Line segment extraction for large scale unorganized point clouds

Article *in* ISPRS Journal of Photogrammetry and Remote Sensing · April 2015

DOI: 10.1016/j.isprsjprs.2014.12.027

CITATIONS
16

READS
608

7 authors, including:

Yangbin Lin
Jimei University
15 PUBLICATIONS **81** CITATIONS
SEE PROFILE

Cheng Wang
Xiamen University
157 PUBLICATIONS **894** CITATIONS
SEE PROFILE

Zhonggui Chen
Xiamen University
16 PUBLICATIONS **215** CITATIONS
SEE PROFILE

Jonathan Li
University of Waterloo
252 PUBLICATIONS **3,240** CITATIONS
SEE PROFILE

ISPRS Journal of Photogrammetry and Remote Sensing 102 (2015) 172–183

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Line segment extraction for large scale unorganized point clouds

Yangbin Lin [a], Cheng Wang [a,*], Jun Cheng [a], Bili Chen [b], Fukai Jia [a], Zhonggui Chen [a], Jonathan Li [c]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Department of Computer Science, Xiamen University, Xiamen, FJ 361005, China
[b] School of Software, Xiamen University, Xiamen, FJ 361005, China
[c] Department of Geography & Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Article history:
Received 28 May 2014
Received in revised form 18 December 2014
Accepted 19 December 2014

Keywords:
Line segment extraction
3D line-support regions
Point clouds
LiDAR
Mobile laser scanning
3D Structure

ABSTRACT

Line segment detection in images is already a well-investigated topic, although it has received considerably less attention in 3D point clouds. Benefiting from current LiDAR devices, large-scale point clouds are becoming increasingly common. Most human-made objects have flat surfaces. Line segments that occur where pairs of planes intersect give important information regarding the geometric content of point clouds, which is especially useful for automatic building reconstruction and segmentation. This paper proposes a novel method that is capable of accurately extracting plane intersection line segments from large-scale raw scan points. The 3D line-support region, namely, a point set near a straight linear structure, is extracted simultaneously. The 3D line-support region is fitted by our Line-Segment-Half-Planes (LSHP) structure, which provides a geometric constraint for a line segment, making the line segment more reliable and accurate. We demonstrate our method on the point clouds of large-scale, complex, real-world scenes acquired by LiDAR devices. We also demonstrate the application of 3D line-support regions and their LSHP structures on urban scene abstraction.

1. Introduction

Benefiting from the advances in sensor technology for both airborne and ground-based mobile laser scanning, dense points clouds have become increasingly common, and the need for new approaches to address these point clouds has become increasingly important. As the common feature in man-made objects, straight linear structures play an important role in a variety of applications, such as: road extraction (Yang et al., 2013); building outline extraction (Baillard et al., 1999); localization (Borges et al., 2010), city model building (Lafarge and Mallet, 2012); calibration (Moghadam et al., 2013); line-based visualization (Chen and Wang, 2011); and more. This paper emphasizes straight line segment extraction for point clouds, whereas most of the existing work concentrates on 2D line segment detection in a single image (Ballard, 1981; Burns et al., 1986; Von Gioi et al., 2010) and 3D line segment reconstruction in multi-view images (Baillard et al., 1999; Woo et al., 2009; Jain et al., 2010). Only a few papers consider point clouds (Lu et al., 2008; Moghadam et al., 2013).

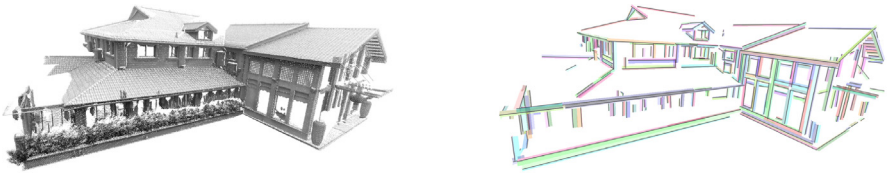
A large number of dense point clouds have been obtained by current scanners; the RIEGL VMX-450 scanner, for example, can yield 1.1 million range measurements per second. Therefore, one of the biggest challenges is to find an efficient way to address the voluminous data. Unorganized point clouds lack normal vector and connectivity information, making the problem even more challenging.

Our method is designed to cope with line segment extraction for large-scale unorganized point clouds from the real word. A line segment here is defined as the intersection of two half-planes. To extract the line segment, we take into account the point region that is near the straight linear structure. Such a region is designated as a "3D line-support region." The word "3D" is used to distinguish the region from the concept of a "line-support region," which has proved to be a robust descriptor to extract line segments in images.

The key idea of our method is to first convert a point cloud into a collection of shaded images by non-photorealistic rendering with different viewpoints; then the LSD algorithm (Von Gioi et al., 2010) is applied to these images to extract the 2D line-support regions. These 2D line-support regions are then back-projected into the original point cloud as 3D line-support regions, with each region containing roughly one line segment. Next, to maintain accuracy, each 3D line-support region is fitted by our Line-Segment-Half-Planes (LSHP) structure. Finally, the 3D line-support regions and their LSHP structures are refined as the output.

* Corresponding author.
E-mail address: cwang@xmu.edu.cn (C. Wang).

http://dx.doi.org/10.1016/j.isprsjprs.2014.12.027
0924-2716/



(a) 3D raw scan point cloud (rendered by ambient occlusion). (b) Extracted 3D line segments with attached half-planes.

Fig. 1. Given an unorganized 3D raw scan point cloud (a), our method is able to extract line segments together with attached half-planes (b), where the line segments are drawn in black color and their attached half-planes are represented by colored 3D rectangles. (a) 3D raw scan point cloud (rendered by ambient occlusion). (b) Extracted 3D line segments with attached half-planes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

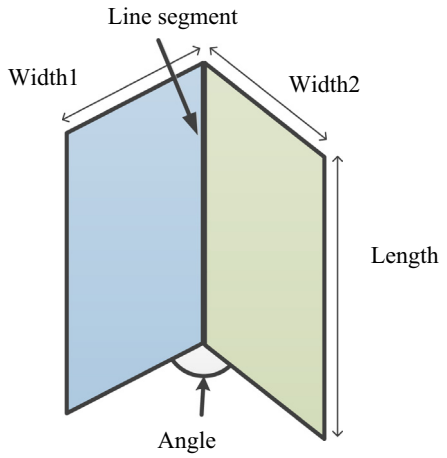


Fig. 2. Line-Segment-Half-Planes (LSHP) structure is characterized by two tangential 3D rectangles.

Fig. 1 presents a result of our method. Given an unorganized 3D raw scan point cloud as the input (Fig. 1(a)), our method extracts the 3D line-support region and LSHP structures as the output, where the line segments are drawn in black and the attached half-planes are represented by colored 3D rectangles (Fig. 1(b)). As a result, the LSHP structure provides an abstraction of a point cloud, and the vegetation in the input is filtered.

2. Related work

2.1. 2D line segment detection for a single image

Image line segment detection has been studied over several decades. The traditional methods combine the Canny edge detector (Canny, 1986) and Hough transform (Ballard, 1981). These methods are generally slow and produce a significant number of false detections. Recently, an efficient line segment detector with false detection control (designated LSD) was presented by Von Gioi et al. (2010). LSD follows the method proposed by Burns et al. (1986). First, the image is partitioned into a collection of straight image regions, named line-support regions; the gradient angles of pixels in each region are roughly oriented along the same direction. Then, a line segment, that best approximates each line-support region, is determined. Finally, a line segment validation, based on the approach of Desolneux et al. (2000) is adopted to control the number of false detections.

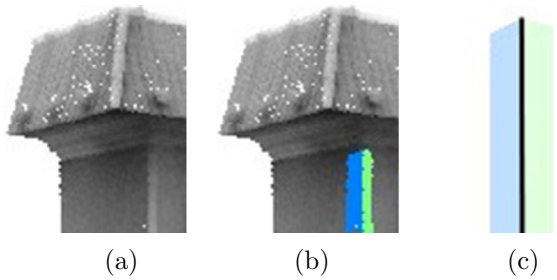


Fig. 3. Example of 3D line-support region and its Line-Segment-Half-Planes (LSHP) structure. (a) Raw scan point cloud. (b) One of the 3D line-support region. (c) The reconstructed LSHP structure of the 3D line-support region in (b).

2.2. 3D line segment reconstruction from multi-view images

Numerous papers on 3D line segment reconstruction for multi-view images have been published in recent years. Taylor and Kriegman (1995) presented a reconstruction algorithm that works by minimizing an objective function that is defined as the total squared distance between the observed edge segments from the image and the projections of the reconstructed lines. Baillard et al. (1999) found the correspondence between lines over stereo images by epipolar geometry and cross correlation scores. Then the attached half-planes are computed for piecewise planar reconstruction of the 3D model. Heuel and Forstner (2001) combined projective geometry and a statistical hypotheses test to reconstruct the 3D line segments. Martinec and Pajdla (2003) reconstructed lines by factoring a matrix containing line correspondences. Jain et al. (2010) used connectivity constraints to reconstruct the line segments from different stereo images independently; the partial reconstructions are then merged into a global result. Chen and Wang (2011) first detected 2D line segments from photos and generated a 3D point cloud by the Structure From Motion (SFM) methods (Snavely et al., 2006). Then, the 3D lines are reconstructed by applying the weak matching method both on 2D photos and a 3D point cloud. The false 3D line segments are filtered via a plane-clustering algorithm. Ceylan et al. (2012) generated 3D lines from image-level edges of urban buildings and then used these lines to simultaneously detect symmetric line arrangements while refining the 3D building model.

Most of the above algorithms use line matching to reconstruct 3D lines. Generally speaking, line matching is a difficult task due to its lacks of geometric constraints. In contrast, via our approach, the images are generated from a point cloud. Because depth information is already known, our approach does not need to apply line matching between multiple images.

2.3. Line segment detection for 3D point clouds

Few papers concentrate on line segment detection for 3D point clouds. Lu et al. (2008) combined the RANdom SAmple Consensus (RANSAC) method and Mahalanobis distance to detect 3D lines.

Borges et al. (2010) proposed an approach for extracting plane intersection lines and depth discontinuity lines from a laser point cloud. To extract plane intersection lines or depth discontinuity lines, their method first classifies the planar points or potential edge points by analyzing the eigenvalues of the covariance matrix that are defined by each point's k-nearest neighbors. Then a region growing procedure iteratively merges planar points or potential edge points into clusters according to the similarity of their normals or line directions. Moghadam et al. (2013) used the same method to extract the plane intersection lines.

However, the results of the method in Lu et al. (2008) are unsatisfactory, because only a few line segments are extracted. Alternately, plane-based methods, such as those of Borges et al. (2010) and Moghadam et al. (2013), have the following drawbacks: (1) It is difficult to determine the boundary of the planes. (2) The task to fit the small and narrow plane is difficult. With a noisy point cloud, the task becomes even more difficult. (3) Moreover, when the data become complex, this type of method may generate unexpected lines at the non-planar surfaces.

Another possible approach is to first extract the sharp features along linear structures; then group them into lines. For details about methods for sharp feature extraction, see Daniels et al. (2007), Weber et al. (2010), and Altantsetseg et al. (2013). However, methods for sharp feature extraction are not robust to noise, because the regions with sharp features and noisy areas have similarly high surface gradients, which are difficult to distinguish. Additionally, it is difficult to design a line-grouping algorithm in 3D space.

3. Overview

This section first introduces the concepts of 3D line-support regions and Line-Segment-Half-Planes (LSHP) structures, and then provides an overview of our approach.

3.1. 3D line-support regions and Line-Segment-Half-Planes (LSHP) structures

For line segment detection in images, Burns et al. (1986) first introduced the term "line-support region," which is defined as a straight region in 2D image space whose pixels share a similar gradient angle. The line-support region has proved to be a robust descriptor to extract line segments in images.

In the context of a point cloud, we define a 3D line-support region as the point set nearest the intersection of two planes, and propose a Line-Segment-Half-Planes (LSHP) structure to fit the 3D line-support region. As illustrated in Fig. 2, the LSHP of a 3D line-support region is characterized by a pair of tangential 3D rectangles. Each rectangle is described by the following parameters: width, length, normal vector and position. A line segment is obtained by computing the intersection of two rectangles. The LSHP structure modeling procedure is given in Section 5.

An example of a 3D line-support region and its Line-Segment-Half-Planes structure is shown in Fig. 3.

3.2. Framework

As shown in Fig. 4, there are three steps to our framework, as follows:

3.2.1. 3D line-support region extraction

Given an unorganized point cloud, $\mathcal{P}$, as input (Fig. 4(a)), $\mathcal{P}$ is projected into a collection of shaded images (Fig. 4(b)). Evenly distributed viewpoints are generated on a sphere surrounding $\mathcal{P}$. For each viewpoint, $\mathcal{P}$ is projected onto a shaded image (Fig. 4(b)). Then, the LSD algorithm (Von Gioi et al., 2010) is applied to extract the 2D line-support regions (the red[1] rectangles in Fig. 4(c)). These extracted 2D line-support regions are then back-projected into the original point cloud as the 3D line-support regions.

3.2.2. LSHP modeling

For each 3D line-support region, we first project the region onto a plane along its principal axis. Then, we employ a method based on dynamic programming to find the "V" shapes in 2D space. Such a "V" shape is fitted by the LSHP structure to validate the 3D line-support region and maintain the accuracy of the line segment. The result of the LSHP structures is shown in Fig. 4(d); the locations where the colors overlap indicate the overlap of LSHP structures from different views.

3.2.3. Refinement of 3D line-support regions and LSHP structures

As shown in Fig. 4(d), many LSHP structures share the same straight linear structure. Therefore, a combination procedure is performed to merge these 3D line-support regions and LSHP structures. Then, the boundary of the LSHP structures is determined by a region growing procedure. The final LSHP structures are shown in Fig. 4(e).

4. 3D line-support region extraction

2D line-support region extraction in images has already been well investigated. The traditional methods group the edge pixels with similar gradient directions into line segments. However, although there are several methods that can extract sharp feature points in 3D space, there is not, to our best knowledge, a method to group these feature points into line segments, especially for scenes containing vegetation and other non-manifold objects.

To cope with this problem, we take the 2D line-support region extraction into account by converting the point cloud into a collection of projected images with different viewpoints. These viewpoints are evenly placed on a sphere surrounding the point cloud. The image-based method reduces the dimensionality and avoids the 3D neighborhood search which is time-consuming for large scale point data. The image-based method also provides a good grouping result for 3D line-support region extraction.

The resolution of the projected images is defined by a parameter, *Res*, and the number of projected images is defined by *K*. Section 7 describes the results of the experiments versus variations of *Res* and *K*.

4.1. Projected images from a point cloud

To obtain a better 2D line-support region result from the projected image, the image must contain clear edge information, i.e., the line structure in a 3D point cloud must correspond to the gray scale variation in the projected images.

Global illumination models, e.g., ambient occlusion, are often used to emphasize the relief of surfaces and edge information. However, such models often require complicated pre-computations and need extra information, such as the normal vector, and thus are unsuitable for large-scale unorganized point clouds. To solve this problem, we adopt Eye Dome Lighting (EDL) (Boucheny, 2009), which uses only depth buffer information and is performed in image coordinate space. Other techniques, such as Screen-Space Ambient Occlusion (SSAO) (Kajalin, 2009), can also be adopted for our situation, but EDL provides better edge perception (see Fig. 5).

[1] For interpretation of color in Figs. 4 and 8, the reader is referred to the web version of this article.

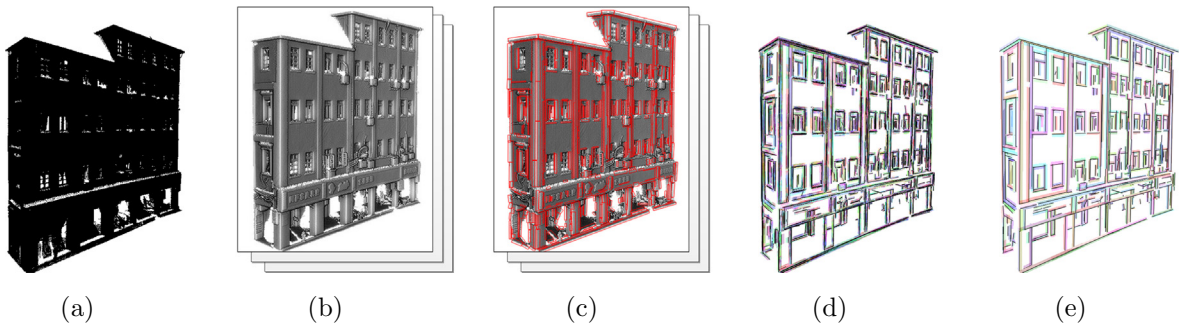


Fig. 4. An overview of our method's pipeline. (a) Original unorganized point cloud. (b) Multi-view images with distinguished edge information. (c) Extracted 2D line-support regions from images. (d) LSHP structures of the extracted line-support regions. (e) The refined LSHP structures.

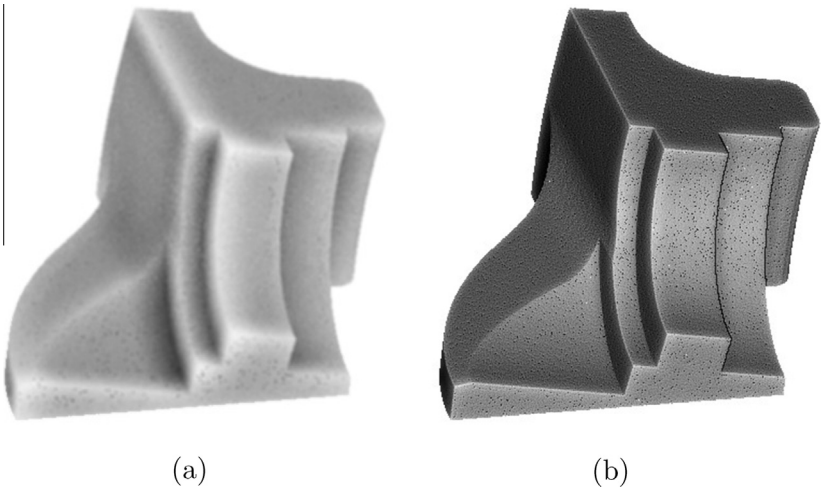


Fig. 5. The comparison results between Crytek Screen-Space Ambient Occlusion (a) and Eye Dome Lighting (b). The EDL provides better edge perception.

EDL is a simple and efficient image-based non-photorealistic shading technique to improve depth perception in scientific visualization images. The shading value for each pixel, p, in the depth image is defined as the amount of occlusion it receives from its neighboring pixels and is expressed as follows:

$$S(p) = \exp(-A * \sum_{q \in V_p} s(p,q)), \tag{1}$$

where $A = 100$ is the factor to simply amplify the intensity of shading; V_p is the neighboring pixels of p; and $s(p,q)$ is the function to measure the occlusion of pixel p against pixel q:

$$s(p,q) = \max\left(\frac{z_p - z_q}{d_{pq}}, 0\right). \tag{2}$$

Here, $z_p \in [0,1]$ and $z_q \in [0,1]$ are the normalized depth values of pixels p and q, respectively; and d_{pq} is the Euclidean distance between pixel p and q.

To take into account the contributions of farther neighboring pixels but avoid increasing the time cost significantly, a multi-resolution approach that applies the same shading function at half and quarter image size resolution is used. Then, a cross filter (Eisemann and Durand, 2004) is used to limit aliasing induced by the lower resolutions. The final image, I, is obtained by weighting these multi-resolution images:

$$I = \frac{4I_0 + 2I_1 + I_2}{7}, \tag{3}$$

where, I_0, I_1 and I_2 are the full, half and quarter resolution shaded images, respectively.

4.2. 2D line-support region extraction

Because it is the state-of-the-art method for image line segment detection, we choose the LSD algorithm (Von Gioi et al., 2010) to extract 2D line-support regions. The LSD algorithm extracts the accurate line segments and 2D line-support regions simultaneously; it does not need to change the values of the parameters. Moreover, the method is fast, and, therefore, efficient, for our multi-view shaded images.

5. LSHP modeling

Each 3D line-support region is fitted by an LSHP structure. The LSHP structure is used to validate the 3D line-support region and 'provide a geometric constraint for line segments.

It is difficult to directly fit the LSHP structure for a 3D line-support region in 3D space. Instead, we can take advantage of the corresponding 2D line-support regions to ascertain an optimal projection direction of the 3D line-support region (Fig. 6(a)). By projecting the 3D line-support region along the projection direction (Fig. 6(b)), we extract the "V" shape from the projected points. The "V" shape is a point set that can be approximated by two line segments that share an endpoint. Such a "V" shape is likely to be the projection of an LSHP structure along its intersection line direction. To find the "V" shape, an approach based on dynamic programming is performed. Once the "V" shape is determined, the projected points with the "V" shape are divided into two subsets according to the corner point (Fig. 6(c)). Correspondingly, the 3D line-support region is also partitioned into two subsets. These

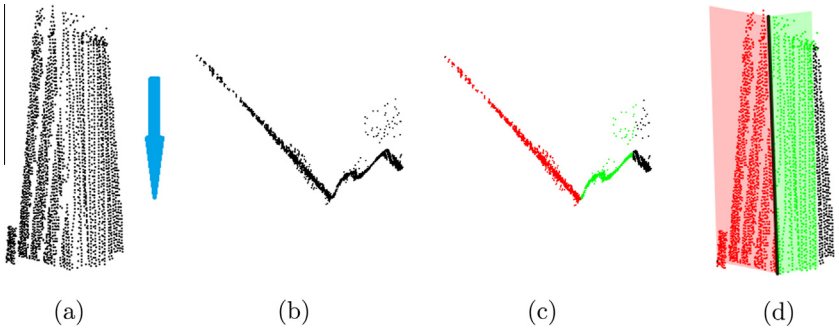


Fig. 6. The LSHP structure of a 3D line-support region is constructed via projective analysis. (a) Given a 3D line-support region, the arrow represents its projection direction. (b) The projected points of the 3D line-support region. (c) The extracted "V" shape, which, is divided into two parts, respectively drawn in red and green. (d) The corresponding 3D line-support region and LSHP structure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

two subsets are fitted separately by two planes. After validation, the LSHP model is constructed using these two planes (Fig. 6(d)).

Through projective analysis, such a construction strategy for the LSHP structure is robust and dramatically reduces the computational complexity. The following three subsections provide further details.

5.1. Finding the projection direction

When a 3D line-support region is projected along a direction v, an aggregation degree of these projected points can be calculated. By considering each projected point as a 2D circle, the aggregation degree can be regarded as the union area of these circles (projected points). The direction v that has the minimum aggregation degree is desired.

As illustrated in Fig. 7, a 3D line-support region L and its corresponding 2D line-support region l in the image I are given. For the sake of convenience, we assume that image I lies on the XY plane and l is parallel to the y-axis in the image coordinate system (if not, we can rotate the point cloud to guarantee it). Thus, the projection direction v can be parameterized by $v = (0, \cos\alpha, \sin\alpha)$, where α is the angle between L and l. Then, the normal vector $\hat{n}$ of L's tangent plane Ω is given by $\hat{n} = (0, \sin\alpha, -\cos\alpha)$. The deviation of the distances between each point of L to Ω can be defined as a function with respect to the unary variable α, i.e.,

$$f(\alpha) = \sigma_{p\in L}(dist(p, \Omega)), \tag{4}$$

where $dist(p, \Omega)$ is the Euclidean distance from p to Ω, and $\sigma(\cdot)$ is the standard deviation.

A smaller value of $f(\alpha)$ indicates a better aggregation degree of the projected points. Ideally, the function $f(\alpha)$ can be treated as a unimodal function versus α; i.e., for some value m, $f(\alpha)$ is monotonically decreasing for $alpha \leqslant m$ and monotonically increasing for $\alpha \geqslant m$, where $\alpha \in (-90, 90)$. In that case, the value of α that minimizes $f(\alpha)$ can be easily found via ternary search or Golden section search (Kiefer, 1953). Once α is determined, the projection direction v can be computed accordingly.

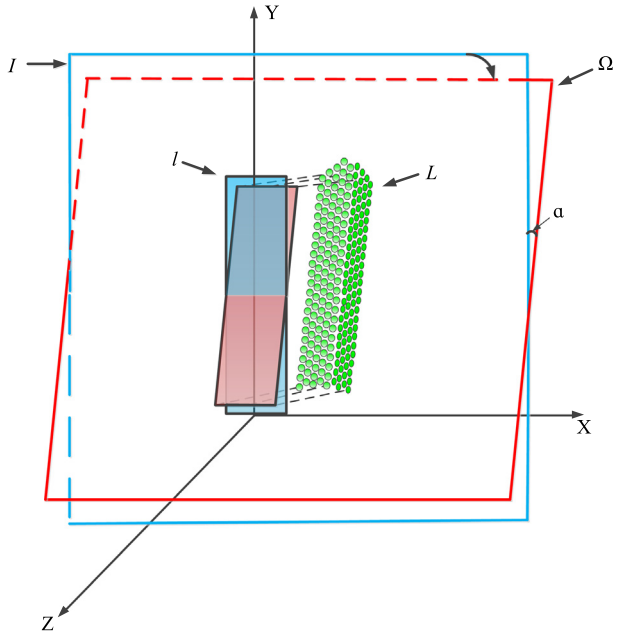


Fig. 7. Finding the primary axis of 3D line-support region.

5.2. "V" shape extraction

Because the normal vector of the tangent plane of a given 3D line-support region is $\hat{n} = (0, \sin\alpha, -\cos\alpha)$ and the projection direction is $v = (0, \cos\alpha, \sin\alpha)$, this implies that the tangent line of the projected points is parallel to the x-axis. Therefore, we can use the distance from each projected point to the tangent line (i.e., the y coordinate value) to analyze the shape of the projected points. We first divide the projected points into n pieces with the same width along the x-axis. The center point of each piece is computed. By storing the y value of the center point associated with the i-th piece in $S[i]$, we obtain a sequence $S[1, \ldots, i]$, which can be used to find the corner point of the projected points.

Let LIS_i be the longest increasing subsequence of $S[1, \ldots, i]$ with the endpoint at S_i, and let $LISR_i$ be the longest increasing subsequence of the reverse sequence $S[n, \ldots, i]$ with the endpoint at S_i. Similarly, LDS_i and $LDSR_i$ are defined as the longest decreasing subsequences of $S[1, \ldots, i]$ and $S[n, \ldots, i]$ with the endpoints at S_i, respectively. The corner point of a "V" shape is computed by finding the index i according to the following equation:

$$\arg\max_{i=1,\ldots,n}(|LIS_i| \times |LISR_i|, |LDS_i| \times |LDSR_i|), \tag{5}$$

where $|\cdot|$ is the size of the subsequence. The "V" shape consists of the points that lie between the pieces of $LIS_i(orLDS_i)$ and $LISR_i(orLDSR_i)$. The center point of i-th piece can be regarded as the corner point of the "V" shape, as it divides the "V" shape into two parts with different monotonicity.

The longest increasing or decreasing subsequences can be solved for by dynamic programming with time complexity $O(nlogn)$ (Fredman, 1975), where n is the number of pieces. Here, n is set to be 100.

5.3. LSHP fitting from the "V" shape

As introduced in Section 3, the LSHP structure is characterized by two tangential rectangles. The LSHP structure can be easily constructed based on the "V" shape. This is achieved by fitting the 3D line-support region into two separated planes according to their projected points in the "V" shape. Denote β as the bisecting plane that bisects the two fitted planes. The 3D line-support region is redivided into two parts by β. Then, the two tangential rectangles can be represented by two planes, and the width of each rectangle is defined as the longest distance from the points on the plane to the intersection line of the two planes.

5.3.1. Least median of squares

The least median of squares (LMS) method is employed to fit the planes. The LMS is a robust regression method that can tolerate up to 50% outliers without requiring pre-set thresholds. The LMS method estimates the parameters of plane β by minimizing the median of the square of the residuals r, which is defined as the distance from the points to the sample plane. Thus, the regression plane β of points $\mathcal{P}$ is estimated by the following equation:

$$\arg\min_{\beta} \underset{p\in\mathcal{P}}{median}(dist(p,\beta)). \tag{6}$$

Eq. (6) can be solved by the following random sampling algorithm. First, m random subsamples of k different points are generated. For each subsample $\mathcal{P}_i$, a plane β_i is fitted by these points. The residual r_i is determined by the median of the distances from each point $p \in \mathcal{P}$ to the plane β_i, and the plane β_i with minimal r_i is retained as the final plane. The value of m can be determined by $P = 1 - (1-\epsilon^k)^m$ according to Rousseeuw and Leroy (2005), where ϵ is the fraction of outliers and P is the probability that at least one of the m subsamples is good. In our implementation, we assume that $k = 3$ and $\epsilon = 50\%$. Therefore, with the requirement of $P = 0.999$, we obtain the number of samples as $m = 51$.

5.3.2. 3D Line-support region validation

A threshold θ for the distance from the point set to the regression plane is used to check if the regression plane can fit its points. A regression plane β is considered to fit the points P if they satisfy:

$$\underset{p\in P}{median}(dist(p,\beta)) \leqslant \theta. \tag{7}$$

Although θ is related to the accuracy of a laser device, it can be estimated by the average distance between the nearest neighbor points, i.e.,

$$\theta = \frac{\sum_{p\in\mathcal{P}}(\min_{q\in\mathcal{P}\wedge q\neq p}(\|p-q\|))}{|\mathcal{P}|}, \tag{8}$$

where $\mathcal{P}$ is the original point cloud.

6. Refinement of 3D line-support regions and LSHP structures

Because the 3D line-support regions are obtained from multi-view images, there are many overlaps. It is necessary to combine the 3D line-support regions and LSHP structures that share the same line segment. At the same time, the boundaries of the 3D line-support regions also need to be refined in 3D space.

6.1. Combination

Let $\mathcal{L} = \{L_1, \ldots, L_n\}$ be the n extracted 3D line-support regions. Each 3D line-support region L_i is divided into two subsets L_i^1 and L_i^2, according to its two associated rectangles of the LSHP structure. $\mathcal{L}$ is sorted in descending order according to the confidence score s_i of each L_i. The confidence score s_i is defined by $s_i = |L_i^1||L_i^2|$ for L_i. A high value of s_i indicates that the half-planes of its LSHP are reliable.

For each pair $L_i (i = 1, \ldots, n)$ and $L_j (j = i+1, \ldots, n)$ that are adjacent, i.e., $L_i \cap L_j \neq \emptyset$: If both subsets of L_i are coplanar with those of L_j, we merge L_j into L_i, i.e., $L_i \leftarrow L_i \cup L_j$. To check whether two subsets are coplanar, equation (7) is adopted. On the contrary, if there are two subsets, L_i^a of L_i and L_j^b of L_j, that share most of their points ($|L_i^a \cap L_j^b| > 0.5|\min(L_i^a, L_j^b)|$ in detail) but are not coplanar, we call L_i and L_j conflicting. In this case, the 3D line-support region with a lower confidence score will be removed from $\mathcal{L}$.

The combination is repeated until there are no changes in the 3D line-support regions of $\mathcal{L}$.

6.2. 3D Line-support region growing

After combination, a region growing procedure is carried out to determine the boundary of the 3D line-support regions. A 3D line-support region grows respectively on its two subsets with the growing direction along the line segment of its LSHP.

Given a 3D line-support region L, assume that plane β bisects the angle between two half-planes of L's LSHP. Thus, β divides L and the original point cloud $\mathcal{P}$ into two components. Let $L' \subset L$ and $\mathcal{P}' \subset \mathcal{P}$ lie on the same side of β. The r-neighborhood of $p \in L'$ is defined by:

$$\mathcal{N}(p,r) = \{q \in \mathcal{P}' | \|p-q\| \leqslant r\}. \tag{9}$$

For each point $p \in L'$, we first compute its neighborhood $\mathcal{N}(p, \frac{w}{2})$, where w is the half width of L''s approximation rectangle. To guarantee that the width of the rectangle remains the same after growing, only the points whose distance to the line segment are smaller than w are grown. An example is given in Fig. 8(a), in which, only the red points of the sphere are grown. We denote these points as $\mathcal{N}'$. If $\mathcal{N}'$ and the regression plane of L' satisfy Eq. (7), we add $\mathcal{N}'$ into L'. The growing procedure is repeated until there are no new points that can be added into L'. The result after growing is shown in Fig. 8(b).

7. Experimental results

7.1. Environment

The method was evaluated using typical street-scene LiDAR point clouds acquired by a RIGEL VMX-450 MLS system; the average density of the point clouds used is approximately 2500 points/m^2. We also tested our method on the classical "sharp sphere" point cloud data.

Information on the test point clouds is summarized in Table 1. The first column of the table is the name of the data, the second column is the figure number of the point clouds as presented in this paper, the third column is the number of points, and the fourth column is the average distance of nearest neighboring points within the point cloud (equal to θ in Eq. (7)).

Our method was implemented in C++ and OpenGL. It was executed on a personal computer with Intel Core(TM) i5-3470 3.2 GHz CPU and 12.0 GB RAM. For the implementation of the LSD algorithm, the reader is referred to Von Gioi et al. (2012), and the algorithm's source code can be found at http://iie.fing.edu.uy/jirafa/lsd.

7.2. Parameters

Only two parameters, the number of viewpoints K and resolution Res, are decided by the users; the others are considered internal parameters and, can be used without tuning.

We first demonstrated the influence of *Res*. Here, we set $K = 128$. As shown in Fig. 9, the value of *Res* depends on the details that the user wants to extract. If *Res* is too large, the features may be missing because holes would arise in the renderings; however, if *Res* is too small, details will be lost. Here, we simply set *Res* according to the following equation.

$$Res = \frac{d}{4\bar{r}} \times \frac{d}{4\bar{r}}, \tag{10}$$

where d is the diagonal length of the bounding box of the input points and $\bar{r}$ is the average distance between the nearest neighboring points (equal to θ in Eq. (8)).

Next, we examined the influence of K. As we know, a larger K means more views, and more points will be visible. However, the time cost will increase along with the increase of K. We demonstrate the percentage of visible points under different numbers of viewpoints K on four point clouds. Here, we use the method proposed by Katz et al. (2007) to check if a point is visible via a specific viewpoint. From Fig. 10, one can observe that when K is large enough, i.e., $K \geqslant 96$, the percentage of invisible points is smaller than 0.5%; this small amount of invisible points can be ignored. Fig. 11 shows the influence of different K values on the number of detected LSHP structures. As seen from Fig. 11, there is an upward trend in the number of detected LSHP structures, and a considerable increase occurred from $K = 32$ to $K = 128$. When $K > 128$, the rate of increase slows down, and it tends to remain stable when $K > 224$. Considering the computation time factor, $K = 128$ is a reasonable configuration.

Finally, we chose $Res = \frac{d}{4\bar{r}} \times \frac{d}{4\bar{r}}$, and $K = 128$. Except for the examples in Fig. 9, in which we demonstrate the influence of parameter changes, these parameters were applied with fixed configurations throughout all presented figures.

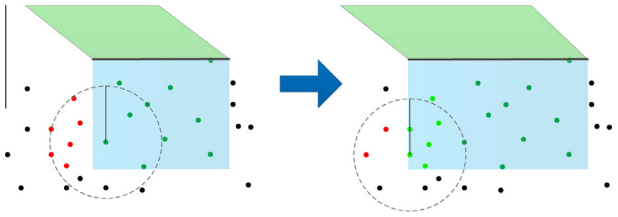


Fig. 8. Illustration of 3D line-support region growing.

Table 1
Information of the test point clouds.

Data	Figs.	# points	$\bar{r}$
Data1	1	781,297	0.014
Data2	9	3,725,565	0.017
Data3	12	1,000,000	0.023
Data4	13	810,190	0.059
Data5	14	3,906,174	0.015
Data6	15	6,043,282	0.013
Data7	16	4,802,365	0.0075

7.3. Extraction results

The results in previous sections have already demonstrated that our method has the ability to handle raw scan point clouds of cluttered scenes (Figs. 1 and 9). Here, we present more results on raw scan point clouds for common scenes in urban areas, such as buildings (Figs. 13 and 16) and roads (Fig. 15); we also show the ability of our method to handle point clouds with sharp curves (Fig. 12).

To quantitatively evaluate the accuracy of the line segment extraction results, we introduce three indices, *Completeness* (*Comp*), *Correctness* (*Corr*) and *Quality*, from (Rutzinger et al., 2009). *Completeness* is the true positive detection rate of line segments, and *Correctness* indicates how well the line segments are detected. They can be derived as

$$Comp = \frac{|TP|}{|N|}, \tag{11}$$

$$Corr = \frac{|TP|}{|M|}, \tag{12}$$

where $|TP|$ is the number of true positive detected line segments, $|N|$ is the total number of line segments in the ground truth, and $|M|$ is the total number of detected line segments. The *Quality* metric provides a balance of completeness and correctness, i.e.,

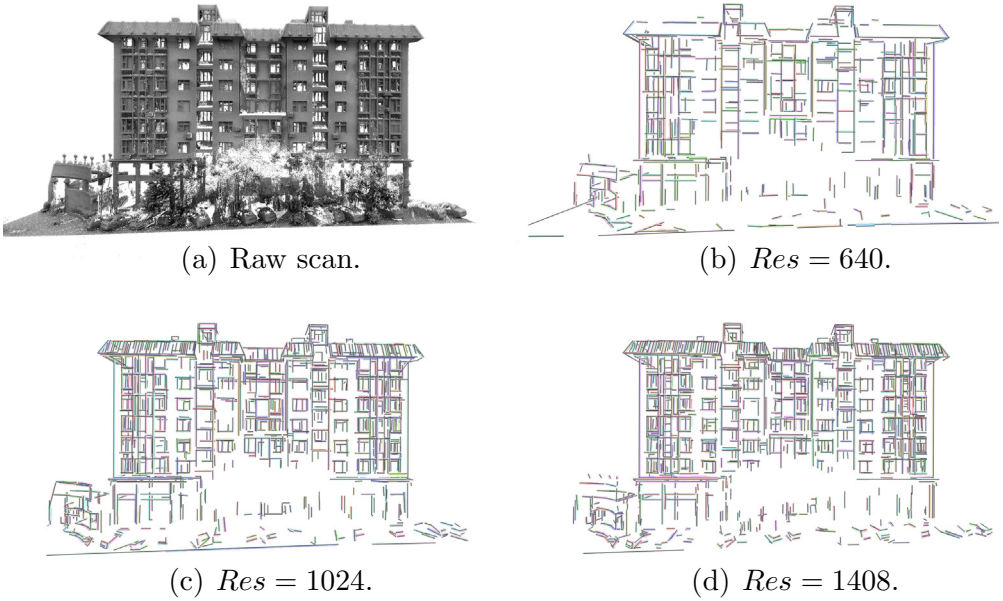


Fig. 9. Extracted 3D line-support regions by different *Res*.

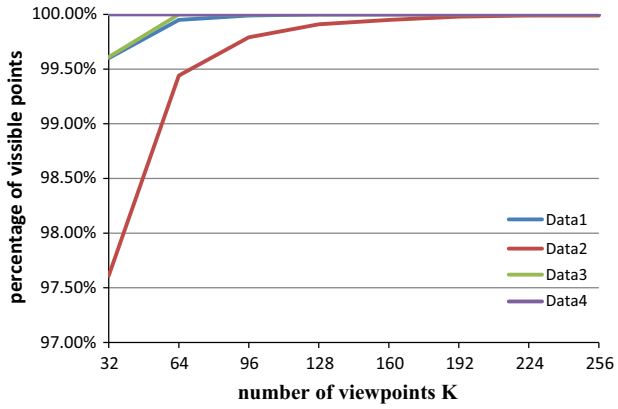


Fig. 10. The percentage of visible points under different numbers of viewpoints K.

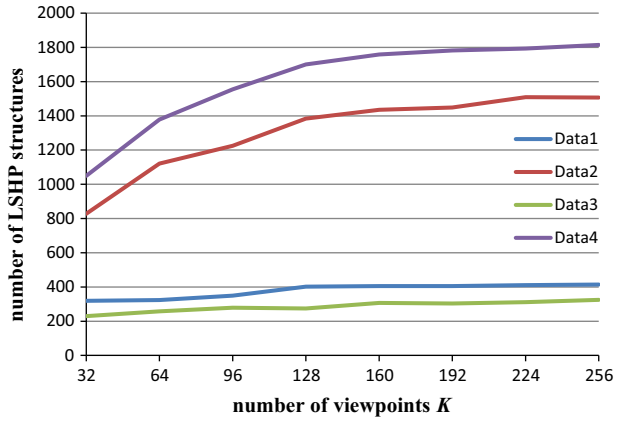


Fig. 11. Influence of different K values on the number of detected LSHP structures.

$$Quality = \frac{Comp \cdot Corr}{Comp + Corr - Comp \cdot Corr}. \tag{13}$$

Let $l\_dist(l_1, l_2)$ be the distance measurement from line segment l_1 to line segment l_2, which is formalized as:

$$l\_dist(l_1, l_2) = \frac{1}{n}\sum_{i=1}^{n} dist(p_i, l_2), \tag{14}$$

where $p_1, \ldots, p_n$ are n points that are evenly picked from l_1 (here n is set to $\lfloor \frac{5|l_2|}{\bar{r}} \rfloor$); and $dist(p, l)$ is the minimum Euclidean distance from point p to line segment l. For each detected line segment l, it is considered to be a true positive if its distance to the nearest line segment in $\mathcal{L}_{best}$ is smaller than $5\bar{r}$, i.e.,

$$min_{l_2 \in \mathcal{L}_{best}} l\_dist(l_1, l_2) < 5\bar{r}, \tag{15}$$

where $\mathcal{L}_{best}$ is the ground truth line segment set that is acquired manually.

We also propose a measure ε to evaluate the accuracy of the detected LSHP structure. Given a detected line segment l, ε is defined as the average distance from l's 3D line support region to l's LSHP. Here, we define l's 3D line support region as:

$$L = \{p | p \in \mathcal{P} \wedge dist(p, l) < 5\bar{r}\} \tag{16}$$

For each point $p \in L$, its distance to l's LSHP is computed by:

$$lshp\_dist(p, LSHP) = min\{dist(p, \beta_1), dist(p, \beta_2)\}, \tag{17}$$

where β_1 and β_2 are two support half-planes of l's LSHP. Thus, ε can be formalized as:

$$\varepsilon = \frac{1}{|L|}\sum_{p \in L} lshp\_dist(p, LSHP). \tag{18}$$

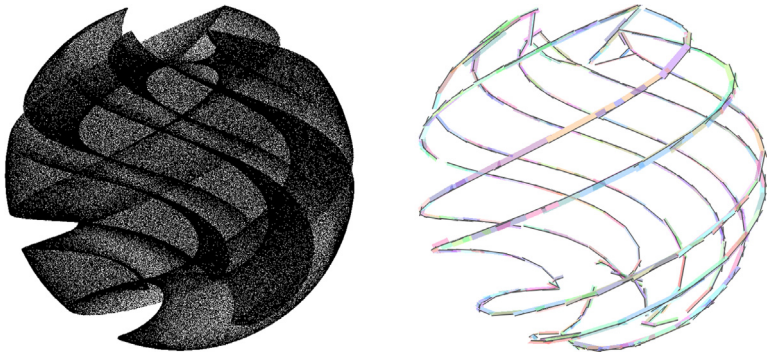
We use $\bar{\varepsilon}$ to represent the average of ε for all detected LSHP structures; it can be regarded as an accuracy measure to estimate the quality of detected LSHP structures. The smaller value of $\bar{\varepsilon}$ is desired.

The accuracy evaluation results are listed in Table 2, where the first column of the table is the name of the test point cloud and the second column is the number of obtained LSHP structures. The measures mentioned before are given in the third through sixth columns. The ground truth in our experiments is acquired manually. On average, the proposed algorithm achieves a completeness greater than 87%, a correctness greater than 95%, and a quality greater than 83%. On the whole, the proposed algorithm extracts line segments accurately from mobile LiDAR point clouds.

The running time (in seconds) is given in the last column of Table 2; only one core of the CPU is used. The typical ratio of points used to perform the nearest neighbor search in 3D is approximately 25% for building scenes, e.g., Data1, and 10% for road scenes, e.g., Data6. Our method can handle 1 M points in an average of $1 \sim 2$ min, which is practical for industrial applications. The speed can be improved by computing the 3D line-support regions for each view in parallel or manually choosing the suitable viewpoints.

7.4. Comparative studies

For comparison, we implemented the method proposed by Borges et al. (2010). The main idea of their method is to first segment the point cloud into planes by a bottom-up procedure, and then extract the line segments that are determined by pairs of



(a) original point cloud (b) extracted LSHP structures

Fig. 12. Extracted LSHP structures of "Sharp sphere".

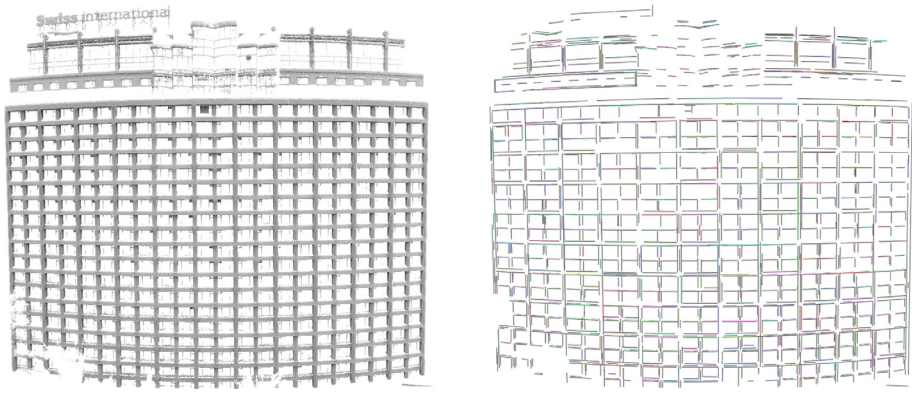


Fig. 13. Extracted LSHP structures of "Swiss Hotel" building.

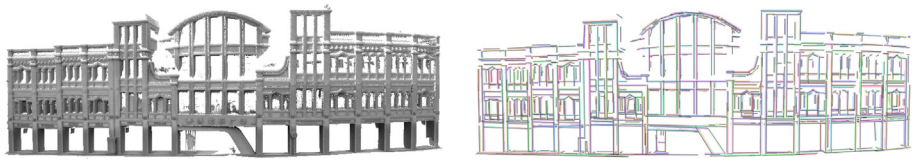


Fig. 14. Extracted LSHP structures of a "ZhongShan Road" building.

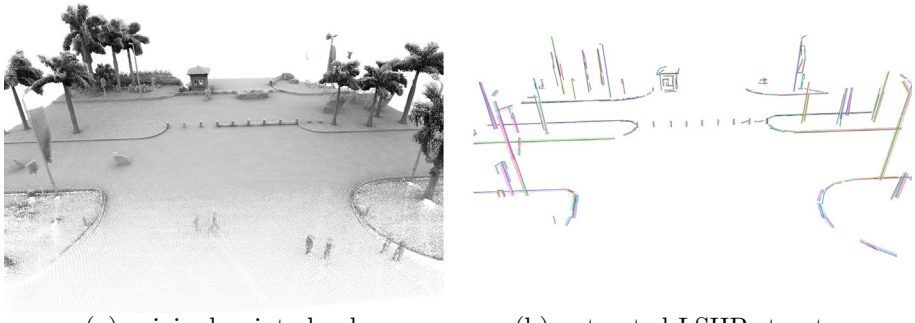


Fig. 15. Extracted LSHP structures of road scene.

Table 2
Accuracy results of the proposed method.

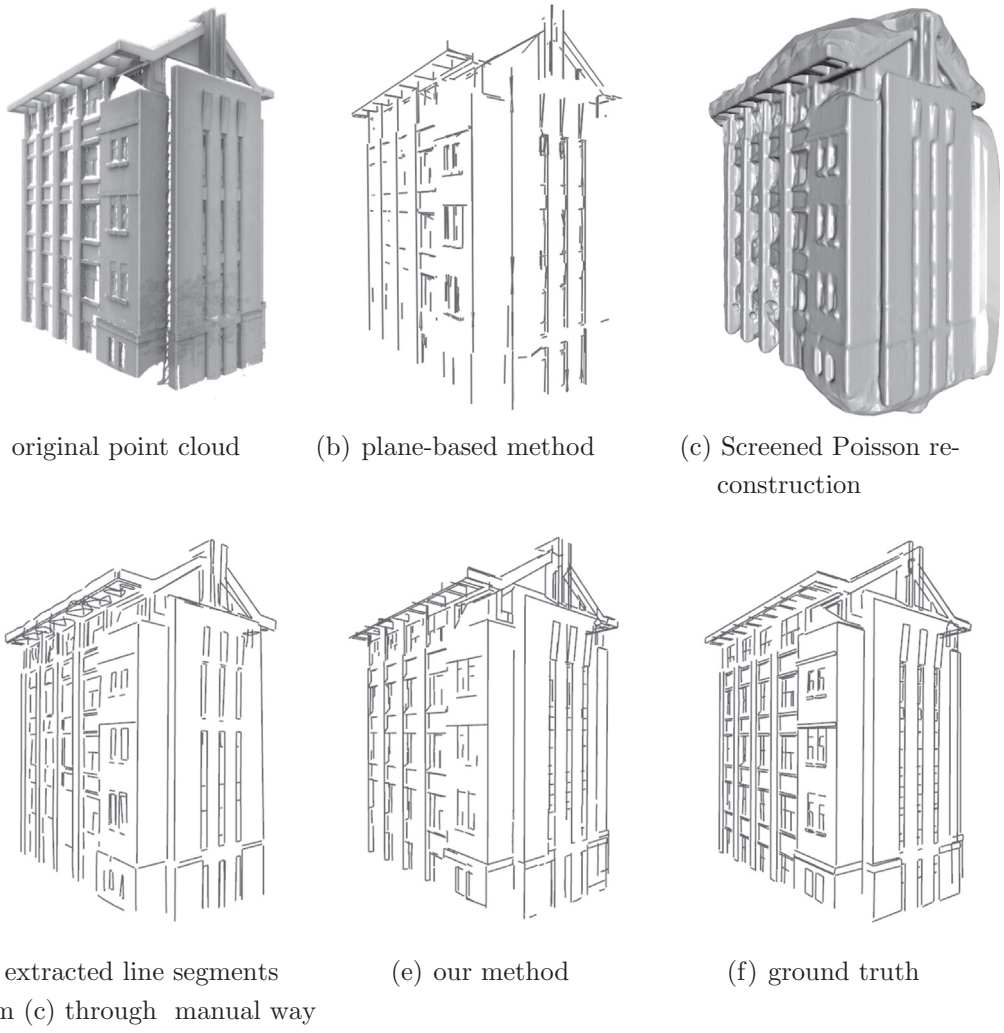
Data	# LSHP	*Comp* (%)	*Corr* (%)	*Quality* (%)	ε	Time (s)
Data1	329	89.3	97.0	86.9	0.0061	94.4
Data2	1384	86.3	90.0	78.7	0.0091	359.0
Data3	266	96.3	97.7	94.2	0.014	59.5
Data4	1700	86.0	92.0	80.0	0.014	222.8
Data5	1164	81.1	97.6	79.5	0.0083	585.8
Data6	286	86.1	96.5	83.5	0.0075	445.1
Data7	558	86.9	95.2	83.1	0.0031	589.4

adjacent planes. The result of this plane-based method is shown in Fig. 16(b), where the parameters are fine-tuned. We also compare our method to reconstruction-based methods. Fig. 16(c) shows the reconstructed mesh model yielded by Screened Poisson reconstruction method (Kazhdan and Hoppe, 2013), and the line segments shown in Fig. 16(d) are extracted manually from Fig. 16(c).

As seen from the line segment extraction results in Fig. 16, our algorithm extracts line segments completely and correctly and obtains better results than the other two methods. The numerical comparison results are given in Table 3, where the value of $\bar{\varepsilon}$ of the Screen Poisson reconstruction method is not presented because the LSHP structure is absent in this method. The reasons the plane-based method can not achieve a high performance are: (1) the boundaries of the planes are difficult to determine, (2) the small and narrow plane is difficult to extract, and (3) the complex data (e.g., plants) is difficult to be fitted by planes.

7.5. Robustness studies

We study the robustness of our method by adding Gaussian noise into the point cloud, and the extraction results are given in Fig. 17. As the noise increases, the processing quality gradually gets worse. We also impose our method on a multi-view stereo point



(a) original point cloud (b) plane-based method (c) Screened Poisson reconstruction

(d) extracted line segments from (c) through manual way (e) our method (f) ground truth

Fig. 16. Extracted line segments by plane-based method (b), reconstruction-based method (d) and our method (e).

Table 3
Accuracy comparison results of plane-based method, reconstruction-based method and our method.

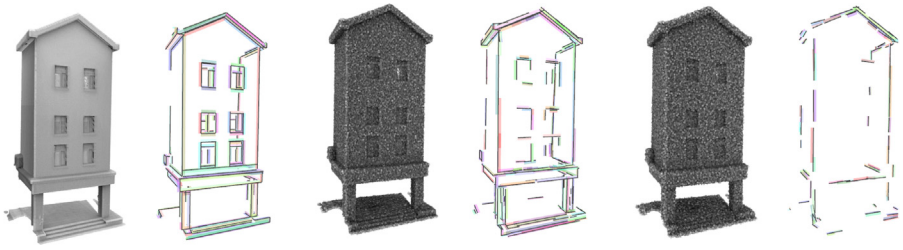
	Plane-based	Screen Poisson reconstruction	Ours
Line segments	567	624	558
True positive	487	377	531
Comp	79.7%	61.7%	86.9%
Corr	85.9%	60.4%	95.2%
Quality	70.5%	43.9%	83.1%
$\overline{\varepsilon}$	0.0061	–	0.0031
Time (s)	170.2	67.1	589.4

cloud with high levels of outliers and noise. Fig. 18 shows the extraction results, in which most of the main line segments are extracted.

8. Application

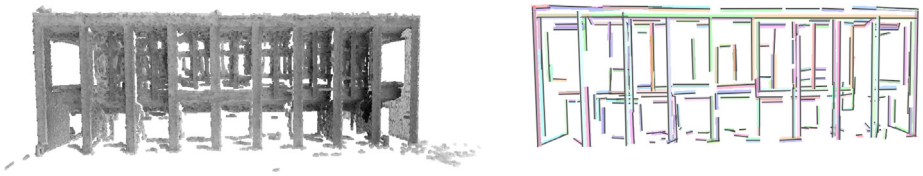
In this section, we illustrate the effectiveness of 3D line-support regions and their relative LSHP structures in urban scene abstraction.

Today, with the widespread use of point clouds acquired by current devices, it is efficient to create large-scale dense point clouds from large outside environments. However, to fully model the



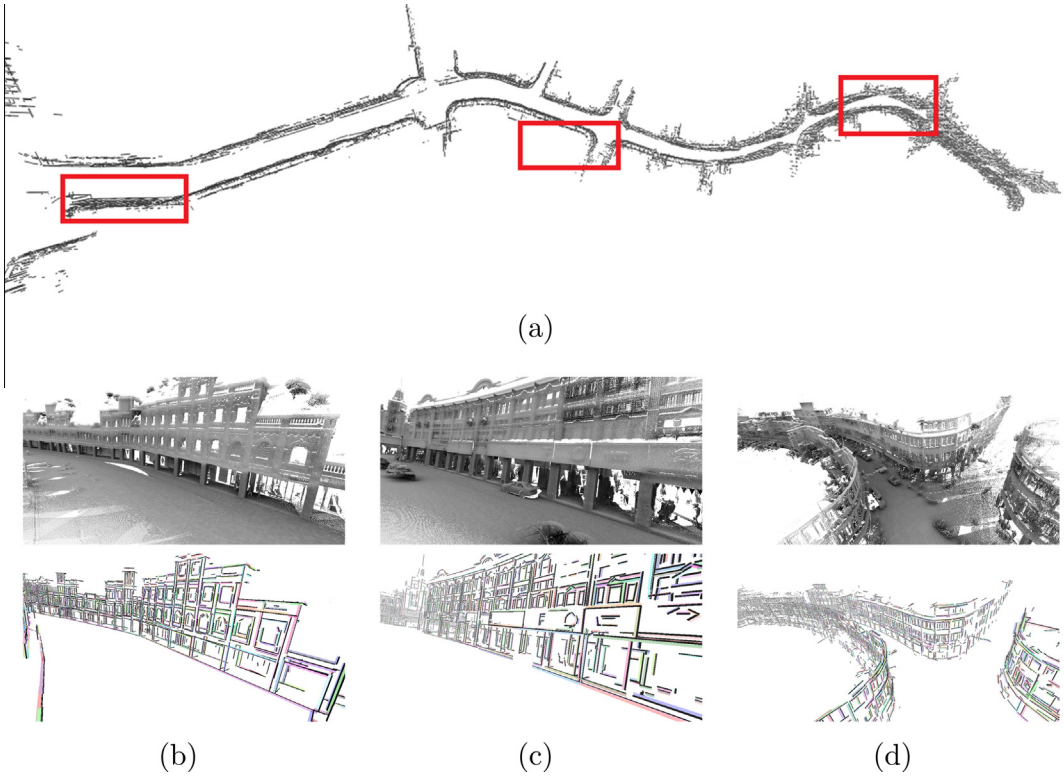
(a) no noise (b) 3cm Gaussian noise (c) 5cm Gaussian noise

Fig. 17. Extracted LSHP structures in presence of noise.



(a) multi-view stereo point cloud (b) extraction result

Fig. 18. Extracted LSHP structures from multi-view stereo point cloud.



(a) (b) (c) (d)

Fig. 19. The overall extracted LSHP structures from a point cloud covering 1.5 km street are shown in (a). The original point cloud is stored in an uncompressed file with a nearly size of 10 GB. And the result is stored in The details in the red boxes are given in the subfigures (b–d) from left to right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

entire scene is time-consuming with the current technology. For other applications, such browsing the data on the Internet via a browser, the bandwidth is the major bottleneck. Our Line-Segment-Half-Planes (LSHP) structure can provide an abstraction of the point cloud, which is a useful tool to solve these problems.

An example of LSHP abstraction for a point cloud covering 1.5 km of a street is given in Fig. 19. The size of the original uncompressed point cloud file is nearly 10 GB, and the number of points is approximately 250 million. These points are manually divided into 86 regions; each region contains nearly 3 M points. The LSHP structures of each region are extracted individually. These structures are then combined into one file. After abstraction, the LSHP result can be stored in a file with a size of only 1.8 MB, and the number of LSHP structures is approximately 40,000. The compression ratio of these binary files is nearly 5000:1. Furthermore, the total computation time is less than 8 h when run on a single core of the CPU.

Due to the high compression ratio, it is possible to transfer the results through narrow bandwidth channels. At the same time, as shown in Fig. 19(b–d), the LSHP structures maintain the major features of the linear structures in the point cloud, which provides a clear overview for users.

9. Conclusion and future work

In this paper, we have proposed an effective line segment extraction method that is capable of accurately extracting line segments from large-scale unorganized raw scan point clouds. The 3D line-support regions are also extracted at the same time. These 3D line-support regions are fitted by our Line-Segment-Half-Planes (LSHP) structure, which provides a geometric constraint for line segments, making the line segments more reliable and accurate.

The proposed method was tested on raw scan point clouds of complex real-world scenes acquired by LiDAR devices. The experimental results show that the proposed algorithm extracts line segments accurately from mobile LiDAR point clouds.

Lastly, we demonstrated the application of 3D line-support regions and their relative LSHP structures on urban scene abstraction.

9.1. Limitations

Though the proposed method works well on the complex real mobile LiDAR point cloud, it has two limitations. Firstly, visibility of the point cloud during rendering is essential for line extraction,

edges that are not visible from the selected views cannot be detected. More sophistical viewpoints selection method will improve the efficiency and robustness. Secondly, the non-photorealistic rendering is sensitive to the rendering resolution, which effected by the density of point cloud and the view parameter used for rendering, adaptive methods are needed to avoid the sparseness between points in the rendered images.

9.2. Future work

In the future, we would like to, through analysis, discover the best image resolution for our multi-view projected images and extend our method to extract small reliefs of surfaces. We also want to further address the viewpoint selection based on the visibility. It is also of interest to apply our method to urban reconstruction.

Acknowledgments

This work was supported by an NSFC Grant (Project No. 61371144) and an NSERC Discovery Grant. The authors would like to thank the anonymous reviewers for their valuable comments.

References

Altantsetseg, E., Muraki, Y., Matsuyama, K., Konno, K., 2013. Feature line extraction from unorganized noisy point clouds using truncated fourier series. Vis. Comput. 29 (6–8), 617–626.

Baillard, C., Schmid, C., Zisserman, A., Fitzgibbon, A., England, O.O., 1999. Automatic line matching and 3d reconstruction of buildings from multiple views. In: ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery 32 (Part 3-2W5). pp. 69–80.

Ballard, D.H., 1981. Generalizing the hough transform to detect arbitrary shapes. Pattern Recogn. 13 (2), 111–122.

Borges, P., Zlot, R., Bosse, M., Nuske, S., Tews, A., 2010. Vision-based localization using an edge map extracted from 3d laser range data. In: IEEE International Conference on Robotics and Automation. IEEE, pp. 4902–4909.

Boucheny, C., 2009. Interactive Scientific Visualization of Large Datasets: Towards a Perceptive-Based Approach (Ph.D. thesis). Joseph Fourier University.

Burns, J.B., Hanson, A.R., Riseman, E.M., 1986. Extracting straight lines. IEEE Trans. Pattern Anal. Mach. Intell. 8 (4), 425–455.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8 (6), 679–698.

Ceylan, D., Mitra, N.J., Li, H., Weise, T., Pauly, M., 2012. Factored Facade Acquisition Using Symmetric Line Arrangements. Wiley Online Library, pp. 671–680.

Chen, T., Wang, Q., 2011. 3d line segment detection for unorganized point clouds from multi-view stereo. In: Computer Vision–ACCV 2010. Springer, pp. 400–411.

Daniels, J., Ha, L.K., Ochotta, T., Silva, C.T., 2007. Robust smooth feature extraction from point clouds. In: IEEE International Conf. on Shape Modeling and Applications. IEEE, pp. 123–136.

Desolneux, A., Moisan, L., Morel, J.-M., 2000. Meaningful alignments. Int. J. Comput. Vis. 40 (1), 7–23.

Eisemann, E., Durand, F., 2004. Flash photography enhancement via intrinsic relighting. ACM Trans. Graph. 23 (3), 673–678.

Fredman, M.L., 1975. On computing the length of longest increasing subsequences. Discr. Math. 11 (1), 29–35.

Heuel, S., Forstner, W., 2001. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. Proceedings IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE, pp. 517–524.

Jain, A., Kurz, C., Thormahlen, T., Seidel, H.-P., 2010. Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In: Proceedings IEEE International Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1586–1593.

Kajalin, V., 2009. Screen space ambient occlusion. Shader X^7, 413–424.

Katz, S., Tal, A., Basri, R., 2007. Direct visibility of point sets. ACM Trans. Graph. 26 (3), 24:1–24:12.

Kazhdan, M., Hoppe, H., 2013. Screened poisson surface reconstruction. ACM Trans. Graph. 32 (3), 29:1–29:13.

Kiefer, J., 1953. Sequential minimax search for a maximum. Proceedings of the American Mathematical Society 4 (3), 502–506.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. Int. J. Comput. Vis. 99 (1), 69–85.

Lu, Z., Baek, S., Lee, S., 2008. Robust 3d line extraction from stereo point clouds. In: IEEE Conference on Robotics, Automation and Mechatronics. IEEE, pp. 1–5.

Martinec, D., Pajdla, T., 2003. Line reconstruction from many perspective images by factorization. Proceedings IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1. IEEE, pp. 497–502.

Moghadam, P., Bosse, M., Zlot, R., 2013. Line-based extrinsic calibration of range and image sensors. In: IEEE International Conference on Robotics and Automation. IEEE, pp. 3685–3691.

Rousseeuw, P.J., Leroy, A.M., 2005. Robust Regression and Outlier Detection. John Wiley & Sons Inc., New York.

Rutzinger, M., Rottensteiner, F., Pfeifer, N., 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. 2 (1), 11–20.

Snavely, N., Seitz, S.M., Szeliski, R., 2006. Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. 25 (3), 835–846.

Taylor, C.J., Kriegman, D.J., 1995. Structure and motion from line segments in multiple images. IEEE Trans. Pattern Anal. Mach. Intell. 17 (11), 1021–1032.

Von Gioi, R.G., Jakubowicz, J., Morel, J.-M., Randall, G., 2010. Lsd: A fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. 32 (4), 722–732.

Von Gioi, R.G., Jakubowicz, J., Morel, J.-M., Randall, G., 2012. Lsd: a line segment detector. Image Process. On Line 2, 35–55.

Weber, C., Hahmann, S., Hagen, H., 2010. Sharp feature detection in point clouds. In: Shape Modeling International Conference. IEEE, pp. 175–186.

Woo, D.-M., Park, D.-C., Han, S.-S., 2009. Extraction of 3d line segment using disparity map. In: IEEE International Conference on Digital Image Processing. IEEE, pp. 127–131.

Yang, B., Fang, L., Li, J., 2013. Semi-automated extraction and delineation of 3d roads of street scene from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 79, 80–93.