

Globally Optimal and Efficient Vanishing Point Estimation in Atlanta World

Haoang Li¹, Pyojin Kim^{2*}, Ji Zhao³, Kyungdon Joo⁴,
Zhipeng Cai⁵, Zhe Liu⁶, and Yun-Hui Liu¹

¹ The Chinese University of Hong Kong, Hong Kong, China

² Simon Fraser University, Burnaby, Canada

³ TuSimple, Beijing, China

⁴ Carnegie Mellon University, Pittsburgh, USA

⁵ The University of Adelaide, Adelaide, Australia

⁶ University of Cambridge, Cambridge, UK

{haoang.li.cuhk, pjinkim1215, zhaoji84, kdjoo369}@gmail.com

zhipeng.cai@adelaide.edu.au, zl457@cam.ac.uk, yhliu@mae.cuhk.edu.hk

Abstract. Atlanta world holds for the scenes composed of a vertical dominant direction and several horizontal dominant directions. Vanishing point (VP) is the intersection of the image lines projected from parallel 3D lines. In Atlanta world, given a set of image lines, we aim to cluster them by the unknown-but-sought VPs whose number is unknown. Existing approaches are prone to missing partial inliers, rely on prior knowledge of the number of VPs, and/or lead to low efficiency. To overcome these limitations, we propose the novel mine-and-stab (MnS) algorithm and embed it in the branch-and-bound (BnB) algorithm. Different from BnB that iteratively branches the full parameter intervals, our MnS directly mines the narrow sub-intervals and then stabs them by probes. We simultaneously search for the vertical VP by BnB and horizontal VPs by MnS. The proposed collaboration between BnB and MnS guarantees global optimality in terms of maximizing the number of inliers. It can also automatically determine the number of VPs. Moreover, its efficiency is suitable for practical applications. Experiments on synthetic and real-world datasets showed that our method outperforms state-of-the-art approaches in terms of accuracy and/or efficiency.

1 Introduction

A set of image lines projected from parallel 3D lines intersect at a common point called the vanishing point (VP). VP has various applications such as camera calibration [19, 22], shape estimation [12] and robot navigation [20, 21, 37]. In structured environments such as man-made scenes, several dominant directions (DDs) exist. The well-known Manhattan world [9] consists of three mutually orthogonal DDs. However, this model is not suitable to represent many structures such as non-orthogonal walls. Atlanta world [30] holds for more general scenes.

* Corresponding author: Pyojin Kim (email: pjinkim1215@gmail.com)

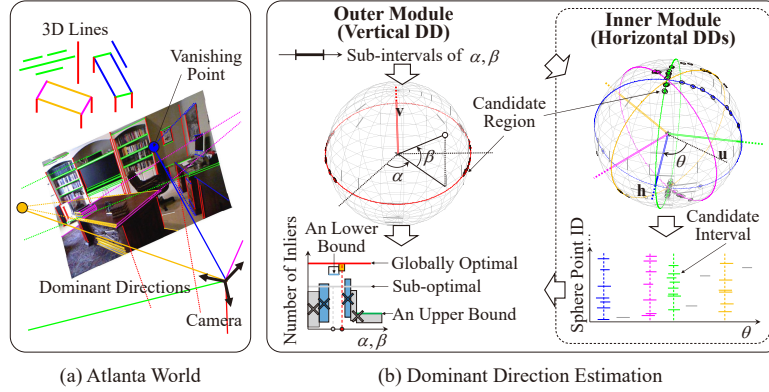


Fig. 1. (a) Atlanta world. (b) Pipeline of our method. Our outer module searches for the vertical DD by BnB. Our inner module searches for the horizontal DDs by MnS.

It is composed of a vertical DD and several horizontal DDs (see Fig. 1(a)). The horizontal DDs are not necessarily orthogonal to each other but orthogonal to the vertical DD. In Atlanta world, given a set of lines in a calibrated image, we aim to cluster them by the unknown-but-sought VPs whose number is unknown.

The direction defined by the camera center and a VP is aligned to a DD [15]. Based on this constraint, VP estimation can be reformulated as computing DDs. Existing VP/DD estimation approaches for Atlanta world are prone to missing partial inliers [1, 20, 35], rely on prior knowledge of the number of VPs [16, 30], and/or lead to low efficiency [17]. To overcome these limitations, we propose the novel mine-and-stab (MnS) algorithm and embed it in the branch-and-bound (BnB) algorithm. Different from BnB that iteratively branches the full parameter intervals, our MnS directly mines the narrow sub-intervals and then stabs them by probes. As shown in Fig. 1(b), we simultaneously search for the vertical DD by our BnB-based outer module, and horizontal DDs by our MnS-based inner module. The proposed collaboration between BnB and MnS guarantees global optimality in terms of maximizing the number of inliers. It can also automatically determine the number of DDs. Moreover, its efficiency is suitable for practical applications. In addition, given the vertical DD obtained by inertial measurement unit (IMU), our inner module can run independently and achieve real-time efficiency. Our main contributions are summarized as follows.

- Our method guarantees global optimality in terms of maximizing the number of inliers thanks to the collaboration between BnB and MnS.
- Our method can automatically determine the number of VPs thanks to MnS.
- Our method leads to high efficiency thanks to low-dimensional search space of BnB and low computational complexity of MnS.
- We established an image dataset with the manually extracted lines as well as ground truth VPs. It is publicly available on our project website⁷.

⁷ https://sites.google.com/view/haoangli/projects/eccv20_vp

2 Related Work

Existing VP estimation methods applicable to Atlanta world can be classified into two main categories in terms of whether prior knowledge of the number of VPs is required [10, 16, 30, 32] or not [1, 17, 18, 20, 29, 35].

Methods with prior knowledge. The expectation-maximization algorithm [8] has been applied to VP estimation [30]. This method assigns each line with a cluster label based on the known number of VPs, and then alternately updates these labels and VPs. However, it is sensitive to the initial labels and prone to getting stuck into a local optimum. Classical RANSAC [10] is inherently suitable for Manhattan world with three VPs [4, 5]. However, in Atlanta world, it requires the known number of VPs to determine the number of the sampled lines at an iteration [17, 38]. An alternative sampling strategy is to fix the number of samples at an iteration [32]. Accordingly, VPs are sequentially estimated on the remaining outliers. However, RANSAC may fail to retrieve all the inliers due to the effect of noise. Joo et al. [16] first proposed an approach that guarantees global optimality in terms of maximizing the number of inliers. They used the known number of VPs to define the parameter set and searched for all these parameters by BnB. While this method provides high accuracy, its efficiency is unsatisfactory (generally more than 10 seconds per image). In addition, Antunes et al. [2] proposed a method that can handle the images with radial distortion. Since prior knowledge of the number of VPs may not be available in practice, the above methods lead to relatively low generality.

Methods without prior knowledge. The Hough transform-based method maps the image lines to the great circles, and generates a histogram of the intersections of these circles [29]. The bins with high cardinalities correspond to VPs. However, this method is sensitive to the histogram resolution. Several methods based on the variants of RANSAC [26, 36] can automatically determine the number of VPs. For example, Tardif et al. [35] leveraged J-Linkage [36] to generate the image line descriptors by numerous samplings, and then clustered the lines based on the descriptor similarity. However, this method is sensitive to noise and also leads to unsatisfactory efficiency. Li et al. [20] used T-Linkage [26] to estimate VPs. While this method improves the accuracy of the above J-Linkage-based approach, it still fails to guarantee global optimality in terms of maximizing the number of inliers. Antunes and Barreto [1] proposed a message passing-based method, but it may also get stuck into a local optimum. Moreover, the above approaches fail to satisfy the orthogonality between the vertical and horizontal DDs. In contrast, our method satisfies this orthogonality. Pham et al. [28] proposed an energy minimization method. However, it requires sampling and thus leads to unsatisfactory accuracy. Joo et al. [17] proposed a Bayesian information criterion-based strategy to determine the number of VPs. They integrated it into the above globally optimal approach [16] as a pre-processing step. However, it is time-consuming and may miss some clusters.

Overall, existing approaches fail to achieve high generality, accuracy, and efficiency simultaneously. Our method overcomes these limitations thanks to the collaboration between BnB and MnS, as will be shown in the experiments.

3 Algorithm Overview

DD estimation in Atlanta world is a high-dimensional multi-model fitting problem subject to constraints. High dimension represents a relatively large number of parameters to estimate; Multiple models represent a set of DDs whose number is unknown; Constraints represent that each horizontal DD is orthogonal to the vertical DD. As introduced above, the original BnB [16] can hardly handle this problem well since it leads to low efficiency and also requires prior knowledge of the number of DDs. To overcome this limitation, we propose the novel MnS and embed it in BnB⁸. Our MnS has three main advantages. First, it can automatically determine the number of horizontal DDs. Second, it leads to low computational complexity. Third, it accelerates BnB by reducing the search space of BnB. As shown in Fig. 1(b), our method satisfies a nested structure. We simultaneously search for the vertical DD by our BnB-based outer module, and the horizontal DDs by our MnS-based inner module. If an image line is fitted by a vertical or horizontal VP/DD, we call it the vertical or horizontal inlier.

Outer Module. As shown in Fig. 1(b-outer), in the camera frame whose origin is the ball center, we use the unknown-but-sought azimuth $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and elevation $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ to parametrize the vertical DD \mathbf{v} by

$$\mathbf{v}(\alpha, \beta) = [\cos \alpha \cdot \cos \beta, \sin \alpha \cdot \cos \beta, \sin \beta]^\top. \quad (1)$$

Our BnB-based outer module iteratively branches the full intervals of α and β , obtaining the wide-to-narrow sub-intervals. Given a pair of sub-intervals of α and β , our outer module computes a perturbed vertical DD based on Eq. (1), and uses this DD to compute the bounds of the number of vertical inliers. Then it passes the sub-intervals of α and β to our inner module.

Inner Module. As shown in Fig. 1(b-inner), we compute a unit vector $\mathbf{u} = [-\sin \alpha, \cos \alpha, 0]^\top$ orthogonal to the vertical DD \mathbf{v} . Then we rotate \mathbf{u} around \mathbf{v} by an unknown-but-sought angle $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ to parametrize a horizontal DD \mathbf{h} by

$$\mathbf{h}(\alpha, \beta, \theta) = [[a_1, b_1]\mathbf{t}, [a_2, b_2]\mathbf{t}, [a_3, b_3]\mathbf{t}]^\top, \quad (2)$$

where $\{a_i, b_i\}_{i=1}^3$ are expressed by the angles α and β , and $\mathbf{t} = [\cos \theta, \sin \theta]^\top$. All the horizontal DDs $\{\mathbf{h}_n\}$ ($n = \text{I, II} \cdots N$) share the common coefficients $\{a_i, b_i\}_{i=1}^3$ but have different rotation angles $\{\theta_n\}$. This parametrization satisfies the orthogonality between the vertical and horizontal DDs. For each image line, our MnS-based inner module directly mines a narrow sub-interval from the full interval of θ . This image line is treated as an inlier within this sub-interval. We call this sub-interval the “candidate interval”. Then our inner module finds a set of probes, each of which stabs more than τ candidate intervals (τ is a threshold). The number of probes is the number N of horizontal DDs; The positions of probes correspond to the angles $\{\theta_n\}$ of horizontal DDs; The number of the candidate intervals stabbed by these probes is the number of horizontal inliers.

⁸ The reason why we do not use MnS independently is that MnS is inherently suitable for low-dimensional problems.

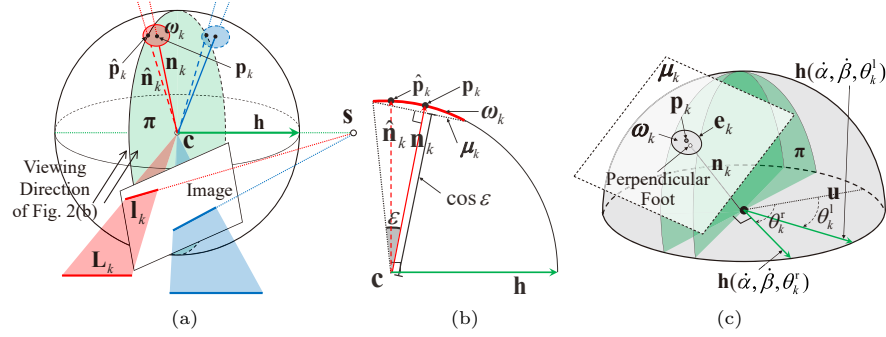


Fig. 2. (a) The noise-free sphere point $\hat{\mathbf{p}}_k$ lies on the dominant plane π , while the observed point \mathbf{p}_k slightly deviates from π . (b) We expand \mathbf{p}_k into the spherical cap ω_k , and call ω_k the candidate region. (c) The candidate interval $[\theta_k^l, \theta_k^r]$ corresponds to the case that the dominant plane π intersects with the candidate region ω_k .

Since the input of our inner module is the sub-intervals (instead of exact values) of α and β , $\{a_i, b_i\}_{i=1}^3$ in Eq. (2) are perturbed, and further the candidate intervals are perturbed. Accordingly, our inner module returns the bounds (instead of exact value) of the number of horizontal inliers to our outer module.

Based on the above bounds of the number of vertical and horizontal inliers, we obtain the globally optimal DDs that maximize the total number of inliers. In Section 4, we consider the simplified case that the vertical DD and candidate intervals are not perturbed. In Section 5, we consider the practical case that the vertical DD and candidate intervals are perturbed.

4 Simplified Case without Perturbation

In this section, we consider the simplified case that BnB generates the coarse-to-fine values (instead of wide-to-narrow sub-intervals) of the angles α and β , which is called the quasi-exhaustive search [3]. Accordingly, the vertical DD \mathbf{v} in Eq. (1) and the coefficients $\{a_i, b_i\}_{i=1}^3$ of the horizontal DD \mathbf{h} in Eq. (2) are not perturbed. Given a pair of exact values $\hat{\alpha}$ and $\hat{\beta}$ of the angles α and β (regardless of accuracy), we aim to identify inliers. Intuitively, if $\hat{\alpha}$ and $\hat{\beta}$ are close to the ground-truth values, the known DD $\mathbf{v}(\hat{\alpha}, \hat{\beta})$ and coefficients $\{a_i(\hat{\alpha}, \hat{\beta}), b_i(\hat{\alpha}, \hat{\beta})\}_{i=1}^3$ are accurate, and thus the number of the identified inliers is large.

4.1 Defining Dominant Plane and Candidate Region

As shown in Fig. 2(a), the image line \mathbf{l}_k is projected from the 3D line \mathbf{L}_k ($k = 1, 2, \dots$). The camera center \mathbf{c} and \mathbf{L}_k define the projection plane. The unit projection plane normal \mathbf{n}_k is computed by the endpoints of \mathbf{l}_k [24]. A set of image lines $\{\mathbf{l}_k\}$ intersect at a horizontal VP \mathbf{s} . The direction defined by \mathbf{s} and the camera center \mathbf{c} is aligned to an unknown-but-sought horizontal

DD $\mathbf{h}(\dot{\alpha}, \dot{\beta}, \theta)$. We define the horizontal dominant plane π , which is orthogonal to the DD $\mathbf{h}(\dot{\alpha}, \dot{\beta}, \theta)$ and also passes through the camera center \mathbf{c} , by

$$\pi(\dot{\alpha}, \dot{\beta}, \theta) : \mathbf{h}(\dot{\alpha}, \dot{\beta}, \theta) \cdot [x, y, z] = 0. \quad (3)$$

Similarly, we use the known vertical DD $\mathbf{v}(\dot{\alpha}, \dot{\beta})$ to define the vertical dominant plane π' by

$$\pi'(\dot{\alpha}, \dot{\beta}) : \mathbf{v}(\dot{\alpha}, \dot{\beta}) \cdot [x, y, z] = 0. \quad (4)$$

As shown in Fig. 2(a), for a set of noise-free inlier image lines $\{\mathbf{l}_k\}$ associated with the same VP \mathbf{s} , their corresponding unit projection plane normals $\{\hat{\mathbf{n}}_k\}$ are orthogonal to the same horizontal DD \mathbf{h} . Accordingly, the terminal points of $\{\hat{\mathbf{n}}_k\}$, which are denoted by the sphere points $\{\hat{\mathbf{p}}_k\}$, lie on the same horizontal dominant plane π (see Fig. 1(b-inner)). Similarly, there are some noise-free inlier sphere points lying on the vertical dominant plane π' (see Fig. 1(b-outer)). In practice, an observed projection plane normal \mathbf{n}_k is affected by noise, and thus the sphere point \mathbf{p}_k does not strictly lie on a dominant plane. We use the candidate region to model this error in the following.

As shown in Fig. 2(b), we assume that the angle between the noise-free projection plane normal $\hat{\mathbf{n}}_k$ and the observed normal \mathbf{n}_k is smaller than the threshold ε ($\varepsilon = 2^\circ$ in our experiments). Accordingly, we expand the observed sphere point \mathbf{p}_k into the spherical cap ω_k that encloses the noise-free point $\hat{\mathbf{p}}_k$. We call ω_k the candidate region. To mathematically express ω_k , we define the 3D secant plane μ_k of the unit sphere. As shown in Figs. 2(b) and 2(c), μ_k is orthogonal to the observed projection plane normal \mathbf{n}_k . The vertical distance between the secant plane μ_k and the sphere center \mathbf{c} is $\cos \varepsilon$. Accordingly, we express μ_k by $\mathbf{n}_k \cdot [x, y, z] + \cos \varepsilon = 0$. Then we define the edge \mathbf{e}_k of the candidate region ω_k as the intersection of μ_k and unit sphere \mathbb{S}^2 as

$$\mathbf{e}_k : \begin{cases} \mu_k : \mathbf{n}_k \cdot [x, y, z] + \cos \varepsilon = 0 \\ \mathbb{S}^2 : x^2 + y^2 + z^2 = 1 \end{cases} \quad (5)$$

The edge \mathbf{e}_k encloses the candidate region ω_k .

Based on the candidate region, we re-define the inlier. Specifically, if the candidate region ω_k intersects with a dominant plane, we treat the sphere point \mathbf{p}_k as an inlier. Since the vertical dominant plane $\pi'(\dot{\alpha}, \dot{\beta})$ is known, identifying the vertical inlier is straightforward. In the following, we introduce how we leverage the proposed MnS to search for the unknown angle θ of the horizontal dominant plane $\pi(\dot{\alpha}, \dot{\beta}, \theta)$ and also identify the horizontal inliers.

4.2 Mining Candidate Interval

For each sphere point, we mine its candidate interval based on the above candidate region. As shown in Fig. 2(c), the candidate interval $[\theta_k^l, \theta_k^r]$ ⁹ of the point \mathbf{p}_k corresponds to the case that the horizontal dominant plane $\pi(\dot{\alpha}, \dot{\beta}, \theta)$ intersects with the candidate region edge \mathbf{e}_k . Mathematically, the quadratic system defined

⁹ For writing simplification, we denote θ_k by θ hereinafter.

by Eqs. (3) and (5) has two distinct real solutions that are the coordinates of two plane-edge intersections. We use basic variable substitutions to eliminate the variables y and z of this system, obtaining a quadratic polynomial equation with respect to a single variable x as

$$\lambda_2(\dot{\alpha}, \dot{\beta}, \theta) \cdot x^2 + \lambda_1(\dot{\alpha}, \dot{\beta}, \theta) \cdot x + \lambda_0(\dot{\alpha}, \dot{\beta}, \theta) = 0, \quad (6)$$

where the coefficients $\{\lambda_2, \lambda_1, \lambda_0\}$ are composed of the known $\dot{\alpha}$ and $\dot{\beta}$ as well as the unknown $\cos \theta$ and $\sin \theta$. Therefore, we formulate the case that the dominant plane intersects with the candidate region edge as the case that the quadratic polynomial in Eq. (6) has two distinct real roots. We compute the discriminant of this polynomial as $\Delta(\dot{\alpha}, \dot{\beta}, \theta) = \lambda_1^2 - 4 \cdot \lambda_0 \cdot \lambda_2$. In the following, we aim to find the candidate interval with respect to θ where $\Delta(\dot{\alpha}, \dot{\beta}, \theta) > 0$.

We first analyze the case that $\Delta(\dot{\alpha}, \dot{\beta}, \theta) = 0$. It corresponds to the case that the dominant plane is tangential to the candidate region edge. The original $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ is a quartic polynomial with respect to $\cos \theta$ and $\sin \theta$. We use the power reduction [7] to simplify it as

$$\Delta(\dot{\alpha}, \dot{\beta}, \theta) = A \cdot \cos(2\theta) + B \cdot \cos(4\theta) + C \cdot \sin(2\theta) + D \cdot \sin(4\theta) + E, \quad (7)$$

where the known coefficients $\{A, B, C, D, E\}$ are computed by $\dot{\alpha}$ and $\dot{\beta}$. Then we substitute $\cos(4\theta) = 2\cos^2(2\theta) - 1$ and $\sin(4\theta) = 2\sin(2\theta)\cos(2\theta)$ into Eq. (7) to transform $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ as a polynomial with respect to only $\cos(2\theta)$ and $\sin(2\theta)$. Finally, we use Weierstrass substitution [7], i.e., $\cos(2\theta) = \frac{1 - \tan^2 \theta}{1 + \tan^2 \theta}$ and $\sin(2\theta) = \frac{2 \tan \theta}{1 + \tan^2 \theta}$ to simplify $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ as

$$\Delta(\dot{\alpha}, \dot{\beta}, \theta) = a \cdot \tan^4 \theta + b \cdot \tan^3 \theta + c \cdot \tan^2 \theta + d \cdot \tan \theta + e. \quad (8)$$

$\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ in Eq. (8) is a quartic polynomial with respect to $\tan \theta$, and its known coefficients $\{a, b, c, d, e\}$ are computed by $\dot{\alpha}$ and $\dot{\beta}$.

We solve the real root $\tan \theta$ of the polynomial $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ in Eq. (8) by SVD [15] and then obtain the zero $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Note that θ has two solutions $\{\theta^l, \theta^r\}$ that both correspond to the case of tangency (see Fig. 2(c)). Given $\{\theta^l, \theta^r\}$, we aim to find the candidate interval corresponding to the case that $\Delta(\dot{\alpha}, \dot{\beta}, \theta) > 0$. As shown in Fig. 3(a), we compute the midpoints θ^m of θ^l and θ^r . If $\Delta(\dot{\alpha}, \dot{\beta}, \theta^m) > 0$, we treat $[\theta^l, \theta^r]$ as the candidate interval. If $\Delta(\dot{\alpha}, \dot{\beta}, \theta^m) < 0$, we treat $[-\frac{\pi}{2}, \theta^l] \cup [\theta^r, \frac{\pi}{2}]$ as the candidate interval. Our candidate interval computation leads to $\mathcal{O}(K)$ complexity.

4.3 Stabbing Candidate Intervals by Probes

Given K candidate intervals mined above, we aim to find a set of probes, each of which stabs as many intervals as possible (i.e., maximizes the number of horizontal inliers). Note that we only consider the probe stabbing more than τ intervals (we compute the adaptive τ following [33]). The reason is that some outliers may coincidentally generate a small number of mutually overlapping intervals, which

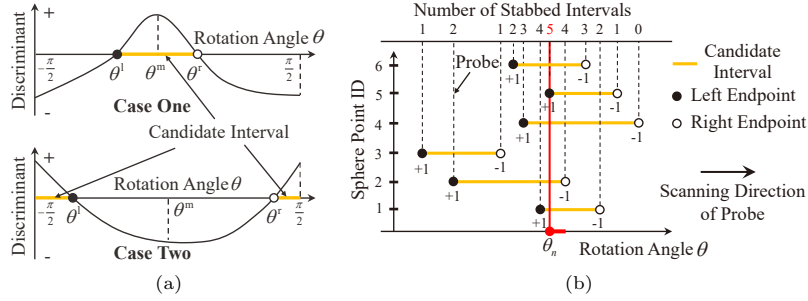


Fig. 3. (a) Given the polynomial roots θ^l and θ^r , we find the candidate interval corresponding to the positive discriminant. (b) We sequentially scan each probe passing through an endpoint of candidate interval.

results in a set of pseudo-horizontal inliers. First, we sort all the interval endpoints in ascending order by the merge sort algorithm [6] whose complexity is $\mathcal{O}(K \log K)$. Then as shown in Fig. 3(b), we define the probes located at each endpoint and then sequentially scan these probes. If we scan a probe that passes through a left/right endpoint, we increase/decrease the number of the stabbed intervals by 1. Our probe scanning leads to $\mathcal{O}(K)$ complexity. In a small region of θ enclosed by two adjacent endpoints (see the red region in Fig. 3(b)), different values of θ correspond to the same number of the stabbed intervals. Without loss of generality, we treat the probe passing through the left endpoint of this region (see the red probe in Fig. 3(b)) as the representative.

After scanning, each probe is associated with the number of the stabbed intervals. We save a probe if its associated number is higher than the numbers of its two neighbors and also higher than the above threshold τ (see the red probe in Fig. 3(b) where $\tau = 3$). We treat the positions of N saved probes as the estimated angles $\{\theta_n\}_{n=1}^N$ and use them to compute the horizontal DDs by Eq. (2). We treat each set of sphere points, whose candidate intervals are stabbed by a saved probe, as a set of horizontal inliers. Therefore, our inner module can automatically determine the number N of the horizontal DDs and also maximizes the cardinality of each horizontal inlier set. The above candidate interval mining, endpoint sorting and probe scanning lead to the total complexity of $\mathcal{O}(K \log K)$. Our inner module can thus run in polynomial time.

5 Practical Case with Perturbation

We extend the above section to the practical case that BnB generates the wide-to-narrow sub-intervals of the angles α and β . Accordingly, the vertical DD \mathbf{v} in Eq. (1) and the coefficients $\{a_i, b_i\}_{i=1}^3$ of the horizontal DD \mathbf{h} in Eq. (2) are perturbed. Given a pair of sub-intervals $[\alpha]$ and $[\beta]$ of the angles α and β , we aim to compute the bounds (instead of exact value) of the number of identified inliers. Note that the exact values $\hat{\alpha}$ and $\hat{\beta}$ in the above section can be treated as the midpoints of the sub-intervals $[\alpha]$ and $[\beta]$.

5.1 Bounds of Number of Inliers

Vertical Inliers. We extend the non-perturbed vertical dominant plane $\pi'(\dot{\alpha}, \dot{\beta})$ in Eq. (4) to the perturbed vertical dominant plane $\pi'([\alpha], [\beta])$. If $\pi'([\alpha], [\beta])$ intersects with the candidate region of the sphere point \mathbf{p}_k , we treat \mathbf{p}_k as a vertical inlier. Mathematically, we follow Section 4.2 to define a system based on Eqs. (4) and (5), and further compute the discriminant $\Delta([\alpha], [\beta])$. Then we employ the interval analysis [27] to compute the range of $\Delta([\alpha], [\beta])$ and denote it by $[\Delta]$. If $[\Delta] > 0$, we treat the sphere point \mathbf{p}_k as an inlier. We increase both lower and upper bounds of the number of vertical inliers by 1. If $[\Delta] \leq 0 \leq \overline{[\Delta]}$, we cannot make sure whether \mathbf{p}_k is an inlier. We only increase the upper bound of the number of vertical inliers by 1. If $\overline{[\Delta]} < 0$, we treat \mathbf{p}_k as an outlier. Our outer module provides $\mathcal{O}(K)$ complexity.

Horizontal Inliers. We follow Sections 4.1 and 4.2 to use the midpoints $\dot{\alpha}$ and $\dot{\beta}$ of the sub-intervals $[\alpha]$ and $[\beta]$ to generate a polynomial $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ and compute its zeros θ^l and θ^r . In addition, we extend this non-perturbed polynomial to the perturbed polynomial $\Delta([\alpha], [\beta], \theta)$. Fig. 4(a-left) shows that $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ is within the “buffer”, i.e., perturbation range of $\Delta([\alpha], [\beta], \theta)$. Mathematically, the non-perturbed coefficients of $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ in Eq. (8) are with respect to $\dot{\alpha}$ and $\dot{\beta}$, while the perturbed coefficients of $\Delta([\alpha], [\beta], \theta)$ are with respect to $[\alpha]$ and $[\beta]$. We employ the above interval analysis to compute the ranges of these perturbed coefficients. Accordingly, we extend the non-perturbed zeros θ^l and θ^r of $\Delta(\dot{\alpha}, \dot{\beta}, \theta)$ to the perturbed zeros $[\theta^l]$ and $[\theta^r]$ of $\Delta([\alpha], [\beta], \theta)$. We leverage the polynomial perturbation theory [11] to compute $[\theta^l]$ and $[\theta^r]$.

Based on the above perturbed zeros $[\theta^l]$ and $[\theta^r]$, we extend the non-perturbed candidate intervals in Section 4.2 to the perturbed candidate intervals. As shown in Fig. 4(a-right), we define the “middle-sized” candidate interval as $[\theta^l, \theta^r]$, and define the “widest” candidate interval as $[[\theta^l], [\theta^r]]$. The middle-sized candidate interval is a subset of the widest candidate interval. Then we follow Section 4.3 to find two sets of probes stabbing these middle-sized and widest candidate intervals, respectively. As shown in Fig. 4(b), if a set of probes stabs at most w_1 middle-sized candidate intervals, we can find another set of probes stabbing at most w_2 ($w_2 \geq w_1$) widest candidate intervals. We treat w_1 and w_2 as the lower and upper bounds of the number of horizontal inliers, respectively.

5.2 Collaboration between BnB and MnS

As shown in Fig. 1(b), given a pair of sub-intervals $[\alpha]$ and $[\beta]$, 1) our BnB-based outer module computes the bounds of the number of vertical inliers, and 2) our MnS-based inner module computes the bounds of the number of horizontal inliers and returns them to our outer module. Our outer module adds these bounds to obtain the bounds of the total number of inliers. We discard a pair of sub-intervals (see blue bins in Fig. 1(b-outer)) if its associated upper bound is smaller than the lower bound associated with another pair of sub-intervals. At convergence, we obtain the optimal pair of (narrow) sub-intervals $[\hat{\alpha}]$ and $[\hat{\beta}]$. For $[\hat{\alpha}]$ and $[\hat{\beta}]$, we 1) use their midpoints to compute the optimal vertical DD by

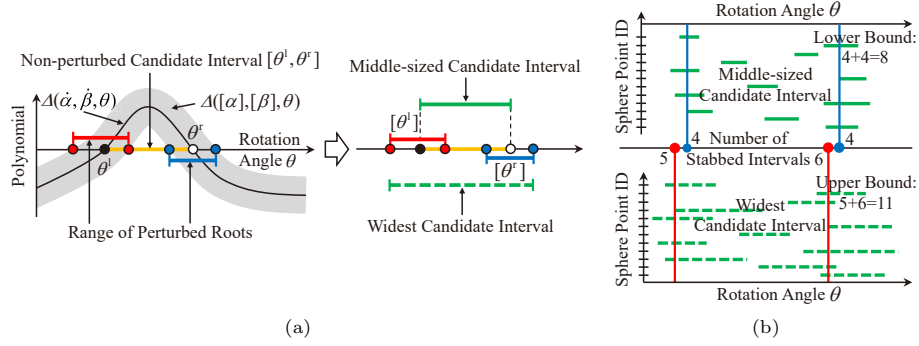


Fig. 4. (a) Left: the perturbed polynomial $\Delta([\alpha], [\beta], \theta)$ leads to the perturbed zeros $[\theta^l]$ and $[\theta^r]$. Right: we use these perturbed zeros to define the perturbed (middle-sized and widest) candidate intervals. (b) We use these candidate intervals to compute the lower and upper bounds of the number of horizontal inliers.

Eq. (1), and 2) use their midpoints and corresponding N optimal angles $\{\theta_n\}_{n=1}^N$ to compute the optimal horizontal DDs by Eq. (2). In addition, to speed up our search, we leverage Hough transform [29] and the orthogonality enforcement method [34] to estimate a sub-optimal DD set. We discard a large number of sub-intervals (see gray bins in Fig. 1(b-outer)) whose upper bounds are smaller than the number of inliers identified by this sub-optimal DD set.

Given K image lines, our complexity is $\mathcal{O}(K \log K)$ for a pair of sub-intervals $[\alpha]$ and $[\beta]$. Our method evaluates 2^2 pairs of sub-intervals at an iteration. It processes totally $I \cdot 2^2$ pairs of sub-intervals where I denotes its number of iterations, leading to $\mathcal{O}(I \cdot 2^2 \cdot K \log K)$ complexity. In contrast, the state-of-the-art pure BnB-based approach [17] provides $\mathcal{O}(K)$ complexity for a list of sub-intervals. It processes totally $I' \cdot 2^{2+N}$ lists of sub-intervals where I' denotes its number of iterations and N denotes the number of horizontal DDs, leading to $\mathcal{O}(I' \cdot 2^{2+N} \cdot K)$ complexity. Experiments show that our method is significantly faster than [17]. The reasons are 1) typically, $\log K < 2^N$ (our complexity only depends on the number of lines K but not the number of DDs N), 2) $I < I'$ (our branched space has lower dimension and redundancy), and 3) determining N by [17] is inefficient.

6 Experiments

We compare the state-of-the-art approaches with our methods:

- The Hough transform-based approach [29] (denoted by **Hough**);
- The T-Linkage-based approach [20] (denoted by **T-Linkage**);
- The BnB-based approach [17] (denoted by **BnB**);
- The integration of our outer and inner modules (denoted by **OnI**);
- Our inner module using the ground truth vertical DD (denoted by **Inner**).

All these methods are implemented in MATLAB and tested on a computer equipped with an Intel Core i7 3.2 GHz CPU and 8GB RAM.

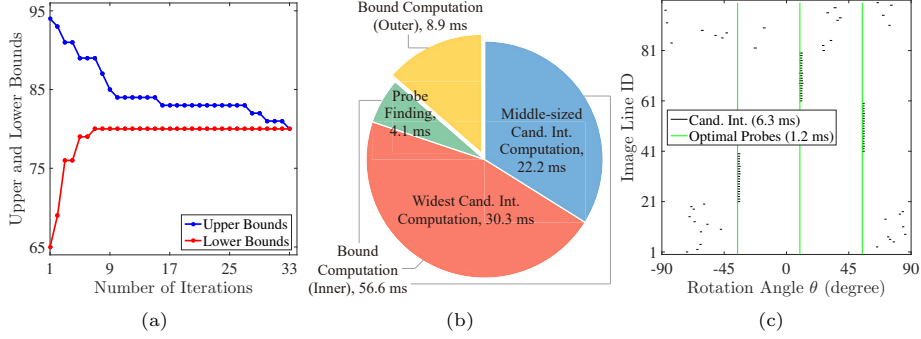


Fig. 5. Representative tests on synthetic data. (a) Evolutions of the highest upper and lower bounds of our **OnI**. (b) Time distribution of our **OnI** at an iteration (processing four pairs of sub-intervals). (c) Candidate intervals and probes of our **Inner**.

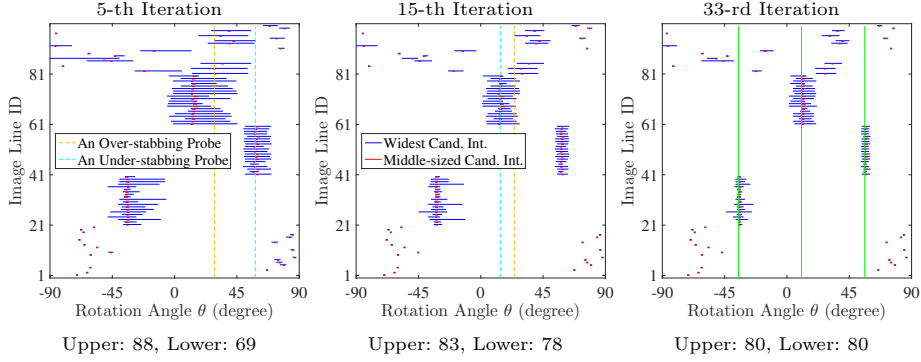


Fig. 6. Representative iterations of Fig. 5(a). Given a pair of sub-intervals of the angles α and β , we mine the widest and middle-sized candidate intervals. The numbers below each image denote the upper and lower bounds of the total number of inliers.

We follow [23, 25] to evaluate the accuracy of image line clustering in terms of precision and recall, and evaluate the VP accuracy in terms of root mean square of the consistency error. Specifically, $\text{precision} = \frac{C}{C+W}$ and $\text{recall} = \frac{C}{C+M}$ where C , W , and M denote the numbers of the correctly identified, wrongly identified, and missing inliers, respectively. We also compute the F_1 -score $= \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$. The consistency error represents the distance from an endpoint of the image line \mathbf{l} to a virtual line defined by the midpoint of \mathbf{l} and an estimated VP.

6.1 Synthetic Dataset

We synthesize several 3D lines aligned to a vertical DD and N ($N \geq 3$) horizontal DDs, and project them to the image to generate inlier lines. We perturb the endpoints of these inlier lines by a zero-mean Gaussian noise. We generate outlier lines by randomizing their endpoints within the image. In the following, we first

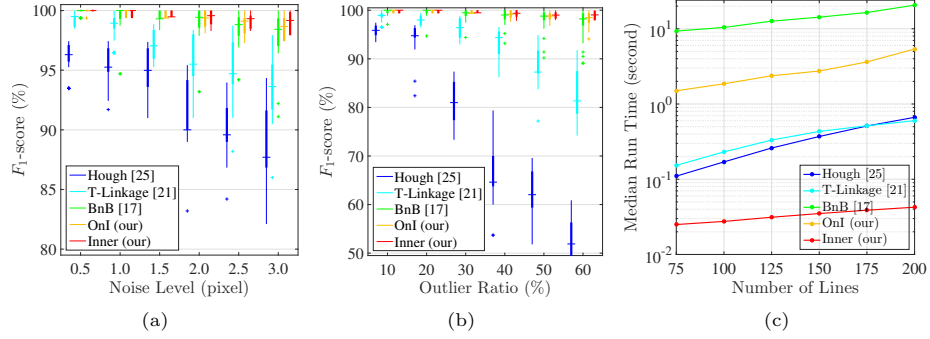


Fig. 7. Comparisons on synthetic datasets (the number of horizontal VPs is 4). (a) Accuracy test with respect to the noise level. (b) Accuracy test with respect to the outlier ratio. (c) Efficiency test with respect to the number of lines.

report some representative tests of our **OnI** and **Inner**. Then we compare our **OnI** and **Inner** with state-of-the-art approaches.

Representative tests. We synthesize 100 image lines. The 1-st to 20-th lines are vertical inliers. The 21-st to 80-th lines are horizontal inliers associated with 3 VPs. The 81-st to 100-th lines are outliers. Fig. 5(a) shows the evolutions of the highest upper and lower bounds of our **OnI**. They converge to the number of inliers 80. We will analyze some representative iterations in the next paragraph. As shown in Fig. 5(b), our inner module is more time-consuming than our outer module due to the candidate interval mining. In addition, our probe finding is efficient thanks to its low computational complexity. As shown in Fig. 5(c), our **Inner** identifies all the 60 horizontal inliers and achieves real-time efficiency.

Fig. 6 shows some representative iterations. Given a pair of sub-intervals of the angles α and β , our **OnI** mines the widest and middle-sized candidate intervals. At the 5-th iteration, the sub-intervals are wide since the space of α and β has not been fully branched. Accordingly, the widest candidate intervals of a set of horizontal inliers and some outliers overlap with each other, leading to an over-stabbing probe and loose upper bound. Moreover, the sub-intervals are not accurate, i.e., they do not contain the ground truth values of α and β . Accordingly, the middle-sized candidate intervals of a set of horizontal inliers deviate from each other, leading to an under-stabbing probe and loose lower bound. At the 15-th iteration, the sub-intervals become narrower. The number of the over-stabbed candidate intervals decreases and thus the upper bound decreases. Moreover, the sub-intervals become more accurate. The number of the under-stabbed candidate intervals decreases and thus the lower bound increases. At the 33-rd iteration, the highest upper and lower bounds both equal to the number of inliers 80, which satisfies our stopping criterion.

Accuracy comparisons. Fig. 7(a) shows the tests with respect to the noise level. We fix the number of lines and outlier ratio to 100 and 20% respectively, and vary the standard deviation of noise from 0.5 to 3 pixels. Fig. 7(b) shows the tests with respect to the outlier ratio. We fix the number of lines and noise














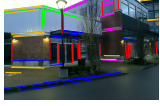
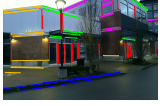
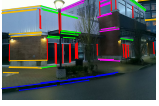
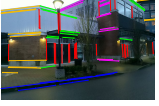







Lines	Hough [29]	T-Linkage [20]	BnB [17]	OnI (our)	Inner (our)
					
Our Dataset 117 lines, 4 VPs	91.74%, 92.59% 1.61 pix., 0.26 sec.	97.35%, 96.49% 0.82 pix., 0.37 sec.	100%, 100% 0.44 pix., 12.85 sec.	100%, 100% 0.45 pix., 3.06 sec.	100%, 100% 0.39 pix., 0.04 sec.
					
NYU [31] 79 lines, 4 VPs	93.42%, 95.95% 1.98 pix., 0.17 sec.	96.15%, 98.68% 1.09 pix., 0.25 sec.	100%, 100% 0.59 pix., 9.92 sec.	100%, 100% 0.52 pix., 2.15 sec.	100%, 100% 0.47 pix., 0.03 sec.
					
Our Dataset 84 lines, 5 VPs	80.76%, 94.02% 5.63 pix., 0.20 sec.	96.15%, 94.93% 1.46 pix., 0.28 sec.	100%, 98.78% 0.78 pix., 11.09 sec.	100%, 98.78% 0.75 pix., 2.48 sec.	100%, 100% 0.61 pix., 0.03 sec.
					
NYU [31] 86 lines, 4 VPs	82.43%, 84.72% 9.03 pix., 0.22 sec.	97.53%, 95.18% 1.38 pix., 0.30 sec.	100%, 98.73% 0.60 pix., 13.10 sec.	100%, 98.73% 0.62 pix., 2.73 sec.	100%, 97.46% 0.55 pix., 0.04 sec.

Fig. 8. Representative comparisons on our and NYU [31] datasets. The first two rows: the manually extracted lines. The last two rows: the lines extracted by LSD [14]. The numbers below image represent the precision, recall, consistency error and run time.

level to 200 and 1 pixel respectively, and vary the outlier ratio from 10% to 60%. Each test is composed of 500 independent trials. **Hough** is sensitive to noise and outliers. **T-Linkage** is only robust under low outlier ratios and its accuracy is prone to being affected by noise since it fails to enforce the orthogonality constraint. **BnB** can handle high noise levels and outlier ratios in most cases. However, its accuracy is affected by some trials without convergence. In contrast, our **OnI** and **Inner** provide high robustness and accuracy. The reason why their F_1 -scores are slightly smaller than 100% is that some lines perturbed by great noise result in the inlier missing and/or cluster ambiguity problems [3].

Efficiency comparisons. Fig. 7(c) shows the test with respect to the number of lines. We fix the noise level and outlier ratio to 1 pixel and 20% respectively, and vary the number of lines from 75 to 200. As the number of lines increases, **Hough** computes a larger number of intersections, and thus its run time increases. The time variation of **T-Linkage** is relatively small due to a fixed number of samplings. The efficiencies of **BnB** and our **OnI** decrease due to more time-consuming bound computation. Our **OnI** is significantly faster than **BnB** since its computational complexity is lower (see Section 5.2), and also it does not require an inefficient pre-processing step to determine the number of VPs. Our **Inner** provides the highest efficiency thanks to our fast probe finding.

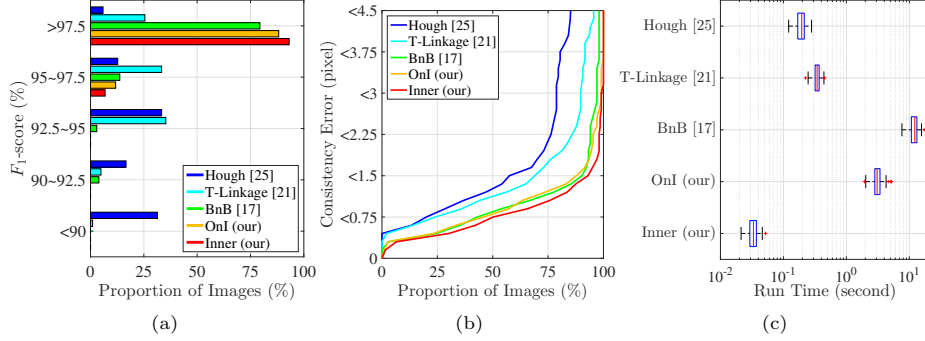


Fig. 9. Comparisons on all the images of our and NYU [31] datasets (using the manually extracted lines). (a) F_1 -score of image line clustering. (b) Culminate histogram of the consistency error. (c) Time distribution of processing a single image.

6.2 Real-world Dataset

We establish an image dataset. It consists of several images satisfying the Atlanta world assumption. We manually extract the lines and assign them with the ground truth cluster labels. We also provide the ground truth VPs. In addition, we select some images satisfying the Atlanta world assumption from the NYU dataset [31]. We manually extract and label the lines. The ground truth VPs are provided by [13]. We also use LSD [14] to automatically extract the lines.

Fig. 8 shows some representative comparisons, and Fig. 9 reports the results on all the images. **Hough** leads to satisfactory efficiency but the lowest accuracy. **T-Linkage** sacrifices partial efficiency to improve its accuracy. **BnB** provides high accuracy but low efficiency. Moreover, it fails to converge on a small number of images, and the best-so-far solution is not accurate enough. In contrast, our **OnI** converges robustly and its accuracy and efficiency are higher than **BnB**. Note that some lines perturbed by great noise slightly affect the overall accuracy of **BnB** and our **OnI**. Our **Inner** exploits the ground truth vertical DD to reduce the effect of noise and search space, achieving the highest accuracy and efficiency.

7 Conclusions

We propose the novel MnS and embed it in BnB to estimate VPs in Atlanta world. Our method efficiently achieves global optimality in terms of maximizing the number of inliers. Moreover, it can automatically determine the number of horizontal VPs. Experiments on synthetic and real-world datasets showed that our method outperforms state-of-the-art approaches in terms of accuracy and/or efficiency.

Acknowledgments. This work is supported in part by the Natural Science Foundation of China under Grant U1613218, in part by the Hong Kong ITC under Grant ITS/448/16FP and Hong Kong Centre for Logistics Robotics, and in part by the VC Fund 4930745 of the CUHK T Stone Robotics Institute.

References

1. Antunes, M., Barreto, J.P.: A global approach for the detection of vanishing points and mutually orthogonal vanishing directions. In: CVPR (2013)
2. Antunes, M., Barreto, J.P., Aouada, D., Ottersten, B.: Unsupervised vanishing point detection and camera calibration from a single Manhattan image with radial distortion. In: CVPR (2017)
3. Bazin, J.C., Demonceaux, C., Vasseur, P., Kweon, I.: Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. IJRR (2012)
4. Bazin, J.C., Seo, Y., Demonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., Pollefeys, M.: Globally optimal line clustering and vanishing point estimation in Manhattan world. In: CVPR (2012)
5. Bazin, J.C., Seo, Y., Pollefeys, M.: Globally optimal consensus set maximization through rotation search. In: ACCV (2012)
6. Berg, M., Cheong, O., Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer, third edn. (2010)
7. Beyer, W.: CRC Standard Mathematical Tables. CRC Press (1987)
8. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
9. Coughlan, J., Yuille, A.: Manhattan world: Compass direction from a single image by Bayesian inference. In: ICCV (1999)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM (1981)
11. Galántai, A., Hegedus, C.J.: Perturbation bounds for polynomials. Numerische Mathematik (2008)
12. Gao, Y., Yuille, A.L.: Exploiting symmetry and/or Manhattan properties for 3D object structure estimation from single and multiple images. In: CVPR (2017)
13. Ghanem, B., Thabet, A., Niebles, J.C., Heilbron, F.C.: Robust Manhattan frame estimation from a single RGB-D image. In: CVPR (2015)
14. Grompone von Gioi, R., Jakubowicz, J., Morel, J., Randall, G.: LSD: A fast line segment detector with a false detection control. TPAMI (2010)
15. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, second edn. (2003)
16. Joo, K., Oh, T., Kweon, I.S., Bazin, J.C.: Globally optimal inlier set maximization for Atlanta frame estimation. In: CVPR (2018)
17. Joo, K., Oh, T., Kweon, I.S., Bazin, J.C.: Globally optimal inlier set maximization for Atlanta world understanding. TPAMI (2019)
18. Kim, P., Coltin, B., Kim, H.J.: Low-drift visual odometry in structured environments by decoupling rotational and translational motion. In: ICRA (2018)
19. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic upright adjustment of photographs with robust camera calibration. TPAMI (2014)
20. Li, H., Xing, Y., Zhao, J., Bazin, J.C., Liu, Z., Liu, Y.H.: Leveraging structural regularity of Atlanta world for monocular SLAM. In: ICRA (2019)
21. Li, H., Yao, J., Bazin, J.C., Lu, X., Xing, Y., Liu, K.: A monocular SLAM system leveraging structural regularity in Manhattan world. In: ICRA (2018)
22. Li, H., Zhao, J., Bazin, J.C., Chen, W., Chen, K., Liu, Y.H.: Line-based absolute and relative camera pose estimation in structured environments. In: IROS (2019)
23. Li, H., Zhao, J., Bazin, J.C., Chen, W., Liu, Z., Liu, Y.H.: Quasi-globally optimal and efficient vanishing point estimation in Manhattan world. In: ICCV (2019)

24. Li, H., Zhao, J., Bazin, J.C., Liu, Y.H.: Robust estimation of absolute camera pose via intersection constraint and flow consensus. *TIP* (2020)
25. Lu, X., Yao, J., Li, H., Liu, Y.: 2-line exhaustive searching for real-time vanishing point estimation in Manhattan world. In: *WACV* (2017)
26. Magri, L., Fusiello, A.: T-Linkage: A continuous relaxation of J-Linkage for multi-model fitting. In: *CVPR* (2014)
27. Moore, R.E., Kearfott, R.B., Cloud, M.J.: *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics (2009)
28. Pham, T.T., Chin, T., Schindler, K., Suter, D.: Interacting geometric priors for robust multimodel fitting. *TIP* (2014)
29. Quan, L., Mohr, R.: Determining perspective structures using hierarchical Hough transform. *PRL* (1989)
30. Schindler, G., Dellaert, F.: Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In: *CVPR* (2004)
31. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: *ECCV* (2012)
32. Sinha, S., Steedly, D., Szeliski, R., Agrawala, M., Pollefeys, M.: Interactive 3D architectural modeling from unordered photo collections. In: *SIGGRAPH Asia* (2008)
33. Stewart, C.V.: MINPRAN: a new robust estimator for computer vision. *TPAMI* (1995)
34. Straub, J., Freifeld, O., Rosman, G., Leonard, J.J., Fisher, J.W.: The Manhattan frame model—Manhattan world inference in the space of surface normals. *TPAMI* (2017)
35. Tardif, J.P.: Non-iterative approach for fast and accurate vanishing point detection. In: *ICCV* (2009)
36. Toldo, R., Fusiello, A.: Robust multiple structures estimation with J-Linkage. In: *ECCV* (2008)
37. Zhou, S., Zhao, H., Chen, W., Miao, Z., Liu, Z., Wang, H., Liu, Y.H.: Robust path following of the tractor-trailers system in GPS-denied environments. *RAL* (2020)
38. Zuliani, M., Kenney, C.S., Manjunath, B.S.: The multiRANSAC algorithm and its application to detect planar homographies. In: *ICIP* (2005)