

ORIGINAL ARTICLE

Open Access



Geotri-VIO: geometrically consistent multi-prism projection for panoramic imaging in visual-inertial-odometry

Ruidong Mei¹, Yihan Tu², Lanxiang Zheng², MingXin Wei³ and Hui Cheng^{2*}

Abstract

Panoramic camera-based Visual-Inertial-Odometry (VIO) systems play a crucial role in robotic navigation, autonomous driving, and virtual reality applications, owing to their large Field-of-View and enhanced localization capabilities. However, the nonlinear distortions caused by the lack of geometric consistency in projection models for panoramic images pose significant challenges to feature extraction and tracking algorithms. In this paper, we present Geotri-VIO, a novel VIO system that addresses these challenges using a multi-prism projection model. By constructing the multi-prism projection planes such that each face is tangent to the inherent projection sphere of the panoramic camera, the proposed model ensures strict geometric consistency in each projection plane while maintaining global geometric consistency, which is supported by mathematical proof. Additionally, we evaluate the impact of increasing the number of projection planes and demonstrate that triangular prism projection outperforms other multi-prism projection models. To validate its effectiveness, Geotri-VIO is tested on public datasets. Experimental results show that the triangular prism projection significantly improves the tracking accuracy of both point and line features, thereby enhancing the overall localization performance of the VIO system.

Keywords Panoramic camera, VIO, Triangular prism projection, Geometric consistency

Introduction

With the rapid advancements in fields such as robotic navigation, autonomous driving, Virtual Reality (VR) and Augmented Reality (AR), the need for precise localization and environmental perception is growing steadily (Hua et al., 2023; Li et al., 2021), particularly in Global Positioning System (GPS) denied environments (Cao et al., 2022). Visual-Inertial-Odometry (VIO) systems, which integrate visual and inertial data to deliver accurate pose

estimation, have become a significant area of research (Qin et al., 2018). While most existing studies focus on VIO systems utilizing pinhole camera models (Campos et al., 2021; Yang et al., 2023; Yan et al., 2024; El-Sheemy & Li, 2021), panoramic cameras (Gao et al., 2022; Wang et al., 2022) are gaining attention due to their large Field-of-View (FoV) and ability to capture rich environmental information (Wu et al., 2024; Li et al., 2024).

In VIO systems, geometric consistency plays a critical role. Geometric consistency refers to the requirement that the projection of image features and their spatial relationship in the physical scene remain consistent in pose estimation (Amato et al., 2011). For effective feature matching and optimization using visual and inertial data, the geometric structure of these data must align with the real-world spatial layout. However, many existing panoramic VIO algorithms utilize models such as equidistant cylindrical projection or omnidirectional

*Correspondence:

Hui Cheng
chengh9@mail.sysu.edu.cn

¹ School of Systems Science and Engineering, Sun Yat-Sen University, Guangzhou 511400, China

² School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou 511400, China

³ School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai 519082, China

projection, which abandon perspective projection to capture scene information. This approach compromises the proportional representation of real-world scenes, causing panoramic camera images to lose geometric consistency with the physical environment. The lack of geometric consistency causes nonlinear distortions, making it more difficult to extract feature points and line segments from panoramic images, reducing tracking accuracy, and ultimately negatively impacting the overall performance of VIO systems.

To address these challenges, this work proposes a novel multi-prism projection model for panoramic cameras and designs a panoramic VIO system, Geotri-VIO, that integrates both point features and combined point-line features to validate the performance of the proposed model. Specifically, we construct the multi-prism projection planes in such a way that each face is tangent to the inherent projection sphere of the panoramic camera. By calculating the azimuth and elevation angles of the points on the projection planes in the spherical coordinate system, pixel values are assigned. Mathematical proof demonstrates that in multi-prism projection, the geometric relationship of pixels within each plane strictly adheres to spatial consistency, preserving the relative positions of feature points and thereby improving the accuracy of feature extraction and tracking. Figure 1 shows the projection results of different models in the same scenario, including equidistant cylindrical projection, panoramic annular projection, the triangular prism projection from

the multi-prism projection model, and Cubemap projection. By examining the highlighted region (red box), it can be observed that equidistant cylindrical projection and panoramic annular projection exhibit significant distortions, causing the images to lose geometric consistency. Additionally, some continuous geometric structures in the real world become discontinuous after being mapped to the projection image through the Cube map projection model, which poses a significant challenge for feature-based VIO systems. In contrast, the triangular prism projection best preserves the linear structure and angular relationships, making the projection model most consistent with the geometric characteristics of the Three-Dimensional (3D) point cloud map.

To determine the optimal number of prism faces, this study systematically compares the projection characteristics of triangular, quadrilateral, pentagonal, and hexagonal prisms, extending the analysis to equidistant cylindrical projection. With rigorous mathematical verification of angular deviations in projection plane interfaces, the results demonstrate that triangular prism projection exhibits superior global geometric consistency compared to other polygonal prism projections. Theoretical analysis indicates that increasing the number of prism faces progressively approximates to equidistant cylindrical projection, leading to a reduction in global spatial consistency. Finally, experimental validation confirms these theoretical findings.

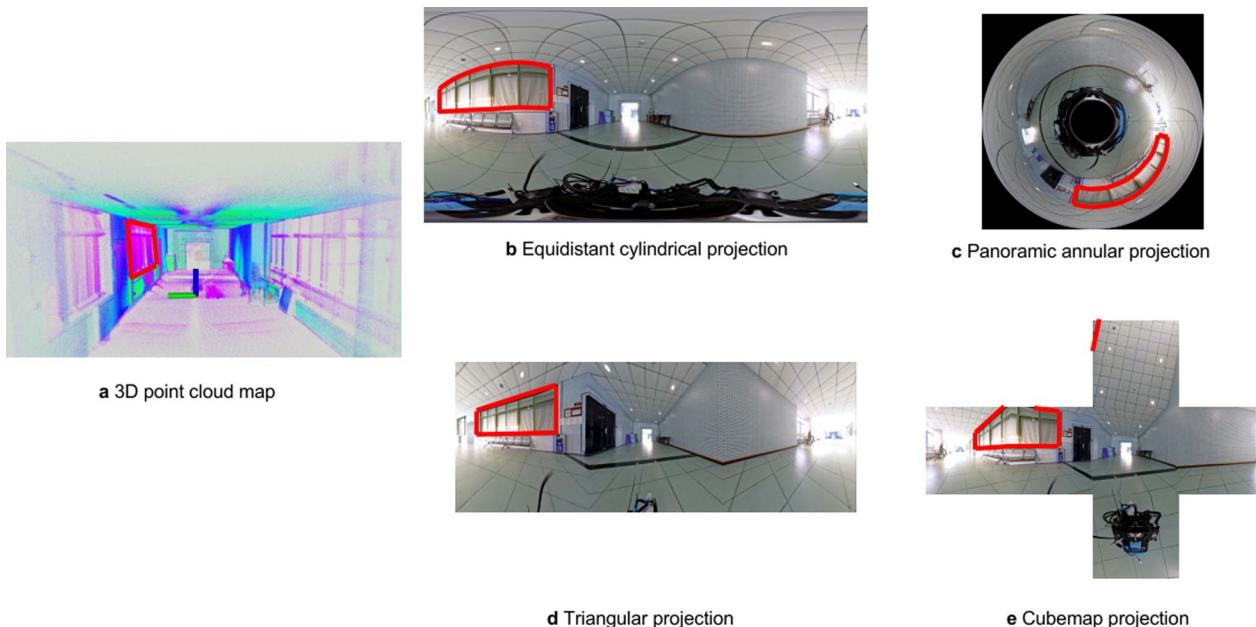


Fig. 1 Projection results of different models under the same scenario. **a** 3D point cloud map. **b** Equidistant cylindrical projection. **c** Panoramic annular projection. **d** Triangular projection. **e** Cubemap projection

In this paper, the main contributions and innovations are summarized as follows:

- A panoramic VIO framework, Geotri-VIO, is designed to enhance the utilization of visual information in panoramic images, improving feature matching and backend optimization accuracy, thereby boosting the overall performance of the VIO system.
- A novel multi-prism projection model is introduced, adaptable to various VIO frontend features, addressing the issue of geometric inconsistency in panoramic images. Theoretical analysis demonstrates the model's effectiveness in improving geometric consistency, and it is further validated that optimal performance is achieved by using a triangular prism projection.
- To validate the effectiveness of the Geotri-VIO framework, extensive experiments are conducted using both point features and point-line combined features. On public datasets, our algorithm demonstrated excellent performance in both front-end feature tracking accuracy and pose estimation precision.

Related work

This section reviews the panoramic camera projection models, point based panoramic VIO systems, and point and line based panoramic VIO systems. These methods illustrate the ability to integrate diverse frontend features within VIO systems, and offer various solutions to the challenges posed by panoramic cameras.

Panoramic camera projection models

Panoramic cameras have gained a widespread attention in the field of VIO due to their ability to capture a larger Field-of-View (FoV) and perceive a broader surrounding environment. However, panoramic cameras abandon perspective projection for capturing scene information, resulting in the images that fail to represent real world scenes. In other words, the panoramic camera images no longer maintain geometric consistency with the real-world scene. Currently, the panoramic camera projection models widely used in panoramic VIO systems include the omnidirectional projection model (Scaramuzza, Martinelli, and Siegwart, 2006), equidistant cylindrical projection model (Yang, 2021) and Cubemap projection model (Wang et al., 2018), as shown in Fig. 2. The panoramic annular image, calibrated with a

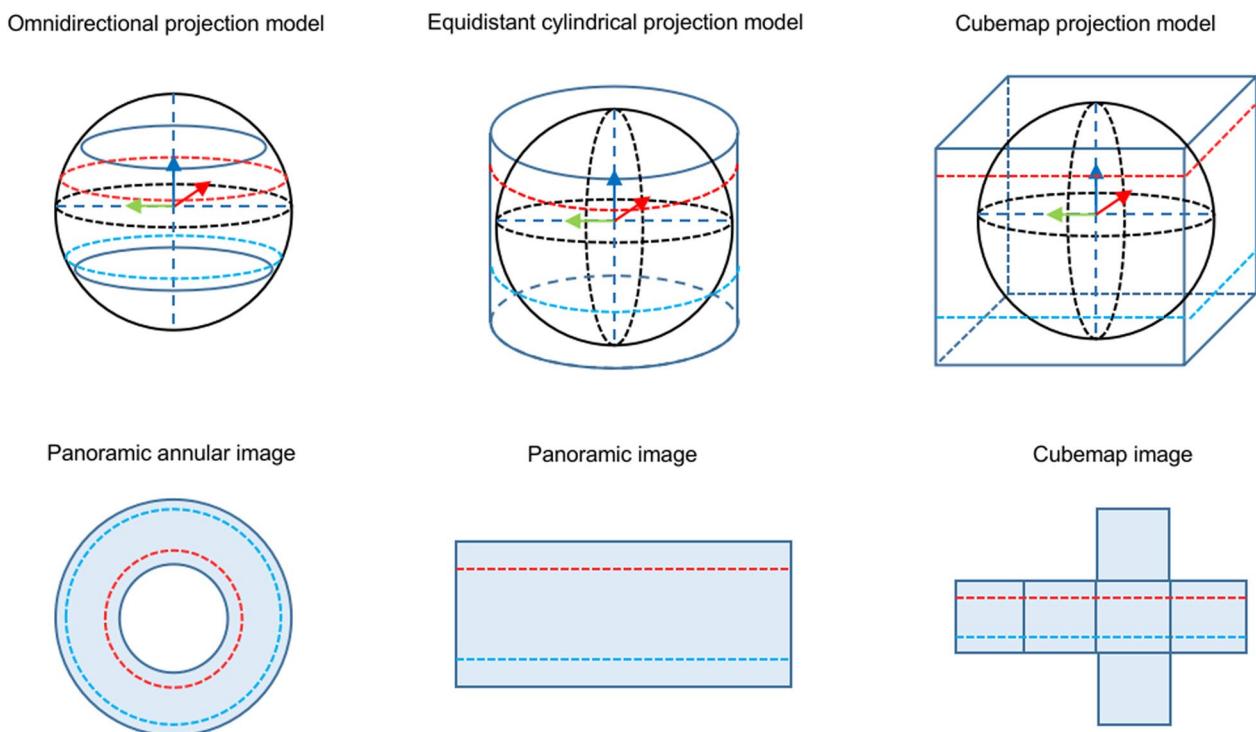


Fig. 2 Three commonly used projection models and corresponding projection images in VIO. This figure illustrates three commonly used projection models and their corresponding projection images in VIO. Omnidirectional projection model and panoramic annualr image. Equidistant cylindrical projection model and panoramic image. Cubemap projection model and Cubemap image

few parameters (Scaramuzza, Martinelli, and Siegwart, 2006), efficiently establishes pixel-to-scene correspondence, making it widely adopted in panoramic VIO systems (Wang et al., 2022; Wang et al., 2022; Wang et al., 2024; Wang et al., 2024). Nevertheless, this projection model completely disrupts the perspective relationship between the real-world scene and the pixel image, creating significant challenges for feature processing in VIO front-end systems. The panoramic image, generated with equidistant cylindrical projection (Yang, 2021), reduces the distortion at the image center and has been utilized in some Simultaneous Localization and Mapping (SLAM) frameworks (Huang & Yeung, 2022; Wu et al., 2024). This advantage, however, comes at the cost of increased distortion in the image edges, which poses additional difficulties for VIO front-end feature extraction and matching. Cubemap projection, introduced in Wang et al. (2018), models a large FoV camera as a combination of multiple pinhole cameras as shown in Fig. 2. The resulting Cubemap consists of undistorted sub-regions seamlessly stitched together, partially restoring the geometric consistency between the real-world scene and the image. While the approaches in Wang et al. (2018) and Xu et al. (2022) demonstrate promising results with this projection model, they lack a comprehensive analysis of the distortion issues occurring in the boundaries between sub-regions.

Point based panoramic VIO systems

Point based panoramic VIO methods utilize point features for localization and mapping. The extended perceptual range provided by panoramic cameras (Jiang et al., 2022, 2024; Yang et al., 2019) facilitates the extraction of a broader set of features, thereby improving the robustness of panoramic VIO systems. As a result, numerous visual odometry, VIO, and SLAM frameworks have been developed specifically for panoramic cameras. For instance, OpenVSLAM (Sumikura et al., 2019), a general-purpose SLAM framework, employs sparse ORB features (Rublee et al., 2011) in its front-end processing and supports panoramic camera. Cubemap-SLAM (Y. Wang et al., 2018) redesigns the ORB-SLAM (Mur-Artal et al., 2015) framework for piecewise-pinhole monocular fisheye cameras, optimizing feature extraction across sub-regions. PAL-SLAM (Wang et al., 2022) and 360ORB-SLAM (Chen et al., 2024) employ ORB (Mur-Artal, Montiel, and Tardos, 2015) feature points for 360° camera SLAM systems. Based on PAL-SLAM (Wang et al., 2022), the recent PAL-SLAM2 (Wang et al., 2024) improves feature tracking in the negative hemisphere of 360° cameras, addressing the limitations of traditional frameworks and effectively handling rapid turns, low-light conditions, and sudden lighting changes. Similarly,

LF-VIO (Wang et al., 2022), developed based on the VINS-MONO (Qin et al., 2018) framework, presents a VIO system specifically designed for 360° cameras, addressing the challenges of feature point tracking and processing in the negative hemisphere. More recently, LF-VISLAM (Wang et al., 2024) is proposed as an extension of LF-VIO (Wang et al., 2022), incorporating a loop closure module for 360° cameras based SLAM system, which further enhances the system's accuracy and robustness. Despite their utility, these approaches rely on traditional feature detection, description, and matching pipelines. The inherent lack of geometric consistency in panoramic images results in significant changes in feature descriptors under varying viewpoints, reducing matching performance. Methods based on optical flow also suffer from similar geometric consistency issues, limiting their effectiveness.

Point and line based panoramic VIO systems

Point and line based VIO methods enhance system performance by integrating line features into the front-end and utilizing line constraints in back-end optimization. As a result, the performance of line feature detection and tracking is critical to the overall efficiency of these systems. Early point and line based VIO systems, such as PL-VIO (He, Zhao, Guo, He, and Yuan, 2018) and PL-VINS (Fu et al., 2020), typically employed traditional line detection algorithms like LSD (Von Gioi et al., 2012) and ELSED (Suárez, Buenaposada, and Baumela, 2022), which rely on local contour-based approaches. The detected line features are commonly described and matched using the Line-Band-Descriptor (LBD) descriptor (Zhang & Koch, 2013) in early point and line based VIO system. In recent years, learning-based methods for line feature detection, such as SOLD2 (Pautrat et al., 2021) and DeepLSD (Pautrat et al., 2023), were proposed, leveraging deep neural networks to improve detection accuracy and robustness. Recently, AirSLAM (Xu et al., 2024) proposed PLNet, a learning-based network capable of real-time line detection and matching. However, these methods rely on input images with geometric consistency and are thus not directly applicable to panoramic VIO systems. Unified Line Segment Detection (ULSD) (Li et al., 2021), another learning-based line detection method, is specifically designed for detecting line segments in both distorted and undistorted images, making it particularly well-suited for panoramic cameras. However, ULSD focuses solely on line detection and lacks the functionality for matching and tracking detected lines. While learning-based methods offer higher flexibility in feature extraction and can perform better in extremely complex scenarios, they often rely on Graphics Processing Unit (GPU) acceleration and incur significant

computational overhead, making it challenging to ensure real-time performance. More recently, LF-PGVIO (Wang et al., 2024) recombined the LBD descriptor and then proposed the RLBD descriptor, which is used for describing and matching line segments in panoramic images. Although this approach addresses some challenges, the lack of geometric consistency in panoramic images often leads to RLBD descriptor instability under varying viewpoints.

In summary, most panoramic VIO algorithms, whether point based or point and line based, rely on the geometric consistency of the input images in the front-end feature processing (as summarized in Table 1). However, these algorithms do not fully account for this issue. Motivated by this, we propose a novel panoramic VIO framework, where the front-end introduces a multi-prism projection model to achieve geometric consistency.

Methodology

This section details the proposed multi-prism projection algorithm and proves its geometric consistency. Additionally, the proposed VIO framework is introduced, which is adaptable to both point features and combined point-line features. The framework is illustrated in Fig. 3. The overall framework consists of three main components: projection, point based VIO, and point and line based VIO. The projection component is discussed in Section Geometric consistency and Section Multi-prism projection, which detail the projection model and prove its geometric consistency. The point based VIO component is in Section Point based VIO, describing the process from front-end feature extraction to back-end residual construction and optimization. Finally, the point and line based VIO component in Section Point and line based VIO extends the framework by integrating line features, detailing the process from feature extraction to residual construction and optimization for enhanced robustness.

Table 1 The Importance of Geometric Consistency in Different Feature Processing Algorithms

Algorithm type	Requires geometric consistency
Harris	Yes
ORB	Yes
LK	Yes
BRIEF	Yes
LSD	Yes
ELSED	Yes
LBD	Yes

Geometric consistency

Geometric consistency can be assessed by examining whether the partial derivatives of the image coordinates with respect to the spatial coordinates exhibit linear behavior under constant depth conditions. The mathematical proof is as follows.

For panoramic images with equidistant cylindrical projection, the projection relationship between pixel coordinates (u, v) and 3D spatial coordinates (X, Y, Z) is as:

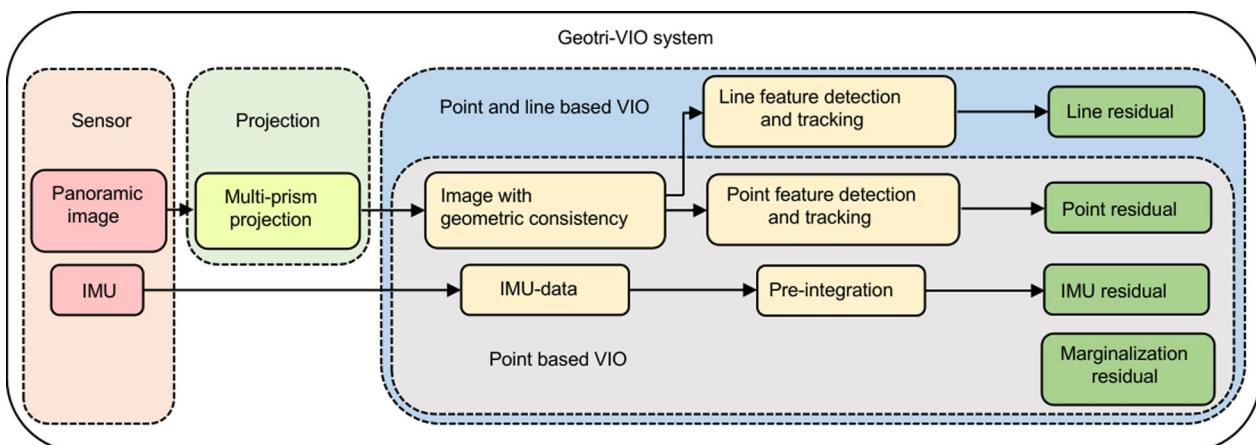


Fig. 3 The framework of Geotri-VIO compatible with both point features and combined point-line features. Projection, point based VIO, and point and line based VIO are the main modules of Geotri-VIO, represented by light green, gray, and blue boxes, respectively

$$\begin{cases} \phi = \arctan \frac{Y}{X} \\ \theta = \arctan \frac{Z}{\sqrt{X^2+Y^2}} \\ u = k_\phi \phi \\ v = k_\theta \theta \end{cases} \quad (1)$$

where k_ϕ and k_θ are scaling factors that denote the number of pixels per radian in the latitude and longitude directions, respectively. For $u = k_\phi \arctan(\frac{Y}{X})$, the partial derivatives with respect to X and Y are derived as follows:

$$\frac{\partial u}{\partial X} = k_\phi \cdot \frac{\partial \phi}{\partial X} = k_\phi \cdot \left(-\frac{Y}{X^2 + Y^2} \right) \quad (2)$$

$$\frac{\partial u}{\partial Y} = k_\phi \cdot \frac{\partial \phi}{\partial Y} = k_\phi \cdot \frac{X}{X^2 + Y^2} \quad (3)$$

From Eq (1): $v = k_\theta \arctan\left(\frac{Z}{\sqrt{X^2+Y^2}}\right)$, the partial derivatives with respect to X , Y and Z are:

$$\frac{\partial v}{\partial X} = k_\theta \cdot \frac{-Z \cdot X}{(X^2 + Y^2)^{3/2}} \quad (4)$$

$$\frac{\partial v}{\partial Y} = k_\theta \cdot \frac{-Z \cdot Y}{(X^2 + Y^2)^{3/2}} \quad (5)$$

$$\frac{\partial v}{\partial Z} = k_\theta \cdot \frac{1}{\sqrt{X^2 + Y^2}} \quad (6)$$

These derivatives vary significantly depending on the position in the image, particularly near the poles or far from the center, where the nonlinearity becomes pronounced. This positional dependence disrupts the uniformity of 3D spatial relationships between pixels, leading to geometric inconsistency. Consequently, such distortions affect the reliability of feature extraction, matching, and motion estimation in VIO systems.

Multi-prism projection

The proposed multi-prism projection model better preserves geometric consistency across different regions of the panoramic image. Taking the triangular prism projection as an example, pixel values from the panoramic image are mapped to the projection image by calculating their azimuth and elevation angles in the projection sphere's coordinate system. As illustrated in Fig. 4, the detailed proof of this process is provided below.

For a spatial point $P(X, Y, Z)$, it is first mapped to the image center coordinate system \hat{O} of the projection plane, with \hat{O} representing the center of the image, using the following formula:

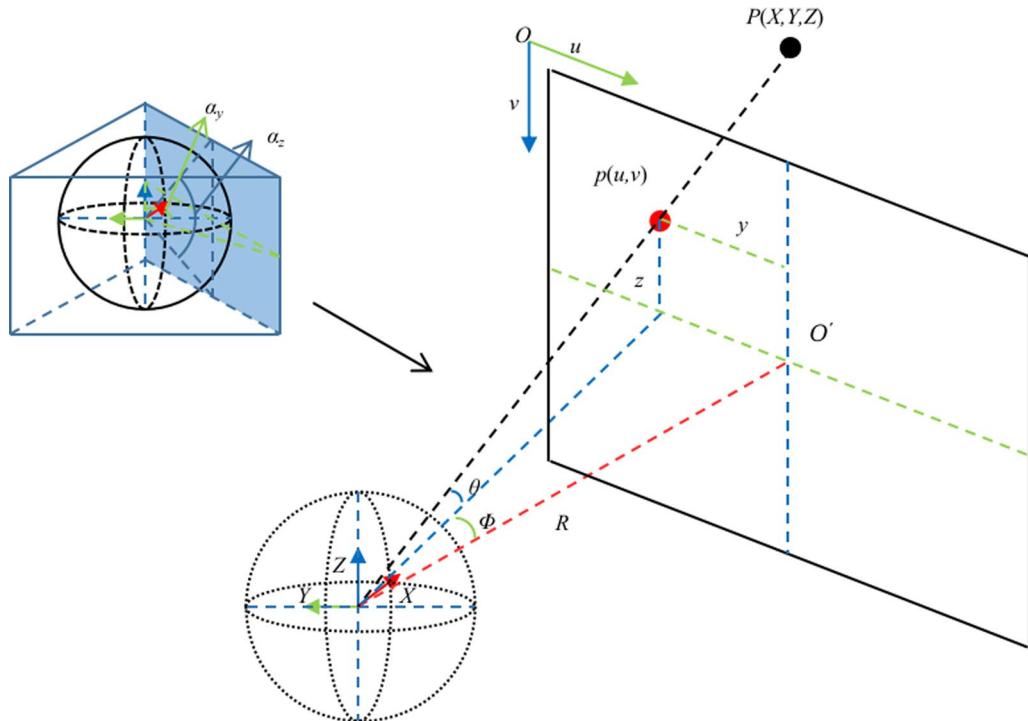


Fig. 4 Illustration of triangular prism projection for panoramic cameras. This figure illustrates the triangular prism projection for panoramic cameras. The top-left inset shows a 3D schematic of the projection sphere inscribed within a triangular prism. The main diagram highlights the 3D to 2D projection process

$$y = R \cdot \frac{Y}{X} \quad (7)$$

$$z = R \cdot \frac{Z}{X} \quad (8)$$

$$R = \frac{2 \times \tan(\alpha_y/2)}{W} = \frac{2 \times \tan(\alpha_z/2)}{H} \quad (9)$$

where W and H represent the horizontal and vertical resolutions of the projection plane, and α_y and α_z denote the horizontal and vertical viewing angles, respectively. R is the projection radius. Then convert y and z to the point $p(u, v)$ in the coordinate system O :

$$u = y + \frac{W}{2} = R \cdot \frac{Y}{X} + \frac{W}{2} \quad (10)$$

$$v = z + \frac{H}{2} = R \cdot \frac{Z}{X} + \frac{H}{2} \quad (11)$$

Finally, the azimuth angle ϕ and elevation angle θ of the point $P(X, Y, Z)$ in the projection spherical coordinate system are computed, and the pixel values are assigned to $p(u, v)$ accordingly:

$$\phi = \arctan\left(\frac{y}{R}\right) \quad (12)$$

$$\theta = \arctan\left(\frac{z}{\sqrt{y^2 + R^2}}\right) \quad (13)$$

Once the panoramic camera parameters are fixed, the mapping relationship for the projection can be computed and stored. During runtime, the projection process simply looks up the computed mapping table, which has a time complexity of $O(1)$ per pixel. This efficient implementation ensures that the computational overhead of the projection process is negligible compared to other components of the system.

The rate of change of the new projection plane can be computed by calculating the partial derivatives of the pixel coordinates with respect to the spatial coordinates. Since the projection plane's depth direction is aligned with the X -axis, the partial derivatives of u with respect to Y and Z are calculated as follows:

$$\frac{\partial u}{\partial Y} = \frac{R}{X} \quad (14)$$

$$\frac{\partial u}{\partial Z} = 0 \quad (15)$$

Similarly, for v :

$$\frac{\partial v}{\partial Y} = 0 \quad (16)$$

$$\frac{\partial v}{\partial Z} = \frac{R}{X} \quad (17)$$

The rates of change, represented by $\frac{\partial u}{\partial Y}$, $\frac{\partial u}{\partial Z}$, $\frac{\partial v}{\partial Y}$ and $\frac{\partial v}{\partial Z}$, follow linear transformations. Consequently, the triangular prism projection achieves local geometric consistency within each projection plane.

Although this work has demonstrated that each projection plane in the multi-prism projection exhibits local geometric consistency, such consistency does not hold in the boundaries between adjacent projection planes. Notably, the degree of global geometric consistency varies with the number of prism faces. To determine the optimal number of prism faces, this paper conducts a rigorous mathematical derivation to quantify the global geometric consistency of the multi-prism projection.

As shown in Fig. 5, a spatial line L is projected onto the projection plane $\Pi(1)$ as $l(1)$ and onto $\Pi(2)$ as $l(2)$. Subsequently, $\Pi(1)$ and $\Pi(2)$ are stitched together to form the two-dimensional image planes $\Pi'(1)$ and $\Pi'(2)$, which simulates the imaging process of multi-prism projection. For clarity, we denote the Two-Dimensional (2D) image planes as $\Pi'(i)$ and the 2D lines on these planes as $l'(i)$. It can be observed that when the projection $l(1)$ of the space line L crosses the intersection line $l(3)$ between $\Pi(1)$ and $\Pi(2)$, it undergoes a deflection. As a result, an angle θ is formed between $l'(1)$ and $l'(2)$ in the image plane, reflecting the geometric inconsistency between the projection planes.

To compute θ , we first calculate the cross product of the normal vectors of $\Pi(1)$ and $\Pi(3)$, as well as $\Pi(2)$ and $\Pi(3)$, to obtain the direction vectors of the intersection lines $l(1)$ and $l(2)$, where $\Pi(3)$ is the plane formed by the spatial line L and the camera center O . This paper uses $\mathbf{n}_{\Pi(i)}$ to represent the normal vector of a plane $\Pi(i)$ and $\mathbf{v}_{l(i)}$ to denote the direction vector of a line $l(i)$:

$$\begin{cases} \mathbf{v}_{l(1)} = (\mathbf{n}_{\Pi(3)} \hat{\times} \mathbf{n}_{\Pi(1)}) \\ \mathbf{v}_{l(2)} = (\mathbf{n}_{\Pi(3)} \hat{\times} \mathbf{n}_{\Pi(2)}) \end{cases} \quad (18)$$

Where $(\hat{\cdot})$ denotes the normalization operation of a vector. Since $l(1)$ and $l(2)$ are 3D projection lines, the angle between them cannot be directly used to represent the angle θ between $l'(1)$ and $l'(2)$ in the 2D image plane. Therefore, we compute the angles between $l(1)$ and $l(3)$, as well as $l(2)$ and $l(3)$, to indirectly determine θ :

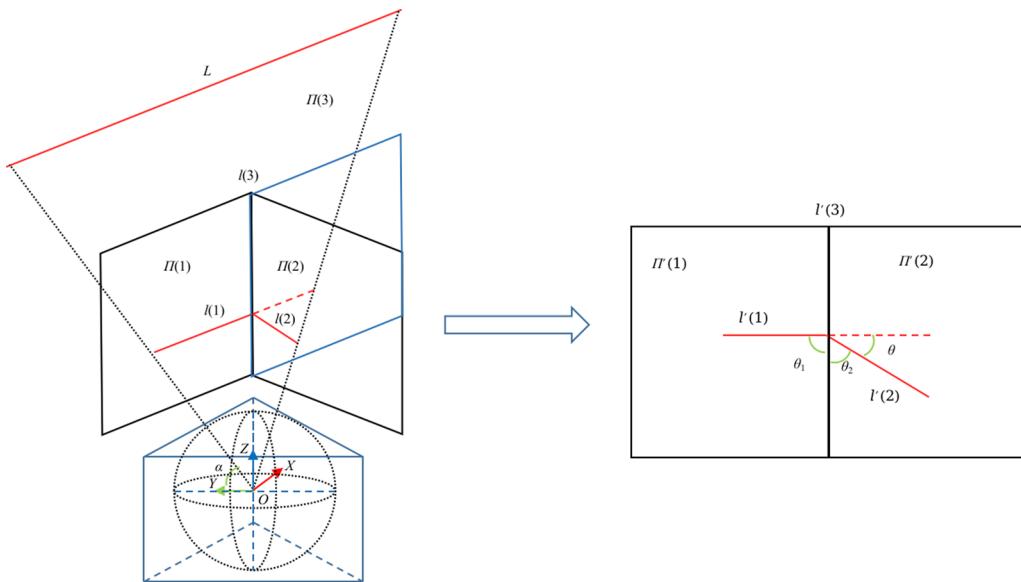


Fig. 5 Geometric consistency analysis of multi-prism projection. The left figure illustrates the process of projecting a spatial line onto multiple projection planes, while the right figure shows the deflection of the line in the 2D image plane

$$\begin{cases} \theta_1 = \arccos \frac{\mathbf{v}_{l(1)} \cdot \mathbf{v}_{l(3)}}{\|\mathbf{v}_{l(1)}\| \cdot \|\mathbf{v}_{l(3)}\|} \\ \theta_2 = \arccos \frac{\mathbf{v}_{l(2)} \cdot \mathbf{v}_{l(3)}}{\|\mathbf{v}_{l(2)}\| \cdot \|\mathbf{v}_{l(3)}\|} \\ \theta = (\pi - \theta_1) - \theta_2 \end{cases} \quad (19)$$

where $l(3)$ can be obtained by computing the cross product of the normal vectors of $\Pi(1)$ and $\Pi(2)$:

$$\mathbf{v}_{l(3)} = (\mathbf{n}_{\Pi(1)} \hat{\times} \mathbf{n}_{\Pi(2)}) \quad (20)$$

Noting that the intrinsic geometric inconsistency between adjacent projection planes depends only on their relative angles and is independent of external factors. We assume without loss of generality that L is parallel to both the plane Π_{XOY} and the plane $\Pi(1)$ to simplify the mathematical proof. In this case, $\mathbf{n}_{\Pi(3)}$ and \mathbf{v}_L satisfy the constraint:

$$\begin{cases} \mathbf{v}_L \cdot \mathbf{n}_{\Pi_{XOY}} = 0 \\ \mathbf{v}_L \cdot \mathbf{n}_{\Pi(1)} = 0 \end{cases} \quad (21)$$

moreover, θ_1 satisfies:

$$\theta_1 = \frac{\pi}{2} \quad (22)$$

Similarly, assuming that the number of faces in the multi-prism projection is k and the angle between $\Pi(3)$ and Π_{XOY} is α , then $\Pi(1)$, $\Pi(2)$ and $\Pi(3)$ satisfy the constraint:

$$\begin{cases} \arccos \frac{\mathbf{n}_{\Pi(1)} \cdot \mathbf{n}_{\Pi(2)}}{\|\mathbf{n}_{\Pi(1)}\| \cdot \|\mathbf{n}_{\Pi(2)}\|} = \pi - \frac{2\pi}{k}, k \in \mathbb{Z}_{\geq 3}^+ \\ \arccos \frac{\mathbf{n}_{\Pi(3)} \cdot \mathbf{n}_{\Pi_{XOY}}}{\|\mathbf{n}_{\Pi(3)}\| \cdot \|\mathbf{n}_{\Pi_{XOY}}\|} = \alpha \end{cases} \quad (23)$$

where, $\frac{2\pi}{k}$ represents the angle between the planes $\Pi(1)$ and $\Pi(2)$, and $\mathbf{n}_{\Pi_{XOY}}$ is aligned with the Z -axis. α denotes the angle between the planes $\Pi(3)$ and $\mathbf{n}_{\Pi_{XOY}}$. In this proof, we restrict our analysis to the case where $\alpha \in [0, \pi/2]$ due to the symmetry of the projection planes with respect to Π_{XOY} .

By solving the system of equations Eqs. (18)–(23), θ can be expressed as a function of k and α :

$$\theta(k, \alpha) = \arcsin \left[\sin \left(\frac{2\pi}{k} \right) \sin(\alpha) \right] \quad (24)$$

Since there are $k - 1$ intersection lines between projection planes when the number of projection planes is k , this work constructs the function I to represent the global geometric inconsistency of the multi-prism projection for different values of k :

$$I(k, \alpha) = \arcsin \left[\sin \left(\frac{2\pi}{k} \right) \sin(\alpha) \right] \cdot (k - 1) \quad (25)$$

where, for a fixed α , $I(k, \alpha)$ is monotonically increasing for $k \in \mathbb{Z}_{\geq 3}^+$, which implies that I attains its minimum value when $k = 3$. This indicates that the global geometric consistency of the triangular prism projection model is the highest.

Figure 6 illustrates the theoretical derivation described above. As the number of projection planes increases, the size of each individual plane decreases, the boundary regions expand, and global geometric consistency deteriorates. In the limiting case, where the multi-prism projection approaches the equidistant

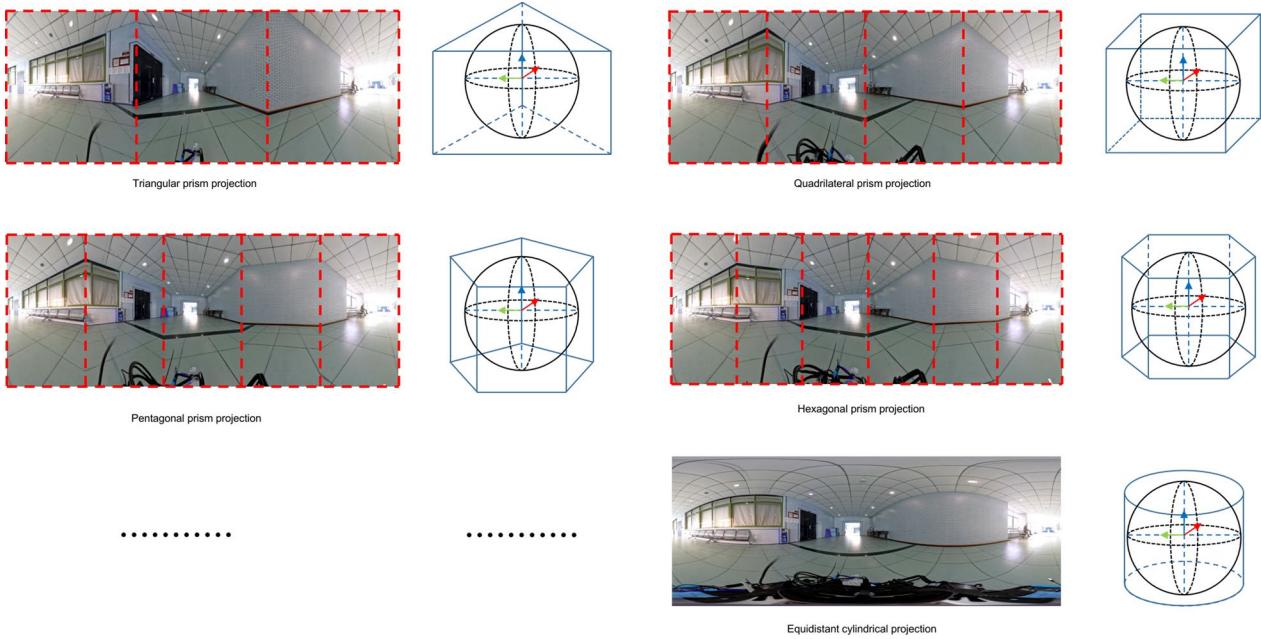


Fig. 6 Projection images of models with varying numbers of projection planes. This figure illustrates how the triangular prism projection image progressively converges to the equidistant cylindrical projection image as the number of projection planes increases

cylindrical projection, geometric consistency is entirely lost. In the subsequent sections, the triangular prism projection is employed as a representative example of multi-prism projections and is utilized for the extraction and matching of both point and line features.

The relationship between the number of projection planes and geometric consistency will be experimentally validated in Section Experiments.

Point based VIO

As mentioned in Section Multi-prism projection, after completing the multi-prism projection, the front-end image achieves geometric consistency. This means that the feature detection and tracking methods commonly used in traditional VIO systems with pinhole cameras can be directly applied to this framework.

To establish the correspondences between different image frames, we first extract Shi-Tomasi corners (Shi et al., 1994) and then track them using the Lucas-Kanade method (Lucas & Kanade, 1981). Next, we apply a two-step geometric outlier rejection process with Random Sample Consensus (RANSAC), as described in Qin et al. (2018), to enhance robustness. For the back-end optimization, to accommodate the large FoV panoramic cameras, we use a similar panoramic VIO work (Z. Wang et al., 2022) to construct the optimization objective function. In the point based VIO system, the state vector variables are shown as:

$$\begin{cases} \boldsymbol{\chi} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N, \boldsymbol{t}_{c,0}^{(B)}, \dots, \boldsymbol{t}_{c,N}^{(B)}, \lambda_1, \dots, \lambda_K] \\ \boldsymbol{x}_n = [\boldsymbol{P}_{B,n}^{(W)}, \boldsymbol{V}_{B,n}^{(W)}, \boldsymbol{q}_{B,n}^{(W)}, \boldsymbol{b}_a, \boldsymbol{b}_g], n \in [0, N] \\ \boldsymbol{t}_{c,i}^{(B)} = [\boldsymbol{P}_{c,i}^{(B)}, \boldsymbol{q}_{c,i}^{(B)}] \end{cases} \quad (26)$$

Here, \boldsymbol{x}_n represents the state of the body in the n -th sliding window. It consists of the position $\boldsymbol{P}_{B,n}^{(W)}$, velocity $\boldsymbol{V}_{B,n}^{(W)}$, orientation quaternion $\boldsymbol{q}_{B,n}^{(W)}$, and the accelerometer and gyroscope biases, denoted as \boldsymbol{b}_a and \boldsymbol{b}_g , respectively. The variable $\boldsymbol{t}_{c,i}^{(B)}$ represents the transformation from the camera coordinate system c_i to the body coordinate system. This transformation includes the position $\boldsymbol{P}_{c,i}^{(B)}$ and the orientation quaternion $\boldsymbol{q}_{c,i}^{(B)}$. The variable λ_K denotes the inverse distance of the k -th feature point relative to its first observation on the unit sphere. N refers to the total number of sliding windows used in the algorithm, while K represents the number of feature points.

By utilizing the state variables defined in Eq. (26), the overall optimization objective for the point based VIO system is formulated as:

$$\begin{aligned} \min_{\boldsymbol{\chi}} \left\{ \| \boldsymbol{r}_p - \boldsymbol{H}_p \boldsymbol{\chi} \|^2 + \sum_{k \in B} \| \boldsymbol{r}_B(\hat{\boldsymbol{z}}_{b,k+1}^{b,k} \boldsymbol{\chi}) \|^2 \right. \\ \left. + \sum_{(i,j) \in C} \rho_p \| \boldsymbol{r}_C(\hat{\boldsymbol{z}}_{p,c,j}^{(c,j)}, \boldsymbol{\chi}) \|^2 \right\} \end{aligned} \quad (27)$$

where χ is the optimization variable. \mathbf{r}_p , \mathbf{r}_B and \mathbf{r}_C represent the residuals for marginalization, IMU pre-integration and point features, respectively. B is the set of all pre-integrated IMU measurements within the sliding window, while C denotes the sets of point measurements from the observed frames. We use the Ceres Solver to solve the nonlinear maximum of the a posteriori estimation problem and employ a Huber loss function (Huber, 1992) to mitigate the influence of outliers, thereby enhancing the system's robustness.

Point feature residual: The input to the front-end is the multi-prism projection image, which achieves geometric consistency and enhances the accuracy of feature detection and tracking. In the back-end optimization, to support the large FoV of the panoramic camera, the point features are projected onto the unit spherical surface using the following projection function:

$$\begin{cases} \alpha = \cos(\theta) \cdot \cos(\phi) \\ \beta = \cos(\theta) \cdot \sin(\phi) \\ \gamma = \sin(\theta) \end{cases} \quad (28)$$

where, α , β and γ represent the coordinates of the feature points on the unit spherical surface, θ and ϕ represent the elevation angle and azimuth angle of the point on the unit sphere, respectively. As mentioned in Section Multi-prism projection, θ and ϕ can be derived from the feature point coordinates (u, v) in the triangular prism projection image using Eqs. (7)-(13). Then denote the transformation from (u, v) to (α, β, γ) as π_c^{-1} .

The point feature reprojection error is computed according to the following equation:

$$\begin{cases} \mathbf{r}_C(\hat{\mathbf{z}}_p^{(c,j)}, \chi) = (\mathbf{b}_1, \mathbf{b}_2)^T \cdot \left(\hat{\mathbf{P}}_l^{(c,j)} - \frac{\mathbf{P}_l^{(c,j)}}{\|\mathbf{P}_l^{(c,j)}\|} \right) \\ \hat{\mathbf{P}}_l^{(c,j)} = \pi_c^{-1} \cdot (\hat{u}_l^{(c,j)}, \hat{v}_l^{(c,j)})^T \\ \mathbf{P}_l^{(c,j)} = \mathbf{R}_b^{(c)} \left\{ \mathbf{R}_{\omega}^{(b,j)} \left\{ \mathbf{R}_{(b,i)}^{(\omega)} \left[\mathbf{R}_c^{(b)} \cdot \frac{1}{\lambda_d} \cdot \pi_c^{-1} \cdot (\hat{u}_l^{(c,i)}, \hat{v}_l^{(c,i)})^T + \mathbf{P}_c^{(b)} \right] \right. \right. \\ \left. \left. + \mathbf{P}_{(b,i)}^{(\omega)} - \mathbf{P}_{(b,j)}^{(\omega)} \right\} - \mathbf{P}_c^{(b)} \right\} \end{cases} \quad (29)$$

where $(\hat{u}_l^{(c,i)}, \hat{v}_l^{(c,i)})$ is the first observation of the l -th feature that happens in the i -th image. $(\hat{u}_l^{(c,i)}, \hat{v}_l^{(c,i)})$ is the observation of the same feature in the j -th image. \mathbf{b}_1 and \mathbf{b}_2 are two arbitrarily selected orthogonal bases which span the tangent plane of $\hat{\mathbf{P}}_l^{(c,j)}$.

Point and line based VIO

Since the input triangular prism projection image is geometrically consistent, the line feature detection and tracking methods commonly used in traditional pinhole

camera SLAM systems, such as He et al. (2018) and Fu et al. (2020), can be directly applied. In this work, line features are detected using ELSED and tracked with LBD descriptors. Additionally, 3D lines are represented and computed using Plücker coordinates (G. Zhang, Lee, Lim, and Suh, 2015).

Before introducing the back-end optimization, the Plücker coordinate system is briefly overviewed, which provides an intuitive and elegant representation of 3D lines. In this coordinate system, a line is expressed as $\mathbf{L}(\mathbf{n}, \mathbf{d}) \in \mathbb{R}^6$, where \mathbf{n} and \mathbf{d} denote the normal and direction vectors, respectively. This representation simplifies the numerical computation processes of 3D line triangulation and reprojection.

However, during the back-end optimization in VIO, the Plücker coordinate introduces a hyperparameter issue, as it represents a 3D line with 6 Degrees of Freedom (DOF), but only 4-DOF are required. To address this, an orthonormal representation with only 4-DOF is adopted to represent 3D lines in the optimization process. This approach has demonstrated a good convergence in previous works (Fu et al., 2020; G. Zhang, Lee, Lim, and Suh, 2015).

The orthonormal representation can be expressed as follows:

$$\mathbf{o} = [\psi, \phi] \quad (30)$$

where ψ is a rotation matrix representing the rotation of the line relative to the camera coordinate system, and ϕ is a scalar representing the minimal distance from the center of the panoramic camera to the line.

In the point and line based VIO system, the state vector is defined as:

$$\begin{cases} \chi = [\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{t}_{(c,0)}^{(B)}, \dots, \mathbf{t}_{(c,N)}^{(B)}, \lambda_1, \dots, \lambda_K, \mathbf{o}_1, \dots, \mathbf{o}_J] \\ \mathbf{x}_n = [\mathbf{P}_{B,n}^{(W)}, \mathbf{V}_{B,n}^{(W)}, \mathbf{q}_{B,n}^{(W)}, \mathbf{b}_a, \mathbf{b}_g], n \in [0, N] \\ \mathbf{t}_{c,i}^{(B)} = [\mathbf{P}_{c,i}^{(B)}, \mathbf{q}_{c,i}^{(B)}] \\ \mathbf{o}_j = [\psi_j, \phi_j], j \in [0, J] \end{cases} \quad (31)$$

Compared to Eq. (26), the above equation's χ includes an additional term, \mathbf{o}_j , which represents the orthonormal representation of the j -th 3D straight line.

Using Eq. (31), a new optimization objective function is constructed:

$$\begin{aligned} \min_{\chi} & \left\{ \|\mathbf{r}_p - \mathbf{H}_p \chi\|^2 + \sum_{k \in B} \left\| \mathbf{r}_B(\hat{\mathbf{z}}_{b,k+1}^{b,k}, \chi) \right\|^2 \right. \\ & \left. + \sum_{(i,j) \in C} \rho_p \left\| \mathbf{r}_C(\hat{\mathbf{z}}_p^{c,j}, \chi) \right\|^2 + \sum_{(i,j) \in L} \rho_l \left\| \mathbf{r}_L(\hat{\mathbf{z}}_l^{c,j}, \chi) \right\|^2 \right\} \end{aligned} \quad (32)$$

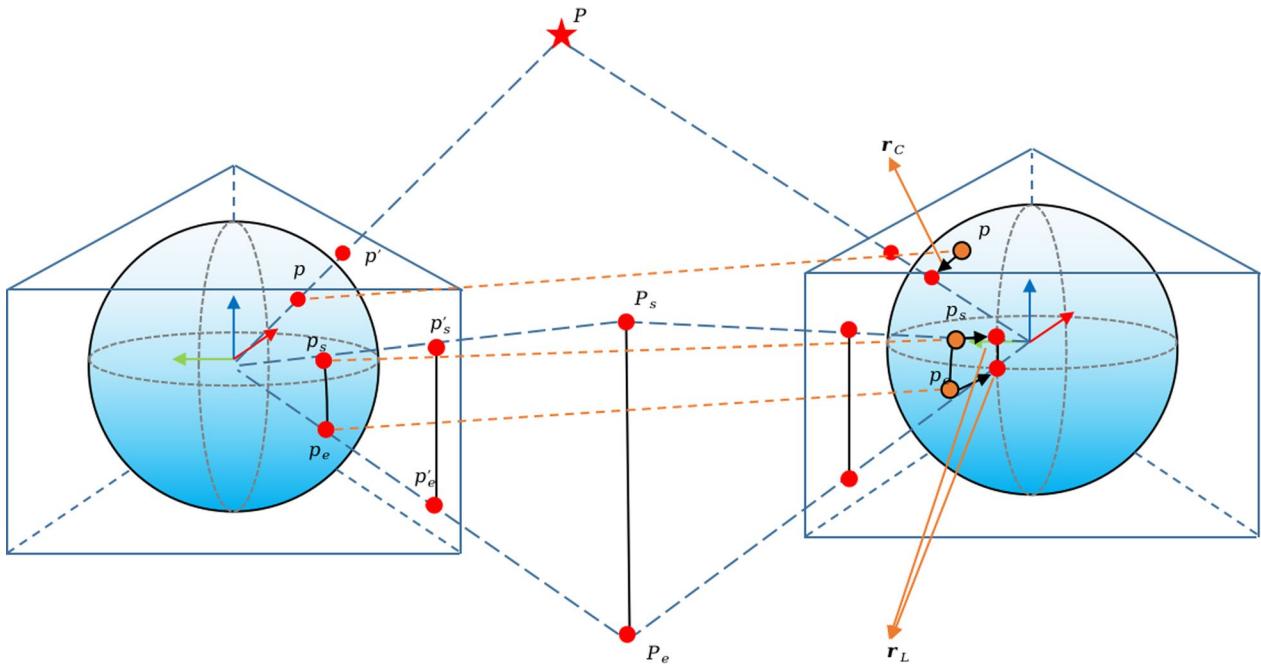


Fig. 7 Illustration of point and line feature residuals in spherical and prism projections. The red 3D point and the red endpoints of the 3D line are projected onto the unit spherical surface in different image frames. The orange point and endpoints are the reprojections from other frames. \mathbf{P} represents an observed 3D point, while \mathbf{P}_s and \mathbf{P}_e represent the observed line segment endpoints. p , p_s , and p_e denote features projected onto the unit sphere. p' , p'_s , and p'_e are the mappings of these features on the triangular prism projection image

where χ is the optimization variable, \mathbf{r}_p , \mathbf{r}_B and \mathbf{r}_C are the same as in Eq. (27). \mathbf{r}_L represents the residuals for line features, where L denotes the set of line measurements.

Line feature residual: As shown in Fig. 7, line feature detection and tracking are performed on the geometrically consistent triangular prism projection image. However, when constructing the line feature residual, the line features distributed in the 360° space need to be projected onto the unit spherical surface. Since a line is defined by two points, this work project both the start and end points using the projection function from Eq. (28), thereby mapping the entire line from the triangular prism projection image onto the unit sphere.

The re-projection error of the line measurement model is defined as follows:

$$\mathbf{r}_L = \begin{bmatrix} d(p_s, \mathbf{n}^l) \\ d(p_e, \mathbf{n}^l) \end{bmatrix} \quad (33)$$

where

$$\left\{ \begin{array}{l} d(p, \mathbf{n}^l) = \frac{\|p \cdot \mathbf{n}^l\|}{\|\mathbf{n}^l\|} \\ p_s = [\alpha_s, \beta_s, \gamma_s] \\ p_e = [\alpha_e, \beta_e, \gamma_e] \end{array} \right. \quad (34)$$

\mathbf{r}_L represents the residual for the line feature, and d denotes the distance between the endpoints of the observed line segment on the unit spherical surface and its reprojected line segment. The vector \mathbf{n}^l is the normal vector to the plane that contains the reprojected line segment. The endpoints of the observed line segment on the unit spherical surface are denoted by p_s and p_e .

Experiments

This section evaluates the effectiveness of the proposed Geotri-VIO framework through a series of experiments, with its performance compared to several state-of-the-art algorithms. They include the benchmark LF-VIO algorithm and the LF-PGVIO algorithm (Z. Wang, Yang, Shi, Zhang, et al., 2024), which are designed for panoramic cameras, as well as the traditional pinhole camera-based VIO algorithms VINS-Mono and PL-VINS. Two datasets are employed for this purpose: a custom dataset collected using the experimental setup shown in Fig. 8 and the publicly available PALVIO dataset. The localization accuracy of Geotri-VIO is compared with these algorithms to validate its advantages.

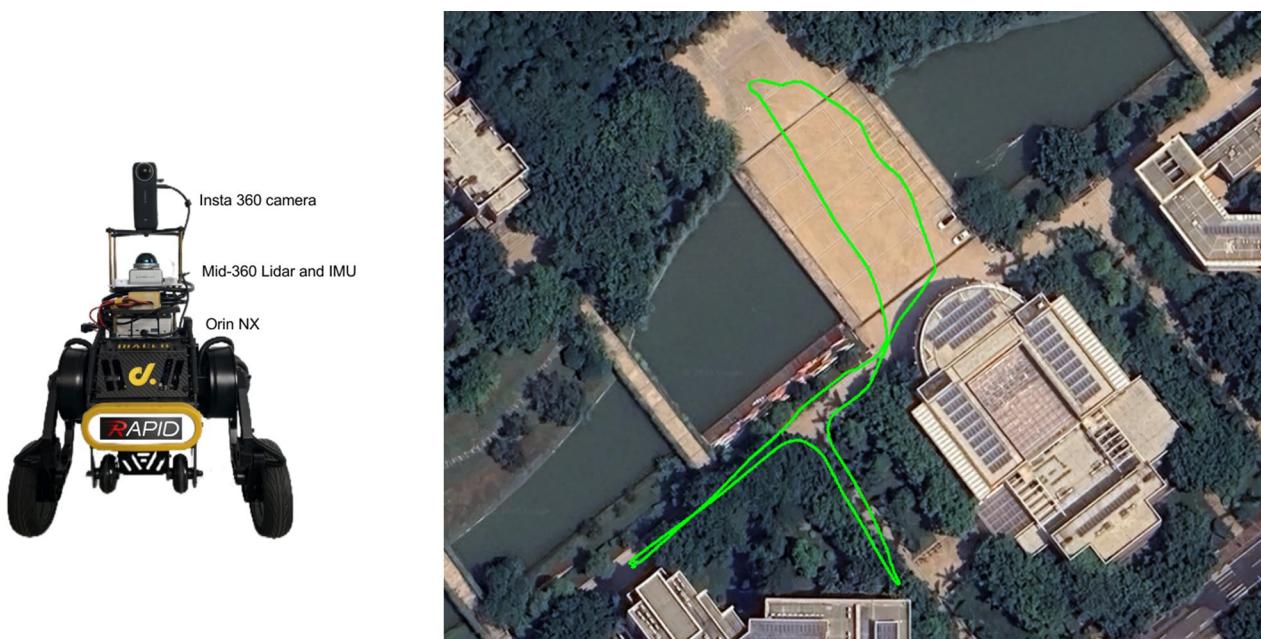


Fig. 8 Hardware setup and outdoor environment for real world data. Our car experiment platform with a Insta 360 panoramic camera, a Livox-Mid-360 LiDAR, an IMU sensor and an onboard computer

Experiments setup and datasets

PALVIO Dataset: The PALVIO dataset is collected using two panoramic annular cameras paired with a CUAV-v5 nano IMU sensor and a RealSense D435 sensor, synchronized with ground-truth location and pose data captured by a motion capture system (Vicon T40s). The panoramic cameras capture monocular images with a resolution of 1280×960 at a rate of 30 Hz and a FoV spanning 360° horizontally and 40° to 120° vertically. The IMU sensor provides angular velocity and acceleration data at 200 Hz, while the motion capture system delivers position and attitude data at 100 Hz, serving as the ground truth. All data are recorded using the Robot Operating System (ROS) and provided in raw format without additional processing. The dataset includes ten indoor sequences (ID01–ID10) and two outdoor sequences (OD01 and OD02). The indoor sequences are collected within an $8\text{ m} \times 10\text{ m}$ indoor environment. Among these, sequences ID01, ID04, ID07, ID08, and ID09 feature complex motion trajectories, including rapid rotations and abrupt changes in direction, while sequences ID02, ID03, ID05, ID06, and ID10 exhibit relatively smooth trajectories. The outdoor sequences are recorded using a small vehicle equipped with a Livox Mid-360 Light Detection and Ranging (LiDAR) sensor, which is used with the Fast-LIO2 algorithm (W. Xu, Cai, He, Lin, and Zhang, 2022) to obtain the ground truth. OD01 covers a large open area, while OD02 includes uneven terrain.

Real world: To further validate the robustness and adaptability of the proposed Geotri-VIO framework in real world scenarios, real world data is collected using a two-wheeled self-balancing robot equipped with an NVIDIA Jetson Orin NX featuring a 6-core Advanced RISC (Reduced Instruction Set Computing) Machine (ARM) Central Processing Unit (CPU) for real-time processing. The robot is outfitted with an Insta360 X4 camera, providing 360-degree imagery at a resolution of 2448×1440 pixels and a frame rate of 30 Frames per Second (FPS), a Livox Mid-360 LiDAR generating point clouds at 10 Hz, and an IMU sensor delivering angular velocity and acceleration data at 200 Hz. The LiDAR data is utilized by the Fast-LIO2 algorithm to establish high-precision ground truth. The real-world data includes two distinct sequences: the first sequence, 360I, is recorded in a large-scale indoor area of $60\text{ m} \times 40\text{ m}$, characterized by structured environments and significant lighting variations, simulating a typical indoor navigation scenario. The second sequence, 360O, is collected in an outdoor campus area of $100\text{ m} \times 120\text{ m}$, featuring unstructured environments and dynamic obstacles such as pedestrians and vehicles, making it a challenging environment for VIO systems. These sequences are designed to comprehensively evaluate the robustness of the proposed method in handling diverse environmental conditions, including structured and unstructured scenes, dynamic obstacles, and varying lighting conditions.

Feature point extraction and tracking evaluation

In the point based VIO system, the quality of feature point tracking directly determines the system's performance. The core of feature point tracking lies in finding matching points between consecutive frames, which serves as the foundation for subsequent pose estimation. To evaluate the tracking performance of the proposed triangular prism projection model on point features, the same image is mapped to both the prism model and the omnidirectional model, using Harris corner (Harris et al., 1988) detection to extract feature points. Optical flow algorithms are then applied to track these feature points. However, tracking with optical flow may be affected by dynamic objects, lighting changes, geometric consistency of the projection model, and other factors, leading to mismatches. Therefore, to enhance the reliability of feature points, the RANSAC algorithm is incorporated to eliminate erroneous matches. Several metrics are used to assess the model's effectiveness in utilizing point features, as follows.

Optical Flow Success Rate (OFSR) measures the accuracy and reliability of feature points tracked using optical flow alone. A higher OFSR indicates that a larger proportion of the feature points tracked through optical flow have been effectively validated. The value of OFSR is defined as:

$$V_{\text{OFSR}} = \frac{N_F}{N_{\text{all}}}, \quad (35)$$

where N_F represents the number of feature points successfully tracked with the optical flow alone, and N_{all} represents the total number of feature points extracted from the previous frame. This metric is used to evaluate the impact of the projection model on the performance of pure optical flow algorithms. By enhancing geometric consistency, the projection model helps the optical flow algorithm better handle challenges such as dynamic objects and lighting variations, thereby improving the robustness of feature tracking.

Tracked Feature Ratio (TFR) reflects the overall tracking capability of feature points. By discarding mismatched feature points, an increase in TFR indicates improved tracking accuracy and stability. The value of TFR is calculated as follows:

$$V_{\text{TFR}} = \frac{N_{\text{FR}}}{N_{\text{all}}}, \quad (36)$$

where N_{FR} represents the number of feature points successfully tracked with optical flow and RANSAC algorithms, and N_{all} represents the total number of feature points extracted from the previous frame. This metric directly evaluates the projection model's impact on the

final tracking precision, which is critical for the performance of VIO systems. A higher TFR indicates that the projection model not only enhances feature point tracking but also improves the overall accuracy and stability of the system, ultimately benefiting VIO localization precision in both static and dynamic environments.

Figure 9 and Table 2 present a comparative analysis of the feature tracking performance between the triangular prism projection model and the omnidirectional projection used in LF-VIO algorithm. In Fig. 9, the feature tracking results with different projection models are clearly illustrated: red points represent features that remain trackable after both the optical flow and RANSAC algorithms, green points indicate features tracked by the optical flow but eliminated by RANSAC, and blue points signify the features, whose optical flow tracking failed. The results show that the triangular prism projection model achieves a higher proportion of red points, demonstrating its improved feature tracking stability. At the same time, the reduced number of green and blue points indicates that the model better maintains geometric consistency in large FoV conditions, thereby minimizing feature tracking failures caused by geometric distortion. This improvement highlights the advantages of the triangular prism projection model in supporting feature detection and matching in panoramic VIO systems.

Table 2 presents a comparison of the TFR and OFSR metrics for all sequences between the triangular prism projection model and the omnidirectional projection used in LF-VIO. Bold values indicate the best results in each comparison, and this convention applies to all subsequent tables. The results demonstrate that the triangular prism projection model consistently outperforms the omnidirectional projection in all indoor and outdoor test sequences. The average TFR for the triangular prism projection model increased from 0.883 (LF-VIO) to 0.929, indicating a significant improvement in feature tracking stability due to its enhanced geometric consistency. In contrast, OFSR, which measures feature tracking accuracy based solely on the optical flow, shows minimal differences between the two models (average values of 0.988 and 0.984, respectively). This is because the optical flow algorithm inherently tolerates a certain level of tracking error, and the accuracy of tracked features can vary. The RANSAC algorithm subsequently eliminates the features with lower tracking precision. Therefore, in terms of both TFR and OFSR metrics, the triangular prism projection model stands out by significantly improving the quality and utilization of feature tracking through enhanced geometric consistency.

In the real-world data, Geotri-VIO further demonstrates its robustness. The 360I sequence (indoor) features structured environments and significant lighting

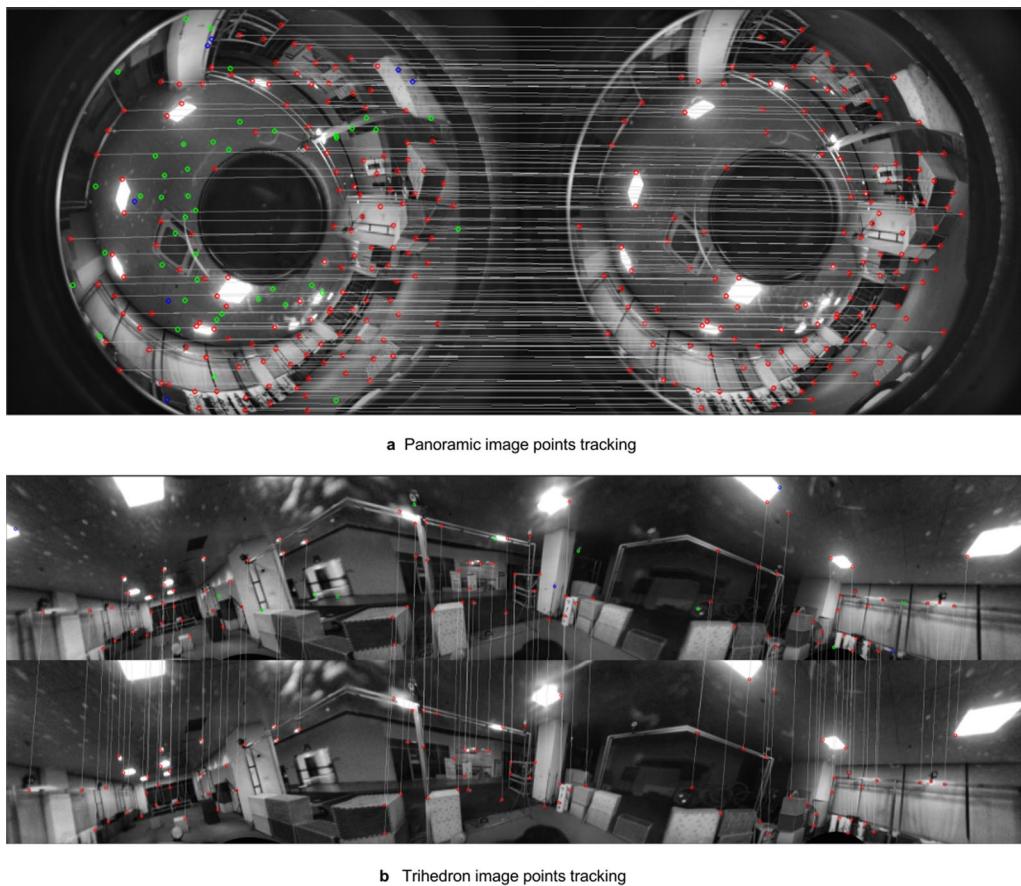


Fig. 9 Feature point extraction and tracking results under different projection models. This figure illustrates feature point extraction and tracking results under **a** anamorphic annular image and **b** triangular prism projection image. Red points represent those that remain trackable after optical flow and RANSAC algorithms, green points indicate those that are tracked by the optical flow method but eliminated by RANSAC, and blue points denote those where optical flow tracking failed

variations, while the 360O sequence (outdoor) includes unstructured environments and dynamic obstacles such as pedestrians and vehicles. In these conditions, Geotri-VIO achieves TFR values of 0.955 and 0.947, and OFSR values of 0.995 and 0.992, significantly outperforming the comparative method LF-VIO (TFR: 0.843 and 0.828; OFSR: 0.987 and 0.981). These results indicate that Geotri-VIO effectively addresses the challenges posed by structured and unstructured scenes, dynamic obstacles, and varying lighting conditions.

To validate the superiority of the triangular prism projection compared to other multi-prism projections, the TFR and OFSR metrics of the multi-prism projection model are analyzed for different numbers of projection planes, as shown in Table 3. The triangular prism projection with three planes achieves the highest scores for both metrics. As the number of projection planes increases, both TFR and OFSR show a declining trend. In the ID01 sequence, the TFR decreases from 0.854 for the three-plane projection to 0.783 for the equidistant

cylindrical projection, while the OFSR drops from 0.975 to 0.964. This result is consistent with the theoretical findings derived from Eq. (25), which demonstrates that as the number of projection planes increases, geometric consistency deteriorates, leading to reduced tracking stability.

Feature point based VIO evaluation

To evaluate the performance of Geotri-VIO using point features, three metrics are used to compare Geotri-VIO with LF-VIO: Relative Pose Error in translation (RPEt), Relative Pose Error in rotation (RPER), and Absolute Trajectory Error (ATE). RPEt measures the error between the estimated translation and the true values between consecutive poses, reflecting the accuracy of local displacement estimation. RPER quantifies the rotational error between consecutive poses, assessing the system's accuracy in estimating changes in orientation. ATE evaluates global consistency by comparing the estimated trajectory with the true trajectory, serving as a key

Table 2 TFR and OFSR Comparison Between the proposed and LF-VIO

Items	Different results of TFR		Different results of OFSR	
	Ours	LF-VIO	Ours	LF-VIO
ID01	0.854	0.766	0.975	0.963
ID02	0.961	0.921	0.994	0.992
ID03	0.968	0.942	0.995	0.993
ID04	0.942	0.913	0.991	0.988
ID05	0.931	0.889	0.987	0.985
ID06	0.920	0.884	0.985	0.983
ID07	0.886	0.848	0.979	0.974
ID08	0.926	0.892	0.987	0.984
ID09	0.924	0.889	0.987	0.983
ID10	0.809	0.771	0.966	0.951
OD01	0.990	0.987	0.998	0.998
OD02	0.989	0.987	0.998	0.996
360I	0.955	0.843	0.995	0.987
360O	0.947	0.828	0.992	0.981
Mean	0.929	0.883	0.988	0.984

performance metric for mapping and long-term navigation tasks. These metrics assess the accuracy of the VIO system from different perspectives and are crucial in practical applications. Additionally, the accuracy of the VIO system is assessed on the ID01 and OD01 sequences for different projection planes.

Table 4 presents the experimental comparison of RPEt, RPER, and ATE between Geotri-VIO and LF-VIO for all test sequences, with the corresponding trajectories shown in Fig. 10. The trajectories of Geotri-VIO (green) are overall closer to the Ground Truth (red) compared to LF-VIO (blue), particularly in complex trajectory regions such as ID01 and ID04, where Geotri-VIO effectively reduces drift errors. In most test sequences, Geotri-VIO outperforms LF-VIO in terms of RPEt, RPER, and ATE. On average, Geotri-VIO achieves a 25% reduction in RPEt, a 10% reduction in RPER, and a 39% reduction in ATE, demonstrating its significant improvement in global

trajectory reconstruction accuracy. These enhancements are attributed to the introduction of the triangular prism projection model, which substantially improves the performance of point feature-based VIO systems.

To further validate the performance of Geotri-VIO in more challenging environments, we analyze the results of the 360I and 360O sequences. As shown in Table 4, Geotri-VIO achieves an RPEt reduction to 6.800, an RPER reduction to 0.295, and an ATE reduction to 0.317 in the 360I sequence, while in the 360O sequence, it achieves an RPEt reduction to 12.001, an RPER reduction to 0.420, and an ATE reduction to 1.126, significantly outperforming LF-VIO. These results further confirm the robustness of Geotri-VIO in handling complex scenarios, as previously demonstrated by its superior performance in TFR and OFSR metrics. The consistent improvements in multiple evaluation metrics highlight the effectiveness of the proposed triangular prism projection model in dynamic and challenging environments.

Additionally, to validate that the triangular prism projection model outperforms other multi-prism projection models in point based panoramic VIO performance, a comparative analysis is conducted using the indoor ID01 sequence and outdoor OD01 sequence for different projection planes. The results are shown in Table 3, with corresponding trajectories presented in Fig. 11. As the number of projection planes increases, the global geometric consistency of the projection model gradually deteriorates, leading to an upward trend in RPEt, RPER, and ATE. These findings confirm that the triangular prism projection model achieves superior performance by preserving geometric consistency and maintaining higher accuracy in trajectory estimation compared to the models with more projection planes.

Finally, to validate the advantages of the proposed Geotri-VIO framework over traditional VIO algorithms designed for pinhole cameras, a comparison is conducted with VINS-Mono, a widely-used and classical VIO algorithm. VINS-Mono is selected for comparison due to its

Table 3 Comparison of Feature Point Based Performance Metrics for Different Surfaces

Surfaces	Different metrics in ID01 sequence					Different metrics in ODO1 sequence				
	TFR	OFSR	RPEt	RPER	ATE	TFR	OFSR	RPEt	RPER	ATE
3	0.854	0.975	1.091	0.994	0.244	0.990	0.998	21.534	0.473	0.205
4	0.852	0.969	1.174	1.022	0.252	0.989	0.998	23.571	0.487	0.228
5	0.851	0.969	1.273	1.112	0.278	0.989	0.998	25.913	0.505	0.272
6	0.835	0.966	1.278	1.149	0.280	0.988	0.998	26.639	0.519	0.308
Infinity	0.783	0.964	1.438	1.238	0.342	0.976	0.998	28.984	0.548	0.341

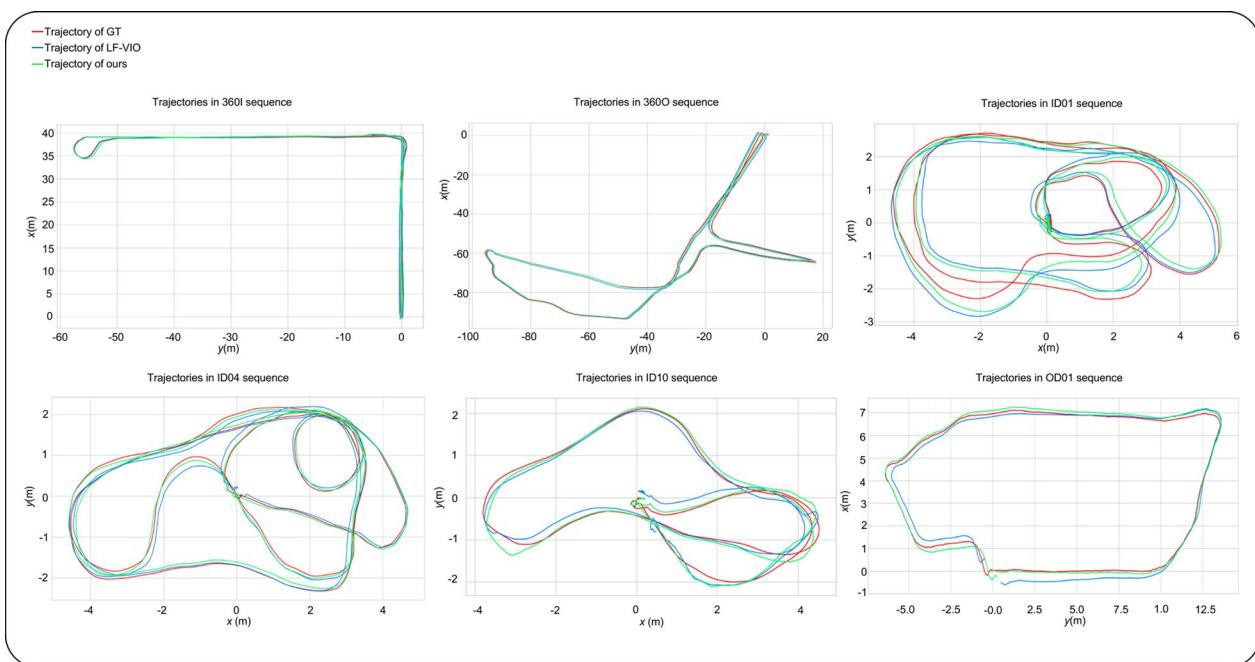


Fig. 10 A portion of trajectories of different Point based VIO systems. This figure illustrates a portion of trajectories of different point based VIO systems on the PALVIO dataset benchmark include sequences: 360l, 360O, ID01, ID04, ID10, OD01

Table 4 Comparison of Performance for RPEt RPEr ATE in Point Feature Based VIO with LF-VIO

Items	Different results of RPEt(%)		Different results of RPEr((°/m)		Different results of ATE (m)	
	Ours	LF-VIO	Ours	LF-VIO	Ours	LF-VIO
ID01	1.091	1.416	0.994	1.139	0.244	0.288
ID02	0.750	0.813	0.394	0.535	0.241	0.409
ID03	0.600	0.626	0.402	0.451	0.195	0.355
ID04	0.694	0.810	0.307	0.332	0.100	0.154
ID05	1.034	0.952	0.324	0.374	0.189	0.274
ID06	0.873	0.910	0.384	0.391	0.063	0.111
ID07	0.926	1.144	0.389	0.398	0.112	0.379
ID08	0.880	1.014	0.629	0.601	0.110	0.186
ID09	0.911	0.873	0.430	0.426	0.158	0.203
ID10	2.075	1.567	0.980	0.972	0.179	0.328
OD01	21.534	29.188	0.473	0.542	0.205	0.344
OD02	14.091	16.386	0.276	0.305	0.122	0.143
360l	6.800	11.242	0.295	0.366	0.317	0.680
360O	12.001	19.214	0.420	0.627	1.126	1.641
Mean	4.590(↓25%)	6.154	0.478(↓10%)	0.533	0.240(↓39%)	0.393

use of optical flow tracking in the front-end, which aligns with the technical approach of the proposed method. The comparison is performed on the PALVIO dataset, and the results are summarized in Table 5. As shown in the table, Geotri-VIO consistently outperforms VINS-Mono for all test sequences. For instance, in the ID01 sequence, Geotri-VIO achieves an RPEt of 1.091, RPEr of 0.994 degree/m, and ATE of 0.244 m, significantly lower than VINS-Mono's 2.446, 1.431 degree/m, and 0.920 m, respectively. Similarly, in the OD01 sequence, Geotri-VIO achieves an RPEt of 21.534, RPEr of 0.473 degree/m, and ATE of 0.205 m, compared to VINS-Mono's 22.181,

sequence, Geotri-VIO achieves an RPEt of 1.091, RPEr of 0.994 degree/m, and ATE of 0.244 m, significantly lower than VINS-Mono's 2.446, 1.431 degree/m, and 0.920 m, respectively. Similarly, in the OD01 sequence, Geotri-VIO achieves an RPEt of 21.534, RPEr of 0.473 degree/m, and ATE of 0.205 m, compared to VINS-Mono's 22.181,

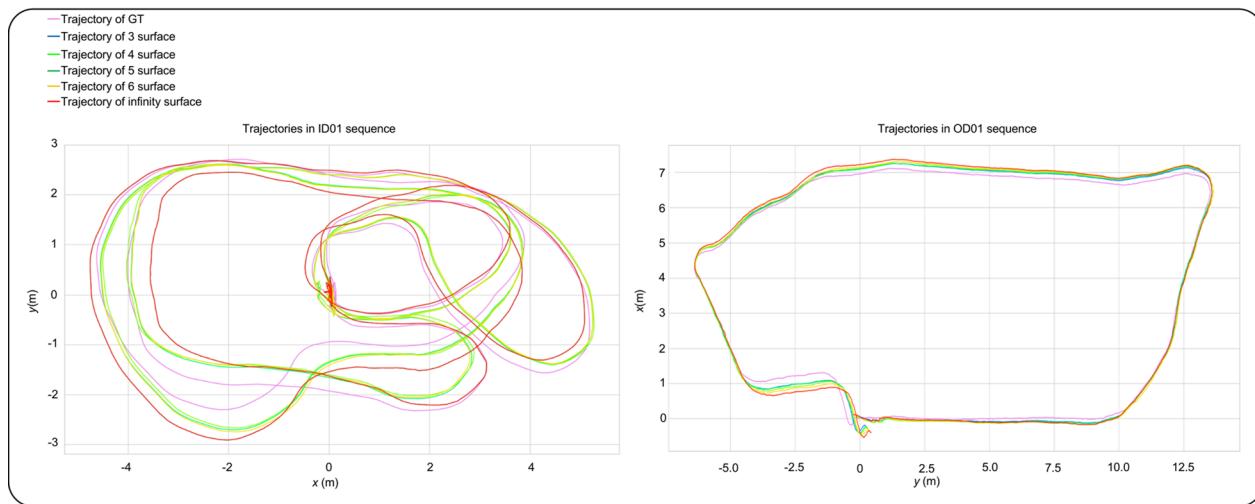


Fig. 11 Point based VIO trajectories with varying numbers of projection planes. This figure illustrates point based VIO trajectories with varying numbers of projection planes on sequences: ID01, OD01

Table 5 Comparison of Performance for Point Feature Based VIO with VINS-mono in PALVIO Dataset

Items	Different results of RPEt(%)		Different results of RPEr($^{\circ}$)/m)		Different results of ATE (m)	
	Ours	VINS	Ours	VINS	Ours	VINS
ID01	1.091	2.446	0.994	1.431	0.244	0.920
ID02	0.750	1.542	0.394	0.614	0.241	0.414
ID03	0.600	1.987	0.402	0.725	0.195	0.345
ID04	0.694	1.719	0.307	0.649	0.100	0.247
ID05	1.034	1.957	0.324	0.578	0.189	0.453
ID06	0.873	2.193	0.384	0.645	0.063	0.204
ID07	0.926	2.045	0.389	0.539	0.112	0.386
ID08	0.880	1.905	0.629	0.752	0.110	0.213
ID09	0.911	2.133	0.430	0.557	0.158	0.391
ID10	2.075	3.774	0.980	1.567	0.179	0.539
OD01	21.534	22.181	0.473	0.539	0.205	1.309
OD02	14.091	failed	0.276	failed	0.122	failed
Mean	3.705 (↓5%)	3.898	0.499 (↓36%)	0.781	0.160 (↓69%)	0.529

0.539 degree/m, and 1.309 m. Notably, VINS-Mono fails to complete the OD02 sequence due to its limited FoV, which leads to insufficient feature tracking in large-scale environments. In contrast, Geotri-VIO successfully achieves an RPEt of 14.091, RPEr of 0.276 degree/m, and ATE of 0.122 m in the same sequence. On average, Geotri-VIO reduces RPEt by 5%, RPEr by 36%, and ATE by 69% compared to VINS-Mono. These results demonstrate the superior performance of the proposed Geotri-VIO framework in handling complex environments, leveraging the significant advantages of wide FoV

cameras and the effectiveness of the triangular prism projection model.

Evaluation of feature line extraction and tracking

In point and line-based VIO systems, a higher success rate in line tracking indicates that more effective line features are utilized for pose estimation, enabling more accurate position estimation and ensuring stability in varying lighting conditions and dynamic environments. To evaluate the performance of the proposed multi-prism projection model for line features, the same images are mapped onto the multi-prism model

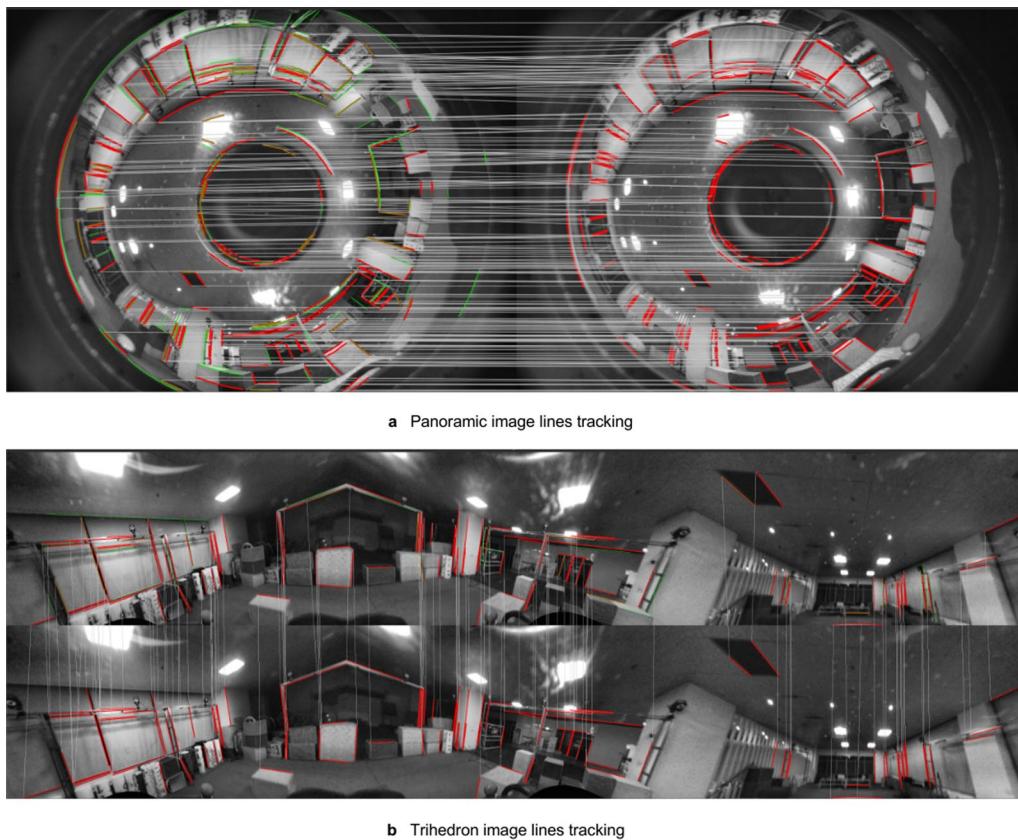


Fig. 12 Feature line extraction and tracking results under different projection models. Red lines represent those that remain trackable, green lines indicate those tracking failed

Table 6 Comparison of Performance for TLR

Items	Ours results of TLR	LF-PGVIO results of TLR
ID01	0.503	0.412
ID02	0.645	0.616
ID03	0.661	0.646
ID04	0.608	0.583
ID05	0.596	0.558
ID06	0.575	0.531
ID07	0.539	0.522
ID08	0.600	0.576
ID09	0.614	0.581
ID10	0.483	0.452
OD01	0.565	0.528
OD02	0.592	0.542
360I	0.457	0.403
360O	0.439	0.358
Mean	0.563	0.522

and the omnidirectional model. The ELSED algorithm is used to extract line features, followed by the application of LBD descriptors for feature description and matching. The following metric is utilized to assess the model's effectiveness in leveraging line features:

Tracked Line Ratio(TLR) reflects the tracking capability of line segments with the current projection model. An increase in TLR indicates improved tracking accuracy of line features. The value of TLR is calculated as:

$$V_{\text{TLR}} = \frac{N_{\text{LT}}}{N_{\text{LA}}} \quad (37)$$

where N_{LT} represents the number of line segments successfully tracked across frames, and N_{LA} denotes the total number of line segments extracted in the previous frame. A higher TLR allows the VIO system to effectively utilize more line features, thereby enhancing positioning accuracy and improving stability in complex environments.

Table 6 and Fig. 12 present a comparative analysis of the line feature tracking performance between the triangular prism projection model and the omnidirectional projection used in LF-PGVIO. In Fig. 12, the results of

line feature extraction and tracking with different projection models are clearly illustrated: red lines represent successfully tracked features, while green lines indicate the features that failed in tracking. The results show that the triangular prism projection model achieves a higher proportion of successfully tracked red lines, demonstrating its superior line feature tracking stability. Meanwhile, the reduced number of green lines indicates that the model maintains better geometric consistency under large FoV, minimizing tracking failures caused by distortions. These improvements underscore the triangular prism projection model's effectiveness in line feature detection and tracking.

Table 6 compares the TLR metrics between the triangular prism projection model and the omnidirectional projection used in LF-PGVIO for all test sequences. The results demonstrate that the triangular prism projection model consistently achieves higher TLR values, reflecting its superior ability to maintain and track line features effectively. For example, in the challenging indoor sequence ID01, the triangular prism projection model achieves a TLR of 0.503, significantly higher than LF-PGVIO's 0.412. Similarly, in ID03 and ID09, the TLR values of 0.661 and 0.614 surpass LF-PGVIO's 0.646 and 0.581, respectively. These improvements highlight the triangular prism projection model's effectiveness in preserving geometric consistency and minimizing tracking failures in complex environments. The superiority of the triangular prism projection model is also evident in outdoor sequences. In OD01, it achieves a TLR of 0.565, compared to LF-PGVIO's 0.528, while in OD02 it reaches 0.592, outperforming LF-PGVIO's 0.542.

Furthermore, in the real-world data, which were collected in highly dynamic environments and significant lighting variations, the triangular prism projection model continues to demonstrate its robustness. In the 360I sequence, it achieves a TLR of 0.457, compared to LF-PGVIO's 0.403, while in the 360O sequence, it attains a TLR of 0.439, significantly higher than LF-PGVIO's 0.358. These results further validate the robustness of

the proposed model in diverse and challenging environments. The consistent improvements for all sequences underscore the advantages of the triangular prism projection model in handling complex scenarios.

Overall, the average TLR for all sequences with the triangular prism projection model is 0.563, representing an improvement over LF-PGVIO's 0.522. While LF-PGVIO employs the OCSD method for extracting line features, its RLBD descriptor remains fundamentally reliant on the LBD descriptor, which requires strict geometric consistency. By preserving geometric consistency and reducing distortions, the triangular prism projection model ensures more reliable and accurate line feature tracking.

To validate that the triangular prism projection model outperforms other multi-prism projection models in line feature tracking, the TLR metric is analyzed for different numbers of projection planes, as shown in Table 7. The results indicate that the triangular prism projection model with three planes achieves the highest TLR in both indoor (ID01) and outdoor (OD01) sequences, with TLR values of 0.503 and 0.565, respectively. As the number of projection planes increases, the TLR steadily decreases. For instance, in the ID01 sequence, the TLR drops from 0.503 for the three-plane configuration to 0.462 for the six-plane configuration. Similarly, in the OD01 sequence, the TLR declines from 0.565 to 0.529 over the same range. When the number of projection planes increases to the equidistant cylindrical projection (infinity planes), the line feature extraction and tracking algorithm fails entirely, underscoring the importance of maintaining geometric consistency for effective line feature tracking.

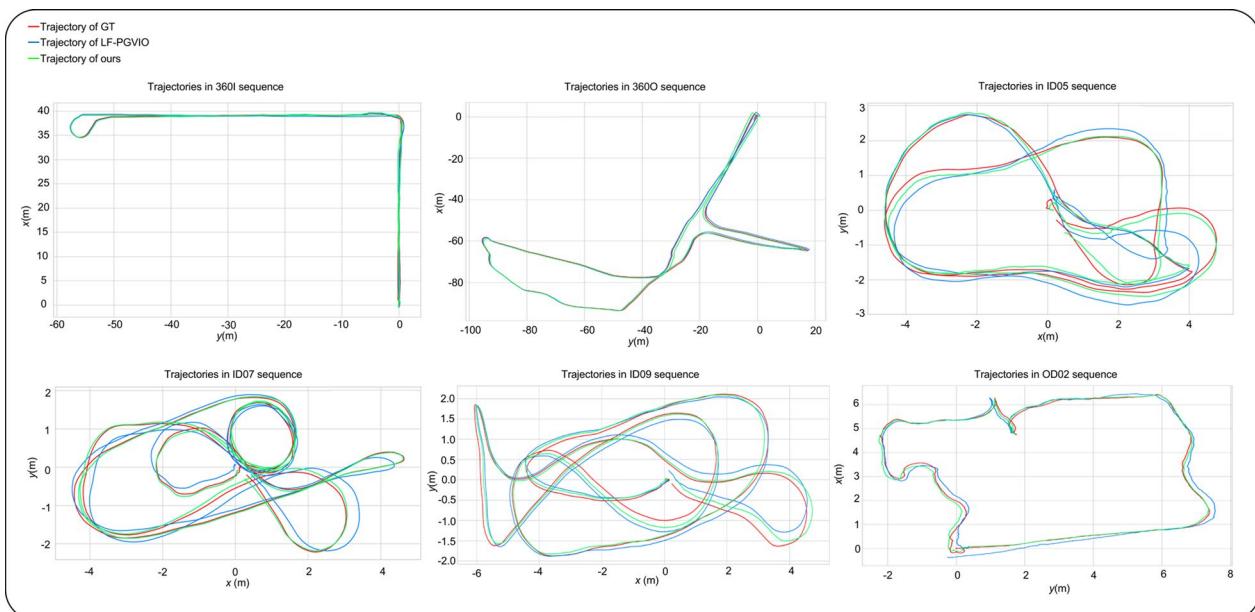
Feature point and line based VIO evaluation

Using the metrics RPEt, RPEr, and ATE, the performance of the point and line-based Geotri-VIO system is evaluated, as summarized in Table 8, with corresponding trajectories shown in Fig. 13. The experimental results reveal that Geotri-VIO consistently outperforms LF-PGVIO for all test sequences. On average, Geotri-VIO reduces RPEt by 27%, RPEr by 11%, and ATE by 41% compared to

Table 7 Performance Comparison of Feature Point and Line-Based Metrics Across Different Projection Models

Table 8 Comparison of RPEt, RPEr, and ATE for the Proposed Framework and LF-PGVIO

Items	Different results of RPEt(%)		Different results of RPEr((°/m)		Different results of ATE (m)	
	Ours	LF-PGVIO	Ours	LF-PGVIO	Ours	LF-PGVIO
ID01	0.958	1.371	0.923	1.080	0.219	0.278
ID02	0.738	0.796	0.374	0.500	0.183	0.306
ID03	0.582	0.609	0.363	0.423	0.091	0.192
ID04	0.692	0.676	0.273	0.268	0.128	0.149
ID05	0.955	0.920	0.302	0.342	0.197	0.444
ID06	0.814	0.903	0.343	0.359	0.051	0.105
ID07	0.918	1.244	0.371	0.385	0.093	0.226
ID08	0.716	1.102	0.627	0.616	0.088	0.184
ID09	0.887	0.819	0.426	0.410	0.144	0.186
ID10	1.985	1.966	0.920	0.941	0.150	0.299
OD01	20.067	28.382	0.466	0.563	0.183	0.335
OD02	11.928	15.694	0.251	0.286	0.096	0.124
360I	5.110	9.934	0.273	0.360	0.263	0.606
360O	11.367	14.071	0.385	0.554	1.064	1.568
Mean	4.113(↓27%)	5.606	0.450(↓11%)	0.506	0.211(↓41%)	0.357

**Fig. 13** A portion of trajectories of different point and line based VIO systems. This figure illustrates a portion of trajectories of different point and line based VIO systems on the PALVIO dataset benchmark and our dataset include sequences: 360I, 360O, ID05, ID07, ID09, OD02

LF-PGVIO, demonstrating significant improvements in trajectory estimation accuracy.

For instance, in the challenging ID01 sequence, Geotri-VIO achieves an RPEt of 0.958, RPEr of 0.923 degree/m, and ATE of 0.219 m, all of which are superior to LF-PGVIO's corresponding values of 1.371, 1.080 degree/m, and 0.278 m. Similarly, in the OD01 sequence,

Geotri-VIO achieves an RPEt of 20.067, RPEr of 0.466 degree/m, and ATE of 0.183 m, significantly outperforming LF-PGVIO. These results highlight the robustness of the Geotri-VIO system in both indoor and outdoor scenarios, effectively reducing trajectory drift and improving accuracy.

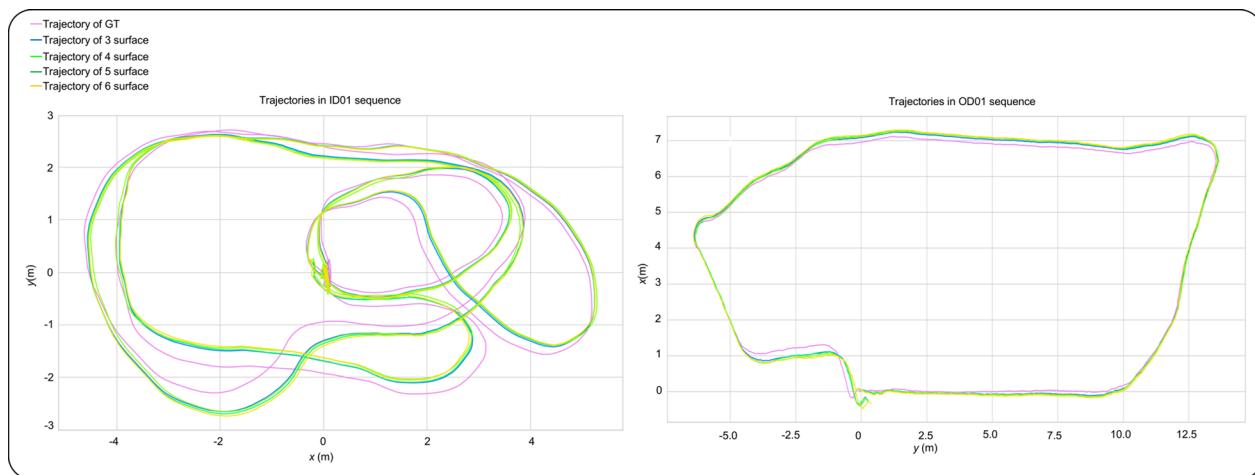


Fig. 14 Point and line based VIO trajectories with varying numbers of projection planes. This figure illustrates point and line based VIO trajectories with varying numbers of projection planes on sequences: ID01, OD01

To further validate the performance of Geotri-VIO in highly dynamic and complex environments, the 360I and 360O sequences are analyzed. In the 360I sequence, Geotri-VIO achieves an RPEt of 5.110, RPER of 0.273 degree/m, and ATE of 0.263 m, compared to LF-PGVIO's 9.934, 0.360 degree/m, and 0.606 m, respectively. Similarly, in the 360O sequence, Geotri-VIO attains an RPEt of 11.367, RPER of 0.385 degree/m, and ATE of 1.064 m, significantly outperforming LF-PGVIO's 14.071, 0.554 degree/m, and 1.568 m. These results demonstrate the robustness of Geotri-VIO in handling dynamic obstacles and varying lighting conditions. The consistent improvements in all metrics underscore the effectiveness of the proposed framework in real world applications.

Additionally, the performance of the VIO system for different numbers of projection planes is evaluated using the ID01 and OD01 sequences. The experimental results are presented in Table 7, with corresponding trajectories shown in Fig. 14. The results indicate that the triangular prism projection model with three planes achieves the best overall performance, with the lowest RPEt, RPER, and ATE values in both sequences. For example, in the ID01 sequence, the RPEt increases from 0.958% (three planes) to 1.201% (six planes), while the ATE increases from 0.219 m to 0.260 m. A similar trend is observed in the OD01 sequence, where the RPEt and ATE also degrade as the number of projection planes increases. When the projection model reaches the equidistant

Table 9 Comparison of RPEt, RPER, and ATE for the Proposed Framework and PL-VINS

Items	Different results of RPEt(%)		Different results of RPER(°/m)		Different results of ATE (m)	
	Ours	PL-VINS	Ours	PL-VINS	Ours	PL-VINS
ID01	0.958	2.102	0.923	1.960	0.219	0.532
ID02	0.738	1.292	0.374	0.903	0.183	0.454
ID03	0.582	1.157	0.363	0.826	0.091	0.300
ID04	0.692	0.972	0.273	0.439	0.128	0.297
ID05	0.955	1.199	0.302	0.421	0.197	0.518
ID06	0.814	1.226	0.343	0.459	0.051	0.195
ID07	0.918	1.138	0.371	0.417	0.093	0.256
ID08	0.716	1.229	0.627	0.809	0.088	0.236
ID09	0.887	1.051	0.426	0.502	0.144	0.240
ID10	1.985	2.293	0.920	1.341	0.150	0.408
OD01	20.067	22.221	0.466	0.590	0.183	0.944
OD02	11.928	31.150	0.251	0.697	0.096	2.524
Mean	3.353(↓38%)	5.503	0.470(↓39%)	0.780	0.135(↓76%)	0.575

cylindrical projection (infinity planes), the system fails entirely, underscoring the importance of geometric consistency in ensuring the accuracy and stability of VIO systems.

Finally, to further validate the performance of the proposed Geotri-VIO framework in point and line feature based VIO, a comparison is conducted with PL-VINS, a state-of-the-art VIO algorithm designed for pinhole cameras that shares a similar technical approach in front-end feature tracking (optical flow for point features and line feature extraction and matching). The comparison is performed on the PALVIO dataset, and the results are summarized in Table 9. As shown in the table, Geotri-VIO consistently outperforms PL-VINS for all test sequences. For instance, in the ID01 sequence, Geotri-VIO achieves an RPEt of 0.958, RPER of 0.923 degree/m, and ATE of 0.219 m, significantly lower than PL-VINS's 2.102, 1.960 degree/m, and 0.532 m, respectively. Similarly, in the OD01 sequence, Geotri-VIO achieves an RPEt of 20.067, RPER of 0.466 degree/m, and ATE of 0.183 m, compared to PL-VINS's 22.221, 0.590 degree/m, and 0.944 m. Notably, in the OD02 sequence, PL-VINS exhibits significantly larger errors, with an RPEt of 31.150, RPER of 0.697 degree/m, and ATE of 2.524 m, while Geotri-VIO achieves an RPEt of 11.928, RPER of 0.251 degree/m, and ATE of 0.096 m. On average, Geotri-VIO reduces RPEt by 38%, RPER by 39%, and ATE by 76% compared to PL-VINS. These results demonstrate the superior performance of the proposed Geotri-VIO framework in handling complex environments, leveraging the advantages of wide FoV cameras and the effectiveness of the triangular prism projection model for both point and line features.

These findings validate the effectiveness of the triangular prism projection model in preserving geometric consistency and enhancing feature tracking performance. By integrating both point and line features, Geotri-VIO achieves superior trajectory estimation accuracy compared to benchmarks, particularly in large FoV scenarios and complex environments.

Ablation Study on Geotri-VIO

To evaluate the impact of different projection models and feature types on the performance of Geotri-VIO, we conducted an ablation study. The results are presented in Table 10, which compares the Mean Relative Pose Error in translation (MRPEt), rotation (MRPER), and Mean Absolute Trajectory Error (MATE) for four configurations: Tri-PL, Tri-P, Omni-PL, and Omni-P. Here, Tri refers to the triangular prism projection model proposed in this paper, while Omni denotes the omnidirectional projection model adopted by state-of-the-art methods (Z. Wang et al., 2022; Z. Wang, Yang, Shi, Li, et al., 2024; Z. Wang, Yang, Shi, Zhang, et al., 2024). PL and P represent the use of point-line features and point-only features, respectively. These metrics were computed by averaging the results over the PALVIO Dataset (ID01-ID10 and OD01-OD02) as well as the Custom Dataset (360I and 360O), ensuring a comprehensive evaluation in diverse scenarios.

In addition to accuracy metrics, we also examined computational efficiency by evaluating CPU utilization and memory consumption for each configuration. To ensure a fair comparison, all methods were executed at a fixed processing frequency of 10 Hz, which is one of the most commonly used frame rates in VIO applications. This setup allows us to assess the computational overhead associated with different projection models and feature types in realistic conditions. The results reveal that while the triangular prism projection model consistently achieves higher accuracy, it also demonstrates a balanced trade-off between accuracy and computational cost, particularly in terms of memory efficiency. These findings provide valuable insights into the practical implications of projection model selection in real world VIO deployments.

Impact of Projection Models: The triangular prism projection model (Tri) consistently outperforms the omnidirectional projection model (Omni) for all feature types. For instance, in the point-line feature setting, Tri-PL reduces MRPEt by 27% (4.113 vs. 5.606), MRPER by 11% (0.450 vs. 0.506), and MATE by 41% (0.211 vs. 0.357) compared to Omni-PL. In the point-only feature setting, Tri-P achieves reductions of 25% in MRPEt (4.590

Table 10 Performance Metrics of Different Projection Models and Feature Types

Settings	MRPEt (%)	MRPER ((°)/m)	MATE (m)	CPU (%)	MEM (GB)	Freq (Hz)
Tri-PL	4.113	0.450	0.211	38.254	0.064	10
Tri-P	4.590	0.478	0.240	25.318	0.048	10
Omni-PL	5.606	0.506	0.357	44.577	0.304	10
Omni-P	6.154	0.533	0.393	25.125	0.048	10

vs. 6.154), 10% in MRPEr (0.478 vs. 0.533), and 39% in MATE (0.240 vs. 0.393) relative to Omni-P. These results demonstrate that the triangular prism projection model is better suited for handling feature data in VIO systems, aligning closely with the theoretical analysis derived from the mathematical proofs in Section Methodology.

In addition to the accuracy, the triangular prism projection model also exhibits advantages in computational efficiency. Compared to Omni-PL, Tri-PL reduces memory usage by 79% (0.064 GB vs. 0.304 GB) while achieving a lower CPU load (38.254% vs. 44.577%). This improvement stems from the local geometric consistency of the triangular prism projection, which enables a more compact representation and efficient tracking of line features. In contrast, the omnidirectional projection model lacks geometric consistency, often requiring more complex methods for describing and tracking line features, leading to higher memory consumption and CPU load. A similar trend is observed in the point-only setting: Tri-P and Omni-P exhibit nearly identical memory consumption (0.048 GB), while Tri-P needs a slightly higher CPU load (25.318% vs. 25.125%). This minor difference arises because, apart from the front-end projection model, both configurations share identical VIO settings. Moreover, since the triangular prism projection is derived from the omnidirectional image through a simple transformation, the additional computational cost is negligible, explaining the nearly identical resource usage.

These results highlight that the triangular prism projection model not only enhances state estimation accuracy but also achieves a more favorable balance between computational cost and performance, making it a compelling choice for VIO systems.

Impact of Feature Types: The use of point-line features (PL) consistently improves performance compared to point-only features (P) with both projection models. For example, in the triangular prism projection setting, Tri-PL reduces MRPEt by 10% (4.113 vs. 4.590), MRPEr by 5% (0.450 vs. 0.478), and MATE by 12% (0.211 vs. 0.240) compared to Tri-P. Similarly, in the omnidirectional projection setting, Omni-PL achieves reductions of 9% in MRPEt (5.606 vs. 6.154), 5% in MRPEr (0.506 vs. 0.533), and 9% in MATE (0.357 vs. 0.393) relative to Omni-P.

This indicates that incorporating line features enhances the system's ability to estimate both translation and rotation, which is consistent with the conclusions of Wang et al. (2024). However, this accuracy improvement comes at the cost of increased resource consumption. Compared to point-only features, incorporating point-line features needs higher CPU load with both projection models: Tri-PL incurs a 51% increase over Tri-P (38.254% vs. 25.318%), while Omni-PL sees a 77% increase over

Omni-P (44.577% vs. 25.125%). Memory usage also rises, with Tri-PL consuming 33% more than Tri-P (0.064 GB vs. 0.048 GB) and Omni-PL using 533% more than Omni-P (0.304 GB vs. 0.048 GB).

These results demonstrate that while line features significantly enhance accuracy, they introduce a notable computational overhead, particularly in the omnidirectional setting. The triangular prism projection model, however, mitigates this cost by enabling more efficient line feature description and tracking algorithms, achieving a better trade-off between accuracy and computational efficiency.

Feature Utilization Across Projection Models: In the *Impact of Feature Types* section, it is observed that the reductions in MRPEt, MRPEr, and MATE achieved by Tri-PL over Tri-P (10%, 5%, 12%) exceed those achieved by Omni-PL over Omni-P (9%, 5%, 9%). This suggests that the triangular prism projection model benefits more from incorporating line features compared to the omnidirectional projection model, likely due to its compatibility with more effective line feature description and tracking methods. This further supports its suitability for handling feature data in VIO systems.

Discussion: The ablation study demonstrates that both the choice of projection model and the inclusion of line features significantly impact VIO performance in terms of accuracy and computational efficiency. The triangular prism projection model not only improves state estimation accuracy but also achieves a better balance between memory usage and CPU load, making it a computationally efficient alternative to the omnidirectional projection model. Additionally, the use of point-line features consistently enhances localization accuracy using both projection models, with the triangular prism projection benefiting more from the inclusion of line features. This suggests that its structured representation is more compatible with efficient line feature tracking. Overall, the results highlight the advantage of adopting a projection model that not only preserves geometric consistency but also facilitates effective feature utilization, leading to a more robust and efficient VIO system.

Conclusion

This paper introduces Geotri-VIO, a panoramic VIO framework that leverages a multi-prism projection model to enhance geometric consistency in panoramic images, significantly improving visual information utilization, feature matching accuracy, and overall VIO system performance. Theoretical analysis confirms that the proposed framework ensures geometric consistency and achieves optimal performance with a triangular prism projection. By adapting to both point features

and combined point-line features, the framework demonstrates strong compatibility with various front-end visual features. Experimental results validate that Geotri-VIO consistently outperforms benchmarks in multiple evaluation metrics, achieving optimal geometric consistency with the triangular prism projection, in alignment with theoretical predictions. However, the proposed framework still has certain limitations. First, like most VIO systems, it suffers from accumulated errors over long trajectories, which is a fundamental challenge in odometry-based systems. Second, the reliance on traditional feature extraction and tracking methods makes the system vulnerable to motion blur in high-speed scenarios. In future, we plan to address these limitations by extending the framework to a full SLAM system with loop closure detection and exploring learning-based methods for feature extraction and tracking. These improvements will enhance the performance and applicability of the proposed framework in real world scenarios.

Acknowledgements

Not applicable.

Author contributions

MRD: Methodology design, software implementation, and data analysis. TYH: Validation, visualization, and investigation. ZXL: Draft preparation, review, and editing. WMX: Draft preparation, review, and editing. CH: Conceptualization of the research, supervision, and project administration. All authors read and approved the final manuscript.

Funding

This work was supported in part by the China National Key Research and Development Program under Grant 2022YFB3903804.

Data availability

The datasets used and analysed in this study are available from the corresponding author on reasonable request.

Code availability

The code used in this study is available from the corresponding author on reasonable request.

Declarations

Competing interests

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Received: 4 January 2025 Revised: 28 May 2025 Accepted: 2 June 2025
Published online: 21 July 2025

References

- Amato, G., Falchi, F., & Gennaro, C. (2011). Geometric consistency checks for knn based image classification relying on local features. In: *Proceedings of the fourth international conference on similarity search and applications* (pp. 81–88). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1995412.1995428>
- Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., & Tardós, J. D. (2021). Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6), 1874–1890. <https://doi.org/10.1109/TRO.2021.3075644>
- Cao, S., Lu, X., & Shen, S. (2022). Gvins: Tightly coupled gnss-visual-inertial fusion for smooth and consistent state estimation. *IEEE Transactions on Robotics*, 38(4), 2004–2021. <https://doi.org/10.1109/TRO.2021.3133730>
- Chen, Y., Pan, Y., Liu, R., Zhang, H., Zhang, G., Sun, B., & Zhang, J. (2024). 360orb-slam: A visual slam system for panoramic images with depth completion network. In: *2024 27th international conference on computer supported cooperative work in design (cscwd)* (pp. 717–722). <https://doi.org/10.1109/CSCWD61410.2024.10580875>
- El-Sheimy, N., & Li, Y. (2021). Indoor navigation: State of the art and future trends. *Satellite Navigation*, 2(1), 7. <https://doi.org/10.1186/s43020-021-00041-3>
- Fu, Q., Wang, J., Yu, H., Ali, I., Guo, F., He, Y., & Zhang, H. (2020). Pl-vins: Real-time monocular visual-inertial slam with point and line features. arXiv preprint [arXiv:2009.07462](https://arxiv.org/abs/2009.07462), <https://doi.org/10.48550/arXiv.2009.07462>
- Gao, S., Yang, K., Shi, H., Wang, K., & Bai, J. (2022). Review on panoramic imaging and its applications in scene understanding. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–34. <https://doi.org/10.1109/TIM.2022.3216675>
- Harris, C., Stephens, M., & others (1988). A combined corner and edge detector. *Alvey vision conference*, 15, 10–5244.
- He, Y., Zhao, J., Guo, Y., He, W., & Yuan, K. (2018). Pl-vio: Tightly-coupled monocular visual-inertial odometry using point and line features. *Sensors*, 18(4), 1159. <https://doi.org/10.3390/s18041159>
- Hua, T., Pei, L., Li, T., Yin, J., Liu, G., & Yu, W. (2023). M2c-gvio: Motion manifold constrained aided gnss-visual-inertial odometry for ground vehicles. *Satellite Navigation*, 4(1), 13. <https://doi.org/10.1186/s43020-023-00102-9>
- Huang, H., & Yeung, S.-K. (2022). 360vo: Visual odometry using a single 360 camera. In: *2022 international conference on robotics and automation (icra)* (pp. 5594–5600). <https://doi.org/10.1109/ICRA46639.2022.9812203>
- Huber, P. J. (1992). Robust estimation of a location parameter. *Breakthroughs in statistics: Methodology and distribution* (pp. 492–518). New York: Springer.
- Jiang, Q., Gao, S., Gao, Y., Yang, K., Yi, Z., Shi, H., & Wang, K. (2024). Minimalist and high-quality panoramic imaging with psf-aware transformers. *IEEE Transactions on Image Processing*, 33, 4568–4583. <https://doi.org/10.1109/TIP.2024.3441370>
- Jiang, Q., Shi, H., Sun, L., Gao, S., Yang, K., & Wang, K. (2022). Annular computational imaging: Capture clear panoramic images through simple lens. *IEEE Transactions on Computational Imaging*, 8, 1250–1264. <https://doi.org/10.48550/arXiv.2206.06070>
- Li, H., Yu, H., Wang, J., Yang, W., Yu, L., & Scherer, S. (2021). Ulsd: Unified line segment detection across pinhole, fisheye, and spherical cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 178, 187–202. <https://doi.org/10.1016/j.isprsjprs.2021.06.004>
- Li, L., Huang, H., Yeung, S.-K., & Cheng, H. (2024). Omnigs: Omnidirectional gaussian splatting for fast radiance field reconstruction using omnidirectional images. *arXiv e-prints*, arXiv–2404, <https://doi.org/10.48550/arXiv.2404.03202>
- Li, X., Wang, X., Liao, J., Li, X., Li, S., & Lyu, H. (2021). Semi-tightly coupled integration of multi-gnss ppp and s-vins for precise positioning in gnss-challenged environments. *Satellite Navigation*, 2, 1–14. <https://doi.org/10.1186/s43020-020-00033-9>
- Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Ijcai'81: 7th international joint conference on artificial intelligence*, 2, 674–679.
- Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5), 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>
- Pautrat, R., Barath, D., Larsson, V., Oswald, M. R., & Pollefeys, M. (2023). Deeplsd: Line segment detection and refinement with deep image gradients. In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)* (p.17327–17336).
- Pautrat, R., Lin, J.-T., Larsson, V., Oswald, M. R., & Pollefeys, M. (2021). Sold2: Self-supervised occlusion-aware line description and detection. In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)* (p.11368–11378).
- Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4), 1004–1020. <https://doi.org/10.1109/TRO.2018.2853729>

- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. *2011 international conference on computer vision* (pp. 2564–2571). <https://doi.org/10.1109/ICCV.2011.6126544>
- Scaramuzza, D., Martinelli, A., & Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras. *2006 ieee/rsj international conference on intelligent robots and systems* (pp. 5695–5701). <https://doi.org/10.1109/IROS.2006.282372>
- Shi, J., & others (1994). Good features to track. *1994 proceedings of ieee conference on computer vision and pattern recognition* (pp. 593–600). <https://doi.org/10.1109/CVPR.1994.323794>
- Suárez, I., Buenaposada, J. M., & Baumela, L. (2022). Elsed: Enhanced line segment drawing. *Pattern Recognition*, 127, Article 108619. <https://doi.org/10.1016/j.patcog.2022.108619>
- Sumikura, S., Shibuya, M., & Sakurada, K. (2019). Openvslam: A versatile visual slam framework. *Proceedings of the 27th acm international conference on multimedia* (pp. 2292–2295). <https://doi.org/10.1145/3343031.335053>
- Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., & Randall, G. (2012). Lsd: A line segment detector. *Image Processing On Line*, 2, 35–55. [https://doi.org/10.5201.ipol.2012.gjmr-lsd](https://doi.org/10.5201/ipol.2012.gjmr-lsd)
- Wang, D., Wang, J., Tian, Y., Fang, Y., Yuan, Z., & Xu, M. (2024). Pal-slam2: Visual and visual-inertial monocular slam for panoramic annular lens. *ISPRS Journal of Photogrammetry and Remote Sensing*, 211, 35–48. <https://doi.org/10.1016/j.isprsjprs.2024.03.016>
- Wang, D., Wang, J., Tian, Y., Hu, K., & Xu, M. (2022). Pal-slam: a feature-based slam system for a panoramic annular lens. *Optics express*, 30(2), 1099–1113. <https://doi.org/10.1364/OE.447893>
- Wang, J., Yang, K., Gao, S., Sun, L., Zhu, C., Wang, K., & Bai, J. (2022). High-performance panoramic annular lens design for real-time semantic segmentation on aerial imagery. *Optical Engineering*, 61(3), 035101–035101. <https://doi.org/10.1117/1.OE.61.3.035101>
- Wang, Y., Cai, S., Li, S.-J., Liu, Y., Guo, Y., Li, T., & Cheng, M.-M. (2018). Cubemap-slam: A piecewise-pinhole monocular fisheye slam system. *Asian conference on computer vision* (pp. 34–49).
- Wang, Z., Yang, K., Shi, H., Li, P., Gao, F., Bai, J., & Wang, K. (2024). Lf-vislam: A slam framework for large field-of-view cameras with negative imaging plane on mobile agents. *IEEE Transactions on Automation Science and Engineering*, 21(4), 6321–6335. <https://doi.org/10.1109/TASE.2023.3324495>
- Wang, Z., Yang, K., Shi, H., Li, P., Gao, F., & Wang, K. (2022). Lf-vio: A visual-inertial-odometry framework for large field-of-view cameras with negative plane. *2022 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 4423–4430). <https://doi.org/10.1109/IROS47612.2022.9981217>
- Wang, Z., Yang, K., Shi, H., Zhang, Y., Xu, Z., Gao, F., & Wang, K. (2024). Lf-pgvio: A visual-inertial-odometry framework for large field-of-view cameras using points and geodesic segments. *IEEE Transactions on Intelligent Vehicles*, 1–13. <https://doi.org/10.1109/TV.2024.3376829>
- Wu, Q., Xu, X., Chen, X., Pei, L., Long, C., Deng, J., & Yu, W. (2024). 360-vio: A robust visual-inertial odometry using a 360° camera. *IEEE Transactions on Industrial Electronics*, 71(9), 11136–11145. <https://doi.org/10.1109/TIE.2023.3337541>
- Xu, H., Zhang, Y., Zhou, B., Wang, L., Yao, X., Meng, G., & Shen, S. (2022). Omni-swarm: A decentralized omnidirectional visual-inertial-uwb state estimation system for aerial swarms. *IEEE Transactions on Robotics*, 38(6), 3374–3394. <https://doi.org/10.1109/TRO.2022.3182503>
- Xu, K., Hao, Y., Yuan, S., Wang, C., & Xie, L. (2024). Airslam: An efficient and illumination-robust point-line visual slam system. *arXiv preprint arXiv: 2408.03520*, <https://doi.org/10.1109/TRO.2025.3539171>
- Xu, W., Cai, Y., He, D., Lin, J., & Zhang, F. (2022). Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4), 2053–2073. <https://doi.org/10.1109/TRO.2022.3141876>
- Yan, J., Zheng, Y., Yang, J., Mihaylova, L., Yuan, W., & Gu, F. (2024). Plpf-vslam: An indoor visual slam with adaptive fusion of point-line-plane features. *Journal of Field Robotics*, 41(1), 50–67. <https://doi.org/10.1002/rob.22242>
- Yang, B. (2021). A mathematical investigation on the distance-preserving property of an equidistant cylindrical projection. *arXiv preprint arXiv: 2101.03972*, <https://doi.org/10.48550/arXiv.2101.03972>
- Yang, H., Yuan, J., Gao, Y., Sun, X., & Zhang, X. (2023). Uplp-slam: Unified point-line-plane feature fusion for rgb-d visual slam. *Information Fusion*, 96, 51–65. <https://doi.org/10.1016/j.inffus.2023.03.006>
- Yang, K., Hu, X., Bergasa, L. M., Romera, E., & Wang, K. (2019). Pass: Panoramic annular semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 21(10), 4171–4185. <https://doi.org/10.1109/TITS.2019.2938965>
- Zhang, G., Lee, J. H., Lim, J., & Suh, I. H. (2015). Building a 3-d line-based map using stereo slam. *IEEE Transactions on Robotics*, 31(6), 1364–1377. <https://doi.org/10.1109/TRO.2015.2489498>
- Zhang, L., & Koch, R. (2013). An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. *Journal of visual communication and image representation*, 24(7), 794–805. <https://doi.org/10.1016/j.jvcir.2013.05.006>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.