Quality Point Cloud Normal Estimation by Guided Least Squares Representation

Xiuping Liu\textsuperscript{a}, Jie Zhang\textsuperscript{b}, Junjie Cao\textsuperscript{a,b}, Bo Li\textsuperscript{b}, Ligang Liu\textsuperscript{b}

\textsuperscript{a}School of Mathematical Sciences, Dalian University of Technology, Dalian, China.
\textsuperscript{b}School of Mathematical Sciences, Nanchang Hangkong University, Nanchang, 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 \cite{8}, surface reconstruction \cite{5,6} and anisotropic smoothing \cite{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 \cite{9}. However, the acquired points are inevitably defect-ridden and normal estimation is sensitive to these defects 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 \cite{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 \cite{13,14,11} estimate consistent sub-neighborhoods to compute normals for feature preserving. However, the most recently proposed statistics-based method \cite{11} 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, Bouch et al. \cite{1} design an uniform sampling strategy. However, in the vicinity of sharp features, some erroneous normals may still persist, as shown in Fig. 1.1. Moreover, the performance of this method drops when the dihedral angel is large. Utilizing the subspace structures of the underlying piecewise surfaces, LRRSGNE \cite{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. Second, to reduce the runtime further...
ther, a subspace structure propagation algorithm is proposed. After analyzing the subspace structures for a small percentage of points near sharp features via LSRSG, the rest candidate features’ 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 [10]. 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 fitting 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. [11] 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. However, 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] 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. [29] 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 anisotropic neighborhoods of the rest candidate feature points. Then quality normals can be estimated by the consistent sub-neighborhood enclosing neighbor points sampled from the same smooth surface patch as the point. 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 $N_i$ 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:

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

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are the singular values of the covariance matrix of $N_i$. 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 $\{w_i\}_{i=1}^N$.

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 4. 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 4.

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.

For a candidate feature point \( p_i \), a larger neighborhood \( N^d_i \) is selected. The \( j \)-th neighbor point \( p_j \) of \( p_i \) is represented as \( x_j = [x_j^1, x_j^2, x_j^3, n_j^1, n_j^2]^T \), where \( n_j^1, n_j^2 \) is its normal computed by PCA and \([x_j^1, x_j^2, x_j^3]\) is local coordinate of \( p_j \) with \( p_i \) as the origin. The data matrix \( X \) is defined as

\[
X = [x_1, x_2, \ldots, x_s]^T
\]

**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 \( X = [X_1, X_2, \ldots, X_n] \in \mathbb{R}^{d \times m} \) sampled from \( n \) subspaces which compose the space \( S \) and \( \dim(S) = d \). The sample set \( X_i \in \mathbb{R}^{d \times m} \) is drawn from subspace \( S_i \) and \( \dim(S_i) = d_i, i = 1, 2, \ldots, n \). Our task is to group the data according to the subspaces from which they are drawn.

Based on the observation that each sample \( x_i \) 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

\[
{\min}_{Z} ||Z||_* \quad \text{s.t.} \quad X = XZ
\]

where \( || \cdot ||_* \) is the matrix nuclear norm \( i.e. \) the sum of singular values. 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 \( x_i \) and \( x_j \) are drawn from different subspaces \( S(i, j) \) 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):

$$\min_{Z} \|Z\|_{F} + \beta \|\Omega \odot Z\|_{F} \quad \text{s.t.} \quad X = XZ \tag{5}$$

where $\beta$ is parameter, $\| \cdot \|_{F}$ represents $\ell_2$-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:

$$\min_{Z} \|f_{FB}(Z)\|_{F} + \beta f_{S}(\Omega \odot Z) \quad \text{s.t.} \quad X = XZ. \tag{6}$$

where $f_{FB}$ 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) $f(U M V) = f(M)$ for all $M \in \mathbb{R}^{m \times n}$ and all unitary matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$.

2) For all square matrices $A$ and $D$, $f\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = f(A) \circ f(D)$ 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 $A \in \mathbb{R}^{m \times n}$, $f(A)$ can represented as:

$$f(A) = f_0\left(\sum_{i} f_{i}(a_{ij})\right), \tag{7}$$

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

The matrix nuclear norm is a favorable block-diagonal function, the $\ell_1$-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 $\ell_1$-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 $\ell_1$-norm in model (5) with it, we have the LSRSG for data without noise:

$$\min_{Z} \|Z\|_{F}^2 + \beta \|\Omega \odot Z\|_{F}^2 \quad \text{s.t.} \quad X = XZ. \tag{8}$$

The effectiveness of it and its generalization (6) are guaranteed by the two following theorems.

**Theorem 1:** If $S_1, S_2, \cdots, S_n$ are independent, the optimal solution to the model (6) is a block-diagonal matrix

$$Z = \begin{bmatrix} Z_{1,1} & 0 & \cdots & 0 \\ 0 & Z_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z_{n,n} \end{bmatrix}, \tag{9}$$

where $Z_{i,i}$ is a $m_i \times m_i$ matrix.

**Theorem 2:** Denote $x_i$ as the $i$-th sample from $X$. Since both $z_{ij}$ and $x_{ij}$ denote the affinity between the sample $x_i$ and $x_j$, it is natural to suppose that $Z$ is symmetric. If $S_1, S_2, \cdots, S_n$ are disjoint and

$$\Omega = \begin{bmatrix} A^*_{1,1} & B^*_{1,1} & C^*_{1,1} & \cdots & C^*_{1,n} \\ B^*_{1,2} & A^*_{2,2} & B^*_{2,2} & \cdots & C^*_{2,n} \\ C^*_{1,3} & B^*_{1,3} & A^*_{3,3} & \cdots & C^*_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C^*_{1,n} & C^*_{2,n} & C^*_{3,n} & \cdots & A^*_{n,n} \end{bmatrix}, \tag{10}$$

where the elements of $A^* \in \mathbb{R}^{m \times m}$ and $B^* \in \mathbb{R}^{m \times m}$ are all zeros, whereas the elements of $C^* \in \mathbb{R}^{m \times m}$ are all ones, there exists $\beta$, 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, $X$ is corrupted by noise, certain relaxation to the equality constraint in model (6) is desirable:

$$\min \|Z\|_{F}^2 + \beta \|\Omega \odot Z\|_{F}^2 + \lambda \|X - XZ\|_{F}^2, \tag{11}$$

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:

$$2Z^* + 2\beta (\Omega \odot Z^*) - 2\lambda (X^T (X - XZ^*)) = 0,$$

$$Z^* + \beta (\Omega \odot Z^*) - \lambda (X^T X - X^T XZ^*) = 0,$$

$$\lambda(I + X^T X)Z^* + \beta (\Omega \odot Z^*) = \lambda(X^T X), \tag{12}$$

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

$$(I + X^T X)Z^*(.; j) + \beta(diag(\Omega(:, j))Z^*(.; j)) = \lambda(X^T X)(.; j), \tag{13}$$

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

$$Z^*(.; j) = \lambda inv(I + X^T X + \beta diag(\Omega(:, j)))XTX(.; j), \tag{14}$$

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

### Table 1: Computation time for the toy examples.

<table>
<thead>
<tr>
<th></th>
<th>LRR</th>
<th>LRRSG</th>
<th>LSRSG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two planes</td>
<td>2.51</td>
<td>2.23</td>
<td>0.44</td>
</tr>
<tr>
<td>One plane &amp; two lines</td>
<td>2.54</td>
<td>2.32</td>
<td>0.48</td>
</tr>
</tbody>
</table>
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 $\Omega$ which is constructed by forty percent prior knowledge with 30% errors. Firstly, an ideal full guiding matrix $G$ is built. Specifically, if two samples $p_j$ and $p_k$ are in the same subspaces, $G(i, j) = 0$, otherwise $G(i, j) = 1$. The guiding matrix is generated by choosing 40% elements of $G$ randomly and the other elements of $\Omega$ 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 number of data increases, the advantage of parallel running strategy becomes more obvious.
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 R and N to store the segmentation results obtained by LSRSG. The values of \( R(i, j) \) and \( N(i, j) \) represent the number of \( p_i \) and \( p_j \) grouped into the same subspace and into different subspaces, respectively. To compute \( R \) and \( N \), 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

\[
    w_i^d = |w_i - w_{\text{ave}}|,
\]

where \( w_{\text{ave}} \) 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 \( N_i^* \), we select a smaller neighborhood \( N_i^\prime \) with size \( S^\prime \). The segmentation of the neighborhood \( N_i^\prime \) is inferred by the subspace structure propagation algorithm, if the current point \( p_i \) and its neighborhood \( N_i^\prime \) are well covered by \( T \). Specifically, if \( N_i^\prime \) \( \cap T = \emptyset \), we add the point \( p_i \) into \( T \), segment \( N_i^\prime \) by LSRSG and modify \( R \) and \( N \). Otherwise, \( N_i^\prime \) is segmented by the propagation algorithm presented as follows. We find a seed point \( p_j \) from \( N_i^\prime \) 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 \( REL_{i,Q} \) between point \( p_i \) and set \( Q \) is defined as:

\[
    REL_{i,Q} = \max_{p_j \in Q} (\text{rel}(p_i, p_j)),
\]

\[
    \text{rel}(p_i, p_j) = \begin{cases} 
        -1, & \text{if } N(l, j) > 0 \\
        R(l, j), & \text{if } N(l, j) = 0
    \end{cases}.
\]

This process is terminated until \( REL_{i,Q} \leq 2 \) or \(|Q| > 0.7 \times S\), where \(|Q|\) denotes the cardinality of the set \( Q \). Then, for the rest of \( p_i \)'s neighbors \( N_i^* = N_i^\prime \setminus Q \), we repeat the process until \( N_i^* \) is empty. If a neighbor \( p_j \) is not belong to any previous segmented neighborhoods, \( i.e. \) to structure information about it is stored \( R \) and \( N \), it is ignored and deleted from the current neighborhood \( N_i^* \). Thus, \( N_i^* \) is segmented into some sub-regions \( Q_1, Q_2, \ldots \).

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 LRRSG-GNE \[15\]. 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:

\[
    \text{RMS}_{\tau} = \left( \frac{1}{|P|} \sum_{p \in P} (f(n_{p,\text{ref}}n_{p,\text{est}})) \right)^{1/2},
\]

where

\[
    f(n_{p,\text{ref}}n_{p,\text{est}}) = \begin{cases} 
        n_{p,\text{ref}}n_{p,\text{est}}, & \text{if } n_{p,\text{ref}}n_{p,\text{est}} < \tau \\
        \pi/2, & \text{otherwise}
    \end{cases}
\]

\( n_{p,\text{ref}} \) and \( n_{p,\text{est}} \) 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 \( \tau \) 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 choice of $S$ and $S^*$ depends on the noise. Since only part sharp neighbors are used to estimate the normals when points are near sharp features, $S^*$ should be larger than $S$. Parameter $S^*$ 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 is large, we should relax the fitting restriction and decrease the value of $\lambda$ and increase the value of $\beta$. In our implementation, $S$, $S^*$, $S^r$, $\lambda$, and $\beta$ are selected as $S = 70$, $S^* = 120$, $S^r = 30$, $r = 0.1$, $\lambda = 1$ and $\beta = 4$.

7.1. Computation time & precision

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. 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...
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\text{points}, HF\text{unif}, HF\text{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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Octahedron</th>
<th>Fandisk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26K</td>
<td>36K</td>
</tr>
<tr>
<td>LRRSGNE</td>
<td>2101</td>
<td>2556</td>
</tr>
<tr>
<td>OUR</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Octahedron (26K)</th>
<th>Fandisk (26K)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40%</td>
<td>50%</td>
</tr>
<tr>
<td>PCA RMS</td>
<td>0.771</td>
<td>0.773</td>
</tr>
<tr>
<td>NBP</td>
<td>6302</td>
<td>6342</td>
</tr>
<tr>
<td>HF\text{points RMS}</td>
<td>0.718</td>
<td>0.912</td>
</tr>
<tr>
<td>NBP</td>
<td>5395</td>
<td>8768</td>
</tr>
<tr>
<td>HF\text{unif RMS}</td>
<td>0.445</td>
<td>0.544</td>
</tr>
<tr>
<td>NBP</td>
<td>2085</td>
<td>3122</td>
</tr>
<tr>
<td>HF\text{cubes RMS}</td>
<td>0.422</td>
<td>0.538</td>
</tr>
<tr>
<td>NBP</td>
<td>1867</td>
<td>3048</td>
</tr>
<tr>
<td>RNE RMS</td>
<td>0.461</td>
<td>0.556</td>
</tr>
<tr>
<td>NBP</td>
<td>2223</td>
<td>3246</td>
</tr>
<tr>
<td>LRRSGNE RMS</td>
<td>0.162</td>
<td>0.228</td>
</tr>
<tr>
<td>NBP</td>
<td>264</td>
<td>525</td>
</tr>
<tr>
<td>OUR RMS</td>
<td>0.165</td>
<td>0.258</td>
</tr>
<tr>
<td>NBP</td>
<td>271</td>
<td>673</td>
</tr>
</tbody>
</table>

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\text{points}, HF\text{cubes and HF\text{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.

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.
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.
Fig. 3 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 variations 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. 7. 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 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 anisotropically. 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, 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 LRS-SG. 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 tuning. In the future, we would like to choose these parameters adaptively according to various noise and sampling density. Furthermore, similar to some existing methods, our method produces jagged features on sparse point clouds. The global labeling techniques 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 2): Suppose the optimal solution of model (6) is

\[
Z = \begin{bmatrix}
Z_{11} & Z_{12} & \cdots & Z_{1n} \\
Z_{21} & Z_{22} & \cdots & Z_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
Z_{n1} & Z_{n2} & \cdots & Z_{nn}
\end{bmatrix},
\]

(20)

where \(Z_{ij}\) is the coefficients of \(X_i\) represented by \(X_j\). We should prove \(Z_{ij} = 0\) for \(i \neq j\). For \(i = 1, 2, \cdots, n\) we have

\[
X_1 = X_1Z_{11} + X_2Z_{21} + \cdots + X_nz_{n1},
\]

(21)

Since these subspaces are independent, \(S_1 \cap \cap_{i=2}^n S_i = \emptyset\). So, we have \(X_1 = X_1z_{11}\). Evidenced by the same theory, \(X_i = X_iZ_{ii}\), for \(i = 1, 2, \cdots, n\). Therefore,

\[
Z = \begin{bmatrix}
Z_{11} & 0 & \cdots & 0 \\
0 & Z_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Z_{nn}
\end{bmatrix},
\]

(22)

is also a solution for model (6). According to the definition of favorable block-diagonal function and separable function, we have \(f_{BBD}(Z) + \beta f_5(\Omega \odot Z) \leq f_{BBD}(\tilde{Z}) + \beta f_5(\Omega \odot \tilde{Z})\). \(Z\) is the optimal solution, so \(f_{BBD}(Z) + \beta f_5(\Omega \odot Z) \geq f_{BBD}(\tilde{Z}) + \beta f_5(\Omega \odot \tilde{Z})\). Therefore, we have \(f_{BBD}(Z) + \beta f_5(\Omega \odot Z) = f_{BBD}(\tilde{Z}) + \beta f_5(\Omega \odot \tilde{Z})\). This equality holds if and only if \(Z_{ij} = 0\) for \(i \neq j\).

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

\[
Z = \begin{bmatrix}
Z_{11} & Z_{12} & 0 & 0 & \cdots & 0 \\
Z_{12} & Z_{22} & 0 & 0 & \cdots & 0 \\
0 & Z_{23} & Z_{33} & Z_{34} & \cdots & 0 \\
0 & 0 & Z_{34} & Z_{44} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & Z_{nn}
\end{bmatrix},
\]

(23)
where $Z_{ij}$ is the coefficients of $X_i$ represented by $X_j$. The pairs of $Z_{ij}$ and $Z_{ij}^+$ make sense, since $Z$ is symmetric. Supposing one element located in the all zeros submatrix of $Z$ is $a \ (a > 0)$, there $\exists \beta = n/a$ make $f_{FBD}(Z) + \beta f_\beta (\Omega \circ Z) > n$. Since $X = XI$, $I$ is one solution. However, $f_{FBD}(I) + \beta f_\beta (\Omega \circ I) = n$. Therefore, this assumption is invalid.

Next, we will prove that if $Z$ is the optimal solution, $Z_{ij} = 0$, for $i = 1, \cdots, n$. For $i = 1$, we have $X_1 = X_1Z_{11} = X_1Z_{11}^T$. Since $S_1$ and $S_2$ are disjoint, we have $S_1 \cap S_2 = 0$. Therefore $X_1 = X_1Z_{11}$ and $X_2Z_{12} = 0$. We construct

$$Z = \begin{bmatrix} Z_{11} & 0 & 0 & 0 & \cdots & 0 \\ 0 & Z_{22} & Z_{23} & Z_{24} & \cdots & 0 \\ 0 & Z_{32} & Z_{33} & Z_{34} & \cdots & 0 \\ 0 & 0 & Z_{43} & Z_{44} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & Z_{nn} \end{bmatrix}$$

(24)

$Z$ and $\tilde{Z}$ are the same except for $Z_{12}$. Because $X_1 = X_1Z_{11}$, we can get $X = X \tilde{Z}$. According to the definition of favorable block-diagonal function and separable function, we have $f_{FBD}(Z) + \beta f_\beta (\Omega \circ Z) \leq f_{FBD}(\tilde{Z}) + \beta f_\beta (\Omega \circ \tilde{Z})$. This equality holds if and only if $Z_{12} = 0$. Evidenced by the same theory, we have

$$Z_{i,i+1} = 0, \text{ for } i = 2, \cdots, n. \quad \Box$$

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