ParallaxBA: Bundle Adjustment using Parallax Angle Feature Parametrization

Liang Zhao[*], Shoudong Huang[†], Yanbiao Sun[‡], Lei Yan[‡], Gamini Dissanayake[†]

Abstract

The main contribution of this paper is a novel feature parametrization based on parallax angles for bundle adjustment (BA) in structure and motion estimation from monocular images. It is demonstrated that under certain conditions, describing feature locations using their Euclidean XYZ coordinates or using inverse depth in BA leads to ill-conditioned normal equations as well as objective functions that have very small gradients with respect to some of the parameters describing feature locations. The proposed parallax angle feature parametrization in BA (ParallaxBA), avoids both of the above problems leading to better convergence properties and more accurate motion and structure estimates. Simulation and experimental datasets are used to demonstrate the impact of different feature parametrizations on BA, and the improved convergence, efficiency and accuracy of the proposed ParallaxBA algorithm when compared with some existing BA packages such as SBA, sSBA and g2o. The C/C++ source code of ParallaxBA is available on OpenSLAM (https://openslam.org/).

Key words: Bundle adjustment, feature parametrization, parallax angle, objective function, singularity.

1 Introduction

Structure from motion or monocular simultaneous localization and mapping (SLAM) is the problem in estimating the camera poses and feature positions

[*] L. Zhao is with the Hamlyn Centre for Robotic Surgery, Department of Computing, Faculty of Engineering, Imperial College London, London SW7 2AZ, United Kingdom. liang.zhao@imperial.ac.uk.

[†] S. Huang and G. Dissanayake are with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia. {Shoudong.Huang,Gamini.Dissanayake}@uts.edu.au.

[‡] Y. Sun and L. Yan are with the Institute of Remote Sensing and GIS, School of Earth and Space Science, Peking University, Beijing China, 100871. {syb51,lyan}@pku.edu.cn.

This work is done when the first author (L. Zhao) was working as a research fellow and the third author (Y. Sun) was a visiting scholar in the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney.

A preliminary version of this paper has been published in proceedings of the 2011 IEEE International Conference on Robotics and Automation (Zhao et al. 2011).

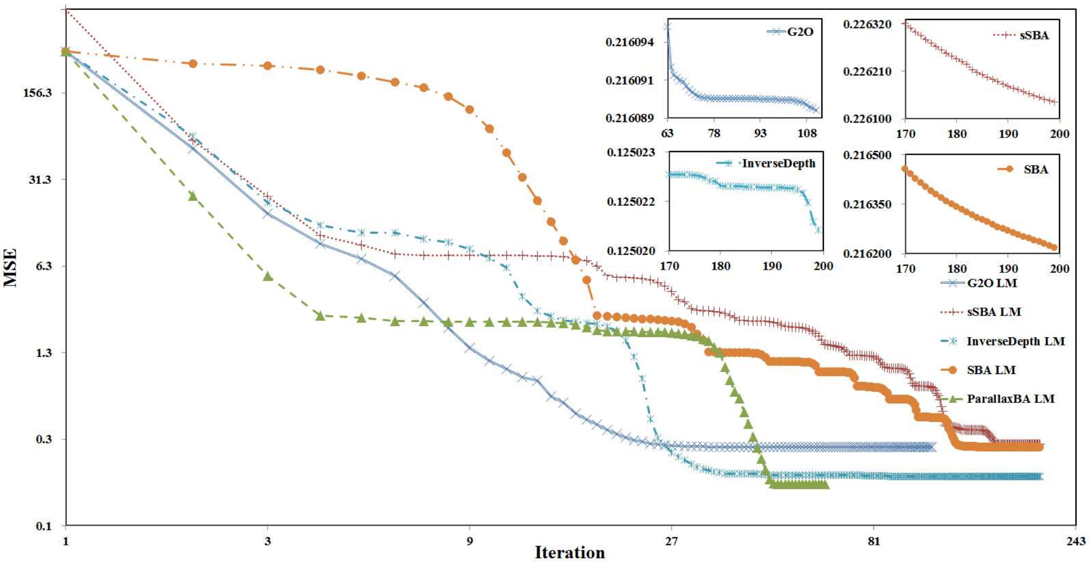


Figure 1: Mean Square Error (MSE) of the reprojections for BA using Malaga dataset. ParallaxBA converges to 0.109209 in 61 iterations; g2o converges to 0.216089 in 111 iterations; BA with inverse depth parametrization, SBA, and sSBA stop because the maximum number of iterations (200) is reached (the MSE value will change very little if the number of iterations is increased further).

using a sequence of images. Bundle adjustment (BA) is regarded as the gold standard for estimating the feature positions and camera poses once the feature correspondences among the images is established.

BA has been a subject of research over a long period of time (Triggs et al. 1999) and has been extensively used in many applications. Currently a number of BA packages are publicly available. For example, the SBA package (Lourakis and Argyros 2009) is very popular among both robotics and computer vision communities. Recently, more efficient implementations BA, sSBA (Konolige 2010) and g2o (Kummerle et al. 2011) have also become available.

All these three algorithms use Euclidean XYZ coordinates for representing the locations of three-dimensional point features. The authors' experiments using BA with XYZ feature parametrization indicated that, when far away features exist or all the camera poses observing a feature are aligned with the feature, the normal equations matrix becomes ill-conditioned, and thus the Gauss-Newton (GN) method will report a singularity and diverge, making it necessary to use the Levenberg-Marquardt (LM) method for optimization. However, LM often requires an extremely large number of iterations. Furthermore, the reduction in the objective function becomes very small after some iterations, while at the same time the changes to state vector remain significant. This makes it very difficult to decide whether the optimal solution has been reached and to determine the appropriate time to terminate the algorithm.

Figure 1 shows an example of how different BA algorithms perform when used with an experimental dataset (Malaga Parking-6L dataset (Blanco et al. 2009)). For this dataset, the final mean square error (MSE) of the reprojections using the ParallaxBA algorithm proposed in this paper is 0.109209. For the popular BA packages SBA, sSBA, and g2o, the MSE of the reprojections quickly reach around 0.22, but the rate of change becomes very small thereafter (Figure 1). For g2o, the MSE does not change significantly after about 30 iterations and the algorithm terminates at iteration 111 with a MSE of 0.216089. For SBA and sSBA, the changes in MSEs are not significant after around 160 iterations and the algorithms were terminated when the maximum number of iterations (200) is reached with the final MSE 0.216219 (SBA) and 0.226131 (sSBA).

In the inverse depth feature parametrization proposed in (Civera et al. 2008), feature locations are described using the reciprocal of the distance between the camera center and the feature, the azimuth and elevation angles of the feature with respect to the camera center, and the position of the camera center in a global coordinate frame typically known as the "anchor". Inverse depth feature parametrization has been shown to perform better than XYZ feature parametrization with a better chance to provide consistent estimates when used in the extended Kalman filter (EKF) framework. The study by the authors indicated that when inverse depth parametrization is used in BA, although far away features will not cause any problem, features that are aligned with the direction of camera motion can cause the normal equations to be ill-conditioned. Under this condition, the GN algorithm diverges and the objective function reduces very slowly in the LM algorithm, even though the changes to the state vector remain large. As seen in Figure 1, the MSE of BA with inverse depth parametrization reaches 0.13 after about 40 iterations but only reduces to 0.125020 when the maximum number of iterations 200 is reached.

There is no easy mechanism to detect and discard these problematic features before performing BA. Furthermore, as the state vector keeps changing during the iterations in BA, even a single feature that satisfies the singularity condition in one particular iteration can have significant impact on the convergence of the algorithm.

This paper proposes ParallaxBA, where a new parallax angle feature parametrization strategy is used in BA. The new feature parameters include three angles – the azimuth and elevation angles of the feature with respect to one camera center, and the parallax angle of the feature from two camera centers. Both the two camera centers are used as anchors in this representation. It is demonstrated that the new feature parametrization results in better convergence when compared with BA using the Euclidean XYZ and inverse depth parametrizations. It can be seen from Figure 1 that ParallaxBA can converge to MSE 0.109209 with only 61 iterations which is significantly smaller than those obtained using existing BA packages (the iteration naturally stopped because the change in state vector is smaller than the threshold).

This paper builds on the previous preliminary work (Zhao et al. 2011), with the following major improvements: (1) an analysis of the ill-conditioning and singularity of the normal equations matrix; (2) an analysis of the structure of the objective functions; (3) an optimized C/C++ implementation of ParallaxBA which is faster than SBA and comparable in speed to that of sSBA and g2o; (4) evaluations using more large-scale datasets; and (5) the availability of the

C/C++ source code on OpenSLAM.

This paper is organized as follows. Section 2 discusses the related work. Section 3 describes the parallax angle parametrization for features. Section 4 provides the details of the ParallaxBA algorithm. Section 5 analyzes the impact of the different parametrizations on the singularity of the normal equations, the structure of the objective function, and the accuracy of the initial feature location estimates. In Section 6, simulations and experimental results are provided to demonstrate the impact of feature parametrizations on BA, as well as the improved convergence and accuracy of ParallaxBA. Finally Section 7 concludes the paper and Appendix A presents the derivation of the observation function used in Section 4.

2 Related Work

In monocular SLAM (Davison 2003), observations from a camera that provides relative bearing and elevation to features in the environment are used to build up the representation of the environment as well as provide an estimate of the camera poses. The most obvious way to represent the location of features in a 3D environment is to use their Euclidean coordinates X, Y and Z. As a direct measurement of feature depth is not available, a good initial estimate of X, Y and Z cannot be obtained without observing a feature from two locations with sufficient large parallax. Representing feature locations using their Euclidean XYZ coordinates has been demonstrated to be a poor choice, particularly when the parallax angle used to initialize the feature location is small (Civera et al. 2008). Parallax angle becomes small in two specific scenarios: when a feature is at a long distance from the camera or when the camera is moving towards or away from a feature along the line from the camera to the feature, regardless of the distance. The many strategies that have been proposed to deal with the issue of initializing feature locations, either add significant complexity or result in a delay in incorporating the information from observed features. For example, Kwok and Dissanayake (Kwok and Dissanayake 2004) and Sola et al. (Sola et al. 2005) proposed to use a mixture of Gaussians to represent multiple hypotheses for the depth of the feature so that the feature location can be initialized as soon as a feature is observed, while Davison et al. (Davison et al. 2007) used a ray to represent the feature when it is observed first, and the feature is initialized as a point after multiple observations when the probability distribution for the feature location can be safely approximated as Gaussian.

For a simpler 2D bearing-only SLAM problem, a framework for dealing with both nearby and distant landmarks is provided in (Trawny and Roumeliotis 2006), where a global bearing angle from an artificial position is used as one of the feature parameters. This strategy leads to a redundant representation when extended to 3D. The inverse depth feature parametrization proposed in (Civera et al. 2008) provides a more robust and accurate representation for far away features, using the reciprocal of the distance between the anchored camera center and the feature. A state vector containing the feature locations and relevant camera centers is then estimated in an estimator such as an extended Kalman filter (EKF) (Civera et al. 2008). A similar representation using homogeneous coordinates is also shown to perform well in optimization frameworks such as BA (Triggs et al. 1999) when far away features are present. Inverse depth method overcomes many of the issues associated with feature initialization and has now become the most accepted way to describe feature locations in monocular SLAM. Sola (Sola 2010) compared different feature parametrization approaches. In an estimator framework based on the EKF, feature parametrizations that incorporate a camera center or "anchor" are shown to be superior to those that do not use anchors. The ParallaxBA proposed in this paper uses the parallax angle feature parametrization which has two "anchors" for one 3D point feature.

Recently, nonlinear optimization based SLAM has become more popular than those based on EKF due to the availability of increased computer power and the recognition that the Jacobians and information matrices involved in the optimization algorithms are sparse, making optimization process computationally efficient. Bundle adjustment, used extensively in computer vision literature (Triggs et al. 1999), is the gold standard for monocular SLAM as it solves the global optimization problem involving all the observations. As such it avoids many of the issues associated with EKF based algorithms (Huang and Dissanayake 2007) (Strasdat et al. 2010) and is able to produce more accurate and consistent estimates. A systematic and comprehensive comparison between BA and EKF is given in (Strasdat et al. 2010) where it was shown that BA outperforms EKF SLAM in terms of consistency and accuracy. SLAM based on BA can now be performed in real-time for example by using relative bundle adjustment (Blanco et al. 2013) or an incremental light bundle adjustment (Indelman et al. 2013).

A number of software packages of BA are now available, for example SBA (Lourakis and Argyros 2009), and a more efficient implementation sSBA (Konolige 2010). A general graph optimization package, g2o (Kummerle et al. 2011), also has a BA component. Closer examination of the behavior of existing BA packages illustrates some interesting issues. The normal equations associated with the optimization problem can easily become ill-conditioned, requiring an algorithm with a damping factor such as Levenberg-Marquardt (LM). Moreover, the objective function may consist of "long flat valleys" due to the lack of adequate depth information for some features. This can result in gradient based algorithms such as LM converging very slowly or failing to find the global optimum. Although several modern methods are proposed to improve the convergence and efficiency of BA (Jeong et al. 2012), the fundamental issues in the normal equations and objective functions have not yet been addressed. The authors have observed that when XYZ parametrization is used, the relevant Jacobians become rank deficient when features that are always far away from the camera are present, indicating that there is insufficient information to compute the distance to these features. A similar situation occurs in the case of inverse depth representation if the camera velocity vector passed through a feature. While this seems to be an extreme scenario, given datasets that contain hundreds of frames and tens of thousands of features are now common, at least one feature that satisfies this condition is always likely to be present, as will be seen in Section 6.

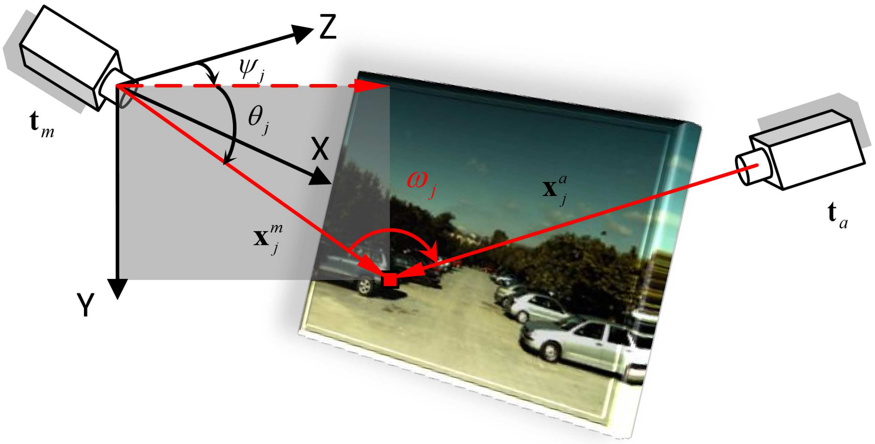


Figure 2: Parallax angle parametrization for feature.

This paper proposes a new feature parametrization strategy where the parallax angle from two camera poses is used as a parameter to describe the feature location. It is demonstrated that the used parallax angle feature parametrization in BA results in an objective function that has a better structure thereby avoiding the issues mentioned in the previous paragraph and resulting in improved convergence.

3 Parallax Angle Feature Parametrization

In this section, the parallax angle feature parametrization for monocular SLAM is presented. The key idea is to use the azimuth angle, elevation angle, and parallax angle, together with the two anchored camera centers to represent the location of a 3D point feature.

3.1 Camera Pose Parametrization

A camera pose is represented by rotation angles and a translation vector relative to the first camera pose, $\mathbf{p}_0$. The i^{th} camera pose is denoted as

$$\mathbf{p}_i = [\alpha_i \quad \beta_i \quad \gamma_i \quad x_i \quad y_i \quad z_i]^T \tag{1}$$

where $\mathbf{r}_i = [\alpha_i \quad \beta_i \quad \gamma_i]^T$ contains the Yaw, Pitch, Roll angles of $\mathbf{p}_i$ and $\mathbf{t}_i = [x_i \quad y_i \quad z_i]^T$ is the translation vector from $\mathbf{p}_0$ to $\mathbf{p}_i$, $\mathbf{p}_0 = [0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0]^T$.

3.2 Parallax Angle Parametrization for Features

If a feature $\mathbf{F}_j$ is only observed once, then the camera center from which it is observed is defined as the main anchor of $\mathbf{F}_j$ and is denoted as $\mathbf{t}_m$. The feature is described by

$$\mathbf{F}_j = [\psi_j \quad \theta_j]^T \tag{2}$$

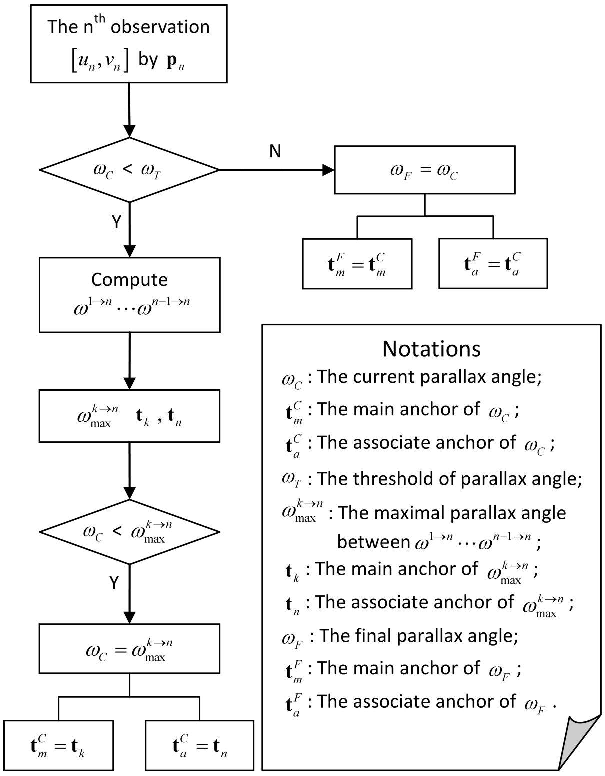

The n^th observation
$[u_n, v_n]$ by $\mathbf{p}_n$
$\omega_C < \omega_T$
N
Y
$\omega_F = \omega_C$
$\mathbf{t}_m^F = \mathbf{t}_m^C$
$\mathbf{t}_a^F = \mathbf{t}_a^C$
Compute
$\omega^{1\to n} \cdots \omega^{n-1\to n}$
$\omega_{\max}^{k\to n}$ $\mathbf{t}_k$, $\mathbf{t}_n$
$\omega_C < \omega_{\max}^{k\to n}$
Y
$\omega_C = \omega_{\max}^{k\to n}$
$\mathbf{t}_m^C = \mathbf{t}_k$
$\mathbf{t}_a^C = \mathbf{t}_n$
Notations
ω_C : The current parallax angle;
$\mathbf{t}_m^C$: The main anchor of ω_C ;
$\mathbf{t}_a^C$: The associate anchor of ω_C ;
ω_T : The threshold of parallax angle;
$\omega_{\max}^{k\to n}$: The maximal parallax angle between $\omega^{1\to n} \cdots \omega^{n-1\to n}$;
$\mathbf{t}_k$: The main anchor of $\omega_{\max}^{k\to n}$;
$\mathbf{t}_n$: The associate anchor of $\omega_{\max}^{k\to n}$;
ω_F : The final parallax angle;
$\mathbf{t}_m^F$: The main anchor of ω_F ;
$\mathbf{t}_a^F$: The associate anchor of ω_F .

Figure 3: Flowchart of choosing the anchors.

where ψ_j and θ_j are the azimuth and elevation angles, so the parameters $[\psi_j \quad \theta_j]^T$ represent the direction from the main anchor $\mathbf{t}_m$ to $\mathbf{F}_j$ in the coordinate frame defined by $\mathbf{p}_0$ (Figure 2).

If the feature $\mathbf{F}_j$ is observed twice or more, then two camera centers will be chosen as its anchors. One is called the main anchor which is denoted as $\mathbf{t}_m$ and the other is called the associate anchor and is denoted as $\mathbf{t}_a$. The feature $\mathbf{F}_j$ is described using

$$\mathbf{F}_j = [\psi_j \quad \theta_j \quad \omega_j]^T, \tag{3}$$

where the parallax angle ω_j is the angle from $\mathbf{x}_j^m$ to $\mathbf{x}_j^a$ with $\mathbf{x}_j^m$ and $\mathbf{x}_j^a$ being the vectors from the main anchor $\mathbf{t}_m$ to $\mathbf{F}_j$ and the associate anchor $\mathbf{t}_a$ to $\mathbf{F}_j$, respectively (Figure 2).

3.3 A Strategy for Choosing the Anchors

When the $\mathbf{F}_j$ is observed twice, the main anchor is defined as the camera center where $\mathbf{F}_j$ is first observed, and the associate anchor is defined as the camera center where $\mathbf{F}_j$ is observed the second time. If the feature has been observed more than twice, the strategy for selecting the anchors is given in the flowchart in Figure 3. That is, the anchors selected will be the two camera centers with either the maximum parallax angle or a parallax angle larger than a threshold. The value of the threshold is found not to be very critical. In the experiments performed in this paper and the code published on OpenSLAM, the threshold is set as 0.5 rad.

In monocular vision, the feature depth information is implied in the parallax between observations from different viewpoints. By directly using parallax angle instead of parameters with depth information, the proposed parallax angle parametrization is closer to the measurement space of monocular sensors, and thus it can be expected to be more accurate than other feature parametrizations. Moreover, the three angles in (3) only define the relative structure of the feature with respect to the two anchors. As such, the scale of the feature is implicitly defined by the relative translation of the two anchors. This is completely different from the XYZ parametrization (in which the scale of the feature is involved in all the three parameters) and the inverse depth parametrization (in which one parameter, the inverse of the depth, contains the scale information of the feature).

4 Bundle Adjustment using Parallax Angle Feature Parametrization

While the presence of many "anchors" can increase the size of the state vector in a typical EKF based estimator, the new parametrization does not increase the size of the optimization problem in a BA based solution as all the camera poses and all the features are present in the state vector.

In this section, the observation function for BA using the new feature parametrization is first presented. Then the least squares optimization formulation for BA, and the procedures for the initialization of poses and features are briefly outlined. For clarity, some detailed derivations are presented in Appendix A.

4.1 Observation Function for BA

The information available for BA is the image coordinates of each feature in each image. This information needs to be described as a function of the camera poses and the feature parameters.

Suppose the main and associate anchors of a feature $\mathbf{F}_j$ are $\mathbf{t}_m$ and $\mathbf{t}_a$ respectively. The image coordinates of $\mathbf{F}_j$ at $\mathbf{p}_i$ can be written as

$$\begin{bmatrix} u_j^i \\ v_j^i \end{bmatrix} = \begin{bmatrix} x_j^i \,/\, t_j^i \\ y_j^i \,/\, t_j^i \end{bmatrix} \tag{4}$$

where

$$\begin{bmatrix} x_j^i \\ y_j^i \\ t_j^i \end{bmatrix} = \begin{cases} K \; R_m \; \mathbf{x}_j^m, & \text{if } \; i = m \\ K \; R_i \;\; \tilde{\mathbf{x}}_j^i \; , & \text{if } \; i \neq m. \end{cases} \tag{5}$$

Here K is the calibration matrix, R_i is the rotation matrix of $\mathbf{p}_i$, which is a function of $[\alpha_i \;\; \beta_i \;\; \gamma_i]^T$,

$$R_i = \mathrm{r}\,(\alpha_i,\, \beta_i,\, \gamma_i). \tag{6}$$

$\mathbf{x}_j^m$ is the unit vector from the main anchor $\mathbf{t}_m$ to feature $\mathbf{F}_j$ given by

$$\mathbf{x}_j^m = \mathrm{v}\,(\psi_j,\, \theta_j) = \begin{bmatrix} \sin\psi_j \cos\theta_j \\ \sin\theta_j \\ \cos\psi_j \cos\theta_j \end{bmatrix}. \tag{7}$$

$\tilde{\mathbf{x}}_j^i$ $(i \neq m)$ represents the scaled vector from $\mathbf{t}_i$ to $\mathbf{F}_j$ which can be computed as

$$\tilde{\mathbf{x}}_j^i = \sin(\omega_j + \varphi_j)\, \|\mathbf{t}_a - \mathbf{t}_m\|\, \mathbf{x}_j^m - \sin\omega_j\, (\mathbf{t}_i - \mathbf{t}_m) \tag{8}$$

where φ_j is the angle from the vector $\mathbf{t}_a - \mathbf{t}_m$ to the vector $\mathbf{x}_j^m$. The angle φ_j can be computed from the dot product of the two vectors $\mathbf{t}_a - \mathbf{t}_m$ and $\mathbf{x}_j^m$ as

$$\varphi_j = \arccos\left(\mathbf{x}_j^m \cdot \frac{\mathbf{t}_a - \mathbf{t}_m}{\|\mathbf{t}_a - \mathbf{t}_m\|}\right). \tag{9}$$

The detailed derivation of (8) is given in Appendix A.

4.2 Least Squares Optimization

Let $\mathbf{Z}$ be the measurement vector containing image coordinates of all the features in each image, let $\mathbf{X}$ be the state vector containing all the poses and features using the proposed parametrization, and let $f(\mathbf{X})$ be the observation function defined by combining (4) to (9) in Section 4.1. The least squares optimization problem in BA is to seek the vector $\hat{\mathbf{X}}$ such that

$$\|\varepsilon\|^2_{\Sigma_Z^{-1}} = (f(\hat{\mathbf{X}}) - \mathbf{Z})^T \Sigma_Z^{-1} (f(\hat{\mathbf{X}}) - \mathbf{Z}) \tag{10}$$

is minimized.

The uncertainty of the image coordinates of all the features are assumed to be identical, which is standard in BA. Then Σ_Z^{-1} is an identity matrix in the following.

4.3 Gauss-Newton Iteration and Levenberg-Marquardt Method

A solution $\hat{\mathbf{X}}$ of (10) can be obtained by starting with an initial estimate $\mathbf{X}_0$ and iterating with $\mathbf{X}_{k+1} = \mathbf{X}_k + \Delta_k$, where the vector Δ_k is the solution to

$$J^T J \Delta_k = -J^T (f(\mathbf{X}_k) - \mathbf{Z}) \tag{11}$$

where J is the linear mapping represented by the Jacobian matrix $\partial f / \partial \mathbf{X}$ evaluated at $\mathbf{X}_k$.

Equation (11) is called normal equations and is used in the Gauss-Newton (GN) iteration (Hartley and Zisserman 2003).

The Levenberg-Marquardt (LM) method is a damped GN method. The step Δ_k is computed by the following modification to (11),

$$(J^T J + \lambda E)\Delta_k = -J^T(f(\mathbf{X}_k) - \mathbf{Z}). \tag{12}$$

where E is the identity matrix. The damping parameter $\lambda > 0$ has several effects: a) For all $\lambda > 0$ the coefficient matrix is positive definite, and thus nonsingular. b) For large values of $\lambda > 0$, (12) results in a short step in the steepest descent direction. This is useful if the current iteration is far from the solution. c) If $\lambda > 0$ is very small, then it is close to GN.

4.4 State Vector Initialization

The i^{th} camera pose, $\mathbf{p}_i = [\alpha_i \quad \beta_i \quad \gamma_i \quad x_i \quad y_i \quad z_i]^T$, can be initialized as

$$\begin{cases} \mathrm{r}\left(\alpha_i,\, \beta_i,\, \gamma_i\right) = \prod_{n=1}^{i} \Delta R_{n-1}^{n} \\ [x_i \;\; y_i \;\; z_i]^T = \sum_{n=1}^{i} R_n^T \Delta \mathbf{t}_{n-1}^{n} \end{cases} \tag{13}$$

where ΔR_{n-1}^{n} is the relative rotation matrix and $\Delta \mathbf{t}_{n-1}^{n} = [\Delta x_{n-1}^{n} \quad \Delta y_{n-1}^{n} \quad \Delta z_{n-1}^{n}]^T$ is the relative translation between camera poses $\mathbf{p}_{n-1}$ and $\mathbf{p}_n$ computed using two-view geometry (Hartley and Zisserman 2003). For initialization, the relative scales between relative translations are assumed to be 1.

If feature $\mathbf{F}_j$ is observed only once at $\mathbf{p}_m$, then $\mathbf{t}_m$ is its main anchor and $\mathbf{F}_j$ can be initialized as

$$\begin{cases} \psi_j = \mathrm{atan2}\left(\hat{x}_j^m,\, \hat{z}_j^m\right) \\ \theta_j = \mathrm{atan2}\left(\hat{y}_j^m,\, \sqrt{(\hat{x}_j^m)^2 + (\hat{z}_j^m)^2}\right) \end{cases} \tag{14}$$

where

$$\begin{bmatrix} \hat{x}_j^m \\ \hat{y}_j^m \\ \hat{z}_j^m \end{bmatrix} = (K R_m)^{-1} \begin{bmatrix} u_j^m \\ v_j^m \\ 1 \end{bmatrix} \tag{15}$$

and R_m is the rotation matrix of $\mathbf{p}_m$, $[u_j^m \quad v_j^m]^T$ is the image coordinates of feature $\mathbf{F}_j$ at $\mathbf{p}_m$ and $\hat{\mathbf{x}}_j^m = [\hat{x}_j^m \;\; \hat{y}_j^m \;\; \hat{z}_j^m]^T$ is the vector from $\mathbf{t}_m$ to feature $\mathbf{F}_j$.

If $\mathbf{F}_j$ is observed at least twice with main anchor $\mathbf{t}_m$ and associate anchor $\mathbf{t}_a$, then the parallax angle ω_j can be initialized as

$$\omega_j = \arccos\left(\frac{\hat{\mathbf{x}}_j^m \cdot \hat{\mathbf{x}}_j^a}{\|\hat{\mathbf{x}}_j^m\| \|\hat{\mathbf{x}}_j^a\|}\right) \tag{16}$$

where $\hat{\mathbf{x}}_j^a$ is the vector from $\mathbf{t}_a$ to feature $\mathbf{F}_j$ computed similarly to $\hat{\mathbf{x}}_j^m$ by using (15).

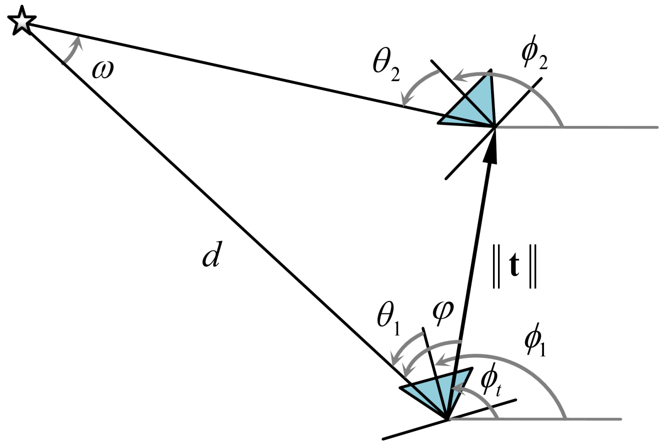


Figure 4: A scenario of 2D bearing only case.

5 Analysis of the Impact of Feature Parametrizations on BA

In this section, the impact of feature parametrizations on the BA problem will be analyzed. In particular, the singularity of the information matrix, the structure of the objective function, and the sensitivity of the feature location representations to the observation noises for different feature parametrizations will be studied. For simplicity, the analysis is performed for the 2D case with a simple scenario shown in Figure 4 and the camera poses are assumed fixed. Consider a 2D bearing only problem where a feature is observed from two camera poses. The orientations of the first and second camera poses are denoted as ϕ_1 and ϕ_2 respectively, and the bearing angles are denoted as θ_1 and θ_2. The orientation and length of the relative translation between the two camera poses are denoted as ϕ_t and $\|\mathbf{t}\|$, respectively (Figure 4).

5.1 The Singularity of Information Matrix $J^T J$

The normal equations (11) plays a key role in both GN and LM algorithms. Its coefficient matrix $J^T J$ is an approximation to the Hessian matrix. Once the algorithm converges, $J^T J$ can also be used as the information matrix (the inverse of the covariance matrix) of the final estimate. Now it is called the "information matrix" and the situations under which this matrix can be singular or ill-conditioned will be analyzed for different feature parametrizations.

5.1.1 XYZ Parametrization

Since depth plays an important role in XYZ parametrization, in the following analysis, it is assumed that the depth d and the global bearing angle θ are used to represent the feature. Assume all the other parameters are fixed except for θ and d. The problem is to find θ and d that minimize the sum of the observation error squared

$$G(\theta, d) = \|\mathbf{f}(\theta, d) - \mathbf{z}\|^2 = (f_1(\theta, d) - \theta_1)^2 + (f_2(\theta, d) - \theta_2)^2 \tag{17}$$

where

$$f_1(\theta, d) = \theta - \phi_1, \quad f_2(\theta, d) = \operatorname{atan2}(d \sin\theta - \|\mathbf{t}\| \sin\phi_t,\ d\cos\theta - \|\mathbf{t}\| \cos\phi_t) - \phi_2. \tag{18}$$

The errors in the observations θ_1 and θ_2 are assumed to be independent, the variances of the observation errors are assumed to be 1.

The Jacobian of $\mathbf{f}(\theta, d)$ is

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial d} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial d} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial d} \end{bmatrix}. \tag{19}$$

The information matrix is

$$I = J^T J = \begin{bmatrix} 1 + (\frac{\partial f_2}{\partial \theta})^2 & \frac{\partial f_2}{\partial \theta}\frac{\partial f_2}{\partial d} \\ \frac{\partial f_2}{\partial \theta}\frac{\partial f_2}{\partial d} & (\frac{\partial f_2}{\partial d})^2 \end{bmatrix}. \tag{20}$$

The determinant of the information matrix I is $\det(I) = (\frac{\partial f_2}{\partial d})^2$. It is clear that I is positive definite unless $\frac{\partial f_2}{\partial d} = 0$.

Note that

$$\frac{\partial f_2}{\partial d} = \frac{-\|\mathbf{t}\| \sin\varphi}{d^2 + \|\mathbf{t}\|^2 - 2d\|\mathbf{t}\| \cos\varphi} \tag{21}$$

where $\varphi = \theta - \phi_t$ is the angle from the relative translation to the first feature observation (see Figure 4).

Thus, the information matrix is singular if (i) $\varphi = 0$, and it will have a very large condition number if φ is close to 0. Also, the information matrix will have a very large condition number if (ii) $d \to \infty$. When the camera motion is in the direction of the feature, condition (i) is satisfied, while condition (ii) occurs when a feature is at a long distance from the camera.

5.1.2 Inverse Depth Parametrization

When inverse depth parametrization is used, the 2D feature is represented by the global bearing θ and the inverse of the depth $\rho = 1/d$. Assume all the other parameters are fixed except θ and ρ. In this case, the problem is to find the θ and ρ that minimize

$$G(\theta, \rho) = \|\mathbf{f}(\theta, \rho) - \mathbf{z}\|^2 = (f_1(\theta, \rho) - \theta_1)^2 + (f_2(\theta, \rho) - \theta_2)^2 \tag{22}$$

where

$$f_1(\theta, \rho) = \theta - \phi_1, \quad f_2(\theta, \rho) = \operatorname{atan2}(\sin\theta - \rho\|\mathbf{t}\| \sin\phi_t,\ \cos\theta - \rho\|\mathbf{t}\| \cos\phi_t) - \phi_2. \tag{23}$$

In this case, the information matrix I is positive definite unless $\frac{\partial f_2}{\partial \rho} = 0$.

Note that

$$\frac{\partial f_2}{\partial \rho} = \frac{\|\mathbf{t}\| \sin \varphi}{1 + \rho^2 \|\mathbf{t}\|^2 - 2\rho \|\mathbf{t}\| \cos \varphi}. \tag{24}$$

Thus, the information matrix is singular if (i) $\varphi = 0$, and it will have a very large condition number if φ is close to 0. Also the information matrix will have a large condition number if (ii) $\rho \to \infty$. As before when the camera motion is aligned with the direction of the feature, condition (i) is satisfied. Condition (ii) occurs only when the feature is very close to the camera, which is impossible in practice. Notice that if the feature is far away, $\rho \to 0$ and this will not cause $\frac{\partial f_2}{\partial \rho} \to 0$ unless $\varphi \to 0$.

5.1.3 Parallax Angle Parametrization

For parallax angle parametrization, the feature is represented by the global bearing θ and the parallax angle ω. By considering the feature parameters as only variables as before, the problem now is to find the θ and ω that minimize

$$G(\theta, \omega) = \|\mathbf{f}(\theta, \omega) - \mathbf{z}\|^2 = (f_1(\theta, \omega) - \theta_1)^2 + (f_2(\theta, \omega) - \theta_2)^2 \tag{25}$$

where

$$f_1(\theta, \omega) = \theta - \phi_1, \quad f_2(\theta, \omega) = \theta + \omega - \phi_2. \tag{26}$$

Note that

$$\frac{\partial f_2}{\partial \theta} = 1, \quad \frac{\partial f_2}{\partial \omega} = 1. \tag{27}$$

Thus, the information matrix is

$$I = J^T J = \begin{bmatrix} 1 & \frac{\partial f_2}{\partial \theta} \\ 0 & \frac{\partial f_2}{\partial \omega} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \omega} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \tag{28}$$

which is always positive definite.

The reason why the singularity of the information matrix can be avoided using parallax angle feature parametrization is that when the camera motion is aligned with the direction to the feature, although no information about the depth of the feature is available, the parallax angle parameter that needs to be estimated, is known to be zero.

In Section 6, it will be shown using 3D simulation and real experiments that for both XYZ parametrization and inverse depth parametrization the information matrices can become singular in practice, confirming the two conditions: (i) the camera motion is aligned with the direction to the feature, and (ii) the feature is at a long distance. The large condition number of the information matrix can cause some convergence issues for both GN and LM. It will also be shown that using parallax angle parametrization helps avoid the singularity issue caused by features in actual 3D BA. It should be noted that in 3D BA with parallax angle feature parametrization, the information matrix is no longer a constant since the observation functions are more complicated than 2D and the poses are also considered as variables.

5.2 Analysis on the Objective Function in BA

In this section, the structure of the objective function in BA when different feature parametrizations are used will be analyzed. As before, the analysis is conducted using the simple scenario as described in Figure 4.

Consider the objective function when all the variables are fixed at the ground truth except for one variable, the depth, the inverse depth or the parallax angle for each feature parametrization. Examples of the 1D objective functions are shown in Figures 5(a) to 5(c). For the objective functions drawn in the figures, the scenario described in Figure 4 is considered and the first camera pose is assumed to be the origin. It is also assumed that the camera moves straight forward $1m$ with no rotation ($\phi_1 = 0$, $\phi_2 = 0$, $\phi_t = 0$ and $\|\mathbf{t}\| = 1m$ in Figure 4). Also, the bearing angles of the feature from the two camera poses are assumed to be perfectly observed ($\theta_1 = 0.2450rad$, $\theta_2 = 0.3218rad$). The ground truth of the poses and feature are $\mathbf{P}_1 = (0,\ 0,\ 0)$, $\mathbf{P}_2 = (1,\ 0,\ 0)$, and $\mathbf{F} = (4,\ 1)$ in the global coordinates. Furthermore, $\varphi = 0.2450rad$, $\omega = 0.0768rad$, $d = 4.1231m$, $\rho = 0.2425/m$. Since the bearing observations have no error, the global minimum is the ground truth and the minimal objective function is zero.

5.2.1 XYZ Parametrization

Suppose the feature is parametrized as $\mathbf{F} = [\theta,\ d]^T$, where θ is the global bearing angle of $\mathbf{F}$ from the first camera and d is the depth. When θ is fixed, the objective function can be written as

$$G(d) = (\theta - \phi_1 - \theta_1)^2 + (\operatorname{atan2}(d\sin\theta - \|\mathbf{t}\|\sin\phi_t,\ d\cos\theta - \|\mathbf{t}\|\cos\phi_t) - \phi_2 - \theta_2)^2. \tag{29}$$

Then the derivative of $G(d)$ is

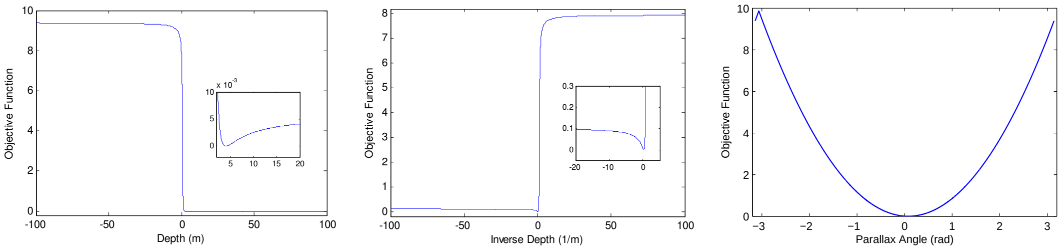
$$\begin{aligned} G'(d) =& 2(\operatorname{atan2}(d\sin\theta - \|\mathbf{t}\|\sin\phi_t,\ d\cos\theta - \|\mathbf{t}\|\cos\phi_t) - \phi_2 - \theta_2) \\ & \frac{-\|\mathbf{t}\|\sin\varphi}{d^2 + \|\mathbf{t}\|^2 - 2d\|\mathbf{t}\|\cos\varphi}. \end{aligned} \tag{30}$$

Therefore, $G'(d) = 0$ not only when

$$\operatorname{atan2}(d\sin\theta - \|\mathbf{t}\|\sin\phi_t,\ d\cos\theta - \|\mathbf{t}\|\cos\phi_t) - \phi_2 = \theta_2 \tag{31}$$

which means the observation computed from the depth d is consistent with the actual observation, but also when $\varphi = 0$ which means the camera motion is aligned with the direction to the feature. Furthermore, $G'(d) \to 0$ if $d \to \infty$ or $\varphi \to 0$.

As shown in Figure 5(a), when the depth of the feature is large, the objective function is almost flat, making it difficult for a numerical optimization algorithm to converge to the true solution.



(a) XYZ (depth) (b) inverse depth (c) parallax angle

Figure 5: One-dimensional objective function of BA using different feature parametrizations.

5.2.2 Inverse Depth Parametrization

Suppose the feature is parametrized as $\mathbf{F} = [\theta,\ \rho]^T$. When θ is fixed the objective function can be written as

$$G(\rho) = (\theta - \phi_1 - \theta_1)^2 + (\operatorname{atan2}(\sin\theta - \rho\|\mathbf{t}\|\sin\phi_t,\ \cos\theta - \rho\|\mathbf{t}\|\cos\phi_t) - \phi_2 - \theta_2)^2. \tag{32}$$

The derivative of $G(\rho)$ is

$$\begin{aligned} G'(\rho) = & 2(\operatorname{atan2}(\sin\theta - \rho\|\mathbf{t}\|\sin\phi_t,\ \cos\theta - \rho\|\mathbf{t}\|\cos\phi_t) - \phi_2 - \theta_2) \\ & \frac{\|\mathbf{t}\|\sin\varphi}{1 + \rho^2\|\mathbf{t}\|^2 - 2\rho\|\mathbf{t}\|\cos\varphi}. \end{aligned} \tag{33}$$

Therefore, $G'(\rho) = 0$ not only when

$$\operatorname{atan2}(\sin\theta - \rho\|\mathbf{t}\|\sin\phi_t,\ \cos\theta - \rho\|\mathbf{t}\|\cos\phi_t) - \phi_2 = \theta_2 \tag{34}$$

which means the observation computed from the inverse depth ρ is consistent with the actual observation, but also when $\varphi = 0$ which means the camera motion is aligned with the direction to the feature. Furthermore, $G'(\rho) \to 0$ if $\rho \to \infty$ or $\varphi \to 0$.

An example of 1D objective function using inverse depth parametrization is shown in Figure 5(b). It can be seen that the true solution is also not easy to find because of the large flat region.

5.2.3 Parallax Angle Parametrization

Suppose the feature is parametrized as $\mathbf{F} = [\theta,\ \omega]^T$. When θ is fixed, the objective function can be written as

$$G(\omega) = (\theta - \phi_1 - \theta_1)^2 + (\theta + \omega - \phi_2 - \theta_2)^2. \tag{35}$$

The derivative of $G(\omega)$ is

$$G'(\omega) = 2(\theta + \omega - \phi_2 - \theta_2). \tag{36}$$

Note that the only condition when $G'(\omega) = 0$ is $\theta + \omega - \phi_2 = \theta_2$ which gives the true solution of ω.

Furthermore, as seen in Figure 5(c), there is no region when the objective function is flat.

5.3 The Error in the Initial Feature Location Estimate

This section discusses the impact of the different feature parametrizations on the errors in the initial feature location estimate. This is described by the sensitivity of errors in different feature parameters w.r.t the observation noise. It will be shown that the initialization of the proposed parallax angle feature parametrization is less sensitive to the observation noise, as compared to the XYZ or inverse depth models.

Consider the 2D bearing only scenario shown in Figure 4. Let the true bearings be $\bar{\theta}_1$ and $\bar{\theta}_2$. The errors of the observation angles are assumed to be zero mean Gaussian with variance δ_θ^2 and are denoted by θ_1^Δ and θ_2^Δ

$$\begin{aligned} \theta_1 &= \bar{\theta}_1 + \theta_1^\Delta, \quad \theta_1^\Delta \sim N(0, \delta_\theta^2) \\ \theta_2 &= \bar{\theta}_2 + \theta_2^\Delta, \quad \theta_2^\Delta \sim N(0, \delta_\theta^2). \end{aligned} \tag{37}$$

It is also assumed that the orientations ϕ_1 and ϕ_2, as well as the relative translations ϕ_t and $\|\mathbf{t}\|$ of the two camera poses, are known.

5.3.1 XYZ Parametrization

Suppose φ and $\bar{\varphi}$ are the computed and true angles from the relative translation to the first feature observation, then

$$\bar{\varphi} = \bar{\theta}_1 + \phi_1 - \phi_t \tag{38}$$

$$\varphi = \theta_1 + \phi_1 - \phi_t = \bar{\varphi} + \theta_1^\Delta. \tag{39}$$

From the sine law, the true depth of the feature can be computed as

$$\bar{d} = \frac{\|\mathbf{t}\| \sin(\bar{\omega} + \bar{\varphi})}{\sin \bar{\omega}}. \tag{40}$$

And the computed depth

$$d = \frac{\|\mathbf{t}\| \sin(\omega + \varphi)}{\sin \omega} = \frac{\|\mathbf{t}\| \sin(\bar{\omega} + \bar{\varphi} + \theta_2^\Delta)}{\sin(\bar{\omega} + \theta_2^\Delta - \theta_1^\Delta)}. \tag{41}$$

Therefore,

$$\begin{aligned} \frac{\partial d}{\partial \theta_1^\Delta} &= \frac{\sin(\bar{\omega} + \bar{\varphi} + \theta_2^\Delta) \cos(\bar{\omega} + \theta_2^\Delta - \theta_1^\Delta)}{\sin^2(\bar{\omega} + \theta_2^\Delta - \theta_1^\Delta)} \|\mathbf{t}\| \\ \frac{\partial d}{\partial \theta_2^\Delta} &= \frac{-\sin(\bar{\varphi} + \theta_1^\Delta)}{\sin^2(\bar{\omega} + \theta_2^\Delta - \theta_1^\Delta)} \|\mathbf{t}\|. \end{aligned} \tag{42}$$

When the parallax angle $\bar{\omega}$ is close to zero, the error of the initial value of feature depth will be very large. The large error in depth d can result in a large error in XYZ.

5.3.2 Inverse Depth Parametrization

Inverse depth parametrization overcomes the issues associated with distant features

$$\rho = \frac{1}{d} = \frac{\sin(\bar{\omega} + \theta_2^\Delta - \theta_1^\Delta)}{\|\mathbf{t}\| \sin(\bar{\omega} + \bar{\varphi} + \theta_2^\Delta)}. \tag{43}$$

Therefore,

$$\begin{aligned} \frac{\partial \rho}{\partial \theta_1^\Delta} &= \frac{-\cos(\bar{\omega} + \theta_2^\Delta - \theta_1^\Delta)}{\|\mathbf{t}\| \sin(\bar{\omega} + \bar{\varphi} + \theta_2^\Delta)} \\ \frac{\partial \rho}{\partial \theta_2^\Delta} &= \frac{\sin(\bar{\varphi} + \theta_1^\Delta)}{\|\mathbf{t}\| \sin^2(\bar{\omega} + \bar{\varphi} + \theta_2^\Delta)}. \end{aligned} \tag{44}$$

While only small parallax angles may not cause any problems, when the angle $\bar{\omega} + \bar{\varphi}$ is close to zero, the error in the initial value of ρ will be large. Note that $\bar{\omega} + \bar{\varphi} \approx 0$ is equivalent to $\bar{\varphi} \approx 0$. This occurs when the camera is moving towards or away from the feature at the instant of initialization.

5.3.3 Parallax Angle Parametrization

The global bearing angles of feature from the two camera poses are $\theta_1 + \phi_1$ and $\theta_2 + \phi_2$, respectively. Suppose $\bar{\omega}$ is the true parallax angle and ω is the computed parallax angle, then

$$\bar{\omega} = (\bar{\theta}_2 + \phi_2) - (\bar{\theta}_1 + \phi_1) \tag{45}$$

$$\omega = (\theta_2 + \phi_2) - (\theta_1 + \phi_1) = \bar{\omega} + \theta_2^\Delta - \theta_1^\Delta. \tag{46}$$

Therefore,

$$\frac{\partial \omega}{\partial \theta_1^\Delta} = -1 \quad \text{and} \quad \frac{\partial \omega}{\partial \theta_2^\Delta} = 1. \tag{47}$$

Thus, the error in the parallax angle has the same scale as the observation error and is not affected by the geometrical configuration. Furthermore, the absolute error of the parallax angle will always be small since the errors in the observation angles θ_1^Δ and θ_2^Δ are typically small.

6 Simulation and Experimental Results

Simulation and experimental datasets have been used to compare the proposed ParallaxBA with BA using the Euclidian XYZ and inverse depth parametrizations, as well as the BA packages SBA, sSBA, g2o in terms of convergency, accuracy and efficiency.

Table 1: Sensor Model in Simulation

Parameter	Description
Sensor.FOV = $[-\pi/4,\ \pi/4]$	FOV: Bearing $[Min,\ Max]$
Sensor.Distance = $[0,\ +\infty]$	Observation distance $[Min,\ Max]$
Sensor.Resolution = [800, 800]	Image resolution $[u,\ v]$
Sensor.PP = [400, 400]	Principle point $[P_u,\ P_v]$ in pixel
Sensor.FL = [400, 400]	Focal length $[F_u,\ F_v]$ in pixel
Sensor.uvNoise = $0.1 \times randn$	uv Noise Gaussian $\sigma = 0.1$

Table 2: Simulation Details

Simulation 1

Parameter	Description
Feature.Dis = [10000, 10000, 10]	Feature distribution area XYZ(m)
Feature.Resolution = [10 10 2]	Uniform Feature Resolution XYZ
Feature.Dis2 = [50, 50, 10]	Second set of features
Feature.Resolution2 = [20 20 4]	
Camera.TotalSteps = 23	Total number of Steps
Camera.Trajectory = ‘Circle’	Trajectory
Camera.Dis = 5	Step Distance (m)

Simulation 2

Feature.Dis = [50, 50, 10]	Camera.TotalSteps = 21
Feature.Resolution = [20 20 4]	Camera.Trajectory = ‘Line’
Camera.Dis = 2	

Simulation 3

Parameter	Description
Feature.Dis = [10000, 10000, 10]	Feature.Dis2 = [50, 50, 10]
Feature.Resolution = [10 10 2]	Feature.Resolution2 = [20 20 4]
Camera.TotalSteps = 66	Camera.Trajectory = ‘Square’
Camera.Dis = 2.5	

6.1 Simulation Results

Computer simulations presented in this section are designed to highlight the difference between ParallaxBA and BA using Euclidean XYZ and inverse depth parametrization in some particular scenarios. In the computer simulations, the camera is modeled with a FOV of $[-\pi/4,\ \pi/4]$, $[0,\ +\infty]$ observation distance, 800×800 image resolution with the principal point at $[400, 400]$ and a focal length of [400, 400]. Random Gaussian noise with $\sigma = 0.1$ is added to the theoretical image coordinates as the observations of features in all the simulations. These are summarized in Table 1. Parameters used in different simulations are detailed in Table 2.

All the relative poses are computed from two consecutive images with known feature matches and the noisy image coordinates using the eight point algorithm

(Hartley and Zisserman 2003) which is the same method used later in the real experiments. Relative scale of poses is assumed to be 1 for calculating the initial values.

To initialize the features using parallax angle parametrization, the method proposed in Section 4.4 is used. In order to make a fair comparison with other parametrizations, the two anchors used in parallax angle parametrization and the two projections corresponding to these two anchors are used to calculate the position of the feature by using the linear triangulation algorithm (Hartley and Zisserman 2003). The initial values of XYZ and inverse depth parametrizations for each feature are both computed from this triangulation result. Thus, the initial estimates of the feature locations in the three parametrizations are equivalent, and the initial objective functions (the initial errors of the reprojections) of BA using different parametrizations are identical.

6.1.1 Singularity of Euclidean XYZ Parametrization

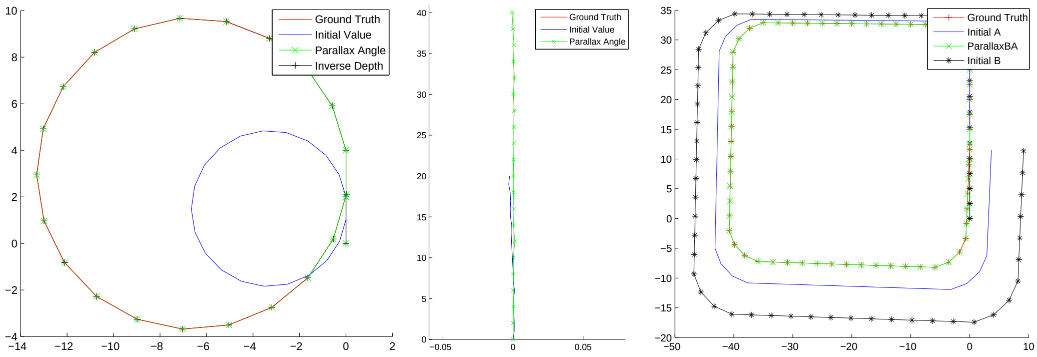
Simulation 1 is designed to show the singularity problem and the existence of the "long flat valley" of the objective function in BA using the Euclidean XYZ parametrization, when some distant features are observed. In particular, one set of features are assumed to be uniformly distributed in the {[-5000,5000],[-5000,5000],[-5,5]} cuboid with resolution [10;10;2] (i.e. $10 \times 10 \times 2 = 200$ features in the cuboid) and another set of features are uniformly distributed in the {[-25,25],[-25,25],[-5,5]} cuboid with resolution [20;20;4]. Simulation 1 has 23 poses with a circular trajectory as shown in Figure 6(a) (the start pose $(0, 0, 0)$ is not on the circle).

Four algorithms are compared using this simulation. These are, ParallaxBA with Gauss-Newton (GN), PaprallaxBA with Levenberg-Marquardt (LM), BA with XYZ and GN, BA with XYZ and LM.

When GN is used for solving BA, BA with XYZ parametrization diverges, with the information matrix becoming singular, while ParallaxBA converges. When LM is used in the BA algorithms, both BA with XYZ and ParallaxBA converge to nearly the same mean square error (MSE) of the reprojections*. The results are shown in Figure 6(a) and Table 3. Thus, the parallax angle parametrization can easily deal with the distant features using simple GN in this scenario, while XYZ parametrization needs the more robust LM algorithm.

The condition numbers of the information matrix $J^T J$ in each iteration for the four algorithms are shown in Figure 7(a). It is clear that the condition number of $J^T J$ in BA with XYZ parametrization is much larger than that of ParallaxBA showing the singularity problem of XYZ parametrization. As shown in Table 3, although LM can handle the singularity of XYZ parametrization by introducing a damping factor (12), the number of iterations required in LM is very large and the final MSE after 88 iterations is still larger than that of ParallaxBA obtained with only 6 iterations of GN.

*The MSE of the reprojections is equal to the objective function defined in (10) divided by the number of observations, as used in SBA (Lourakis and Argyros 2009).



(a) Simulation 1 (b) Simulation 2 (c) Simulation 3

Figure 6: ParallaxBA results of Simulation 1, 2 and 3, and the converged results of BA with inverse depth using GN for Simulation 1.

Furthermore, when BA with XYZ and LM is used, the depths of some distant features converge to very large values while all the other variables are nearly the same as the converged results from ParallaxBA. When these large depth values are changed, the objective function does not change significantly. And when changing other parameters around the obtained result, both the positive and negative direction of each parameter will make the objective function increase. Also, in the last few iterations during LM, the objective function and all the variables do not change except these depth values. When the LM result is used as the initial guess to GN, GN does not converge with information matrix singular. Thus, BA with XYZ parametrization and LM for this simulation converges to a point in a "long flat valley" instead of a real local minimum. The 1D and 2D surface and contour of the objective function around the converged result are not shown to save space but they are very similar to those in Figure 11. It is clear that the existence of such "long flat valleys" make it difficult to obtain the true optimal solution.

It should be mentioned that for this simulation, BA with inverse depth parametrization and GN can also converge very easily to the same MSE as ParallaxBA, demonstrating that the inverse depth parametrization can also deal with the distant features effectively. The result of BA with inverse depth is also shown in Figure 6(a).

6.1.2 Singularity of Inverse Depth Parametrization

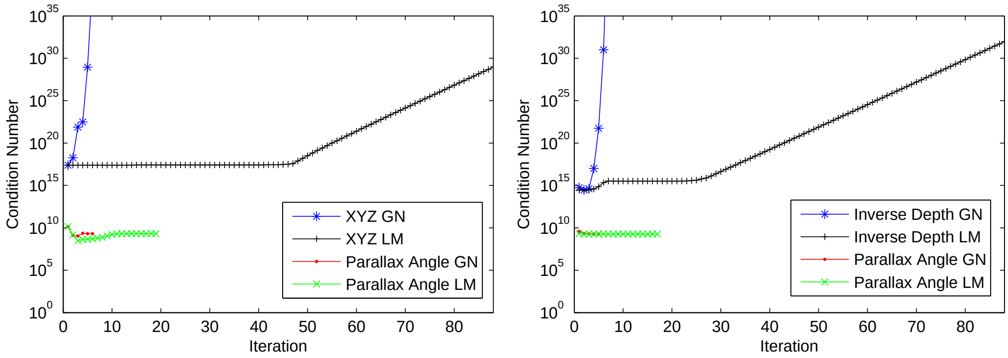
To demonstrate the possible singularity of BA using inverse depth parametrization, a more extreme scenario is designed. In Simulation 2 the camera moves along a straight line for 21 steps with the features uniformly distributed in the {[-25,25],[-25,25],[-5,5]} cuboid with resolution [20;20;4]. In this simulation, there are 5 features in the direction of the camera motion and each of these features is observed only twice.

Four algorithms are tested using this simulation. These are, ParallaxBA with GN, ParallaxBA with LM, BA with inverse depth and GN, and BA with inverse depth and LM. The final result and details of the convergence behavior

Table 3: Convergence of Simulation 1 and 2

	Parallax Angle GN		Parallax Angle LM		XYZ GN			XYZ LM	
	Final MSE	Iteration	Final MSE	Iteration	Final MSE	Iteration	Singularity	Final MSE	Iteration
Simu 1	0.01342625	6	0.01342625	19	N		Singular	0.01342868	88
	Parallax Angle GN		Parallax Angle LM		Inverse Depth GN			Inverse Depth LM	
	Final MSE	Iteration	Final MSE	Iteration	Final MSE	Iteration	Singularity	Final MSE	Iteration
Simu 2	0.01594779	5	0.01594779	17	N		Singular	0.01594781	88

*The maximum number of iterations is 88.



(a) Simulation 1

(b) Simulation 2

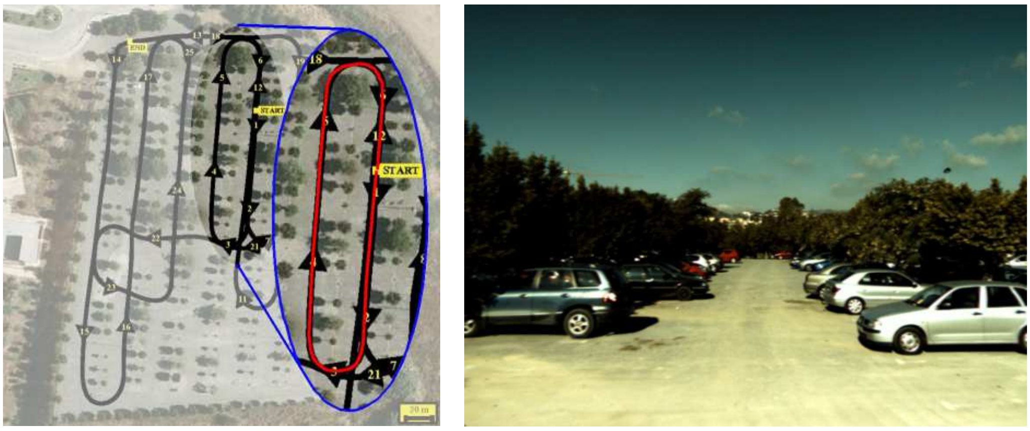
Figure 7: Condition numbers of the information matrices in the simulations.

of Simulation 2 are shown in Figure 6(b) and Table 3, respectively. As seen in Table 3, ParallaxBA converges with both GN and LM. BA using inverse depth parametrization and GN does not converge. BA using inverse depth parametrization and LM converges to almost the same MSE as that of ParallaxBA.

The condition numbers of the information matrix J^TJ in each iteration for the four algorithms are shown in Figure 7(b). It is clear that the condition number of J^TJ in BA with inverse depth parametrization is much larger than that of ParallaxBA showing the singularity problem of inverse depth parametrization in this scenario. As shown in Table 3, although LM can handle the singularity of the inverse depth parametrization by introducing a damping factor (12), the number of iterations required in LM is very large and the final MSE after 88 iterations is still larger than that of ParallaxBA achieved with only 5 iterations of GN.

As analyzed in Section 5.1, the inverse depth parametrization leads to singularity due to the existence of features in the direction of camera motion. Since there is no information about the depth (apart from noise) for these features, the Jacobian is (close to) rank deficient. Thus, the matrix J^TJ is ill-conditioned.

For Simulation 2, similar to Simulation 1, when BA with inverse depth and LM is used, the inverse depth of a feature in the direction of camera motion goes to a large value while all the other variables are nearly the same as the converged results from ParallaxBA. The 1D and 2D objective function are not shown to save space but they are very similar to those shown in Figure 12. It is clear that the existence of the "long flat valleys" in the objective function make the BA



(a) Trajectory (b) Image

Figure 8: The trajectory and image of PARKING-6L dataset.

with inverse depth parametrization difficult to converge in this scenario.

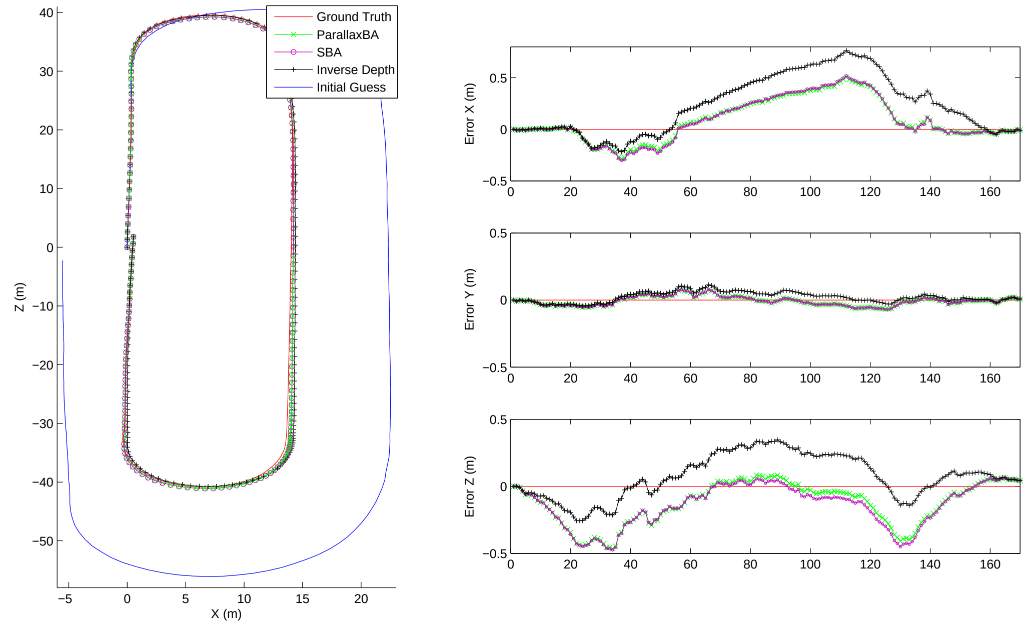
BA with XYZ and GN also fails in this simulation. When LM is used, BA with XYZ parametrization reaches the maximum iteration number with the final MSE 0.01594811, which is slightly larger than that of ParallaxBA and BA with inverse depth and LM. The singularity problem also exits for XYZ parametrization in this scenario since the features in the direction of camera motion lack depth information.

6.1.3 The Impact of the Initialization of Pose Estimation

In the previous analysis and simulations, the impacts of different parametrizations of features are considered. As BA is a high dimensional nonlinear optimization problem, the initial guess of poses also plays an important role. In this simulation, it will be shown that, although a good feature parametrization is used in the proposed ParallaxBA algorithm, a poor initialization of camera poses can cause the divergence of GN and the singularity of the information matrix. That is, the initial value of poses is critical in all the BA algorithms including ParallaxBA.

Simulation 3 is designed to have the same environment as Simulation 1, which contains both nearby and distant features. The camera is simulated as a square trajectory with 66 steps. The ground truth of the camera trajectory, as well as the result of converged ParallaxBA, are shown in Figure 6(c).

For normal initialization of the camera poses solved by every two images with relative scale assumed to be 1, ParallaxBA converges with GN. When relative scales are randomly selected between 1 to 1.2 for initialization (one example is given as Initial A in Figure 6(c)), ParallaxBA can converge very easily with GN. However when the relative scales are selected to be between 1 to 1.5 (e.g. Initial B in Figure 6(c)), ParallaxBA with GN diverges, with some of the pose estimates in the state vector becoming very poor, and the information matrix becomes singular. When LM is used as the optimization algorithm, ParallaxBA converges. For ParallaxBA with LM, the step size is controlled by the damping



(a) Trajectories from different algorithms, initial guess and the ground truth

(b) Errors in X,Y,Z Axis

Figure 9: ParallaxBA, BA using inverse depth parametrization with LM and SBA results of the Malaga PARKING-6L dataset containing 170 images (only the 170 camera poses are shown). The final MSE: 0.109209 (ParallaxBA), 0.125020 (Inverse Depth), 0.216219 (SBA).

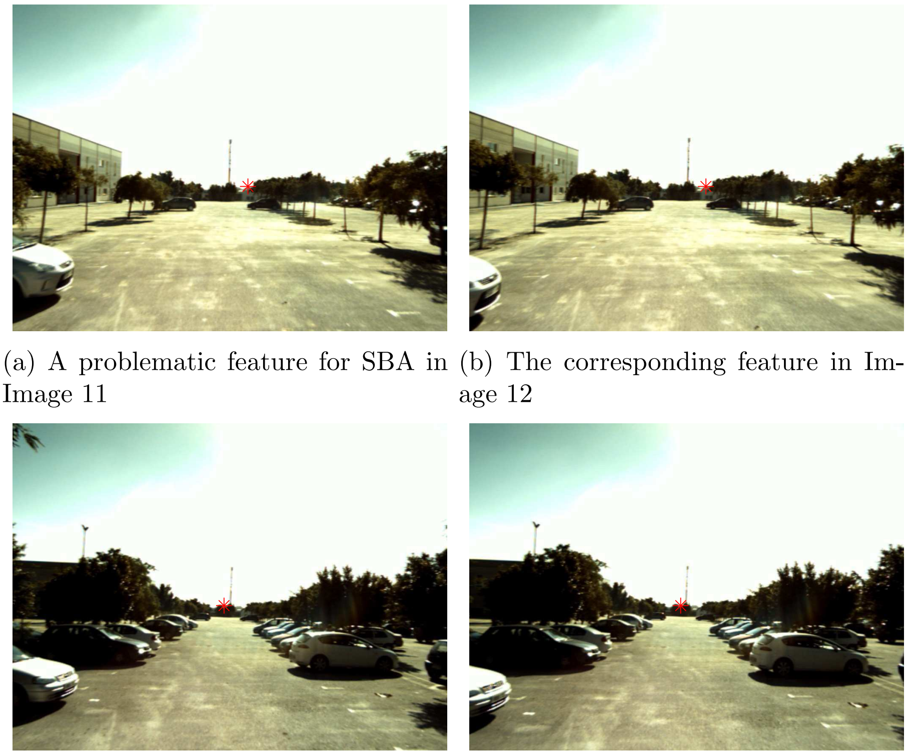
factor and an update causing the increase of objective function is not accepted preventing the pose estimates from becoming very poor. Thus, the condition number is not very large and the information matrix is not ill-conditioned.

6.2 Experimental Results

This section presents some results using a series of real experimental datasets. First, a small-scale (250m single loop) publicly available dataset is used to compare BA with XYZ, inverse depth and parallax angle parametrizations. The results confirm that the singularity issue and the "long flat vallley" in the objective function appear in practice when XYZ and inverse depth parametrizations are used in BA, confirming the analysis in Section 5 and the results from the simulations in Section 6.1. These issues do not appear in ParallaxBA.

To demonstrate the accuracy and efficiency of the proposed ParallaxBA algorithm, four large-scale real experimental datasets are used to compare ParallaxBA with the state-of-art BA packages g2o and sSBA. Each BA package is treated as a black box. BA with inverse depth parametrization is not compared in the large-scale experiments as such a BA package is not publicly available.

The source code of ParallaxBA is released on OpenSLAM under project "ParallaxBA". All the experimental datasets used here are within the released code.



(a) A problematic feature for SBA in Image 11

(b) The corresponding feature in Image 12

(c) A problematic feature for inverse depth parametrization in Image 151

(d) The corresponding feature in Image 152

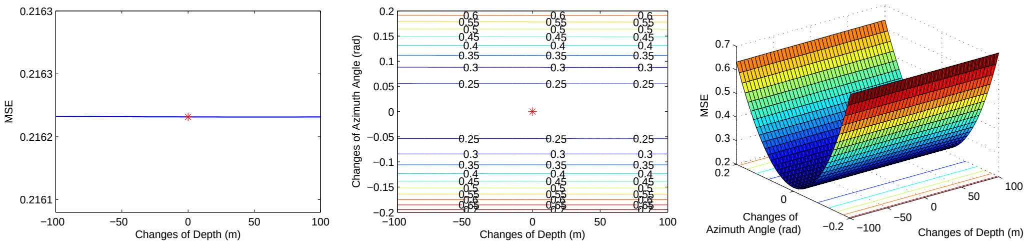
Figure 10: Problematic features in Malaga PARKING-6L dataset.

6.2.1 Small Loop of Malaga PARKING-6L Dataset

Publicly available Malaga 2009 Robotic Dataset Collection (Blanco et al. 2009) is used for demonstrating that the singularity issues also occur in a real-life situation. This dataset was collected using an electric car equipped with laser scanners, cameras, IMU, and GPS receivers. As described in (Blanco et al. 2009), a centimeter-level ground truth is provided making this dataset an ideal test bed for SLAM.

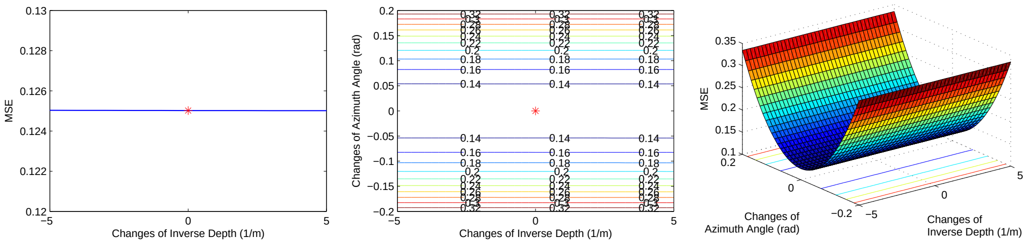
Images captured by the right camera are used. The image resolution is 1024×768 and camera calibration parameters are provided in the dataset. SIFT (Lowe 2004) is used for feature detection with subpixel accuracy and for matching, including the loop closure detection. A multi-level RANSAC (Fischler and Bolles 1981) with thresholds as 2, 0.5, 0.1 and 0.05 is used to remove the outliers (Zhao et al. 2010). The eight point algorithm (Hartley and Zisserman 2003) is used for relative pose computation because it provides a result that is adequate for BA (Zhao et al. 2010).

A sequence of images collected from a 250m close loop trajectory named the PARKING-6L dataset is selected (Figure 8). Data is sub sampled at 2.5 Hz from the original 7.5Hz reducing the number of images to 170 from original 508 images as the keyframes for this loop. BA with this keyframe dataset involves 170 poses, 58,404 features and 167,285 projections in total. The initial guess is obtained from visual odometry as shown in Figure 9. ParallaxBA diverges when



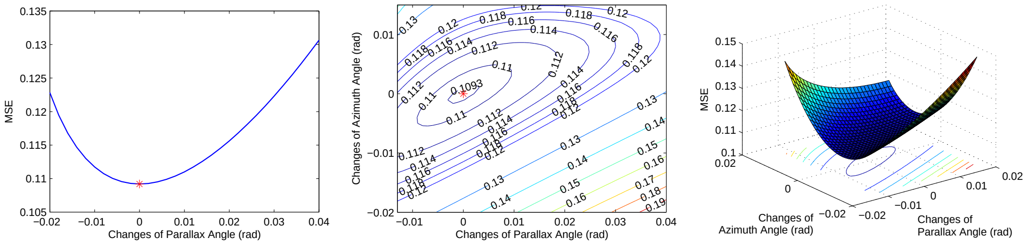
(a) 1D curve (b) 2D Contour (c) 3D Surface

Figure 11: The objective function of SBA as a function of the depth (and the azimuth angle) for the problematic feature in the Malaga PARKING-6L dataset. The red dot corresponds to the final result of SBA.



(a) 1D curve (b) 2D Contour (c) 3D Surface

Figure 12: The objective function of BA with inverse depth as a function of the inverse depth (and the azimuth angle) for the problematic feature in the Malaga PARKING-6L dataset. The red dot corresponds to the final result of BA with inverse depth parametrization.



(a) 1D curve (b) 2D Contour (c) 3D Surface

Figure 13: The objective function of ParallaxBA as a function of the parallax (and the azimuth angle) for the problematic feature for inverse depth in the Malaga PARKING-6L dataset. The red dot corresponds to the final result of ParallaxBA.

Table 4: The Depth (d)/Inverse Depth (ρ) of Problematic Features and the Corresponding MSE in Some Iterations of BA for Malaga Dataset

SBA (BA + XYZ)

Iteration	136	142	149	157	167	175	189
d	595.623	649.973	682.323	693.037	682.557	660.176	602.594
MSE	0.222991	0.218274	0.217356	0.216907	0.216624	0.216485	0.216325

BA + Inverse Depth

Iteration	101	133	154	156	157	158	200
ρ	14.2796	16.0329	17.6923	20.1636	23.8773	28.0191	33.7862
MSE	0.125334	0.125143	0.125084	0.125073	0.125060	0.125044	0.125020

GN is used as the optimization algorithm. The reason for this is speculated to be similar to that described in Section 6.1.3. ParallaxBA with the LM algorithm takes 61 iterations to converge. The ground truth and the result from ParallaxBA are shown in Figure 9. BA using both inverse depth parametrization and XYZ parametrization are also implemented. The SBA package (Lourakis and Argyros 2009) is used for BA with XYZ parametrization. Algorithms converge when LM is used while GN fails to produce a result due to divergence. Results obtained are also shown in Figure 9. The final MSE of ParallaxBA, BA using inverse depth parametrization and SBA are 0.109209, 0.125020, and 0.216219, respectively. The details of the convergence are shown in Figure 1.

Similar to Simulation 1 and Simulation 2, it was found that there are 55 problematic features when XYZ parameterization is used, and 2 problematic features when inverse depth parameterization is used for BA. These are either distant features or features in the direction of the camera motion. This is the same as that mentioned in Section 6.1.1 and 6.1.2. Some of these features are shown in Figure 10 (the feature with ID 3251 for SBA and the feature with ID 52280 for BA with inverse depth in the dataset released with the ParallaxBA code). The depth/inverse depth obtained for such problematic features and the MSE in some iterations of BA are shown in Table 4. This table demonstrates that these problematic features cause the optimization algorithm to converge very slowly, pointing to the presence of a “long flat valley”. The 1D and 2D objective functions around the estimated parameters for the problematic features are shown in Figure 11, Figure 12 and Figure 13. It is clear that the objective function is not well behaved for BA with both XYZ parameterization and inverse depth parameterization while the parallax angle parameterization produces a good result.

A formulation of BA that uses angles rather than the image coordinates in formulating the objective function is described in (Mouragnon et al. 2009). To examine the impact of this alternative formulation, BA is implemented with both XYZ parametrization (Angle+XYZ) and parallax angle parametrization (Angle+Parallax) using the Malaga dataset. The LM algorithm is used for the optimization. Angle+Parallax converges to the objective function of 0.01701969, while Angle+XYZ converges to objective function of 0.03383568. The information matrix is singular in the latter case. It is clear that the issues of XYZ presentation as described before still exist and parallax angle parametrization remains superior for this alternative formulation of the objective function.

6.2.2 Aerial Photogrammetric Datasets

In this and the next two subsections, the performance of ParallaxBA is compared with those of sSBA and g2o in terms of convergence and efficiency using four large-scale datasets. As described in (Konolige 2010), sSBA is much more efficient than SBA, and thus SBA is not used in this evaluation. All the three BA implementations are run on an Intel i5-3210M@2.5GHz CPU. The parameter τ to initialize the damping factor in LM is set such that each of the algorithms produced the best performance in terms of convergence and final MSE, resulting in τ equal to 10^{-6}, 10^{-6}, and 10^{-8} for ParallaxBA, sSBA and g2o, respectively. ParallaxBA uses the identical implementation of the LM algorithm as used in SBA while different implementations are used in sSBA and g2o. Therefore, it is not feasible to use the same thresholds for convergence criteria. Thus, the default thresholds in the releases of sSBA and g2o are used, and the thresholds used in ParallaxBA are the defaults in SBA. Further detailed discussions of these issues are given in Section 6.2.5.

The Village and College datasets in aerial photogrammetry are used first. In the Village dataset, there are 90, 7680 × 13824 pixels resolution images, taken with a digital mapping camera in a snake track. After SIFT and RANSAC, 305,719 features and 779,268 projections are extracted and used as input to BA. In the College dataset, 468, 5616 × 3744 images captured by a Canon camera are available. For this dataset, 1,236,502 features and 3,107,524 projections are obtained after processing with SIFT and RANSAC.

In order to make a fair comparison, the same feature initialization method as described in Section 6.1 is used. The initial MSE, final MSE, number of iterations, number of linear equation solving, time per iteration and total time are listed in Table 5 and Table 6. Moreover, in Figure 14(a) and Figure 15(a), the changes of MSE in each iteration are depicted. Figure 14(b) and Figure 15(b) show the results of ParallaxBA, where the blue points represent the estimated 3D features, and the triangular cones represent the camera centers.

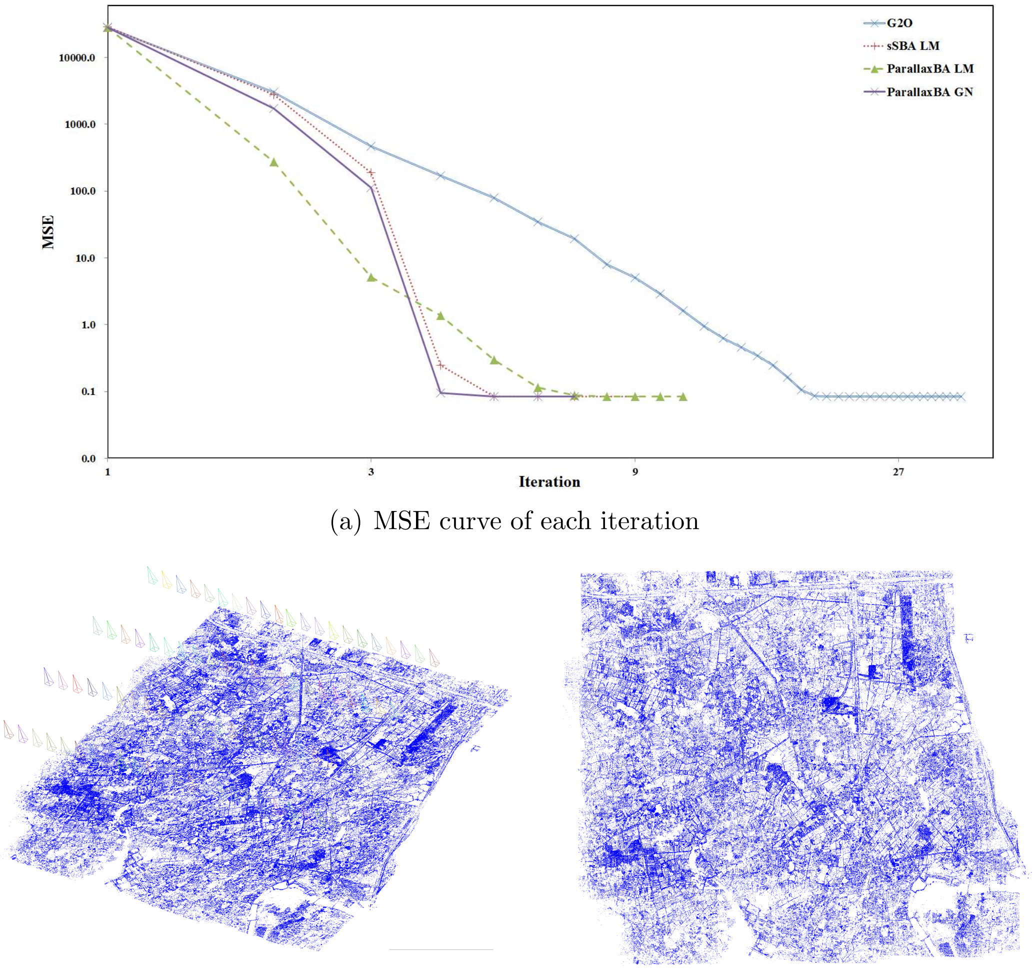
In these results, the small difference seen in the initial MSE between g2o and ParallaxBA is due to the use of a transformation from Euler angles to quaternion. The reason why the initial MSE of sSBA is different from that of g2o and ParallaxBA in Table 6 is that two additional operations are performed in the sSBA code. First, the original quaternion is changed in some scenarios. Second, the reprojection errors of the points located at the back of the camera are set to zero. If these operations are not performed in sSBA, then the initial MSE will be the same as that of g2o, but the convergence will in general not be as good. Section 6.2.5 provides a discussion on this issue. It should be noted here that publicly available sSBA and g2o packages without any modifications are used in all comparisons in Section 6.2.2–6.2.4.

For the Village dataset, identical values for final MSE are achieved by all

Table 5: Aerial Photogrammetric Village Dataset

90 poses, 305,719 3D features, 779,268 projections. Time in Seconds.

	g2o GN	g2o LM	sSBA	ParallaxBA GN	ParallaxBA LM
Initial MSE	28174.105559	28174.105559	28968.738359	28170.979450	28170.979450
Final MSE	N/A	0.083716	0.083716	0.083716	0.083716
Iteration	N/A	34	8	6	11
Solve	N/A	55	8	6	11
Time/Iter	N/A	N/A	N/A	0.67(Windows)	0.67(Windows)
	N/A	0.62(Linux)	0.56(Linux)	0.96(Linux)	0.96(Linux)
Total Time	N/A	N/A	N/A	5.07(Windows)	7.92(Windows)
	N/A	27.46(Linux)	4.54(Linux)	6.98(Linux)	12.23(Linux)



(a) MSE curve of each iteration

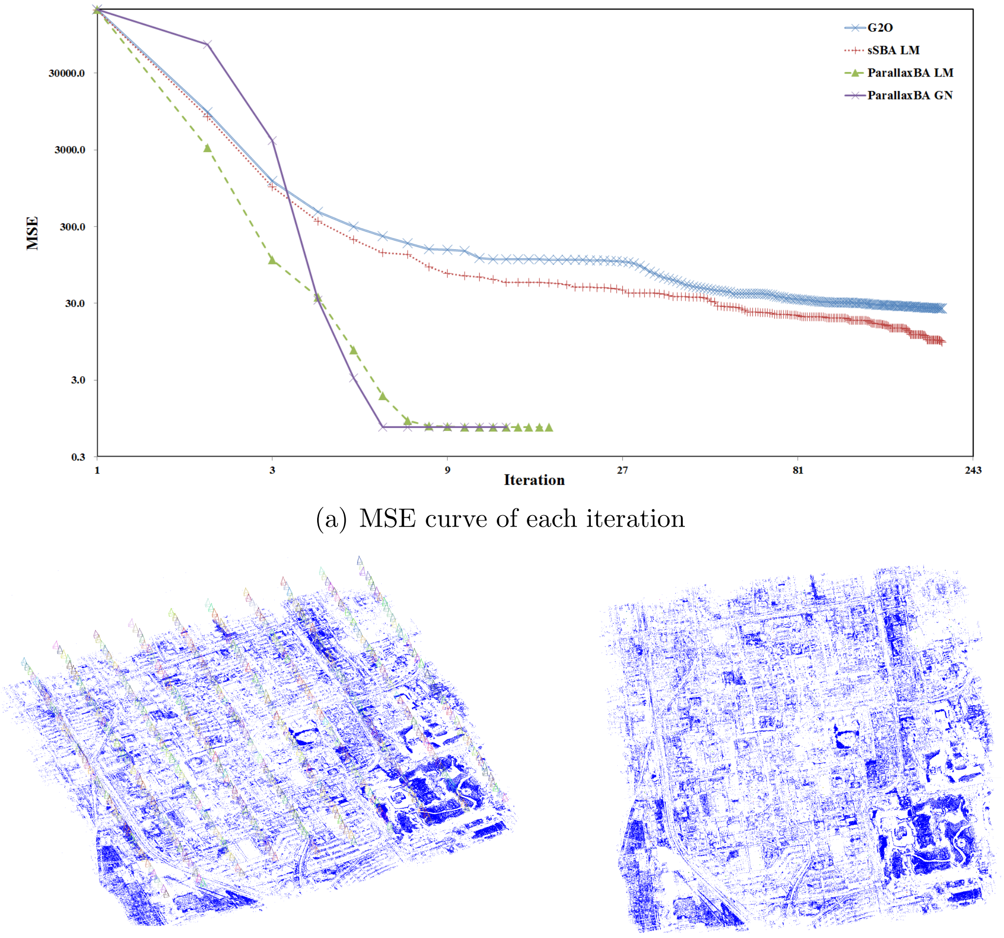
(b) Estimated 3D features (blue points) and camera centers (triangular cones) by ParallaxBA (3D view and top view)

Figure 14: Result of aerial photogrammetric Village dataset.

Table 6: Aerial Photogrammetric College Dataset

468 poses, 1,236,502 3D features, 3,107,524 projections. Time in Seconds.

	g2o GN	g2o LM	sSBA	ParallaxBA GN	ParallaxBA LM
Initial MSE	202329.445139	202329.445139	201046.610863	202329.447636	202329.447636
Final MSE	N/A	25.723307	9.272481	0.734738	0.734738
Iteration	N/A	200	200	12	17
Solve	N/A	349	228	12	17
Time/Iter	N/A	N/A	N/A	2.71(Windows)	2.71(Windows)
	N/A	2.51(Linux)	2.72(Linux)	3.85(Linux)	3.85(Linux)
Total Time	N/A	N/A	N/A	37.14(Windows)	49.68(Windows)
	N/A	674.83(Linux)	453.22(Linux)	51.55(Linux)	69.58(Linux)



(a) MSE curve of each iteration

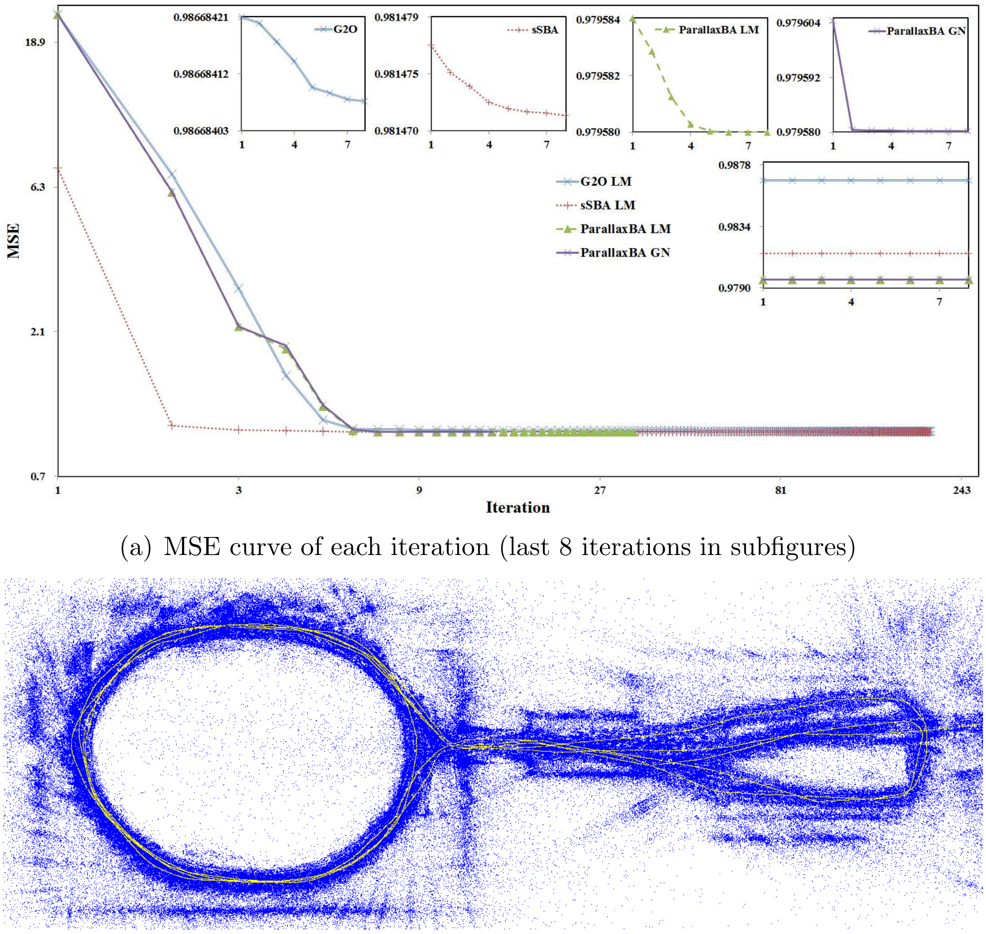
(b) Estimated 3D features (blue points) and camera centers (triangular cones) by ParallaxBA (3D view and top view)

Figure 15: Result of aerial photogrammetric College dataset.

Table 7: New College Dataset

3,500 poses, 449,096 3D features, 2,124,449 projections. Time in Seconds.

	g2o GN	g2o LM	sSBA	ParallaxBA GN	ParallaxBA LM
Initial MSE	23.354248	23.354248	7.295188	23.354933	23.354933
Final MSE	N/A	0.986684	0.981471	0.979580	0.979580
Iteration	N/A	200	200	13	33
Solve	N/A	313	219	13	33
Time/Iter	N/A	N/A	N/A	5.35(Windows)	5.35(Windows)
	N/A	4.12(Linux)	8.55(Linux)	7.60(Linux)	7.60(Linux)
Total Time	N/A	N/A	N/A	77.52(Windows)	183.25(Windows)
	N/A	3262.52(Linux)	1972.35(Linux)	108.39(Linux)	255.17(Linux)



(a) MSE curve of each iteration (last 8 iterations in subfigures)

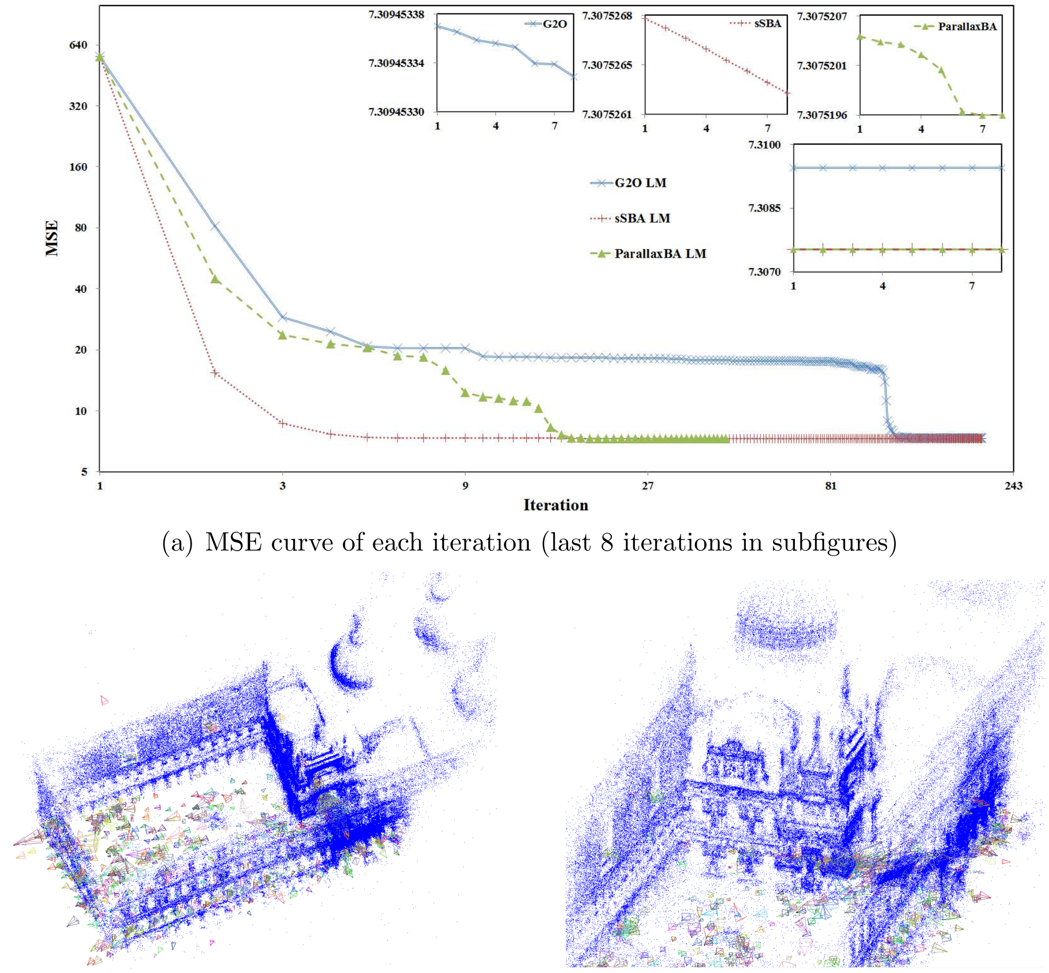
(b) Estimated 3D features (blue points) and camera centers (yellow line) by ParallaxBA

Figure 16: Result of New College dataset (as monocular vision).

Table 8: Venice Dataset

871 poses, 530,304 3D features, 2,838,740 projections. Time in Seconds.

	g2o GN	g2o LM	sSBA	ParallaxBA GN	ParallaxBA LM
Initial MSE	560.605910	560.605910	559.752359	560.817183	560.817183
Final MSE	N/A	7.309453	7.307526	N/A	7.307519
Iteration	N/A	200	200	N/A	43
Solve	N/A	300	219	N/A	74
Time/Iter	N/A	N/A	N/A	N/A	9.69(Windows)
	N/A	7.72(Linux)	18.75(Linux)	N/A	12.45(Linux)
Total Time	N/A	N/A	N/A	N/A	666.83(Windows)
	N/A	2191.27(Linux)	4367.73(Linux)	N/A	783.23(Linux)



(a) MSE curve of each iteration (last 8 iterations in subfigures)

(b) Estimated 3D features (blue points) and camera centers (triangular cones) by ParallaxBA

Figure 17: Result of Venice dataset.

four algorithms. However, sSBA and ParallaxBA with GN or LM require fewer iterations than g2o. For the College dataset, ParallaxBA with GN and LM can converge to MSE of 0.734738 within 12 and 17 iterations respectively, while g2o reaches MSE 25.723307, and sSBA reaches MSE 9.272481 after the maximum 200 iterations.

6.2.3 New College Dataset

The New College dataset provided with g2o on OpenSLAM is used for another comparison. As the original New College dataset is in stereo, all the observations from the right camera as well as all the features that are observed only once are deleted to obtain a monocular dataset. There are 3,500 camera poses, 449,096 features and 2,124,449 projections in this monocular dataset. The initial values provided in the dataset are directly used in sSBA and g2o. Initial values of camera poses and feature XYZ locations are used to compute the initial values of the parallax angle feature parameters to make sure that different algorithms start from the same initial guess. Detailed convergence information is shown in Table 7. The MSE curves are shown in Figure 16(a). Figure 16(b) shows the result of ParallaxBA.

The initial MSE of three algorithms are different due to the same reasons explained in Section 6.2.2. It is seen that, ParallaxBA requires fewer iterations as compared with g2o and sSBA with the final MSE being slightly smaller. The reason why sSBA converges faster at the beginning is discussed in Section 6.2.5.

6.2.4 Venice Dataset

The Venice dataset provided with g2o on OpenSLAM is also used for a further comparison. In this dataset, there are 871 camera poses, 530,304 features and 2,838,740 projections. The same strategy used in the previous section was used for initialization. Detailed convergence information is shown in Table 8. The MSE curves are shown in Figure 17(a). Figure 17(b) shows the result of ParallaxBA.

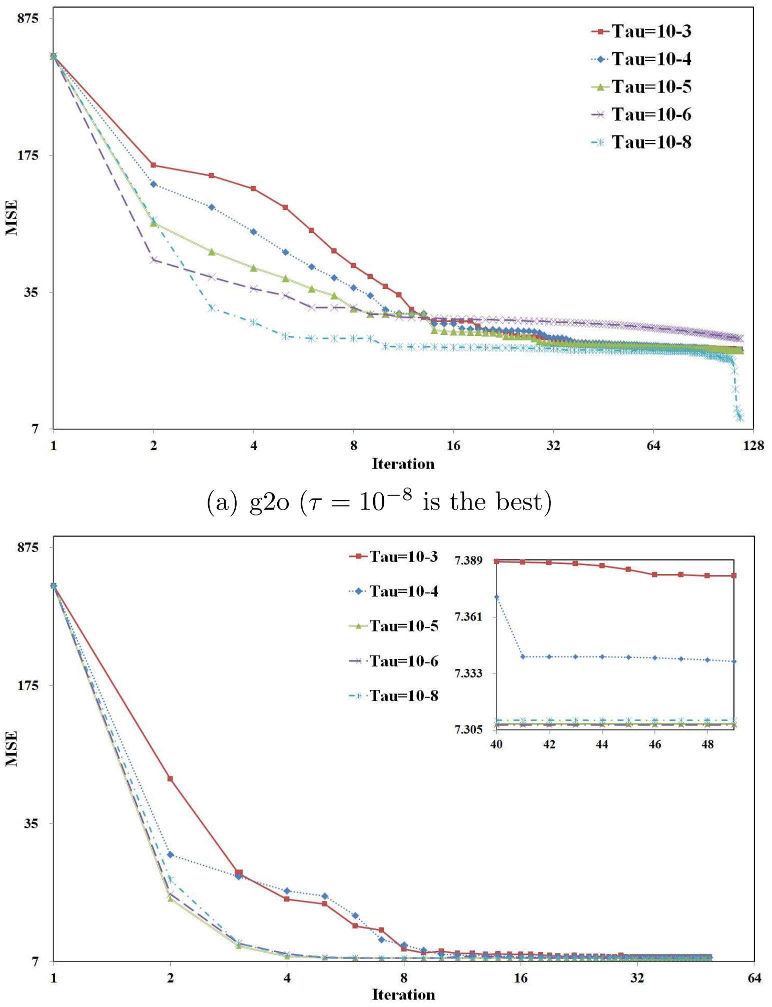
It was found that the GN algorithm fails to converge in ParallaxBA, probably because of the poor initial value of camera poses and/or the outliers in the data.

6.2.5 A Discussion on the Use of Different BA Packages

Despite the authors' best endeavours, different LM implementations and different parameters used in the publicly available BA packages make it difficult to have completely fair comparisons.

There are two sets of important parameters in the LM algorithm. One is the initial damping parameter (λ in (12)), which affects the convergence speed in LM; and the other is the set of thresholds, which decide when to stop the iteration.

The performance of the algorithms may heavily depend on the initial damping parameter (Kummerle et al. 2011), and thus the parameter τ to initialize the damping parameter in g2o (10^{-8}) and sSBA (10^{-6}) are chosen such that they



(a) g2o ($\tau = 10^{-8}$ is the best)

(b) sSBA ($\tau = 10^{-6}$ is the best)

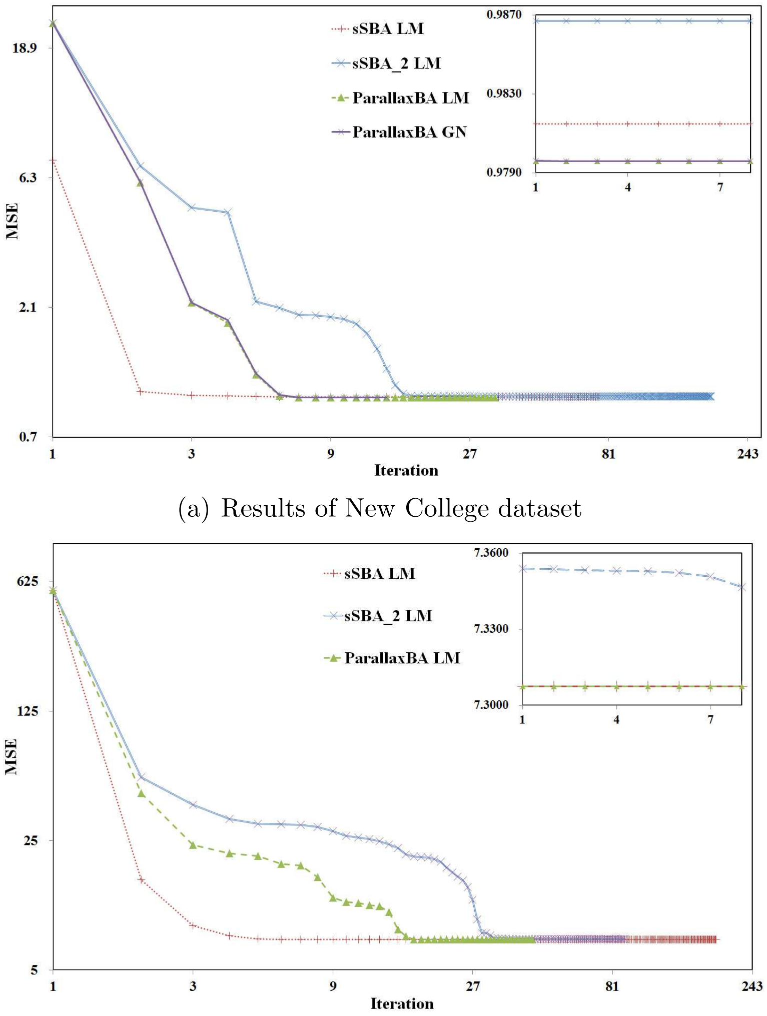
Figure 18: The impact of the parameter τ, which initializes the damping factor, on the convergence and final MSE with the Venice dataset.

provide the best performance. Impact of τ in these two algorithms is shown in Figure 18.

The LM implementations in sSBA, g2o and ParallaxBA are all different, so it is not feasible to use the same thresholds for convergence. Thus, in this paper, the thresholds used in all the three BA packages are their defaults. ParallaxBA uses the same LM algorithm as SBA, and the thresholds are also the defaults in SBA. In LM algorithms for BA, there are, in general, four conditions to stop the iteration: 1. the update of the state vector (Δ_k in (12)) is very small; 2. the value of the objective function (10) does not change much; 3. the gradient of the objective function is close to zero; and 4. the maximum number of iterations is reached. ParallaxBA and SBA use identical implementations of LM where all four conditions are used as the stopping criteria, while conditions 1 and 4 are used in sSBA and conditions 2 and 4 are used in g2o. Condition 4 is to avoid the algorithm iterating forever, while the results obtained by this criterion may not be a minimum. In addition, the maximum number of iterations is difficult to specify without the prior knowledge of the convergence information. It was set to 200 for all the results reported in this paper. Thus, the most reasonable stop conditions are 1-3. It should be mentioned that a large damping factor λ in LM will make both Δ_k and the change of the objective function close to zero. Thus, if the thresholds for conditions 1 and 2 are not small enough, LM can easily stop before convergence with poor quality results (for example, g2o will stop at MSE around 20 for Venice dataset if the thresholds are not small enough, see Figure 17(a)). This happens very often in BA and that is why the defaults of these thresholds are all set to be very small in SBA, g2o and sSBA (e.g. in SBA, 10^{-12} for condition 1 and 0 for condition 2). For a well-conditioned nonlinear least squares problem, when the algorithm converges, either condition 1, 2 or 3 will be satisfied because if a minimum of the objective function is reached, the gradient should be zero, and both the state vector and the objective function will not change anymore. However in an ill-conditioned problem, conditions 1-3 will be very hard to reach since both the state vector and the objective function will keep changing, and LM is often stopped by reaching the maximum iteration number, as shown in the experiments for BA with XYZ and inverse depth in this paper. To avoid the ill-conditioned systems in BA is one of the motivations of this paper (Section 1).

In the four evaluations with large-scale datasets, ParallaxBA outperforms g2o in most of the scenarios in terms of efficiency and accuracy. sSBA converges more rapidly in certain situations because in its implementation, the reprojection error of features located at the back of the camera is set as zero in each iteration in sSBA, while this is not the case in g2o and ParallaxBA. This special implementation is somewhat similar to the iterative re-weighted least squares used in the robust kernel methods (Zhang 1997). When the special implementation is removed, the convergence of sSBA is not as good as that of ParallaxBA for both New College and Venice datasets, as shown in Figure 19. Even with this special implementation, sSBA performance is inferior to that of ParallaxBA in some datasets (Figure 1 and Figure 15(a)). Also, as the objective function has been changed in sSBA, it is not reasonable to directly compare the timing and MSE of ParallaxBA/g2o with that of sSBA. In four out of the five experimental datasets, a number of image points are ignored in the calculation of the final MSE in sSBA (280 image points (Malaga), 0 (Village), 86 (College), 438 (New College), and 223 (Venice), respectively).

Based on the experimental results of using the four large-scale datasets, it is observed that: (1) The time per iteration in ParallaxBA is comparable to that of sSBA (out of four datasets, two are shorter, one is almost the same, one is longer). The time per iteration in ParallaxBA is slightly longer than that of g2o; (2) ParallaxBA always achieves the smallest MSE; and (3) Fewer iterations are needed in ParallaxBA. The last two points are due to the fact that ParallaxBA avoids the singularity of the information matrix and the “long flat valley” of the objective function caused by features, which occurs in the other BA packages.



(a) Results of New College dataset

(b) Results of Venice dataset

Figure 19: The convergence of sSBA, ParallaxBA, and sSBA without ignoring the reprojection error of features located at the back of the camera (sSBA_2). The subfigure shows the last 8 iterations.

6.3 Summary of the Simulation and Experimental Results

In summary, the simulation and experimental results confirm that:

(i) The far away features can result in a singular information matrix in BA when used with XYZ parametrization. The features in the direction of camera motion can cause singularity of the information matrix for both BA using XYZ and BA using inverse depth. These make it impossible to apply GN if such problematic features are present. When LM is applied, the singularity can be controlled to some extent due to the introduce of a damping factor (12). However, such features can cause the LM algorithm to converge very slowly due to the "long flat valley" of the objective function and might be trapped into such a "local minimum" (a point in the "long flat valley").

(ii) Parallax angle feature parametrization does not have issues for such features, and thus can avoid this kind of singularity. This makes it possible for GN to converge sometimes (still no guarantee, depending on the initial value, data quality, etc.). In general, ParallaxBA can converge more quickly (and with smaller MSE) as compared with BA using XYZ or inverse depth parametrization.

In theory, it may be possible to first identify the problematic features (for XYZ and inverse depth parametrizations) and then remove them before performing BA. However, in practice, the ground truth of the camera poses and the feature positions are not available, and thus it is not clear which feature may cause a problem. Since BA requires iterations and the state vector keeps changing and there are many features, whenever the values of one feature and its corresponding camera poses in one particular iteration satisfy the singularity condition, the convergence of the algorithm may become an issue. Thus, it is the best to avoid the singularity through feature parametrization such as using parallax angle.

7 Conclusion

This paper proposed ParallaxBA – a new bundle adjustment algorithm based on a novel feature parametrization that uses parallax angle as the key parameter for representing the three-dimensional location of the environment structure. The new representation is close to the measurement space and does not explicitly contain scale information. Rigorous analysis is presented as to why this parametrization is superior to existing feature parametrizations in BA.

Simulation and large-scale experimental results demonstrated that, in most cases, the proposed ParallaxBA has better accuracy, efficiency and convergence properties in comparison to the state-of-art BA packages, namely SBA, sSBA and g2o. The main reason is that the singularity caused by the existence of near zero parallax features in the existing BA algorithms can be avoided in ParallaxBA, resulting in fewer iterations for convergence and smaller final values for the objective function. Thus, the proposed ParallaxBA algorithm can be expected to be more likely to produce an acceptable result when used in practice. The source code of ParallaxBA is made available to the research community on OpenSLAM.

The current implementation of ParallaxBA is efficient for reasonably large datasets. For larger datasets, map joining ideas (e.g. (Ni et al. 2007)(Pinies and Tardos 2008)(Zhao et al. 2011)) can be used to further reduce the computational cost. As shown in this paper, the proposed ParallaxBA can converge with only Gauss-Newton iterations for some datasets. But for some other datasets, the Levenberg-Marquardt algorithm is required. Since BA is a very high dimensional nonlinear optimization problem, all methods available for BA, including the method proposed in this paper, cannot guarantee convergence to the true minimum under all conditions. The initial guess plays an important role in BA and requires further investigation. This paper has focused on the parametrization of features. It will be interesting to investigate the impact of the parametriza-

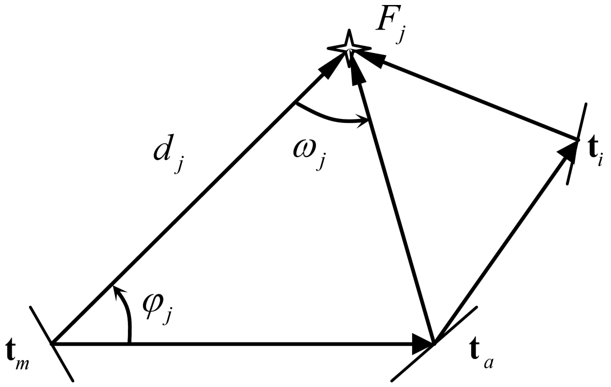


Figure 20: Computation of vector from different pose to feature.

tion of poses. Also, the robustness of ParallaxBA with respect to outliers (e.g. M-estimators (Zhang 1997)) requires further analysis and evaluations.

Interesting problems also include the analysis of the number of local minima (Huang et al. 2012) present in ParallaxBA and the estimate of the basin of attraction of the global minimum in ParallaxBA. Furthermore, the extension of ParallaxBA for stereo images, handling images taken by uncalibrated cameras (Agarwal et al. 2010), and robust parametrization for line features (Eade and Drummond 2009)(Zhao et al. 2014) are also very interesting avenues for future research.

A Observation Function for Bundle Adjustment

This appendix describes the derivation of $\tilde{\mathbf{x}}_j^i$ ((8) in Section 4), when the current projective camera center $\mathbf{t}_i$ is not the main anchor $\mathbf{t}_m$ of feature $\mathbf{F}_j$.

First, the sine law is used to compute the depth of feature $\mathbf{F}_j$ from the main anchor $\mathbf{t}_m$ and then the vector from $\mathbf{t}_i$ to feature $\mathbf{F}_j$ is computed (Figure 20).

A.1 Computation of φ_j

The vector from the main anchor $\mathbf{t}_m$ to the associate anchor $\mathbf{t}_a$ can be computed as the relative translation $\mathbf{t}_a - \mathbf{t}_m$. Thus, the angle from the vector $\mathbf{t}_a - \mathbf{t}_m$ to the vector $\mathbf{x}_j^m$ can be computed from the dot product between the two vectors $\mathbf{t}_a - \mathbf{t}_m$ and $\mathbf{x}_j^m$ by using (9) in Section 4.

A.2 Computation of d_j

The depth d_j of feature $\mathbf{F}_j$ from the main anchor $\mathbf{t}_m$ can be computed by using the sine law in the triangle bounded by $\mathbf{t}_m$, $\mathbf{t}_a$ and $\mathbf{F}_j$

$$d_j = \frac{\sin(\omega_j + \varphi_j)}{\sin \omega_j} \|\mathbf{t}_a - \mathbf{t}_m\|. \tag{48}$$

A.3 Computation of the Vector from $\mathbf{t}_i$ to Feature $\mathbf{F}_j$

The vector from $\mathbf{t}_i$ to feature $\mathbf{F}_j$ can be computed using

$$\mathbf{x}_j^i = d_j \mathbf{x}_j^m - (\mathbf{t}_i - \mathbf{t}_m). \tag{49}$$

The vector $\mathbf{x}_j^i$ computed by (49) can be directly used in the projective function. To avoid numerical errors caused by $\sin \omega_j$ in the denominator in d_j (48), and because different lengths of vector from pose to feature will not change the projective values of the feature in the images, the vector $\mathbf{x}_j^i$ is multiplied by $\sin \omega_j$. From (48) and (49),

$$\tilde{\mathbf{x}}_j^i = \sin \omega_j \, \mathbf{x}_j^i = \sin(\omega_j + \varphi_j) \, \|\mathbf{t}_a - \mathbf{t}_m\| \, \mathbf{x}_j^m - \sin \omega_j \, (\mathbf{t}_i - \mathbf{t}_m). \tag{50}$$

This is (8) in Section 4.

Acknowledgment

This work is supported in part by Australian Research Council (ARC) Linkage Project LP0884112 and ARC Discovery Project DP120102786. The authors would like to thanks the authors of g2o, sSBA and SBA for making their source code available which has significantly helped the research in this paper. The authors would also like to thank the editors and anonymous reviewers for their constructive comments on an earlier version of the manuscript.

References

Agarwal, S., Snavely, N., Seitz, S. M. and Szeliski, R. (2010). Bundle Adjustment in the Large. 11^{th} *European Conference on Computer Vision (ECCV)*, Heraklion, Greece, pp. 29-42.

Blanco, J. L., Gonzalez-Jimenez, J. and Fernandez-Madrigal, J. A. (2013). Sparser Relative Bundle Adjustment (SRBA): Constant-Time Maintenance and Local Optimization of Arbitrarily Large Maps. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 70-77.

Blanco, J. L., Moreno, F. A. and Gonzalez, J. (2009). A Collection of Outdoor Robotic Datasets with Centimeter-Accuracy Ground Truth. *Autonomous Robots*, 27: 327-351.

Civera, J., Davison, A. J. and Montiel, J. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5): 932-945.

Davison, A. J. (2003). Real-time Simultaneous Localisation and Mapping with a Single Camera. *International Conference on Computer Vision (ICCV)*, pp. 1403-1410.

Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6): 1052-1067.

Eade, E. and Drummond, T. (2009). Edge Landmarks in Monocular SLAM. *Image and Vision Computing*, 27(5): 588-596.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6): 381-395.

Hartley, R. and Zisserman, A. (2003). Multiple View Geometry in Computer Vision, 2nd Ed. *Cambridge University Press.*

Huang, S. and Dissanayake, G. (2007). Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM. *IEEE Transactions on Robotics*, 23(5): 1036-1049.

Huang, S., Wang, H., Frese, U., and Dissanayake, G. (2012). On the Number of Local Minima to the Point Feature Based SLAM Problem, *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, pp. 2074-2079.

Indelman, V., Melim, A. and Dellaert, F. (2013). Incremental Light Bundle Adjustment for Robotics Navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1952-1959.

Jeong, Y., Nister, D., Steedly, D., Szeliski, R. and Kweon, I. S. (2012). Pushing the Envelope of Modern Methods for Bundle Adjustment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8): 1605-1617.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K. and Burgard, W. (2011). G2O: A General Framework for Graph Optimization. *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, pp. 3607-3613.

Kwok, N. M. and Dissanayake, G. (2004). An Efficient Multiple Hypothesis Filter for Bearing-only SLAM. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, pp. 736-741.

Konolige, K. (2010). Sparse Sparse Bundle Adjustment. *British Machine Vision Conference (BMVC)*, pp. 1-11.

Lourakis, M. I. A. and Argyros, A. A. (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Transactions on Mathematical Software*, 36(1): 2:1-2:30.

Lowe, D. G. (2004). Distinctive Image Features from Scale Invariant Keypoints. *International Journal of Computer Vision*, 60(2): 91-110.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser F. and Sayd, P. (2009). Generic and Real Time Structure from Motion using Local Bundle Adjustment. *Image and Vision Computing*, 27(8): 1178-1193.

Ni, K., Steedly, D., and Dellaert, F. (2007). Out-of-core Bundle Adjustment for Large-scale 3D Reconstruction. *IEEE International Conference on Computer Vision (ICCV)*, pp. 1-8.

Pinies, P. and Tardos, J. D. (2008). Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision. *IEEE Transactions on Robotics*, 24(5): 1094-1106.

Sola, J., Monin, A., Devy, M. and Lemaire, T. (2005). Undelayed Initialization in Bearing Only SLAM. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, pp. 2499-2504.

Sola, J. (2010). Consistency of the Monocular EKF-SLAM Algorithm for Three Different Landmark Parametrizations. *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, pp. 3513-3518.

Strasdat, H., Montiel, J. and Davison, A. J. (2010). Real-time Monocular SLAM: Why Filter? *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, pp. 2657-2664.

Trawny, N. and Roumeliotis, S. I. (2006). A Unified Framework for Nearby and Distant Landmarks in Bearing-Only SLAM. *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, USA, pp. 1923-1929.

Triggs, B., McLauchlan, P. F., Hartley, R. I. and Fitzgibbon, A. W. (1999). Bundle Adjustment—A Modern Synthesis. *Vision Algorithms: Theory and Practice*, 1883/2000: 298-373.

Zhang, Z. (1997). Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. *Image and Vision Computing*, 15(1): 59-76.

Zhao, L., Huang, S., Yan, L., Wang, J., Hu, G. and Dissanayake, G. (2010). Large-Scale Monocular SLAM by Local Bundle Adjustment and Map Joining. 11^{th} *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, pp. 431-436.

Zhao, L., Huang, S., Yan, L. and Dissanayake, G. (2011). Parallax Angle Parametrization for Monocular SLAM. *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, pp. 3117-3124.

Zhao, L., Huang, S., Yan, L. and Dissanayake, G. (2014). A New Feature Parametrization for Monocular SLAM using Line Features. *Robotica*.