

RSL-BA: Rolling Shutter Line Bundle Adjustment

Yongcong Zhang^{1,3,*}, Bangyan Liao^{2,*}, Yifei Xue^{1,3}, Chen Lu⁴,
Peidong Liu², and Yizhen Lao^{1,†}

¹Hunan University ²Westlake University ³DaHe.AI
⁴Dreame Technology Company

Abstract. The line is a prevalent element in man-made environments, inherently encoding spatial structural information, thus making it a more robust choice for feature representation in practical applications. Despite its apparent advantages, previous rolling shutter bundle adjustment (RSBA) methods have only supported sparse feature points, which lack robustness, particularly in degenerate environments. In this paper, we introduce the first rolling shutter line-based bundle adjustment solution, *RSL-BA*. Specifically, we initially establish the rolling shutter camera line projection theory utilizing Plücker line parameterization. Subsequently, we derive a series of reprojection error formulations which are stable and efficient. Finally, we theoretically and experimentally demonstrate that our method can prevent three common degeneracies, one of which is first discovered in this paper. Extensive synthetic and real data experiments demonstrate that our method achieves efficiency and accuracy comparable to existing point-based rolling shutter bundle adjustment solutions.

Keywords: Rolling Shutter · Bundle Adjustment

1 Introduction

Bundle Adjustment (BA) is a crucial step in multi-view 3D reconstruction, as it jointly optimizes camera poses and scene structure. Although feature point-based BA [6, 23, 26] currently dominates the academic landscape, line-based BA has been gaining increasing attention from researchers [16, 28]. Lines, being one of the most common features in man-made environments, effectively describe the structural information of 3D scenes, in contrast to points which rely solely on local image patches. This structural information provides robust constraints for visual reconstruction [27], making line a more suitable choice in challenging scenarios.

In parallel with classical bundle adjustment [26], several methods [1, 13] have incorporated the rolling shutter camera model to simultaneously estimate camera

* Equal contribution

† Corresponding author: (yizhenlao@hnu.edu.cn)

Project page: <https://github.com/zhangtaxue/RSL-BA>

poses, structure, and instantaneous motion parameters. However, all existing rolling shutter bundle adjustment (RSBA) methods rely solely on point features, while **line features have never been addressed in RSBA**. We identify two challenges that hinder the direct combination of line-based BA and the rolling shutter camera model.

The first challenge arises from the time-dependent exposure of RS cameras. Unlike CCD cameras and their global shutter (GS) counterparts, RS cameras capture images in a scanline-by-scanline manner. Consequently, as illustrated in Fig. 1, images taken by moving RS cameras exhibit distortions known as the RS effect. This effect curves the projection of each 3D straight line, making it non-trivial to directly transfer the line-based BA to the rolling shutter camera setting.

The second challenge is associated with degeneracy. As shown in [1, 30], degeneracy is one of the main obstacles in RSBA since iterative optimization can easily be collapsed in some degenerate solutions which are far from the ground truth solutions. Although several strategies have been proposed to mitigate degeneracy in RSBA [1, 12, 14], a specific strategy to address degeneracy in line-based RSBA is still lacking.

In conclusion, an **accurate** and **robust** solution to line-based RSBA is still missing. Such a method would be vital in the potential widespread deployment of 3D vision with RS imaging systems.

1.1 Related Works

Video-based RSBA. The assumption of smooth continuous trajectories is widely employed for RS video inputs to reduce the optimization parameter space and enhance algorithm robustness. In [8], Hedborg *et al.* present an RSBA algorithm that interpolates the motion between consecutive frames. Zhuang *et al.* [29] further propose an optical flow-based RSBA, which are developed to recover the relative pose of an RS camera that undergoes constant velocity and acceleration motion, respectively. Following this assumption, a spline-based camera trajectory motion model is proposed by [20].

Unordered RSBA. As the unordered image set is the standard input for SfM. Albl *et al.* first addresses this setting by explicitly velocity modeling and optimization. Besides, a planar degeneracy configuration has also been disclosed in [1]. While in [12], Lao *et al.* propose a camera-based RSBA to simulate the actual camera projection, exhibiting the degeneracy resilience ability. In [10], Ito *et al.* establish the equivalence between self-calibrated SfM and RSSFM based on the pure rotation instantaneous motion model and affine camera assumption, while the work of [13] draws the equivalence between RSSfM and non-rigid SfM. Recently, Liao *et al.* proposed two techniques to boost the accuracy and robustness of RSBA in [14].

Line-based GSBA. In [2, 25], the authors propose a comprehensive line-based SFM pipeline, while [9, 18] introduce line-based incremental SFM frameworks. Wei *et al.* in [28] employs planes and points to guide matching, achieving efficient line reconstruction. Recently, Lui *et al.* introduced LIMAP in [16], where the

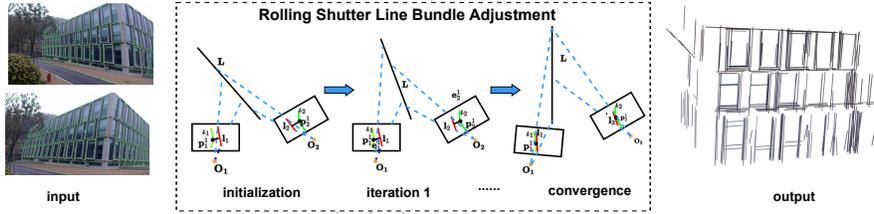


Fig. 1: The pipeline of proposed *RSL-BA* with some example results. Starting with the input of some RS images (left), the optimization of camera poses and 3D line coordinates is performed using proposed *RSL-BA* (middle), yielding well-reconstructed 3D line segments and accurately estimated camera poses (right).

authors devised various strategies and algorithms to achieve efficient and precise line reconstruction. In [15, 21, 31], hybrid strategies combining lines and feature points are utilized to enhance the accuracy and robustness of SLAM.

To this end, we propose a novel *RSL-BA* algorithm to solve the line-based RSBA problem. Specifically, we first establish the rolling shutter line projection theory using the Plücker line parameterization and subsequently derive a series of reprojection error formulations. We claim that the proposed reprojection error formulations are efficient and exhibit the degeneration-resistant ability. We also provide complete degeneration-resistant proof of our proposed error formulation. Our contributions are summarized as follows:

- To the best of our knowledge, this is the first line-based RSBA solution, which serves as the foundation for line-based or point-line-hybrid RS-SfM and RS-SLAM.
- we theoretically and experimentally demonstrate that the proposed *RSL-BA* can prevent three common degeneracies in RSBA, one of the degenerate cases is first discovered in this paper.
- The extensive evaluations in both synthetic and real datasets exhibit the comparable efficiency and accuracy of the proposed method over previous works.

2 Background

In this section, we review the rolling shutter projection model in Sec. 2.1 and the parameterization of 3D lines in Sec. 2.2.

2.1 Rolling Shutter Camera Projection Mode

Let $\mathbf{R}(v) \in \mathbf{SO}(3)$ and $\mathbf{t}(v) \in \mathbb{R}^3$ represent the camera rotation and translation, respectively, when the row index v of measurement is acquired. Denote the intrinsic camera parameters as \mathbf{K} , and the initial rotation and translation of the

camera at $v = 0$ as \mathbf{R}_0 and \mathbf{t}_0 , respectively. The projection matrix of the global shutter camera is defined as $\mathbf{P}_0 = \mathbf{K}[\mathbf{R}_0, \mathbf{t}_0]$. Given the assumption of constant camera motion during frame capture, which is commonly made in RS 3D vision tasks [1, 4, 13], we can model the instantaneous motion as:

$$\mathbf{R}(v) = (\mathbf{I} + [\boldsymbol{\omega}]_{\times} v) \mathbf{R}_0, \quad \mathbf{t}(v) = \mathbf{t}_0 + \mathbf{d}v, \quad (1)$$

where $\mathbf{d} = [d_x, d_y, d_z]^{\top}$ is the translational velocity vector and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^{\top}$ is the rotational velocity vector, and $[\boldsymbol{\omega}]_{\times}$ represents the skew-symmetric matrix of vector $\boldsymbol{\omega}$. Then, the projection matrix of the camera under the RS model can be defined as:

$$\begin{aligned} \mathbf{P}_v &= \mathbf{K}[\mathbf{R}_v, \mathbf{t}_v] = \mathbf{P}_0 + v\mathbf{Q} \\ \mathbf{P}_0 &= \mathbf{K}[\mathbf{R}_0, \mathbf{t}_0] \quad \mathbf{Q} = \mathbf{K}[[\boldsymbol{\omega}]_{\times} \mathbf{R}_0, \mathbf{d}]. \end{aligned} \quad (2)$$

2.2 3D Line Segments Parameterization

We employ the Plücker matrix to parameterize the 3D line [7]. Specifically, given two homogeneous 3D points $P_a = [a_1, a_2, a_3, 1]^{\top}$ and $P_b = [b_1, b_2, b_3, 1]^{\top}$. The Plücker matrix \mathbf{L} can be defined as:

$$\begin{aligned} \mathbf{L} &= P_a P_b^{\top} - P_b P_a^{\top} \\ &= \begin{bmatrix} 0 & a_1 b_2 - a_2 b_1 & a_1 b_3 - a_3 b_1 & a_1 - b_1 \\ a_2 b_1 - a_1 b_2 & 0 & a_2 b_3 - a_3 b_2 & a_2 - b_2 \\ a_3 b_1 - a_1 b_3 & a_3 b_2 - a_2 b_3 & 0 & a_3 - b_3 \\ b_1 - a_1 & b_2 - a_2 & b_3 - a_3 & 0 \end{bmatrix} = \begin{bmatrix} 0 & l_{12} & l_{13} & l_{14} \\ -l_{12} & 0 & l_{23} & l_{24} \\ -l_{13} & -l_{23} & 0 & l_{34} \\ -l_{14} & -l_{24} & -l_{34} & 0 \end{bmatrix}, \end{aligned} \quad (3)$$

where the $[l_{14}, l_{24}, l_{34}]^{\top}$ represents the direction of this straight line while the $[-l_{23}, l_{13}, -l_{12}]^{\top}$ represents the normal direction of this line. The projection of 3D Plücker matrix is the skew-symmetric matrix of a 2D normalized line. It can be related to each other as:

$$[\mathbf{l}_{gs}]_{\times} = [\mathbf{R}, \mathbf{t}] \mathbf{L} [\mathbf{R}, \mathbf{t}]^{\top} \quad (4)$$

3 Methodology

In Sec. 3.1, we initially establish the rolling shutter projection equations using the Plücker line parameterization. These equations establish a connection between the 3D lines and the curve parameters on the image plane. Then in Sec. 3.2, we present a series of line-based reprojection error equations. On top of the error definition, we can formulate the entire rolling shutter line bundle adjustment problem in Sec. 3.3. Finally, in Sec. 3.4, we demonstrate the degeneracy resilience capability of the proposed error equations.

3.1 Rolling Shutter Line Projection Formulation

As previously explained, the curved rolling shutter projection of 3D lines occurs because of the camera’s nonlinear motion. Using the Plücker parameterization method, we can derive explicit expressions for the curve parameters.

Firstly, we construct the skew-symmetric matrix $[\mathbf{l}_{rs}]_{\times}$ of the instantaneous projection line by combining Eq. (2) and Eq. (4):

$$\begin{aligned} [\mathbf{l}_{rs}]_{\times} &= \mathbf{P}_v \mathbf{L} \mathbf{P}_v^{\top} \\ &= \mathbf{P}_0 \mathbf{L} \mathbf{P}_0^{\top} + v(\mathbf{P}_0 \mathbf{L} \mathbf{Q}^{\top} + \mathbf{Q} \mathbf{L} \mathbf{P}_0^{\top}) + v^2 \mathbf{Q} \mathbf{L} \mathbf{Q}^{\top} \\ &= \mathbf{A}_1 + v \mathbf{A}_2 + v^2 \mathbf{A}_3 \end{aligned} \quad (5)$$

Secondly, as $\mathbf{l}_{rs} = [l_1, l_2, l_3]^{\top}$ and $[\mathbf{l}_{rs}]_{\times} = \mathbf{A}_1 + v \mathbf{A}_2 + v^2 \mathbf{A}_3$, we can determine the instantaneous projected line parameter as follows:

$$\begin{cases} l_1 = \mathbf{A}_1^{32} + v \mathbf{A}_2^{32} + v^2 \mathbf{A}_3^{32} \\ l_2 = \mathbf{A}_1^{13} + v \mathbf{A}_2^{13} + v^2 \mathbf{A}_3^{13} \\ l_3 = \mathbf{A}_1^{21} + v \mathbf{A}_2^{21} + v^2 \mathbf{A}_3^{21} \end{cases} \quad (6)$$

Ultimately, we note that the instantaneous projected line parameter fulfills this $l_1 u + l_2 v + l_3 = 0$ equation, which we can expand to obtain the final curve equation, denoted as $\boldsymbol{\iota}$:

$$\begin{aligned} \boldsymbol{\iota} : \mathbf{A}_3^{13} v^3 + \mathbf{A}_3^{32} u v^2 + (\mathbf{A}_2^{13} + \mathbf{A}_3^{21}) v^2 + \mathbf{A}_2^{32} u v \\ + (\mathbf{A}_1^{13} + \mathbf{A}_2^{21}) v + \mathbf{A}_1^{32} u + \mathbf{A}_1^{21} = 0 \end{aligned} \quad (7)$$

Our conclusion aligns with that of Lao *et al.* [11], yielding an identical polynomial curve degree. Nevertheless, our utilization of a 4-DoF orthogonal representation for 3D lines has distinct benefits when compared to Lao’s 5-DoF over-parameterized representation. These advantages include the capability to conduct unconstrained optimization on spatial lines directly.

Remark. The geometric interpretation underlying this derivation is quite straightforward. Given the parameters of a spatial line and the current camera pose, if point (u, v) lies on the curve obtained by projecting this spatial line, then when the RS camera scans to the v -th row, the projection of the spatial line onto the image plane follows Eq. (6). Therefore, it is evident that point (u, v) must be located on this line. Based on this reasoning, we can deduce Eq. (7).

3.2 Line-based Rejection Errors

As claimed in Sec. 3.1, a 3D straight line will become a polynomial curve on the captured image. In this section, we will establish a series of re-projection errors which are stable and robust. We first leverage the similar intuition in Sec. 3.1 to obtain the distance error representation. As shown in Fig. 2(a), Let there be a point $\mathbf{q} = [u, v]^{\top}$ on the observed curve $\boldsymbol{\iota}$, then the pose of the camera when exposing that point is $\mathbf{R}_v, \mathbf{t}_v$, then projecting the 3D line \mathbf{L} onto the image

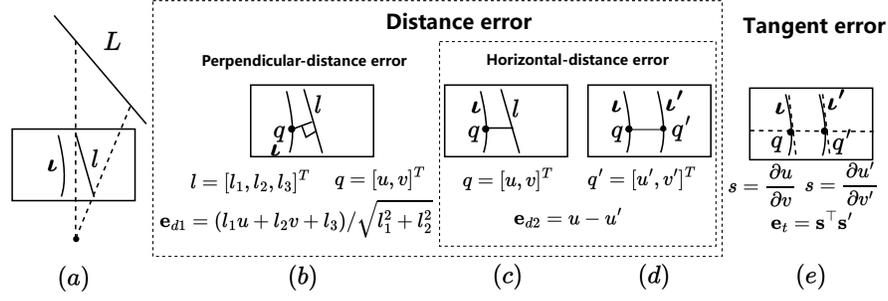


Fig. 2: The illustration of proposed line-based reprojection errors given the spatial line L and observed curve ι . Instead of directly measuring the distance between the observed curve and the projected curve, we split each curve into multiple points q , and aggregate the distance errors. Specifically, we first utilize the virtual projected line l when observing it, as shown in (a). To measure the distance between each point q and the corresponding virtual lines, we can use the perpendicular distance (b), the horizontal distance (c,d) or the tangent distance (e) as our basic metrics.

plane, this results in the virtual 2D line $\mathbf{l} = [l_1, l_2, l_3]^\top$. Under the condition of ground truth, this virtual line must theoretically intersect with that point \mathbf{q} . This constraint directly leads us to the two variants of distance error, which we have defined below.

Perpendicular Distance Error e_{d1} . The first distance error formulation, named perpendicular distance error, measures the shortest distance from point \mathbf{q} to the line \mathbf{l} , as shown in Fig. 2(b). The error is defined as:

$$e_{d1} = \frac{l_1 u + l_2 v + l_3}{\sqrt{l_1^2 + l_2^2}}, \quad (8)$$

where the line parameters $[l_1, l_2, l_3]$ can be calculated through Eq. (6).

Horizontal Distance Error e_{d2} . The second distance error formulation, named horizontal distance error, measures the u -axis distance from point \mathbf{q} to the line \mathbf{l} , as shown in Fig. 2(c). The error can be defined as:

$$e_{d2} = u - u', \quad (9)$$

where the corresponding u' coordinate of virtual line \mathbf{l} at the v -th row can be calculated as:

$$u' = -\frac{l_2 v + l_3}{l_1}, \quad (10)$$

As shown in Fig. 2(d), the e_{d2} distance error is equivalent to the u -axis distance from the point to the projection curve.

Tangent Error \mathbf{e}_t . While distance error provides an elegant constraint, it is insufficient to rely solely on distance error in certain scenarios. To further enhance the robustness, we introduce a second type of tangent error. As shown in Fig. 2(e), we constrain the tangent of the projected curve $\boldsymbol{\nu}'$ to match the observed tangent direction. The error can be defined as:

$$\mathbf{e}_t = \mathbf{s}^\top \mathbf{s}', \quad (11)$$

where \mathbf{s} represents the tangent direction observation. The \mathbf{s}' can be calculated analytically following the implicit function theorem as:

$$\mathbf{s}' = \left[\frac{1}{\sqrt{1+s^2}}, \frac{s}{\sqrt{1+s^2}} \right], \quad s = \frac{\partial u}{\partial v} = -\frac{\partial \boldsymbol{\nu} / \partial v}{\partial \boldsymbol{\nu} / \partial u} \quad (12)$$

3.3 Rolling Shutter Line Bundle Adjustment

The non-linear least squares solvers are used to find an optimal solution $\boldsymbol{\theta}^*$ including camera poses $\mathbf{R}^*, \mathbf{t}^*$, instantaneous motion $\boldsymbol{\omega}^*, \mathbf{d}^*$ and 3D lines $\boldsymbol{\tau}^*$ by minimizing the reprojection error \mathbf{e}_i^j from line i to camera j over all the camera index in set \mathcal{F} and corresponding observed 3D lines index in subset \mathcal{P}_j :

$$\boldsymbol{\theta}^* = \{\boldsymbol{\tau}^*, \mathbf{R}^*, \mathbf{t}^*, \boldsymbol{\omega}^*, \mathbf{d}^*\} = \arg \min_{\boldsymbol{\theta}} \sum_{j \in \mathcal{F}} \sum_{i \in \mathcal{P}_j} \left\| \mathbf{e}_i^j \right\|_2^2. \quad (13)$$

On top of errors proposed in Sec. 3.2 , we provide two variants of reprojection error:

$$\mathbf{e}_1 = \mathbf{e}_{d1} + \lambda_1 \mathbf{e}_t, \quad \mathbf{e}_2 = \mathbf{e}_{d2} + \lambda_2 \mathbf{e}_t, \quad (14)$$

where λ_1, λ_2 are the relative weight. We have experimentally validated the optimality of these two error variants in Sec. 4.1. After definition, this non-linear least square problem can be iteratively solved through Levenberg-Marquardt [19] method.

Besides, the full analytical Jacobian has been derived in the supplementary material.

3.4 Degeneracy Analysis

In this section, we conducted theoretical analysis on the resilience of *RSL-BA* to three types of degeneracy scenarios. These three degeneracy scenarios include plane degeneracy case [1], 2-views pure translation case [30], and X-Y pure translation degeneracy case disclosed by us. We assume the positive direction of X Y Z axis is right down forward, respectively.

① Resistance to Plane Degeneracy [1]. We first briefly describe the conditions under which the plane degeneration occurs: when the camera’s y-axis is parallel, if a rotational velocity $\boldsymbol{\omega} = [-1, 0, 0]^\top$ is given to the camera, then the

reconstructed 3D points will be compressed onto a plane perpendicular to the y -axis, and satisfy that the reprojection error is 0.

For line features, we also assume $\mathbf{R}_0 = \mathbf{I}$, $\mathbf{t}_0 = \mathbf{0}$, $\boldsymbol{\omega} = [-1, 0, 0]^\top$, $\mathbf{d} = \mathbf{0}$. Let there be two points, \mathbf{P}_a and \mathbf{P}_b , on the spatial line \mathbf{L} . $\mathbf{P}_a = [a_1, a_2, a_3, 1]^\top$ and $\mathbf{P}_b = [b_1, b_2, b_3, 1]^\top$. When the space is compressed into a plane, the new coordinates of \mathbf{P}_a and \mathbf{P}_b are $\mathbf{P}'_a = [a_1, 0, a_3, 1]^\top$ and $\mathbf{P}'_b = [b_1, 0, b_3, 1]^\top$, respectively. By referring to Sec. 3.1, we can solve for the projection parameters of the two-dimensional curve at this moment. The parameters of the curve at this time are:

$$\begin{cases} \mathbf{A}_3^{13} = \mathbf{A}_3^{32} = \mathbf{A}_2^{13} = \mathbf{A}_3^{21} = \mathbf{A}_2^{32} = \mathbf{A}_1^{32} = \mathbf{A}_1^{21} = 0 \\ \mathbf{A}_1^{13} = a_3b_1 - a_1b_3 \\ \mathbf{A}_2^{21} = a_1b_3 - a_3b_1 \end{cases} \quad (15)$$

Substituting the parameters into the curve expression Eq. (7), we obtain $0 = 0$, indicating that u and v are unrestricted. Under the influence of $\boldsymbol{\omega} = [-1 \ 0 \ 0]^\top$, the camera's imaging rows are always coplanar with the line, making the distance from any point on the image to the line zero and satisfying the point-to-line distance error function (Sec. 3.2). However, the slope constraint cannot be met. Using this projection equation to calculate the tangent direction of any point on the curve results in indeterminate forms $[\frac{0}{0}, \frac{0}{0}]$, which do not match the measured tangent direction. Therefore, using the tangent direction of points on the curve as the error function prevents this degeneration. As verified in Fig 5, our proposed *RSL-BA* can suppress plane degeneration, whereas conventional rolling shutter point-based bundle adjustment method NMRSSBA [1] converges to a degenerate solution.

② Resistance to 2-views Pure Translation Degeneracy [30]. We first briefly describe the conditions under which this degeneration occurs: when the camera moves in a straight line to take two pictures, assigning any translational velocity to the camera for both shots, it is possible to find new 3D points that satisfy the condition that the reprojection error of the points under this velocity is zero.

Let's assume the real pose of the camera before the degeneration is: $\mathbf{R}_0 = \mathbf{I}$, $\mathbf{t}_0 = \mathbf{0}$, $\boldsymbol{\omega} = [0, 0, 0]^\top$, $\mathbf{d} = [d_1, d_2, d_3]^\top$, $\mathbf{L} = (l_{12}, l_{13}, l_{23}, l_{14}, l_{24}, l_{34})$. In this case, substitute these parameters into Eq. (7) we can get curve expression:

$$(l_{14}d_3 - l_{34}d_1)v^2 + (l_{34}d_2 - l_{24}d_3)uv + (l_{24}d_1 - l_{14}d_2 + l_{13})v - l_{23}u - l_{12} = 0 \quad (16)$$

After the degeneration occurs, $\mathbf{R}'_0 = \mathbf{I}$, $\mathbf{t}'_0 = \mathbf{0}$, $\boldsymbol{\omega}' = [0 \ 0 \ 0]^\top$, $\mathbf{d}' = r[d_1, d_2, d_3]$, $\mathbf{L} = (l'_{12}, l'_{13}, l'_{23}, l'_{14}, l'_{24}, l'_{34})$. In this case, the curve expression is:

$$r(l'_{14}d_3 - l'_{34}d_1)v^2 + r(l'_{34}d_2 - l'_{24}d_3)uv + (r(l'_{24}d_1 - l'_{14}d_2) + l'_{13})v - l'_{23}u - l'_{12} = 0 \quad (17)$$

To make the reprojection error zero, the curve expressions before and after degeneration must be the same, meaning the corresponding coefficients are in

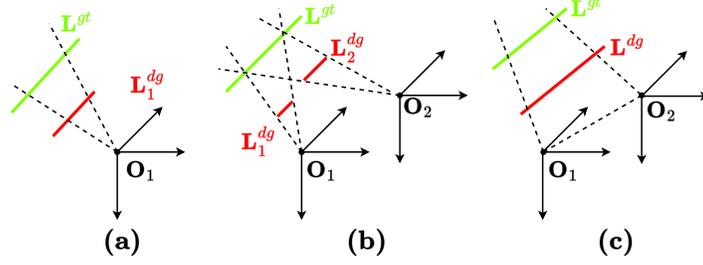


Fig. 3: The illustration of degeneracy resistance of our proposed method in 2-views pure translation degeneracy. (a) Degeneration in single-view scenarios. (b) Suppression of degeneracy using line features in dual-view scenarios. (c) Inability of line features to suppress degeneracy in special dual-view scenarios

proportion.

$$\begin{cases} l_{14}d_3 - l_{34}d_1 = sr(l'_{14}d_3 - l'_{34}d_1) \\ l_{34}d_2 - l_{24}d_3 = sr(l'_{34}d_2 - l'_{24}d_3) \\ l_{24}d_1 - l_{14}d_2 + l_{13} = sr(l'_{24}d_1 - l'_{14}d_2) + sl'_{13} \\ l_{23} = sl'_{23} \\ l_{12} = sl'_{12} \end{cases} \quad (18)$$

Substituting the first and second equations into the third equation yields $l_{13} = sl'_{13}$. Combining this with:

$$\begin{cases} l_{12}l_{34} - l_{13}l_{24} + l_{14}l_{23} = 0 \\ l'_{12}l'_{34} - l'_{13}l'_{24} + l'_{14}l'_{23} = 0 \end{cases} \quad (19)$$

we obtain:

$$l'_{12} = \frac{l'_{12}}{s}, \quad l'_{13} = \frac{l_{13}}{s}, \quad l'_{23} = \frac{l_{23}}{s}, \quad l'_{14} = \frac{l_{14}}{sr}, \quad l'_{24} = \frac{l_{24}}{sr}, \quad l'_{34} = \frac{l_{34}}{sr} \quad (20)$$

Note that the direction and normal direction of the line remain unchanged, indicating that the degenerated line stays in the original plane and parallel to the original line, as shown in Fig. 3(1), where \mathbf{L}^{gt} is the original line, and \mathbf{L}_1^{dg} is the degenerated line. In the dual-frame scenario (Fig.3(2)), when the plane $\mathbf{O}_1\mathbf{L}^{gt}$ is not coplanar with the plane $\mathbf{O}_2\mathbf{L}^{gt}$, their degenerated states \mathbf{L}_1^{dg} and \mathbf{L}_2^{dg} cannot coincide (except at \mathbf{L}^{gt}). In this case, line constraints can prevent degeneracy. However, when \mathbf{L}^{gt} , \mathbf{O}_1 , and \mathbf{O}_2 are coplanar (Fig.3(3)), line constraints cannot suppress degeneracy, though this is a very special condition requiring all lines to meet this criterion. This demonstrates that line constraints are more effective than point constraints in preventing degeneracy.

③ Resistance to X-Y Pure Translation Degeneracy (our new finding). We present a pure translation scenario without the z-direction, suitable for

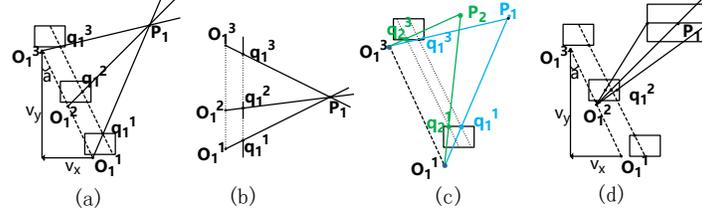


Fig. 4: In the case of X-Y pure translation, optimization compresses points into a straight line. In a single view, pixel points parallel to the motion trajectory back-project to the same 3D point ((a) and (b)). (c) Different intersection points in space are equidistant from the camera plane. (d) All intersection points formed by back-projection constitute a straight line.

multi-view BA, compressing space points into a straight line. Unlike the second type of degeneration, this does not depend on the number of images.

Firstly, considering a single view, as shown in Fig.4(1), during exposure, the camera moves from position \mathbf{O}_1^1 to \mathbf{O}_1^3 , where pixel \mathbf{q}_1^1 is located on the first row and \mathbf{q}_1^3 is located on the H -th row. Their corresponding spatial points are located on the rays $\mathbf{O}_1^1\mathbf{q}_1^1$ and $\mathbf{O}_1^3\mathbf{q}_1^3$, intersecting at point \mathbf{P}_1 . Actually, any point on the line $\mathbf{q}_1^1\mathbf{q}_1^3$ intersects with its corresponding 3D point at \mathbf{P}_1 during exposure. Let \mathbf{O}_1^2 be the camera position when exposed to the h_1^2 -th row, with corresponding feature point \mathbf{q}_1^2 . We only consider the plane $\mathbf{O}_1^1\mathbf{P}_1\mathbf{O}_1^3$ in Fig.4(2). The motion of \mathbf{O} is uniform rectilinear motion thus, there exists a length relationship:

$$\begin{cases} l_{\mathbf{q}_1^1\mathbf{q}_1^2} = \frac{h_1^2}{H}l_{\mathbf{q}_1^1\mathbf{q}_1^3}, & l_{\mathbf{q}_1^3\mathbf{q}_1^2} = (1 - \frac{h_1^2}{H})l_{\mathbf{q}_1^1\mathbf{q}_1^3} \\ l_{\mathbf{O}_1^1\mathbf{O}_1^2} = \frac{h_1^2}{H}l_{\mathbf{O}_1^1\mathbf{O}_1^3}, & l_{\mathbf{O}_1^3\mathbf{O}_1^2} = (1 - \frac{h_1^2}{H})l_{\mathbf{O}_1^1\mathbf{O}_1^3} \end{cases} \quad (21)$$

so we have:

$$\frac{l_{\mathbf{q}_1^1\mathbf{q}_1^2}}{l_{\mathbf{O}_1^1\mathbf{O}_1^2}} = \frac{l_{\mathbf{q}_1^3\mathbf{q}_1^2}}{l_{\mathbf{O}_1^3\mathbf{O}_1^2}} = \frac{l_{\mathbf{q}_1^1\mathbf{q}_1^3}}{l_{\mathbf{O}_1^1\mathbf{O}_1^3}} \quad (22)$$

Therefore, the ray $\mathbf{O}_1^2\mathbf{q}_1^2$ will also intersect at point \mathbf{P}_1 . For every pixel on a line parallel to $\mathbf{O}_1^1\mathbf{O}_1^3$, it can be traced back to a different spatial point. For pixel points on different lines parallel to the motion trajectory, as shown in Fig. 4(3), because the distance $l_{\mathbf{q}_1^1\mathbf{q}_1^3}$ is the same and the camera focal length remains unchanged, according to similar triangles, these spatial points have equal distances to the image plane. Simultaneously, due to symmetry, these points also lie in the plane that is perpendicular to and bisects the line segment $\mathbf{O}_1^1\mathbf{O}_1^3$. Therefore, the intersection line of these two planes represents the final degenerated configuration, which is a line parallel to the x-axis, as shown in Fig.4(4). In multiple views, with no z-direction movement, the imaging planes of different cameras coincide. We assign the same camera configuration to different views, including rotation, translation, angular velocity, and linear velocity. From the single-view case, corresponding points on the pixel plane parallel to the motion trajectory reconstruct to the same spatial point, with zero reprojection error.

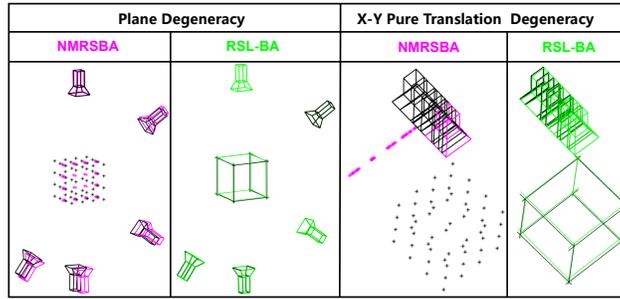


Fig. 5: Comparison of degenerate resistance ability between NMRSBA [1] and proposed *RSL-BA*. Ground truth camera poses and structure are colored with black. This example illustrates that our proposed *RSL-BA* has the resistance ability against the plane and X-Y pure translation degeneracy.

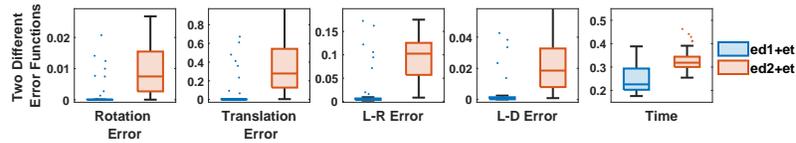


Fig. 6: Accuracy and computational time comparison of the proposed two combinations of reprojection errors.

For the line constraints proposed in this paper, *RSL-BA* can still effectively suppress degeneracy. The proof process is similar to the previous section Sec. 3.4, with the only modification being setting $d_3 = 0$ in the camera’s motion direction. As verified in Fig 5, our proposed *RSL-BA* can suppress X-Y pure translation degeneracy, whereas conventional rolling shutter point-based bundle adjustment method NMRSBA [1] converges to a degenerate solution.

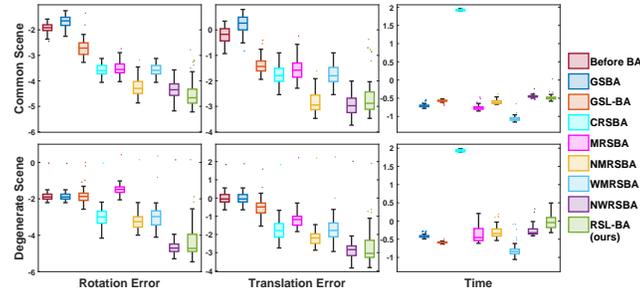
4 Experiments

We compare our method with two SOTA GS-based-method: 1) GSBA [17], 2) GLBA [25], and four SOTA RS-based-method: 1) MRSBA [5], 2) NMRSBA [1], 3) WMRSBA [14], 4) NWRSBA [14]. The experiments are conducted on a laptop with an Intel i7 CPU and all algorithms are implemented in MATLAB.

Evaluation metrics. The accuracy measures employed encompass standard metrics such as rotation error, translation error, and algorithm efficiency. When conducting experiments related to line BA, we also incorporate error metrics that assess the accuracy of line reconstruction. Given the ground truth rotation \mathbf{R}_g , translation \mathbf{t}_g , and space line $\mathbf{L}_g = (\mathbf{n}_g^\top, \mathbf{a}_g^\top)^\top$ with an arbitrary point \mathbf{P}_g on it. Similarly, the optimized parameters are denoted as $\mathbf{R}_d, \mathbf{t}_d, \mathbf{L}_d = (\mathbf{n}_d^\top, \mathbf{a}_d^\top)^\top$,

Table 1: Robustness analysis against multiple levels of Gaussian noise (px).

noise level (px)	0.1	0.5	1.0	1.5	2.0
Rotation Error	7.96e-6	9.24e-6	1.50e-5	1.71e-5	2.71e-5
Translation Error	2.32e-4	5.93e-4	8.91e-4	9.54e-4	1.55e-3
L-R Error	3.36e-3	3.83e-3	5.14e-3	5.22e-3	6.42e-3
L-D Error	2.22e-4	2.43e-4	3.90e-4	4.43e-4	4.75e-4

**Fig. 7:** Comparison of accuracy and time among different algorithms.

\mathbf{P}_d , respectively. The evaluation metrics are defined as:

$$\begin{cases} e_{Rotation} = \arccos\left(\frac{\text{Trace}(\mathbf{R}_g^T \mathbf{R}_d) - 1}{2}\right) \\ e_{Translation} = \arccos\left(\frac{\mathbf{t}_g^T \mathbf{t}_d}{\|\mathbf{t}_g\| \cdot \|\mathbf{t}_d\|}\right) \\ e_{L-R} = \arccos\left(\frac{\mathbf{a}_g^T \mathbf{a}_d}{\|\mathbf{a}_g\| \cdot \|\mathbf{a}_d\|}\right) \\ e_{L-D} = \frac{|(\mathbf{a}_g \times \mathbf{a}_d) \cdot (\mathbf{P}_d - \mathbf{P}_g)|}{\|\mathbf{a}_g \times \mathbf{a}_d\|} \end{cases} \quad (23)$$

4.1 Synthetic Experiments

Which error term in Sec. 3.2 is better? In the synthetic environment, We set up a cubic box with 12 edges in 3D space and simulated a series of cameras around it to perform RS imaging. The experimental results are shown in Fig. 6. It is evident that, the first type of reprojection error combination performs much better than the second combination in terms of accuracy and time. Therefore, in the forthcoming experiments, we shall exclusively employ the first type of reprojection error combination for comparison.

How significant is the impact of noise on *RSL-BA*? Following the same synthetic experimental setup as described above, we additionally introduce Gaussian noise to the endpoints of each line in this experiment. The noise levels are varied from 0.1 pixels to 2.0 pixels, and four different types of errors are recorded for the algorithm at each noise level. For each noise level, 50 experiments are conducted, and the median value is used as the final result. The results, presented in Table 1, demonstrate that *RSL-BA* provides stable estimations as the noise level increases from 0.1 pixels to 2 pixels.

Table 2: The median absolute trajectory error (ATE) of different methods on WHU-RSVI [3] dataset. The best and second results are shown in green and blue, respectively.

	WHU-RSVI1	WHU-RSVI2	WHU-RSVI3	WHU-RSVI4
GSBA	0.080992	0.061310	0.030404	0.023698
GLBA	0.076173	0.065985	0.033554	0.024961
NMRSBA	0.050969	0.041629	0.042317	0.028183
NWRSBA	0.040640	0.045313	0.035451	0.022666
RSL-BA(ours)	0.0443502	0.039314	0.023351	0.020675

What are the advantages of *RSL-BA* over point-based methods? Within this part, we will conduct a comparative analysis of our method alongside current SOTA methods, including point- and line-based methods. We set up a cubic box of points in the same positions as the 3D lines, totaling 56 points. To ensure fairness, we set up 8 lines with 7 points each to construct error functions, resulting in a total of precisely 56 points. Detailed experimental settings on the number of lines and the number of sampling points on each line are provided in the supplementary materials. We construct two scenarios: regular and classic scenarios with degeneration parallel to the y-axis. Due to significant differences in the experimental results of different methods, we uniformly take the logarithm (base 10) of the results for better observation. The experimental findings are shown in Fig. 7. It is evident that in both situations, our *RSL-BA* and the current NWRSBA exhibit comparable accuracy, outperforming alternative approaches based on points and lines. Additionally, the computation time is equivalent to that of NWRSBA. This indicates that the application of lines on Rolling Shutter camera holds significant promise.

4.2 Synthetic Images

In this section, we conduct experiments on input synthetic images. We use the WHU-RSVI [3] dataset, from which we select two sets of data from trajectory1-fast and trajectory2-fast for 3D reconstruction and pose estimation. We first employ [22] to detect RS curves by segmenting curves into multiple short-line segments and performing line fitting for initialization. The GS line-based SfM [16] is applied to initialize the *RSL-BA* parameters. The comparative methods include GSBA, NMRSBA, NWRSBA, and GSLBA. Table 2 shows the median absolute trajectory error of different methods, it can be observed that the proposed *RSL-BA* method is the most stable one, achieving optimal or near-optimal results in all cases. Qualitative comparison are also provided in Fig. 8. Unlike point-based methods, line-based methods often achieve good results with fewer feature lines. However, the GSL-BA method is not sufficiently stable when RS effects are prominent.

4.3 Real Images

In this section, we conduct experiments on the real image dataset TUM-RSVI [24]. The experimental setup and comparison methods are similar to those in Sec. 4.2.

Table 3: The absolute trajectory error (ATE) comparison of different methods in TUM-RSVI [24] dataset. The best and second results are shown in green and blue, respectively.

	GSBA	GLBA	NMRSBA	NWRSBA	RSL-BA(ours)
seq1	0.069484	0.086460	0.052366	0.045064	0.037816
seq2	0.029821	0.030379	0.026028	0.023227	0.025132
seq3	0.065160	0.063718	0.057918	0.055001	0.048187
seq4	0.049613	0.052026	0.032214	0.030534	0.031903
seq5	0.031860	0.035839	0.019407	0.016066	0.017659
seq6	0.061966	0.061792	0.032434	0.024658	0.026448
seq7	0.051621	0.056534	0.039154	0.039620	0.039701
seq8	0.026403	0.028690	0.024807	0.025926	0.024983
seq9	0.098334	0.098212	0.082481	0.073580	0.080523
seq10	0.81174	0.81180	0.59390	0.53296	0.57477

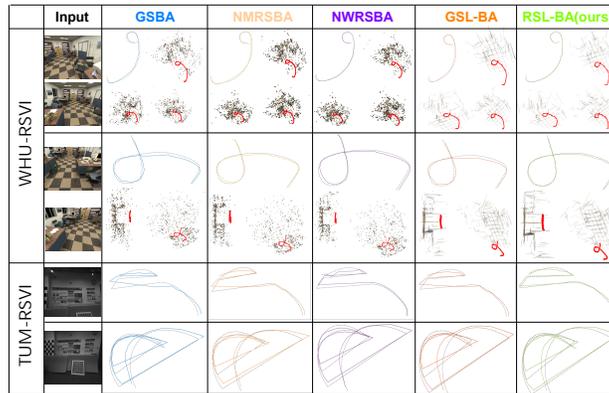


Fig. 8: Comparison of trajectories and 3D reconstruction on WHU-RSVI [3] and TUM-RSVI [24] dataset. Each column represents a different bundle adjustment algorithm, and each row represents a different sequence.

Table 3 and Fig 8 present some of the experimental results. As can be seen, in most cases, the *RSL-BA* method outperforms other methods but is slightly weaker than *NWRSBA*. This is because the *TUM-RSVI* lacks line features and they are not visually prominent, making it less suitable for *RSL-BA*.

5 Conclusion

This paper presents the first solution of line-based *RSL-BA*. By utilizing points and tangent directions on curves, we have established a series of faster and more robust curve-to-line re-projection errors. The proposed *RSL-BA* method can prevent three common degeneracies in *RSBA*, one of which is newly introduced by us. Extensive experiments in real and synthetic data verify the effectiveness and efficiency of the proposed *RSL-BA* method.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2022ZD01190030), Nature Science Foundation of China (No. 62102145), Jiangxi Provincial 03 Special Foundation and 5G Program (Grant No. 20224ABC03A05), Lushan Lab Research Funding, and Changsha Science Fund for Distinguished Young Scholars (kq2306002).

References

1. Albl, C., Sugimoto, A., Pajdla, T.: Degeneracies in rolling shutter sfm. In: ECCV (2016)
2. Bartoli, A., Sturm, P.: Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer vision and image understanding* **100**(3), 416–441 (2005)
3. Cao, L., Ling, J., Xiao, X.: The whu rolling shutter visual-inertial dataset. *IEEE Access* **8**, 50771–50779 (2020)
4. Dai, Y., Li, H., Kneip, L.: Rolling shutter camera relative pose: generalized epipolar geometry. In: CVPR (2016)
5. Duchamp, G., Ait-Aider, O., Royer, E., Lavest, J.M.: A rolling shutter compliant method for localisation and reconstruction. In: VISAPP (2015)
6. Fisher, A., Cannizzaro, R., Cochrane, M., Nagahawatte, C., Palmer, J.L.: Colmap: A memory-efficient occupancy grid mapping framework. *Robotics and Autonomous Systems* **142**, 103755 (2021)
7. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
8. Hedborg, J., Forssen, P.E., Felsberg, M., Ringaby, E.: Rolling shutter bundle adjustment. In: CVPR (2012)
9. Holynski, A., Geraghty, D., Frahm, J.M., Sweeney, C., Szeliski, R.: Reducing drift in structure from motion using extended features. In: 2020 International Conference on 3D Vision (3DV). pp. 51–60. IEEE (2020)
10. Ito, E., Okatani, T.: Self-calibration-based approach to critical motion sequences of rolling-shutter structure from motion. In: CVPR (2016)
11. Lao, Y., Ait-Aider, O.: A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In: CVPR (2018)
12. Lao, Y., Ait-Aider, O., Araujo, H.: Robustified structure from motion with rolling-shutter camera using straightness constraint. *Pattern Recognition Letters* (2018)
13. Lao, Y., Ait-Aider, O., Bartoli, A.: Solving rolling shutter 3d vision problems using analogies with non-rigidity. *IJCV* (2021)
14. Liao, B., Qu, D., Xue, Y., Zhang, H., Lao, Y.: Revisiting rolling shutter bundle adjustment: Toward accurate and fast solution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4863–4871 (2023)
15. Lim, H., Jeon, J., Myung, H.: Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping. *IEEE Robotics and Automation Letters* **7**(2), 1518–1525 (2022)
16. Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., Larsson, V.: 3d line mapping revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21445–21455 (2023)

17. Lourakis, M.I., Argyros, A.A.: Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)* **36**(1), 1–30 (2009)
18. Micusik, B., Wildenauer, H.: Structure from motion with line segments under relaxed endpoint constraints. *International Journal of Computer Vision* **124**, 65–79 (2017)
19. Moré, J.J.: The levenberg-marquardt algorithm: implementation and theory. In: *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. pp. 105–116. Springer (2006)
20. Patron-Perez, A., Lovegrove, S., Sibley, G.: A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *IJCV* (2015)
21. Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F.: Pl-slam: Real-time monocular visual slam with points and lines. In: *2017 IEEE international conference on robotics and automation (ICRA)*. pp. 4503–4508. IEEE (2017)
22. Purkait, P., Zach, C., Leonardis, A.: Rolling shutter correction in manhattan world. In: *ICCV*. pp. 882–890 (2017)
23. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4104–4113 (2016)
24. Schubert, D., Demmel, N., von Stumberg, L., Usenko, V., Cremers, D.: Rolling-shutter modelling for direct visual-inertial odometry. In: *IROS* (2019)
25. Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(11), 1021–1032 (1995)
26. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. pp. 298–372. Springer (2000)
27. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence* **32**(4), 722–732 (2008)
28. Wei, D., Wan, Y., Zhang, Y., Liu, X., Zhang, B., Wang, X.: Elsr: Efficient line segment reconstruction with planes and points guidance. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 15807–15815 (2022)
29. Zhuang, B., Cheong, L.F., Lee, G.H.: Rolling-shutter-aware differential sfm and image rectification. In: *ICCV* (2017)
30. Zhuang, B., Tran, Q.H., Ji, P., Cheong, L.F., Chandraker, M.: Learning structure-and-motion-aware rolling shutter correction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4551–4560 (2019)
31. Zuo, X., Xie, X., Liu, Y., Huang, G.: Robust visual slam with point and line features. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1775–1782. IEEE (2017)