Hong Kong World: Leveraging Structural Regularity for Line-based SLAM

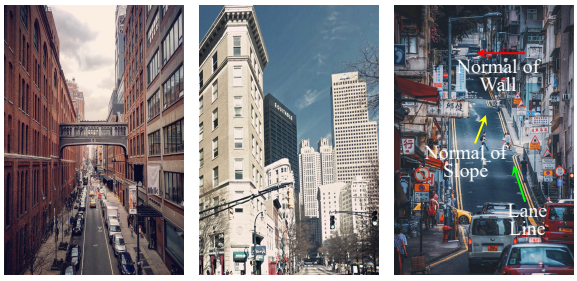
Haoang Li, Ji Zhao, Jean-Charles Bazin, Pyojin Kim, Kyungdon Joo, Zhenjun Zhao, and Yun-Hui Liu

Abstract—Manhattan and Atlanta worlds hold for the structured scenes with only vertical and horizontal dominant directions (DDs). To describe the scenes with additional sloping DDs, a mixture of independent Manhattan worlds seems plausible, but may lead to unaligned and unrelated DDs. By contrast, we propose a novel structural model called Hong Kong world. It is more general than Manhattan and Atlanta worlds since it can represent the environments with slopes, e.g., a city with hilly terrain, a house with sloping roof, and a loft apartment with staircase. Moreover, it is more compact and accurate than a mixture of independent Manhattan worlds by enforcing the orthogonality constraints between not only vertical and horizontal DDs, but also horizontal and sloping DDs. We further leverage the structural regularity of Hong Kong world for the line-based SLAM. Our SLAM method is reliable thanks to three technical novelties. First, we estimate DDs/vanishing points in Hong Kong world in a semi-searching way. We use a new consensus voting strategy for search, instead of traditional branch and bound. This method is the first one that can simultaneously determine the number of DDs, and achieve quasi-global optimality in terms of the number of inliers. Second, we compute the camera pose by exploiting the spatial relations between DDs in Hong Kong world. This method generates concise polynomials, and thus is more accurate and efficient than existing approaches designed for unstructured scenes. Third, we refine the estimated DDs in Hong Kong world by a novel filter-based method. Then we use these refined DDs to optimize the camera poses and 3D lines, leading to higher accuracy and robustness than existing optimization algorithms. In addition, we establish the first dataset of sequential images in Hong Kong world. Experiments showed that our approach outperforms state-of-the-art methods in terms of accuracy and/or efficiency.

Index Terms—Structural Regularity, Line, SLAM, Dominant Direction, Vanishing Point, Camera Pose.

1 Introduction

Structured environments have been well studied in computer vision and robotics fields [1], [2], [3]. As shown in Figs. 1(a) and 2(a), Manhattan world [1] holds for the scenes with two horizontal dominant directions (DDs) and a vertical DD. These DDs are mutually orthogonal. We can model Manhattan world by a frame whose axes correspond to DDs. Figures 1(b) and 2(b) show that Atlanta world [2] describes the scenes with multiple horizontal DDs and a vertical DD. Horizontal DDs are unnecessarily orthogonal to each other but all orthogonal to the vertical DD. We can model Atlanta world by a set of frames sharing a common vertical axis. The main shortcoming of Manhattan and Atlanta worlds is limited generality. They can only describe the structured scenes on the flat lands, but not the structured environments with slopes (see Fig. 1(c)). A straightforward solution is a mixture of Manhattan worlds [3] (see Fig. 2(c)). This model assumes that all the Manhattan worlds are independent. However, in practice, some Manhattan worlds share a common DD. When expressing such type of scenes, this model may result in unrelated and unaligned DDs. Thus, the main limitation of a mixture of Manhattan worlds lies in unsatisfactory compactness and accuracy. To solve the above problems, in this paper, we propose a novel structural model called *Hong Kong world*.



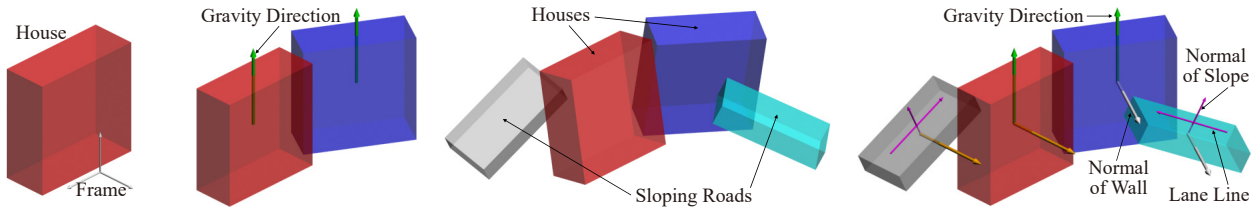
(a) Manhattan (b) Atlanta (c) Hong Kong

Fig. 1. Representative cities exhibiting various structural regularities. (a) Manhattan with a vertical DD and two horizontal DDs. (b) Atlanta with a vertical DD and multiple horizontal DDs. (c) Hong Kong with a vertical DD, multiple horizontal DDs (see red arrow), and multiple sloping DDs (see yellow and green arrows). Red, yellow, and green arrows are mutually orthogonal.

As shown in Figs. 1(c) and 2(d), our Hong Kong world consists of a vertical DD, multiple horizontal DDs, and multiple sloping DDs.[1] These DDs satisfy three constraints. First, the vertical DD is orthogonal to all the horizontal DDs. For example, houses are built along the gravity direction (see {red, blue} blocks in Fig. 2(d)). Second, a horizontal DD

- *H. Li is with the Department of Informatics, Technical University of Munich, Munich, Germany. Email: haoang.li@tum.de*
- *J. Zhao is in Beijing, China. Email: zhaoji84@gmail.com*
- *J.-C. Bazin is with the School of Electrical Engineering and the Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Email: bazinjc@kaist.ac.kr*
- *P. Kim is with the Department of Mechanical Systems Engineering, Sookmyung Women's University, Seoul, South Korea. Email: pjinkim@sookmyung.ac.kr*
- *K. Joo is with the Artificial Intelligence Graduate School and the Department of Computer Science, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea. Email: kdjoo369@gmail.com*
- *Z. Zhao and Y.-H. Liu are with the Department of Mechanical and Automation Engineering and CUHK T Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong. Email: {zjzhao, yhliu}@mae.cuhk.edu.hk*
- *(Corresponding authors: Yun-Hui Liu and Kyungdon Joo.)*

1. In our context, we do not differentiate between the positive and negative orientations of DDs, following [4], [5].



(a) Manhattan World (b) Atlanta World (c) A Mixture of Manhattan Worlds (d) Hong Kong World

Fig. 2. Illustration of various structural models. (a) Manhattan world [1] corresponds to a single block or frame. (b) Atlanta world [2] corresponds to multiple blocks sharing a common vertical DD, e.g., gravity direction. (c) A mixture of independent Manhattan worlds [3] corresponds to multiple unaligned and unrelated blocks. (d) In our Hong Kong world, {red, blue} blocks share a common vertical DD, e.g., gravity direction. {Blue, cyan} or {red, gray} blocks share a common horizontal DD, e.g., a normal of wall (see Fig. 1(c)).

is orthogonal to a set of sloping DDs. For example, a wall of house adjoins a sloping road (see Fig. 1(c)). Accordingly, the normal of this wall that corresponds to a horizontal DD is orthogonal to both lane line and normal of slope that correspond to two sloping DDs (see {blue, cyan} or {red, gray} blocks in Fig. 2(d)). Note that there may be more than one horizontal DDs (see yellow and gray arrows in Fig. 2(d)), each of which is orthogonal to a set of sloping DDs. Third, several pairs of horizontal or sloping DDs are orthogonal. For example, normals of two orthogonal walls that correspond to two horizontal DDs are orthogonal; Lane line and normal of slope that correspond to two sloping DDs are orthogonal (see magenta arrows in Fig. 2(d)).

Based on the above DD constraints, we model our Hong Kong world by $(1+N)$ sets of frames. The first set of frames share a common vertical axis (see {red, blue} blocks in Fig. 2(d)). The n-th $(2 \leqslant n \leqslant 1+N)$ set of frames share a common horizontal axis (see {blue, cyan} and {red, gray} blocks in Fig. 2(d)). To some extent, our Hong Kong world is a generalization of Atlanta world. Specifically, a single set of frames in Atlanta world share a common vertical axis, while $(1+N)$ sets of frames in Hong Kong world share $(1+N)$ common vertical or horizontal axes. Our Hong Kong world has two main advantages. First, it is more general than Manhattan and Atlanta worlds since it can represent the environments with slopes, e.g., a city with hilly terrain, a house with sloping roof, and a loft apartment with staircase. Second, it is more compact and accurate than a mixture of independent Manhattan worlds since frames are tightly coupled based on the orthogonality constraints.

Our Hong Kong world has a large variety of potential application fields, such as 3D reconstruction [6], scene understanding [7] and robot navigation [8]. In this paper, we focus on applying it to a crucial technology of robot navigation, i.e., simultaneous localization and mapping (SLAM). Existing point-based SLAM methods [9], [10] are unstable in textureless environments. To solve this problem, several line-based methods [11], [12], [13], [14] have been proposed. However, they neglect the spatial relations between 3D lines that can provide effective geometric constraints. Accordingly, their accuracy is unsatisfactory. By contrast, recent line-based methods [15], [16] consider particular spatial relations between 3D lines such as parallelism and orthogonality in structured scenes. While these methods improve the accuracy, they lead to low generality since they are only applicable to Manhattan and Atlanta worlds. To overcome this limitation, we propose a monocular line-based SLAM method by leveraging the structural regularity of Hong Kong world.

Our SLAM method is reliable thanks to three technical novelties. First, we estimate DDs/vanishing points[2] in Hong Kong world in a semi-searching way. We use a new consensus voting strategy for search, instead of traditional branch and bound (BnB) [18], [19]. This method is the first one that can simultaneously determine the number of DDs, and achieve quasi-global optimality in terms of the number of inliers. Second, we compute the camera pose by exploiting the spatial relations between DDs in Hong Kong world. This method generates concise polynomials, and thus is more accurate and efficient than existing approaches designed for unstructured scenes [20], [21], [22]. Third, we refine the estimated DDs in Hong Kong world by a novel filter-based method. Then we use these refined DDs to optimize the camera poses and 3D lines, leading to higher accuracy and robustness than existing optimization algorithms [15], [16], [23]. For experiments, we establish the first dataset of sequential images in Hong Kong world. It is composed of 7 sequences (9077 images). We provide calibration parameters and ground truth trajectories. Experiments showed that our approach outperforms state-of-the-art methods in terms of accuracy and/or efficiency. Our dataset and source code will be publicly available on our project website[3]. Our main contributions are:

- We propose a general, compact and accurate structural model called Hong Kong world.
- We design a quasi-globally optimal and efficient DD estimation method.
- We introduce an accurate and efficient camera pose estimation approach.
- We present a novel DD refinement algorithm, and an accurate and robust SLAM optimization strategy.
- We establish the first dataset of sequential images in Hong Kong world.

2 Related Work

In this section, we review existing line-based SLAM methods. We classify them into three categories in terms of application scenarios, i.e., arbitrary scenes (regardless of structured or unstructured scenes), Manhattan world, and Atlanta world. Most of them consist of two main modules, i.e., front-end for motion estimation and back-end for optimization [24].

2. Vanishing point is the intersection of a set of image lines projected from parallel 3D lines [17]. DDs and vanishing points are equivalent in our context, which will be introduced in Section 3.

3. https://sites.google.com/view/haoangli/projects/hk-slam

2.1 SLAM in Arbitrary Scenes

Front-end. The front-ends of SLAM methods [11], [12], [25], [26] use various camera pose estimation algorithms. A straightforward approach assumes the constant velocity motion of camera [27]. While it is efficient, it becomes unstable when the camera accelerates or turns. Despite different formulations, the line correspondence-based algorithms all process each correspondence independently, i.e., neglect the spatial relations between 3D lines. Accordingly, they provide high generality. However, they generate complex formulas such as high-order polynomial equations [20], [21]. These formulas lead to relatively unsatisfactory numerical stability, which affects the algorithm accuracy. Moreover, solving these formulas is relatively time-consuming.

Back-end. The back-end of earlier SLAM method [11] uses the extended Kalman filter. It is prone to generating significant drift error. By contrast, the back-ends of recent SLAM methods [12], [25], [26] are based on the graph, i.e., exploit bundle adjustment that minimizes the re-projection error of lines. The accuracy of the graph-based methods is typically higher than that of the filter-based approaches [24]. While the back-ends of the above methods are general, they neglect the structural regularity when working in structured scenes. Therefore, there is still room for improvement in accuracy.

2.2 SLAM in Manhattan World

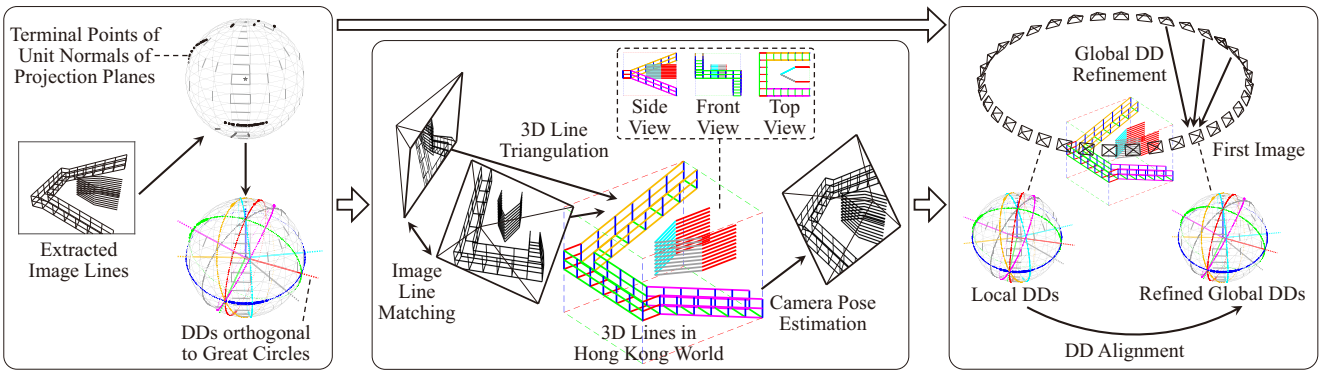
Front-end. The front-ends of SLAM methods [8], [15], [23], [28] fail to effectively leverage the structural regularity. Specifically, their camera pose estimation algorithms neglect the spatial relations between DDs. By contrast, the front-end of a recent SLAM method [29] employs a camera pose estimation algorithm that explicitly considers the orthogonality between DDs. However, this algorithm cannot handle non-orthogonal 3D lines in Atlanta and Hong Kong worlds, leading to relatively low generality.

Back-end. Different from the above front-ends, the back-ends of SLAM methods [8], [15], [23], [28], [29] effectively leverage the structural regularity. This regularity is encoded by DDs/vanishing points. To estimate DDs, SLAM methods [15], [23] exploit the data sampling-based algorithms such as RANSAC [30] and J-Linkage [31]. However, these algorithms are sensitive to noise. A SLAM method [28] uses the parameter search-based algorithm [32] to estimate DDs. While this algorithm provides high accuracy, its efficiency is unsatisfactory, especially when prior information of camera rotation is unavailable. Given the estimated DDs, SLAM methods [15], [23] first estimate the structural lines and then treat them as the landmarks to reduce the error accumulation. The other SLAM methods [8], [28] directly exploit the geometric constraints provided by DDs to optimize the camera pose. While the above methods are reliable in Manhattan world, their generality is low. Specifically, when working in more complex structured scenes, e.g., Atlanta world, they may lead to unsatisfactory accuracy due to insufficient estimated structural lines and DDs. Moreover, they directly use DDs estimated by a single image without refinement and outlier detection. These DDs may be unreliable, which affects the SLAM accuracy and robustness.

2.3 SLAM in Atlanta World

Front-end. The front-ends of SLAM methods [5], [16] fail to effectively leverage the structural regularity. Specifically, their camera pose estimation algorithms do not consider the spatial relations between DDs. This limitation is similar to that of the above SLAM methods in Manhattan world.

Back-end. The back-ends of SLAM methods [5], [16] both use the constraints related to DDs for optimization. Their main difference lies in DD estimation. Recall that in Atlanta world, the number of DDs is unknown and the vertical DD is orthogonal to any horizontal DD, which increases the difficulty of DD estimation. The above SLAM methods fail to fully overcome these challenges. Specifically, Zou et al. [16] first used the inertial measurement unit (IMU) to obtain the vertical DD, and then employed RANSAC to estimate the horizontal DDs. However, this algorithm is prone to missing some DDs since RANSAC may fail to sample inlier image lines associated with these DDs. This algorithm also leads to relatively low generality since it relies on the extra IMU. Li et al. [5] exploited T-Linkage [33] to non-iteratively estimate DDs based on numerous samplings. While this algorithm can robustly retrieve all the DDs, it fails to satisfy the orthogonality constraint between the vertical



(a) Regularity Encoder (b) Front-end (c) Back-end

Fig. 3. Overview of our SLAM framework in Hong Kong world. The 3D scene in this example is analogous to part of a shopping mall with a floor and two staircases. (a) We extract image lines and compute their associated normals of projection planes. Then we use these normals to estimate DDs. (b) We match lines between two images, and use 2D-2D line correspondences to triangulate 3D lines. We match 3D lines aligned to DDs against image lines, and use 3D-2D line correspondences to estimate the camera pose. (c) We recursively refine the global DDs and use these refined DDs to optimize the camera poses and 3D lines.

and horizontal DDs. Moreover, the above algorithms are both sensitive to noise due to the uncertainty of sampling. The other DD estimation algorithms in Atlanta world [2], [4] are not suitable for SLAM due to high computational complexity or requirement of prior knowledge regarding the number of DDs.

Overall, leveraging the structural regularity can improve the SLAM accuracy, but existing structural regularity-based SLAM methods still have some limitations. First, they rely on strict Manhattan or Atlanta world assumption, leading to relatively low generality. Second, the accuracy and/or efficiency of their DD and camera pose estimation algorithms are unsatisfactory. Third, their SLAM optimization algorithms are sensitive to noise and outliers of DDs. By contrast, our SLAM method is applicable to not only Manhattan and Atlanta worlds, but also Hong Kong world. Moreover, our algorithms of both DD and camera pose estimation are accurate and efficient. In addition, our DD-based SLAM optimization is reliable thanks to the proposed refinement and outlier removal of DDs.

3 Overview of Our SLAM Framework

Our SLAM framework in Hong Kong world consists of regularity encoder, front-end, and back-end (see Fig. 3).

Regularity Encoder. We use DDs/vanishing points to encode the structural regularity of Hong Kong world. Vanishing point in a calibrated image[4] is equivalent to DD. Intuitively, a DD is aligned to the 3D direction defined by a vanishing point and the camera center (see green direction in Fig. 4). Mathematically, a vanishing point $\mathbf{c}$ in homogeneous coordinates can be expressed by $\mathbf{c} = \mathbf{K}\mathbf{d}$ where $\mathbf{K}$ is the calibration matrix and $\mathbf{d}$ is a DD [17]. Based on several image lines extracted by LSD [34], we compute the unit normals of projection planes. Let $\mathbf{o}_1$ and $\mathbf{o}_2$ denote the homogeneous coordinates of two endpoints of an image line $\mathbf{l}$. A normal $\mathbf{n}$ of projection plane can be computed by $\mathbf{n} = \hat{\mathbf{o}}_1 \times \hat{\mathbf{o}}_2$ where $\hat{\mathbf{o}}_1 = \mathbf{K}^{-1}\mathbf{o}_1$ and $\hat{\mathbf{o}}_2 = \mathbf{K}^{-1}\mathbf{o}_2$. Then given the computed normals, we cluster them by the unknown-but-sought DDs. We will introduce this DD estimation method in Section 4. Our regularity encoder provides the geometric constraints related to DDs for both front-end and back-end to improve their performances.

Front-end. We match image lines across two views by LBD descriptor [35]. Then we use a 2D-2D line correspondence to triangulate a 3D line. Accordingly, this 3D line is associated with the descriptors of image lines. We match this 3D line against an extracted line in a new image based on their descriptors following [25], [29]. Given a set of 3D-2D line correspondences, we estimate the camera pose by exploiting the spatial relations between DDs estimated by our regularity encoder. We will present this camera pose estimation method in Section 5.

Back-end. We follow conventional SLAM methods [9], [11] to treat the first camera frame as the world frame. The global and local DDs represent DDs in the world and camera frames, respectively. Given frame-by-frame local DDs estimated by our regularity encoder, we propose a filter-based strategy to refine the global DDs. By aligning the local DDs to the refined global DDs, we optimize the camera pose. In addition, we align 3D lines to the global DDs when conducting optimization. We will introduce our global DD refinement and DD-based SLAM optimization in Section 6.

4. In our context, we follow conventional SLAM methods [9], [10], [11] to assume a calibrated camera.

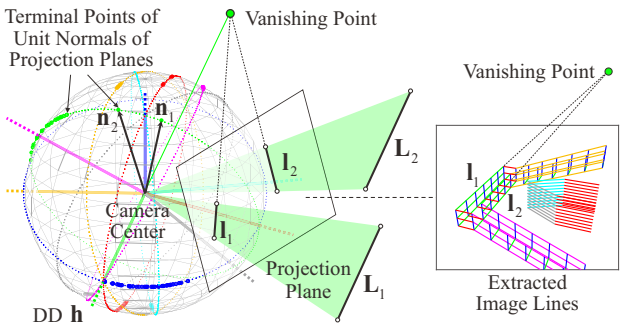


Fig. 4. Illustration of the projective geometry in Hong Kong world. A 3D depiction of this Hong Kong world is shown in Fig. 3(b). Representative 3D lines $\mathbf{L}_1$ and $\mathbf{L}_2$ are both aligned to a horizontal DD $\mathbf{h}$. Their corresponding image lines $\mathbf{l}_1$ and $\mathbf{l}_2$ intersect at a vanishing point.

4 DD Estimation

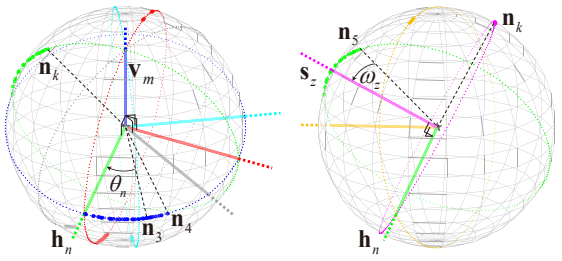
In Hong Kong world, given K image lines $\{\mathbf{l}_k\}_{k=1}^K$, we aim to estimate the vertical, horizontal, and sloping DDs. Note that we estimate DDs instead of vanishing points in the image for two main reasons. First, the image plane is unbounded, i.e., a vanishing point may be very far from the image center. By contrast, the unit sphere is a bounded space, which facilitates the parameter search. Second, on the unit sphere, we can explicitly enforce the orthogonality constraint between DDs.

As shown in Fig. 4, we compute the unit normals of projection planes $\{\mathbf{n}_k\}_{k=1}^K$ (hereinafter we call them "normals") that are associated with the input image lines. We treat a normal $\mathbf{n}_k$ orthogonal to a vertical, horizontal or sloping DD as a vertical, horizontal or sloping inlier, respectively. We consider a normal that is not orthogonal to any DD as an outlier. Note that for a set of inliers with respect to the same DD, they are all orthogonal to this DD (see horizontal inliers $\{\mathbf{n}_1, \mathbf{n}_2\}$ and horizontal DD $\mathbf{h}$ in Fig. 4).

4.1 Sequential DD Estimation

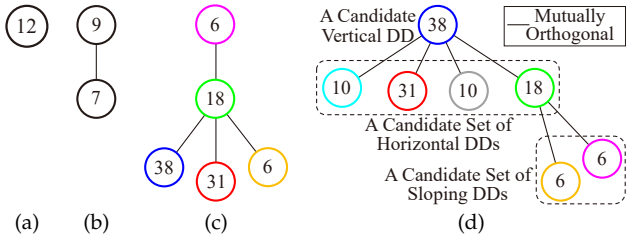
We propose to sequentially estimate the vertical, horizontal, and sloping DDs in Hong Kong world. We first use sampling or IMU to estimate the vertical DD, followed by searching for the horizontal and sloping DDs. We call this strategy the semi-searching strategy. Our approach overcomes the limitations of existing DD estimation methods introduced in Section 2. Specifically, it can simultaneously determine the number of DDs, and achieve high accuracy and efficiency.

Vertical DD. Figure 5(a) shows that we can compute a unit vertical DD $\mathbf{v}$ using any two vertical inliers $\mathbf{n}_3$ and $\mathbf{n}_4$, i.e., $\mathbf{v} = (\mathbf{n}_3 \times \mathbf{n}_4)/\|\mathbf{n}_3 \times \mathbf{n}_4\|$. However, in practice, we do not have prior knowledge regarding which two normals are vertical inliers. To solve this problem, we employ RANSAC [30]. Specifically, we sample two normals M times to guarantee at least one valid sampling, i.e., sampling



(a) (b)

Fig. 5. Illustration of our sequential DD estimation in Hong Kong world. A 3D depiction of this Hong Kong world is shown in Fig. 3(b). (a) We first compute the vertical DD $\mathbf{v}_m$ (see blue DD) based on sampling or IMU. Then we search for a set of horizontal DDs $\{\mathbf{h}_n\}$ (see red, green, cyan, and gray DDs) that are orthogonal to $\mathbf{v}_m$. (b) Given a horizontal DD $\mathbf{h}_n$, we search for a set of sloping DDs $\{\mathbf{s}_z\}$ (see yellow and magenta DDs) that are orthogonal to $\mathbf{h}_n$.



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

Fig. 6. Illustration of our sequential DD estimation in Hong Kong world. A 3D depiction of this Hong Kong world is shown in Fig. 3(b). A node of tree represents an estimated DD. The number associated with a node represents the number of identified inliers with respect to DD. (a) A tree generated by sampling two outliers. (b) A tree generated by sampling two inliers associated with different DDs. (c) A tree generated by sampling two sloping inliers. (d) A tree generated by sampling two vertical inliers.

two vertical inliers (computation of M is available in the supplementary material). Accordingly, we generate M candidate vertical DDs $\{\mathbf{v}_m\}$ $(m = \mathrm{I}, \mathrm{II} \cdots M)$[5]. In addition, if an IMU is available, we use it to obtain the vertical DD, following [16].

Horizontal DDs. As shown in Fig. 5(a), given a candidate vertical DD $\mathbf{v}_m$, we search for a set of horizontal DDs $\{\mathbf{h}_n\}$ $(n = \mathrm{I}, \mathrm{II} \cdots N)$ orthogonal to $\mathbf{v}_m$. We use a new consensus voting strategy for search, instead of traditional BnB [18], [36] (details of our strategy and reason for disusing BnB will be introduced in the next subsection). Our method can automatically determine the number N of horizontal DDs. Moreover, it achieves quasi-global optimality in terms of the number of horizontal inliers. For illustration, let us assume a candidate vertical DD estimated by a valid sampling. This DD is correct but may be affected by noise. Under the condition that this DD should be orthogonal to all the estimated horizontal DDs, our search guarantees to retrieve the maximum number of horizontal inliers. In practice, given M candidate vertical DDs generated above, we can obtain M candidate sets of horizontal DDs. In addition, our method can achieve global optimality if the input vertical DD is obtained by a (nearly) noise-free sensor [37], [38].

Sloping DDs. Given a horizontal DD $\mathbf{h}_n$ of a candidate set estimated above, we search for a set of sloping DDs $\{\mathbf{s}_z\}$ $(z = \mathrm{I}, \mathrm{II} \cdots Z)$ orthogonal to $\mathbf{h}_n$ (see Fig. 5(b)). Similar to the above horizontal DD estimation, we use a consensus voting strategy for search. Our method can automatically determine the number Z of sloping DDs, and achieve quasi-global optimality in terms of the number of sloping inliers. The reason for quasi-global optimality is that our estimated sloping DDs should be orthogonal to the input horizontal DD, but this horizontal DD may be affected by noise.

Figure 6 shows that each two-normal sampling and its follow-up search correspond to a tree. An invalid sampling results in a small number of the identified inliers. Specifically, an invalid sampling refers to sampling two outliers (see Fig. 6(a)), or two inliers associated with different DDs (see Fig. 6(b)), or two horizontal/sloping inliers (see Fig. 6(c)). By contrast, a valid sampling, i.e., sampling two vertical inliers leads to a large number of the identified inliers (see Fig. 6(d)). We save the tree associated with the largest number of the identified inliers. Our full sequential DD estimation is described in Algorithm 1. In addition, our approach is also applicable to Atlanta or Manhattan world by only conducting the above first two steps, i.e., estimating the vertical and horizontal DDs.

Algorithm 1: Sequential DD Estimation

Input: Extracted image lines.
Output: A vertical DD, a set of horizontal DDs, and several sets of sloping DDs.

```
Use image lines {l_k}_{k=1}^K to compute the normals of
  projection planes {n_k}_{k=1}^K (see Fig. 4);
Sample a pair of normals M times;
for each pair of normals do
    Compute a candidate vertical DD v_m (see Fig. 5(a));
end
for each candidate vertical DD v_m do
    Estimate a candidate set of horizontal DDs {h_n}_{n=I}^N
    (see Fig. 5(a));
    for each horizontal DD h_n of a candidate set do
        Estimate a candidate set of sloping DDs {s_z}_{z=I}^Z
        (see Fig. 5(b));
    end
end
Select the optimal tree that maximizes the number of
  inliers (see Fig. 6(d));
```

4.2 Searching for Horizontal and Sloping DDs

In the following, we first illustrate our consensus voting-based horizontal DD search, and then extend it to our sloping DD search. As shown in Fig. 5(a), let us assume that we have obtained a correct but noisy candidate vertical DD $\mathbf{v}_m$ and its associated vertical inliers. To parametrize a horizontal DD $\mathbf{h}_n$, we rotate an arbitrary vertical inlier, e.g., $\mathbf{n}_3$ by an unknown-but-sought angle $\theta_n \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ around the known vertical DD $\mathbf{v}_m$ as

$$\mathbf{h}_n(\theta_n) = \mathbf{R}_{\langle \mathbf{v_m}, \theta_n \rangle} \mathbf{n}_3, \tag{1}$$

where $\mathbf{R}_{\langle \text{axis, angle} \rangle}$ denotes a rotation based on axis-angle representation [17]. Accordingly, each element of $\mathbf{h}_n(\theta_n)$ is a linear combination of $\cos\theta_n$ and $\sin\theta_n$. Figure 5(a) shows that in the noise-free case, the horizontal DD $\mathbf{h}_n(\theta_n)$ is strictly orthogonal to a horizontal inlier $\mathbf{n}_k$, i.e., $\mathbf{h}_n(\theta_n)^\top \mathbf{n}_k = 0$. By substituting Eq. (1) into this constraint, we obtain $a \cdot \cos\theta_n + b \cdot \sin\theta_n = 0$, where a and b are known.

5. We use Arabic numerals, e.g., 1 and 2 to denote the indices of normals, while we use Roman numerals, e.g., I and II to denote the indices of DDs.

Deviations of a and b are available in the supplementary material. Under the presence of noise, we define the residual function $f_k(\theta_n)$ of the horizontal inlier $\mathbf{n}_k$ by

$$f_k(\theta_n) = |a \cdot \cos\theta_n + b \cdot \sin\theta_n| \leqslant \varepsilon, \tag{2}$$

where ε denotes the inlier threshold ($\varepsilon = \cos(\frac{\pi}{2} - \frac{\pi}{90})$ in the experiments). In practice, we do not have prior knowledge regarding whether a normal $\mathbf{n}_k$ is an inlier or outlier. Given K normals $\{\mathbf{n}_k\}_{k=1}^{K}$ corrupted by outliers, we aim to find N horizontal DDs (N is unknown) that maximize the number of horizontal inliers. Mathematically,

$$\max_{N,\{\theta_n\}_{n=1}^{N}} \sum_{n=1}^{N} \underbrace{\sum_{k=1}^{K} \mathcal{I}\big(f_k(\theta_n)\big)}_{c_n}, \text{ subject to } c_n > \tau, \tag{3}$$

where

$$\mathcal{I}\big(f_k(\theta_n)\big) = \begin{cases} 1, & \text{if } f_k(\theta_n) \leqslant \varepsilon; \\ 0, & \text{otherwise,} \end{cases}$$

c_n represents the cardinality of an inlier set associated with a horizontal DD $\mathbf{h}_n$, and τ represents the cardinality threshold ($\tau = 5$ in our experiments). Note that we only save an inlier set whose cardinality is higher than τ. The reason is that some outliers and a small number of inliers may coincidently define a fake DD [39]. Experiments on value setting of the above thresholds ε and τ are available in the supplementary material.

To solve the problem in Eq. (3), traditional BnB [18] seems feasible. Specifically, it first divides the ranges of the unknown angles $\{\theta_n\}_{n=1}^{N}$ into several sub-ranges, and computes the bound of Eq. (3) for each sub-range. Then BnB identifies whether a sub-range is valid (i.e., potentially contains the optimal solution) by comparing the bound in this sub-range with the best-so-far bound. An invalid sub-range is discarded, while a valid sub-range is further divided until the optimal solution is found. BnB leads to unsatisfactory efficiency due to numerous range divisions, especially when the number N of angles $\{\theta_n\}_{n=1}^{N}$ is large. Moreover, BnB assumes that the number N is known a priori. To overcome these limitations, we propose a novel voting-based approach as follows.

We begin with computing a "valid interval" of each normal $\mathbf{n}_k$, regardless of whether $\mathbf{n}_k$ is an inlier or outlier. Specifically, within the valid interval with respect to the angle θ, we can treat the normal $\mathbf{n}_k$ as a horizontal inlier based on Eq. (2), i.e.,

$$f_k(\theta) = |a \cdot \cos\theta + b \cdot \sin\theta| \leqslant \varepsilon \tag{4a}$$

$$\Leftrightarrow |\sqrt{a^2+b^2} \cdot \sin\big(\theta + \arctan(a/b)\big)| \leqslant \varepsilon \tag{4b}$$

$$\Leftrightarrow |\sin\big(\theta + \arctan(a/b)\big)| \leqslant \varepsilon/\sqrt{a^2+b^2} \triangleq \tilde{\varepsilon}. \tag{4c}$$

Eq. (4b) is based on the harmonic addition theorem [40]. The evolution of the function $|\sin\big(\theta + \arctan(a/b)\big)|$ in Eq. (4c) within a one-period interval is shown in Fig. 7. Accordingly, we can obtain a valid interval with respect to the angle θ, i.e., $[-\arcsin(\tilde{\varepsilon}) - \arctan(a/b), \arcsin(\tilde{\varepsilon}) - \arctan(a/b)]$. Intuitively, for a set of inliers with respect to the same horizontal DD $\mathbf{h}_n$, their valid intervals enclose the same unknown-but-sought angle θ_n, i.e., follow a consensus. In another word, these valid intervals overlap with each other. By contrast, valid intervals of outliers are disordered. We leverage this fact to solve the problem in Eq. (3).

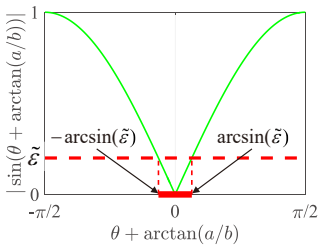


Fig. 7. Evolution of the function $|\sin\big(\theta + \arctan(a/b)\big)|$ in Eq. (4c).

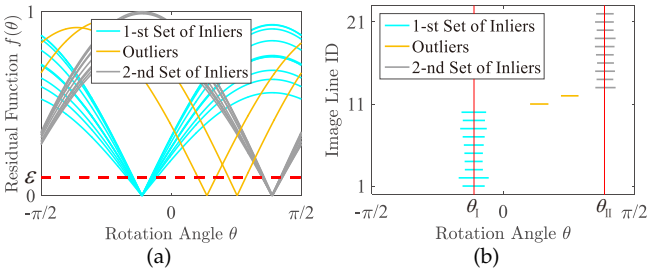


Fig. 8. A consensus of a set of inliers with respect to the same horizontal DD. (a) Residual functions of these inliers lead to adjacent "troughs". (b) Valid intervals of these inliers overlap with each other.

Let us consider some normals corrupted by outliers for illustration. The residual functions and valid intervals of these normals are shown in Figs. 8(a) and 8(b), respectively. At the unknown-but-sought position θ_{I} or θ_{II}, the number of overlapping intervals achieves a local maximum, and also is higher than the cardinality threshold τ. Therefore, we treat both θ_{I} and θ_{II} as the positions of maximum consensus, and substitute them into Eq. (1) to compute the horizontal DDs $\mathbf{h}_{\mathrm{I}}$ and $\mathbf{h}_{\mathrm{II}}$. We introduce how we search for the positions θ_{I} and θ_{II} in the next subsection. Therefore, our approach can automatically determine the number of horizontal DDs, and also obtain the horizontal DDs that maximize the number of inliers.

We extend the above horizontal DD search to the sloping DD search. As shown in Fig. 5(b), let us assume that we have obtained a correct but noisy horizontal DD $\mathbf{h}_n$ and its associated horizontal inliers. To parametrize a sloping DD $\mathbf{s}_z$, we rotate an arbitrary horizontal inlier, e.g., $\mathbf{n}_5$ by an unknown-but-sought angle $\omega_z \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ around the known horizontal DD $\mathbf{h}_n$. Then for each normal $\mathbf{n}_k$, we compute its valid interval with respect to the angle ω. Based on these intervals, we search for the positions of maximum consensus $\{\omega_z\}$ ($z = \mathrm{I}, \mathrm{II} \cdots Z$). Finally, we use these positions to compute the sloping DDs.

4.3 Searching for Positions of Maximum Consensus

In the above subsection, we use several positions of maximum consensus $\{\theta_n\}$ ($n = \mathrm{I}, \mathrm{II} \cdots N$) to compute the horizontal DDs. In the following, we introduce how we search for these positions. Given a set of valid intervals, we use a probe to scan the endpoints of these intervals (see Fig. 9). If the probe scans a left/right endpoint of a valid interval, we increase/decrease the number of votes by 1. After probe scanning, each endpoint is associated with the number of votes. This number of votes equals to the number of overlapping valid intervals. For example, the number of votes at

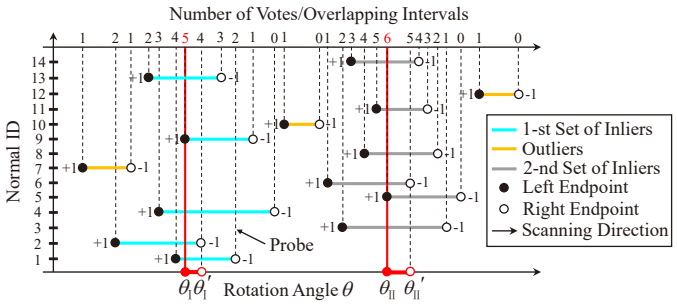


Fig. 9. Illustration of how we search for the positions of maximum consensus θ_{I} and θ_{II}. Here, the cardinality threshold $\tau = 3$.

the endpoint θ_{I}, i.e., 5 equals to the number of overlapping valid intervals within the region $[\theta_{\mathrm{I}}, \theta_{\mathrm{I}}')$. Recall that we aim to maximize the number of overlapping valid intervals. Accordingly, we select the midpoint of each region[6] whose associated number of votes achieves a local maximum and also is higher than the cardinality threshold τ in Eq. (3). For example, we select the midpoints of the regions $[\theta_{\mathrm{I}}, \theta_{\mathrm{I}}')$ and $[\theta_{\mathrm{II}}, \theta_{\mathrm{II}}')$ whose associated number of votes, i.e., 5 and 6 are both local maxima and also higher than the threshold τ. Finally, we treat the selected midpoints as the positions of maximum consensus. We also use the above strategy to search for the positions of maximum consensus $\{\omega_z\}$ $(z = \mathrm{I}, \mathrm{II} \cdots Z)$ for the sloping DD estimation.

In practice, the above method may result in multiple cardinality peaks. Specifically, if the noise level of image lines is too high, the computed valid intervals deviate from each other to some extent. Accordingly, more than one cardinality peaks occur, leading to adjacent under-stabbing probes. We solve this problem by merging two sets of valid intervals if their corresponding under-stabbing probes are close (distance is smaller than $\frac{\pi}{90}$ in our experiments). In addition, we propose a simple but effective strategy to enforce the orthogonality constraint between a pair of horizontal or sloping DDs for consensus voting. Due to limited space, we introduce it in the supplementary material.

5 Camera Pose Estimation

In Hong Kong world, 3D line directions are aligned to DDs. Given several 3D lines in the camera frame and their corresponding image lines, we aim to estimate the camera pose aligning the camera frame to the world frame. Recall that we treat the first camera frame as the world frame.

5.1 Sampling Correspondences

In practice, 3D-2D line correspondences are inevitably corrupted by outliers. To solve this problem, we employ RANSAC [30]. Specifically, we sample three correspondences several times to guarantee at least one valid sampling, i.e., sampling pure inliers. For each three sampled correspondences, their 3D lines constitute a 3D line triplet. Figure 10 shows that in Hong Kong world, three types of 3D line triplets exist. A fully-orthogonal triplet represents three mutually orthogonal 3D lines. A partially-orthogonal triplet corresponds to three 3D lines where two lines are not orthogonal to each other but both orthogonal to the third. A non-orthogonal triplet holds for three non-orthogonal 3D lines. Note that we can easily identify which type a 3D line triplet belongs to based on its known coordinates in the world frame. Our camera pose estimation approach is applicable to all the above types of 3D line triplets.

6. In terms of maximizing the cardinality of consensus set, any other positions within this region are equivalent to the midpoint. Briefly, the probes at arbitrary positions within this region stab the same set of intervals, i.e., correspond to the same consensus set.

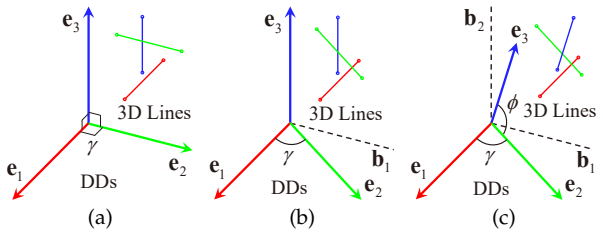


Fig. 10. Typical 3D line triplets aligned to DDs in Hong Kong world. (a) Fully-orthogonal triplet. (b) Partially-orthogonal triplet. (c) Non-orthogonal triplet.

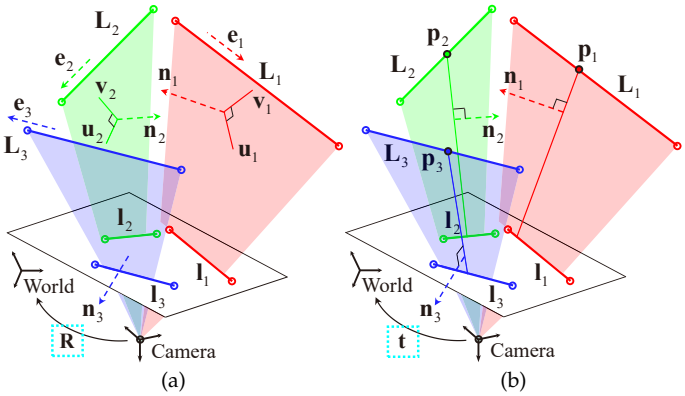


Fig. 11. Illustration of camera pose estimation problem in Hong Kong world. Given three 3D-2D line correspondences $\{(\mathbf{L}_k, \mathbf{l}_k)\}_{k=1}^{3}$, we separately estimate the (a) rotation $\mathbf{R}$ and (b) translation $\mathbf{t}$ that align the camera frame to the world frame.

Instead of random sampling, we preferentially sample three correspondences whose 3D lines constitute a fully-orthogonal triplet, and give the lowest priority to a non-orthogonal triplet. The reason is that the formulas generated by a fully-orthogonal triplet is the easiest one to solve, and a non-orthogonal triplet leads to the highest difficulty (details are available in the next subsections). In addition, we preferentially sample the line correspondences whose image lines are long since long lines are robust to noise.

As shown in Fig. 11, let us assume that we have sampled three inlier line correspondences. We estimate the rotation and translation separately, and propose a novel rotation estimation method. A key step of our rotation estimation is to compute the directions of a 3D line triplet in the camera frame. We parametrize these directions by unknown angles in Section 5.2, and then solve these angles in Section 5.3.

5.2 Parametrizing Line Directions by Angles

We aim to parametrize the directions of a 3D line triplet in the camera frame by unknown angles. As shown in Fig. 11(a), we first compute the normal $\mathbf{n}_k$ of projection plane associated with an image line $\mathbf{l}_k$. We then compute a unit orthogonal basis $\{\mathbf{u}_1, \mathbf{v}_1\}$ of the projection plane orthogonal to the normal $\mathbf{n}_1$. The 3D line $\mathbf{L}_1$ lies within

this projection plane. Therefore, we can use $\{\mathbf{u}_1, \mathbf{v}_1\}$ to parametrize the unit 3D line direction $\mathbf{e}_1$ by an unknown-but-sought rotation angle α as

$$\mathbf{e}_1(\alpha) = \cos\alpha \cdot \mathbf{u}_1 + \sin\alpha \cdot \mathbf{v}_1. \tag{5}$$

Similarly, we compute a unit orthogonal basis $\{\mathbf{u}_2, \mathbf{v}_2\}$ of the projection plane orthogonal to the normal $\mathbf{n}_2$. The 3D line $\mathbf{L}_2$ lies within this plane. Therefore, we can use $\{\mathbf{u}_2, \mathbf{v}_2\}$ to parametrize the unit 3D line direction $\mathbf{e}_2$ by an unknown-but-sought rotation angle β as

$$\mathbf{e}_2(\beta) = \cos\beta \cdot \mathbf{u}_2 + \sin\beta \cdot \mathbf{v}_2. \tag{6}$$

In the following, we use the above angles α and β to parametrize the unit direction $\mathbf{e}_3$ of the 3D line $\mathbf{L}_3$ of different 3D line triplets.

Fully-orthogonal Triplet. Figure 10(a) shows that the directions $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are mutually orthogonal. Therefore, we can compute $\mathbf{e}_3$ by

$$\mathbf{e}_3(\alpha, \beta) = \mathbf{e}_1(\alpha) \times \mathbf{e}_2(\beta). \tag{7}$$

Partially-orthogonal Triplet. As shown in Fig. 10(b), we know the angle γ between the line directions $\mathbf{e}_1$ and $\mathbf{e}_2$. We define an auxiliary unit direction $\mathbf{b}_1$ that is orthogonal to the direction $\mathbf{e}_1$ and also lies on the plane spanned by the directions $\mathbf{e}_1$ and $\mathbf{e}_2$. We compute $\mathbf{b}_1$ by $\mathbf{b}_1(\alpha, \beta) = \frac{\mathbf{e}_2(\beta) - \cos\gamma \cdot \mathbf{e}_1(\alpha)}{\sin\gamma}$. The directions $\mathbf{e}_1$, $\mathbf{b}_1$, and $\mathbf{e}_3$ are mutually orthogonal. Therefore, we can compute $\mathbf{e}_3$ by

$$\mathbf{e}_3(\alpha, \beta) = \mathbf{e}_1(\alpha) \times \mathbf{b}_1(\alpha, \beta). \tag{8}$$

Non-orthogonal Triplet. Similar to the above partially-orthogonal triplet, we first compute the auxiliary direction $\mathbf{b}_1$ (see Fig. 10(c)). In addition, we know the angle ϕ between the line directions $\mathbf{e}_2$ and $\mathbf{e}_3$. We define an auxiliary unit direction $\mathbf{b}_2$ orthogonal to the plane spanned by the directions $\mathbf{e}_1$ and $\mathbf{b}_1$. We compute $\mathbf{b}_2$ by $\mathbf{b}_2(\alpha, \beta) = \mathbf{e}_1(\alpha) \times \mathbf{b}_1(\alpha, \beta)$. Then we can express the direction $\mathbf{e}_3$ by

$$\mathbf{e}_3(\alpha, \beta) = \sin\phi \cdot \mathbf{b}_2(\alpha, \beta) + \cos\phi \cdot \mathbf{e}_2(\beta). \tag{9}$$

In the next subsection, we will solve the above angles α and β to obtain the line directions $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$.

5.3 Solving Angles of Line Directions

We introduce two constraints to solve the above angles α and β. First, based on the above known angle γ between the line directions $\mathbf{e}_1$ and $\mathbf{e}_2$ (see Fig. 10), we have

$$\mathbf{e}_1^\top \mathbf{e}_2 = \cos\gamma. \tag{10}$$

We call it the "angle constraint". Second, Fig. 11(a) shows that the line direction $\mathbf{e}_3$ is orthogonal to the normal $\mathbf{n}_3$ of projection plane, i.e.,

$$\mathbf{e}_3^\top \mathbf{n}_3 = 0. \tag{11}$$

We call it the "projection constraint". In the following, we use the above constraints to generate formulas with respect to the unknown angles α and β and then solve these angles. We define a vector $\boldsymbol{\kappa} = [\cos\alpha \cdot \cos\beta,\ \cos\alpha \cdot \sin\beta,\ \sin\alpha \cdot \cos\beta,\ \sin\alpha \cdot \sin\beta]^\top$ that will be used below. We provide detailed derivations of the known coefficients/coefficient vectors of the following polynomial equations in the supplementary material.

Fully-orthogonal Triplet. We substitute the directions $\mathbf{e}_1(\alpha)$ in Eq. (5) and $\mathbf{e}_2(\beta)$ in Eq. (6) into the angle constraint in Eq. (10), obtaining

$$\boldsymbol{\kappa}^\top \mathbf{m}_1 = \cos\gamma = 0, \tag{12}$$

where $\boldsymbol{\kappa}$ is defined above, $\mathbf{m}_1$ is a known coefficient vector, and $\cos\gamma = 0$ since the directions $\mathbf{e}_1$ and $\mathbf{e}_2$ are orthogonal (see Fig. 10(a)). Then we substitute the direction $\mathbf{e}_3(\alpha, \beta)$ in Eq. (7) into the projection constraint in Eq. (11), obtaining

$$\boldsymbol{\kappa}^\top \mathbf{m}_2 = 0, \tag{13}$$

where $\boldsymbol{\kappa}$ is defined above and $\mathbf{m}_2$ is a known coefficient vector. Polynomials in Eqs. (12) and (13) are both quadratic ones with respect to $\cos\alpha$, $\sin\alpha$, $\cos\beta$ and $\sin\beta$. We reduce these polynomials to linear ones by Werner formulas [40], i.e., $\sin\alpha \cdot \sin\beta = \frac{\cos m - \cos p}{2}$, $\cos\alpha \cdot \cos\beta = \frac{\cos m + \cos p}{2}$, $\sin\alpha \cdot \cos\beta = \frac{\sin p + \sin m}{2}$, and $\cos\alpha \cdot \sin\beta = \frac{\sin p - \sin m}{2}$, where $p = \alpha + \beta$ and $m = \alpha - \beta$. Accordingly, we can transform Eqs. (12) and (13) into

$$\begin{cases} \cos m = A_1 \cdot \cos p + B_1 \cdot \sin p, \\ \sin m = A_2 \cdot \cos p + B_2 \cdot \sin p, \end{cases} \tag{14}$$

where A_1, B_1, A_2 and B_2 are known coefficients. We substitute Eq. (14) into the constraint $\cos^2 m + \sin^2 m = 1$, obtaining

$$\left[\cos^2 p,\ \cos p \cdot \sin p,\ \sin^2 p,\ 1\right] \mathbf{m}_3 = 0, \tag{15}$$

where $\mathbf{m}_3$ is a known coefficient vector. We simplify Eq. (15) by power reduction formulas [40], i.e., $\cos^2 p = \frac{1+\cos(2p)}{2}$, $\sin p \cdot \cos p = \frac{\sin(2p)}{2}$, and $\sin^2 p = \frac{1-\cos(2p)}{2}$. Accordingly, we have

$$a \cdot \cos(2p) + b \cdot \sin(2p) + c = 0 \tag{16a}$$

$$\Leftrightarrow \sqrt{a^2 + b^2} \cdot \sin\left(2p + \arctan(a/b)\right) = -c \tag{16b}$$

where a, b and c are known coefficients, and Eq. (16b) is based on the harmonic addition theorem [40]. We compute p as $p = \frac{\arcsin(-c/\sqrt{a^2+b^2}) - \arctan(a/b)}{2}$ and substitute it into Eq. (14) to compute m. Finally, we obtain the angles $\alpha = \frac{p+m}{2}$ and $\beta = \frac{p-m}{2}$

Partially-orthogonal Triplet. We substitute the directions $\mathbf{e}_1(\alpha)$ in Eq. (5) and $\mathbf{e}_2(\beta)$ in Eq. (6) into the angle constraint in Eq. (10), obtaining

$$\boldsymbol{\kappa}^\top \mathbf{a}_1 = \cos\gamma, \tag{17}$$

where $\boldsymbol{\kappa}$ is defined above, $\mathbf{a}_1$ is a known coefficient vector, and $\cos\gamma \neq 0$ (see Fig. 10(b)). Then we substitute the direction $\mathbf{e}_3(\alpha, \beta)$ in Eq. (8) into the projection constraint in Eq. (11), obtaining

$$\boldsymbol{\kappa}^\top \mathbf{a}_2 = 0, \tag{18}$$

where $\mathbf{a}_2$ is a known coefficient vector. Based on the above Werner formulas, we can transform Eqs. (17) and (18) into

$$\begin{cases} \cos m = A_1 \cdot \cos p + B_1 \cdot \sin p + C_1, \\ \sin m = A_2 \cdot \cos p + B_2 \cdot \sin p + C_2. \end{cases} \tag{19}$$

TABLE 1
Equations and solvers with respect to different 3D line triplets

Triplet	Equation	Solver
Fully-orthogonal	Sine (Eq. (16b))	Inverse sine
Partially-orthogonal	Polynomial (Eq. (21))	Eigenvalues
Non-orthogonal	Polynomials×2 (Eq. (24))	Gröbner Basis

where A_1, B_1, C_1, A_2, B_2 and C_2 are known coefficients, $p = \alpha + \beta$, and $m = \alpha - \beta$. We substitute Eq. (19) into the constraint $\cos^2 m + \sin^2 m = 1$, obtaining

$$\left[\cos^2 p,\ \cos p \cdot \sin p,\ \sin^2 p,\ \cos p,\ \sin p,\ 1\right] \mathbf{a}_3 = 0, \quad (20)$$

where $\mathbf{a}_3$ is a known coefficient vector. Note that Eq. (20) contains additional linear terms with respect to $\cos p$ and $\sin p$, compared with Eq. (15). Accordingly, we do not use the above power reduction formulas to simplify Eq. (20). Instead, we exploit Weierstrass substitution [40]. Specifically, we define $w = \tan(\frac{p}{2})$, and substitute $\cos p = \frac{1-w^2}{1+w^2}$ and $\sin p = \frac{2w}{1+w^2}$ into Eq. (20). Accordingly, we can obtain a quartic polynomial equation with respect to w, i.e.,

$$[w^4,\ w^3,\ w^2,\ w,\ 1]^\top \mathbf{a}_4 = 0, \quad (21)$$

where $\mathbf{a}_4$ is a known coefficient vector. We solve this polynomial equation using the eigenvalue-based method [17] to obtain w. Then we substitute w back into the above Weierstrass substitution to compute p, and further compute m based on Eq. (19). Finally, we obtain the angles $\alpha = \frac{p+m}{2}$ and $\beta = \frac{p-m}{2}$.

Non-orthogonal Triplet. As shown in Fig. 10(c), the angle constraint of the non-orthogonal triplet is the same as that of the partially-orthogonal triplet. Accordingly, we have

$$\boldsymbol{\kappa}^\top \mathbf{h}_1 = \cos\gamma, \quad (22)$$

where $\boldsymbol{\kappa}$ is defined above and $\mathbf{h}_1$ is a known coefficient vector. Then we substitute the direction $\mathbf{e}_3(\alpha, \beta)$ in Eq. (9) into the projection constraint in Eq. (11), obtaining

$$\boldsymbol{\rho}^\top \mathbf{h}_2 = 0, \quad (23)$$

where $\boldsymbol{\rho} = [\boldsymbol{\kappa}^\top, \cos\alpha, \sin\alpha, \cos\beta, \sin\beta]^\top$ and $\mathbf{h}_2$ is a known coefficient vector. Note that $\boldsymbol{\rho}$ contains additional terms $\cos\alpha$, $\sin\alpha$, $\cos\beta$, and $\sin\beta$, compared with $\boldsymbol{\kappa}$. Accordingly, we do not use the above Werner formulas to simplify Eqs. (22) and (23). Instead, we use Weierstrass substitution. Specifically, we define $q_1 = \tan(\frac{\alpha}{2})$ and $q_2 = \tan(\frac{\beta}{2})$, and substitute $\cos\alpha = \frac{1-q_1^2}{1+q_1^2}$, $\sin\alpha = \frac{2q_1}{1+q_1^2}$, $\cos\beta = \frac{1-q_2^2}{1+q_2^2}$, $\sin\beta = \frac{2q_2}{1+q_2^2}$ into Eqs. (22) and (23). Accordingly, we can generate a quartic polynomial system with respect to q_1 and q_2, i.e.,

$$\begin{cases} \left[q_1^2q_2^2,\ q_1^2q_2,\ q_1^2,\ q_1q_2^2,\ q_1q_2,\ q_1,\ q_2^2,\ q_2,\ 1\right]^\top \mathbf{h}_3 = 0 \\ \left[q_1^2q_2^2,\ q_1^2q_2,\ q_1^2,\ q_1q_2^2,\ q_1q_2,\ q_1,\ q_2^2,\ q_2,\ 1\right]^\top \mathbf{h}_4 = 0 \end{cases} \quad (24)$$

where $\mathbf{h}_3$ and $\mathbf{h}_4$ are known coefficient vectors. We solve this system based on the Gröbner basis [41] to obtain q_1 and q_2. Then we substitute q_1 and q_2 back into the above Weierstrass substitutions to obtain the angles α and β.

Overall, based on the angle and projection constraints, we generate equations with respect to the unknown angles α and β of our 3D line directions. Then we solve these equations to obtain the angles α and β. We summarize our equations and solvers in Table 1.

5.4 Computing Rotation and Translation

Given the angles α and β estimated above, we compute the line directions $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ in the camera frame (see Section 5.2). In addition, we compute the corresponding line directions in the world frame using the known coordinates of 3D lines in the world frame. Based on these line direction correspondences, we use [42] to obtain the closed-form solution of the rotation $\mathbf{R}$. Then based on the known rotation $\mathbf{R}$, we compute the translation $\mathbf{t}$. Specifically, as shown in Fig. 11(b), a 3D point $\mathbf{p}_k$ in the world frame lies on a 3D line $\mathbf{L}_k$. The direction defined by $\mathbf{p}_k$ and the camera center is orthogonal to the normal $\mathbf{n}_k$ of projection plane. We express this constraint as $\mathbf{n}_k^\top\left(\mathbf{R}^\top(\mathbf{p}_k - \mathbf{t})\right) = 0$ in the camera frame, and transform it as $(\mathbf{n}_k^\top\mathbf{R}^\top)\mathbf{t} = \mathbf{n}_k^\top\mathbf{R}^\top\mathbf{p}_k$. Each 3D-2D line correspondence can provide such a linear equation with respect to the unknown translation $\mathbf{t}$. Given three correspondences, we combine their linear equations as a linear system to compute the translation $\mathbf{t}$.

6 Back-end

As introduced in Section 3, the global and local DDs represent DDs in the world and camera frames, respectively. We use our regularity encoder to estimate N local DDs $\{\mathbf{d}_i^n\}_{n=1}^N$ in the i-th camera frame. In addition, we use our front-end to estimate the rotation $\mathbf{R}_i$ of the i-th camera. As shown in Fig. 12(a), given the local DD $\mathbf{d}_i$ and rotation $\mathbf{R}_i$, we can compute the global DD $\mathbf{g}_i$ by

$$\mathbf{g}_i = \mathbf{R}_i\mathbf{d}_i. \quad (25)$$

In practice, both rotation $\mathbf{R}_i$ and local DD $\mathbf{d}_i$ are inevitably affected by noise. Accordingly, the computed global DD $\mathbf{g}_i$ (e.g., $\mathbf{g}_1 = \mathbf{R}_1\mathbf{d}_1$ used by existing SLAM methods [15], [16]) may deviate from the unknown ground truth one. To obtain accurate global DDs, we propose to recursively refine the computed global DDs in Section 6.1. Then we use these refined global DDs to optimize the camera poses and 3D lines in Section 6.2.

6.1 Refining Global DDs

We propose a Kalman filter-based method to recursively refine the global DDs. Recall that a local DD corresponds to a vanishing point that is the intersection of a set of image lines. We model the uncertainty of an extracted image line by a covariance matrix, following [43]. For a set of image lines associated with a vanishing point/local DD, we use their covariance matrices to compute the covariance matrix of a local DD by error propagation [17]. We then compute the observed global DD $\mathbf{g}_i^{\text{o}}$ and its covariance matrix $\boldsymbol{\Sigma}_i^{\text{o}}$ based on Eq. (25) and error propagation. In addition, let us assume that we have obtained the refined global DD $\hat{\mathbf{g}}_{i-1}$ and its covariance matrix $\hat{\boldsymbol{\Sigma}}_{i-1}$ of the $(i-1)$-th camera. Considering that the unknown ground truth global DD is a

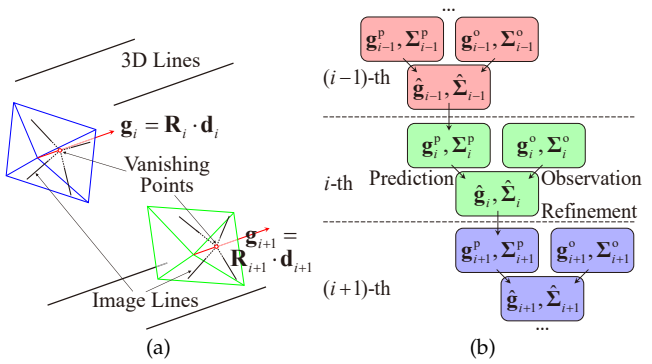


Fig. 12. Illustrations of the relation between the local and global DDs, as well as our global DD refinement. (a) The camera rotation $\mathbf{R}_i$ from the camera frame to the world frame aligns each local DD $\mathbf{d}_i$ to the global DD $\mathbf{g}_i$ denoted by the red arrow. (b) We recursively refine a global DD based on Kalman filter.

constant, we treat the above refined result $(\hat{\mathbf{g}}_{i-1}, \hat{\mathbf{\Sigma}}_{i-1})$ as the prediction $(\mathbf{d}_i^{\mathrm{p}}, \mathbf{\Sigma}_i^{\mathrm{p}})$ of the i-th camera.

By combining the above observation and prediction, we can refine the global DD and update its covariance matrix (see Fig. 12(b)). Specifically, we first use the predicted covariance matrix $\mathbf{\Sigma}_i^{\mathrm{p}}$ and the observed covariance matrix $\mathbf{\Sigma}_i^{\mathrm{o}}$ to compute the Kalman gain G_i [27]. Then we use G_i, the predicted global DD $\mathbf{g}_i^{\mathrm{p}}$, and the observed global DD $\mathbf{g}_i^{\mathrm{o}}$ to obtain the refined global DD $\hat{\mathbf{g}}_i$ by

$$\hat{\mathbf{g}}_i = \mathbf{g}_i^{\mathrm{p}} + G_i \cdot (\mathbf{g}_i^{\mathrm{o}} - \mathbf{g}_i^{\mathrm{p}}). \tag{26}$$

Moreover, we use Kalman gain G_i and the predicted covariance matrix $\mathbf{\Sigma}_i^{\mathrm{p}}$ to obtain the updated covariance matrix $\hat{\mathbf{\Sigma}}_i$ by

$$\hat{\mathbf{\Sigma}}_i = (1 - G_i) \cdot \mathbf{\Sigma}_i^{\mathrm{p}}. \tag{27}$$

We treat the refined result $(\hat{\mathbf{g}}_i, \hat{\mathbf{\Sigma}}_i)$ of the i-th camera in Eqs. (26) and (27) as the prediction $(\mathbf{g}_{i+1}^{\mathrm{p}}, \mathbf{\Sigma}_{i+1}^{\mathrm{p}})$ of the $(i+1)$-th camera, and thus complete an iteration.

6.2 Optimizing Camera Poses and 3D Lines

Based on the above refined global DDs, we propose a DD alignment-based method to optimize the camera poses and 3D lines. Specifically, traditional landmarks like 3D points or lines can be only observed by a limited number of cameras. By contrast, DDs can be observed in the global scene. Let us consider the i-th camera to illustrate our DD alignment-based rotation optimization. Given N estimated local DDs $\{\mathbf{d}_i^n\}_{n=1}^N$ and their corresponding refined global DDs $\{\hat{\mathbf{g}}_i^n\}_{n=1}^N$, we aim to find the optimal rotation $\mathbf{R}_i$ to align each pair of global-local DDs based on Eq. (25). Such an alignment is independent of the other cameras, and thus can significantly reduce the error accumulation of rotation. A straightforward way for alignment is combining all pairs of global-local DDs $\{\hat{\mathbf{g}}_i^n, \mathbf{d}_i^n\}_{n=1}^N$ to generate the equation $[\hat{\mathbf{g}}_i^1, \hat{\mathbf{g}}_i^2, \cdots, \hat{\mathbf{g}}_i^N] = \mathbf{R}_i[\mathbf{d}_i^1, \mathbf{d}_i^2, \cdots, \mathbf{d}_i^N]$. While $\mathbf{R}_i$ can be easily solved by [42], this combination is prone to being affected by outlier, i.e., false association between a pair of global and local DDs [23]. To overcome this limitation, we propose a robust method as follows.

We repeatedly sample three non-coplanar DDs (e.g., a vertical DD, a horizontal DD, and a sloping DD) to define a DD triplet. Given N DDs, we can generate S different DD triplets. Note that $S \leqslant \binom{N}{3}$ since we do not consider three coplanar DDs (e.g., three horizontal DDs) that may result in degeneration. We treat the DD triplets in the world and camera frames as the global and local triplets, respectively. In the noise- and outlier-free case, the rotation $\mathbf{R}_i$ aligns a local DD triplet $\mathcal{D}_i$ to a global DD triplet $\mathcal{G}_i$ by $\mathcal{G}_i = \mathbf{R}_i\mathcal{D}_i$, which is similar to Eq. (25). In practice, given S pairs of global-local DD triplets $\{(\mathcal{G}_i^s, \mathcal{D}_i^s)\}_{s=1}^S$, we compute S candidate rotations $\{\mathbf{R}_i^s\}_{s=1}^S$ by $\mathbf{R}_i^s = \mathcal{G}_i^s\mathcal{D}_i^{s-1}$. Then we leverage [44] to average these candidate rotations, obtaining the optimal rotation $\hat{\mathbf{R}}_i$, i.e.,

$$\min_{\hat{\mathbf{R}}_i} \sum_{s=1}^{S} L_1(\hat{\mathbf{R}}_i, \mathbf{R}_i^s), \tag{28}$$

where $L_1(\cdot,\cdot)$ denotes the L_1-mean. Our method is robust to outliers and noise thanks to rotation averaging based on L_1-mean. In addition, we follow [8], [23] to conduct the DD-constrained bundle adjustment. Specifically, we align 3D line directions to their corresponding refined global DDs when minimizing re-projection error.

7 Experiments

We first introduce our dataset of sequential images in Hong Kong world. Then we compare our SLAM method with state-of-the-art approaches introduced in Section 2. Finally, considering that the overall SLAM accuracy depends on various modules (e.g., DD/vanishing point estimation, camera pose estimation, and back-end optimization), we conduct ablation study of each module independently. Additional experimental results are available in the supplementary material. All the methods in our experiments are implemented in C++. We conduct tests on a computer equipped with an Intel Core i7 3.2 GHz CPU and 16 GB RAM.

7.1 Our CUHK-SLAM Datasets

Existing SLAM datasets [16], [38] are only suitable for the experiments in Manhattan and Atlanta worlds. To evaluate the proposed algorithms, we establish the first dataset of sequential images in Hong Kong world. We collect data on the campus of The Chinese University of Hong Kong (CUHK) and call our dataset "CUHK-SLAM dataset". Figure 13 and Table 2 show that our dataset is composed of 7 image sequences (9077 images). Each sequence corresponds to a scene with at least one sloping DD, e.g., sloping road, stairway, and sloping roof. We obtain data by

TABLE 2
Information regarding image sequences of our CUHK-SLAM dataset.

Sequence	Trajectory Length	Number of Images
Lady_Shaw_Road	96.35 m	1653
Admin_Building	74.33 m	1671
Central_Ave	73.02 m	1007
Faculty_Law	44.67 m	895
Faculty_Arts	74.23 m	1519
Student_Canteen	56.58 m	1354
Station_Road	45.19 m	978

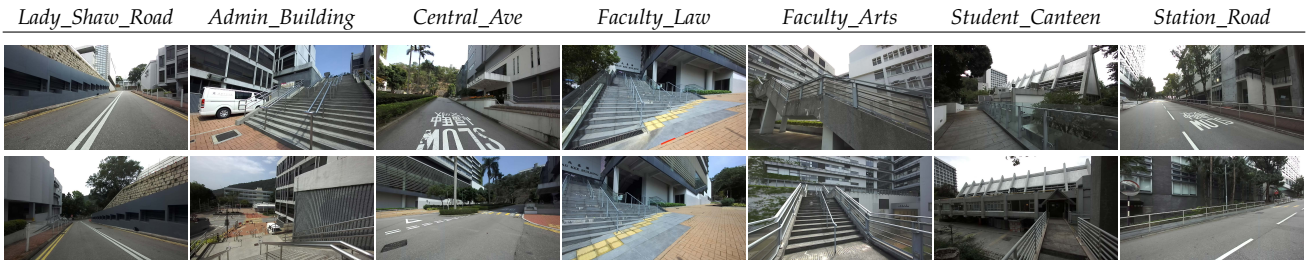


Fig. 13. Sample images of our CUHK-SLAM dataset. Each column corresponds to an image sequence satisfying Hong Kong world configuration.

a handhold platform equipped with camera, IMU and real-time kinematic (RTK) positioning module. We associate data of different sensors by the timestamps of robot operating system. We provide calibration parameters of each sensor. The image size is 1280×720 pixels, and we undistort images beforehand.[7] We use IMU to obtain the accurate vertical DD in the camera frame, following [47]. We use RTK that can provide centimeter-level positioning to obtain the ground truth trajectory.

7.2 SLAM Methods

Evaluation Criteria. We employ Umeyama algorithm [48] to align the estimated and ground truth trajectories, following [23], [37]. Accordingly, two aligned trajectories share the same scale and start point. Given the ground truth position $\hat{\mathbf{x}}_i$ and the estimated position $\mathbf{x}_i$ that are associated by the timestamps, we compute the absolute trajectory error [37] by $\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|$. To evaluate the overall accuracy of the estimated trajectory, we compute the root mean square of the absolute trajectory errors.

Methods for Comparison. We compare the following line-based SLAM methods:

- FI-SLAM [11]: The filter-based method without considering the structural regularity.
- GR-SLAM [26]: The graph-based method without using the structural regularity.[8]
- AT-SLAM: The filter-based method exploiting the structural regularity of Atlanta world [16].
- MA-SLAM [23]: The graph-based method utilizing the structural regularity of Manhattan world.
- HK-SLAM: Our optimization-based method leveraging the structural regularity of Hong Kong world.

On our CUHK-SLAM dataset, while sub-trajectories may intersect with each other, it is difficult to detect loop closure due to different viewpoints of cameras. Accordingly, our reported results do not involve loop correction.

Experimental Results. As shown in Table 3, as well as Figs. 14 and 16, we compare trajectories estimated by various SLAM methods. We also present 3D lines reconstructed by our HK-SLAM in Fig. 15. FI-SLAM leads to unsatisfactory accuracy since its filter is relatively sensitive to noise and also it does not exploit the structural regularity for optimization. GR-SLAM improves the accuracy to some extent by bundle adjustment. However, it also neglects the structural constraints and thus becomes unstable at sharp turns. AT-SLAM treats all the sloping inliers as outliers. Accordingly, it fails to use sufficient observations to compensate for noise. In addition, similar to FI-SLAM, its unreliable filter results in inferior performance to GR-SLAM on some sequences. MA-SLAM cannot enforce the structural constraints related to the sloping and partial horizontal DDs. Despite this limitation, it still outperforms GR-SLAM on most sequences, demonstrating the effectiveness of the structural constraint-based optimization. Our HK-SLAM achieves the highest accuracy since it can exploit information regarding the sloping DDs and also our DD alignment-based optimization is effective.

7.3 DD/Vanishing Point Estimation

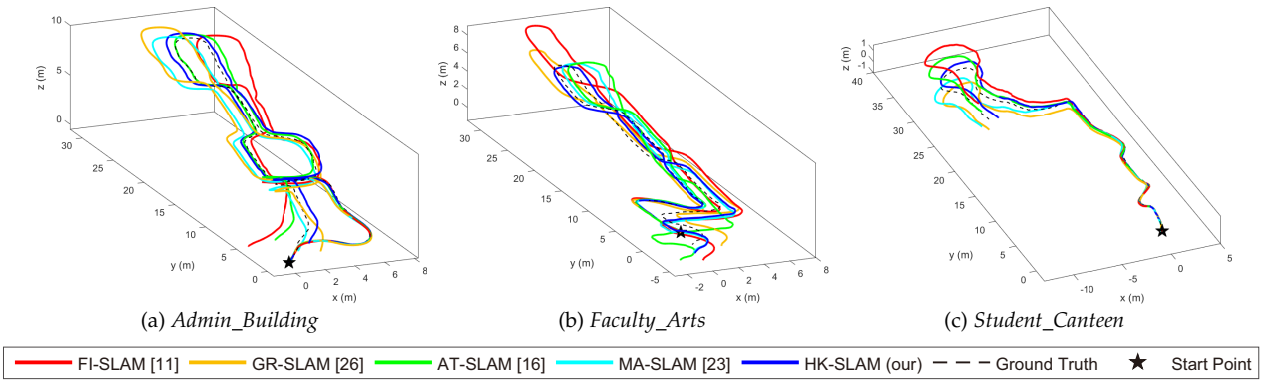
Evaluation Criteria. We randomly sample 500 images from our CUHK-SLAM dataset and treat them as the testing images (see Fig. 17). We follow [49] to manually assign image lines with ground truth cluster labels. Based on these labels, we follow [50] to evaluate the algorithm accuracy in terms of precision and recall of image line clustering. Specifically, the precision is defined by $\frac{C}{C+W}$, and recall is defined by $\frac{C}{C+M}$,

7. An alternative strategy is to directly detect arcs [45] or curves [46] in the distorted image. Then we can use these arcs or curves to estimate the distortion parameters, and further undistort these arcs or curves into straight lines.

8. For a fair comparison, we let [26] to use only lines but not points.

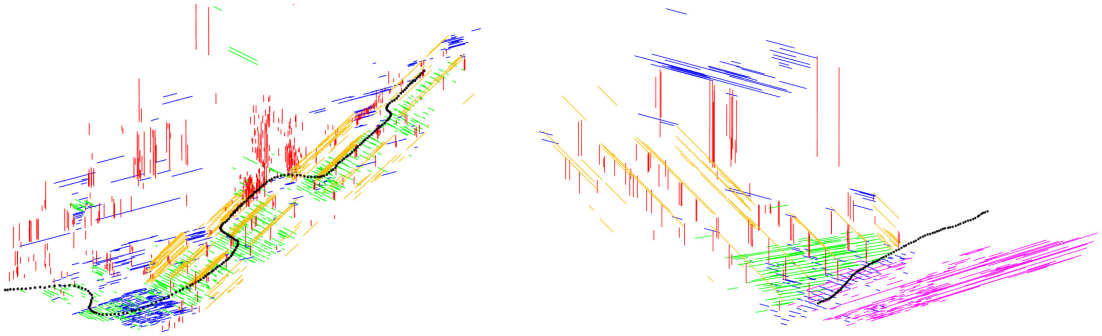
TABLE 3
Absolute trajectory errors of various SLAM methods on all the image sequences of our CUHK-SLAM dataset.

Sequence	FI-SLAM [11]	GR-SLAM [26]	AT-SLAM [16]	MA-SLAM [23]	HK-SLAM (our)
Lady_Shaw_Road	2.54 m	2.52 m	1.35 m	1.66 m	**1.02** m
Admin_Building	1.29 m	1.19 m	0.60 m	0.72 m	**0.47** m
Central_Ave	2.80 m	1.24 m	2.02 m	2.11 m	**0.95** m
Faculty_Law	0.81 m	0.89 m	0.95 m	0.74 m	**0.58** m
Faculty_Arts	3.16 m	2.07 m	2.86 m	1.47 m	**1.08** m
Student_Canteen	1.69 m	1.04 m	0.72 m	0.67 m	**0.41** m
Station_Road	0.78 m	0.72 m	0.45 m	0.49 m	**0.37** m



(a) *Admin_Building* (b) *Faculty_Arts* (c) *Student_Canteen*

Fig. 14. Trajectories estimated by various SLAM methods on three image sequences of our CUHK-SLAM dataset.



4 DDs (1 Sloping DD) 5 DDs (2 Sloping DDs)

(a) *Admin_Building* (b) *Faculty_Law*

Fig. 15. 3D lines reconstructed by our HK-SLAM on two image sequences of our CUHK-SLAM dataset. 3D lines aligned to the vertical, horizontal, and sloping DDs are shown in red, {green, blue}, and {yellow, magenta}, respectively. The black dotted lines denote the trajectories estimated by our HK-SLAM.

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 that simultaneously encodes the precision and recall by $F_1 = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$. For an unbiased comparison, we follow [51] to report the original results without least-squares global optimization, as we will do in Section 7.4.

Methods for Comparison. We compare the following DD estimation methods:

- BR-DD [18]: The branch and bound-based method designed for Manhattan world.
- IMU-RA-DD [16]: The IMU and RANSAC-based method used by AT-SLAM.
- JL-DD [31]: The J-Linkage-based method used by MA-SLAM.
- HK-SA-DD: Our method that is designed for Hong Kong world and estimates the vertical DD by sampling (see Section 4).
- HK-IMU-DD: Our method that is designed for Hong Kong world and estimates the vertical DD by IMU (see Section 4).

Experimental Results. We present the accuracy comparisons in Figs. 17 and 18. BR-DD can only estimate three orthogonal DDs. Accordingly, in addition to sloping DDs, it may neglect partial horizontal DDs. While IMU-RA-DD can retrieve all the horizontal DDs, it fails to estimate the sloping DDs. Moreover, its estimated horizontal DDs are relatively inaccurate since RANSAC is sensitive to noise. While JL-DD can theoretically determine the number of DDs, in practice, it is prone to resulting in over/under clustering due to the uncertainty of sampling. Moreover, its accuracy is unsatisfactory since it neglects the orthogonality between DDs and also its line descriptor generated by sampling is sensitive to noise. Our HK-SA-DD can reliably identify all the DDs. Moreover, it is more robust to noise than IMU-RA-DD and JL-DD since it estimates most DDs by search instead of sampling. Our HK-IMU-DD further improves the accuracy. Thanks to IMU, it obtains the accurate vertical DD and also reduces the error propagation when searching for the horizontal and sloping DDs.

Figure 17 and Table 4 show the efficiency comparisons. BR-DD leads to relatively low time cost since it only computes three DDs. The efficiency of IMU-RA-DD is also satisfactory. This method does consider the vertical inliers identified by IMU when estimating the horizontal DDs, which increases the probability of valid sampling. JL-DD results in unsatisfactory efficiency since it generates line descriptors by more than 1000 samplings in general. By contrast, our HK-SA-DD achieves near real-time efficiency for two main reasons. First, it uses a small number of samplings to reduce the search space. Second, the computational complexity of our search strategy, i.e., consensus voting is relatively low. Our HK-IMU-DD achieves real-time efficiency since it uses IMU to avoid sampling, and only searches on a single tree.

7.4 Camera Pose Estimation

Evaluation Criteria. We randomly sample 500 images from our CUHK-SLAM dataset and treat them as query im-

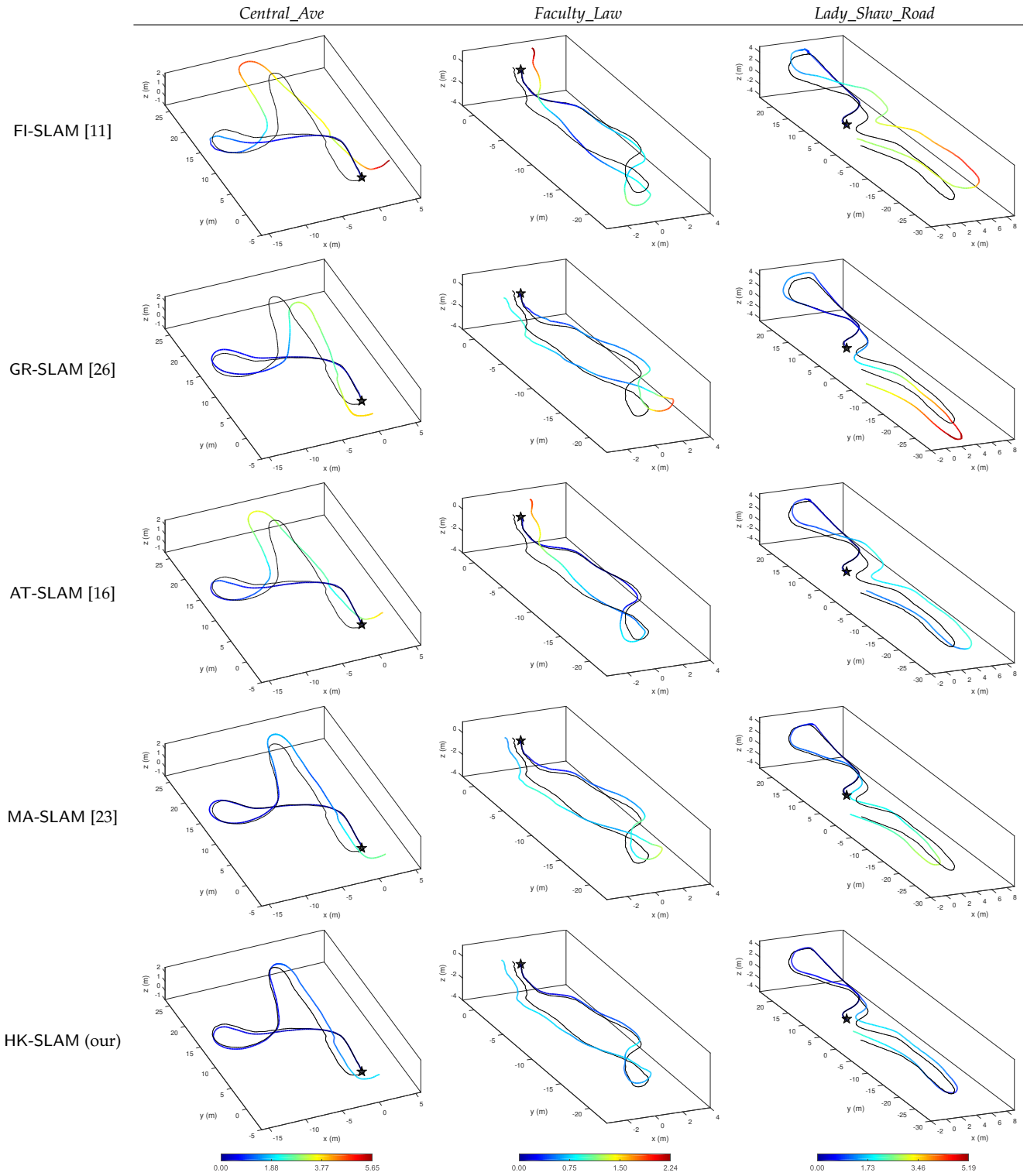


Fig. 16. Trajectories estimated by various SLAM methods on three image sequences of our CUHK-SLAM dataset. The colored and black lines denote the estimated and ground truth trajectories, respectively. Pentagram represents the starting point of trajectory. Color bar indicates the magnitude of the absolute trajectory error.

ages (see Fig. 19). We follow [52] to associate each query image with a set of 3D-2D line correspondences. We use these correspondences to estimate the pose of query image. Given the ground truth rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}$, we follow [21], [29] to evaluate the estimated rotation $\mathbf{R}$ and translation $\mathbf{t}$. The rotation error is defined by the mean of $\{\arccos(\mathbf{r}_s^\top \hat{\mathbf{r}}_s) \times 180/\pi\}_{s=1}^3$ (degree), where $\mathbf{r}_s$ and $\hat{\mathbf{r}}_s$ represent the s-th columns of $\mathbf{R}$ and $\hat{\mathbf{R}}$, respectively. The translation error is defined by $\|\hat{\mathbf{t}} - \mathbf{t}\|/\|\mathbf{t}\| \times 100$ (%). In addition, we follow [52] to evaluate the accuracy in terms of visual alignment between 2D edges. Specifically, we manually extract some ground truth edges in the image (see cyan edges in Fig. 19). Then we manually match these 2D edges between images and use the ground truth camera poses

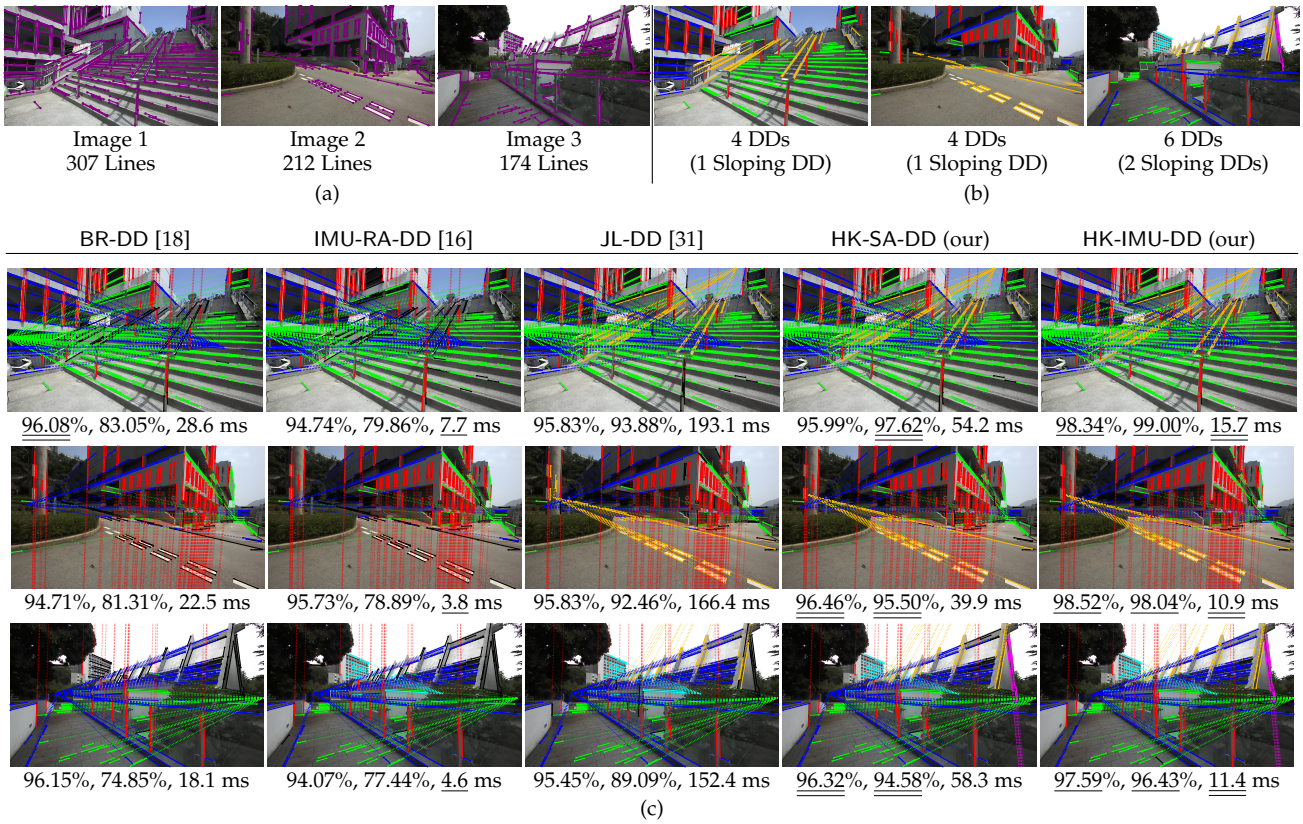


Fig. 17. Accuracy and efficiency comparisons between various DD estimation methods on three testing images of our CUHK-SLAM dataset. (a) Image lines extracted by LSD [34]. (b) The ground truth image line clusters. (c) The clustered image lines associated with the vertical, horizontal, and sloping DDs are shown in red, {green, blue, cyan}, and {yellow, magenta}, respectively. The unclustered image lines are shown in black. A dotted line represents the connection between the midpoint of a clustered image line and an estimated vanishing point. The numbers below each image represent the precision, recall, and run time, respectively. We highlight the best and second-best results by "__" and "__", respectively.

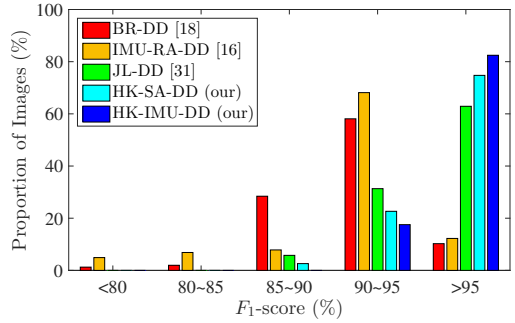


Fig. 18. Accuracy comparison between various DD estimation methods in terms of F_1-score of image line clustering on all the testing images of our CUHK-SLAM dataset.

TABLE 4
Efficiency comparison between various DD/vanishing point estimation methods in terms of average run time on all the testing images of our CUHK-SLAM dataset.

Method	Run Time
BR-DD [18]	21.9 ms
IMU-RA-DD [16]	5.3 ms
JL-DD [31]	170.8 ms
HK-SA-DD (our)	55.4 ms
HK-IMU-DD (our)	13.6 ms

TABLE 5
Efficiency comparison between various camera pose estimation methods in terms of average run time on all the query images of our CUHK-SLAM dataset.

Method	Run Time
CO-CP [27]	0.2 ms
EN-CP [21]	49.5 ms
OR-CP [29]	9.1 ms
PR-CP [22]	75.3 ms
HK-CP (our)	36.9 ms

to reconstruct 3D edges by triangulation. We project these 3D edges back to the query image using the camera pose estimated by each method, obtaining the projected 2D edges (see yellow edges in Fig. 19). A better alignment between the ground truth and projected 2D edges represents higher accuracy of the estimated camera pose.

Methods for Comparison. We compare the following camera pose estimation methods:

- CO-CP [27]: The constant velocity motion model used by FI-SLAM and AT-SLAM.[9]
- EN-CP [21]: The endpoint constraint-based method used by GR-SLAM.
- OR-CP [29]: The orthogonality constraint-based method.
- PR-CP [22]: The projection constraint-based method without exploiting the structural regularity.
- HK-CP: Our method designed for Hong Kong world (see Section 5).

Experimental Results. We present the accuracy comparisons in Figs. 19 and 20. CO-CP leads to low accuracy since the constant velocity motion model is unreliable when the camera accelerates or turns. EN-CP is relatively sensitive to noise. The reason is that image lines may not be completely

9. CO-CP does not use 3D-2D line correspondences, which is different from the other camera pose estimation methods.

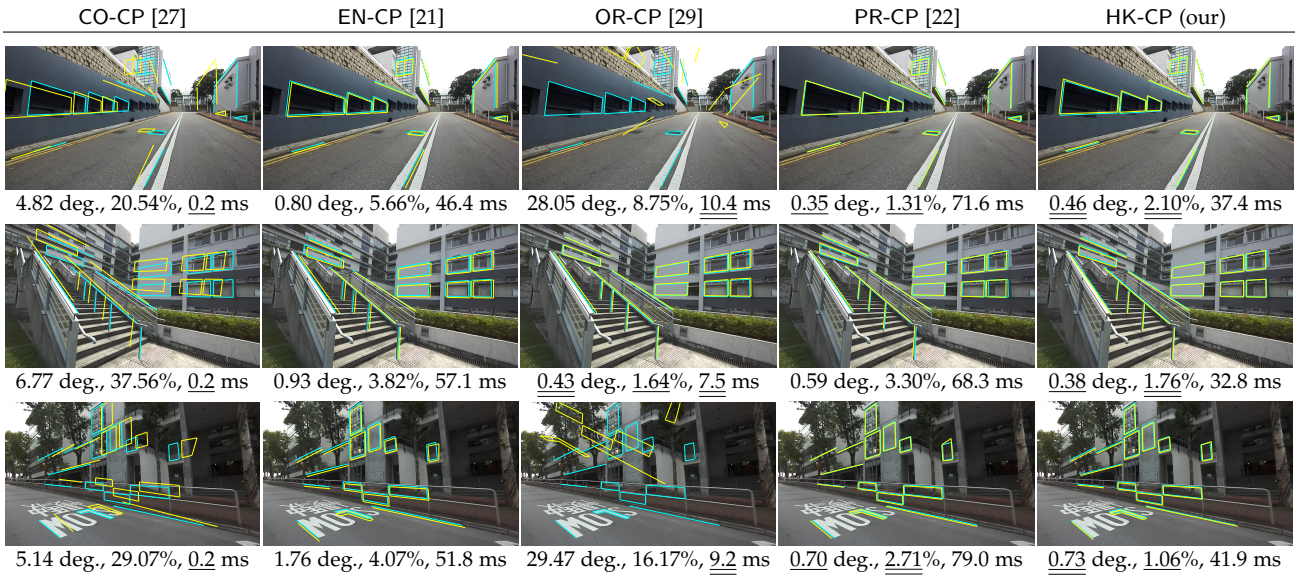


Fig. 19. Accuracy and efficiency comparisons between various camera pose estimation methods on three query images of our CUHK-SLAM dataset. The numbers below each image represent the rotation error, translation error and run time, respectively. The ground truth and projected 2D edges are shown in cyan and yellow, respectively. We highlight the best and second-best results by "_" and "=", respectively.

TABLE 6
Absolute trajectory errors of baseline and our SLAM methods on different image sequences of our CUHK-SLAM dataset.

Sequence	NEI	ALI	ALI-FIL (our)
Lady_Shaw_Road	2.11 m	1.04 m	1.02 m
Admin_Building	1.03 m	0.47 m	0.47 m
Central_Ave	1.10 m	0.95 m	0.95 m
Faculty_Law	0.93 m	0.58 m	0.58 m
Faculty_Arts	1.89 m	1.08 m	1.08 m
Student_Canteen	0.95 m	0.46 m	0.41 m
Station_Road	0.73 m	0.45 m	0.37 m

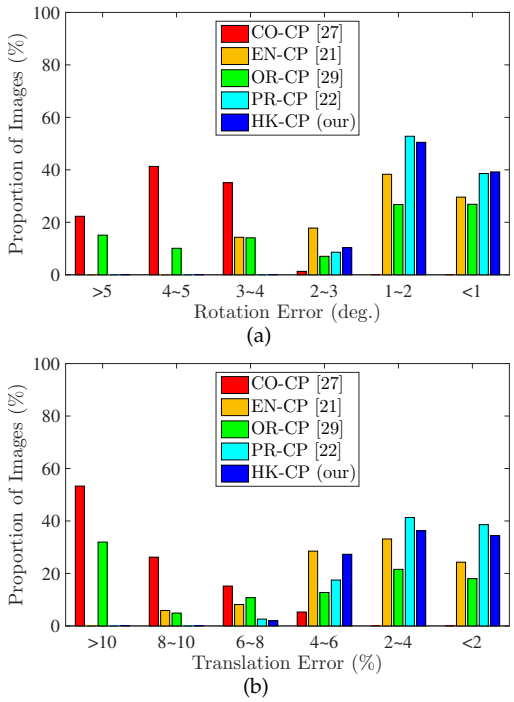


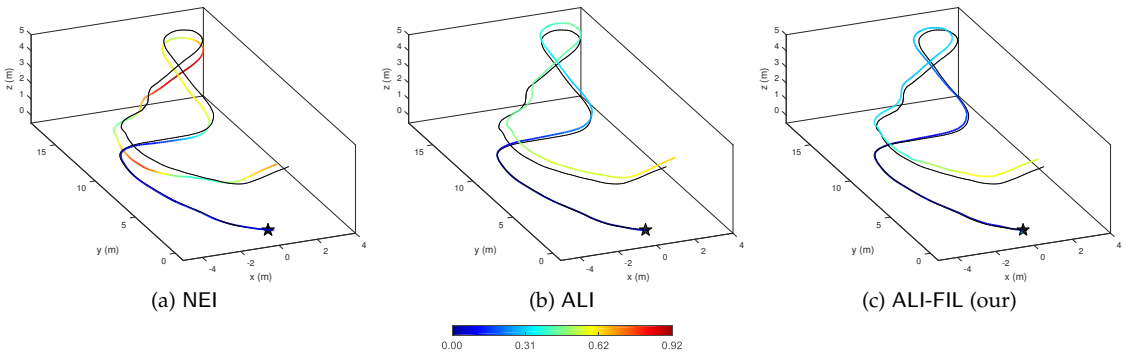
Fig. 20. Accuracy comparison between various camera pose estimation methods in terms of (a) rotation error and (b) translation error on all the query images of our CUHK-SLAM dataset.

extracted and thus the endpoints of 3D-2D line correspondences may not be associated. The accuracy of OR-CP is satisfactory when the fully-orthogonal triplets of 3D-2D line correspondences are sufficient. However, when this configuration is unobservable, OR-CP mistakenly estimates the camera pose. For example, on the first and third images of Fig. 19, we can only observe the partially-orthogonal triplets but not fully-orthogonal triplets. PR-CP and our HK-CP both achieve high accuracy thanks to their respective advantages. Specifically, on partial images (e.g., the second image of Fig. 19), our HK-CP leverages the structural regularity to generate lower-order polynomial equations, and thus provides higher numerical stability/accuracy than PR-CP. On the other images (e.g., the first image of Fig. 19), 3D line triplets of 3D-2D line correspondences are perturbed by noise, which affects the structural constraint. Accordingly, our HK-CP is slightly inferior to PR-CP.

Figure 19 and Table 5 show the efficiency comparisons. CO-CP is the fastest method since the constant velocity motion model is simple. EN-CP is relatively time-consuming due to its complex endpoint-based constraint. OR-CP provides high efficiency since it uses the orthogonality constraint to simplify equations. The run time of PR-CP is relatively unsatisfactory due to its high-order polynomial equations. Our HK-CP is slower than OR-CP, but faster than PR-CP. Specifically, when our HK-CP uses the fully-orthogonal triplet, its efficiency approximates to that of OR-CP. When our HK-CP uses the partially-orthogonal or non-orthogonal triplet, it is slower than OR-CP. Still, it is faster than PR-CP thanks to lower-order polynomial equations.

7.5 Back-end

In this section, we conduct ablation study of our Kalman filter-based global DD refinement (see Section 6.1) and DD



(a) NEI (b) ALI (c) ALI-FIL (our)

Fig. 21. Trajectories estimated by baseline approaches and our method on the image sequence *Station_Road* of our CUHK-SLAM dataset. Pentagram denotes the starting point of trajectory. Color bar indicates the magnitude of the absolute trajectory error.

alignment-based camera pose optimization (see Section 6.2).

Evaluation Criteria. We use the absolute trajectory error introduced in Section 7.2 to evaluate the trajectory accuracy.

Methods for Comparison. We design two baseline approaches. Except for partial modules of back-ends, these approaches are the same as the proposed SLAM method. We compare these approaches with our method as follows:

- NEI: Baseline approach that uses neither of the filter-based global DD refinement and DD alignment-based camera pose optimization.
- ALI: Baseline approach that only exploits the DD alignment-based camera pose optimization.
- FIL+ALI: Our method that leverages both filter-based global DD refinement and DD alignment-based camera pose optimization.

All the above methods employ the DD-constrained bundle adjustment (see Section 6.2).

Experimental Results. As shown in Fig. 21 and Table 6, NEI results in significant error due to unrefined global DDs and lack of effective camera pose optimization. ALI reduces the error to some extent thanks to our DD alignment-based camera pose optimization. On partial sequences, our FIL+ALI is more accurate than ALI. Specifically, the estimated local DDs in the first camera frame are not accurate enough. Accordingly, the global DD initialization is unreliable. Our filter-based global DD refinement mitigates the effect of unreliable initialization, and thus improves the SLAM accuracy. On the other sequences, the accuracy improvement is limited since the global DD initialization is reliable.

8 Conclusions

In this paper, we propose a novel structural model called Hong Kong world to describe the structured scenes with vertical, horizontal, and sloping DDs. It is more general than Manhattan and Atlanta worlds, and also more compact and accurate than a mixture of independent Manhattan worlds. We further leverage the structural regularity of Hong Kong world for the line-based SLAM. Our SLAM method is reliable thanks to three technical novelties. First, our method to estimate DDs in Hong Kong world is the first one that can simultaneously determine the number of DDs, and achieve quasi-global optimality in terms of the number of inliers. Second, our camera pose estimation method exploits the spatial relations between DDs in Hong Kong world. It is more accurate and/or efficient than existing methods designed for unstructured scenes. Third, we refine DDs in Hong Kong world by a novel filter-based method. Then we use these refined DDs to optimize the camera poses and 3D lines, leading to higher accuracy and robustness than existing optimization algorithms. In addition, we establish the first dataset of sequential images in Hong Kong world. Experiments showed that our approach outperforms state-of-the-art methods in terms of accuracy and/or efficiency.

A main limitation of our method is that it can only handle structured environments. In an unknown environment, our method can automatically determine whether it can work or not. Specifically, on sequential images, if the regularity encoder always fails to detect DDs associated with sufficient inliers, our method can (indirectly) identify the scene as a non-structured environment. Accordingly, we can suppose that our method is unsuitable for this scene. In addition, we treat the extension to a visual-inertial odometry or multi-camera system as our future work.

Acknowledgments

Yun-Hui Liu was supported by the InnoHK of the Government of Hong Kong via the Hong Kong Centre for Logistics Robotics, the CUHK T Sone Robotics Institute, and the Shenzhen Portion of Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under HZQB-KCZYB-20200089. Kyungdon Joo was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00907, Development of AI Bots Collaboration Platform and Self-organizing AI and No.2020-0-01336, Artificial Intelligence Graduate School Program (UNIST)). Pyojin Kim was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1061397).

References

[1] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by Bayesian inference," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, 1999, pp. 941–947.

[2] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, 2004, pp. 203–209.

[3] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher, "The Manhattan frame model—Manhattan world inference in the space of surface normals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 235–249, 2017.

[4] K. Joo, T. Oh, J. Kim, and I. S. Kweon, "Robust and globally optimal manhattan frame estimation in near real time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 682–696, 2019.

[5] H. Li, Y. Xing, J. Zhao, J.-C. Bazin, Z. Liu, and Y.-H. Liu, "Leveraging structural regularity of Atlanta world for monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 2412–2418.

[6] Y. Gao and A. L. Yuille, "Exploiting symmetry and/or Manhattan properties for 3D object structure estimation from single and multiple images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6718–6727.

[7] A. Flint, D. Murray, and I. Reid, "Manhattan scene understanding using monocular, stereo, and 3D features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2228–2235.

[8] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 7247–7253.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Rob.*, 2015.

[11] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2006, pp. 3.1–3.10.

[12] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-D line-based map using stereo SLAM," *IEEE Trans. Rob.*, vol. 31, no. 6, pp. 1364–1377, 2015.

[13] Q. Wang, Z. Yan, J. Wang, F. Xue, W. Ma, and H. Zha, "Line flow based simultaneous localization and mapping," *IEEE Trans. Rob.*, vol. 37, no. 5, pp. 1416–1432, 2021.

[14] L. Xu, H. Yin, T. Shi, D. Jiang, and B. Huang, "EPLF-VINS: Real-time monocular visual-inertial SLAM with efficient point-line flow features," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 752–759, 2023.

[15] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1364–1375, 2015.

[16] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-inertial odometry with structural regularity of man-made environments," *IEEE Trans. Rob.*, vol. 35, no. 4, pp. 999–1013, 2019.

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[18] H. Li, J. Zhao, J.-C. Bazin, W. Chen, Z. Liu, and Y.-H. Liu, "Quasi-globally optimal and efficient vanishing point estimation in Manhattan world," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1646–1654.

[19] H. Li, J. Zhao, J.-C. Bazin, and Y.-H. Liu, "Quasi-globally optimal and near/true real-time vanishing point estimation in Manhattan world," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1503–1518, 2022.

[20] F. M. Mirzaei and S. I. Roumeliotis, "Globally optimal pose estimation from line correspondences," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011, pp. 5581–5588.

[21] A. Vakhitov, J. Funke, and F. Moreno-Noguer, "Accurate and linear time pose estimation from points and lines," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 583–599.

[22] C. Xu, L. Zhang, L. Cheng, and R. Koch, "Pose estimation from line correspondences: A complete analysis and a series of solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1209–1222, 2017.

[23] H. Li, J. Yao, J.-C. Bazin, X. Lu, Y. Xing, and K. Liu, "A monocular SLAM system leveraging structural regularity in Manhattan world," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 2518–2525.

[24] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Rob.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[25] R. Gomez-Ojeda, F. Moreno, D. Zuiga-Nol, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Rob.*, vol. 35, no. 3, pp. 734–746, 2019.

[26] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 4503–4508.

[27] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[28] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon, "Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment," *Int. J. Rob. Res.*, vol. 31, no. 1, pp. 63–81, 2012.

[29] H. Li, J. Zhao, J.-C. Bazin, W. Chen, K. Chen, and Y.-H. Liu, "Line-based absolute and relative camera pose estimation in structured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 6914–6920.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 1250–1257.

[32] R. Hartley and F. Kahl, "Global optimization through rotation space search," *Int. J. Comput. Vision*, vol. 82, no. 1, pp. 64–79, 2009.

[33] L. Magri and A. Fusiello, "T-Linkage: A continuous relaxation of J-Linkage for multi-model fitting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 3954–3961.

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

[35] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *J. Visual Commun. Image Represent.*, vol. 24, no. 7, pp. 794 – 805, 2013.

[36] H. Li, P. Kim, J. Zhao, K. Joo, Z. Cai, Z. Liu, and Y.-H. Liu, "Globally optimal and efficient vanishing point estimation in Atlanta world," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 153–169.

[37] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012.

[38] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Rob. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[39] C. V. Stewart, "MINPRAN: a new robust estimator for computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 925–938, 1995.

[40] W. Beyer, *CRC Standard Mathematical Tables*. CRC Press, 1987.

[41] J. Zhao, L. Kneip, Y. He, and J. Ma, "Minimal case relative pose computation using ray-point-ray features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1176–1190, 2020.

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

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

[44] R. Hartley, K. Aftab, and J. Trumpf, "L1 rotation averaging using the weiszfeld algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 3041–3048.

[45] M. Antunes, J. P. Barreto, D. Aouada, and B. Ottersten, "Unsupervised vanishing point detection and camera calibration from a single Manhattan image with radial distortion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6691–6699.

[46] H. Li, H. Yu, J. Wang, W. Yang, L. Yu, and S. Scherer, "ULSD: unified line segment detection across pinhole, fisheye, and spherical cameras," *ISPRS J. Photogramm. Remote Sens.*, vol. 178, pp. 187–202, 2021.

[47] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Rob.*, vol. 34, no. 4, pp. 1004–1020, 2018.

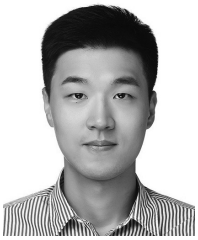
[48] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, 1991.

[49] P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating Manhattan frames in urban imagery," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 197–210.
[50] H. Li, K. Chen, P. Kim, K.-J. Yoon, Z. Liu, K. Joo, and Y.-H. Liu, "Learning icosahedral spherical probability map based on Bingham mixture model for vanishing point estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 5641–5650.
[51] J.-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in Manhattan world," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 638–645.
[52] H. Li, J. Zhao, J.-C. Bazin, and Y.-H. Liu, "Robust estimation of absolute camera pose via intersection constraint and flow consensus," *IEEE Trans. Image Process.*, vol. 29, pp. 6615–6629, 2020.

Haoang Li is a postdoctoral researcher at the Computer Vision Group, the Department of Informatics, Technical University of Munich, Germany. He received the B.E. and M.E. degrees from the School of Remote Sensing and Information Engineering, Wuhan University, China in 2016 and 2018, respectively, and the Ph.D. degree from the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China in 2022. He was a visiting Ph.D. student at the Computer Vision and Geometry Group, the Department of Computer Science, ETH Zurich, Switzerland in 2021. He won an outstanding reviewer award at CVPR 2021. His research interests include 3D computer vision and visual SLAM.

Ji Zhao received a B.S. degree in automation from Nanjing University of Posts and Telecommunication, Nanjing, China, in 2005 and a Ph.D. degree in control science and engineering from Huazhong University of Science and Technology, Wuhan, China, in 2012. From 2012 to 2014, he was a Post-Doctoral Research Associate at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include computer vision and machine learning.

Jean-Charles Bazin received the M.S. degree in Computer Science from the Université de Technologie de Compiègne, France in 2006, and the Ph.D. degree in Electrical Engineering from KAIST, South Korea in 2011. He is currently an Assistant Professor at KAIST. He worked as Associate Research Scientist at Disney Research Zurich, Switzerland (2014-2016). Before joining Disney Research, he was a postdoc jointly at the Computer Graphics Laboratory headed by Prof. M. Gross and the Computer Vision and Geometry Group headed by Prof. M. Pollefeys at ETH Zurich, Switzerland (2011-2014). He also worked as Postdoctoral Fellow at the Computer Vision Lab of Prof. K. Ikeuchi, University of Tokyo, Japan (2010-2011). He is an Area Chair for ICCV 2019 and CVPR 2020, and an invited AI expert in the World Economic Forum Expert.



Pyojin Kim is an assistant professor of the Department of Mechanical Systems Engineering at Sookmyung Women's University, South Korea. He received the B.S. degree in Mechanical Engineering from Yonsei University in 2013, and the M.S. and Ph.D. degrees in the Department of Mechanical and Aerospace Engineering at Seoul National University, Seoul, South Korea in 2015 and 2019, respectively. Before joining Sookmyung Women's University, he was a postdoctoral researcher at Simon Fraser University, Canada. He was a research intern at Google (ARCore Tracking), Mountain View in 2018. His research interests include indoor localization, 3D computer vision, visual odometry, and visual SLAM for robotics.

Kyungdon Joo is an assistant professor of the Artificial Intelligence Graduate School and the Department of Computer Science at UNIST, South Korea. He received the B.E. degree in School of Electrical and Computer Engineering from University of Seoul, South Korea in 2012, and the M.S. and Ph.D. degrees in Robotics Program and School of Electrical Engineering from KAIST, South Korea in 2014 and 2019, respectively. Before joining UNIST, he was a postdoctoral researcher at CMU RI, US. He was a member of "Team KAIST," which won the first place in DARPA Robotics Challenge Finals 2015. He was a research intern at Oculus research (Facebook Reality Labs), Pittsburgh in 2017. His research interests include robust computer vision, geometry and machine learning.

Zhenjun Zhao received the B.E. degree and the M.E. degree from the college of automation engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2012 and 2015, respectively. He is pursing the Ph.D. degree with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China. His research interests include 3D computer vision and visual SLAM.

Yun-Hui Liu received the B.E. degree in applied dynamics from the Beijing Institute of Technology, Beijing, China, the M.E. degree in mechanical engineering from Osaka University, Suita, Japan, and the Ph.D. degree in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1992. He was with the Electrotechnical Laboratory of Japan, Tsukuba, Japan, as a Research Scientist, then he joined The Chinese University of Hong Kong, Hong Kong, in 1995 and is currently a Choh-Ming Li Professor with the Department of Mechanical and Automation Engineering, the Director of the CUHK T-Stone Robotics Institute, and the Director of the Hong Kong Centre for Logistics Robotics. He has published over 500 papers in refereed journals and refereed conference proceedings. Dr. Liu was a recipient of the Highly Cited Author (Engineering) Award by Thomson Reuters in 2013 and numerous research awards from international journals and international conferences in robotics and automation and government agencies. He was the Editor-in-Chief of Robotics and Biomimetics and served as an Associate Editor of the IEEE Transactions on Robotics and Automation, and General Chair of IROS 2006.