Facet Segmentation-Based Line Segment Extraction for Large-Scale Point Clouds

Yangbin Lin, Cheng Wang, *Senior Member, IEEE*, Bili Chen, Dawei Zai, and Jonathan Li, *Senior Member, IEEE*

***Abstract*— As one of the most common features in the man-made environments, straight lines play an important role in many applications. In this paper, we present a new framework to extract line segments from large-scale point clouds. The proposed method is fast to produce results, easy for implementation and understanding, and suitable for various point cloud data. The key idea is to segment the input point cloud into a collection of facets efficiently. These facets provide sufficient information for determining linear features in the local planar region and make line segment extraction become relatively convenient. Moreover, we introduce the concept "number of false alarms" into 3-D point cloud context to filter the false positive line segment detections. We test our approach on various types of point clouds acquired from different ways. We also compared the proposed method with several other methods and provide both quantitative and visual comparison results. The experimental results show that our algorithm is efficient and effective, and produce more accurate and complete line segments than the comparative methods. To further verify the accuracy of the line segments extracted by the proposed method, we also present a line-based registration framework, which employs these line segments on point clouds registration.**

***Index Terms*— Facet segmentation, false detection control, feature extraction, line segment extraction, point clouds, registration.**

I. Introduction

In RECENT years, light detection and ranging (LiDAR) technology has been rapidly developed. It can quickly and directly acquire 3-D geospatial information, which is generally identified as point clouds. These point clouds are usually high density and have large volume. For example, an RIEGL VMX-450 MLS system with two full-view RIEGL VQ-450 laser scanners can produce 1.1 million range measurements per second, and yields nearly 100-GB point clouds data in 1 h. These large-scale point clouds have encouraged numerous applications, e.g., urban analysis [1], building construction [2], and intelligent transportation systems [3], [4].

However, the large-scale point clouds, which are acquired by the current variety of LiDAR systems, also present a great challenge in data processing. To reduce computational time, some applications rasterize the high-density 3-D point clouds into 2-D images [4]–[6], and efficient 2-D data processing methods can be applied without many adaptations. Another common method is to process point clouds on feature space. This kind of method, which relies on the feature extraction, is particularly useful for the applications like object recognition and classification [7]. A large amount of features are presented in the literature, where most of them are point-based. A good review of these features is given in [8].

In this paper, rather than the point-based features, we suggest using straight line features to represent the large-scale point clouds. As one of the most common features in man-made environments, straight line features give important information regarding the geometric content of point clouds. Many applications, such as building reconstruction [9]–[11], symmetry detection [12], calibration [13], data registration [14], and localization [15], involve straight line extraction as the fundamental step. Unlike line segment detection in images, which is already a well-investigated topic [16], [17], it has received considerably less attention in 3-D point clouds. Our previous method [5] first converts point clouds into a collection of shaded images, and then applies 2-D image line segment extraction method to extract 2-D line segments. These 2-D line segments in multiple shaded images are finally combined into 3-D line segments. However, this method requires high-quality shaded images with explicit edge information, which is sensitive to the image resolution and noise. In this paper, we present a facet segmentation-based line segment extraction method, which works directly on the original point cloud. Compared with the method [5], the proposed method is more efficient and extracts more correct and complete line segments.

A. Related Work

Most of the techniques for 3-D line segment extraction can be classified into multiview stereo and surface fitting-based methods.

1) Multiview Stereo Methods: The core idea of multiview stereo is to regard a 3-D model as a collection of

Manuscript received November 15, 2015; revised August 30, 2016 and October 29, 2016; accepted December 6, 2016. Date of publication June 9, 2017; date of current version August 25, 2017. This work was supported in part by the National Natural Science Foundation of China under Project U1605254, Project 61371144, and Project 41471379 and in part by the Natural Sciences and Engineering Research Council Discovery Grant under Project 11368. *(Corresponding author: Cheng Wang.)*

Y. Lin is with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China, and also with the Computer Engineering College, Jimei University, Xiamen 361021, China.

C. Wang and D. Zai are with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: cwang@xmu.edu.cn).

B. Chen is with the School of Software, Xiamen University, Xiamen 361005, China.

J. Li is with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2016.2639025

2-D images from different views. Because 2-D line segments can be detected relatively easily in image data, the major challenge here is the line matching over multiple views. Taylor and Kriegman [18] attempted to reconstruct 3-D lines by minimizing the objective function, which measures the total squared distance in the image plane between the observed 2-D lines and the projection of 3-D reconstructed lines. Heuel and Forstner [19] first extracted geometric entities points, lines, and planes in 2-D and 3-D with their uncertainty by projective geometry; they then found the relations between these geometric entities by statistical hypotheses tests. Using these tools, they introduced an algorithm, based only on geometric information in multiple views to solve the line matching problem. In the approach [20], lines were observed by factoring a matrix containing line correspondences. Jain *et al.* [21] reconstructed 3-D line segments from each image independently by evaluating depth value and connectivity constraints. Finally, they merged partial reconstructions from different images into a global result.

All the above-mentioned algorithms required images as input data. By contrast, Lin *et al.* [5] projected the scanned point cloud into shaded images by nonphotorealistic rendering from different views and then extracted 3-D line segments by taking into account both shaded images and 3-D point clouds. However, this method relies on the quality of the shaded images, and therefore is limited.

2) Surface Fitting-Based Methods: Surface fitting is a widely used approach to extract line features from 3-D scattered data. Ohtake *et al.* [22] extracted ridge-valley lines from the given mesh via implicit surface fitting. Here, the ridge-valley lines were defined as the curves on an implicit surface of mesh vertices along which the surface bends sharply. After detecting the ridge vertices by estimating their curvature tensors and curvature derivatives on the fitted surface, the ridge-valley lines were easily traced by connecting the detected ridge vertices. Similarly, Kim [23] used a modified moving-least-squares (MLSs) surface to locally fit the points and extracted the ridge-valley lines by connecting the ridge vertices along the principal curvature direction. Daniels *et al.* [24] projected the points to the intersection of multiple surfaces that were locally fitted via robust MLS and constructed the polylines through the projected points.

Because most man-made scenes are piecewise-planar, a plane fitting-based method is another suitable choice for 3-D line extraction. As a special case of the surface fitting-based methods, plane fitting-based methods approximate the point cloud by multiple planes and extract the lines between the pairs of adjacent planes or discontinuous planes [13], [15]. Compared with surface fitting-based methods, plane fitting-based methods are usually more robust under a high level of noise and outliers. However, these methods tend to fit the points into a large plane and, therefore, lose details that may be important for visual perception.

3) Other Techniques: Demarsin *et al.* [25] extracted candidate feature points via segmentation and recovered the closed sharp feature lines based on graph theory. Sunkel *et al.* [26] proposed a framework that learned line features from user input.

B. Contribution

The main contributions of this paper are summarized as follows.

1) The proposed method works directly on the original point cloud, which has advantages over the methods requiring projected images or reconstructed models.
2) We suggest using facets (small planes) to represent the original point cloud. The advantage is that it avoids the loss of details as compared with the traditional plane fitting-based methods. We also present a two-pass method to efficiently segment the original point cloud into a set of facets.
3) Based on the segmented facets, we present an efficient line segment extraction method and introduce the concept "number of false alarms (NFAs)" into 3-D point cloud context to filter the false positive detections.
4) To verify the accuracy of our extracted line segments, we present a reliable and efficient line-based registration framework, which employs our extracted line segments on solving point clouds registration problem.

II. Overview

Our aim is to develop an efficient and reliable algorithm, which is capable of extracting line segments from large-scale point clouds. Since the point clouds can be acquired from different ways, we also expect that the algorithm is applicable to various types of point clouds.

As illustrated in Fig. 1, the main steps of our method are as follows. Given a set of 3-D points as input [Fig. 1(a)], we first segment these points into multiple facets [Fig. 1(b)]. Then, we extract the boundary points from these facets, and equip each boundary point with a direction vector. [Fig. 1(c)]. Finally, these boundary points are grouped into line segments [Fig. 1(d)].

III. Facets Segmentation

Planar structure is the key to extract lines from point clouds. Rather than undersegmentation, we oversegment the point clouds into a set of small planes, which are denoted as facets. The reason is that oversegmented parts can be merged later when they share the similar properties, but undersegmented parts are difficult to split again [27]. Besides that, small structures can better maintain the boundary information without losing the details.

Our aim is to seek an oversegmentation method that is capable of efficiently partitioning a point cloud into facets while simultaneously preserving the boundary details. A feasible solution is to adopt voxel cloud connectivity segmentation (VCCS) [28], which is a state-of-the-art oversegmentation method and is proven to conform better to the object boundaries than other methods in depth images segmentation problems. However, for the following reasons, the VCCS cannot be used directly in our application. First, the neighborhood in the VCCS is defined by a constant distance R_{voxel}. However, it is difficult to determine the value of R_{voxel} for nonuniform point clouds, which are typically obtained by mobile laser devices; too small a value of R_{voxel} may cause discontinuities

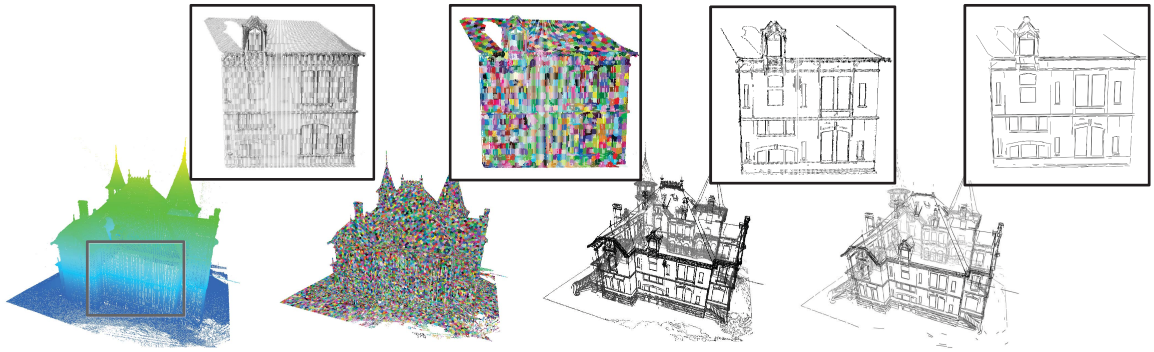


Fig. 1. Pipeline of the proposed method. (a) Input point cloud (from AIM @ shape repository). (b) Facets obtained by segmentation. (c) Boundary points of facets. (d) Final line segments. The subfigures in black boxes are the close-up views.

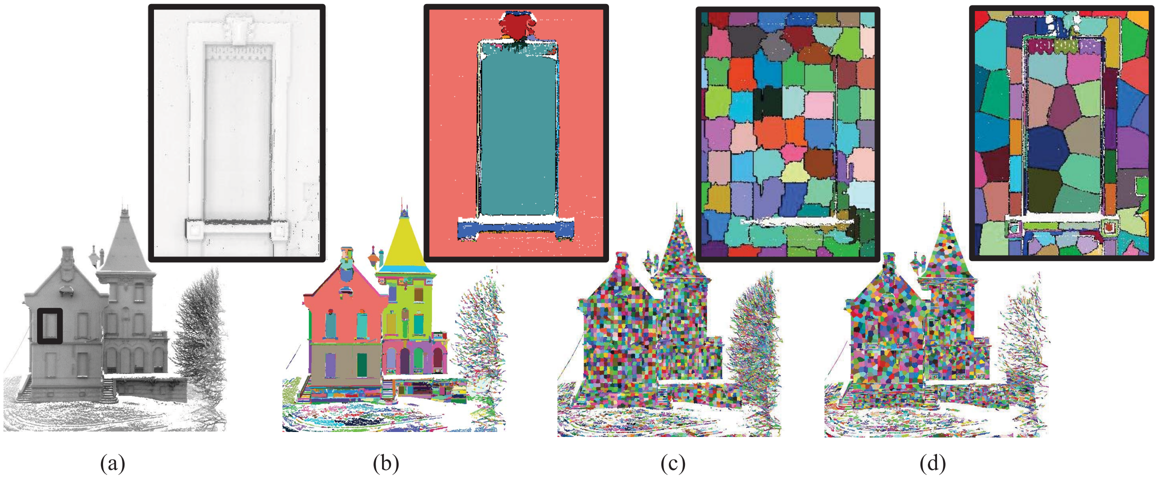


Fig. 2. Comparison results of three segmentation methods. (a) Input point cloud (from AIM @ shape repository). (b) Plane segmentation results by region growing-based method [30]. (c) Segmentation results by VCCS [28]. (d) Our result.

in sparse areas; in contrast, too large a value of R_{voxel} means the loss of details. Second, the initial seed points are evenly chosen by partitioning 3-D space with a resolution R_{seed}, which implies that structure with a radius smaller than R_{seed} may not be confirmed by an individual supervoxel cluster, and the boundaries of these structures may not be well-preserved [see Fig. 2(c)].

To address the above-mentioned problems, we present a two-pass method. In the first pass, we segment the point cloud into facets by region growing, which has the advantage of high speed but coarse boundaries of the facets. In the second pass, we refine the boundaries of these facets by local k-means clustering. Considering the nonuniform point cloud, we adopt the k-nearest neighborhood for region growing. Moreover, unlike VCCS, which works on a voxelized point cloud, the proposed method works directly on the original point cloud, resulting in the preservation of more geometric features. Greater detail is given in Sections III-A–III-C.

A. Coarse Facet Segmentation

Rather than the evenly partitioned strategy in the VCCS, the proposed method attempts to represent every planar region by at least one facet. A facet is represented by a triple $f_i = (P_i, c_i, \widehat{n_i})$ with its constituents point set P_i, centroid c_i, and unit normal vector $\widehat{n_i}$. Two thresholds are required here: σ, a distance threshold to determine if a point lies on a facet; and θ, an angle threshold to determine whether two facets are coplanar. Further discussion for setting the values of these parameters is presented in Section V-A.

For each input point, x_i, from the original point cloud $\mathcal{X}$, we first compute the tangent plane T_i for x_i by iteratively refitting. Let K_i^0 be the K-nearest neighbors of x_i (including x_i itself), where K is a tunable parameter, whose value is discussed in Section V-A. Then, the initial tangent plane, T_i^0, is estimated by performing principal component analysis (PCA) on K_i^0. T_i^0 is constrained to pass through the center point of K_i^0. In the mth iteration, we remove the points from K_i^m, whose distance to T_i^m exceeds $\sigma/2$ and denote the remaining points as K_i^{m+1}. Then, each tangent plane T_i^{m+1} is refitted to the K_i^{m+1} by using PCA again in the further iteration. The iteration ceases when K_i^m remains constant.

After all tangent planes T_i have been computed for all x_i values, we repeatedly select a seed point from the original point cloud. Because the points with smoother local neighborhoods are more likely to belong to planes, they are tested first. Let λ_1, λ_2, and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) be the three eigenvalues of the covariance matrix formed by K_i^m. The smoothness of x_i in its local neighborhood is estimated by

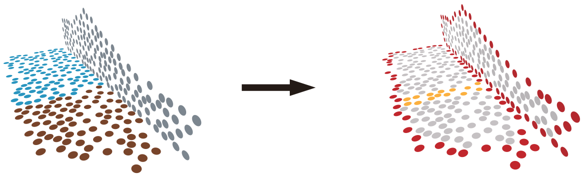


Fig. 3. Given a point cloud with three facets, the boundary points (drawn in red) are defined as the α-shape vertices of each facet with the exception of the vertices at the intersection between two adjacent coplanar facets (drawn in yellow).

λ_2/λ_3. The larger the smoothness, the higher the priority of the point.

When a point, x_i, is selected as the seed, we create a new facet $f_i = (\{K_i^m\}, x_i, \widehat{T_i})$, where $\widehat{T_i}$ is the unit normal vector of T_i. Then, a region growing procedure commencing with f_i is performed. In each iteration, the points adjacent to the facet f_i are tested. We add each adjacent point x_j to facet f_i, if x_j satisfies the following three criteria: 1) the distance from x_j to x_i does not exceed R_{seed}; 2) the angle between $\widehat{T_i}$ and $\widehat{T_j}$ does not exceed the tolerance θ; and 3) the orthogonal distance from x_j to f_i is smaller than $\sigma/2$. Here, R_{seed} is used to constrain the radius of facets, so that the big planes can be segmented into smaller pieces. Once a point is added to the region, it will be marked and never tested again. The region growing is terminated when no more points can be added.

Finally, we obtain an initial set of facets.

B. Boundary Refinement

To obtain a better boundary for each facet, we adopt the local k-means clustering related to [28]. Local k-means clustering iteratively assigns the points to a facet. Specially, it ensures that the distance from each point in $\mathcal{X}$ to its facet f_i is smaller than the distance to the adjacent facets of f_i. Here, the distance is formed as the weighted sum of three components: spatial extent distance D_s, color distance D_c, and normals distance D_n, that is

$$D = W_s * D_s + W_c * D_c + W_n * D_n. \tag{1}$$

Following the suggestion in [29], we set $W_s = 1$, $W_n = 4$, and $W_c = 0$ (only the geometric features are considered here). In this paper, we expect each facet to be more planar than VCCS's. For this purpose, consider a point, p, with normal vector, $\widehat{v}$, and a facet, $f_i = (P_i, c_i, \widehat{n_i})$, if the angle between $\widehat{v}$ and $\widehat{n_i}$ exceeds θ, or the distance from p to c_i along the direction $\widehat{n_i}$ exceeds σ, we set the distance D between them as infinity.

Once all points have been assigned, for each facet f_i with its constituents P_i, we set its centroid c_i be the mean of P_i and set its normal vector $\widehat{n_i}$ be the principal direction of P_i in the form of linear least squares. The clustering process terminates when all the facets are stable.

C. Segmentation Results

The advantage of our segmentation method is that it conforms better to the facet boundaries while not reducing the speed.

Fig. 2 shows the comparison results of a region growing-based plane detection (RGPD) [30], VCCS [28], and the proposed method. RGPD tends to fit the points into a large plane, hence lose the details [see the window in Fig. 2(b)]. The VCCS produces the partitions with nearly the same size, but misses the boundaries of small objects [see Fig. 2(c)]. In contrast, the proposed method produces the adaptive area of facets, which well represent the local planar structures and thus preserve the boundaries [see Fig. 2(d)].

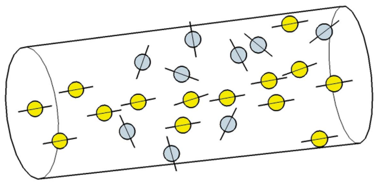


Fig. 4. Cylinder structure for line segment grouping.

IV. Line Segment Extraction

After segmenting the point clouds into facets, we can only consider the line segments located on the boundaries of these facets. With the aid of classical 2-D α-shape [31], the boundary points can be efficiently detected. The remaining problem is to group the boundary points into multiple line segments.

However, in the real world, natural scenes always have plenty of surface textures and irregular shapes, which give rise to the clutter in the boundaries. It is not a simple task to group points from these boundaries into line segments. Therefore, a validation to filter the false positive detections in the clutter scenes is required. On the other hand, parameter-tuning is a generic issue in many detection problems. Since we seek the method that can cope with the various point clouds, a robust line segment grouping method with self-tuning parameters is desired.

In Sections IV-A and IV-B, we first describe how boundary points are extracted from the facets, and then present a line segment grouping method with a false detection control.

A. Boundary Points Extraction

We define the boundary points as those α-shape vertices of each facet with the exception of the vertices at the intersection between two adjacent coplanar facets (see Fig. 3). Assume a finite set of points P in the plane, the α-shape of P is a generalization of the convex hull. Any vertices of the α-shape are the member of P, and an edge of the α-shape is drawn between two points of P whenever there exists a circle of radius α passing though these two points but not containing any other points of P. The α-shape can be derived from Delaunay triangulation and formed as a polygon with holes.

For each facet $f_i = (P_i, c_i, \widehat{n_i})$, the vertices of f_i's α-shape can be regarded as the boundary points of f_i. However, this boundary contains the points that lie on the intersection between adjacent coplanar facets, which are required to exclude. For this purpose, we first compute a facet set F_i of f_i, which is defined as the set of adjacent facets coplanar with and including the current facet f_i. Then, we extract the vertices of the α-shape of F_i, and compute their intersection with P_i as

Algorithm 1 LineGrouping

Input: A boundary points set B, an angle tolerance θ.
Output: A list L of line segments.

```
$Used(B) \leftarrow false.$
for each boundary point $p_b$ in $B$ do
  Initialize a cylinder $c(r, l)$, with radius $r = +\infty$, and
  center line $l$ determined by $p_b$ and $p_b.\vec{v}$
  repeat
    $R \leftarrow p$
    for each point $p$ in $R$ do
      $Used(p) \leftarrow true$
      for each neighbor point $q$ of $p$ do
        if $q$ is aligned to $c$ and $Used(q) = false$ then
          Add $q$ to $R$
          $Used(q) \leftarrow true$
        end if
      end for
    end for
    $c' \leftarrow RegionToCylinder(R)$
    if $c'.r < c.r$ then
      $c \leftarrow c'$
      $Used(R) \leftarrow false$
    end if
  until the parameters of $c$ remain constant
  if $NFA(c) \leq 1$ then
    Add $c.l$ to $L$
  else
    $Used(R) \leftarrow false$
  end if
end for
return L
```

the desired boundary points. Here, the value of α is set to the average spacing of F_i, that is

$$\alpha = \frac{1}{|F_i|} \sum_{p \in F_i} \min_{q \in F_i \wedge q \neq p} \|p - q\|. \tag{2}$$

In addition, we also equip each boundary point p of F_i with a direction vector $\vec{v}$, which is defined as the principal direction of p and its nearby boundary points of F_i in the form of linear least squares. More concretely, we form the α-shape of F_i as a link of vertices, and the nearby points of p contain the next two points and previous two points of p along the link. These direction vectors are useful for grouping the boundary points into line segments.

B. Line Segment Grouping

As mentioned before, line segment grouping is an essential part of the proposed line segment extraction method. The main idea of our line segment grouping method is to dynamically maintain a cylinder structure for each candidate line segment. The cylinder, determined by a center line segment l and a radius r, is a geometric region in 3-D space (see Fig. 4).

Rather than directly grouping the boundary points into line segments, grouping to cylinders provides a way to filter the false positive detections, which is especially useful for the point clouds in the clutter scene. Concretely, since each boundary point is equipped with a direction vector, a concept of "aligned," between the boundary point and the cylinder, is defined as follows.

Algorithm 2 RegionToCylinder

Input: A region, R, which is consisted of aligned boundary points; number of repeated times, m.
Output: A fitting cylinder c of R.

```
$r_{min} \leftarrow +\infty$
for $i \leftarrow 1$ to $m$ do
  Randomly choose two boundary points $p_1$, $p_2$ from $R$
  Compute a line $l$ passing through the chosen points $p_1$
  and $p_2$
  $r \leftarrow \text{median}_{p \in R}\, \text{dist}(p, l)$
  if $r < r_{min}$ then
    $r_{min} \leftarrow r$
    $c \leftarrow cylinder(l, r)$
  end if
end for
return $c$
```

1) Definition (Aligned): We say that a boundary point p with direction vector $\vec{v}$ is aligned to a cylinder c, if the angle between $\vec{v}$ and the center line segment of c is smaller than the angle threshold θ.

Note that, we adopt the same angle threshold θ as the one in Section III.

The concept "aligned" gives us an intuitive way to determine whether or not a cylinder can be approximated by a line segment. One naive method is to verify if the fraction of aligned boundary points over all boundary points in the cylinder is larger than a threshold. However, such hard-threshold method is clearly not appropriate for the various types of input point clouds. To address this problem, we introduce the concept "NFAs," which is originally presented in [32] for solving alignment issue in 2-D images, and be further applied in [16] for 2-D line segment detection.

Let B be the extracted boundary points set. Suppose that we have another boundary point set B', having the same size of points as B. For each boundary point in B', its direction vector is randomly assigned. B' is called a random (or nonstructured) model. According to the Helmholtz principle [33], only a very few line segments could have detected from B'. The validation is based on a contrario theory, i.e., we accept only the structures that unlikely appear in B'. Specially, for each cylinder c obtained in our line segment grouping procedure, we can evaluate the expected number of c that occurs in B' by chance. This expected number, which is named as "NFA," is defined as follows.

2) Definition (NFA): Given a cylinder c with n total boundary points and k aligned boundary points, the NFA of c is defined as

$$\text{NFA}(c) = N^2 \cdot \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{3}$$

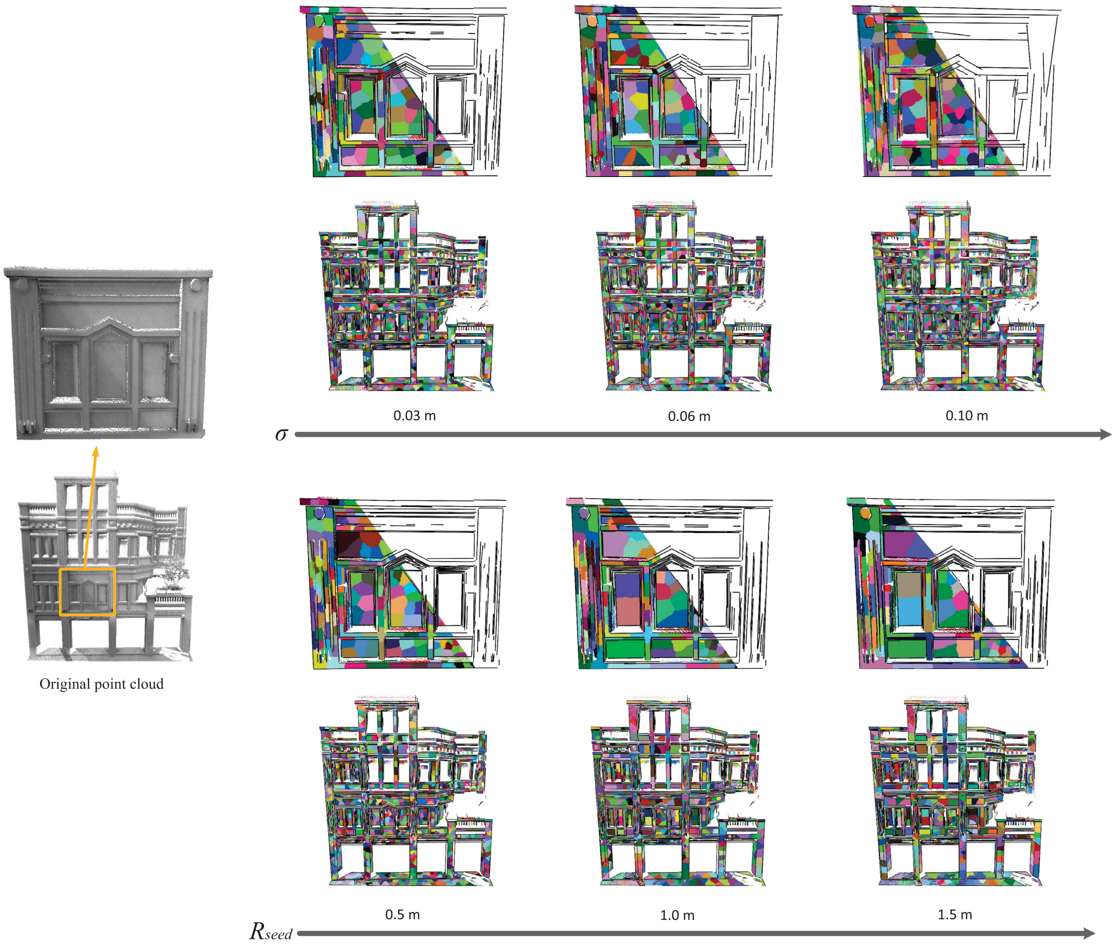


Fig. 5. (Top) Effect that σ has on the obtained facets and the final extracted line segments. (Bottom) Effect that R_{seed} has on the obtained facets and the final extracted lines.

TABLE I
COMPUTATIONAL RESULTS OF VARIOUS POINT CLOUDS

Data information			Computational results			
Description	Fig.	Number of points	Number of lines	Segmentation (s)	Line extraction (s)	Total running time (s)
Castle [1]	1	32,754,893	21412	296.8	246.2	543.0
Shophouse [2]	5	2,184,871	1756	22.0	14.6	36.6
Trim star [1]	6 (a)	1,000,000	608	7.1	7.1	14.2
Swiss hotel [2]	6 (b)	810,190	2358	6.6	3.2	9.8
Fandisk [1]	6 (c)	1,000,000	325	6.5	8.8	15.3
Huazhilin hotel [2]	6 (d)	782,197	1179	6.1	2.7	8.8
StSulpice [3]	6 (e)	6,632,479	8816	80.9	37.7	118.6
Huandao road [2]	6 (f)	43,517,926	6561	393.9	450.3	844.2
Haiyun building [2]	7(f)	4,802,365	2272	37.9	37.0	74.9
Bildstein station1 [4]	7(k)	2,245,638	1104	19.5	17.6	37.1
Keyan building [2]	10	2,185,329	2482	22.3	13.0	35.3

[1] The data is download from AIM@SHAPE: http://visionair.ge.imati.cnr.it/ontologies/shapes/
[2] The data is provided by Fujian Key Laboratory of Sensing and Computing for Smart Cities (Xiamen University).
[3] The data is download from libE57: http://www.libe57.org/data.html
[4] The data is download from http://semantic3d.net/

where N^2 represents the number of potential cylinders in N boundary points, due to the fact that every two points can determine a cylinder. And, $p = \theta/\pi$ is the probability that two random directions are aligned.

A cylinder c is called ϵ-meaningful, if $\text{NFA}(c) \leq \epsilon$. Following the suggestion and experiments in [32] and [16], we can fix ϵ to 1, i.e., allowing one false positive detection per random model. Now, given a cylinder c, we can simply accept it if $\text{NFA}(c) \leq 1$, and reject it otherwise.

Upon the cylinder structures and NFA false positive control, we give the complete line segment grouping algorithm. As shown in Algorithm 1, we first find the cylinder with a minimal radius by iteratively refitting. More concretely, for each boundary point p_b, a cylinder c, whose center line is determined by p_b and p_b's direction vector, is initialized. And the initial radius of c is set to infinity. Then, a region growing commencing with p_b is performed to obtain a region R that is consisted of the boundary points aligned to c

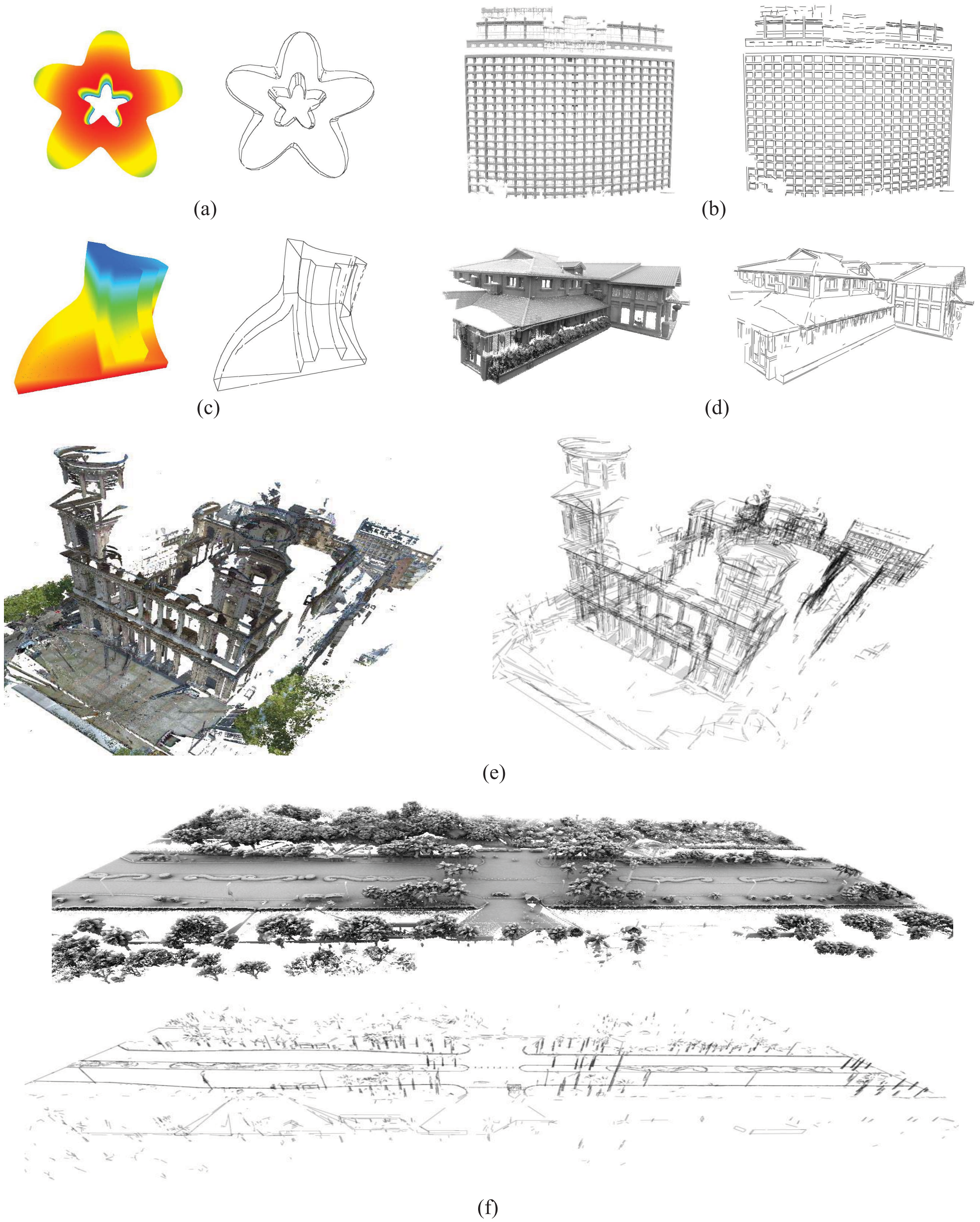

(a) (b) (c) (d) (e) (f)

Fig. 6. Extracted line segments of various types of point clouds. (a) Trim star. (b) Swiss hotel. (c) Fandisk. (d) Huazhilin hotel. (e) StSulpice. (f) Huandao road.

(see Algorithm 1, lines 4–14). After the region growing is terminated, cylinder c is updated to fit the region R. Once a best fitting cylinder c is obtained, we verify it by $NFA(c)$ value. If a cylinder has been rejected, its region points R are marked as unused, means that they can be reused for the regions commencing from the other points. Otherwise, we add a line segment, $c.l$, to the output list.

The subroutine **RegionToCylinder** computes a cylinder to fit the points of a region. Here, we adopt the least median of squares (LMS) method [34], which is a robust regression method and can tolerate up to 50% outliers without requiring threshold. The LMS method estimates the cylinder c of region points R by solving the following nonlinear minimization problem:

$$\arg\min_{c.l} \operatorname*{median}_{p\in R}(\text{dist}(p, c.l)) \tag{4}$$

where $\text{dist}(p, c.l)$ is the nearest Euclidean distance from p to c's center line segment, l. Then, the radius of c is set to the farthest distance from $p \in R$ to $c.l$. The common way to solve (4 is based on a random sampling framework: repeatedly estimate a line segment determined by two randomly selected points, and store the best line segment in

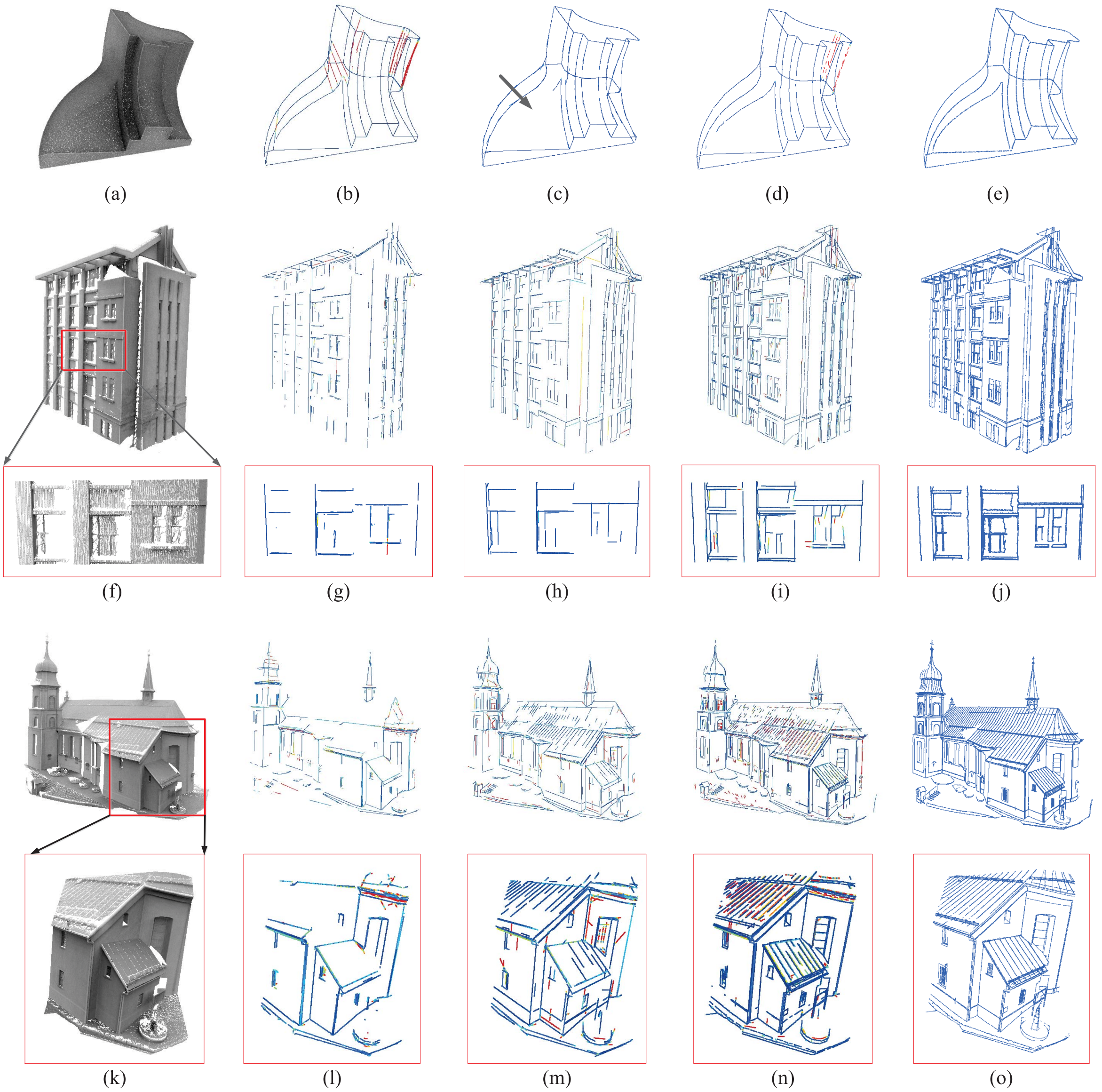


Fig. 7. Extracted line segments by different methods. The color indicates the distance of line from its nearest ground-truth point: the red marks the highest distance value, and the blue marks the lowest. (a), (f), and (k) Original point clouds. (b), (g), and (l) Plane extraction based method [15]. (c), (h), and (m) Multi-view stereo based method [5]. (d), (i), and (n) Proposed method. (e), (j), and (o) Ground-truth.

the form of LMS. Let m be the repeated times, it can be determined by $P = 1 - (1 - \phi^2)^m$ according to [34], where ϕ is the fraction of outliers and P is the probability that at least one estimated line segment is inlier. Here, let $P = 99.99\%$ and $\phi = 50\%$, we can deduce $m \approx 32$.

V. Experimental Results

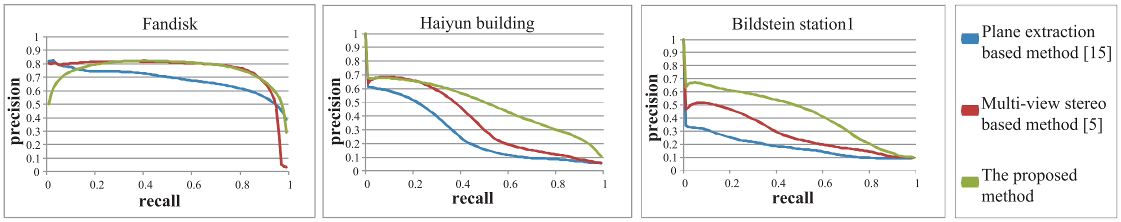
A. Environment and Parameters

We implemented the proposed method using C++ and OpenGL. The processing times were obtained in a PC with Intel Core i5-3470 3.2-GHz CPU and 12-GB RAM. The operation system is Ubuntu 14.04.

The following four parameters are involved in the proposed method: σ, θ, R_{seed}, and K. Below, we describe how to set these parameters.

As given in Section III, σ is the scale distance used to determine whether a point lies on a facet. Generally, σ cannot be determined by a fixed value; rather, it depends on the scale that is of interest to the user, the scale of the data, and the noise level. An example of the effect that σ has on the segmentation results and the final extracted line segments is given in Fig. 5. The higher the value of σ, the fewer the detected facets and line segments. And when the value of σ is set too large (e.g., 0.1 m in Fig. 5), the obtained line segments are less accurate. Here, we estimated σ through statistical analysis, i.e., set σ to twice the average spacing of the input point cloud. This value works well on a broad spectrum of data sets.

θ is another important global parameter used to check whether two direction vectors are aligned. According to the suggestion from [32] and [16], $\theta = 22.5°$ is an



(a) (b) (c)

Fig. 8. Precision-recall curves of different methods. (a) Fandisk. (b) Haiyun building. (c) Bildstein station1.

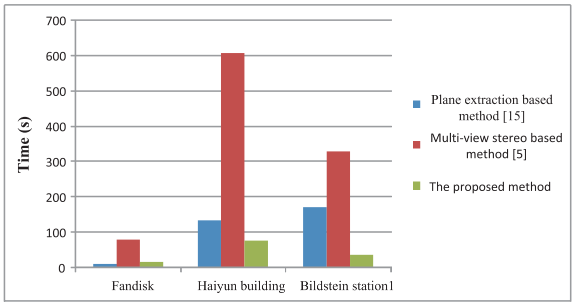


Fig. 9. Time performance of three methods.

appropriate value depending on both Psychophysics and numerical experimentation.

R_{seed} gives the upper bound of the resolution of facets. As shown in Fig. 5, the obtained line segments are insensitive to R_{seed}. Here, we empirically set $R_{\text{seed}} = 15\sigma$.

K is the number of nearest neighbors used to define the neighborhood of the point cloud. In our experiments, $K = 15$ is adequate. However, this number may be increased if the point density is highly nonuniform.

We used these parameter values, without tuning, throughout all of the following experiments. It is also worth to mention that fine-tuning these parameters for different types of point clouds would obtain better results.

B. Computational Results

We tested the proposed method on various types and sizes of point clouds, including three point clouds sampled from three mesh models ("Fandisk," "Trim star," and "Sharp sphere") and eight laser scanning point clouds ("Castle," "Shophouse," "Swiss hotel," "Huazhilin hotel," "StSulpice," "Haiyun building," "Bildstein station1," and "Huandao road"). The information of test data and the computational results are summarized in Table I.

Fig. 6(a) and (c) shows two point clouds uniformly sampled from mesh models. Such mesh models, typical of CAD models, contain obviously sharp edges. These mesh models allow us to intuitively check the obtained line segments. For these two cases, almost all the expected line segments are extracted. For the curve edges, our concern is not to generate a set of perfectly connected line segments, but rather approximate the curve edges by generating a set of locally straight line segments with small spacing between the line segments.

For real world laser scanning data, it is difficult to define ground truth, especially at the clutter area such as plants. The Helmholtz principle states that no structure should be perceived in a uniform random image. Similarly, we do not want to detect any lines in the area that can be seen as uniform noise. As shown in Fig. 6(d), our algorithm preserves most of the important geometric contents, while filtering out the bush. Another example is given in Fig. 6(f), where our algorithm successfully detects the trunk of each tree and obtains much fewer structures at their leaves.

In general, the proposed method successfully extracts the major line segments of the test data within the acceptable time. These line segments, including sharp edges and boundaries, represent well the structures of the input point clouds.

C. Comparative Studies

In order to quantitatively evaluate the quality of line segments extracted by different methods, we first manually selected the points that lie on the plane intersection lines, plane boundaries, and linear structures (e.g., light pole) from the original point clouds. These points are regarded as the ground-truth points. Then, we measured how well the extracted line segments can approximate the ground-truth points by computing the precision-recall curves [35]. Here, precision is defined as the probability that a point lying on the extracted line segment belongs to ground-truth points; recall is defined as the probability that a ground-truth point lies on the extracted line segment. High precision relates to low false positive of resulting line segments, and high recall relates to low false negative of resulting line segments. We used a distance threshold t to determine whether a point lies on the line segment. With changing the value of t, the precision-recall curves are produced. A curve with high recall but low precision will return many line segments, but most of line segments are incorrect. In contrast, a curve with high precision but low recall will return very few line segments, but most of line segments are correct. The ideal curve should have both high precision and high recall, i.e., the area under the curve should be as large as possible.

As shown in Fig. 7, we compared the proposed method with a plane extraction-based method [15], a multiview stereo-based method [5], and the ground-truth data. The color of each line indicates the distance of the line from its nearest ground-truth point: the red marks the highest distance value, and the blue marks the lowest.

The first test data is the small man-made object ("Fandisk"), which only contains the simple and regular structures. Because

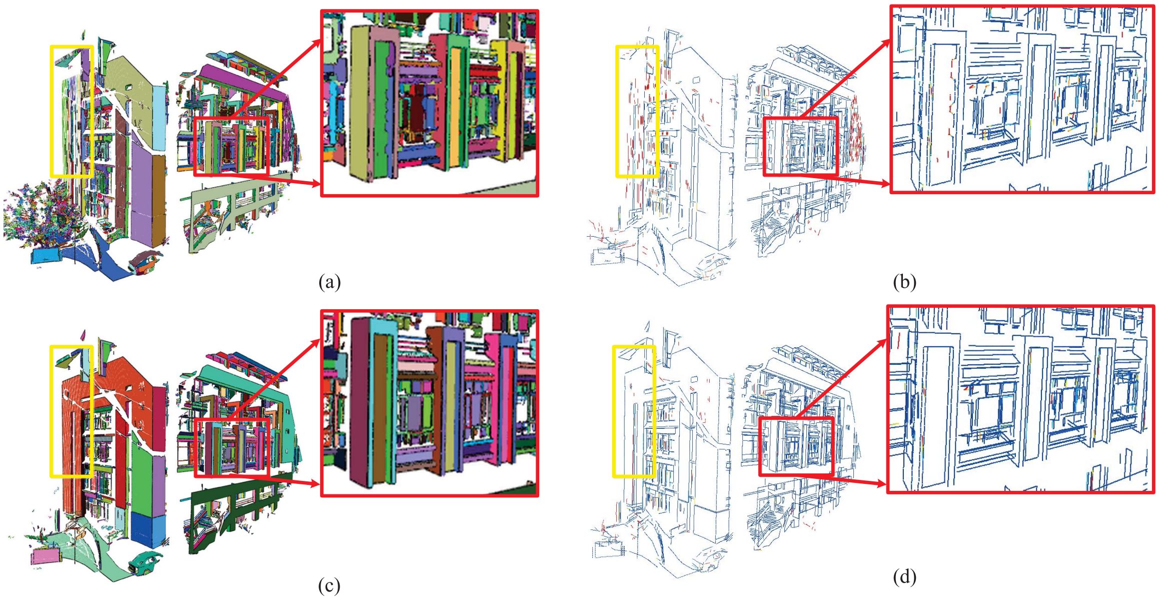


Fig. 10. Comparison between VCCS [28] and our facet segmentation method. (a) VCCS segmentation result. (b) The obtained line segments by VCCS method. (c) Our facet segmentation result. (d) The obtained line segments by our method.

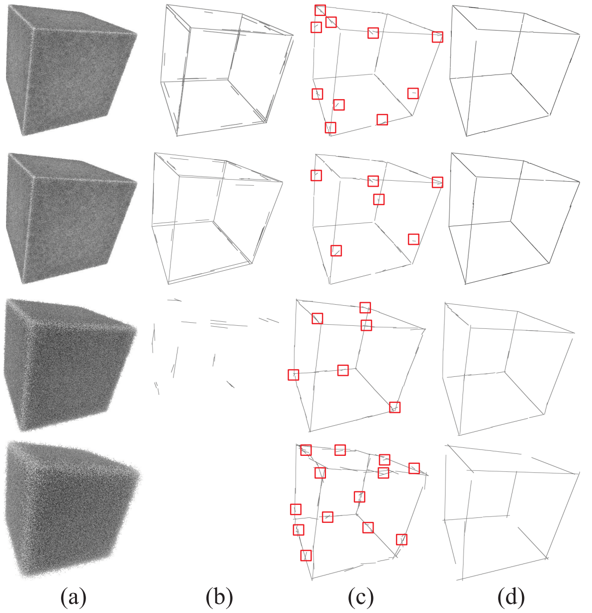
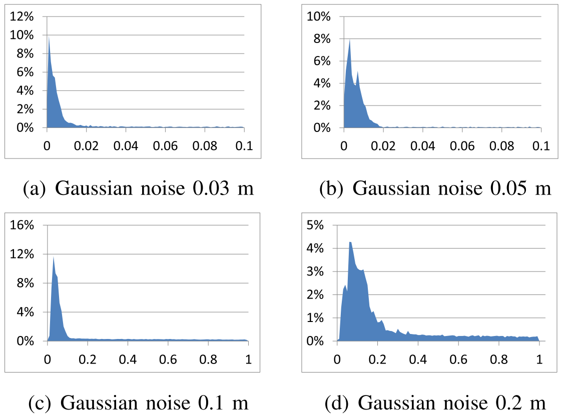


Fig. 11. Extracted line segments in the presence of different levels of Gaussian noise. From top to bottom: 0.03 m Gaussian noise, 0.05 m Gaussian noise, 0.1 m Gaussian noise and 0.2 m Gaussian noise. (a) Input point cloud with Gaussian noise. (b) Extracted line segments in [5]. (c) Extracted line segments by the proposed method without NFA control (red boxes emphasize the false detections). (d) Extracted line segments with NFA control.



(a) Gaussian noise 0.03 m (b) Gaussian noise 0.05 m

(c) Gaussian noise 0.1 m (d) Gaussian noise 0.2 m

Fig. 12. Histograms show the Hausdorff distances between the ground-truth data and our line segment results in the different Gaussian noise levels. The horizontal axis is the distance value, and the vertical axis is the corresponding percentage to each distance value. (a) Gaussian noise 0.03 m. (b) Gaussian noise 0.05 m. (c) Gaussian noise 0.1 m. (d) Gaussian noise 0.2 m.

it cannot fit the smooth curved surface into a single plane, plane extraction-based method [15] produces several false positive plane intersection lines [see the red lines in Fig. 7(b)]. Multiview stereo-based method [5] dose not produce obvious false positive lines. However, it misses the lines, which are not obvious for visual perception [see the area in Fig. 7(c) where the black arrow points to]. The proposed method obtains more complete lines than method [5], although also generates several false positives. Once the high curvature smooth surface is segmented into facets, which are mistakenly regarded as noncoplanar facets, the false positive lines are then produced. In summary, for this test case, all of the three methods are not perfect. The same conclusion can also be observed from Fig. 8(a).

The second test data are a modern architecture ("Haiyun building"), which contains a great deal of plane structures. However, because this point cloud is acquired by a terrestrial laser scanner, the incident angle of lasers is restricted, resulting in incomplete plane boundaries and less plane intersection lines. As shown in Fig. 7(g) and (h), the plane extraction-based method and the multiview stereo-based method only extracts the plane intersection lines. Nevertheless, our proposed method obtains more accurate and complete lines than the other methods. The precision-recall curve in Fig. 8(b) also reveals that our result outperforms others.

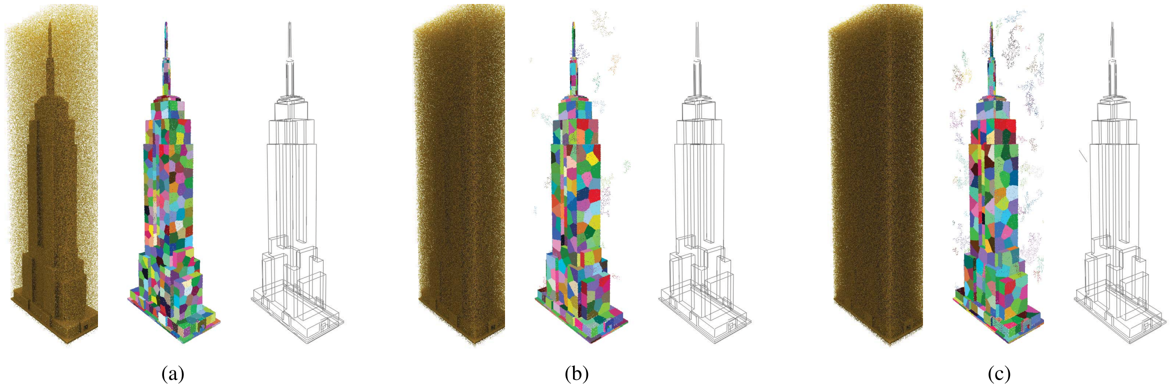


Fig. 13. Segmented facets and extracted line segment result in the presence of outliers. The percentage of outliers is given in the corresponding figures. (a) 20%. (b) 50%. (c) 60%.

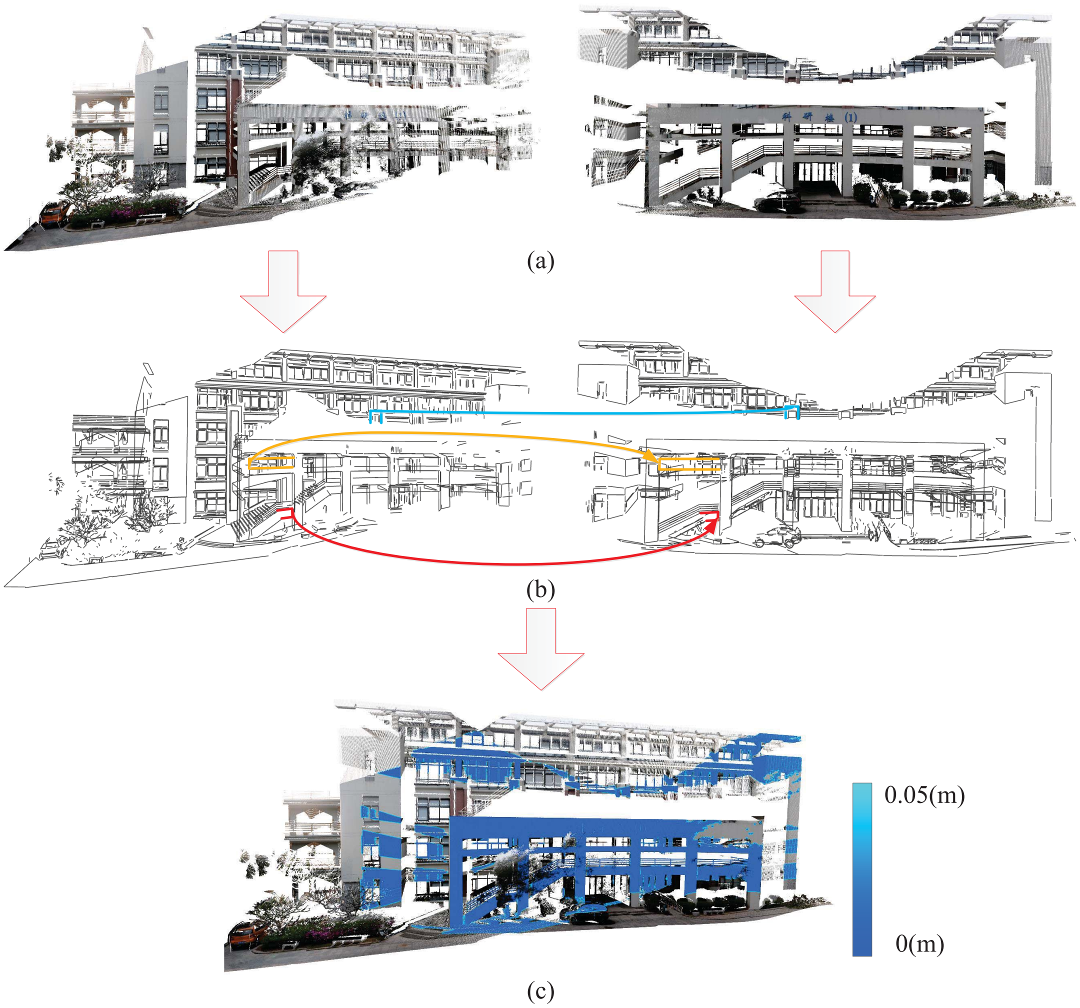


Fig. 14. Line-based registration pipeline. (a) Original point clouds. (b) Line extraction and corresponding matching. (c) Aligned point clouds after ICP refinement. The gray-scale color indicates the distance between common points.

The last test data are a typical church ("Bildstein station1"). Similar to the previous one, the plane extraction-based method [15] extracts only the main lines; the method [5] obtains a more complete result, but fails to detect some plane boundaries, because it is concerned only with the plane intersection lines. Our proposed method obtains a better result, which is reflected in both precision and recall measures [Fig. 8(c)].

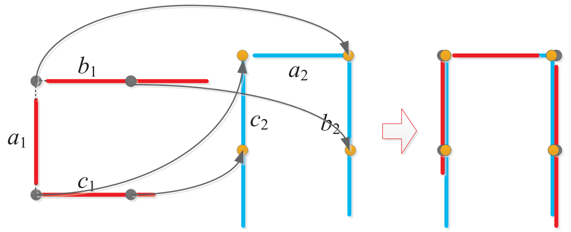


Fig. 15. Two three-line structures alignment.

We also evaluated the time performance of three methods. As shown in Fig. 9, our proposed method is faster than other two methods except for “Fandisk,” where the proposed method is a little slower than plane extraction-based method.

Because the proposed line segment extraction method is based on facet segmentation, we also demonstrate the superiority of our facet segmentation method over VCCS [28]. Fig. 10(a) and (c) shows the segmentation results obtained by VCCS and the proposed facet segmentation method, respectively. To show these boundaries preserved in segmentation results, the adjacent coplanar facets (or supervoxels) are assigned the same color. Compared with VCCS, the proposed method obtains better fitting facets and preserves more accurate boundaries (see red box area in Fig. 10), even when point density varies (see yellow box area in Fig. 10).

D. Robustness Studies

To show the robustness of the proposed method, lines were extracted from the point clouds with noise or outliers added.

First, we studied the robustness of the proposed method by adding Gaussian noise to a synthetic point cloud. As shown in Fig. 11, the original point cloud is uniformly sampled from the surface of a 10 m × 10 m × 10 m cube. One can observe that as Gaussian noise increases, the number of detected lines by the method [5] decreases, and when Gaussian noise reaches 0.2 m, the method [5] is totally failed. The reason is that the method [5] extracts line segments based on projected 2-D images, when noise increases, the edges of projected images are blurred and cannot be detected by 2-D image line segment detection algorithms. In contrast, the proposed method successfully extracts the line segments, and benefits from the NFA control [see the comparison between Fig. 11(b) and (c)], no false detections are made [Fig. 11(d)].

Meanwhile, to demonstrate the accuracy of the proposed method, we computed the Hausdorff distance from the ground truth to the extracted line segments. More specifically, we evenly sampled 10 000 points from the boundary edges of the cube, and computed the minimum distance from each sampled point to extracted line segments. Fig. 12 shows Hausdorff distance results in different Gaussian noise levels, where the horizontal axis is the distance value between the ground-truth data and our line segments results, and the vertical axis is the corresponding percentage to each distance value.

Second, we demonstrated the stability of the proposed method under high amount of outliers. Fig. 13 shows the segmented facets and extracted line segment results in the presence of different levels of outliers. We can observe that small amount (20%) of outliers cannot influence our facets segmentation results. However, when the percentage of outliers reaches 50%, several false facets are produced. Despite this, nearly perfect line segments are produced, even when the percentage of outliers reaches 60%, where the number of original points is 1 million, and the number of outliers is 1.5 million. The reason is our NFA-based validation method successfully filters the false detections.

To summarize, our algorithm produces the accurate and complete line segments in the presence of nonuniform density, noise, or high amount of outliers. The average time of processing 1 million points is under 10 s, which is efficient in practice.

VI. Application

To further verify the accuracy of the line segments extracted by the proposed method, we also present a line-based registration framework, which employs these line segments on point clouds registration.

Point cloud registration is the process of estimating a rigid transformation that minimizes the distance between two point clouds. According to [36], the approaches for point cloud registration can be divided into three categories: surface-based, feature-based, and point-based methods; each has both advantages and disadvantages, where the latter two categories are more popular and appropriate for alignment of cluttered scene [8]. As a common feature, line-based feature had already been applied in point cloud registration. Compared with the point-based feature, line-based feature reduces the data size and involves the boundary information, which can be used to refine the registration. However, most of these algorithms only consider plane intersection lines [37] or require additional photogrammetric data [38]. The line segments obtained by the proposed method are more complete, and can better represent the original point cloud. Benefited from these line segments, our registration approach is d and efficient.

As shown in Fig. 14, our line-based registration contains three stages: 1) line segment extraction; 2) corresponding matching; and 3) iterative closest point (ICP) refinement. More details are given in Sections VI-A and VI-B.

A. Methodology

Given two point clouds P and Q in arbitrary initial positions, our problem is to find a best transformation that aligns P to Q. For rigid transformation, two non-parallel corresponding line-pairs are sufficient to uniquely determine the transformation parameters. Suppose there are n and m line segments extracted from P and Q, respectively. A brute force alignment scheme is to find the best transformation within totally $O(n^2m^2)$ corresponding line-pairs, which is time-consuming. To reduce the complexity, we consider a three-line structure, which we call it *base*.

As shown in Fig. 15, a base consists of three lines, which are labeled by a, b, and c, respectively. These three lines satisfy:

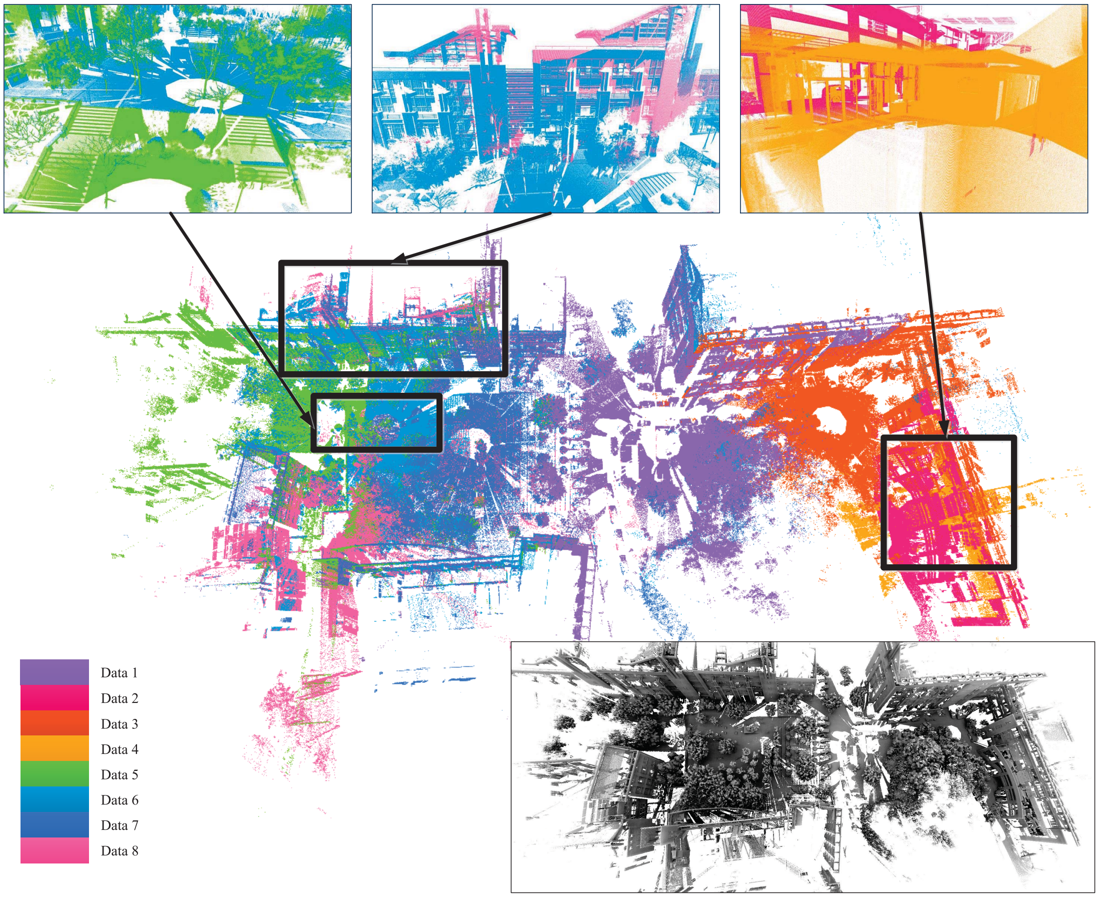


Fig. 16. Registration results of eight scanned point clouds. The gray-scale subfigure shows the overall aligned point cloud rendered with ambient occlusion light.

1) $a \perp b$, $a \perp c$, and $b \parallel c$; 2) a is adjacent to b and c, i.e., the distance from one of a's endpoint to one of b(or c)'s endpoint is smaller than 2σ. Such structure exists widely in man-made environment, especially in architectures. A pair of corresponding bases can uniquely determine a transformation, and the amount of potential corresponding pairs is reduced to $O(k^2)$, where k is the number of bases.

As shown in Fig. 15, given a base $B_1 \equiv \{a_1, b_1, c_1\}$ from P and a base $B_2 \equiv \{a_2, b_2, c_2\}$ from Q, if the length of a_1 is (roughly) equal to the length of a_2, we can always find four (approximately) congruent point-pairs. Then, the transformation can be estimated by [39].

After a transformation is computed, the quality of alignment can be measured by the number of common points (NCP). We define NCP(A, B) be the number of points in the point set A whose closest distance to point set B is smaller than a threshold δ. Rather than using the original point clouds, only the points sampled from the extracted line segments are tested. Furthermore, following the idea of [40], we first test a small constant number of sampled points, and only when most of these points are well matched, we test the remaining points. This scheme significantly reduces the computation time. Finally, we align the points P to Q according to the transformation with largest NCP, followed by employing iteratively closest point (ICP) method on the sampled points [41] to refine the solution.

B. Results

The method is tested on the point clouds acquired by the terrestrial laser scanner RIEGEL VZ-1000, which provides high-speed, noncontact data acquisition using a narrow infrared laser beam and a fast scanning mechanism. The accuracy is 8 mm, at the same time, the precision reaches 5 mm. The scan range varies from 2.5 to 1000 m, and the scan angle range covers 100° in the vertical direction and total 360° in the horizontal direction. The size of the scanned data, including seven outdoor scenes and one indoor scene, is about 3 GB. The surveyed area, the campus of Xiamen University, is a typical man-made environment, which is covered with high buildings, dense vegetation, and cars.

We apply the proposal line-based registration method on the scanned point clouds. After alignment, these aligned point clouds are integrated into one complete point cloud. As shown in Fig. 16, the test point clouds are cluttered, uncompleted, and full with occlusion. Three typical scenes are emphasized by black boxes, including trees, ground, building from different views, and indoor scene. Despite all this, one can find that the boundaries of these areas are well aligned, which means that the registration is reliable.

The computational results are summarized in Table II, where the fourth column denotes the percentage of NCP over the input points. Here, NCP is calculated between each point cloud

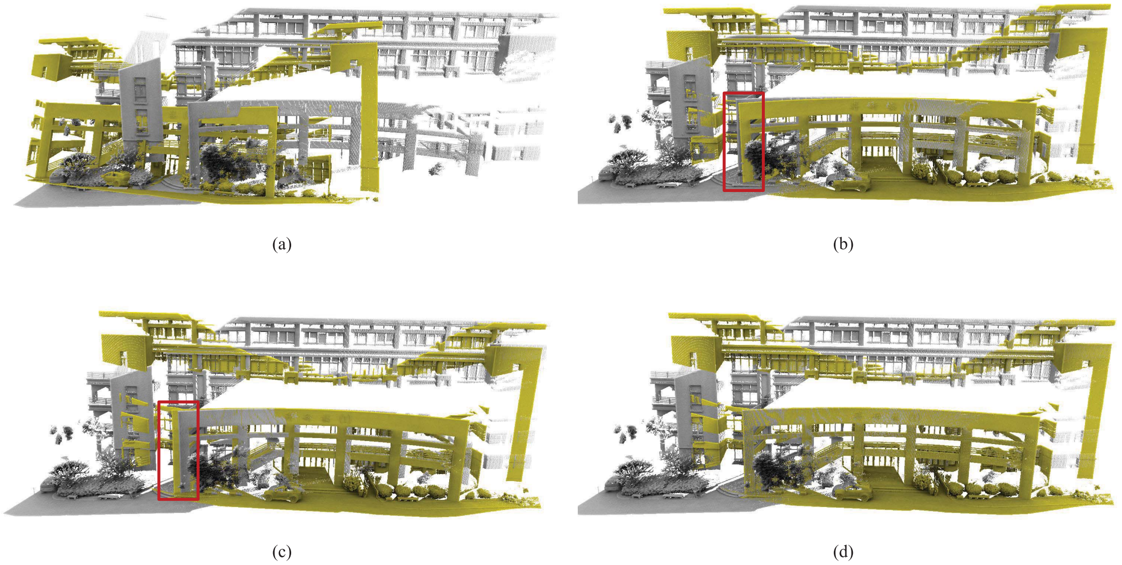


Fig. 17. Comparison results of four registration methods. (a) Sparse ICP. (b) 4PCS. (c) Super 4PCS. (d) Ours.

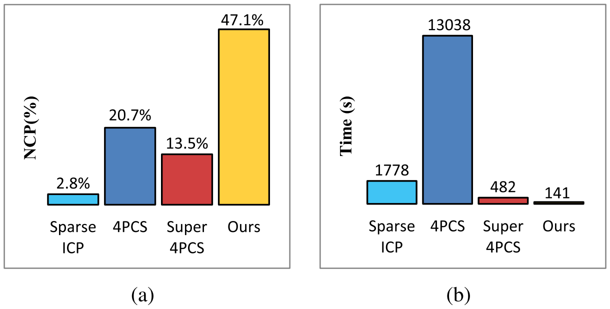


Fig. 18. Comparison of the registration performance. (a) Percentage of NCP over the original point clouds. (b) Running time in seconds.

TABLE II
COMPUTATIONAL RESULTS OF REGISTRATION

Data	# points	# lines	# bases	NCP (%)
1	14,590,438	16784	648	22%
2	4,818,481	7420	806	53%
3	13,998,082	16374	1087	14%
4	14,597,774	7170	1588	38%
5	16,782,619	19764	771	49%
6	16,664,319	20944	751	66%
7	17,018,888	21894	769	55%
8	17,547,442	19452	1153	36%

to the other point clouds after alignment, the threshold distance of NCP is set to 0.05 m.

From Table II, one can observe that the number of extracted line segments accounts for nearly 0.1% of the original point clouds, while the number of bases is under 10% of the line segments. Thus, it is efficient to find an optimal pair of bases for alignment.

Furthermore, we compared our results to an advanced, ICP-based method sparse ICP [42], an efficient four-congruent-points-based 4PCS [43], and a speed-up version of 4PCS, Super 4PCS [44]. We use available implementations of the algorithms and fine tune the parameters.

As shown in Fig. 17, first, sparse ICP totally fails, because it is not suitable for the alignment in arbitrary initial positions. Second, 4PCS and Super 4PCS obtain a coarse registration, but less accurate [see the red box in Fig. 17 (b) and (c)]. The major reason is that 4PCS and Super 4PCS only use the coplanar congruent-points without considering the boundary information. In contrast, the proposed method takes advantage of line boundary information, thus obtains a better result [see Fig. 18(a) for NCP comparison]. We also evaluate time performance of four methods, where the proposed method is fastest (the time of the proposed method has involved line segment extraction, which spends 116 s).

As a conclusion, after extracting the line segments, the size of data is significantly reduced, which makes our registration efficient. Meanwhile, our line segments well preserve the boundary and structure information of the original point cloud. These information, which is consistent to the different view of scans, is reliable for registration.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an efficient line segment extraction method for 3-D large-scale point clouds with noise and outliers. The given point clouds are first segmented into facets by performing local k-means clustering on carefully selected seeds. The proposed facet segmentation method avoids the loss of plane boundary details compared with the other oversegmentation methods. The obtained facets provide sufficient information for determining linear features in the local planar region. In order to group these linear features into 3-D line segment, we present an efficient 3-D line segment grouping method, which introduces the "NFAs" into 3-D point cloud to filter the false positive detections.

The proposed method has been tested on various point cloud data, including the point clouds sampled from mesh models, the point clouds obtained by laser devices, and the synthetic point clouds with noise or outliers added. We also compared the proposed method with several other methods and provide both quantitative and visual comparison results. The experimental results show that our algorithm is efficient, insensitive to noise and outliers, and produces more accurate and complete line segments than the comparison methods.

We believe that many applications can benefit from our extracted line segments. As an example, we employ the extracted line segments on solving the point cloud registration problem. The proposal line-based registration method is reliable and efficient, especially useful for the man-made environment, which contains many line and planar structures.

In future, we will improve our line grouping method by considering latest research results on human perception of oriented stimulus [45]. And we also want to consider more global constraints (such as parallelism, orthogonality, and symmetry in urban environments) to further reduce the false detections. As a further work of 3-D line segment-based registration, it is also interesting to study the problem of aligning 2-D line segments [46] to 3-D line segments, which can be used in 2-D–3-D matching.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] D. Gonzalez-Aguilera, E. Crespo-Matellan, D. Hernandez-Lopez, and P. Rodriguez-Gonzalvez, "Automated urban analysis based on Lidar-derived building models," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 3, pp. 1844–1851, Mar. 2013.

[2] M. Shahzad and X. X. Zhu, "Robust reconstruction of building facades for large areas using spaceborne TomoSAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 752–769, Feb. 2015.

[3] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, "Automated road information extraction from mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 194–205, Feb. 2015.

[4] H. Guan *et al.*, "Iterative tensor voting for pavement crack extraction using mobile laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1527–1537, Mar. 2015.

[5] Y. Lin *et al.*, "Line segment extraction for large scale unorganized point clouds," *J. Photogram. Remote Sens.*, vol. 102, pp. 172–183, Apr. 2015.

[6] G. Zheng, L. M. Moskal, and S. H. Kim, "Retrieval of effective leaf area index in heterogeneous forests with terrestrial laser scanning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 777–786, Feb. 2013.

[7] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.

[8] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.

[9] J.-Y. Rau and L.-C. Chen, "Robust reconstruction of building models from three-dimensional line segments," *Photogram. Eng. Remote Sens.*, vol. 69, no. 2, pp. 181–188, Feb. 2003.

[10] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1554–1567, Mar. 2010.

[11] D. Ceylan, N. J. Mitra, H. Li, T. Weise, and M. Pauly, "Factored facade acquisition using symmetric line arrangements," *Comput. Graph. Forum*, vol. 31, no. 2, pp. 671–680, May 2012.

[12] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, and A. Schilling, "Symmetry detection using feature lines," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 697–706, Apr. 2009.

[13] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3685–3691.

[14] A. Habib, M. Ghanma, M. Morgan, and R. Al-Ruzouq, "Photogrammetric and LiDAR data registration using linear features," *Photogram. Eng. Remote Sens.*, vol. 71, no. 6, pp. 699–707, Jul. 2005.

[15] P. Borges, R. Zlot, M. Bosse, S. Nuske, and A. Tews, "Vision-based localization using an edge map extracted from 3D laser range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4902–4909.

[16] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[17] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, Oct. 2011.

[18] C. J. Taylor and D. J. Kriegman, "Structure and motion from line segments in multiple images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 11, pp. 1021–1032, Nov. 1995.

[19] S. Heuel and W. Forstner, "Matching, reconstructing and grouping 3D lines from multiple views using uncertain projective geometry," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Dec. 2001, pp. 517–524.

[20] D. Martinec and T. Pajdla, "Line reconstruction from many perspective images by factorization," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2003, pp. 497–502.

[21] A. Jain, C. Kurz, T. Thormählen, and H.-P. Seidel, "Exploiting global connectivity constraints for reconstruction of 3D line segments from images," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1586–1593.

[22] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Ridge-valley lines on meshes via implicit surface fitting," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 609–612, Aug. 2004.

[23] S.-K. Kim, "Extraction of ridge and valley lines from unorganized points," *Multimedia Tools Appl.*, vol. 63, no. 1, pp. 265–279, Mar. 2013.

[24] J. Daniels, L. K. Ha, T. Ochotta, and C. T. Silva, "Robust smooth feature extraction from point clouds," in *Proc. IEEE Int. Conf. Shape Modeling Appl.*, Jun. 2007, pp. 123–136.

[25] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of closed sharp edges in point clouds using normal estimation and graph theory," *Comput. Aided Design*, vol. 39, no. 4, pp. 276–283, Apr. 2007.

[26] M. Sunkel, S. Jansen, M. Wand, E. Eisemann, and H.-P. Seidel, "Learning line features in 3D geometry," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 267–276, Apr. 2011.

[27] S. Pu and G. Vosselman, "Knowledge based reconstruction of building models from terrestrial laser scanning data," *J. Photogram. Remote Sens.*, vol. 64, no. 6, pp. 575–584, May 2009.

[28] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation—Supervoxels for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2027–2034.

[29] S. C. Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 304–311.

[30] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1261–1268.

[31] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Inf. Theory*, vol. 29, no. 4, pp. 551–559, Jul. 1983.

[32] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 7–23, Oct. 2000.

[33] A. Desolneux, L. Moisan, and J. M. Morel, *From Gestalt Theory to Image Analysis. A Probabilistic Approach*. New York, NY, USA: Springer, 2008.

[34] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. Hoboken, NJ, USA: Wiley, 2005.

[35] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[36] A. Gressin, C. Mallet, J. Demantké, and N. David, "Towards 3D LiDAR point cloud registration improvement using optimal neighborhood knowledge," *J. Photogram. Remote Sens.*, vol. 79, pp. 240–251, May 2013.

[37] M. Poreba and F. Goulette, "A robust linear feature-based procedure for automated registration of point clouds," *Sensors*, vol. 15, no. 1, pp. 1435–1457, Jan. 2015.

[38] E. Renaudin, A. Habib, and A. P. Kersting, "Featured-based registration of terrestrial laser scans with minimum overlap using photogrammetric data," *ETRI J.*, vol. 33, no. 4, pp. 517–527, 2011.

[39] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A*, vol. 4, no. 4, pp. 629–642, 1987.

[40] S. Irani and P. Raghavan, "Combinatorial and experimental results for randomized point matching algorithms," in *Proc. 12th Annu. Symp. Comput. Geometry*, 1996, pp. 68–77.

[41] P. J. Besl and D. N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[42] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Comput. Graph. Forum*, vol. 32, no. 5, pp. 113–123, Aug. 2013.

[43] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 15–19, Aug. 2008.

[44] N. J. Mitra, N. Mellado, and D. Aiger, "Super 4PCS fast global pointcloud registration via smart indexing," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, Aug. 2014.

[45] C. A. Ramachandra and B. W. Mel, "Computing local edge probability in natural scenes from a population of oriented simple cells," *J. Vis.*, vol. 13, no. 14, pp. 1–15, Dec. 2013.

[46] G. Papari and N. Petkov, "Edge and line oriented contour detection: State of the art," *Image Vis. Comput.*, vol. 29, nos. 2–3, pp. 79–103, 2011.

Yangbin Lin received the B.Sc., M.Sc., and Ph.D. degrees from Xiamen University, Xiamen, China, in 2008, 2011, and 2016, respectively, all in computer science.

He was a Research Assistant with the City University of Hong Kong, Hong Kong, in 2010, and a Software Engineer with Google Company (Shanghai) from 2011 to 2012. He is currently a Faculty Member with Computer Engineering College, Jimei University, Xiamen. His research interests include point cloud, graphics, and optimization.

Cheng Wang (M'11–SM'16) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2002.

He is currently a Professor and Deputy Director with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, and Vice Dean of the School of Information Science and Technology with Xiamen University, Xiamen, China. His research interests include pattern recognition, remote sensing image processing, mobile LiDAR data analysis, and multisensor fusion.

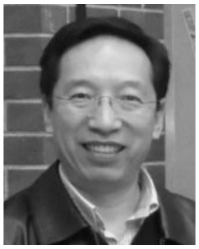
Dr. Wang is a Council Member of the Chinese Society of Image and Graphics, a Senior Member of the IEEE Geoscience and Remote Sensing Society, and the Chair of the ISPRS WG I/6 from 2016 to 2020. He has co-authored over 130 papers published in refereed journals, such as the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Transactions on Intelligent Transportation Systems, the IEEE Geoscience and Remote Sensing Letters, the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, and the *ISPRS Journal of Photogrammetry and Remote Sensing* and conferences, such as IGARSS and ISPRS.

Bili Chen received the B.Sc. degree in software engineering from Fuzhou University, Fuzhou, China, in 2008, and the M.Sc. and Ph.D. degrees in computer science from Xiamen University, Xiamen, China, in 2011 and 2015, respectively.

Her research interests include multi-objective optimization, evolutionary computation, meta heuristics, and portfolio optimization problem.

Dawei Zai received the B.S. degree in aircraft design and engineering from Xi'an Jiaotong University, Xi'an, China, in 2010. He is currently pursuing the Ph.D. degree with the Department of Communication Engineering, Xiamen University.

His research interests include computer vision, machine learning, and mobile LiDAR point cloud data processing.



Jonathan Li (M'00–SM'11) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa.

He is currently a Full Professor and the Head of the Mobile Sensing and Geodata Science Laboratory with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON, Canada. He is also with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China. His research interests include information extraction from LiDAR point clouds and from earth observation images.

Dr. Li has co-authored over 300 publications, over 150 of which were published in refereed journals, including the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Transactions on Intelligent Transportation Systems, the IEEE Geoscience and Remote Sensing Letters, the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, the *ISPRS Journal of Photogrammetry and Remote Sensing*, the *International Journal of Remote Sensing*, *Photogrammetric Engineering & Remote Sensing*, and the *Remote Sensing of Environment*. He is the Chair of the ICA Commission on Sensor-Driven Mapping from 2015 to 2019, the Chair of the ISPRS Working Group I/2 on LiDAR, Air- and Spaceborne Sensing Systems for 3-D Mapping from 2016 to 2020, an Associate Editor of the IEEE Transactions on Intelligent Transportation Systems and the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, and an Associate Editor-in-Chief of *Sensors*.