag{3}$$

where n^j and n^k are the normals of p_i^j and p_i^k, $<\cdot,\cdot>$ represents the inner product of two vectors, and $|a|$ is the absolute value of a. $\mathbf{D}_i(j,k)$ represents the score of points p_i^j and p_i^k belonging to different planes. The value of $\mathbf{\Omega}(j,k)$ is set to 1 if $\mathbf{D}_i(j,k)$ is large enough. In order to improve the reliability of the guiding matrix, two strategies are introduced. Firstly, the neighborhood segmentation starts from the point p_i with smaller w_i since the normals estimated by PCA are reliable for points away from sharp features, and the segmentation results are used to update the guiding matrix of a point computed later. Secondly, when p_i^j or p_i^k is near sharp features, the value of $\mathbf{\Omega}(j,k)$ is decreased. More details are referred to section 4.3.2 of [15].

The optimal coefficient matrix $\mathbf{Z}$ is computed by solving problem (2). The affinity matrix $\mathbf{S}$ is defined as $\mathbf{S} = (|\mathbf{Z}| + |\mathbf{Z}'|)/2$, where $\mathbf{Z}'$ is the transpose of matrix $\mathbf{Z}$ and $|\mathbf{Z}|$ represents a matrix which is defined as $|\mathbf{Z}|(i,j) = |\mathbf{Z}(i,j)|$. Then we segment the neighborhood $\mathcal{N}_i^*$ into several anisotropic sub-neighborhoods by Normalized Cuts [33]. The number of sub-neighborhoods is determined by the iterative segmentation process described in [15]. For each sub-neighborhood, a plane is fitted and then the distance between p_i and the plane is computed. The sub-neighborhood with the minimum distance is identified as the consistent sub-neighborhood of p_i, and an accurate normal is estimated using the consistent sub-neighborhood.

5. Least squares representation with structure guiding

5.1. Basic model

For the sake of analysis, we will discuss LSRSG model on the hypothesis that the data does not contain noise in this subsection. Given m data points $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n] \in \mathbb{R}^{d\times m}$ sampled from n subspaces which compose the space S and $dim(S) = d$. The sample set $\mathbf{X}_i \in \mathbb{R}^{d\times m_i}$ is drawn from subspace $\mathcal{S}_i$ and $\dim(\mathcal{S}_i) = d_i$, $i = 1, 2, \cdots, n$. Our task is to group the data according to the subspaces from which they are drawn.

Based on the observation that each sample $\mathbf{x}_i$ can be represented as a linear combination of other samples drawn from the same subspace, Liu *et al.* [28] propose LRR which is a powerful tool to recover subspace structure. The model is written as

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* \ \ s.t.\ \ \mathbf{X} = \mathbf{X}\mathbf{Z}, \tag{4}$$

where $\|\cdot\|_*$ is the matrix nuclear norm *i.e.* the sum of singular value. Liu *et al.* [28] also prove that LRR obtain a block diagonal solution when the subspaces are independent. This is perfect for segmentation, since when $\mathbf{x}_i$ and $\mathbf{x}_j$ are drawn from different subspaces $\mathbf{Z}(i,j)$ is zero. However, LRR tends to fail when the subspaces are dependent.

To handle more general subspace segmentation problem, Zhang *et al.* [15] propose the low-rank representation with structure guiding (LRRSG):

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \beta\|\mathbf{\Omega}\odot\mathbf{Z}\|_1 \ \ s.t.\ \ \mathbf{X} = \mathbf{X}\mathbf{Z} \tag{5}$$

where β is parameter, $\|\cdot\|_1$ represents ℓ_1-norm. Compared with LRR, it can handle more general subspace segmentation problem and more suitable for neighborhood segmentation [15].

The model (5) can be generalized as:

$$\min_{\mathbf{Z}} f_{FBD}(\mathbf{Z}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{Z}) \;\; s.t. \;\; \mathbf{X} = \mathbf{XZ}. \tag{6}$$

where f_{FBD} and f_S represent arbitrary *favorable block-diagonal function* and *separable function*, respectively.

Favorable block-diagonal function: A matrix function f is regarded as a favorable block-diagonal function, *iff* it satisfying: 1) $f(\mathbf{UMV}) = f(\mathbf{M})$ for all $\mathbf{M} \in \mathbb{R}^{m\times n}$ and all unitary matrices $\mathbf{U} \in \mathbb{R}^{m\times m}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$.
2) for all square matrices A and D, $f(\begin{bmatrix} A & B \\ C & D \end{bmatrix}) \geq f(\begin{bmatrix} A & 0 \\ 0 & D \end{bmatrix})$ and the equality holds if and only if $B = C = 0$.

Separable function: A matrix function f is regarded as a separable function, *iff* for all $\mathbf{A} \in \mathbb{R}^{m\times n}$, $f(\mathbf{A})$ can be represented as:

$$f(\mathbf{A}) = f_0(\sum f_{ij}(|\mathbf{a}_{ij}|)), \tag{7}$$

where $\mathbf{a}_{ij}$ is the (i,j)-th entry of matrix $\mathbf{A}$, f_0 and $f_{ij}, i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$ are increasing functions.

The matrix nuclear norm is a favorable block-diagonal function, the ℓ_1-norm is a separable function, and the square of the F-norm is both a favorable block-diagonal function as well as a separable function. The matrix nuclear norm and ℓ_1-norm are not smooth and it is rather time-consuming to solve Eq. 5. However, the square of the F-norm is smooth. Replacing the matrix nuclear norm and ℓ_1-norm in model (5) with it, we have the LSRSG for data without noise:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_F^2 + \beta\|\mathbf{\Omega} \odot \mathbf{Z}\|_F^2 \;\; s.t. \;\; \mathbf{X} = \mathbf{XZ}. \tag{8}$$

The effectiveness of it and its generalization (6) are guaranteed by the two following theorems [2]:

Theorem 1: If $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_n$ are independent, the optimal solution to the model (6) is a block-diagonal matrix

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1^* & 0 & \cdots & 0 \\ 0 & \mathbf{Z}_2^* & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Z}_n^* \end{bmatrix}, \tag{9}$$

where $\mathbf{Z}_i^*$ is a $m_i \times m_i$ matrix.

Theorem 2: Denote $\mathbf{x}_i$ as the i-th sample from $\mathbf{X}$. Since both $\mathbf{z}_{ij}$ and $\mathbf{z}_{ji}$ denote the affinity between the sample $\mathbf{x}_i$ and $\mathbf{x}_j$, it is natural to suppose that $\mathbf{Z}$ is symmetric. If $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_n$ are disjoint and

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{A}_1^* & \mathbf{B}_1^* & \mathbf{C}_{1,3}^* & \cdots & \mathbf{C}_{1,n}^* \\ \mathbf{B}_1^{*T} & \mathbf{A}_2^* & \mathbf{B}_2^* & \cdots & \mathbf{C}_{2,n}^* \\ \mathbf{C}_{1,3}^{*T} & \mathbf{B}_1^{*T} & \mathbf{A}_3^* & \vdots & \mathbf{C}_{n-2,n}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{1,n}^{*T} & \mathbf{C}_{2,n}^{*T} & \mathbf{C}_{3,n}^{*T} & \cdots & \mathbf{A}_n^* \end{bmatrix}, \tag{10}$$

[2]The proofs are presented in Appendix A.

Table 1: Computation time for the toy examples.

	LRR	LRRSG	LSRSG
Two planes	2.31	2.23	0.44
One plane & two lines	2.54	2.32	0.48

where the elements of $\mathbf{A}_i^* \in \mathbb{R}^{m_i\times m_i}$ and $\mathbf{B}_i^* \in \mathbb{R}^{m_i\times m_{i+1}}$ are all zeros, whereas the elements of $\mathbf{C}_{i,j}^* \in \mathbb{R}^{m_i\times m_j}$ are all ones, there exists β, which makes the optimal solution of model (6) to be a block-diagonal matrix.

Theorem 1 shows that LSRSG achieves the same conclusions as those of LRR when the subspaces are independent. Theorem 2 means that with some predefined guiding matrix, LSRSG can exactly segment multiple disjoint subspaces which is more challenging and can not be handled by LRR.

5.2. Robust model

Since, in most practical cases, $\mathbf{X}$ is corrupted by noise, certain relaxation to the equality constraint in model (8) is desirable:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_F^2 + \beta\|\mathbf{\Omega} \odot \mathbf{Z}\|_F^2 + \lambda\|\mathbf{X} - \mathbf{XZ}\|_F^2, \tag{11}$$

where $\lambda > 0$ is a parameter determined by the noise-scale. From optimization theory, it is well known that problems (2) and (11) share the same solution [34]. Here we consider solving the unconstrained convex optimization (11). The solution of it is the point where the derivative is zero:

$$\begin{aligned} 2\mathbf{Z}^* + 2\beta(\mathbf{\Omega} \odot \mathbf{Z}^*) - 2\lambda(\mathbf{X}^T(\mathbf{X} - \mathbf{XZ}^*)) = 0, \\ \mathbf{Z}^* + \beta(\mathbf{\Omega} \odot \mathbf{Z}^*) - \lambda(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{XZ}^*) = 0, \\ (\mathbf{I} + \mathbf{X}^T\mathbf{X})\mathbf{Z}^* + \beta(\mathbf{\Omega} \odot \mathbf{Z}^*) = \lambda(\mathbf{X}^T\mathbf{X}), \end{aligned} \tag{12}$$

where $\mathbf{Z}^*$ is the optimal solution to problem (11) and $\mathbf{I}$ is an identity matrix. When only see the j-th column of matrix $\mathbf{Z}^*$, we have:

$$\begin{aligned} (\mathbf{I} + \mathbf{XTX})\mathbf{Z}^*(:, j) + \beta(diag(\mathbf{\Omega}(:, j))\mathbf{Z}^*(:, j)) \\ = \lambda(\mathbf{XTX}(:, j)), \end{aligned} \tag{13}$$

where $\mathbf{XTX} = \mathbf{X}^T\mathbf{X}$, $\mathbf{A}(:, j)$ is the j-th column of matrix $\mathbf{A}$, and $diag(\mathbf{a})$ is a diagonal matrix with the elements of $\mathbf{a}$ on the main diagonal. Therefore the solution of problem (11) is:

$$\begin{aligned} \mathbf{Z}^*(:, j) = \lambda(inv(\mathbf{I} + \mathbf{XTX} + \beta diag(\mathbf{\Omega}(:, j)))\mathbf{XTX}(:, j)), \\ j = 1, 2, \cdots, S^*, \end{aligned} \tag{14}$$

where $inv(\mathbf{A})$ represents the inverse matrix of $\mathbf{A}$. The columns of $\mathbf{Z}^*$ are computed by Eq. 14 independently, therefore this solving process is easy to be implemented in parallel.

5.3. Toy examples

Some toy examples are provided to verify the effectiveness and efficiency of LSRSG. Some data are drawn from several disjoint subspaces and segmented by LRR, LRRSG and LSRSG; see Fig. 3. LRRSG and LSRSG use the same guiding matrix $\mathbf{\Omega}$ which is constructed by forty percent prior knowledge with 30% errors. Firstly, an ideal full guiding matrix $\mathbf{G}$ is built.

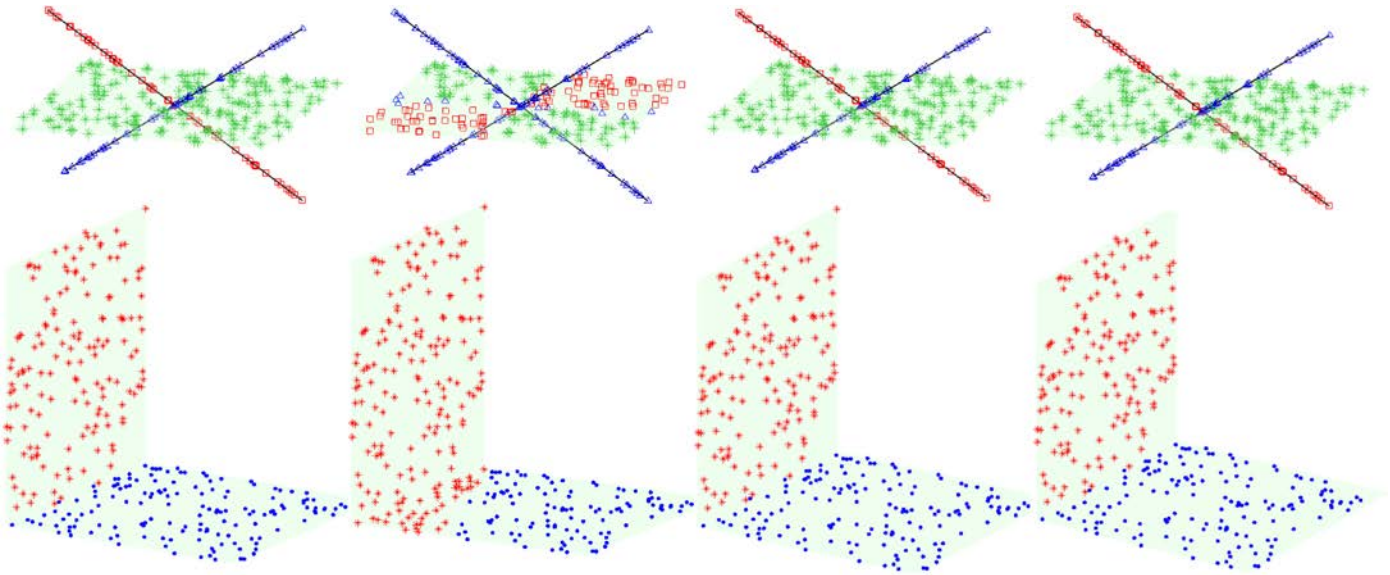


Figure 3: Segmentation results of LRR, LRRSG and LSRSG. The first column is the input data. The segmentation results of LRR, LRRSG and LSRSG are shown from the second column to the last.

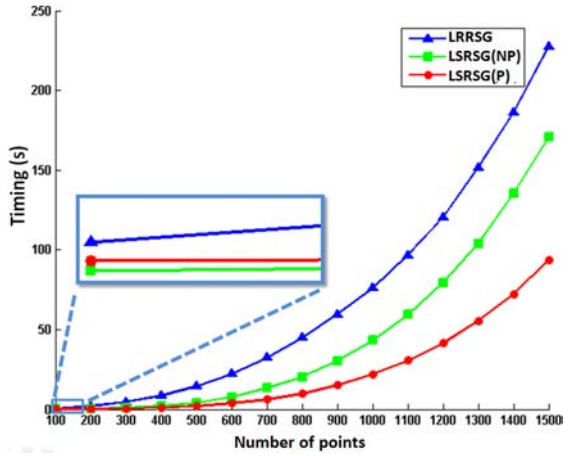


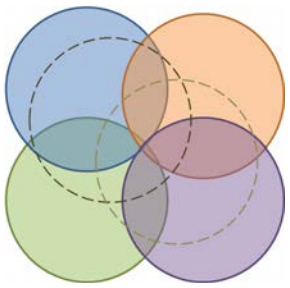
Figure 4: Computation time of LRRSG, LSRSG(NP), and LSRSG(P) with varies number of data.

Specifically, if two samples p_j and p_k are in the same subspaces, $\mathbf{G}(i, j) = 0$, otherwise $\mathbf{G}(i, j) = 1$. The guiding matrix is generated by choosing 40% elements of $\mathbf{G}$ randomly and the other elements of $\mathbf{\Omega}$ are all set to zeros. It is further corrupted by randomly selecting 30% elements from the 40% elements and switching their values, *i.e.* from ones to zeros (or zeros to ones) if they are ones (or zeros) originally. As illustrated in Fig. 3, LRR fails to segment dependent subspaces that is consistent with previous analysis in above subsections. Benefitting by the partial prior knowledge, even with considerable levels of errors, LRRSG and LSRSG segment the data faithfully. The computing times of the two methods are shown in Tab. 1. These results suggest that LSRSG performs as well as LRRSG, but spends less time.

When solving LRRSG, each iteration relies on the result of last iteration. However, the column vectors of the solution of LSRSG are computed independently. Therefore, the solving process of LSRSG is easy to be implemented using parallel running strategy. For convenience, we use LSRSG(P) and LSRSG(NP) to denote the LSRSG implemented in parallel and non-parallel version, respectively. The timings of LRRSG, LSRSG(NP) and LSRSG(P) are shown in Fig. 4. The data is uniformly sampled on two 2-dim subspaces and the number of data varies from 100 to 1500. This experiment is implemented in Matlab and performs with 4 CPU Intel(R) Xeon(R) 2.53 GHz. The solving of LRRSG consumes more time than that of LSRSG whether or not the process is implemented in parallel. As the zoomed view of Fig.4 shows, LSRSG(NP) is more faster than LSRSG(P) when the number of data is small. That is because the integration and separation of variables in parallel computing takes a part of the time. When the number of data is large, such cost is negligible. LSRSG(P) achieves lower computational costs. This means that we can use this approach to further increase the efficiency when handling the large-scale dataset.

6. Neighborhood segmentation by propagation

Although the solving of LSRSG takes less time than LRRSG, employing it for each candidate feature point is still somewhat expensive. Actually the neighborhoods of candidate feature points are overlapping. Inferring the segmentation of a neighborhood from computed results of the neighborhoods overlapping with it, the runtime will be significantly reduced. As the wrapped figure shown, if the



four neighborhoods represented by circles with solid borders have been segmented by LSRSG, we want to infer the segmentations of the neighborhoods marked by circles with dashed borders. In this section, we design a subspace structure propagation algorithm to accomplish this objective.

First, we construct two matrices $\mathbf{R}$ and $\mathbf{N}$ to store the segmentation results obtained by LSRSG. The values of $\mathbf{R}(i, j)$ and $\mathbf{N}(i, j)$ represent the number of p_i and p_j grouped into the same subspace and into different subspaces, respectively. To compute $\mathbf{R}$ and $\mathbf{N}$, a small percent candidate feature points are extracted and their neighborhoods are segmented by LSRSG. We denote the selected candidate feature points by $T = \{t_1, t_2, \ldots, t_k\}$, where $k = N \times r$ is the number of points selected, $r \in [0, 1]$. The selected points T are expected to cover the candidate feature points well and the estimated subspace structures around them should be as faithful as possible. Since the points with larger w_i (problematic points) are usually near a complex structure which makes the segmentation more challenging, it is better to select none of them. On the contrary, only choosing the points with smaller w_i may un-cover the problematic points. Therefore, we rank the candidate feature points by

$$w_i^d = |w_i - w_{ave}|, \tag{15}$$

where w_{ave} is the average w of all the candidate feature points. The points with smaller w_i^d are selected.

Given one of the rest candidate feature points p_i and its neighborhood $\mathcal{N}_i^*$, we select a smaller neighborhood $\mathcal{N}_i^s$ with size S^s. The segmentation of the neighborhood $\mathcal{N}_i^*$ is inferred by the subspace structure propagation algorithm, if the current point p_i and its neighborhood $\mathcal{N}_i^*$ are well covered by T. Specifically, if $\mathcal{N}_i^s \bigcap T = \emptyset$, we add the point p_i into T, segment $\mathcal{N}_i^*$ by LSRSG and modify $\mathbf{R}$ and $\mathbf{N}$. Otherwise, $\mathcal{N}_i^*$ is segmented by the propagation algorithm presented as follows. We find a seed point p_j from $\mathcal{N}_i^*$ with the smallest w and initialize a set $Q = \{p_j\}$ representing the points in the same plane with p_j. The next step is to iteratively add points to the set Q one by one. At each iteration, we chose the point having the largest *relation* with Q. The *relation* $REL_{l,Q}$ between point p_l and set Q is defined as:

$$REL_{l,Q} = \max_{p_j \in Q}(rel(p_l, p_j)), \tag{16}$$

$$rel(p_l, p_j) = \begin{cases} -1, & if \;\; N(l, j) > 0 \\ R(l, j), & if \;\; N(l, j) = 0 \end{cases}. \tag{17}$$

This process is terminated until $REL_{l,Q} \leq 2$ or $|Q| > 0.7 \times S$, where $|Q|$ denotes the cardinality of the set Q. Then, for the rest of p_i's neighbors $\mathcal{N}_i^* = \mathcal{N}_i^* \setminus Q$, we repeat the process until $\mathcal{N}_i^*$ is empty. If a neighbor p_j is not belong to any previous segmented neighborhoods, *i.e.* no structure information about it is stored R and N, it is ignored and deleted from the current neighborhood $\mathcal{N}_i^*$. Thus, $\mathcal{N}_i^*$ is segmented into some sub-neighborhoods $Q_1, Q_2, \cdots$.

7. Results

To evaluate the performance of our approach, a variety of point clouds with sharp features and synthetic Gaussian noise are tested. The deviation is defined as a percentage of average distance between points. We compare our method with some classic and state-of-the-art methods: PCA [9], robust normal estimation (RNE) [1], hough transform (HF) [2], and LRRSGNE [15]. According to the sampling strategy, HF has three versions: HF_points, HF_cubes, and HF_unif.

The Root Mean Square (RMS) measure which has been used in [15, 2] is introduced to quantitatively analyze the results. It is defined as:

$$RMS_{-\tau} = \sqrt{\frac{1}{|\mathcal{P}|}\sum_{p \in \mathcal{P}}(f(n_{p,ref}\hat{n}_{p,est}))}, \tag{18}$$

where

$$f(n_{p,ref}\hat{n}_{p,est}) = \begin{cases} n_{p,ref}\hat{n}_{p,est}, & if \;\; n_{p,ref}\hat{n}_{p,est} < \tau \\ \pi/2 \quad , & otherwise \end{cases}, \tag{19}$$

$n_{p,ref}$ and $n_{p,est}$ are the reference and estimated normals at p, respectively. As proposed by [15, 2], we take $\tau = 10$ degrees and regard the points with the measure greater than τ degrees as bad points. In this section, all experiments have been performed with 2 CPUs Inter(R) Core(TM) i5-3230M 2.60GHZ. We quantitatively analyze the estimation results of the data with synthetic noise: a centered Gaussian noise with deviation defined as a percentage of average distance between points.

The parameters of our algorithm are summarized below:

- S: the number of neighbors used to PCA.
- S^*: the number of neighbors used to segmentation.
- S^S: the number of neighbors used to compute the overlap with T.
- r: the percentage of neighborhoods segmented by LSRSG.
- λ and β: parameters used to balance the three items in Eq. 11.

The choice of S and S^* depends on the noise. Since only part neighbors are used to estimate the normals when points are near sharp features, S^* should be larger than S. Parameter S^S is used to guarantee that for each neighborhood segmented by propagation algorithm, there is enough points recovered by the neighborhoods of T. A smaller value of it represents more recovered points but higher computational costs. The larger the value r is, the more neighborhoods segmented by LSRSG, which represents more accurate normal estimation results and higher computational costs. If the noise ia large, we should relax the fitting restriction and decrease the value of λ and increase the value of β. In our implementation, S, S^*, S^s, r, λ, and β are selected as $S = 70$, $S^* = 120$, $S^s = 30$, $r = 0.1$, $\lambda = 1$ and $\beta = 4$.

7.1. Computation time & precision

Fig. 5 shows the computation time, number of bad points (NBP) and RMS of PCA, RNE, HF_points, HF_cubes, HF_unif, LRRSGNE and our method on the Octahedron and Fandisk models. The sampling number of these models varies from 20K to 100K. For each model, we add 50% noise. In the second and third columns, the value of vertical axes is shown in logarithmic scale. The computation time of PCA is less than other methods, but its precision is the worst. The LRRSGNE obtains

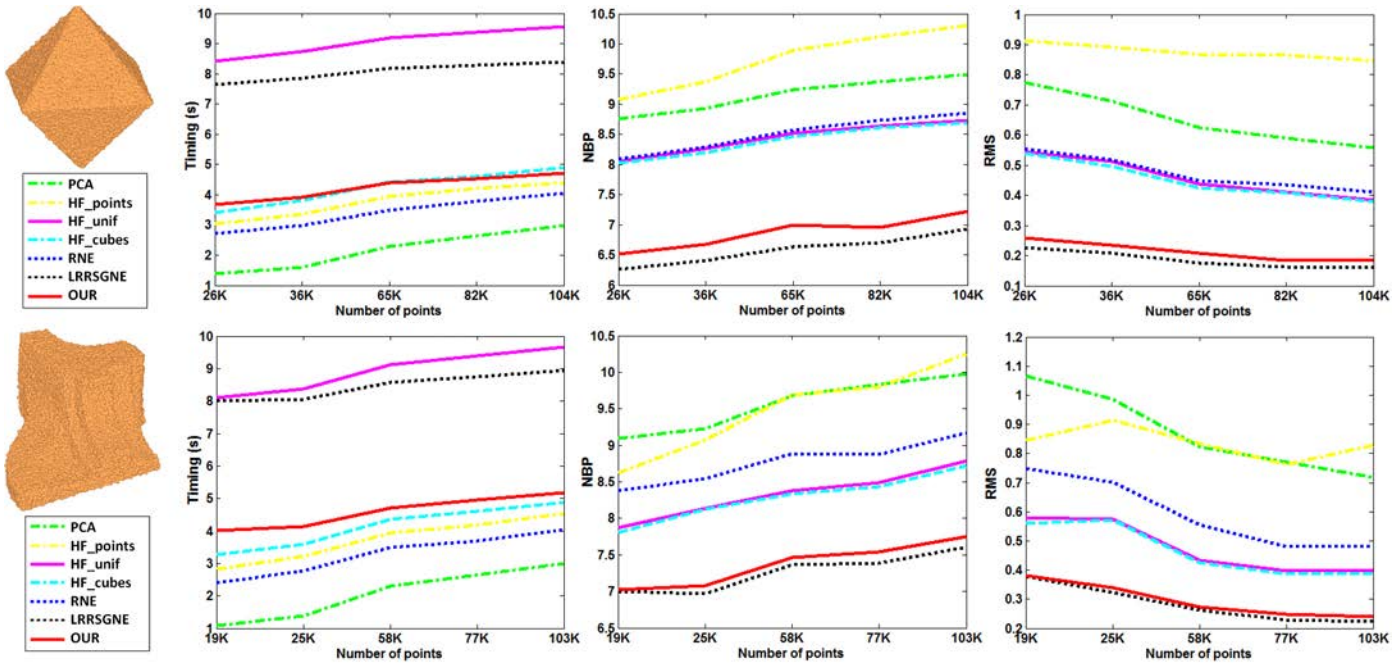


Figure 5: Comparison of the speed and performance on the Octahedron and Fandisk models. The computation time, NBP, and RMS are shown from the second column to the fourth column.

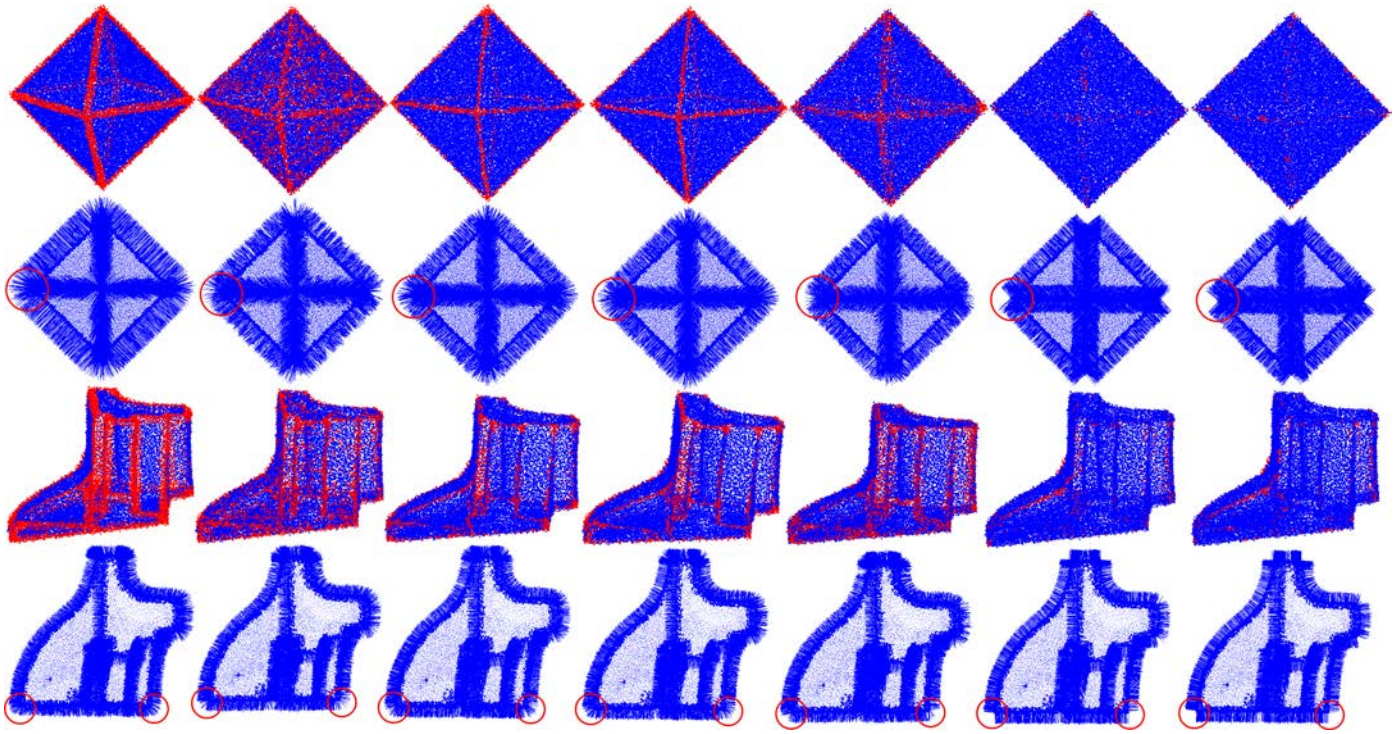


Figure 7: Visual rendering of bad points (the first and third rows) and top view of computed normals near sharp features (the second and fourth rows). 50% noise is added. The results of PCA, HF_points, HF_unif, HF_cubes, RNE, LRRSGNE, and our method are shown from the first column to the last. Our method respects the sharp features and generates fewer bad points.

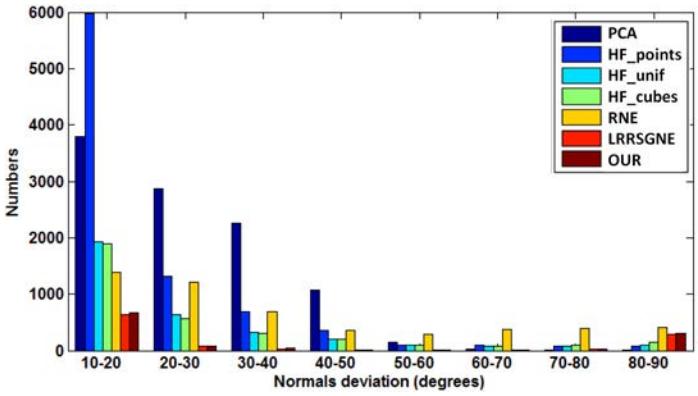


Figure 6: The distribution of bad points of the Fandisk model (26K) with 50% noise.

Table 2: Computation time of our method and LRRSGNE on Octahedron and Fandisk models. All times are in seconds

Method	Octahedron				
	26K	36K	65K	82K	104K
LRRSGNE	2101	2556	3594	3991	4414
OUR	40	50	80	93	112
Method	Fandisk				
	19K	25K	58K	77K	103K
LRRSGNE	2967	3176	5273	6261	7593
OUR	55	62	112	141	176

the minimum NBP and the lowest RMS, but it is time consuming. HF_points and RNE is slightly slower than PCA and much faster than LRRSGNE. But the quality of normals by them is far worse than that of LRRSGNE. The NBP and RMS of HF_unif, HF_cubes, and RNE are comparable, however HF_unif is much lower than HF_cubes and RNE. HF_unif generates more faithful results when the point cloud is sampled non-uniformly. Our method is much faster than LRRSGNE and slightly slower than HF_cubes, HF_points and RNE. But its results are comparable with those of LRRSGNE and much better than the other methods. It balances speed with quality well among all these methods.

Fig. 5 illustrates the computation time in logarithmic scale. Tab 2 lists the timings of LRRSGNE and our method for the Octahedron and Fandisk models under different samplings. We see that our method is about 40 times faster than LRRSGNE for models with 100k points.

To evaluate the quality of the results more precisely, we divide the normal deviation region of bad point ($10^\circ - 90^\circ$) into eight regions and show NBP in each region in Fig. 6. The visual representation of bad points and computed normals near sharp features are shown in Fig. 7. Near the sharp features, normals estimated by PCA are overly smoothed and the NBP generated by it is larger than other methods. Most of bad points generated by PCA fall in the regions from 10 to 50 degrees. It is because that the normals generated by PCA are excessively smoothed and the largest deviations are almost 40-60 degrees. Other edge preserving normal estimation methods generate less bad points especially in the regions between 20 and 80 degrees. Compared with HF_points, HF_cubes and HF_unif, RNE preserves the sharp features better. But, the normals estimated by them are still overly smoothed when extremely near the sharp features (see the regions marked by the red circles in Fig. 7). Only LRRSGNE and our method can recover the sharp features well. The NBP generated by our method is similar with LRRSGNE and much less than the other methods. For LRRSGNE and our method, the frequencies of bad points fallen in 80-90 region are higher. It is because that heavy noise makes the intersection of two planes becoming a ribbon from a line, where the points are supposed to have two directions. Therefore, if the normals preserve the sharp features well, the frequency of bad points fallen in 80-90 region maybe high.

Table 3: Comparison of RMS and NBP on Octahedron and Fandisk models with different noise levels. LRRSGNE and our method are comparable and much better than the other methods.

Method		Octahedron (26K)			Fandisk (26K)		
		40%	50%	60%	40%	50%	60%
PCA	RMS	0.771	0.773	0.774	0.984	0.986	0.994
	NBP	6302	6342	6341	10136	10183	10340
HF_points	RMS	0.718	0.912	1.080	0.756	0.914	1.060
	NBP	5395	8768	12329	5920	8690	11721
HF_unif	RMS	0.445	0.544	0.614	0.454	0.575	0.701
	NBP	2085	3122	3972	2130	3425	5115
HF_cubes	RMS	0.422	0.538	0.608	0.451	0.571	0.689
	NBP	1867	3048	3893	2100	3375	4925
RNE	RMS	0.461	0.556	0.661	0.578	0.701	0.819
	NBP	2223	3246	4592	3472	5110	6981
LRRSGNE	RMS	**0.162**	**0.228**	**0.346**	**0.248**	**0.324**	**0.426**
	NBP	**264**	**525**	**1239**	**624**	**1067**	**1860**
OUR	RMS	**0.165**	**0.258**	**0.369**	**0.266**	**0.341**	**0.460**
	NBP	**271**	**673**	**1412**	**719**	**1180**	**2175**

7.2. Robustness to noise and sampling density

We corrupt the Fandisk and Octahedron models with 40%, 50%, and 60% noise. Tab 3 shows RMS and NBP of different methods on these models. Our method achieves comparable results with LRRSGNE and is much better than the other methods.

Fig. 8 shows the bad points on the Tetrahedron models sampled with face-specific levels of density and corrupted with 50% noise. Since PCA, HF_points, and RNE are not devised to deal with non-uniform point distribution, they are severely affected. HF_cubes and HF_unif are designed to handle density variation and perform better than PCA, HF_points, and RNE. However, in the vicinities of sharp features, many erroneous normals may still persist. LRRSGNE and our method preserve the sharp features well, even when the sampling is very anisotropic around the sharp features. The NBP and RMS of different methods on these models with variational density and noise are furthermore illustrated in Fig. 9. The NBP is shown in logarithmic scale. As expected, the results of LRRSGNE and our method are comparable and more precise than the other methods.

7.3. More results

In Fig. 10, we apply our method to the scanned point clouds in which the typical imperfections, such as noise, outliers and sampling anisotropy, are common and the sharp features are

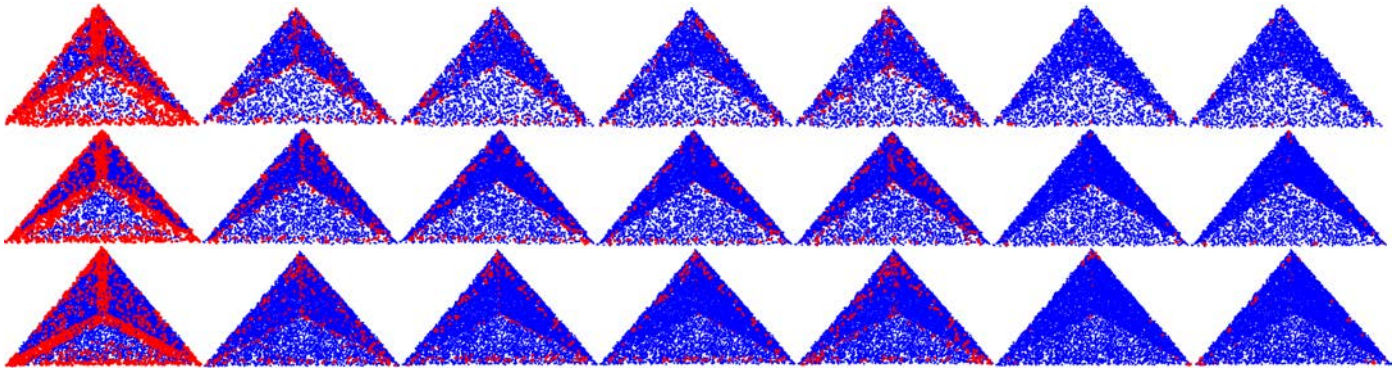


Figure 8: Visual rendering of bad points on the Tetrahedron models with 50% noise and variational density. Density is uniform on each face. From the top to bottom row, the ratios of sampling number on four faces are 1 : 2 : 3 : 4, 1 : 3 : 5 : 7, and 1 : 4 : 6 : 8. The results of PCA, HF_points, HF_cubes, HF_unif, RNE, LRRSGNE, and our method are shown from the first column to the last. LRRSGNE and our method handle the anisotropic sampling well.

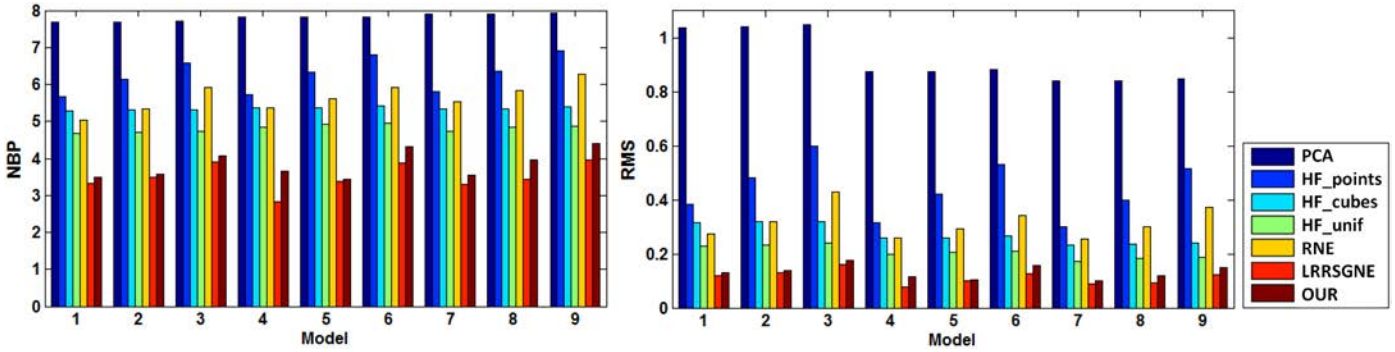


Figure 9: Comparison of the NBP and RMS on the Tetrahedron models. For each model, the ratio of sampling number on four faces is 1 : 2 : 3 : 4, 1 : 3 : 5 : 7 or 1 : 4 : 6 : 8 and the noise added to them is 40%, 50%, or 60%. The results of LRRSGNE and our method are comparable and more precise than the other methods.

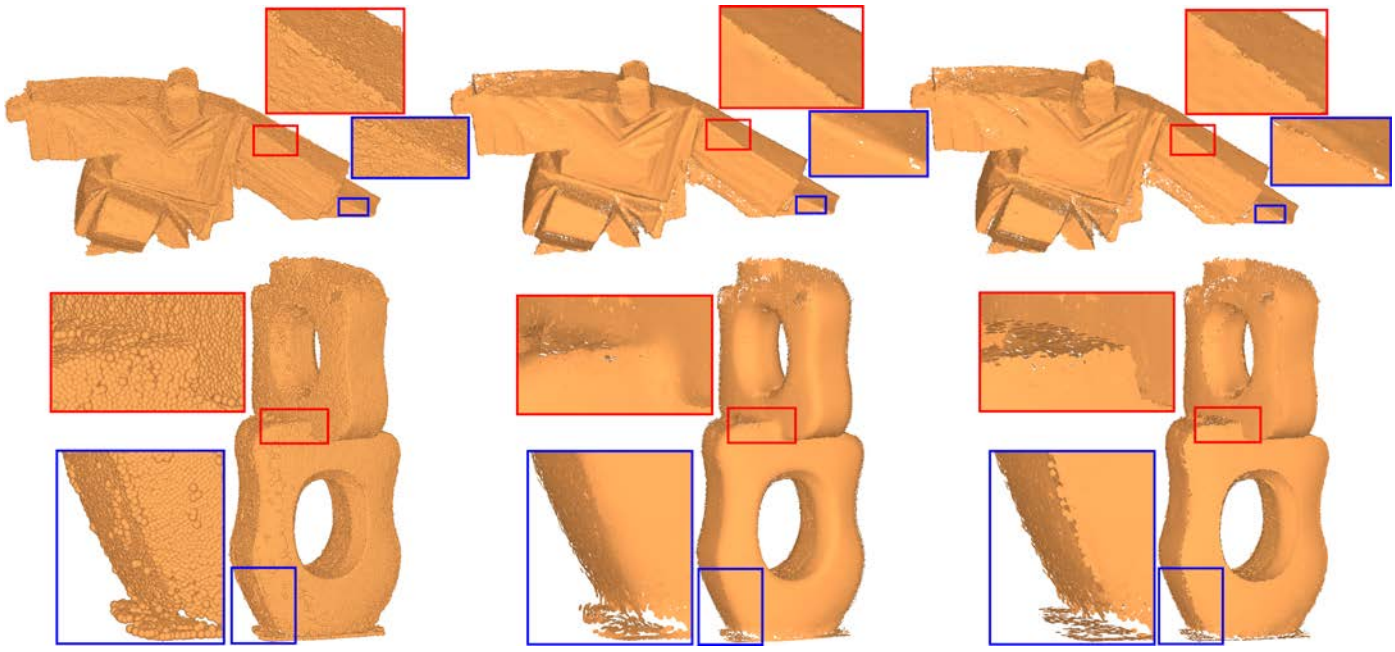


Figure 10: Normal estimation for raw scans of real objects: Genus2 and Taichi. Left to right are the input model, the results of PCA and our algorithm, respectively.

usually corrupted by these imperfections. Our method can recover the edges of Taichi and Genus2 models faithfully. Nearby surface sheets which are contained in Genus2 model and marked by blue circle always challenge normal estimation. The normals estimated by approaches based on distance, such as PCA and its variants, tend to be greatly affected by the points lying on the other sheet, while our structure based method not. Moreover, our method is competent in dealing with raw point clouds with non-uniform sampling. In the Genus2 model, the region marked by red circle is sampled anisotrpoically. The normals estimated by our method preserve the sharp features quite well.

8. Conclusions

In this paper, we present a fast and feature preserving approach to estimate quality normals for point clouds even in the presence of heavy noise and non-uniform point distribution. Following the framework of LRRSGNE [15], which generates more faithful normals than previous methods but at the price of a longer runtime, two improvements are presented. We first propose a novel linear subspace segmentation model - LSRSG. A rapid numerical scheme of LSRSG and its parallel implementation are both devised. Besides less runtime, experiments and theoretical analysis show that it generates subspace segmentation as quality as the low-rank subspace segmentation model used in LRRSGNE. To reduce the runtime of the normal estimation framework further, we develop a subspace structure propagation algorithm. Instead of computing the subspace structures for all the candidate feature points via subspace segmentation, only parts of them are estimated by LSRSG. The neighborhood structures of the rest candidate points are inferred using the propagation algorithm which is faster than LSRSG. It speeds up the normal estimation significantly and reduces the process from hours to minutes. The experiments exhibit that LRRSGNE and our method generate more faithful normals than other state-of-the-art methods. Furthermore, our method generates comparable normals as LRRSGNE but with far less runtime - about at least 40 times faster than LRRSGNE for models with 100k points.

Although we estimate quality normals in acceptable runtime with the parameters fixed in all of our experiments, more faithful normals can be generated with delicate parameters turning. In the future, we would like to choose these parameters adaptively according to various noise and sampling density. Furthermore, similar to some existing methods [1, 2], our method produces jagged features on sparse point clouds. The global labeling techniques [35] may be helpful to smooth out these jagged feature lines. Another future work is to apply our LSRSG model to more computer vision and computer graphics applications, such as shape labelling and co-segmentation.

9. Appendix

Proof (of Theorem 1): Suppose the optimal solution of model (6) is

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \cdots & \mathbf{Z}_{1n} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} & \cdots & \mathbf{Z}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{n1} & \mathbf{Z}_{n2} & \cdots & \mathbf{Z}_{nn} \end{bmatrix}, \tag{20}$$

where $\mathbf{Z}_{ij}$ is the coefficients of $\mathbf{X}_j$ represented by $\mathbf{X}_i$. We should prove $\mathbf{Z}_{ij} = \mathbf{0}$, for $i \neq j$. For $i = 1$, we have

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{X}_1\mathbf{Z}_{11} + \mathbf{X}_2\mathbf{Z}_{21} + \cdots + \mathbf{X}_n\mathbf{Z}_{n1}, \\ \mathbf{X}_1 - \mathbf{X}_1\mathbf{Z}_{11} &= \sum_{i=2}^{n} \mathbf{X}_i\mathbf{Z}_{i1}. \end{aligned} \tag{21}$$

Since these subspaces are independent, $\mathcal{S}_1 \cap \oplus_{i=2}^{n}\mathcal{S}_i = 0$. So, we have $\mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_{11}$. Evidenced by the same theory, $\mathbf{X}_i = \mathbf{X}_i\mathbf{Z}_{ii}$, for $i = 1, 2, \cdots, n$. Therefore,

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{22} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Z}_{nn} \end{bmatrix}, \tag{22}$$

is also a solution for model (6). According to the definition of favorable block-diagonal function and separable function, we have $f_{FBD}(\mathbf{Z}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{Z}) \leq f_{FBD}(\tilde{\mathbf{Z}}) + \beta f_S(\mathbf{\Omega} \odot \tilde{\mathbf{Z}})$. $\mathbf{Z}$ is the optimal solution, so $f_{FBD}(\mathbf{Z}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{Z}) \geq f_{FBD}(\tilde{\mathbf{Z}}) + \beta f_S(\mathbf{\Omega} \odot \tilde{\mathbf{Z}})$. Therefore, we have $f_{FBD}(\mathbf{Z}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{Z}) = f_{FBD}(\tilde{\mathbf{Z}}) + \beta f_S(\mathbf{\Omega} \odot \tilde{\mathbf{Z}})$. This equality holds if and only if $\mathbf{Z}_{ij} = \mathbf{0}$, for $i \neq j$. □

Proof (of Theorem 2): First, we will prove that there exist β making the solution to be the following form:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Z}_{12}^T & \mathbf{Z}_{22} & \mathbf{Z}_{23} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{23}^T & \mathbf{Z}_{33} & \mathbf{Z}_{34} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Z}_{34}^T & \mathbf{Z}_{44} & \vdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Z}_{nn} \end{bmatrix}. \tag{23}$$

where $\mathbf{Z}_{ij}$ is the coefficients of $\mathbf{X}_j$ represented by $\mathbf{X}_i$. The pairs of $\mathbf{Z}_{i,i+1}$ and $\mathbf{Z}_{i,i+1}^T$ make sense, since $\mathbf{Z}$ is symmetric. Supposing one element located in the all zeros submatrix of $\mathbf{Z}$ is a ($a > 0$), there $\exists \beta = n/a$ make $f_{FBD}(\mathbf{Z}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{Z}) > n$. Since $\mathbf{X} = \mathbf{XI}$, $\mathbf{I}$ is one solution. However, $f_{FBD}(\mathbf{I}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{I}) = n$. Therefore, this assumption is invalid.

Next, we will prove that if $\mathbf{Z}$ is the optimal solution, $\mathbf{Z}_{i,i+1} = \mathbf{0}$, for $i = 1, \cdots, n$. For $i = 1$, we have $\mathbf{X}_1 - \mathbf{X}_1\mathbf{Z}_{11} = \mathbf{X}_2\mathbf{Z}_{12}^T$. Since $\mathcal{S}_1$ and $\mathcal{S}_2$are disjoint, we have $\mathcal{S}_1 \cap \mathcal{S}_2 = \mathbf{0}$. Therefore,

$\mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_{11}$ and $\mathbf{X}_2\mathbf{Z}_{12}^T = \mathbf{0}$. We construct

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{22} & \mathbf{Z}_{23} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{23}^T & \mathbf{Z}_{33} & \mathbf{Z}_{34} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Z}_{34}^T & \mathbf{Z}_{44} & \vdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Z}_{nn} \end{bmatrix}. \quad (24)$$

$\mathbf{Z}$ and $\tilde{\mathbf{Z}}$ are the same except for $\mathbf{Z}_{12}$. Because $\mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_{11}$, we can get $\mathbf{X} = \mathbf{X}\tilde{\mathbf{Z}}$. According to the definition of favorable block-diagonal function and separable function, we have $f_{FBD}(\mathbf{Z}) + \beta f_S(\mathbf{\Omega} \odot \mathbf{Z}) \leq f_{FBD}(\tilde{\mathbf{Z}}) + \beta f_S(\mathbf{\Omega} \odot \tilde{\mathbf{Z}})$. This equality holds if and only if $\mathbf{Z}_{12} = 0$. Evidenced by the same theory, we have $\mathbf{Z}_{i,i+1} = \mathbf{0}$, for $i = 2, \cdots, n$. □

Acknowledgement

The authors would like to thank all the reviewers for their valuable comments. Thanks to Bao Li and Alexandre Boulch for providing the code used for comparison. Xiuping Liu is supported by the NSFC Fund (Nos. 61173102 and 61370143). Junjie Cao is supported by the NSFC Fund (No. 61363048). Bo Li is supported by the NSFC Fund (No. 61262050).

References

[1] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang, S. Jin, Robust normal estimation for point clouds with sharp features, Computers & Graphics 34 (2) (2010) 94–106.

[2] A. Boulch, R. Marlet, Fast and robust normal estimation for point clouds with sharp features, Comput. Graph. Forum 31 (5) (2012) 1765–1774.

[3] S. Rusinkiewicz, M. Levoy, Qsplat: a multiresolution point rendering system for large meshes, in: SIGGRAPH, 2000, pp. 343–352.

[4] M. Zwicker, H. Pfister, J. van Baar, M. H. Gross, Surface splatting, in: SIGGRAPH, 2001, pp. 371–378.

[5] J. Wang, D. Gu, Z. Yu, C. Tan, L. Zhou, A framework for 3d model reconstruction in reverse engineering, Computers & Industrial Engineering 63 (4) (2012) 1189–1200.

[6] J. Wang, Z. Yu, W. Zhu, J. Cao, Feature-preserving surface reconstruction from unoriented, noisy point data, Comput. Graph. Forum 32 (1) (2013) 164–176.

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

[8] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. M. Ascher, H. R. Zhang, Edge-aware point set resampling, ACM Trans. Graph. 32 (1) (2013) 9.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, in: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1992, 1992, pp. 71–78.

[10] G. Guennebaud, M. H. Gross, Algebraic point set surfaces, ACM Trans. Graph. 26 (3) (2007) 23.

[11] F. Cazals, M. Pouget, Estimating differential quantities using polynomial fitting of osculating jets, Computer Aided Geometric Design 22 (2005) 121–146.

[12] N. J. Mitra, A. Nguyen, L. J. Guibas, Estimating surface normals in noisy point cloud data, Int. J. Comput. Geometry Appl. 14 (4-5) (2004) 261–276.

[13] S. Fleishman, D. Cohen-Or, C. T. Silva, Robust moving least-squares fitting with sharp features, ACM Trans. Graph. 24 (3) (2005) 544–552.

[14] M. Yoon, Y. Lee, S. Lee, I. P. Ivrissimtzis, H. Seidel, Surface and normal ensembles for surface reconstruction, Computer-Aided Design 39 (5) (2007) 408–420.

[15] J. Zhang, J. Cao, X. Liu, J. Wang, J. Liu, X. Shi, Point cloud normal estimation via low-rank subspace clustering, Computers & Graphics 37 (6) (2013) 697–706.

[16] K. Klasing, D. Althoff, D. Wollherr, M. Buss, Comparison of surface normal estimation methods for range sensing applications, in: IEEE International Conference on Robotics and Automation, 2009, pp. 3206–3211.

[17] F. Cazals, M. Pouget, Estimating differential quantities using polynomial fitting of osculating jets, Computer Aided Geometric Design 22 (2005) 121–146.

[18] M. Pauly, R. Keiser, L. Kobbelt, M. H. Gross, Shape modeling with point-sampled geometry, ACM Trans. Graph. 22 (3) (2003) 641–650.

[19] T. R. Jones, F. Durand, M. Zwicker, Normal improvement for point rendering, IEEE Computer Graphics and Applications 24 (4) (2004) 53–56.

[20] F. Calderón, U. Ruiz, M. Rivera, Surface-normal estimation with neighborhood reorganization for 3d reconstruction, in: Progress in Pattern Recognition, Image Analysis and Applications, 2007, pp. 321–330.

[21] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, Point set surfaces, in: IEEE Visualization 2001, October 24-26, 2001, San Diego, CA, USA, Proceedings, 2001, pp. 21–28.

[22] A. C. Öztireli, G. Guennebaud, M. H. Gross, Feature preserving point set surfaces based on non-linear kernel regression, Comput. Graph. Forum 28 (2) (2009) 493–501.

[23] N. Amenta, M. W. Bern, Surface reconstruction by voronoi filtering, Discrete & Computational Geometry 22 (4) (1999) 481–504.

[24] T. K. Dey, S. Goswami, Provable surface reconstruction from noisy samples, Comput. Geom. 35 (1-2) (2006) 124–141.

[25] P. Alliez, D. Cohen-Steiner, Y. Tong, M. Desbrun, Voronoi-based variational reconstruction of unoriented point sets, in: Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Barcelona, Spain, July 4-6, 2007, 2007, pp. 39–48.

[26] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, H. Zhang, Edge-aware point set resampling, ACM Transactions on Graphics 32 (2013) 9:1–9:12.

[27] Y. Wang, H.-Y. Feng, F.-É. Delorme, S. Engin, An adaptive normal estimation method for scanned point clouds with sharp features, Computer-Aided Design 45 (11) (2013) 1333 – 1348.

[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184.

[29] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, CoRR abs/1404.6736.

[30] X. Liu, J. Zhang, R. Liu, B. Li, J. Wang, J. Cao, Low-rank 3d mesh segmentation and labeling with structure guiding, Computers & Graphics 46 (2015) 99–109.

[31] K. Tang, R. Liu, Z. Su, J. Zhang, Structure-constrained low-rank representation, IEEE Trans. Neural Netw. Learning Syst. 25 (12) (2014) 2167–2179.

[32] Y.-K. Lai, Q.-Y. Zhou, S.-M. Hu, J. Wallner, D. Pottmann, et al., Robust feature classification and editing, Visualization and Computer Graphics, IEEE Transactions on 13 (1) (2007) 34–45.

[33] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[34] J. Yang, Y. Zhang, Alternating direction algorithms for $_1$-problems in compressive sensing, SIAM J. Scientific Computing 33 (1) (2011) 250–278.

[35] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, Pattern Analysis and Machine Intelligence, IEEE Transactions on 23 (11) (2001) 1222–1239.