Fast Regularity-Constrained Plane Reconstruction

Yangbin Lin\textsuperscript{1}, Jialian Li\textsuperscript{1}, Cheng Wang\textsuperscript{2}, Zhonggui Chen\textsuperscript{2}, Zongyue Wang\textsuperscript{1}, and Jonathan Li\textsuperscript{2}

\textsuperscript{1}Computer Engineering College, Jimei University
\textsuperscript{2}Xiamen University

Abstract

Man-made environments typically comprise planar structures that exhibit numerous geometric relationships, such as parallelism, coplanarity, and orthogonality. Making full use of these relationships can considerably improve the robustness of algorithmic plane reconstruction of complex scenes. This research leverages a constraint model requiring minimal prior knowledge to implicitly establish relationships among planes. We introduce a method based on energy minimization to reconstruct the planes consistent with our constraint model. The proposed algorithm is efficient, easily to understand, and simple to implement. The experimental results show that our algorithm successfully reconstructs planes under high percentages of noise and outliers. This is superior to other state-of-the-art regularity-constrained plane reconstruction methods in terms of speed and robustness.

1. Introduction

With the proliferation of three-dimensional (3D) scanning devices, it is easy to acquire large volumes of 3D point clouds. Point clouds provide a direct and convenient way to describe outdoor and indoor scenes. However, the method is imposing when modeling such massive data. Automatic modeling methods typically produce excessively complex meshes, which are cumbersome for later applications. However, many applications (e.g., simultaneous localization and mapping and level-of-detail generation) do not require overly fine models. Considering that man-made environments contain many planar structures, plane reconstruction remains a suitable choice for 3D-scene description. Furthermore, plane reconstruction or extraction is typically used as a prior step in various tasks, such as architecture modeling [2], place recognition [15], object recognition [30], and registration [42].

In man-made environments, planar structures generally conform to one of the following relationships: parallelism, orthogonality, coplanarity, and angular equality. Making full use of these relationships can considerably improve the robustness of the plane reconstruction algorithms for complex scenes [27]. However, traditional plane extraction techniques (e.g., region growing, Hough transform [40, 4], and random-sample consensus (RANSAC) [16, 34]) do not take advantage of these geometric constraints and suffer from a lack of robustness under high percentages of noise and outliers.

Next, we refer to the task of reconstructing planes from 3D unstructured point clouds while maintaining a geometric relationship between reconstructed planes as regularity-constrained plane reconstruction (RCPR), which is increasingly used in industrial manufacturing and architectural modeling.

A trivial method of RCPR is the detection-then-regularization strategy, which has two steps. In the first step, the method extracts planes without regard to regularity constraints. Regularity is reconstructed in the second step. This method relies heavily on the quality of the extracted planes in the first step and is easily renders local solutions. Sophisticated algorithms, on the other hand, are traditionally time-consuming, hampering practicality.

In this paper, we formulate RCPR as a global $L_0$ gradient minimization problem, proposing a simple but efficient algorithm to solve it. The proposed algorithm solves the problems of simultaneous plane reconstruction and regularization without losing efficiency. In fact, our algorithm is 3-1,000 times faster than extant RCPR algorithms.

2. Related work

In this section, we review works immediately related to our problem. We cover the following three main aspects: plane extraction, multimodel fitting, and regularity-
constrained fitting.

**Plane extraction** Plane extraction is an old but recurrent problem. It requires many different techniques and has many variants, depending on the tasks and input data. A very common technique is region growing, often employed by more sophisticated algorithms to extract initial planes [10, 27, 29, 22]. Another popular approach is RANSAC [16] and variants (e.g. [34, 33, 18, 43]). RANSAC is a robust data estimator when facing a high percentage of outliers. However, it was originally designed to cope with a single model and was not built to maintain regularity. The randomized Hough transform technique [40] naturally handles multiple structures. However, it cannot cope with a high percentage of outliers [24].

Recent planar-structure extraction techniques (e.g., supervoxel segmentation [31, 23] and structural-scale planar-shape detection [14]) constrain the scale of planar structures but do not constrain their geometric relationships.

In summary, traditional plane extraction methods do not aim to maintain regularity. However, they are often the initial step in the RCPR methods.

**Multimodel fitting** The goal of multimodel fitting is to improve the robustness when fitting multiple models to data contaminated by noise and outliers. Multimodel fitting is typically formulated as an optimization problem. For example, Delong et al. [12] and Amayo et al. [1] converted the problem to a multilabeling problem and solved it via α-expansion [5] and a primal-dual algorithm [9], respectively. Magri and Fusiello [25] cast the multimodel fitting problem as a set-coverage problem and solved it with integer linear programming. Moreover, clustering-based methods (e.g. J-linkage [37] and T-linkage [24]) and hypergraph-based methods [39] have gained the popularity.

In our application, because of the huge input point cloud, we favor methods having low time complexity. Unfortunately, most multimodel fitting methods do not meet this requirement.

**Regularity-constrained fitting** Regularity-constrained fitting has gained popularity in recent years. Li et al. [21] presented the GlobFit framework, which adopted an iterative detection-then-regularization strategy to fit primitives. It first used RANSAC to locally fit primitives and to detect the geometric relationships among the primitives. Then, it refitted these primitives using the detected constraints. Similarly, Zhou, and Neumann [41] adopted the same strategy to model buildings. Departing from GlobFit, Oesau et al. [29] sought to progressively detect regularities rather than iterating between complete detection and regularization. Monszpart et al. [27] focused on constraining the angles between pairs of planes. Their method first generated candidate planes with fitting distances. Then, it searched the optimal planes from candidates using mixed-integer linear programming. Although the method achieved good performance with cluttered scenes, it was time-consuming because of the need to solve large-scale mixed-integer programming problems. More recently, Joo et al. [17] presented a fast branch-and-bound framework to estimate the optimal Manhattan frame.

### 3. Motivation and constraint models

RCPR’s regularity depends on the assumptions made of the real world. Here, we refer to this group of assumptions as a constraint model. Moreover, we define the constraint model as a collection of plane sets satisfying a particular constraint. Thus, the task of RCPR can be regarded as finding an element from the constraint model that best fits the given data. A typical constraint model is the Manhattan model [11].

**Manhattan model** A plane set, \( \mathcal{P} \), is a Manhattan model if and only if it satisfies

\[
\angle v_i, v_j \in \{0^\circ, 90^\circ\}, \forall v_i, v_j \in V, \quad (1)
\]

where \( V \) is the set of unit normal vectors of \( \mathcal{P} \). Because a plane has two orientations, we specify that the angle between the normal vector in \( V \) and positive z-axis is no more than 90°.

In practice, the Manhattan model is a very strong constraint. It can simplify a few tough problems, such as line clustering and vanishing-point estimation in two-dimensional (2D) images [3]. However, the real world usually does not conform to the Manhattan model. Therefore, Straub et al. [36] proposed a mixture of Manhattan frames. The idea was to utilize multiple Manhattan frames (MMF) to represent complex scenes.

**Multiple Manhattan frames model** A plane set \( \mathcal{P} \), is an \( n \)-MMF model if it can be divided into \( n \) non-overlapping subsets, with each subset being a Manhattan model.

The limitation of Manhattan and MMF models is obvious. For example, they only consider parallelism and orthogonality. RAPTER [27] adopted a constraint model that considered more angular constraints. Because RAPTER’s constraint model is difficult to formulate, we only consider its variant, which we call the generalized Manhattan model.

**Generalized Manhattan model** A plane set, \( \mathcal{P} \), is a generalized Manhattan model, if and only if it satisfies

\[
\angle v_i, v_j \in \mathcal{S}, \forall v_i, v_j \in V, \quad (2)
\]
where $S$ is a user-defined set of constraint angles. It is easy to note that, if $S = \{0^\circ, 90^\circ\}$, the generalized Manhattan model is equivalent to the Manhattan model.

A comparison of these constraint models on a line-fitting task is illustrated in Fig. 1. We notice that the Manhattan model cannot fit this simple data well. The MMF obtains a better result but still misses one line. The generalized Manhattan model can fit the data well. However, it should introduce more angle constraints. For example, in Fig. 1, the set of constraint angles should be set to $\{0^\circ, 25.565^\circ, 63.435^\circ, 90^\circ, 116.565^\circ, 153.435^\circ\}$. Usually, it is difficult to know the exact constraint angles. Furthermore, more constraint angles considerably increase the complexity of regularization.

![Figure 1. Line-fitting results using four constraint models: (a) Manhattan model; (b) MMF model, where $n$, the parameter that controls the number of Manhattan frames, is set to 2; (c) generalized Manhattan model, where the set of constraint angles, $S$, is set to $\{0^\circ, 25.565^\circ, 63.435^\circ, 90^\circ, 116.565^\circ, 153.435^\circ\}$; (d) our directional constraint model, where the number of different directions, $m$, is set to 4.](https://example.com/figure1.png)

We consider, therefore, using implicit angle constraints instead of explicit angle constraints. Specifically, we introduce a parameter, $m$, to constrain the number of different normal vectors of the reconstruction planes. Our motivation is based on a simple observation: In man-made scenes, the number of planes, $|\mathcal{P}|$, is always much more than the number of different normal vectors, $|\mathcal{V}|$. For example, in Fig. 1, we have eight lines, but only four different directions (i.e., normal vectors). We refer to our constraint model as a directional constraint (DC) model, defined as follows.

**Directional constraint model** Given a plane set $\mathcal{P}$, we denote $\mathcal{V}$ as the set of different normal vectors of $\mathcal{P}$. $\mathcal{P}$ is a $m$-DC model, if and only if it satisfies

$$|\mathcal{V}| <= m. \quad (3)$$

Apart from the other three models, the DC model cannot maintain orthogonality. However, it gains the advantage of fewer priori parameters. In fact, the DC model only needs one priori parameter, $m$, (i.e., the number of different normals). An example is given in Fig. 1 (d). There are eight lines but only four different directions. Therefore, we can set $m = 4$ to obtain the correct fitting result. Moreover, by constraining the number of normal vectors, we implicitly establish the connection between planes, which helps obtain better fitting results under noise constraints and outliers (See Fig. 3).

In the next section, we discuss how to reconstruct a set of planes satisfying the DC model.

4. Methodology

4.1. Problem formulation

Given a point cloud, $\mathcal{D}$, equipped with normal vectors, $I$, our task is to find a plane set, $\mathcal{P}$, that best fits the input points while satisfying the DC model. If the input point cloud has no normal vector information, we use the principal component analysis algorithm to estimate their normal vectors.

Our method comprises two steps. First, we reconstruct the normal vector for each point while maintaining the constraints of DC model. Second, the desired planes can be easily obtained by grouping nearby points having the same reconstructed normal. Next, we focus on the first step, because the second step is easily implemented. More specifically, we consider the following problem:

$$\min_{\mathcal{V}} E(I, \mathcal{V}) \quad \text{s.t.} \quad |\mathcal{V}| <= m, \quad (4)$$

where, $\mathcal{V}$ is the set of reconstructed normal vectors; $E$ is the energy that measures how well $\mathcal{V}$ fits $I$. Here, we adopt $L_0$ energy for $E$, owing to its performance under high noise. Moreover, the constraint term, $|\mathcal{V}| <= m$, can be relaxed by introducing a regularization parameter $\lambda_g$. Therefore, Eq. 4 is rewritten as

$$\min_{\mathcal{V}} \sum_{i=1}^{M} \left( ||V_{z_i} - I_i||^2 + \lambda_t \sum_{j \in \mathcal{N}_i} ||V_{z_i} - V_{z_j}||_0 \right) + \lambda_g |\mathcal{V}|, \quad (5)$$

where $M$ is the number of points; $\mathcal{N}_i$ is the neighboring set of the $i$-th point; and $z_i$ is the index of $\mathcal{V}$ for the $i$-th point. Note that Eq. 5 can also satisfy the DC model if we choose the appropriate $\lambda_g$ value.

In Eq. 5, the first term, $||V_{z_i} - I_i||^2$, is used to guide the output normal vectors as close as possible to the input. The second term, $\sum_{j \in \mathcal{N}_i} ||V_{z_i} - V_{z_j}||_0$, is used to constrain the sparsity in the local sense. It guides the normal vector of each point close to its neighboring normals. The third term, $|\mathcal{V}|$, is used to constrain the global sparsity. Two parameters, $\lambda_t$ and $\lambda_g$, are used to balance the influence of the two constraint terms. Eq. 5 can be further divided into the following two subproblems:
Subproblem 1

\[
E_1(I, \lambda) = \min_{V} \sum_{i}^{M} \left( ||V_{z_i} - I_i||^2 + \lambda \sum_{j \in \mathcal{N}_i} ||V_{z_j} - V_{z}||_0 \right). \tag{6}
\]

Subproblem 2

\[
E_2(I, \lambda) = \min_{V \subseteq I} \sum_{i}^{M} ||V_{z_i} - I_i||^2 + \lambda |V|. \tag{7}
\]

If we limit \( V \subseteq I \), then Eq. 7 can be rewritten as:

\[
E_2(I, \lambda) = \min_{V \subseteq I} \sum_{i}^{M} \min_{v \in V} ||v - I_i||^2 + \lambda |V|. \tag{8}
\]

We notice that Eq. 6 is an \( L_0 \) gradient minimization problem and that Eq. 8 is a subset selection problem. This provides a simple notion: we can optimize Eq. 5 by alternately solving two subproblems. By reviewing the state-of-the-art \( L_0 \) minimization [28] and subset selection techniques [13], we can provide a simple but efficient algorithm called, "Global-\( L_0 \)", to solve the problem.

4.2. Global-\( L_0 \) algorithm

The framework of the Global-\( L_0 \) algorithm is given in Algorithm 1, comprising the following two steps.

Algorithm 1 Global-\( L_0 \) algorithm

Input: The input point cloud, \( D \); the normal vectors, \( I \); the neighboring set, \( \mathcal{N} \); the local and global regularization parameter, \( \lambda_l \) and \( \lambda_g \).

Output: the constrained reconstruction normal vectors \( V \).

1: \( \lambda = \text{median}\{\min_{p, \in D} ||I_i - I_N||\} \)
2: \( V = I \)
3: repeat
4: \( L_0 \) minimization: \( V^o \leftarrow \arg\min_{V^o} E_1(V, \lambda) \)
5: Subset selection: \( V^n \leftarrow \arg\min_{V^n} E_2(V^o, \lambda \cdot \lambda_g) \)
6: \( V = V^n \)
7: \( \lambda \leftarrow 2\lambda \)
8: until \( \lambda > \lambda_l \)
9: return \( V \)

\( L_0 \) minimization To solve subproblem 6, we adopt the algorithm based on region fusion [28]. Here, we only need to perform one iteration with the fixed \( \lambda \) value. The time complexity is \( O(M) \).

Subset selection To solve subproblem 8, we adopt greedy unconstrained submodular optimization. Let, \( F(V) = -E_2(V^o, \lambda) \). \( F(V) \) is a submodular function satisfying \( F(X \cup \{x\}) - F(X) \geq F(Y \cup \{x\}) - F(Y) \) for any \( X \subseteq Y \subseteq V \). Therefore, \( F(V) \) is an unconstrained submodular maximization (USM) problem. We can use the linear-time algorithm proposed in [8] to solve it.

However, the complexity of using Algorithm [8] directly is \( O(M^2) \), which is unacceptable. We notice that, after performing region-fusion-based \( L_0 \) minimization, the normal vectors, \( V^o \), are divided into several disjoint regions. Each region shares the same normal vector. Therefore, \( F(V) \) can be simplified to

\[
F(V) = \min_{V \subseteq S} \sum_{v_j \in V, v_i \in S} ||\hat{s}_i - v_j||^2 |s_i| + \lambda |V|, \tag{9}
\]

where \( S \) is the set of regions; \( \hat{s}_i \) is the normal vector of the \( i \)-th region; and \( V^S \) is the set of normal vectors consisting of \( \hat{s}_i, i \in [1, |S|] \).

Additionally, we can ignore the regions whose number of points is less than a threshold, \( \tau \). \( \tau \) can be explained as the minimal support points for a plane (i.e., the filtered regions are treated as outliers). Thus, the time complexity of our subset selection step is reduced to \( O(\min(M/\tau, |S|^2)) \). In practice, it can assume \( O(\min(M/\tau, |S|^2)) < O(M) \).

More details are described in Algorithm 2.

Algorithm 2 Subset selection

Input: The disjoint region set \( S \); each region shares the same normal vector \( \hat{s}_i \).

Output: the selected normal vectors, \( V \).

1: Sort \( \mathcal{S} \) in descending order according to their number of points.
2: Initialize: \( V \leftarrow \emptyset \); \( X \leftarrow \emptyset \); \( Y \leftarrow S \)
3: for \( i = 1, \ldots, |S| \) and \( |S_i| > = \tau \) do
4: \( a \leftarrow F(X \cup \{S_i\}) - F(X) \)
5: \( b \leftarrow F(Y \setminus \{S_i\}) - F(Y) \)
6: if \( a \geq b \) then
7: \( X \leftarrow X \cup \{S_i\} \)
8: \( V \leftarrow V \cup \{\hat{s}_i\} \)
9: else
10: \( Y \leftarrow Y \setminus \{S_i\} \)
11: end if
12: end for
13: return \( V \)

Algorithm 2 has a 1/3 approximation guarantee (i.e., the energy of its solution is always more than 1/3 times the optimal energy). The randomized version [8] has a better approximate rate of 1/2. However, we choose the deterministic algorithm, because it is more stable. Furthermore, with a heuristic sequence (i.e., descending order by cardi-
nality), our solution is usually better than the theoretical upper bound.

The Global-$L_0$ algorithm can be seen as an extension of $L_0$ gradient minimization algorithms. The traditional $L_0$ gradient minimization methods only consider $L_0$ regularity between local neighbors. Our algorithm introduces global regularity by limiting the number of different normal vectors. It establishes implicit relationships between the disjoint regions. In Section 5, we demonstrate that these global relationships are very useful, especially in man-made environments.

Plane reconstruction For a post-processing, the constrained planes can be constructed from disjoint regions. Additionally, the normal vector of each plane is set to the reconstruction normal vector of the corresponding region.

### 4.3. Difference to the multilabeling problem

Eq. 5 looks like the multilabeling problem of [12, 1]. However, they are quite different. First, the energy formulation in [12, 1] only constrained the number of models (planes in our context). In contrast, Eq. 4 constrains the number of plane normals, because, it constrains the size of the label set. Second, in the multi-labeling problems, the label set, $L$ (or $V$ in our formulation), is usually known. For example, Pearl [12] must first compute an initial set of models or labels via sampling. Our algorithm is designed to solve the problem without an initial label set. In other words, the size of the initial label set in our problem is much larger than the label size in multilabeling problems. Therefore, an efficient algorithm is desired.

### 5. Experiments

Our method was implemented in C++ and run on Linux Ubuntu 16.04 with one core of Intel i7 CPU (2.5 GHz) and 8-GB memory.

#### 5.1. Parameter setting

There are four parameters adopted in our method. They are $\tau$, the minimal number of points that support a plane; $K$, the number of neighbors for each point; $\lambda_l$, the local regularization for normal vectors in $L_0$ minimization; and $\lambda_g$, the global regularization for normal vectors. $\tau$ and $K$ are common parameters, and their settings have been fully studied. Here, we set $\tau = 100$ and $K = 10$ for the following experiments.

Fig. 2 demonstrates the effects of $\lambda_l$ and $\lambda_g$. We use the same color to draw the points with the same normal vector. In Fig. 2(b-d), we first set $\lambda_g = 0$ and increase the value of $\lambda_l$. One can observe that, as the $\lambda_l$ increases, the number of regions decreases. However, the normal vectors for each region are different (i.e., $|V|$ equals the number of regions). In Fig. 2(e-h), we set $\lambda_l = 0.5$ and increase the value of $\lambda_g$. The number of regions and the different normal vectors decrease simultaneously. When $\lambda_g$ reaches 10,000, only the three most frequently normal are preserved, satisfying the Manhattan model.

In practice, for data having high noise, we choose higher values for $\lambda_l$ and $\lambda_g$. One can also simply set $\lambda_l$ equal to the one in [28]. Additionally, if the user has a prior knowledge of the scene (the value of $m$), then one can choose an appropriate $\lambda_g$ value.

#### 5.2. 2D line fitting

We first study the robustness of the proposed method on the 2D line-fitting problem with respect to seq-RANSAC, J-linkage [37], T-linkage [24], and Pearl [12]. We implement seq-RANSAC, J-linkage, and T-linkage ourselves, whereas the implementation of Pearl is obtained from [38].

As shown in Fig. 3, we generate eight lines, four of which are parallel, and four of which form a square. Each line comprises of 100 inliers, contaminated by Gaussian noise and outliers of different percentages. When we say 50% outliers, it means that the number of outliers is half of the entire set. Note that the ground truth lines only have three different directions. Our method can benefit from this constraint.

The results are collected in Fig. 3, where $e_n$ is the root-mean-square (RMS) error of the angles between the reconstruction normal vectors and ground truth normals. It is defined as

$$e_n = \sqrt{\frac{\sum_{p \in \mathcal{G}} (\angle(n_p, \hat{n}_p))^2}{|\mathcal{G}|}},$$

where $\mathcal{G}$ represents the ground truth inliers.
As noise and outliers increase, Seq-RANSAC, J-linkage, T-linkage, and Pearl miss the ground-truth line segments, and generate more false positives. J-linkage and T-linkage aggregate the preference set without exploiting the regularity of the data. Moreover, J-linkage and T-linkage rely on the selection of the initial set. The tendency of decreasing the percentage of inliers affects the performance of J-linkage and T-linkage. Pearl solves the line-fitting problem by presenting an optimization framework. However, it only constrains the number of fitted lines and does not constrain the number of directions. It also relies on the selection of the initial set.

Our method achieves the best $e_{\text{NI}}$ in all situations. The reason can be ascribed to the constraint on the number of directions and the independence of the initial set. Note that, the number of directions is a non-local constraint, which similar to non-local denoising methods [7]. When outliers reach 80% (20% inliers), our method generates two false positives and misses one line segment. However, the directions of the fitted lines are correct, and the number of different directions (including false positives) remains three. Moreover, our method has the best time performance, significantly faster than the other three methods.

5.3. Plane reconstruction on 3D manual point sets

We further study the robustness of the proposed method on 3D data with respect to RAPTER [27], piecewise planar surface reconstruction (PPSR) [10], and Pearl [12]. We obtain the implementation of RAPTER from [26]. Here, we use the region-growing technique of [10] to extract the initial planes for Pearl, leading to better reconstruction results.

Noisy dodecahedron We first demonstrate the performance of these methods on a noisy 3D point set. As shown in Fig. 4, we sampled 100,000 points from a dodecahedron and perturbed each point by adding 0.02l Gaussian noise, where l is the diagonal length of the axis-aligned bounding box of the original point cloud.

![Figure 4. Comparison results between our method and various plane reconstruction methods on a noisy dodecahedron. The mesh models are constructed by point-set structuring [19] based on the reconstructed planes. The normal vectors of the obtained planes are shown in black boxes. In parentheses, we provide the running time (s) of each method.](image)

RAPTER is susceptible to Gaussian noise, and, because it relies on the initial planes obtained from region growing, its results will be less than satisfactory when region growing fails. PPSR improves the performance of traditional region growing by carefully selecting seed points. However,
it still generates too many redundant planes. Pearl [12] obtains a much better result but loses accuracy at the boundaries. Benefiting from the robustness of $L_0$ optimization and parallel faces constraints, our method estimates the correct model. Furthermore, our method is much faster than the others.

Empire State Building with outliers We next test our method with RAPTER on the Empire State Building with different level of outliers. The ground truth model is downloaded from [20]. We sample 1,000,000 points from ground-truth model and add 250,000 and 1,000,000 outliers, respectively.

In addition to $e_n$, we evaluate the accuracy of the reconstruction planes by computing the RMS of the distance from the ground-truth planes, $\mathcal{P}$, i.e.,

$$e_p = \sqrt{\frac{\sum_{p \in \mathcal{G}} (\text{dist}(p, P_L(p))^2)}{|\mathcal{G}|}},$$  

where, $L(p)$ denotes the index of the plane closest to $p$.

$e_p$ in Eq. 11 is used to penalize the insufficient reconstruction of planes. However, it does not penalize redundant planes, which is critical for point clouds with outliers. Therefore, for each inlier point, $p$, we calculate its projection point, $p'$, on the reconstruction planes. Then, we evaluate the distance from $p'$ to the nearest point using ground truth points, i.e.,

$$e_q = \sqrt{\frac{\sum_{p \in P \wedge L(p) \neq \text{Outlier}} (\min_{q \in \mathcal{G}} \text{dist}(p', q))^2}{|P|}}.$$  

Note that $e_q$ can penalize the inaccurate and redundant reconstruction of planes simultaneously.

Table 1. Comparisons on the Empire State Building in terms of $e_n$, $e_p$, $e_q$, and running time. The first two rows are the comparison results for the data with 20% outliers and the next two rows are the results for the data with 50% outliers.

| Method          | #Planes | $e_n$ | $e_p$  | $e_q$  | Time |
|-----------------|---------|-------|--------|--------|------|
| RAPTER          | 195     | 3.5   | 1.1e-4 | 6.5e-5 | 1.3e5|
| Ours            | 171     | 3.5   | 1.5e-4 | 3.8e-5 | 55   |
| RAPTER          | 211     | 3.5   | 1.1e-4 | 7.8e-5 | 1.5e5|
| Ours            | 178     | 3.5   | 1.5e-4 | 3.8e-5 | 38   |

Table 2. Plane segmentation comparison results on RGBD dataset. The time is in seconds and averaged over the images.

| Method          | GCE   | LCE   | Time |
|-----------------|-------|-------|------|
| seq-RANSAC      | 0.44  | 0.36  | 7.98 |
| Pearl           | 0.36  | 0.28  | 60.11|
| Global-$L_0$ (Ours) | 0.33  | 0.24  | 21.63|

Figure 5. Comparison of our method and RAPTER on "Empire state building" with 20% and 50% outliers. The black boxes show the distribution of normal vectors.

As shown in the black boxes in Fig. 5, the normal vector distribution of the original point clouds is very noisy. Nevertheless, both methods achieved visually good plane reconstruction results. As a defect, neither method can produce a complete plane reconstruction at the top part of the building (see close-up views in the yellow boxes). The reason is that the small plane is more susceptible to outliers, resulting in incorrect reconstruction planes. Despite that, we can observe that RAPTER produces several planes with inaccurate directions; whereas our method produces relatively few but more accurate planes. In the close-up views in the red boxes, we also observe that our method outperforms RAPTER in details.

Quantitative comparison results are summarized in Table 1. Compared to RAPTER, our method achieves better $e_n$ and $e_q$ but worse $e_p$. The normal vectors reconstructed by our method are significantly more accurate than RAPTER. This is also indicated in the black boxes of Fig. 5. Furthermore, RAPTER is very time consuming, because it needs to solve a large-scale mixed-integer programming system. In contrast, our method is approximately 2,000 times faster.

5.4. Plane reconstruction on real-world point clouds

To quantitatively evaluate the performance of the proposed method in practice, two real-world dataset are considered. In the following experiments, we set $\lambda_l = 10$ and $\lambda_q = 1,000$. The parameters for sequential RANSAC and Pearl are fine-tuned.

We first test our method on the NYU-depth dataset [35]. This dataset was originally designed for object detection. We use it to quantitatively evaluate the performance of plane segmentation. A subset of 218 images, mainly containing planar objects, is selected from the dataset. The subset includes kitchen, bedroom, and bathroom scenes. Scenes containing too many non-planar objects, such as offices and stores, are excluded, because our method is not designed for non-planar scenes. Two standard segmentation metrics, global consistency error (GCE), and local consistency error (LCE) from [32], are introduced here for evaluation.

The quantitative results are summarized in Table 2.
where two typical scenes are shown in Fig. 6. It can be observed that our method is significantly superior to sequential RANSAC and Pearl in terms of visual and quantitative results.

We further test our method on indoor laser-scanning point clouds \(^1\), including five datasets: UZH Ifi (12 data), UZH Irchel (10 data), ETH (18 data), Rooms detection datasets (9 data), and Full 3D (8 data). Owing to the memory limitations of PEARL, the input data was downsampled by performing Poisson-disk sampling \([6]\) using a resolution \(r = 0.02m\).

The quantitative comparison results are shown in Fig. 7 and some typical reconstruction results are shown in Fig. 8. Owing to the lack of ground truth, only \(e_p\) and \(e_q\) are evaluated here. Nevertheless, we can easily observe that our method is superior to the other two methods. PEARL has a high computational complexity and has difficulty maintaining small-scale planar structures. RANSAC-based methods tend to merge unconnected planes, making it difficult to maintain the edges of local structures. The proposed method not only maintains the local structure, it also guarantees the global constraints. Therefore, we achieve the best results.

6. Conclusion

In this paper, we presented a fast algorithm for RCPR problems. Our algorithm was based on a constraint model, requiring less prior knowledge compared to traditional RCPR algorithms. We showed that, by considering the constraints, plane reconstruction results could significantly improved, especially for data with high-level outliers and noise. Furthermore, our algorithm is efficient, between 3 and 1,000-times faster than the existing RCPR algorithms.

References

[1] P. Amayo, P. Pinis, L. M. Paz, and P. Newman. Geometric multi-model fitting with a convex relaxation algorithm. In *Computer Vision and Pattern Recognition*. IEEE, 2018.

[2] M. Arikan, M. Wimmer, and S. Maierhofer. O-snap:optimization-based snapping for modeling architecture. *ACM Transactions on Graphics*, 32(1):1–15, 2013.

[3] J. C. Bazin, Y. Seo, C. Demonceaux, and P. Vasquez. Globally optimal line clustering and vanishing point estimation in

---

\(^1\)https://www.ifi.uzh.ch/en/vmml/research/datasets.html
Figure 8. Plane reconstruction results of indoor point clouds.

Manhattan world. In Computer Vision and Pattern Recognition, pages 638–645. IEEE, 2012.

[4] D. Borrmann, J. Elseberg, L. Kai, and A. Nchter. The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. 3D Research, 2(2):32:1–32:13, 2011.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In International Conference on Computer Vision, pages 377–384. IEEE, 2002.

[6] R. Bridson. Fast poisson disk sampling in arbitrary dimensions. In ACM SIGGRAPH 2007 Sketches, SIGGRAPH ’07, New York, NY, USA, 2007. ACM.

[7] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In Computer Vision and Pattern Recognition, pages 60–65. IEEE, 2005.

[8] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In Foundations of Computer Science, pages 649–658, 2012.

[9] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging & Vision, 40(1):120–145, 2011.

[10] A. L. Chauve, P. Labatut, and J. P. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In Computer Vision and Pattern Recognition, pages 1261–1268. IEEE, 2010.

[11] J. M. Coughlan and A. L. Yuille. Manhattan world: compass direction from a single image by Bayesian inference. In International Conference on Computer Vision, volume 2, pages 941–947. IEEE, 1999.

[12] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. International Journal of Computer Vision, 96(1):1–27, 2012.

[13] E. Elhamifar and M. C. D. P. Kaluza. Online summarization via submodular and convex optimization. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1818–1826, 2017.

[14] H. Fang, F. Lafarge, and M. Desbrun. Planar shape detection at structural scales. In Computer Vision and Pattern Recognition, Salt Lake City, United States, 2018.

[15] E. Fernandez-Moral, W. Mayol-Cuevas, V. Arevalo, and J. Gonzalez-Jimenez. Fast place recognition with plane-based maps. In International Conference on Robotics and Automation, pages 2719–2724. IEEE, 2013.

[16] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):726–740, 1981.

[17] K. Joo, T. Oh, J. Kim, and I. S. Kweon. Robust and globally optimal Manhattan frame estimation in near real time. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2018.

[18] S. Korman and L. Roce. Latent RANSAC. In Computer Vision and Pattern Recognition, pages 6693–6702. IEEE, 2018.

[19] F. Lafarge and P. Alliez. Surface reconstruction through point set structuring. Computer Graphics Forum, 32(2):225–234, 2013.

[20] F. Lafarge and P. Alliez. Surface reconstruction through point set structuring. http://www-sop.inria.fr/members/Florent.Lafarge/benchmark/surface reconstruction/reconstruct 2013.

[21] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. ACM Transaction on Graphics, 30(4):52:1–52:12, 2011.

[22] Y. Lin, C. Wang, B. Chen, D. Zai, and J. Li. Facet segmentation-based line segment extraction for large-scale point clouds. IEEE Transactions on Geoscience & Remote Sensing, 55(9):4839–4854, 2017.

[23] Y. Lin, C. Wang, D. Zhai, W. Li, and J. Li. Toward better boundary preserved supervoxel segmentation for 3D point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 143:39–47, 2018.

[24] L. Magri and A. Fusiello. T-linkage: A continuous relaxation of J-Linkage for multi-model fitting. In Computer Vision and Pattern Recognition, pages 3954–3961. IEEE, 2014.

[25] L. Magri and A. Fusiello. Multiple models fitting as a set coverage problem. In Computer Vision and Pattern Recognition, pages 3318–3326. IEEE, 2016.

[26] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra. RAPlter: rebuilding man-made scenes with regular arrangements of planes. Acm Transactions on Graphics, 34(4):103, 2015.
[28] R. M. H. Nguyen and M. S. Brown. Fast and effective $L_0$ gradient minimization by region fusion. In *IEEE International Conference on Computer Vision*, pages 208–216, 2015. 4, 5

[29] S. Oesau, F. Lafarge, and P. Alliez. Planar shape detection and regularization in tandem. *Computer Graphics Forum*, page 14, 2015. 2

[30] S. Oesau, F. Lafarge, and P. Alliez. Object classification via planar abstraction. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:225–231, 2016. 1

[31] J. Papon, A. Abramov, M. Schoeler, and F. Wrugtter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *Computer Vision and Pattern Recognition*, pages 2027–2034. IEEE, 2013. 2

[32] M. Polak, H. Zhang, and M. Pi. An evaluation metric for image segmentation of multiple objects. *Image and Vision Computing*, 27(8):1223–1227, 2009. 7

[33] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm. USAC: a universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 35(8):2022–2038, 2013. 2

[34] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for pointcloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2010. 1, 2

[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, pages 746–760, 2012. 7

[36] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher. A mixture of Manhattan frames: Beyond the Manhattan world. In *Computer Vision and Pattern Recognition*, pages 3770–3777. IEEE, 2014. 2

[37] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-Linkage. In *European Conference on Computer Vision*, pages 537–547. IEEE, 2008. 2, 5

[38] O. Veksler and A. Delong. Pearl implementation. http://vision.csd.uwo.ca/code/, 2012. 5

[39] H. Wang, G. Xiao, Y. Yan, and D. Suter. Searching for representative modes on hypergraphs for robust geometric model fitting. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):1–1, 2018. 2

[40] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990. 1, 2

[41] Q. Y. Zhou and U. Neumann. 2.5D building modeling by discovering global regularities. In *Computer Vision and Pattern Recognition*, pages 326–333. IEEE, 2012. 2

[42] Z. Zhou, H. Jin, and Y. Ma. Robust plane-based structure from motion. In *Computer Vision and Pattern Recognition*, pages 1482–1489. IEEE, 2012. 1

[43] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multi-RANSAC algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, volume 3, pages III–153. IEEE, 2005. 2