ISPRS Journal of Photogrammetry and Remote Sensing 137 (2018) 112–133

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

An efficient global energy optimization approach for robust 3D plane segmentation of point clouds

Zhen Dong [a,b], Bisheng Yang [a,*], Pingbo Hu [a], Sebastian Scherer [b,*]

[a] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
[b] The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 19 July 2017
Received in revised form 6 November 2017
Accepted 16 January 2018

Keywords:
Plane segmentation
Multiscale supervoxel
Hybrid region growing
Energy optimization
Simulated annealing
Guided sampling

ABSTRACT

Automatic 3D plane segmentation is necessary for many applications including point cloud registration, building information model (BIM) reconstruction, simultaneous localization and mapping (SLAM), and point cloud compression. However, most of the existing 3D plane segmentation methods still suffer from low precision and recall, and inaccurate and incomplete boundaries, especially for low-quality point clouds collected by RGB-D sensors. To overcome these challenges, this paper formulates the plane segmentation problem as a global energy optimization because it is robust to high levels of noise and clutter. First, the proposed method divides the raw point cloud into multiscale supervoxels, and considers planar supervoxels and individual points corresponding to nonplanar supervoxels as basic units. Then, an efficient hybrid region growing algorithm is utilized to generate initial plane set by incrementally merging adjacent basic units with similar features. Next, the initial plane set is further enriched and refined in a mutually reinforcing manner under the framework of global energy optimization. Finally, the performances of the proposed method are evaluated with respect to six metrics (i.e., plane precision, plane recall, under-segmentation rate, over-segmentation rate, boundary precision, and boundary recall) on two benchmark datasets. Comprehensive experiments demonstrate that the proposed method obtained good performances both in high-quality TLS point clouds (i.e., SEMANTIC3D.NET dataset) and low-quality RGB-D point clouds (i.e., S3DIS dataset) with six metrics of (94.2%, 95.1%, 2.9%, 3.8%, 93.6%, 94.1%) and (90.4%, 91.4%, 8.2%, 7.6%, 90.8%, 91.7%) respectively.

1. Introduction

Automatic 3D plane segmentation is necessary in many applications, including point cloud registration (Dold and Brenner, 2006; Theiler and Schindler, 2012; Xiao et al., 2013a; Yang et al., 2016; Lin et al., 2017), building information model (BIM) reconstruction (Vosselman et al., 2004; Sampath and Shan, 2010; Jung et al., 2014; Oesau et al., 2014; Yan et al., 2014; Yang and Wang, 2016; Xu et al., 2017), simultaneous localization and mapping (SLAM) (Xiao et al., 2013b; Pham et al., 2016; Lenac et al., 2017), and point cloud data compression (Vaskevicius et al., 2010; Kaushik and Xiao, 2012). Three-dimensional point clouds generated by various 3D sensing technologies (e.g. 3D laser scanners, multi-view stereo techniques and RGB-D cameras) are frequently contaminated with outliers, noise, occlusion, and clutter, raising great challenges for robust and efficient plane segmentation. In the last decades, extensive studies have been done to improve the efficiency and robustness of 3D plane segmentation, which can be roughly categorized into four categories, i.e., region growing based methods (Tóvári and Pfeifer, 2005; Deschaud and Goulette, 2010; Nurunnabi et al., 2012; Xiao et al., 2013b; Yang and Dong, 2013; Vo et al., 2015), model fitting based methods (Vosselman et al., 2004; Boulaassal et al., 2007; Schnabel et al., 2007; Tarsha-Kurdi et al., 2007; Oehler et al., 2011; Chen et al., 2014; Xu et al., 2017), feature clustering-based methods (Filin, 2002; Filin and Pfeifer, 2006; Biosca and Lerma, 2008; Zhou et al., 2016; Kim et al., 2016), and global energy optimization based methods (Kim and Shan, 2011; Yan et al., 2014; Oesau et al., 2014; Wang et al., 2016; Pham et al., 2016).

1.1. Region growing based methods

Region growing based methods extract 3D planes by progressively merging adjacent points or voxels with similar features (e.g. normal vector). Tóvári and Pfeifer (2005) proposed a

* Corresponding authors.
E-mail addresses: bshyang@whu.edu.cn (B. Yang), basti@andrew.cmu.edu (S. Scherer).

https://doi.org/10.1016/j.isprsjprs.2018.01.013

point-based region growing (PBRG) method that picked randomly a seed point and then merged its neighboring points if they fulfilled the predefined criteria, i.e. similar normal vector and small distance to the adjusting plane. Deschaud and Goulette (2010) presented a voxel-based region growing (VBRG) algorithm to improve the efficiency of PBRG by replacing points with voxels during region growing. Nurunnabi et al. (2012) improved the robustness of the PBRG by estimating more accurate normal vectors using fast minimum covariance determinant based robust principal component analysis approach. Xiao et al. (2013b) proposed a novel hybrid region growing algorithm (HRG) to detect planes in both structured and unstructured environments by using point and voxel as growth units. More precisely, the algorithm utilized single point and voxel as growth units for nonplanar and planar regions respectively, which were fast to compute, and yet exhibiting good accuracy. Vo et al. (2015) introduced an adaptive Octree-based region growing algorithm for the fast surface patch segmentation by incrementally grouping adjacent voxels with similar saliency feature, which was faster and able to achieve better precision, recall, and fitness scores compared with the conventional region growing method.

Region growing based methods are widely used for plane segmentation as they are easily implemented (Vo et al., 2015). However, they are not robust to noise, varying point density, and occlusion, and the segmentation quality strongly depends on the predefined growing criteria and the strategies for seeds selecting (Teboul et al., 2010).

1.2. Model fitting based methods

The most widely employed model fitting-based methods for plane segmentation are random sample consensus (RANSAC) (Fischler and Bolles, 1981) and Hough transform (HT) (Ballard, 1981). Both have been proven to successfully detect shapes in 2D as well as 3D (Schnabel et al., 2007). The RANSAC paradigm generates a number of model proposals by randomly sampling data points and then selects the model with the largest set of inliers with respect to some fixed threshold (Isack and Boykov, 2012). Many publications proposed various generalizations of RANSAC for plane segmentation. For example, Boulaassal et al. (2007) utilized RANSAC algorithm to extract planar parts of facades scanned by a terrestrial laser scanner. Chen et al. (2014) proposed an enhanced RANSAC algorithm for building roof segmentation, which significantly improved the efficiency of plane segmentation by using a novel localized sampling strategy. Schnabel et al. (2007) extended the RANSAC algorithm to detect multiple geometric primitives (i.e., planes, spheres, cylinders, cones and tori) in unorganized point clouds. Xu et al. (2015) introduced a new weighted RANSAC algorithm that combined the point-plane distance and the normal vector consistency for roof point cloud segmentation. In order to improve the efficiency of multiple geometric primitive extraction, Xu et al. (2017) first divided the point cloud into some individual segments using Locally Convex Connected Patches (LCCP) (Christoph et al., 2014), and then recognized and extracted geometric primitives from each segment using the method of Schnabel et al. (2007).

The HT algorithm maps every point into a discretized parameter space and then extracts planes by selecting those parameter space with a significant amount of votes. Vosselman et al. (2004) proposed an efficient variant of the HT for plane segmentation, which improved the efficiency and reliability of conventional HT by determining the plane parameters in two separate steps, i.e., determination of the plane normal vector and establishment of the distance from the plane to the origin. Tarsha-Kurdi et al. (2007) compared the performance of HT and RANSAC algorithms for 3D plane segmentation in terms of processing time and sensitivity to point cloud characteristics. The analytic comparison shows that RANSAC algorithm is more efficient and robust than HT algorithm. Oehler et al. (2011) presented an efficient multi-resolution approach to segment a 3D point cloud into planar components by combining HT and RANSAC algorithms. More precisely, the algorithm first detected the coplanar clusters with the HT technology then extracted connected components on these clusters and determined the best plane through RANSAC.

Although the reported RANSAC and HT approaches work very well for 3D plane segmentation on point cloud with low levels of noise and clutter, these algorithms still have some disadvantages. First, HT and RANSAC are sensitive to the selection of parameter value (i.e., the size of the cell in discretized parameter space, and the threshold for inlier). Second, the segmentation quality is sensitive to the point cloud characteristics (i.e., density, positional accuracy, and noise) (Vo et al., 2015). Moreover running RANSAC sequentially to detect multiple planes is widely known to be sub-optimal since the inaccuracies in detecting the first planes will heavily affect the subsequent planes (Pham et al., 2016).

1.3. Feature clustering-based methods

Feature clustering-based methods organize the point clouds into primitives based on certain pre-calculated local surface properties (e.g., normal vector, saliency feature). Filin (2002) employed mode-seeking algorithm based on seven-dimensional feature space (i.e., position (3), plane function (3) and height difference (1)) to cluster airborne laser scanning data into 3D planes. In a follow-up work, Filin and Pfeifer (2006) improved the quality of the computed features by utilizing a novel slope adaptive neighborhood system. Biosca and Lerma (2008) used Possibilistic C-Means (PCM) algorithm based on the similar feature space with Filin (2002) to the segmentation of planar surfaces, which could automatically determine the number of planes and improve the robustness to noise and outliers. Zhou et al. (2016) proposed a new method to extract planar features from the range image of a point cloud scanned by terrestrial laser scanning system. Similar with Filin and Pfeifer (2006), a plane was parameterized by its normal vector and the distance between the origin and the plane, then the planar parameters was segmented using the Iso cluster unsupervised classification method. Kim et al. (2016) proposed a robust and efficient segmentation methodology for segmentation of planar surfaces from airborne and terrestrial laser scanning data. Specifically, they increased the homogeneity of the laser point attributes by defining a neighborhood based on the shape of the surface and reduced the dimensions of the attribute space using the magnitude of the normal vector. In spite of the popularity and efficiency of feature clustering-based methods, they suffer the difficulty in neighborhood definition and are sensitive to noise and outliers (Yan et al., 2014).

1.4. Global energy optimization based methods

More recently, the global energy optimization based methods have been widely used to geometric primitive extraction both in the 2D image and 3D point cloud, which formulate the geometric primitive extraction as an energy optimization problem. The widespread applications of energy optimization based methods in the field of 2D image process can be found in (Yu et al., 2011; Isack and Boykov, 2012; Pham et al., 2014; Tennakoon et al., 2016). In the case of 3D point cloud process, (Kim and Shan, 2011) adopted the multiphase level set approach to segment planar roof primitives under an energy minimization formulation. In follow-up work, Yan et al. (2014) presented a global plane fitting approach for roof segmentation from lidar point clouds. They first segmented the roof into initial planes based on conventional region growing approach, and then refined the initial planes by minimizing a global energy function consisting of data cost term, smooth cost term

and label cost term. Similar to Yan et al. (2014), Oesau et al. (2014) proposed a graph-cut based formulation for abstracting primitives in indoor scenes. Wang et al. (2016) detected multiplane from 3D point clouds by labeling points instead of greedy searching planes. Specifically, they first generated initial planes using RANSAC algorithm, and then constructed the energy function and minimized the energy function (Delong et al., 2012). Pham et al. (2016) formulated the plane extraction task as a global energy function, which jointly considered plane fidelities and geometric consistencies between planes (i.e., mutually orthogonal or parallel relations).

Plane segmentation can be better formulated as an optimization problem with a global energy function describing the quality of the overall solution. Therefore, these methods can produce spatially coherent planes and improve the plane segmentation quality (Yan et al., 2014). What's more, these methods are more robust to high levels of noise and clutter compared with other methods (Isack and Boykov, 2012). However, the energy optimization based methods are computationally expensive for plane segmentation (Pham et al., 2016) and heavily dependent on the adequacy of initial inputs.

1.5. Contributions

Although the reported methods can generally provide satisfactory plane segmentation results, they still have limitations. First, many existing greedy methods (e.g., region growing based methods, model fitting based methods, and feature clustering-based methods) work only on examples with low levels of noise and clutter (Isack and Boykov, 2012). Second, the global energy optimization based methods are computationally expensive in large-scale point cloud (Pham et al., 2016), and heavily dependent on the adequacy of initial inputs. Third, many existing algorithms have difficulty in detecting small-scale planes in point cloud with high levels of noise and clutter. To overcome these limitations, this paper presents a Robust and Efficient 3D Plane Segmentation method (REPS) and validates its performance on two benchmark datasets. The proposed method first divides the raw point cloud into multiscale supervoxels or individual points to reduce the search space for initial plane segmentation. Then, the initial plane set is generated by an efficient hybrid region growing algorithm. Finally, the initial plane set is further enriched and refined under the framework of global energy optimization. The main contributions of the proposed method are as follows.

(1) The proposed method improves the robustness to noise, outliers, and clutter by formulating plane segmentation issue as an energy optimization problem with a global energy function balancing geometric errors, spatial coherence, and the number of planes.
(2) The proposed method improves the efficiency of global energy optimization based plane segmentation. First, the proposed method represents the raw point cloud with the hybrid representation of multiscale supervoxels and individual points, significantly decreasing the number of basic units for plane segmentation. Second, the proposed method leverages a novel guided sampling technique to improve the probabilities of selecting undetected planes, thus dramatically accelerating the convergence speed of global energy optimization.
(3) Another important feature of our approach is that initial plane set enrichment and refinement are conducted in a mutually reinforcing manner. This synergy enhances the capability of the algorithm to extract undetected planes, especially planes with few points, or small scales, thus improving the completeness of plane segmentation.

The rest of this paper is organized as follows. Following this introduction, Section 2 gives a detailed description of the proposed 3D plane segmentation method. Then the proposed method is validated in experimental studies in Section Section 3. Finally, the conclusions and future research directions are presented in Section Section 4.

2. The robust and efficient 3D plane segmentation

In this section, we propose a novel 3D plane segmentation method with a successive scheme that includes multiscale supervoxel generation (Section 2.1), hybrid region growing (Section Section 2.2) and global energy optimization (Section Section 2.3). An illustration of the proposed 3D plane segmentation method is given in Fig. 1. The proposed method first divides the raw point cloud into multiscale supervoxels, and considers planar supervoxels and individual points corresponding to nonplanar supervoxels as basic units, as shown in Fig. 1b. Then an efficient hybrid region growing algorithm is utilized to generate initial plane set by incrementally merging adjacent basic units with similar features, as shown in Fig. 1c. Finally, the initial plane set is further enriched and refined in a mutually reinforcing manner under the framework of a global energy optimization method, as shown in Fig. 1d.

2.1. Multiscale supervoxel generation

Plane segmentation from the raw point cloud data is very time-consuming. The proposed method represents the raw point cloud using the hybrid of multiscale supervoxels and individual points to reduce the number of points for plane segmentation. The benefit of using hybrid representation is its high efficiency. The pseudo-code of the multiscale supervoxel generation algorithm is detailed in Algorithm 1.

Algorithm 1 (*Multiscale supervoxel generation*).

Notation:
PC: point cloud;
$r_{\max}, r_{\min}, r, r_\Delta$: the maximum, minimum, current and decreasing rate of supervoxel scale;
S_r: the set of supervoxels at scale r;
Λ_r, Λ'_r: the set of planar and nonplanar supervoxels at scale r;
Λ: the set of planar supervoxels at all scales;
Λ': the remaining points not belonging to any planar supervoxel;
$\varnothing$: the empty set.

Input: PC
Output: Λ, Λ'
1 Calculate the geometric features of each point in point cloud PC (Section 2.1.1)
2 **initialization:** $r \leftarrow r_{\max}, \Lambda \leftarrow \varnothing$
3 **while** $PC \neq \varnothing$ && $r \geqslant r_{\min}$ **do**
4 Oversegment point cloud PC into supervoxels S_r (Section 2.1.2)
5 Classify supervoxels S_r into Λ_r and Λ'_r (Section 2.1.2)
6 $PC \leftarrow \Lambda'_r$
7 $\Lambda \leftarrow \Lambda \cup \Lambda_r$
8 $r \leftarrow r * r_\Delta$
9 **end while**
10 $\Lambda' \leftarrow PC$

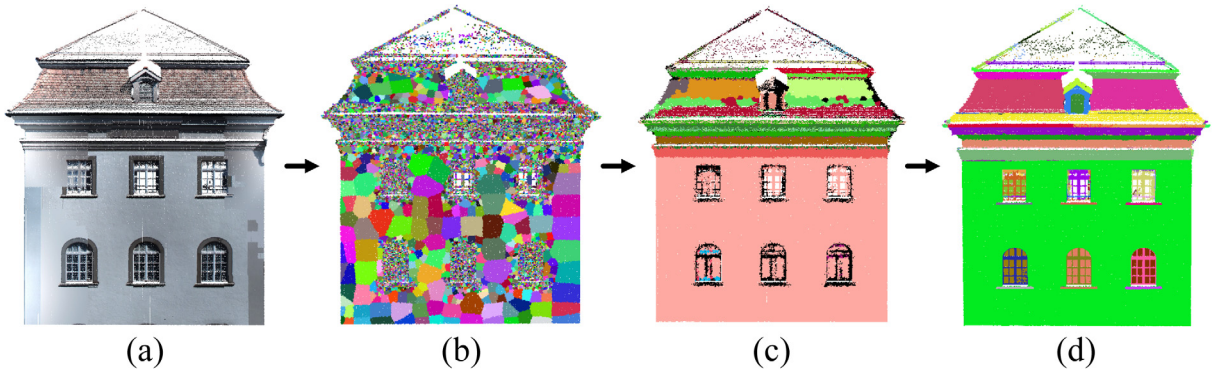

Fig. 1. The illustration of the proposed 3D plane segmentation algorithm: (a) the original point cloud, (b) the output of multiscale supervoxel generation, (c) the output of hybrid region growing, and (d) the output of global energy optimization.

2.1.1. Pointwise geometric features calculation

Before supervoxel generation, the proposed method calculates the geometric features (i.e., parameters of tangent plane, and curvature) of each individual point (Lalonde et al., 2006). The geometric features of each point $\boldsymbol{p}(x_p, y_p, z_p)$ are calculated as follows. The covariance matrix $\boldsymbol{M}_{3\times3}$ is constructed using point $\boldsymbol{p}$ and its k adjacent points according to Eq. (1).

$$\boldsymbol{M}_{3\times3} = \frac{1}{k}\sum_{i=1}^{k}(\boldsymbol{q}_i - \boldsymbol{p})(\boldsymbol{q}_i - \boldsymbol{p})^T \tag{1}$$

where $\boldsymbol{q}_i$ is the adjacent point of $\boldsymbol{p}$. The eigenvalues $\lambda_1^{(M)} \geqslant \lambda_2^{(M)} \geqslant \lambda_3^{(M)}$ and the corresponding eigenvectors $\boldsymbol{e}_1^{(M)}, \boldsymbol{e}_2^{(M)}, \boldsymbol{e}_3^{(M)}$ are calculated by performing an eigenvalue decomposition on $\boldsymbol{M}_{3\times3}$. The geometric features of $\boldsymbol{p}$ are calculated as Eq. (2).

$$\begin{cases} \boldsymbol{n}_p(a_p, b_p, c_p) = \boldsymbol{e}_3^{(M)} \\ d_p = -(a_p x_p + b_p y_p + c_p z_p) \\ f_p = \frac{\lambda_3^{(M)}}{\lambda_1^{(M)} + \lambda_2^{(M)} + \lambda_3^{(M)}} \end{cases} \tag{2}$$

where $\boldsymbol{n}_p$ and d_p are the parameters of the tangent plane of point $\boldsymbol{p}$, and f_p is the curvature. Specifically, $\boldsymbol{n}_p$ is the normal vector and d_p is the distance from the origin to the tangent plane. Each point $\boldsymbol{p}$ and its geometric features are described by one 8-tuple, which will be used for the following processes.

$$\boldsymbol{F}_p = (x_p, y_p, z_p, a_p, b_p, c_p, d_p, f_p) \tag{3}$$

2.1.2. Supervoxels generation and classification

After the pointwise geometric features calculation, the Voxel Cloud Connectivity Segmentation (VCCS) method (Papon et al., 2013) is employed to oversegment the points into supervoxels at each scale (i.e., supervoxel resolution). In this work, we are only interested in geometric features, so we use the spatial distance, and normal vector deviation for supervoxel generation at each scale. The proposed method first oversegments the point cloud into supervoxels at scale r, and classifies them as planar and nonplanar supervoxels. Then the nonplanar supervoxels are further divided into supervoxels at next smaller scale $r * r_\Delta$. The algorithm repeats the above steps until all supervoxels are classified as planar supervoxels or the current scale r is less than the predefined minimum scale $r_{\min}$. More specifically, each supervoxel $\boldsymbol{s}$ is classified as a planar or nonplanar supervoxel as following.

First, the covariance matrix $\boldsymbol{C}_{3\times3}$ is constructed using points belonging to supervoxel $\boldsymbol{s}$ according to Eq. (4).

$$\begin{cases} \boldsymbol{C}_{3\times3} = \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{p}_j - \boldsymbol{p}_s)(\boldsymbol{p}_j - \boldsymbol{p}_s)^T \\ \boldsymbol{p}_s = \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{p}_j \end{cases} \tag{4}$$

where $\boldsymbol{p}_j$ is the point belonging to supervoxel $\boldsymbol{S}$, $\boldsymbol{p}_s$ is the center point of supervoxel $\boldsymbol{S}$. Let $\lambda_1^{(C)} \geqslant \lambda_2^{(C)} \geqslant \lambda_3^{(C)}$ and $\boldsymbol{e}_1^{(C)}, \boldsymbol{e}_2^{(C)}, \boldsymbol{e}_3^{(C)}$ be eigenvalue and eigenvector of $\boldsymbol{C}_{3\times3}$.

Then the saliency features g_1, g_2 and g_3 (Yang et al., 2015) of supervoxel $\boldsymbol{s}$ are calculated as Eq. (5).

$$\begin{cases} g_1 = \frac{\sqrt{\lambda_1^{(C)}} - \sqrt{\lambda_2^{(C)}}}{\sqrt{\lambda_1^{(C)}}} \\ g_2 = \frac{\sqrt{\lambda_2^{(C)}} - \sqrt{\lambda_3^{(C)}}}{\sqrt{\lambda_1^{(C)}}} \\ g_3 = \frac{\sqrt{\lambda_3^{(C)}}}{\sqrt{\lambda_1^{(C)}}} \end{cases} \tag{5}$$

Finally, $\boldsymbol{s}$ is classified into two categories (i.e., planar supervoxel or nonplanar supervoxel) according to its saliency features, as Eq. (6).

$$h_s = \begin{cases} \text{planar}, & \text{if } (g_2 > g_1) \wedge (g_2 > g_3) \\ \text{nonplanar}, & \text{otherwise} \end{cases} \tag{6}$$

And the geometric features of each supervoxel $\boldsymbol{s}$ is calculated as Eq. (7).

$$\begin{cases} \boldsymbol{n}_s(a_s, b_s, c_s) = \boldsymbol{e}_3^{(C)} \\ d_s = -(a_s x_s + b_s y_s + c_s z_s) \\ f_s = \frac{\lambda_3^{(C)}}{\lambda_1^{(C)} + \lambda_2^{(C)} + \lambda_3^{(C)}} \end{cases} \tag{7}$$

where $\boldsymbol{n}_s$ and d_s are the parameters of the tangent plane, and f_s is the curvature. The supervoxel $\boldsymbol{S}$ and its features are described by one 8-tuple, which will be used for the following processes.

$$\boldsymbol{F}_s = \{(x_s, y_s, z_s), (a_s, b_s, c_s), d_s, f_s\} \tag{8}$$

After multiscale supervoxel generation, the planar supervoxels and the individual points corresponding to the nonplanar supervoxels are considered as the basic units for the following hybrid region growing and global energy optimization procedures. Specifically, a basic unit $\boldsymbol{\omega}(x_\omega, y_\omega, z_\omega)$ and its features derived from the corresponding supervoxel or individual point are described by one 8-tuple:

$$\boldsymbol{F}_\omega = \{(x_\omega, y_\omega, z_\omega), (a_\omega, b_\omega, c_\omega), d_\omega, f_\omega\} \tag{9}$$

where $(x_\omega, y_\omega, z_\omega)$ is the coordinate, $\boldsymbol{n}_\omega(a_\omega, b_\omega, c_\omega)$ and d_ω are the parameters of the tangent plane of $\boldsymbol{\omega}$, and f_ω is the curvature.

Fig. 2 illustrates an example of multiscale supervoxel generation and classification. Fig. 2a shows the original point cloud (1,077,794 points) colored by its texture. The proposed method first oversegments the point cloud into supervoxels at the largest scale. Fig. 2b shows the corresponding result of supervoxel generation at the largest scale, where each supervoxel is dotted in one color. Then the algorithm classifies them as planar and nonplanar supervoxels. Fig. 2c denotes the corresponding supervoxel classification result, where red and blue regions represent planar and nonplanar supervoxels respectively. Next the nonplanar supervoxels in Fig. 2c (i.e., the blue regions) are further divided into supervoxels at a smaller scale and classified them as planar and nonplanar supervoxels, as shown in Fig. 2d, and e respectively. The algorithm repeats the above steps until all supervoxels are classified as planar supervoxels or the scale is less than the predefined minimum scale. Fig. 2f and g represent the final multiscale supervoxels and the classification results. Fig. 2h shows the number of basic unites (36,047 units) used for the following processes, where each planar supervoxel is represented by its center point. It is worth noting that, the number of basic units is notably smaller than the number of raw points, thereby significantly reducing the search space (i.e., the number of points) and accelerating the following hybrid region growing and global energy optimization processes.

2.2. Hybrid region growing

In this section, we propose an efficient hybrid region growing method to generate the initial plane set, then further enrich and refine the initial plane set in Section Section 2.3. The planar supervoxels and the individual points corresponding to the nonplanar supervoxels are considered as the basic units for hybrid region growing. The algorithm incrementally merges adjacent basic units with similar features into a set of plane surfaces. The benefit of using hybrid region growing method is its high efficiency. The number of basic units is notably smaller than the number of points, thereby significantly accelerating the region growing process. The pseudo-code of the hybrid region growing algorithm is described in Algorithm 2.

Algorithm 2 (*Hybrid region growing*).

Notation:

Ψ: the set of all the basic units, $\Re$, $\Re'$: the set of basic units belonging to planar and nonplanar surfaces respectively;

$\Psi \setminus (\Re \cup \Re')$: remove basic units belonging to $\Re$ and $\Re'$ from Ψ;

$\boldsymbol{\Omega}$: inliers belonging to the current planar surface, $|\boldsymbol{\Omega}|$: the number of inliers;

$\boldsymbol{\varphi}$: initial seed, $\boldsymbol{\varpi}$: current seed, ζ: candidate basic unit;

$N(\boldsymbol{\varpi})$: the set of adjacent basic units of $\boldsymbol{\varpi}$;

Q: the queue of seeds;

$\theta(\boldsymbol{\varphi}, \zeta)$: the angle between the normal vectors of $\boldsymbol{\varphi}$ and ζ;

$\delta(\boldsymbol{\varphi}, \zeta)$: the difference between d_φ and d_ζ;

τ_θ, τ_δ: the threshold of $\theta(\boldsymbol{\varphi}, \zeta)$, and $\delta(\boldsymbol{\varphi}, \zeta)$ respectively;

κ: the threshold of minimum number for a valid plane;

$\varnothing$: the empty set.

Input: Ψ

Output: $\Re$, $\Re'$

```
initialization: ℜ ← ∅, ℜ′ ← ∅,
while Ψ \ (ℜ ∪ ℜ′) ≠ ∅ do
  Ω ← ∅
  select φ with minimum curvature f_φ in Ψ \ (ℜ ∪ ℜ′)
  Ω ← φ, Q ← φ
  while Q ≠ ∅ do
    ϖ ← Q_pop_front();
    for each unit ζ in N(ϖ) do
      if θ(φ, ζ) ⩽ τ_θ && δ(φ, ζ) ⩽ τ_δ then
        Ω ← Ω ∪ ζ; Q ← Q ∪ ζ;
      end if
    end for
  end while
  if |Ω| ⩾ κ then
    ℜ ← ℜ ∪ Ω
  end if
  else
    ℜ′ ← ℜ′ ∪ Ω
  end else
end while
```

The hybrid region growing method returns the inliers of extracted planar surfaces. The parameters $\boldsymbol{\eta}_t(a_t, b_t, c_t, d_t)$ (i.e., plane function) of each plane $\boldsymbol{P}_t$ are calculated using its corresponding inliers $\boldsymbol{\Omega}_t$ by the least square estimation as Eq. (10).

$$\begin{cases} \hat{\boldsymbol{\eta}}_t = \arg\min_{\boldsymbol{\eta}_t} \sum_{\boldsymbol{\omega} \in \boldsymbol{\Omega}_t} \mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{\eta}_t)^2 \\ \mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{\eta}_t) = \frac{a_t x_\omega + b_t y_\omega + c_t z_\omega + d_t}{\sqrt{a_t^2 + b_t^2 + c_t^2}} \end{cases} \tag{10}$$

Fig. 3 shows an example of the segmented initial plane set and the ground truth, where undetected planes are represented with black, and detected planes are represented with other different colors. There are mainly four kinds of errors in the initial plane set: inaccurate and incomplete boundaries, undetected planes, under-segmentation planes (i.e., planes with similar parameters are segmented into one plane) and over-segmentation planes (i.e., one plane surface is segmented into multiple planes). These issues will be addressed through the following global energy optimization procedure.

2.3. Global energy optimization

In this section, we formulate the plane segmentation issue as a global energy optimization problem due to its robustness to high levels of noise and clutter. Most of the existing energy optimization based approaches first generate abundant initial planes, then determine an optimal subset of the initial planes and their corresponding inliers by minimizing a predefined energy function (Wang et al., 2016; Pham et al., 2016). However, to ensure the adequacy of the initial plane set, the algorithm needs to generate a huge number of initial planes (i.e., the number of generated initial planes is considerably larger than the ground truth), which is time-consuming and also significantly increases the computational effort for the energy optimization (Pham et al., 2016). The proposed method first generates relatively few numbers of initial planes by Algorithm 2, then further enriches and refines the initial plane set alternatingly and in a mutually reinforcing manner, thus improving the efficiency of the energy optimization method and enhancing the capability of extracting undetected planes with few points, or small scales. Meanwhile, in order to avoid local

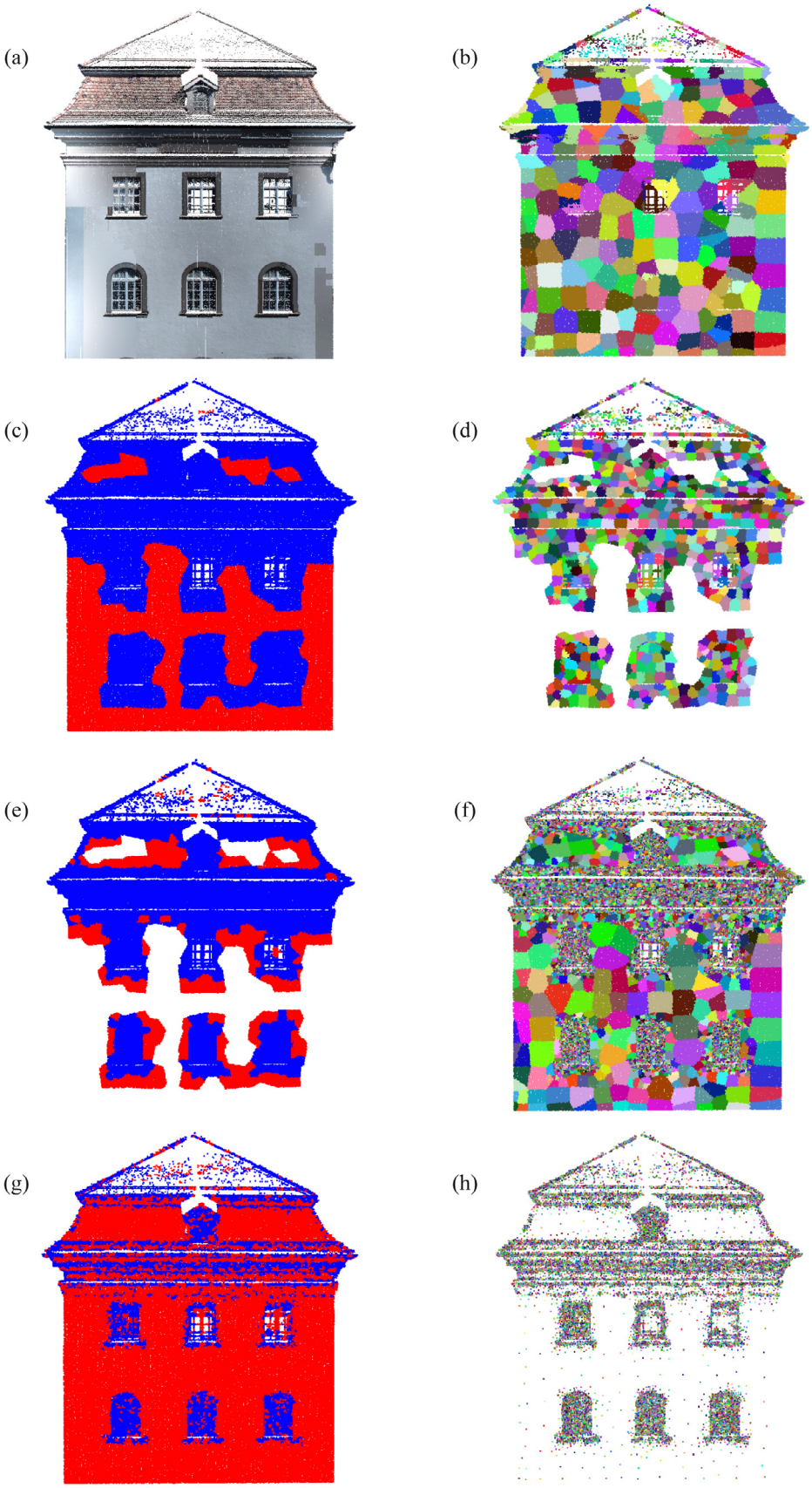

Fig. 2. Multiscale Supervoxel generation: (a) the original point cloud, (b) the generated supervoxels at the largest scale, (c) the supervoxels classification result at the largest scale, (d) the generated supervoxels at the second largest scale, (e) the supervoxels classification result at the second largest scale, (f) the generated multiscale supervoxels, (g) the multiscale supervoxels classification result, and (h) the basic units for the following processes.

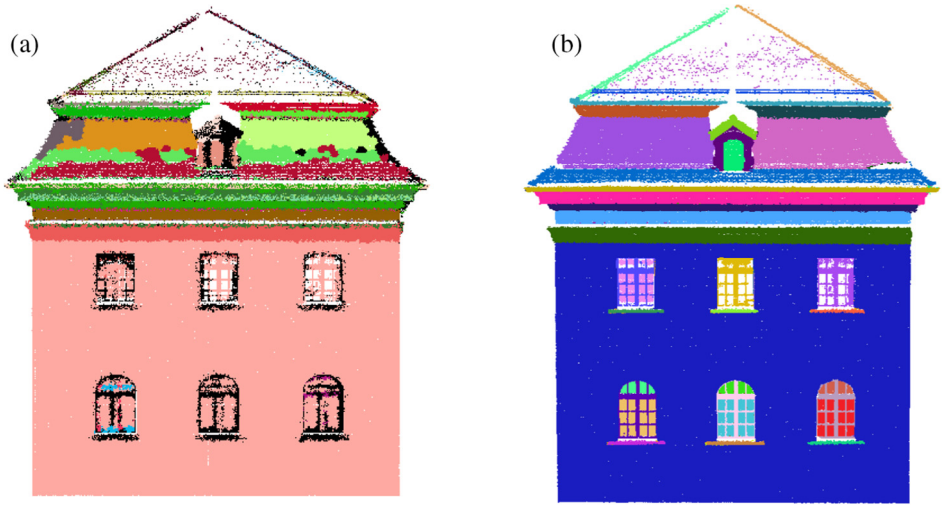

Fig. 3. An example of the segmented initial plane set: (a) the hybrid region growing result, and (b) the ground truth.

minima and ensure the algorithm converge to the global optimum, the simulated annealing approach is leveraged to minimize the overall energy function. The pseudo-code of the global energy optimization algorithm is detailed in Algorithm 3.

Algorithm 3 (*Global energy optimization based plane refinement*).

Notation:
Ψ: the set of all the basic units;
$\mathbf{\Phi} = \{\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2, \cdots \boldsymbol{P}_t \cdots \boldsymbol{P}_{N_u}\}$: the initial plane set;
$\mathbf{\Phi}^{(R)} = \{\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2, \cdots \boldsymbol{P}_t \cdots \boldsymbol{P}_{M_u}\}$: the enriched plane set;
N_u, M_u: the number of planes in $\mathbf{\Phi}$ and $\mathbf{\Phi}^{(R)}$ respectively.
$T, T_\Delta, T_{\min}$: initial temperature, decreasing coefficient, minimum temperature;
PN: the number of the increased new planes in each iteration;
Θ: the set of the increased new planes in each iteration;
CN: the number of consecutive rejecting plane set $\mathbf{\Phi}^{(R)}$;
τ_{CN}: the threshold of CN;
E_{old}, E_{new}: the energy before and after new planes generation;
l_ω: the label of basic unit $\boldsymbol{\omega}$, which means $\boldsymbol{\omega}$ is an inlier of plane $\boldsymbol{P}_{l_\omega}$;
$\boldsymbol{L} = \{l_\omega | \boldsymbol{\omega} \in \mathbf{\Psi}_u\}$: all the labels of basic units.

Input: $\mathbf{\Phi}$ and Ψ
Output: $\mathbf{\Phi}^{(R)}$
initialization:
$T \leftarrow 10, \Delta T \leftarrow 0.99, T_{\min} \leftarrow 0.5, CN \leftarrow 0, \tau_{CN} \leftarrow 5$
while $(CN \leqslant \tau_{CN}) \&\& (T \geqslant T_{\min})$ **do**
  $\mathbf{\Phi}^{(R)} \leftarrow \mathbf{\Phi}$
  Formulate energy function Eq. (11) using $\mathbf{\Phi}^{(R)}$ (Section Section 2.3.1);
  Calculate the energy E_{old} of Eq. (11);
  Generate PN new planes Θ through a guided sampling technique (Section Section 2.3.2).
  $\mathbf{\Phi}^{(R)} \leftarrow \Theta \cup \mathbf{\Phi}^{(R)}$
  Formulate energy function Eq. (11) using $\mathbf{\Phi}^{(R)}$.
  Obtain the labels $\boldsymbol{L}$ by minimizing Eq. (11) via the ∂-expansion algorithm (Section Section 2.3.3).
  Remove the planes without inliers from $\mathbf{\Phi}^{(R)}$.
  Recalculate parameters of planes in $\mathbf{\Phi}^{(R)}$ using their corresponding inliers (Section Section 2.3.3)
  Recalculate the energy E_{new} of Eq. (11).
  if $E_{new} < E_{old}$ **then**
    Accept the new plane set $\mathbf{\Phi}^{(R)}$: $\mathbf{\Phi} \leftarrow \mathbf{\Phi}^{(R)}$, $CN \leftarrow 0$,
  end if
  else
    Accept the new plane set $\mathbf{\Phi}^{(R)}$ with a small probability $e^{\frac{E_{old}-E_{new}}{T}}$: $\mathbf{\Phi} \leftarrow \mathbf{\Phi}^{(R)}$.
    $CN \leftarrow CN + 1$
  end else
  Update the temperature of simulated annealing $T \leftarrow T * T_\Delta$.
end while

2.3.1. Energy function formulation

Inspired by (Isack and Boykov, 2012), we formulate plane segmentation as an optimal labeling problem with a global energy function balancing geometric errors (i.e., data cost), spatial coherence (smooth cost), and the number of planes (label cost), as Eq. (11).

$$E(\boldsymbol{L}; \mathbf{\Phi}^{(R)}) = \overbrace{\sum_{\boldsymbol{\omega} \in \mathbf{\Psi}} \mathrm{ndist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega})}^{\text{data cost}} + \overbrace{\sum_{\boldsymbol{\omega}, \boldsymbol{\varpi} \in Nei} \delta(l_\omega, l_\varpi)}^{\text{smooth cost}} + \overbrace{\kappa \cdot |\mathbf{\Phi}^{(R)}|}^{\text{label cost}} \tag{11}$$

The data cost term $\sum_{\boldsymbol{\omega} \in \mathbf{\Psi}} \mathrm{ndist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega})$ measures the sum of geometric errors between each basic unit $\boldsymbol{\omega}$ to its corresponding plane $\boldsymbol{P}_{l_\omega}$. More specifically, the geometric error is calculated as the normalized perpendicular distance from $\boldsymbol{\omega}$ to its corresponding plane $\boldsymbol{P}_{l_\omega}$, as Eq. (12).

$$\begin{cases} \mathrm{ndist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega}) = -\ln\left(\frac{1}{\sqrt{2\pi}\Delta d} \cdot \exp\left(-\frac{\mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega})^2}{2\Delta d^2}\right)\right) \\ \mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega}) = \begin{cases} \frac{a_{l_\omega} x_\omega + b_{l_\omega} y_\omega + c_{l_\omega} z_\omega + d_{l_\omega}}{\sqrt{a_{l_\omega}^2 + b_{l_\omega}^2 + c_{l_\omega}^2}} & l_\omega \neq 0 \\ 2\Delta d & l_\omega = 0 \end{cases} \end{cases} \tag{12}$$

where $l_\omega = 0$ is an extra label for basic units not belonging to any plane and Δd is a distance threshold in plane fitting. The distance between any basic unit $\boldsymbol{\omega}$ and $\boldsymbol{P}_0$ (i.e., outlier plane) is a constant value $2\Delta d$, which means that the basic units with distances to their corresponding planes larger than $2\Delta d$ are more likely to be outliers. The shorter the distance $\mathrm{ndist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega})$ is, the less the penalty for assigning the basic unit $\boldsymbol{\omega}$ to the plane $\boldsymbol{P}_{l_\omega}$ becomes.

The smooth cost term $\sum_{\boldsymbol{\omega},\boldsymbol{\varpi}\in Nei}\delta(l_\omega, l_\varpi)$ penalizes the label inconsistency between neighboring basic units (Yan et al., 2014). In this paper, the neighborhood system is based on a 3D Triangulated Irregular Networks (TINs) constructed by all units in Ψ. And every pair of vertices in each triangle are neighboring points. The smoothness penalty between neighbor basic units $\boldsymbol{\omega}$ and $\boldsymbol{\varpi}$ is described by the Potts model (Boykov et al., 2001), as Eq. (13), which means that if a pair of neighboring basic units $\boldsymbol{\omega}$ and $\boldsymbol{\varpi}$ belong to the same plane, the smooth cost between them is 0; otherwise, the smooth cost is 1.

$$\delta(l_\omega, l_\varpi) = \begin{cases} 0 \text{ if } l_\omega = l_\varpi \\ 1 \text{ if } l_\omega \neq l_\varpi \end{cases} \tag{13}$$

The label cost term $\kappa \cdot |\boldsymbol{\Phi}^{(R)}|$ penalizes the number of planes, where κ is the minimum number of points required for a valid plane and $|\boldsymbol{\Phi}^{(R)}|$ is the number of planes in plane set $\boldsymbol{\Phi}^{(R)}$. The label cost term can reduce the number of redundant planes, thus alleviating the over-segmentation issue. More alternative strategies to define data cost, smooth cost and label cost can be found in Yan et al. (2014), Delong et al. (2012) and Isack and Boykov (2012).

2.3.2. Guided sampling

The proposed algorithm generates a series of new planes in each iteration to further enrich the initial plane set, which improves the adequacy of the initial plane set and enhances the capability of the algorithm to extract undetected planes with few points, or small scales (i.e., step 6 of Algorithm 3). We can randomly select several basic units from Ψ, and add their tangent planes into the initial plane set. While the random selection is simple, it is not the most efficient way. The generated plane by random sampling has a high probability of already existing in the initial plane set, or being an invalid plane (e.g., the selected unit locates in the nonplanar surface). In this paper, we propose a guided sampling strategy to improve the probability of hitting an undetected plane. On one hand, large residual $\mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega})$ denotes that $\boldsymbol{\omega}$ is not well fitted by existing initial planes. On the other hand, small curvature f_ω means that the unit $\boldsymbol{\omega}$ has a high probability of locating at a planar surface. Therefore, a basic unit $\boldsymbol{\omega}$ with small curvature and large residual is more likely to be sampled. The sampling probability $\mathrm{pro}(\boldsymbol{\omega})$ of each basic unit $\boldsymbol{\omega}$ is calculated as Eq. (14).

$$\begin{cases} \mathrm{pror}(\boldsymbol{\omega}) = \frac{\mathrm{dist}(\boldsymbol{\omega}, \boldsymbol{P}_{l_\omega})}{\max_{\xi\in\boldsymbol{\Psi}}(\mathrm{dist}(\xi, \boldsymbol{P}_{l_\xi}))} \\ \mathrm{proc}(\boldsymbol{\omega}) = \frac{\max_{\xi\in\boldsymbol{\Psi}}(f_\xi) - f_\omega}{\max_{\xi\in\boldsymbol{\Psi}}(f_\xi) - \min_{\xi\in\boldsymbol{\Psi}}(f_\xi)} \\ \mathrm{pro}(\boldsymbol{\omega}) = \frac{\mathrm{pror}(\boldsymbol{\omega}) + \mathrm{proc}(\boldsymbol{\omega})}{2} \end{cases} \tag{14}$$

where $\mathrm{pror}(\boldsymbol{\omega})$ and $\mathrm{proc}(\boldsymbol{\omega})$ are the sampling probability based on residual and curvature respectively; $\max_{\xi\in\boldsymbol{\Psi}}(f_\xi)$ and $\min_{\xi\in\boldsymbol{\Psi}}(f_\xi)$ is the maximum and minimum curvature; $\max_{\xi\in\boldsymbol{\Psi}}(\mathrm{dist}(\xi, \boldsymbol{P}_{l_\xi}))$ is the maximum residual. Therefore, the proposed algorithm samples PN basic units according to Eq. (14) in each iteration and adds their corresponding tangent planes into the initial plane set.

Fig. 4 shows an illustration of the sampling probability of each unit. Fig. 4a is the extracted initial planes by Algorithm 2, where undetected planes are represented with black, and detected planes are represented with other different colors. As shown in Fig. 4b, the undetected regions (e.g., window sills) and under-segmentation regions (e.g., windows) have large residual, whereas the well-fitted regions (e.g., façade) have smaller residual. Fig. 4c shows that the planar surfaces have small curvature, whereas the intersection areas of multiple planes and nonplanar areas have large curvature. Fig. 4d denotes the sampling probability of each basic unit, where the unit with large residual and small curvature has a high sampling probability.

2.3.3. Energy optimization

After the new planes have been generated, a more adequate initial plane set $\boldsymbol{\Phi}^{(R)}$ is acquired (i.e., step 7 of Algorithm 3). In order to simultaneously refine the new plane set $\boldsymbol{\Phi}^{(R)}$ and verify the validity of the added planes, we first formulate the energy function using $\boldsymbol{\Phi}^{(R)}$ (i.e., step 8 of Algorithm 3), and refine the planes in $\boldsymbol{\Phi}^{(R)}$ by inliers assignment (i.e., step 9 of Algorithm 3) and parameters reestimation (i.e., step 11 of Algorithm 3). Then we verify the validity of the new plane set $\boldsymbol{\Phi}^{(R)}$ by comparing the energy before and after new planes generation (steps 13–19 of Algorithm 3).

More specifically, the inliers of each plane can be achieved by determining an $\boldsymbol{L} = \{l_\omega | \boldsymbol{\omega} \in \boldsymbol{\Psi}_u\}$ (i.e., the labels of all units) which minimizes the energy function Eq. (11). However, minimizing the energy function is NP-hard (Delong et al., 2010). Therefore, the extended ∂-expansion algorithm (Delong et al., 2012) is adopted to achieve the optimal $\boldsymbol{L} = \{l_\omega | \boldsymbol{\omega} \in \boldsymbol{\Psi}_u\}$ by minimizing the energy function Eq. (11) (i.e., inliers assignment). Then we remove the planes without inliers from $\boldsymbol{\Phi}^{(R)}$, and reestimate the parameters of remaining planes using their corresponding inliers according to Eq. (10) (i.e., parameters reestimation).

After a round of inliers assignment and parameters reestimation, we recalculate the energy E_{new} of Eq. (11) (step 12 of Algorithm 3). Then we verify the validity of the new plane set $\boldsymbol{\Phi}^{(R)}$ by comparing the energy before and after new planes generation. Specifically, if the energy after refinement E_{new} is less than the energy before the new planes generation E_{old}, then the algorithm accepts the new plane set $\boldsymbol{\Phi}^{(R)}$; otherwise, the algorithm accepts the new plane set $\boldsymbol{\Phi}^{(R)}$ with a small probability $e^{\frac{E_{old}-E_{new}}{T}}$. Accepting an inferior hypothesis probabilistically can avoid local minima and ensure the algorithm can converge to the global optimum (Pham et al., 2014). After each iteration, we decrease the temperature as $T \leftarrow T * T_\Delta$ (step 20 of Algorithm 3). Repeat the steps 3–20 of Algorithm 3, until the new energy stops decreasing (i.e., $CN > \tau_{CN}$) or the temperature is close to $T_{\min}$. Note that, the energy optimization algorithm does not consider the spatial connectivity of the extracted planes, so each output plane may contain a set of coplanar but unconnected segments. Therefore, the connected component analysis technique (Oehler et al., 2011) is adopted to separate such coplanar planes after global energy optimization.

Fig. 5 illustrates an example of global energy optimization. Fig. 5a and b denote the extracted initial plane set by Algorithm 2, and the sampling probability of each basic unit respectively. Fig. 5c shows the sampled basic units (the tangent planes of the sampled basic units are added into the initial plane set) in the first iteration, where most of the sampled basic units derive from undetected or under-segmented regions. Fig. 5d is the corresponding plane segmentation result of the first iteration, where several new planes located on windows, window sills and roof are detected after a round of new planes generation and energy optimization. Fig. 5e and f represent the final plane segmentation result and the ground truth respectively. As shown in Fig. 5e, most of the inaccurate boundaries, undetected planes, under-segmentation, and over-segmentation issues are addressed through the global energy optimization procedure. Fig. 5g and h shows the final residual of each basic unit before and after global energy optimization. It is worth

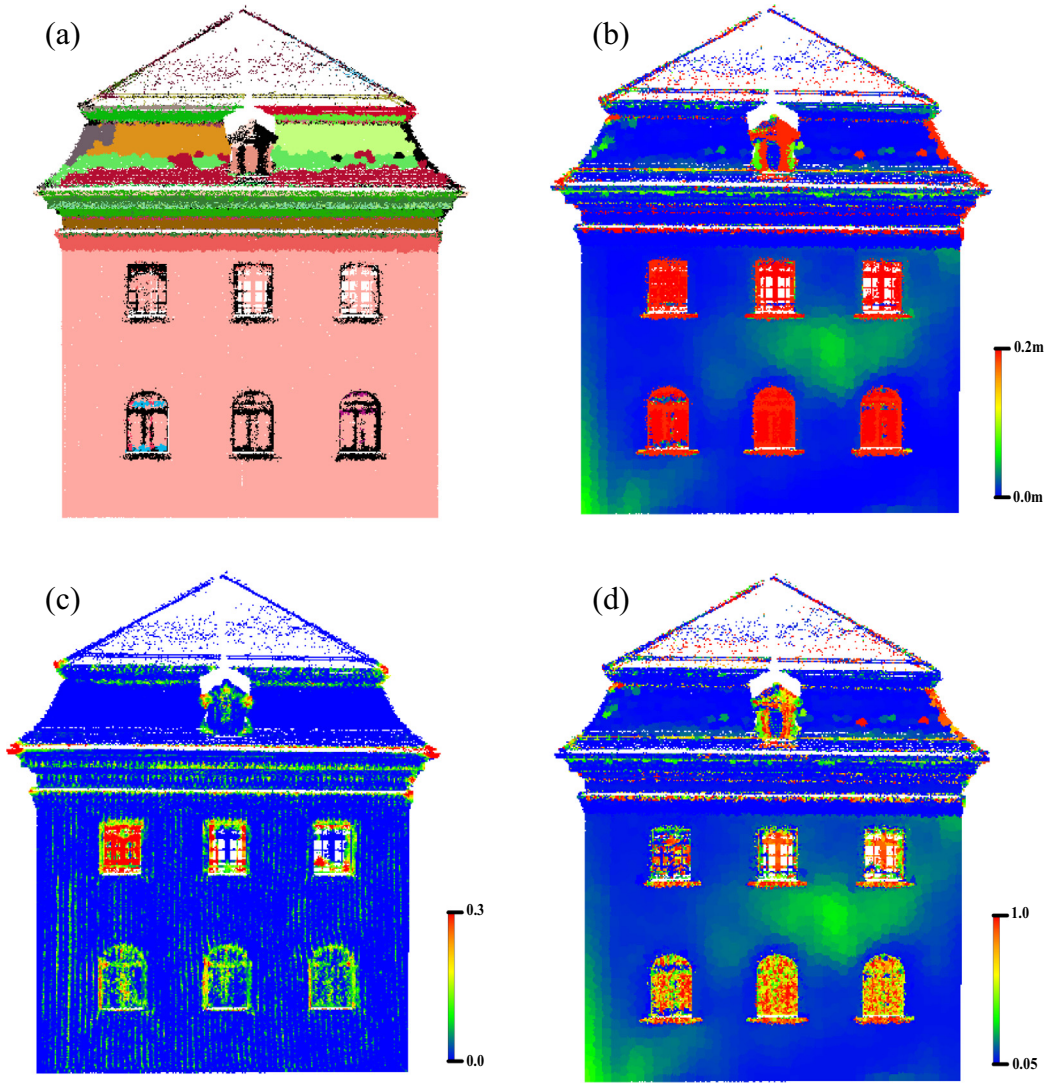

Fig. 4. An illustration of the sampling probability: (a) plane extraction result, (b) the residual of each basic unit to its corresponding plane, (c) the curvature of each basic unit, and (d) the sampling probability of each basic unit.

noting that, most of the basic units are well fitted through global energy optimization with the maximum residual of 0.05 m.

3. Experiments and analysis

3.1. Experimental setup

The implementation details of the experiments, including the description of benchmark datasets, the evaluation criteria, and parameter settings of the proposed methods are described in this section. The experiments were implemented on a computer with 16 GB RAM and an Intel Core i7-6700HQ @ 2.60 GHz CPU.

3.1.1. Datasets description

The performance of the proposed method was evaluated using two benchmark datasets (i.e., SEMANTIC3D.NET, and Stanford large-scale 3D Indoor Spaces Dataset (S3DIS)) with different point densities, point position precisions, and levels of noise and outliers. SEMANTIC3D.NET dataset (Hackel et al., 2017) consists of 30 published terrestrial laser scans with 4 billion 3D points and contains urban and rural scenes, like farms, town halls, sports fields, a castle and market squares. Colorization was performed in a post-processing step by deploying a high-resolution cubemap, which was generated from camera images (Hackel et al., 2017). S3DIS dataset (Armeni et al., 2016) was collected using the Matterport Camera, which combines 3 structured-light sensors to capture 18 RGB and depth images during a 360◦ rotation at each scan location. It consists of 6 large-scale indoor areas with 695,878,620 points and contains various indoor scenes, like conference rooms, offices, copy rooms, hallways, pantries, and lounges. In general, the S3DIS dataset is more challenging than SEMANTIC3D.NET dataset since point cloud collected by structured-light sensors contains a higher level of noise and lower point position precision, all of which pose great challenges for robust 3D plane segmentation. Fig. 6 shows an overview of the sample scenes of the two benchmark datasets.

3.1.2. Evaluation criteria

We evaluate the performance of the proposed method in terms of six metrics (i.e., plane precision, plane recall, under-segmentation rate, over-segmentation rate, boundary precision, and boundary recall), which are commonly used for the evaluation of plane segmentation (Estrada and Jepson, 2009; Awrangjeb et al., 2010; Yan et al., 2014).

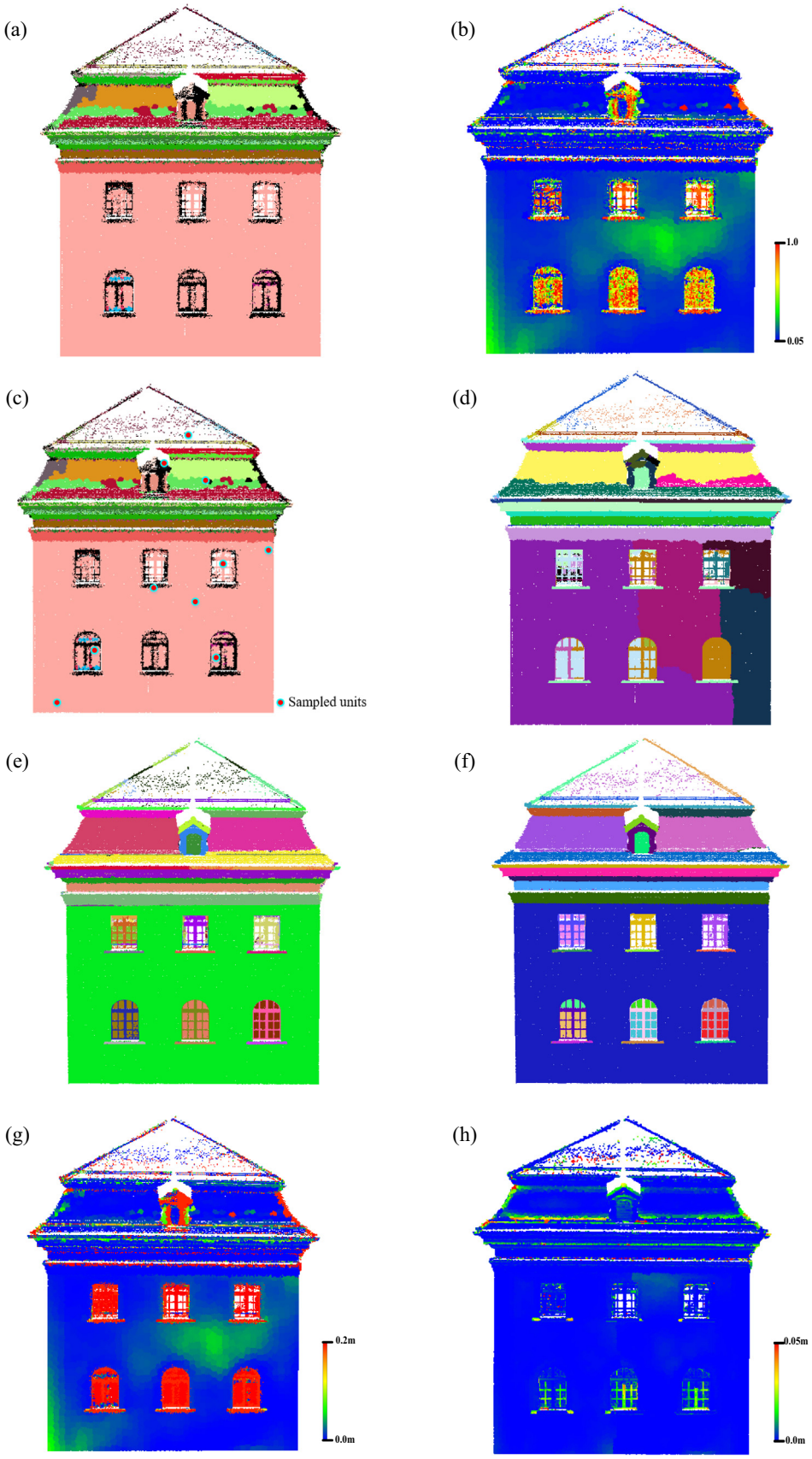

Fig. 5. An illustration of the global energy optimization: (a) the initial plane set, (b) the sampling probability of each basic unit, (c) the sampled basic units in the first iteration, (d) the plane segmentation result of the first iteration, (e) the final output of plane segmentation by global energy optimization, (f) the ground truth, (g) the residual of each basic unit before global energy optimization, and (h) the residual of each basic unit after global energy optimization.

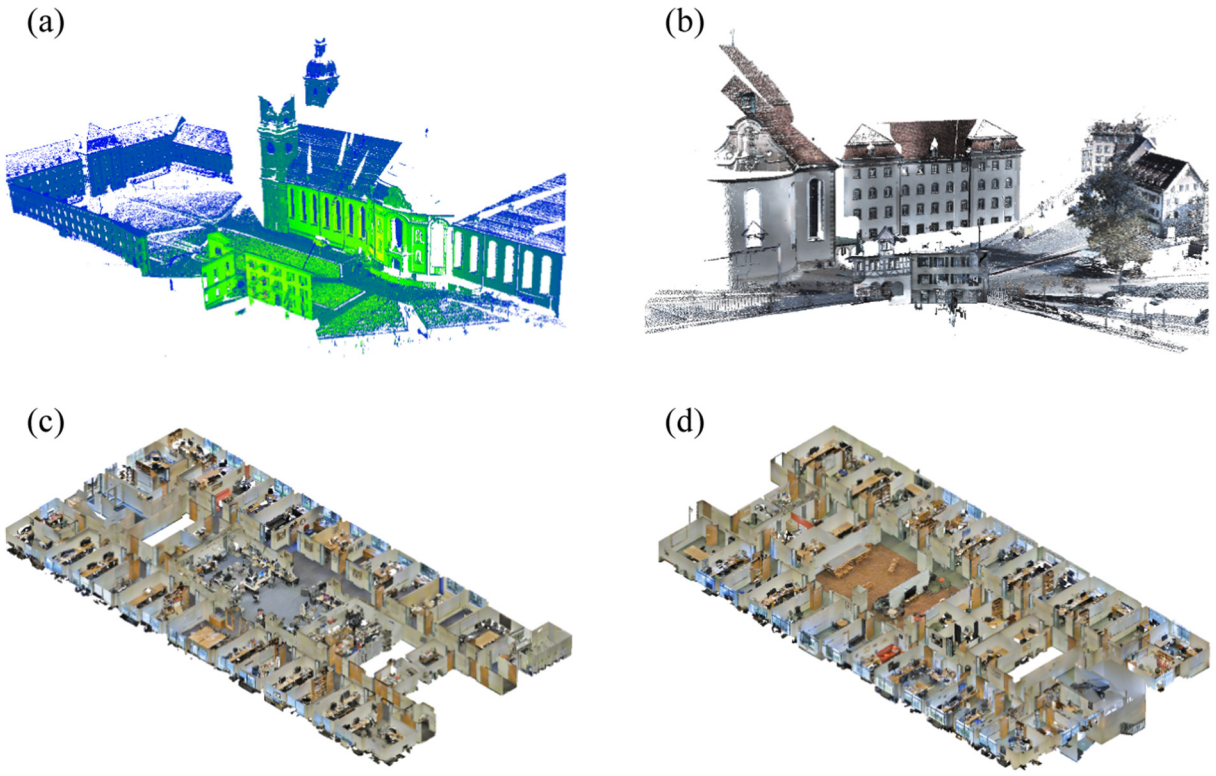

Fig. 6. Overview of sample scenes of the benchmark datasets: (a) Stgallen cathedral station 3 colored by intensity of each point, (b) Stgallen cathedral station 6 colored by RGB values of each point, (c) Area 1 colored by RGB values of each point, and (d) Area 6 colored by RGB values of each point. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Plane precision (PP) and plane recall (PR) are defined as the number of correctly segmented planes with respect to the total number of planes in the segmentation results and ground truth respectively (Yan et al., 2014), as Eqs. (15) and (16).

$$\mathrm{PP} = \frac{N_C}{N_S} \tag{15}$$

$$\mathrm{PR} = \frac{N_C}{N_G} \tag{16}$$

where N_C, N_S, and N_G represent the number of correctly segmented planes, the total number of planes in the segmentation results and the total number of planes in ground truth respectively. To be a correctly segmented plane, a minimum overlap of 80% with the corresponding plane in the ground truth is required.

Under-segmentation rate (USR) is defined as the percentage of detected planes that overlap multiple corresponding planes in the ground truth, and over-segmentation rate (OSR) is defined as the planes in the ground truth that overlap multiple detected planes (Awrangjeb et al., 2010; Yan et al., 2014), as Eqs. (17) and (18).

$$\mathrm{USR} = \frac{N_U}{N_S} \tag{17}$$

$$\mathrm{OSR} = \frac{N_O}{N_G} \tag{18}$$

where N_U is the number of detected planes that overlap more than one planes of the ground truth, and N_O represent the number of planes in the ground truth that overlap multiple detected planes.

Boundary precision (BP) and boundary recall (BR) are defined as the number of correctly detected boundary points with respect to all the detected boundary points and boundary points in ground truth respectively (Estrada and Jepson, 2009; Yan et al., 2014), as Eqs. (19) and (20).

$$\mathrm{BP} = \frac{|B_S \cap B_G|}{|B_S|} \tag{19}$$

$$\mathrm{BR} = \frac{|B_S \cap B_G|}{|B_G|} \tag{20}$$

where B_S and B_G are the set of boundary points in the segmentation result and ground truth, | | denotes the number of points in a dataset.

3.1.3. Parameter settings

Table 1 shows the parameter settings of the proposed 3D plane segmentation method, set by trial and error. All these parameter settings, unless otherwise specified, are used for all the experiments in this paper.

There are five parameters in multiscale supervoxel generation procedure, i.e., w_d, w_n, $r_{\max}$, $r_{\min}$ and r_Δ. w_d and w_n are the weights associated with spatial distance and normal similarity, which control the trade-off between supervoxel compactness and boundary adherence (Papon et al., 2013; Christoph et al., 2014). The parameters w_d = 1 and w_n = 4 are set according to the recommended parameter settings by Christoph et al. (2014). $r_{\max}$, $r_{\min}$ and r_Δ are the maximum, minimum, and decreasing rate of supervoxel resolution, which determine the number of basic units for hybrid region growing and energy optimization. In practice we specified $r_{\min}$, $r_{\max}$ and r_Δ as 5 times average point distance μ, 10 times $r_{\min}$ and $(\frac{r_{\min}}{r_{\max}})^{\frac{1}{10}}$ respectively, where μ is calculated as the average distance from each point to its nearest point.

The hybrid region growing algorithm has three important parameters: the threshold of the angle between two normal

Table 1
Parameters of 3D plane segmentation.

Procedure	Parameter	Descriptor	Value
Multiscale supervoxel generation	w_d	The weight of spatial distance	1
	w_n	The weight of normal deviation	4
	$r_{\min}$	The minimum resolution of supervoxel	5μ
	$r_{\max}$	The maximum resolution of supervoxel	$10r_{\min}$
	r_Δ	The decreasing rate of supervoxel resolution	$\left(\frac{r_{\min}}{r_{\max}}\right)^{\frac{1}{10}}$
Hybrid region growing	τ_θ	The threshold of the angle between two normal vectors	10°
	τ_δ	The threshold of the distance between two planes	0.2 m
	κ	The minimum point number for a valid plane	20
Global energy optimization	T	The initial temperature of simulated annealing	10 °C
	$T_{\min}$	The minimum temperature of decreasing coefficient	0.5 °C
	T_Δ	Decreasing coefficient of simulated annealing	0.99
	τ_{CN}	The maximum number of consecutive rejecting new plane set	5
	PN	The number of increased new planes in each iteration	10
	Δd	A distance threshold of outliers	0.1 m

vectors (i.e., τ_θ), the threshold of the distance between two planes (i.e., τ_δ), and the minimum point number for a valid plane (i.e., κ). τ_θ and τ_δ control the quality of 3D plane segmentation results. In particular, when τ_θ and τ_δ are large, less similar units will be merged and most of the detected planes are under-segmented. When τ_θ and τ_δ are small, only quite similar units will be merged and most of the detected planes are over-segmented. In this paper, we prefer to generate relatively few number of initial planes using hybrid region growing algorithm, and then further enrich and refine the initial plane set in energy optimization procedure. Therefore, we set τ_θ and τ_δ as relatively large values with τ_θ = 10° and τ_δ = 0.2 m to get an under-segmentation initial plane set.

The Global energy optimization procedure contains seven parameters. T, $T_{\min}$, and T_Δ are the initial temperature, minimum temperature and temperature decreasing coefficient of simulated annealing, which ensures convergence to the global optimum regardless of the initial configuration (Lafarge et al., 2010). In this experiment, we set $T = 10$ °C, $T_{\min} = 0.5$ °C and $T_\Delta = 0.99$ according to the recommended parameter settings of (Yang et al., 2013). τ_{CN} is the maximum number of the consecutive rejecting new plane set. Consecutive rejecting new planes means that all planes have been detected and the energy function has converged to the global optimum. It is found in trials that $\tau_{CN} = 5$ can guarantee the convergence of energy function. PN is the number of increased new planes in each iteration, which controls the efficiency of the energy optimization. A crucial component of Algorithm 3 is the generating of new planes (i.e., step 6 of Algorithm 3). The quicker all good planes are generated, the faster Algorithm 3 reaches the global optimal. In particular, when PN is a small value, each iteration costs fewer time but a large number of iterations is needed to detect all good planes. On the contrary, when PN is a large value, Algorithm 3 can detect all good planes with a few number of iterations but each iteration costs more time. So we need to strike a balance between the number of iterations and runtime of each iteration to acquire a minimum total runtime. Experimental results find that the total runtime of energy optimization reduced as PN increases from 1 to 10. And further increasing PN from 10 to 100 increased the total runtime of energy optimization. Therefore, we use PN = 10 as the number of increased new planes in each iteration. Δd and κ are important parameters in global energy optimization procedure, which determine the quality of plane segmentation results. More specifically, Δd is a distance threshold of outliers, which means that the units with distances to their corresponding planes larger than $2\Delta d$ are more likely to be outliers. A large Δd may result in under-segmentation, whereas a small one may produce over-segmentation. κ is the minimum point number for a valid plane. A large κ results in lots of omissive planes, whereas a small one produces some spurious planes. The performance of the proposed method under different settings of parameters Δd and κ is tested using the Recall versus 1-Precision Curve (RP Curve) (Ke and Sukthankar, 2004). Fig. 7a illustrates the RP Curves for Δd ranging from 0.01 m to 0.2 m and Fig. 7b illustrates the RP Curves for κ ranging from 5 to 40. Ideally, the RP Curve would fall in the top-left corner of the plot, which means that the method obtains both high recall and precision. As shown in Fig. 7, the best plane segmentation results were achieved with the value of parameters $\Delta d = 0.1$ m and κ = 20.

3.2. Experiment results

Figs. 8–12 show the outcomes of multiscale supervoxel generation, hybrid region growing, global energy optimization, and the main differences between the plane segmentation result and the ground truth for five selected point clouds (i.e., building 1, building 2, building 3, conference room, and office) from the SEMANTIC3D.NET and S3DIS datasets. Figs. 8–12a represent the three selected point clouds, colored by the RGB values of each point. Figs. 8–12b show the generated multiscale supervoxel results for selected point clouds, where each supervoxel is dotted in one color. It is worth noting that, the number of basic units is notably smaller than the number of raw points, thereby significantly reducing the

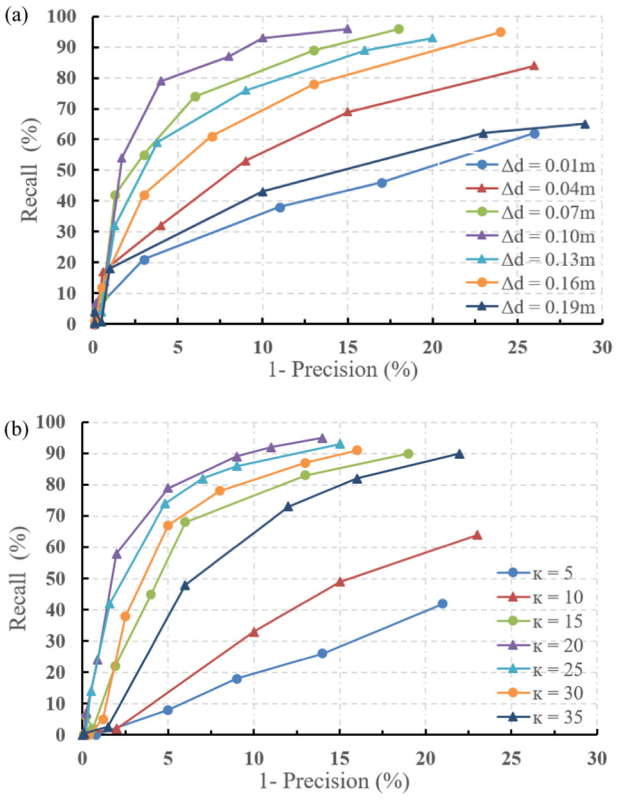

Fig. 7. RP Curves: (a) the RP Curves for varying Δd, and (b) the RP Curves for varying κ.

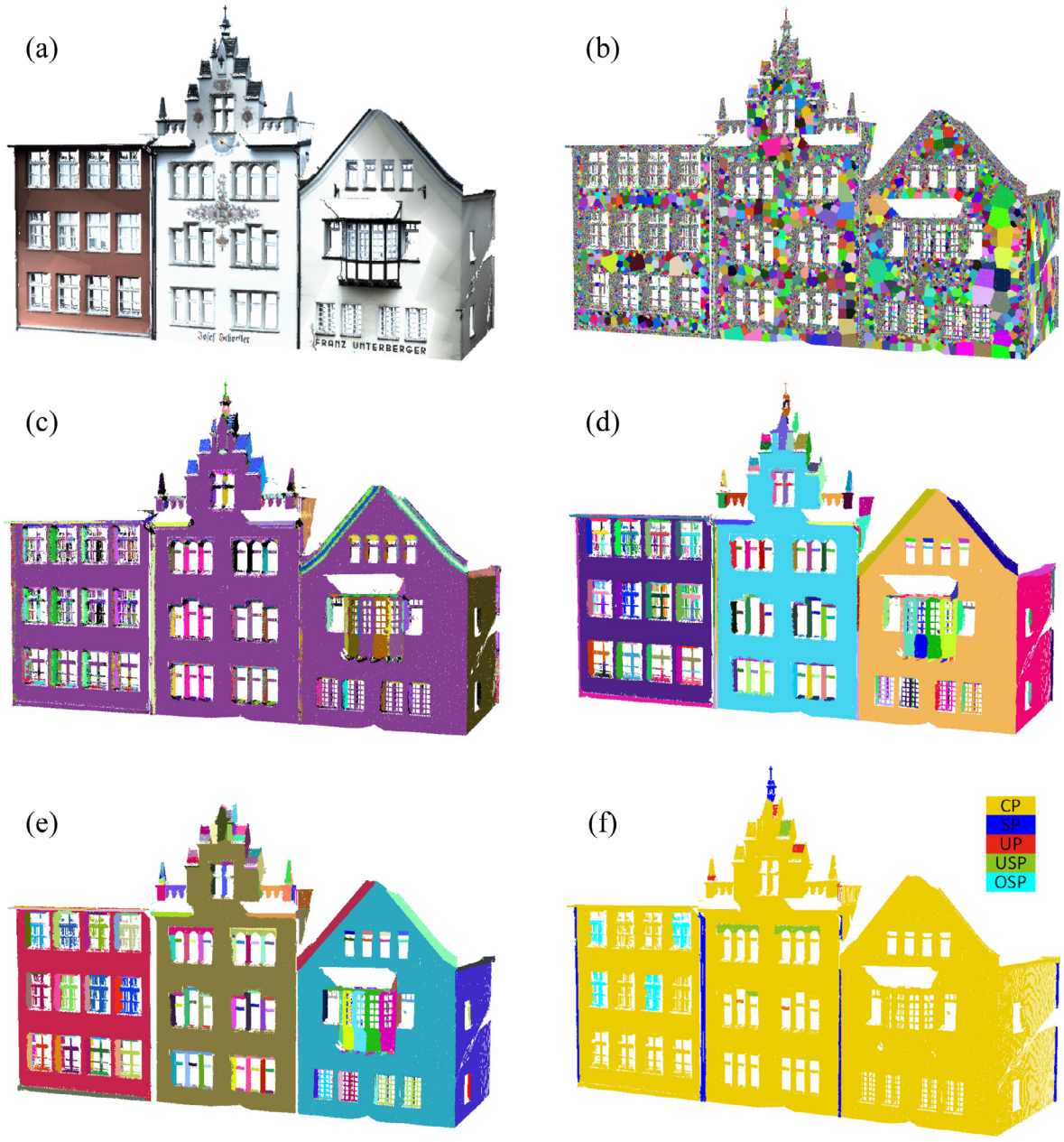

Fig. 8. Plane segmentation result of building 1: (a) the original point cloud, (b) multiscale supervoxel generation result, (c) initial plane segmentation result, (d) plane segmentation result of energy optimization, (e) the ground truth and (f) the main differences between the plane segmentation result and the ground truth.

search space (i.e., the number of points) and accelerating the following hybrid region growing and global energy optimization processes. Figs. 8–12c denote the initial planes segmentation results using the hybrid region growing algorithm, where each plane is drawn in one color. Note that mainly four kinds of errors (i.e., inaccurate boundaries, undetected planes, under-segmentation, and over-segmentation planes) exist in the initial plane set. These issues will be addressed through the following energy optimization procedures. Figs. 8–12d represent the planes segmentation results after global energy optimization, where each plane is drawn in one color. And Figs. 8–12e are the ground truth of each point cloud, which are manually labeled using the CloudCompare point cloud processing software. Figs. 8–12f show the main differences between the plane segmentation results and the ground truth, where the yellow, red, blue, green, and cyan regions represent the correctly segmented planes (CP), undetected plans (UP), spurious planes (SP), under-segmented planes (USP), and over-segmented planes (OSP) respectively. More specifically, the proposed approach fails to detect planes with quite small scale or less than κ points (e.g., the red regions in Figs. 8–12f) due to the insufficient initial plane set or the use of label cost term. Most of the spurious planes (e.g., the blue regions in Figs. 8–12f) are caused by segmenting a nonplanar surface (e.g., cylindrical drainpipe and cambered façade) into a plane or multiple planes. The under-segmented planes (e.g., the green regions in Figs. 8–12f) are caused by merging two or more adjacent planes with exactly similar parameters into one plane due to the insufficient initial plane set or a large distance threshold Δd. The over-segmented planes (e.g., the cyan regions in Figs. 8–12f) are caused by dividing one plane into multiple planes due to a small distance threshold Δd or heavy occlusion (e.g., the window regions). In general, the S3DIS dataset is more challenging than SEMANTIC3D.NET dataset

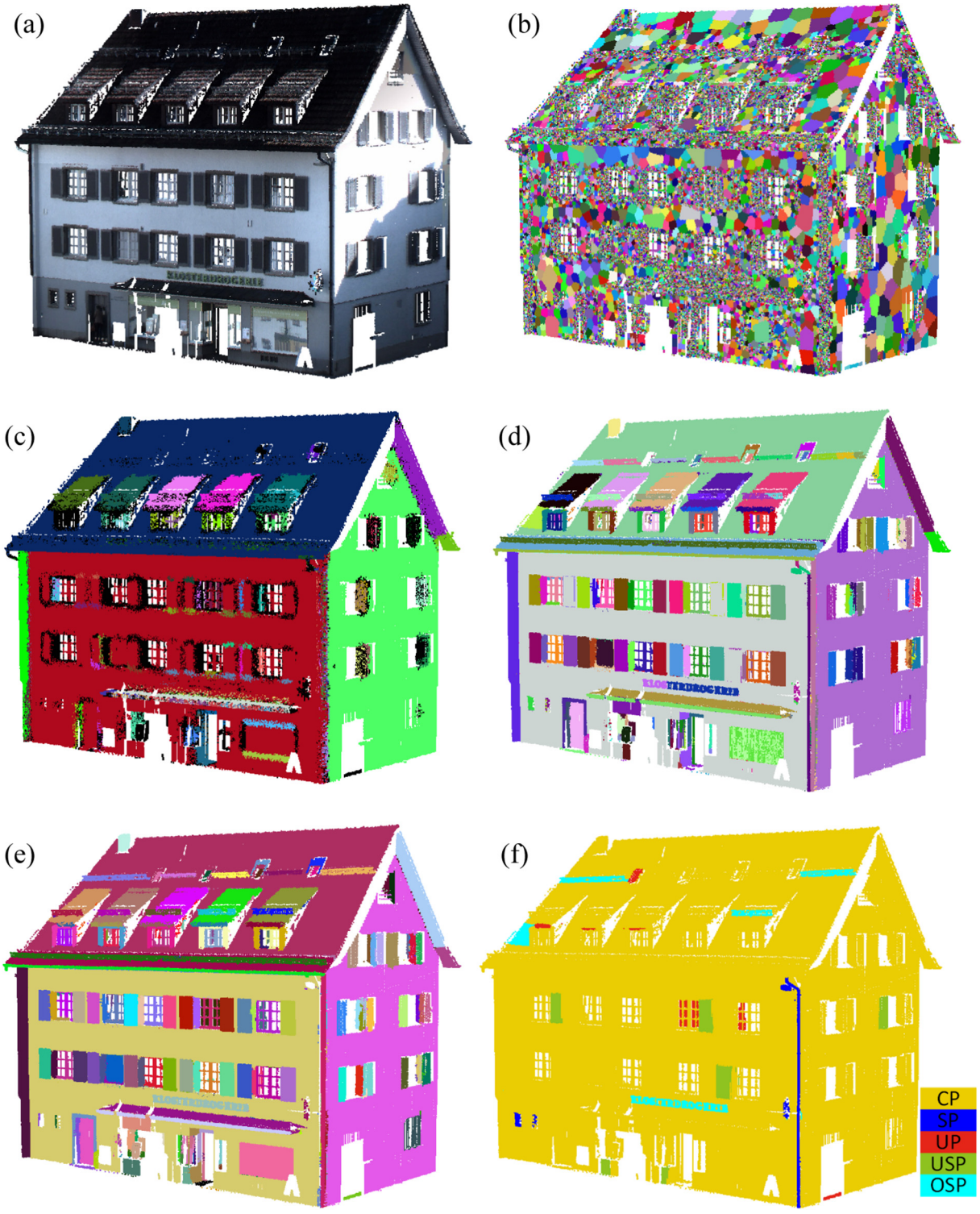

Fig. 9. Plane segmentation result of building 2: (a) the original point cloud, (b) multiscale supervoxel generation result, (c) initial plane segmentation result, (d) plane segmentation result of energy optimization, (e) the ground truth and (f) the main differences between the plane segmentation result and the ground truth.

since the point clouds collected by structured-light sensors contain a higher level of noise and lower point position precision, all of which pose great challenges for robust 3D plane segmentation. The experimental results show that the proposed method can acquire a good performance of 3D plane segmentation both in high-quality TLS point clouds and low-quality RGB-D point clouds.

3.3. Evaluation and analysis

3.3.1. Quantitative evaluation

To evaluate the performance of the proposed method for 3D plane segmentation, the plane segmentation result was compared with those manually marked ground truths in terms of six metrics (i.e., plane precision and plane recall, under-segmentation rate,

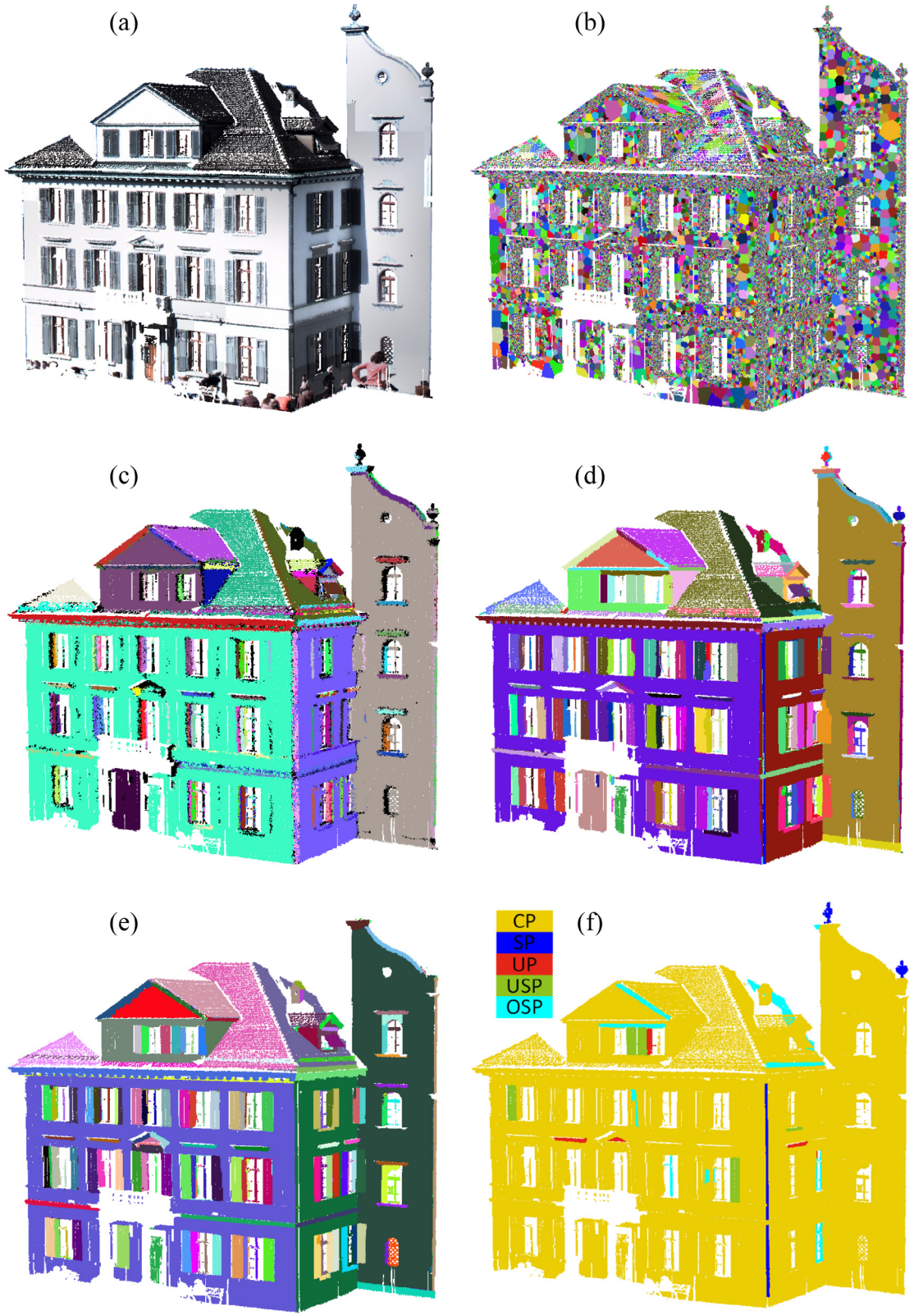

Fig. 10. Plane segmentation result of building 3: (a) the original point cloud, (b) multiscale supervoxel generation result, (c) initial plane segmentation result, (d) plane segmentation result of energy optimization, (e) the ground truth and (f) the main differences between the plane segmentation result and the ground truth.

over-segmentation rate, boundary precision, and boundary recall). To ensure that the evaluation is reliability, we performed 50 runs on each dataset and took the median of these values over 50 runs as the final values. Table 2 reports the number of points and basic units, six metrics and the runtime on the selected point clouds. It is worth note that, the number of basic units is notably smaller than the number of points (i.e., two orders of magnitude smaller), thereby significantly improving the efficiency of the plane

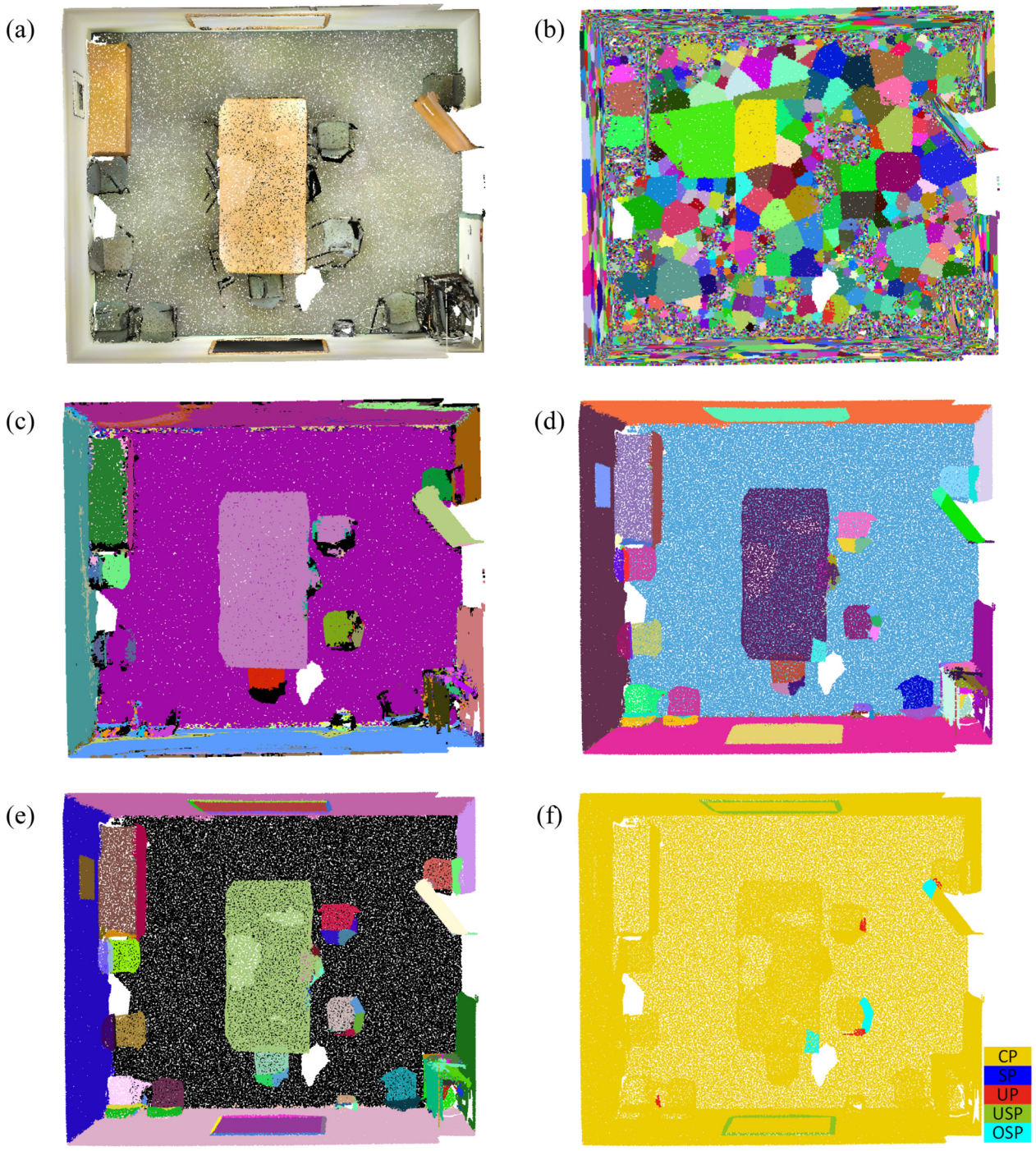

Fig. 11. Plane segmentation result of a conference room: (a) the original point cloud, (b) multiscale supervoxel generation result, (c) initial plane segmentation result, (d) plane segmentation result of energy optimization, (e) the ground truth and (f) the main differences between the plane segmentation result and the ground truth.

segmentation method. The metrics and the runtime listed in Table 2 show that the proposed method can acquire a good performance of 3D plane segmentation both in high-quality TLS point clouds (e.g., SEMANTIC3D: 94.2% (precision), 95.1% (recall), 2.9% (USR), 3.8% (OSR), 93.6% (BP), 94.1% (BR)) and low-quality RGB-D point clouds (e.g., S3DIS: 90.4% (precision), 91.4% (recall), 8.2% (USR), 7.6% (OSR), 90.8% (BP), 91.7% (BR)) with high efficiency.

3.3.2. Performance analysis

To further investigate the effects of each procedure (i.e., multiscale supervoxel generation, hybrid region growing, global energy optimization and new planes generation) on plane segmentation, we also calculate the metrics, iterations and the runtime of REPS-1, REPS-2, REPS-3, and REPS-4, which are the variants of the proposed method. More specifically, REPS-1 does not divide point cloud into supervoxels and directly utilizes original points as basic units for region growing and global energy optimization. REPS-2 only contains multiscale supervoxel generation and hybrid region growing two procedures, so does not refine and enrich the initial plane set. REPS-3 refines the initial plane set without new planes generation process. REPS-4 uses random sampling for new planes generation instead of guided sampling. The corresponding metrics, iterations, and runtime on the selected point clouds are listed in Table 3. It is found that the proposed REPS outperformed its

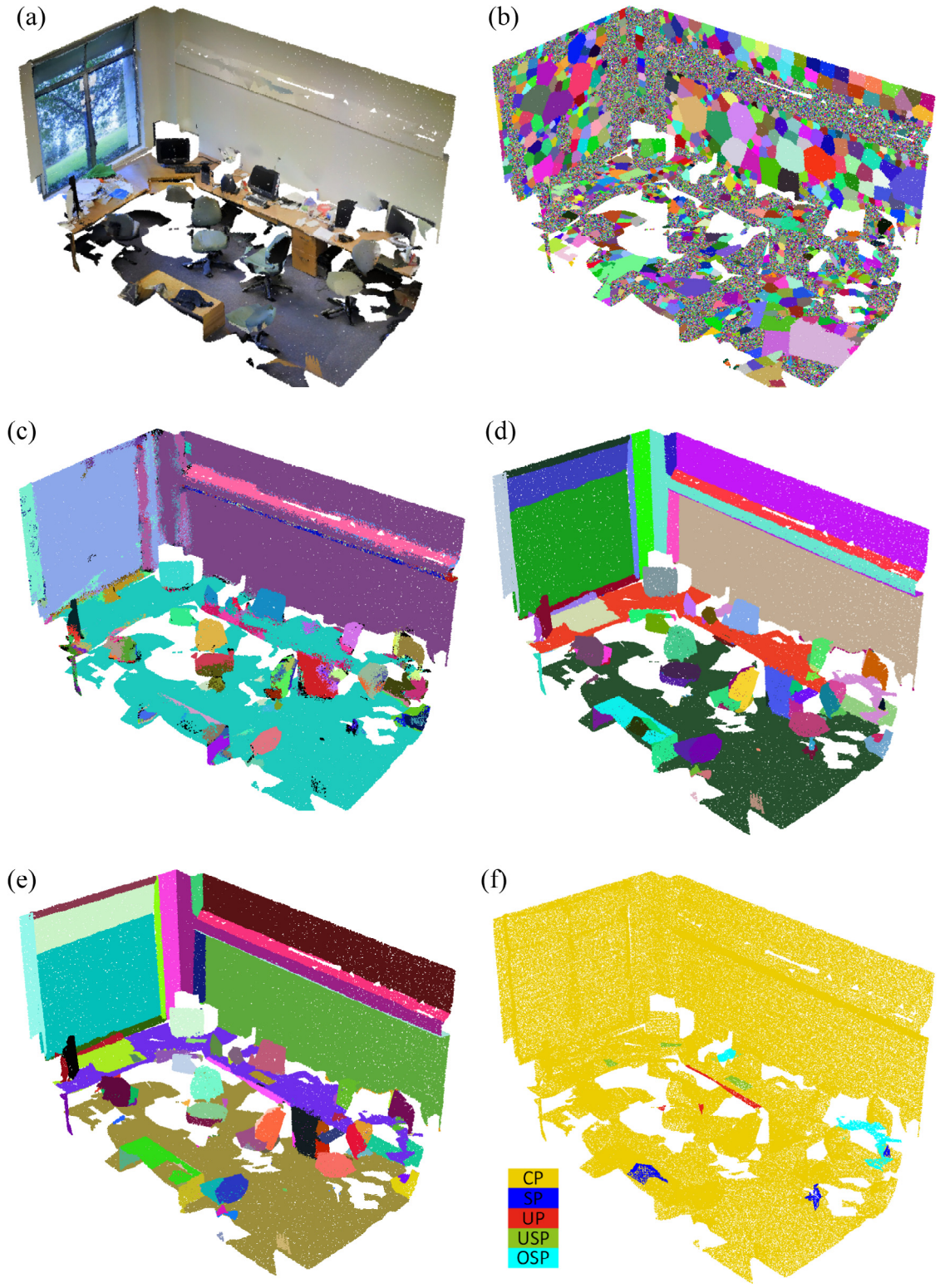

Fig. 12. Plane segmentation result of an office: (a) the original point cloud, (b) multiscale supervoxel generation result, (c) initial plane segmentation result, (d) plane segmentation result of energy optimization, (e) the ground truth and (f) the main differences between the plane segmentation result and the ground truth.

variants on the selected point clouds. First, REPS spent two orders of magnitude fewer runtime and achieved almost the same metrics compared with REPS-1, showing that the hybrid representation of supervoxels and individual points can significantly improve the efficiency of plane segmentation. Second, REPS increased the metrics of plane segmentation from (84.6% (precision), 80.3% (recall), 16.4% (USR), 6.4% (OSR), 86.4% (BP), 80.6% (BR)) and (79.2% (precision), 74.1% (recall), 19.6% (USR), 10.7% (OSR), 79.5% (BP), 75.3% (BR)) to (94.5% (precision), 95.7% (recall), 2.6% (USR), 3.7% (OSR), 94.4% (BP), 92.7% (BR)) and (90.4% (precision), 92.1% (recall), 7.6% (USR), 7.8% (OSR), 90.7% (BP), 91.2% (BR)) on the selected two point clouds compared with REPS-2, which demonstrates that

Table 2
Quantitative evaluation of 3D plane segmentation result.

Data	# Point (million)	# Unit (million)	Precision (%)	Recall (%)	USR (%)	OSR (%)	BP (%)	BR (%)	Runtime (s)			
									A1	A2	A3	Total
Building 1	13.552	0.141	94.3	95.6	2.7	3.8	94.2	92.6	29.5	5.6	200.7	235.8
Building 2	6.942	0.082	93.8	94.5	4.2	3.7	93.1	90.8	18.6	4.1	166.3	189.0
Building 3	8.652	0.093	95.2	94.9	3.1	4.6	93.6	94.3	21.3	4.9	177.7	203.9
SEMANTIC3D	642.8	7.963	94.2	95.1	2.9	3.8	93.6	94.1	Total: 461.5 min			
Conference room	1.067	0.013	90.6	91.3	8.6	7.3	91.1	92.4	3.6	2.3	34.9	40.8
Office	0.825	0.011	90.3	91.8	7.9	8.1	90.6	91.2	3.2	1.9	31.8	36.9
S3DIS	695.9	8.699	90.4	91.4	8.2	7.6	90.8	91.7	Total: 502.3 min			

A1, A2, A3: the abbreviations of Algorithm 1, Algorithm 2, and Algorithm 3 respectively.

Table 3
Performance analysis of the proposed method.

Data		Precision (%)	Recall (%)	USR (%)	OSR (%)	BP (%)	BR (%)	Iterations	Runtime (s)
Building 1	REPS-1	94.5	95.7	2.6	3.7	94.4	92.7	22	40654.5
	REPS-2	84.6	80.3	16.4	6.4	86.4	80.6	\	35.1
	REPS-3	93.3	84.6	9.5	4.6	91.1	83.6	16	180.4
	REPS-4	93.2	94.5	2.9	4.2	93.8	92.1	2346	22421.1
	REPS	94.3	95.6	2.7	3.8	94.2	92.6	21	235.8
Office	REPS-1	90.4	92.1	7.6	7.8	90.7	91.2	15	4612.3
	REPS-2	79.2	74.1	19.6	10.7	79.5	75.3	\	7.1
	REPS-3	86.2	75.6	17.2	9.4	86.9	74.9	12	29.7
	REPS-4	89.5	90.2	8.2	8.6	88.4	89.3	1028	2179.4
	REPS	90.3	91.8	7.9	8.1	90.6	91.2	15	36.9

global energy optimization procedure can dramatically improve the quality of plane segmentation, especially for the low-quality S3DIS dataset. Third, REPS far surpassed REPS-3 in terms of plane recall, under-segmentation rate and boundary recall, which indicates that new plane generation process enhances the capability of the algorithm to extract undetected planes, especially planes with few points, or small scales. Fourth, REPS took two orders of magnitude fewer runtime and iterations than REPS-4, which demonstrates that the proposed method significantly accelerate the convergence speed of global energy optimization by leveraging the guided sampling technique.

3.3.3. Performance comparison

To further compare the performance of the proposed method with other state-of-the-art approaches, we conducted 3D plane segmentation using different selected methods on the point clouds of building 1 and office. Efficient RANSAC (Schnabel et al., 2007) is the most popular and cited 3D Shape detection method and has proven useful for planes, spheres, cylinders, cones and tori detection. Octree-based region growing (ORG) (Vo et al., 2015) is a recently proposed point cloud segmentation method that exhibits superior performance in terrestrial and aerial LiDAR point clouds. Global energy optimization approach (GEO) (Yan et al., 2014) also formulates the plane segmentation issue as a global energy optimization problem, which is the most similar method to our proposed method. Therefore, these state-of-the-art 3D plane segmentation methods are selected as the benchmark methods for performance comparison. The key parameters of all the benchmark methods are set according to the parameter settings recommended in the original articles.

Figs. 13 and 14 show the outcomes of the four benchmark methods on the two selected point clouds. Table 4 lists their corresponding metrics and runtime. It is found that the proposed REPS outperformed other benchmark methods on the selected point clouds. More specifically, a number of observations can be noted based on these comparison results. (1) The local optimal algorithms (i.e., RANSAC and ORG) obtained relatively lower precision, recall, boundary precision, boundary recall, and higher over-segmentation rate and over-segmentation rate than global energy optimization methods (i.e., GEO and REPS), especially for low-quality RGB-D point clouds. The lower performances are attributable to the following two factors. First, the local optimal algorithms are less robust to high levels of noise and clutter. Second, the sub-optimal character of extracting planes one by one tends to result in mistakes at plane transitions, which in turn causes inaccurate and incomplete boundaries among segmented planes. Plane fitting can be better formulated as an optimization problem with a global energy function describing the quality of the overall solution. Therefore, these global energy optimization methods can produce spatially coherent planes and significantly improve the performance of plane segmentation. (2) The proposed REPS method spent two orders of magnitude fewer runtime compared with GEO, which indicates that the hybrid representation of multiscale supervoxels and individual points can significantly reduce the search space and improve the efficiency of plane segmentation. (3) The proposed REPS achieved better performance than the GEO method, especially in terms of plane recall, under-segmentation rate, and boundary recall. This is because the GEO method is powerless to under-segmentation planes and undetected planes since it cannot produce more number of planes than the initial segmentation set. And such improvements of the proposed REPS method are mainly due to the new planes generation procedure, and the mutually reinforcing manner between new planes generation and planes refinement. This synergy enhances the capability of the algorithm to extract undetected planes, especially planes with few points, or small scales, thus improving the completeness of plane segmentation.

3.3.4. Limitations

We also investigate the circumstances in which our method fails to detect the planes. Fig. 15 shows the error cases. The proposed method only can segment planar surfaces, so it has a tendency to segment a nonplanar surface into a plane (e.g., the elliptical regions in Fig. 15a) or multiple planes (e.g., Fig. 15b). Fig. 15a shows the error case of segmentation of a cylindrical drainpipe into a plane. And Fig. 15b shows the error case of

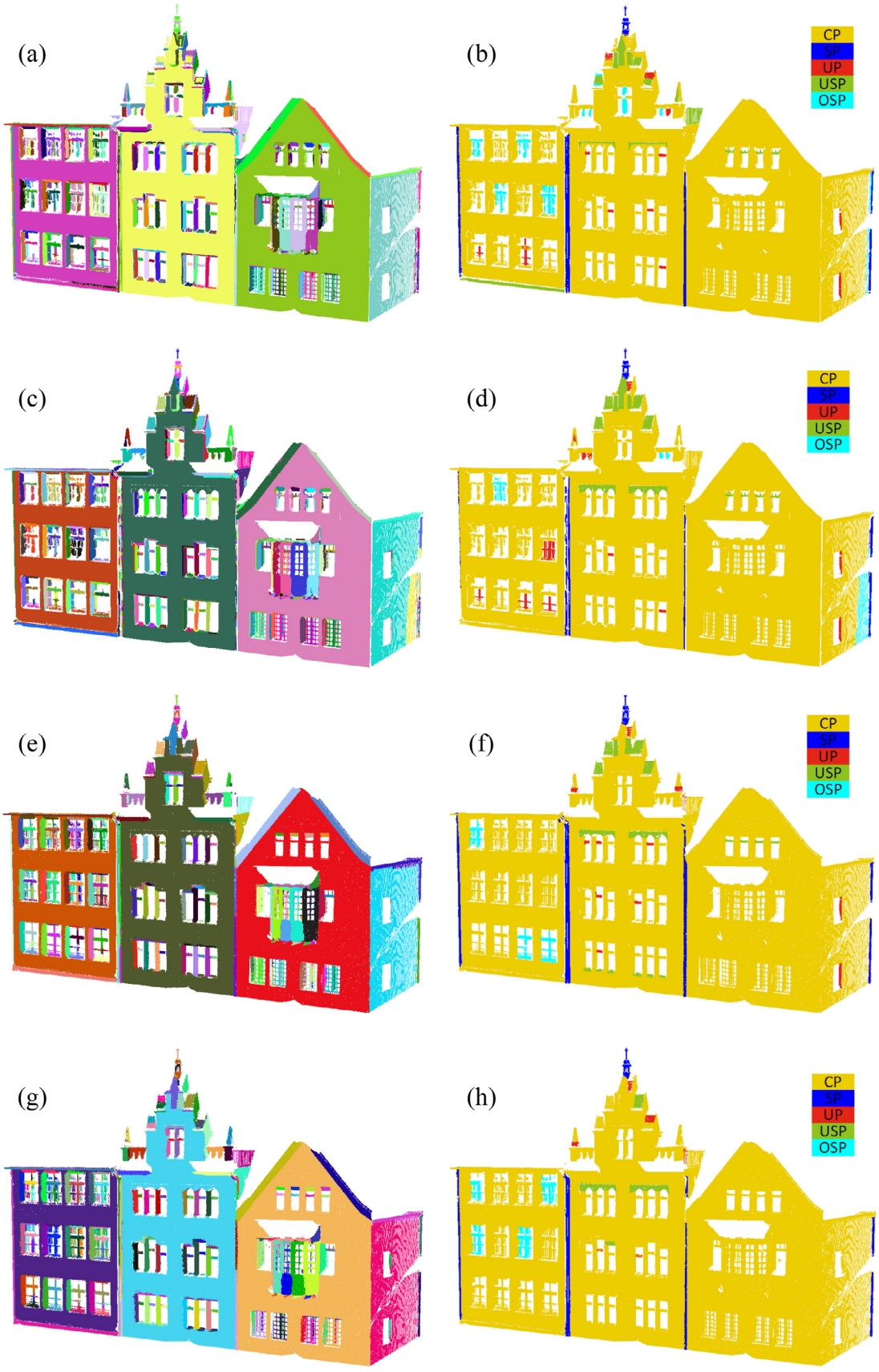

Fig. 13. Plane segmentation result comparison: (a) the output of RANSAC, (b) the main differences between the output of RANSAC and the ground truth, (c) the output of ORG, (d) the main differences between the output of ORG and the ground truth, (e) the output of GEO, (f) the main differences between the output of GEO and the ground truth, (g) the output of REPS, (h) the main differences between the output of REPS and the ground truth.

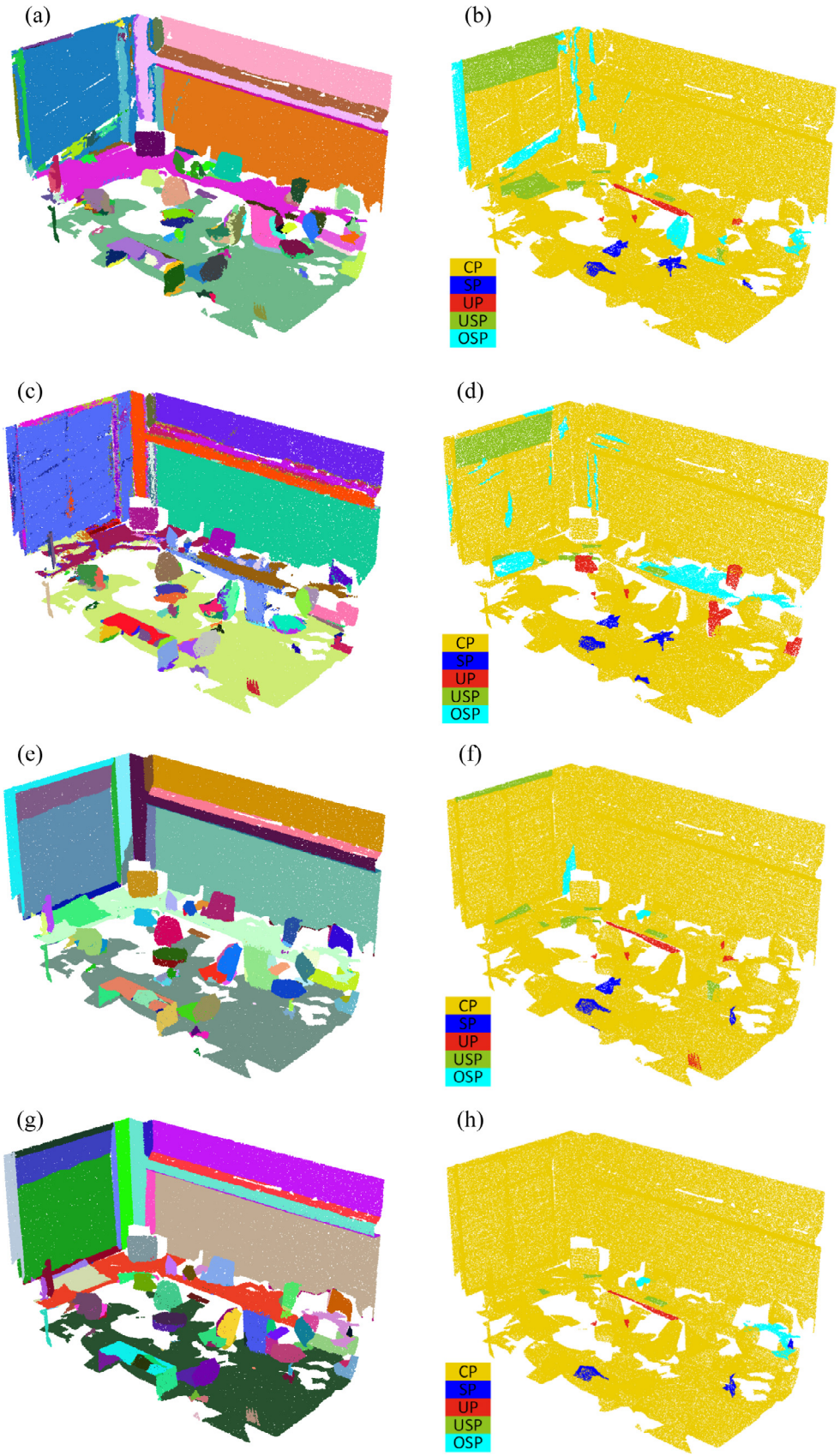

Fig. 14. Plane segmentation result in comparison: (a) the output of RANSAC, (b) the main differences between the output of RANSAC and the ground truth, (c) the output of ORG, (d) the main differences between the output of ORG and the ground truth, (e) the output of GEO, (f) the main differences between the output of GEO and the ground truth, (g) the output of REPS, (h) the main differences between the output of REPS and the ground truth.

Table 4
Performance comparison of 3D plane segmentation results.

Data		Precision (%)	Recall (%)	USR (%)	OSR (%)	BP (%)	BR (%)	Runtime (s)
Building 1	Efficient RANSAC	87.2	87.9	13.9	6.1	82.3	83.1	46.9
	ORG	90.6	90.1	9.3	7.2	84.7	82.1	76.4
	GEO	93.2	90.5	8.5	3.4	91.3	86.7	39792.2
	REPS	94.3	95.6	2.7	3.9	94.2	92.6	235.8
Office	Efficient RANSAC	76.5	75.6	12.8	16.6	74.2	73.1	14.6
	ORG	73.3	74.2	11.4	21.8	75.1	71.6	22.8
	GEO	88.8	84.4	11.2	8.8	87.9	84.8	4304.6
	REPS	90.3	91.8	6.9	5.4	90.6	91.2	36.9

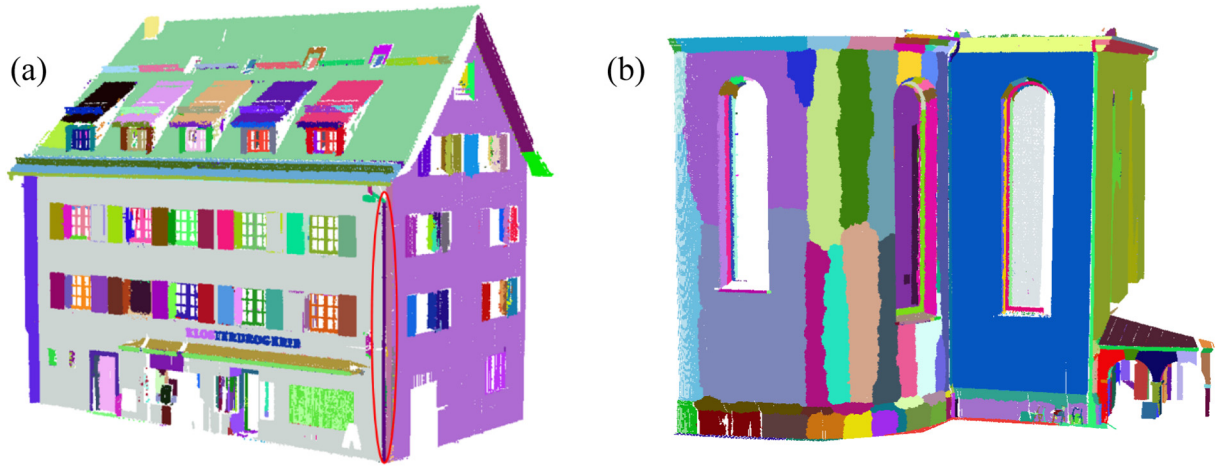

Fig. 15. Error cases of proposed method: (a) segmentation of a cylindrical drainpipe into a plane, and (b) segmentation of a cambered façade into multiple planes.

segmentation of a cambered façade into multiple planes. To overcome these limitations, we will further extend the method to simultaneously detect multiple geometric primitives (e.g., plane, sphere, cylinder, and cone) from point clouds.

4. Conclusions

This paper presented a novel global energy optimization based 3D plane segmentation method and evaluated its performances on two popular benchmark datasets. Comprehensive experiments demonstrated that the proposed method obtained good performances both in high-quality TLS point clouds and low-quality RGB-D point clouds, and outperformed the state-of-the-art approaches. However, the proposed solution also has limitations. The proposed method can only segment planar surfaces, so it has a tendency to segment a small-scale nonplanar surface into a plane and segment a large-scale nonplanar surface into multiple planes. In future work, we will further extend the method to simultaneously detect multiple geometric primitives (e.g., plane, sphere, cylinder, and cone), and reconstruct compact and detailed 3D models using the extracted geometric primitives.

Acknowledgements

The authors would like to acknowledge the following institutions: ETH Zurich for providing the SEMANTIC3D.NET dataset; Computer Science Department of Stanford University for providing the S3DIS dataset. This research is jointly supported by the National Science Fund for Distinguished Young Scholars (No. 41725005), NSFC projects (No. 41531177, No. 41071268), National Key Technology Support Program (No. 2014BAL05B07) and National Key Research and Development Program of China (2016YFF0103500).

References

Armeni, I., Sener, O., Zamir, A.R., et al., 2016. 3d semantic parsing of large-scale indoor spaces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, 1534–1543.
Awrangjeb, M., Ravanbakhsh, M., Fraser, C.S., 2010. Automatic detection of residential buildings using LIDAR data and multispectral Imagery. ISPRS J. Photogramm. Rem. Sens. 65 (5), 457–467.
Ballard, D.H., 1981. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn. 13 (2), : 111–122.
Biosca, J.M., Lerma, J.L., 2008. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS J. Photogramm. Rem. Sens. 63 (1), 84–98.
Boulaassal, H., Landes, T., Grussenmeyer, P., et al., 2007. Automatic segmentation of building facades using terrestrial laser data. Laser 07 (2007), 65.
Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23 (11), 1222–1239.
Chen, D., Zhang, L., Mathiopoulos, P.T., et al., 2014. A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 7 (10), 4199–4217.
Christoph, Stein.S., Schoeler, M., Papon, J., et al., 2014. Object partitioning using local convexity. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, 304–311.
Delong, A., Osokin, A., Isack, H.N., et al., 2010. Fast approximate energy minimization with label costs. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, pp. 2173–2180.
Delong, A., Osokin, A., Isack, H.N., Boykov, Y., 2012. Fast approximate energy minimization with label costs. Int. J. Comput. Vis. 1 (96), 1–27.
Deschaud, J.E., Goulette, F., 2010. A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In: Proceedings of 3D Processing, Visualization and Transmission Conference (3DPVT2010).
Dold, C., Brenner, C., 2006. Registration of terrestrial laser scanning data using planar patches and image data. Int. Arch. Photogramm., Rem. Sens. Spatial Inform. Sci. 36 (5), 78–83.
Estrada, F.J., Jepson, A.D., 2009. Benchmarking image segmentation algorithms. Int. J. Comput. Vis. 85 (2), 167–181.
Filin, S., 2002. Surface clustering from airborne laser scanning data. Int. Arch. Photogramm. Rem. Sens. Spatial Inform. Sci. 34 (3/a), 119–124.
Filin, S., Pfeifer, N., 2006. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. ISPRS J. Photogramm. Rem. Sens. 60 (2), 71–80.
Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

Hackel, T., Savinov, N., Ladicky, L., et al., 2017. Semantic3D. net: A new Large-scale Point Cloud Classification Benchmark. arXiv preprint arXiv:1704.03847.
Isack, H., Boykov, Y., 2012. Energy-based geometric multi-model fitting. Int. J. Comput. Vis. 97 (2), 123–147.
Jung, J., Hong, S., Jeong, S., et al., 2014. Productive modeling for development of as-built BIM of existing indoor structures. Autom. Constr. 42, 68–77.
Kaushik, R., Xiao, J., 2012. Accelerated patch-based planar clustering of noisy range images in indoor environments for robot mapping. Rob. Auton. Syst. 60 (4), 584–598.
Ke, Y., Sukthankar, R., 2004. PCA-SIFT: a more distinctive representation for local image descriptors. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2. IEEE. II–II.
Kim, C., Habib, A., Pyeon, M., et al., 2016. Segmentation of planar surfaces from laser scanning data using the magnitude of normal position vector for adaptive neighborhoods. Sensors 16 (2), 140.
Kim, K.H., Shan, J., 2011. Building roof modeling from airborne laser scanning data based on level set approach. ISPRS J. Photogramm. Rem. Sens. 66 (4), 484–497.
Lafarge, F., Gimel'farb, G., Descombes, X., 2010. Geometric feature extraction by a multimarked point process. IEEE Trans. Pattern Anal. Mach. Intell. 32 (9), 1597–1609.
Lalonde, J.F., Vandapel, N., Huber, D.F., Hebert, M., 2006. Natural terrain classification using three-dimensional ladar data for ground robot mobility. J. Field Rob. 23 (10), 839–861.
Lenac, K., Kitanov, A., Cupec, R., et al., 2017. Fast planar surface 3D SLAM using LIDAR. Rob. Auton. Syst.
Lin, Y., Wang, C., Chen, B., et al., 2017. Facet segmentation-based line segment extraction for large-scale point clouds. IEEE Trans. Geosci. Rem. Sens. https://doi.org/10.1109/TGRS.2016.2639025.
Nurunnabi, A., Belton, D., West, G., 2012. Robust segmentation in laser scanning 3D point cloud data. In: Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on. IEEE, pp. 1–8.
Oehler, B., Stueckler, J., Welle, J., et al., 2011. Efficient multi-resolution plane segmentation of 3D point clouds. Intell. Robot. Appl. 2011, 145–156.
Oesau, S., Lafarge, F., Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS J. Photogramm. Rem. Sens. 2014 (90), 68–82.
Papon, J., Abramov, A., Schoeler, M., et al., 2013. Voxel cloud connectivity segmentation-supervoxels for point clouds. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013, 2027–2034.
Pham, T.T., Chin, T.J., Yu, J., et al., 2014. The random cluster model for robust geometric fitting. IEEE Trans. Pattern Anal. Mach. Intell. 36 (8), 1658–1671.
Pham, T.T., Eich, M., Reid, I., et al., 2016. Geometrically consistent plane extraction for dense indoor 3D maps segmentation. In: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, pp. 4199–4204.
Sampath, A., Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Trans. Geosci. Rem. Sens. 48 (3), 1554–1567.
Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point cloud shape detection. Computer Graphics Forum. Blackwell Publishing Ltd 26(2), 214–226.
Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. Proceedings of the ISPRS Workshop on Laser Scanning 2007 (36), 407–412.
Teboul, O., Simon, L., Koutsourakis, P., et al., 2010. Segmentation of building facades using procedural shape priors. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, pp. 3105–3112.
Tennakoon, R.B., Bab-Hadiashar, A., Cao, Z., et al., 2016. Robust model fitting using higher than minimal subset sampling. IEEE Trans. Pattern Anal. Mach. Intell. 38 (2), 350–362.
Theiler, P.W., Schindler, K., 2012. Automatic registration of terrestrial laser scanner point clouds using natural planar surfaces. ISPRS Ann. Photogramm., Rem. Sens. Spatial Inform. Sci. 3, 173–178.
Tóvári, D., Pfeifer, N., 2005. Segmentation based robust interpolation – a new approach to laser data filtering. Int. Arch. Photogramm., Rem. Sens. Spatial Inform. Sci. 36 (3/19), 79–84.
Vaskevicius, N., Birk, A., Pathak, K., et al., 2010. Efficient representation in three-dimensional environment modeling for planetary robotic exploration. Adv. Rob. 24 (8–9), 1169–1197.
Vo, A.V., Truong-Hong, L., Laefer, D.F., et al., 2015. Octree-based region growing for point cloud segmentation. ISPRS J. Photogramm. Rem. Sens. 104, 88–100.
Vosselman, G., Gorte, B.G.H., Sithole, G., et al., 2004. Recognizing structure in laser scanner point clouds. Int. Arch. Photogramm., Rem. Sens. Spatial Inform. Sci. 46 (8), 33–38.
Wang, L., Shen, C., Duan, F., et al., 2016. Energy-based multi-plane detection from 3D point clouds. In: International Conference on Neural Information Processing. Springer International Publishing, pp. 715–722.
Xiao, J., Adler, B., Zhang, J., et al., 2013a. Planar segment based three-dimensional point cloud registration in outdoor environments. J. Field Rob. 30 (4), 552–582.
Xiao, J., Zhang, J., Adler, B., et al., 2013b. Three-dimensional point cloud plane segmentation in both structured and unstructured environments. Rob. Auton. Syst. 61 (12), 1641–1652.
Xu, B., Jiang, W., Shan, J., et al., 2015. Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds. Rem. Sens. 8 (1), 5.
Xu, Y., Tuttas, S., Hoegner, L., et al., 2017. Geometric primitive extraction from point clouds of construction sites using VGS. IEEE Geosci. Rem. Sens. Lett. 14 (3), 424–428.
Yan, J., Shan, J., Jiang, W., 2014. A global optimization approach to roof segmentation from airborne lidar point clouds. ISPRS J. Photogramm. Rem. Sens. 2014 (94), 183–193.
Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. ISPRS J. Photogramm. Rem. Sens. 2013 (81), 19–30.
Yang, B., Dong, Z., Zhao, G., Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS J. Photogramm. Rem. Sens. 99, 45–57.
Yang, B., Dong, Z., Liang, F., et al., 2016. Automatic registration of large-scale urban scene point clouds based on semantic feature points. ISPRS J. Photogramm. Rem. Sens. 2016 (113), 43–58.
Yang, B., Wang, J., 2016. Mobile mapping with ubiquitous point clouds. Geospatial Inform. Sci. 19 (3), 169–170.
Yang, B., Xu, W., Dong, Z., 2013. Automated extraction of building outlines from airborne laser scanning point clouds. IEEE Geosci. Rem. Sens. Lett. 10 (6), 1399–1403.
Yu, J., Chin, T.J., Suter, D., 2011. A global optimization approach to robust multi-model fitting. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 2041–2048.
Zhou, G., Cao, S., Zhou, J., 2016. Planar segmentation using range images from terrestrial laser scanning. IEEE Geosci. Rem. Sens. Lett. 13 (2), 257–261.