

# Parallax Angle Parametrization for Monocular SLAM

Liang Zhao, Shoudong Huang, Lei Yan, and Gamini Dissanayake

**Abstract**—This paper presents a new unified feature parametrization approach for monocular SLAM. The parametrization is based on the parallax angle and can reliably represent both nearby and distant features, as well as features in the direction of camera motion and features observed only once. A new bundle adjustment (BA) algorithm using the proposed parallax angle parametrization is developed and is shown to be more reliable as compared with existing BA algorithms that use Euclidean XYZ or inverse-depth parametrizations. A new map joining algorithm that allows combining a sequence of local maps generated using BA with the proposed parametrization, that avoids the large computational cost of a global BA, and can automatically optimize the relative scales of the local maps without any loss of information, is also presented. Large-scale simulation and a publicly available large-scale real dataset with centimeter accuracy ground truth are used to demonstrate the accuracy and consistency of the map joining technique using the new parametrization. Especially, since the relative scales are optimized automatically in the proposed BA and map joining algorithms, there is no need to compute any relative scales even for a loop more than 1km.

## I. INTRODUCTION

**S**IMULTANEOUS localization and mapping (SLAM) is the problem where a mobile robot needs to build a map of its environments and simultaneously use the map to locate itself. The monocular SLAM considered in this paper is the SLAM problem where the only sensor onboard the robot is a single camera, it is assumed that the robot can move arbitrarily in 3D environments and there is no odometry information available.

Because there is no range information available from a single camera, the 3D feature initialization in monocular SLAM becomes a challenge. When the parallax angle is small, traditionally XYZ parametrization tends to have large errors. The inverse-depth feature parametrization is becoming popular since it is more robust especially for far away features [1]. But it can still be inaccurate for close features with camera moving towards them. In the inverse-depth representation, each feature has a camera pose associated with it and was called anchor in [4]. In this paper,

we propose a new feature parametrization where parallax angle is a feature parameter with two camera poses as anchors. This parametrization can accurately describe all the different features under different situations.

Bundle adjustment (BA) has been the golden standard for monocular SLAM. It is more accurate and consistent as compared with filter based algorithm [7]. The proposed parallax angle feature parameterization is particularly suitable for BA where all the camera poses and all the features are used as the parameters of the optimization problem. In this paper, the BA algorithm using the new parametrization is implemented and compared with the existing BA algorithms. It is demonstrated that the new feature parametrization allow the BA to converge to global minimum easily.

Local map joining is an efficient strategy for solving large-scale SLAM problems. The map joining algorithm developed in [11, 12] achieved significant computation saving with no information loss. However, the local maps need to have the same scale before they can be joined together. In this paper, we develop a new map joining algorithm with local maps built by BA using the new parallax angle feature representation. The map joining algorithm can optimize the relative scales of the local maps during the optimization process.

This paper is organized as follows. Section II discusses the recent works related to this paper. Section III states parallax angle parametrization for features. Section IV details the bundle adjustment using this parallax angle parametrization. In Section V, initial value errors for different feature parametrization is analyzed. Section VI states the relative scale optimized local submap joining algorithm in short. In Section VII, some simulation and experimental results are provided. Finally Section VIII concludes the paper and Appendix details the observation function in Section IV.

## II. RELATED WORKS

There has been significant progress on monocular SLAM research in the past few years. Some of the work closely related to this paper is discussed below.

For feature parametrization, [1] created an inverse depth parametrization for feature in monocular SLAM. It uses the inverse of the depth from its anchor camera pose as one of the parameters and works very well for distant features. But if the feature is close to the camera and the camera is moving towards the feature, the initialization error will be large and the algorithm may fail (more detailed analysis is given in Section V with simulation results given in Section VI).

Manuscript received September 15, 2010.

Liang Zhao and Lei Yan are with the Institute of Remote Sensing and GIS, School of Earth and Space Science, Peking University, Beijing China, 100871 (e-mail: {lzhao, lyan}@pku.edu.cn).

Shoudong Huang and Gamini Dissanayake are with the Australian Research Council (ARC) Centre of Excellence for Autonomous Systems (CAS), Faculty of Engineering, University of Technology, Sydney, NSW 2007, Australia (e-mail: {sdhuang, gdissa}@eng.uts.edu.au).

Liang Zhao is now with the CAS as a visiting research student supported by China Scholarship Council (corresponding author phone: +61-2-9514 3141; fax: +61-2-95142655; e-mail: lzhao@eng.uts.edu.au).

Sola [4] compared a few different feature parametrization approaches. Especially, a number of anchored feature parametrization appears to be better than no-anchored feature parametrization in the Extended Kalman Filter (EKF) framework. Our parallax angle parametrization belongs to the anchored one but we use two anchors instead of one anchor for each 3D feature.

For a simpler 2D bearing-only SLAM problem, a unified framework for both nearby and distant landmarks is provided in [5]. It uses a global bearing angle from an artificial position to parametrize features and there will be redundant parameters involved if it is extended to 3D and thus constraints need to be applied. As compared with the delayed feature initialization in [6], our new feature parametrization allows the feature to be initialized whenever it is observed.

For all the above mentioned papers, EKF instead of optimization methods were used in the estimation. BA is an optimization based approach for monocular SLAM. Some systematic and comprehensive comparison between BA and EKF are given in [7] and it was shown that BA outperforms EKF SLAM in terms of consistency and accuracy. In our current work, a new BA algorithm is developed where the features are represented using the parallax angle feature parametrization.

Local map joining has been shown to be one of the efficient strategies for large-scale SLAM. In [9], a real-time visual mapping method for large-scale SLAM problem was developed. It first uses local BA to get the relative pose information between key frames; then an optimization is performed to optimize the key frames. Some good results are demonstrated for SLAM using stereo vision but may be non-trivial to be applied to monocular SLAM where scale drift can have a significant impact on the performance. Furthermore, the deleting of common features can introduce significant information loss. Local map joining has also been applied to monocular SLAM in [10], where conditionally independent local maps are built by EKF and then carefully combined together to avoid information reuse. Constant velocity camera motion model is assumed when building the local maps with EKF and the scales of local maps need to be relatively accurate. In this paper, we use BA with the new parametrization to build high quality, statistically independent local maps and then combine the local map together without information loss. The relative scales between local maps are not needed before the map joining since they can be optimized automatically through the map joining process.

Very recently, a scale drift-aware large scale monocular SLAM algorithm was proposed in [13]. It uses split window BA to get the locally optimized poses and a separate algorithm is used to optimize the relative scales when loop closure is detected. Then a structure only BA is applied. Near real-time performance is demonstrated using a camera looking sideways. In our current work, common features in local maps are kept in the global map in order to optimize the

poses, features and the scale factors automatically in the map joining process. Because in monocular SLAM the relative scale information only comes from the common features, high performance can be achieved at the cost of an optimization process of the selected poses and features.

### III. PARALLAX ANGLE FEATURE PARAMETRIZATION

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

#### A. Camera Pose Parametrization

A camera pose is represented by rotation angles and translation vector relative to the first camera pose,  $Pose_0$ .

The  $i^{th}$  camera pose is:

$$Pose_i = [\varphi_{p_i}, \omega_{p_i}, \kappa_{p_i}, X_{p_i}, Y_{p_i}, Z_{p_i}]^T$$

where  $[\varphi_{p_i}, \omega_{p_i}, \kappa_{p_i}]^T$  are the Yaw, Pitch, Roll angles of  $Pose_i$  and  $[X_{p_i}, Y_{p_i}, Z_{p_i}]^T$  is the translation vector from  $Pose_0$  to  $Pose_i$ , where  $Pose_0 = [0, 0, 0, 0, 0, 0]^T$ .

#### B. Parametrization for Non-depth Feature

We divide the point features into two classes. One is non-depth feature meaning that the feature is observed only once. The other is 3D feature meaning that the feature is observed at least twice.

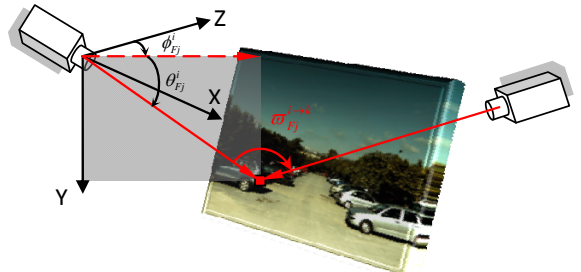


Fig. 1. Parametrization for non-depth and 3D feature.

Introducing the non-depth feature is important for the local map joining scenarios, for example, a feature may be observed once in local map 1 and then observed again in local map 2. The feature will be initialized as a non-depth feature in local map 1 and then changed into 3D feature after joining local maps 1 and 2. Similarly a feature may be observed only once at the beginning until it is observed again when closing the loop. With non-depth feature involved, there is no information loss in the mapping process. Thus this avoids the need of building and joining conditional independent local maps as described in [10]. Furthermore, the feature can be initialized whenever it is first observed without any delay.

Suppose feature  $F_j$  is only observed at  $Pose_i$ , we define the feature as a non-depth feature and define  $Pose_i$  as the main anchor of  $F_j$ . The feature is described by:

$$F_j = [\phi_{F_j}^i, \theta_{F_j}^i]^T$$

where  $\phi_{F_j}^i$  and  $\theta_{F_j}^i$  are the azimuth and elevation angles, so the vector  $[\phi_{F_j}^i, \theta_{F_j}^i]^T$  represents the direction from the main anchor to  $F_j$  in  $Pose_0$  (Figure 1).

This is the same as inverse-depth feature parametrization but without the depth information [1].

### C. Parallax Angle Parametrization for 3D features

When feature  $F_j$  is observed two times or more, we define the feature as a 3D feature with 2 anchors. Suppose the main anchor of feature  $F_j$  is  $Pose_i$  and the associated anchor is  $Pose_k$ , then  $F_j$  is described using the azimuth and elevation angles together with the parallax angle  $\varpi_{F_j}^{m \rightarrow a} = \varpi_{F_j}^{i \rightarrow k}$  as follows:

$$F_j = [\phi_{F_j}^i, \theta_{F_j}^i, \varpi_{F_j}^{i \rightarrow k}]^T.$$

The adding parameter, parallax angle  $\varpi_{F_j}^{i \rightarrow k}$  is the angle from  $\bar{X}_{F_j}^i$  to  $\bar{X}_{F_j}^k$ , where  $\bar{X}_{F_j}^i$  and  $\bar{X}_{F_j}^k$  are the vectors from the main anchor  $Pose_i$  and the associated anchor  $Pose_k$  to  $F_j$  (Figure 1).

The key difference between this parametrization and the inverse-depth parametrization is that there is no scale information involved in the three parameters and we use two anchors instead of one anchor [1].

### D. A strategy for choosing the anchors

A simple way is to define the main anchor as the pose where  $F_j$  is first observed, and define the associated anchor as the pose where  $F_j$  is observed the second time. However, in this paper, when the feature is observed more than two times, we will compare the new parallax angles with the one used in the feature parameter and decide whether to change the anchors or not. We will choose the anchor poses such that the parallax angle is not too small (if possible).

Whether the anchors changing or not will not affect the convergence and the result of BA. But we change the anchors in this way to make the location more accurate if one wants to transform the 3D feature to XYZ representation.

## IV. BUNDLE ADJUSTMENT USING PARALLAX ANGLE PARAMETRIZATION

While this new feature parametrization can be applied to EKF based approach, the computational cost will be significantly increased due to the involvement of the two anchored poses for each feature. On the other hand, no additional computational cost is introduced in the optimization based approach such as bundle adjustment (BA) where all the camera poses are part of the parameters.

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 initialization of poses and features are briefly outlined. Some detailed derivations are given in Appendix for clarity.

### A. Observation Function for BA

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

Suppose the main and associate anchors of feature  $F_j$  are  $Pose_i$  and  $Pose_k$  respectively. The image coordinates of  $F_j$  at  $Pose_l$  can be written as:

$$\begin{bmatrix} u_j^l \\ v_j^l \end{bmatrix} = \begin{bmatrix} x_j^l / t_j^l \\ y_j^l / t_j^l \end{bmatrix}$$

$$\text{where } \begin{bmatrix} x_j^l \\ y_j^l \\ t_j^l \end{bmatrix} = \begin{cases} K R_{pl} \bar{X}_{F_j}^i \text{Unit}, & \text{if } l = i, \\ K R_{pl} \sin \varpi_{F_j}^{i \rightarrow k} \bar{X}_{F_j}^l, & \text{if } l \neq i. \end{cases} \quad (1)$$

Here  $K$  is the calibration matrix and  $R_{pl}(\phi_{pl}, \omega_{pl}, \kappa_{pl})$  is the rotation matrix of  $Pose_l$ .  $\bar{X}_{F_j}^i \text{Unit}$  is the unit vector from the main anchor to feature  $F_j$  given by

$$\bar{X}_{F_j}^i \text{Unit} = \begin{bmatrix} X_{F_j}^i \text{Unit} \\ Y_{F_j}^i \text{Unit} \\ Z_{F_j}^i \text{Unit} \end{bmatrix} = \begin{bmatrix} \sin \phi_{F_j}^i \cos \theta_{F_j}^i \\ \sin \theta_{F_j}^i \\ \cos \phi_{F_j}^i \cos \theta_{F_j}^i \end{bmatrix}$$

$\bar{X}_{F_j}^l$  is a function of  $Pose_i$ ,  $Pose_k$ ,  $Pose_l$  and  $F_j$  which represent the vector from  $Pose_l$  to  $F_j$ . The detailed equation for  $\bar{X}_{F_j}^l$  and the derivation of the remainder of the observation function is given in the Appendix.

### B. Least Squares Optimization

Let  $X$  be the measurement vector containing image coordinates of all the features in each image, let  $P$  be the parameter vector containing all the poses and features using the proposed parametrization, and let  $f(P)$  be the observation function defined above. The least squares optimization problem in BA is to seek the vector  $\hat{P}$  such that

$$\|\varepsilon\|_{\Sigma_X}^2 = (f(\hat{P}) - X)^T \Sigma_X^{-1} (f(\hat{P}) - X)$$

is minimized. We suppose the uncertainty of the image coordinates of all the features are the same, so  $\Sigma_X^{-1} = I$  is used here for BA.

### C. Parameter vector initialization

The  $l^{\text{th}}$  camera pose,  $Pose_l = [\phi_{pl}, \omega_{pl}, \kappa_{pl}, X_{pl}, Y_{pl}, Z_{pl}]^T$ , can be initialized as

$$\begin{cases} R_{p_l}(\varphi_{p_l}, \omega_{p_l}, \kappa_{p_l}) = \prod_{n=1}^l \Delta R_{n-1} \\ [X_{p_l}, Y_{p_l}, Z_{p_l}]^T = \sum_{n=1}^l R_{p_n}^T [\Delta X_{n-1}, \Delta Y_{n-1}, \Delta Z_{n-1}]^T \end{cases}$$

where  $\Delta R_{n-1}^n$  is the relative rotation matrix and  $[\Delta X_{n-1}, \Delta Y_{n-1}, \Delta Z_{n-1}]^T$  is the relative translation between two nearby images computed using two-view geometry [16]. For initialization, the relative scales between relative poses are supposed to be the same.

If feature  $F_j$  is observed only once at  $Pose_i$ , then  $Pose_i$  is its main anchor and  $F_j$  can be initialized as

$$\begin{cases} \phi_{F_j}^i = \text{atan2}(X_{F_j}^i / Z_{F_j}^i) \\ \theta_{F_j}^i = \text{atan2}(Y_{F_j}^i / \sqrt{X_{F_j}^{i,2} + Z_{F_j}^{i,2}}) \end{cases}$$

where  $\begin{bmatrix} X_{F_j}^i \\ Y_{F_j}^i \\ Z_{F_j}^i \end{bmatrix} = (K R_{p_i})^{-1} \begin{bmatrix} u_j^i \\ v_j^i \\ 1 \end{bmatrix}$ .

$R_{p_i}$  is the rotation matrix of  $Pose_i$ ,  $[u_j^i, v_j^i]$  is the image coordinates of feature  $F_j$  at  $Pose_i$  and  $\vec{X}_{F_j}^i = [X_{F_j}^i, Y_{F_j}^i, Z_{F_j}^i]^T$  is the vector from  $Pose_i$  to feature  $F_j$ .

If  $F_j$  is observed at least twice with main anchor  $Pose_i$  and associated anchor  $Pose_k$ , then the adding parallax angle  $\varpi_{F_j}^{i \rightarrow k}$  can be initialized as

$$\varpi_{F_j}^{i \rightarrow k} = \arccos \left( \frac{\vec{X}_{F_j}^i \cdot \vec{X}_{F_j}^k}{\|\vec{X}_{F_j}^i\| \|\vec{X}_{F_j}^k\|} \right),$$

where  $\vec{X}_{F_j}^k$  is the vector from  $Pose_k$  to feature  $F_j$ .

## V. ANALYSIS ON INITIAL VALUE ERRORS FOR DIFFERENT FEATURE PARAMETRIZATIONS

One key property of the proposed parallax angle parametrization is that the feature initialization error only depends on the error of the observed image coordinates. This is discussed and compared with the characteristics of alternative parametrizations. For simplicity, this analysis is only demonstrated for the 2D scenario.

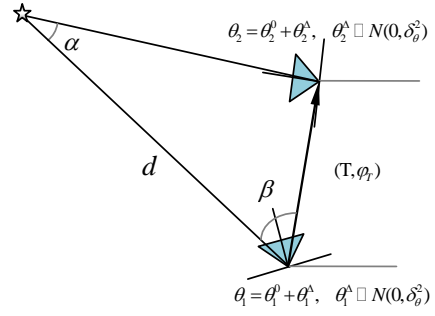


Fig. 2. Initial value errors analysis for the 2D bearing only case.

Consider a 2D bearing only problem where a feature is observed from two camera poses as shown in Figure 2. The orientations of the first and second camera poses are denoted as  $\varphi_1$  and  $\varphi_2$  respectively, and the bearing angles are denoted as  $\theta_1$  and  $\theta_2$ . Let the true bearings be  $\theta_1^0$  and  $\theta_2^0$ , and the errors of the observation angles are zero mean Gaussian  $\theta_1^A \sim N(0, \delta_{\theta_1}^2)$  and  $\theta_2^A \sim N(0, \delta_{\theta_2}^2)$ . The orientation and the distance of relative translation between the two camera poses are  $\varphi_T$  and  $T$ .

### A. Parallax angle parametrization

$\theta_1 + \varphi_1$  and  $\theta_2 + \varphi_2$  are the global bearing angles of feature from the two camera poses. Suppose  $\alpha_0$  is the true parallax angle and  $\alpha$  is the computed parallax angle, then

$$\alpha_0 = (\theta_2^0 + \varphi_2) - (\theta_1^0 + \varphi_1),$$

$$\alpha = (\theta_2 + \varphi_2) - (\theta_1 + \varphi_1) = \alpha_0 + \theta_2^A - \theta_1^A, \text{ so we have}$$

$$\frac{\partial \alpha}{\partial \theta_1^A} = -1 \text{ and } \frac{\partial \alpha}{\partial \theta_2^A} = 1$$

Thus the error of the parallax angle only depends on 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  $\theta_1^A, \theta_2^A$  are small.

### B. XYZ parametrization

Suppose  $\beta$  and  $\beta_0$  are the computed and true angles from the relative translation to the first feature observation, then

$$\beta_0 = \theta_1^0 + \varphi_1 - \varphi_T,$$

$$\beta = \theta_1 + \varphi_1 - \varphi_T = \beta_0 + \theta_1^A,$$

$$\text{From the sine law, } d_0 = \frac{T \sin(\alpha_0 + \beta_0)}{\sin \alpha_0}$$

$$d = \frac{T \sin(\alpha + \beta)}{\sin \alpha} = \frac{T \sin(\alpha_0 + \beta_0 + \theta_2^A)}{\sin(\alpha_0 + \theta_2^A - \theta_1^A)}. \text{ So we have}$$

$$\frac{\partial d}{\partial \theta_1^A} = \frac{\sin(\alpha_0 + \beta_0 + \theta_2^A) \cos(\alpha_0 + \theta_2^A - \theta_1^A)}{\sin^2(\alpha_0 + \theta_2^A - \theta_1^A)} T$$

$$\frac{\partial d}{\partial \theta_2^A} = \frac{-\sin(\beta_0 + \theta_1^A)}{\sin^2(\alpha_0 + \theta_2^A - \theta_1^A)} T$$

When the parallax angle  $\alpha_0$  is close to zero, the error of

the initial value of feature depth will be very large. The large error in depth can result in a large error in XYZ.

### C. Inverse depth parametrization

Inverse depth parametrization is an improvement of XYZ parametrization, particularly useful for distant features.

$$\rho = \frac{1}{d}, \text{ so } \rho = \frac{\sin(\alpha_0 + \theta_2^\Delta - \theta_1^\Delta)}{T \sin(\alpha_0 + \beta_0 + \theta_2^\Delta)},$$

$$\frac{\partial \rho}{\partial \theta_1^\Delta} = \frac{-\cos(\alpha_0 + \theta_2^\Delta - \theta_1^\Delta)}{T \sin(\alpha_0 + \beta_0 + \theta_2^\Delta)}$$

$$\frac{\partial \rho}{\partial \theta_2^\Delta} = \frac{\sin(\beta_0 + \theta_1^\Delta)}{T \sin^2(\alpha_0 + \beta_0 + \theta_2^\Delta)}$$

It can be seen that when the camera is moving towards the feature, the angle  $\alpha_0 + \beta_0$  will be close to zero and the error of initial value of  $\rho$  will be large. The simulation results in Section VI-A confirm this analysis.

## VI. RELATIVE SCALE OPTIMIZED LOCAL SUBMAP JOINING ALGORITHM

Although the proposed BA algorithm using parallax angle parametrization can provide an accurate solution for monocular SLAM, as with BA using XYZ parametrization, it is computationally intractable for very large-scale problems.

Local map joining has shown to be an efficient strategy for large-scale SLAM. This section will briefly describe the new map joining algorithm that is used to join the local maps built by BA using the new parametrization. The details of the algorithm are omitted due to the space limitation.

### A. Local Map Building using BA with Parallax angle parametrization

First the original data are divided into groups and used to build local maps using BA with the new parametrization. The grouping can be selected so that the resulting optimization problem converges within an acceptable time period. As the initial values of the parameter vector are quite accurate and there are a large number of features observed from different poses, BA converges very fast with all the pose translations **up to one scale**.

### B. Deleting Features and Poses from Local Maps

Only parts of the local map are kept during map joining because only these parts contribute to the optimization of the global map:

- Common features appear in at least two local maps;
- The end pose of each local map;
- Translations of the second pose of each local map and all the poses served as the anchors of the kept features;
- Information matrix corresponding to all the above variables (computed using Schur complement).

In practice this removes more than 90% of the features and 50% of the pose parameters, which reduces the computation cost greatly without any impact on the quality of the global map.

### C. Relative scale optimized using least squares

After the features and pose elements are selected from the local maps, a least squares optimization is performed similarly to [12]. Because the common features among local maps are kept, the relative scales between local maps are optimized during the optimization process. So the final global map is accurate up to one scale.

## VII. SIMULATION AND EXPERIMENTAL RESULTS

Simulation and real datasets have been used to check the validity and accuracy of the BA and map joining algorithms using the new parametrization.

### A. Simulation results

In all the simulations, the camera is modeled as  $[-\pi/4, \pi/4]$  of FOV,  $[0, +\infty)$  observation distance,  $800 \times 800$  image,  $[400, 400]$  principle point and  $[400, 400]$  focal length. A random Gaussian noise with  $\sigma = 0.1$  is added on the true image coordinates as the observations of features.

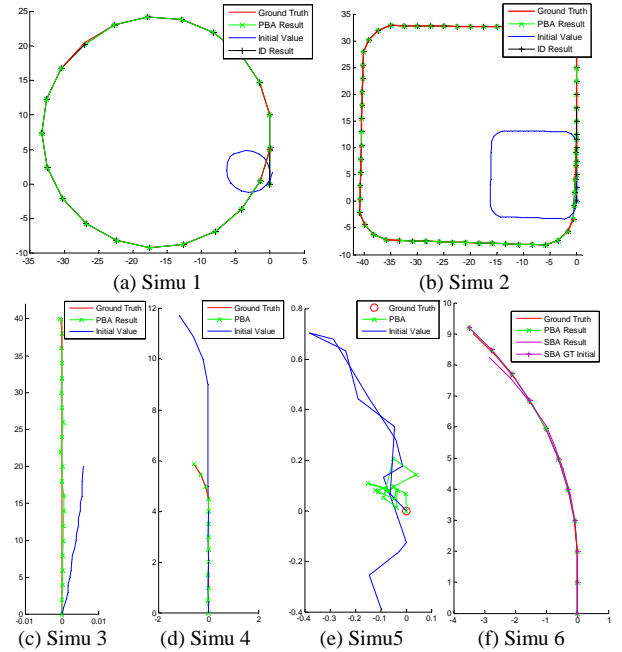


Fig. 3. BA results of Simulation 1-6. Simu 1-5 compare the BA with different parametrizations using only Gauss-Newton method. Simu 6 compares BA with parallax angle and SBA. BA with parallax angle converged in Simu 1-5, BA with XYZ diverged in Simu 1-5, BA with inverse depth converged in Simu 1-2 but diverged in Simu 3-5. In Simu 6, SBA converged to a local minimum but BA with parallax angle converged to the global minimum. When ground truth of features and poses are used as the initial value for SBA, it converged to the same result as BA with parallax angle.

### a. Comparison of BA using different feature parametrization with Gauss-Newton algorithm

For solving the least squares problem involved in BA, the Levenberg-Marquardt (LM) method is commonly used [14]. LM has a lot of advantages as compared with the simple Gauss-Newton algorithm. However, there are many different versions of LM implementations and there are quite a few parameters involved in the LM algorithm that will

significantly influence the convergence and performance of the algorithms. In order to have a fair comparison among the effects of different feature parametrizations for BA, we thus choose to use only Gauss-Newton iterations with the same convergence criteria. We believe that if an algorithm can converge to global minimum with a simple Gauss-Newton algorithm, then the algorithm must be a robust one.

For the Simulation 1-5, three BA algorithms using different parametrization are compared. They are

- XYZ parametrization;
- Inverse depth parametrization;
- Parallax angle parametrization.

Simulation 1 and 2 simulate the regular conditions with 23 poses circular trajectory and 66 poses square trajectory with both nearby and distant features. The features are assumed to be uniformly distributed in the  $[-5000\sim 5000; -5000\sim 5000; -5\sim 5]$  cuboid with resolution  $[10; 10; 2]$  and also in the  $[-25\sim 25; -25\sim 25; -5\sim 5]$  cuboid with resolution  $[20; 20; 4]$ . For these two simulations, BA with XYZ parametrization diverged, and BA with inverse depth parametrization and parallax angle parametrization converged. The results are shown in Figure 3 (a-b). So both inverse depth and parallax angle parametrization can deal with the distant features very well, but XYZ parametrization cannot.

To check whether inverse-depth can deal with arbitrary scenarios, we design the Simulation 3-5 with some extreme conditions. Simulation 3 is the condition that the camera moves straight towards features with 21 steps. Simulation 4 is the condition that the camera goes straight towards features for 9 steps and then starts to turn when it is very close to the feature. Simulation 5 is a more extreme case that the camera does not move but only turn around: no motion at steps 1-2, turning  $0.3\text{rad}$  at steps 3-7 and  $-0.2\text{rad}$  at steps 8-16. In these 3 simulations, BA with inverse depth parametrization diverged but BA with parallax angle parametrization converged. The results are shown in figure 3 (c-e).

### b. Compare the parallax angle BA with SBA

For XYZ BA, there is a robust implementation available called Sparse Bundle Adjustment (SBA) [14]. SBA is a publicly available C/C++ software package for realizing generic BA with high efficiency and flexibility regarding parametrization.

In this section, we would like to see how our BA with parallax angle parametrization performs as compared with SBA. For Simulations 1-5, the final mean squares errors of SBA and BA with parallax angle parametrization are listed in Table I.

We further design another extreme scenario for testing SBA. In Simulation 6, the environment is the same as Simulation 1&2 with 11 poses. But only four close features and about 300 distant features in each image. The trajectories and the mean square errors are shown in Figure 3 (f) and Table I. Here we also use the ground true poses and features as the initial value of SBA to check whether SBA has been trapped in local minimum. We can see from Figure 3 (f) and

Table I that, SBA converged to local minimum because of the inaccurate initial value.

Same thing happens in Simulations 1 and 2 as shown in the first two rows in Table I. In these three simulations, there are many distant features. This proves again that XYZ parametrization cannot deal with distant features very well. BA with parallax angle parametrization does not have this issue as analyzed in Section V.

TABLE I  
MEAN SQUARE ERRORS OF SIMULATION 1-6

Simulation	BA with Parallax Angle		SBA	
	Initial	Final	Initial	Final
1	2922.05506	<b>0.00484460</b>	116071	<b>0.00701422</b>
2	9.76708967	<b>0.00585854</b>	126091	<b>0.0114854</b>
3	7.56421243	0.00574064	2524.38	0.00574053
4	0.06028423	0.00562767	3248.64	0.00562767
5	14.8777296	0.00520672	0.0280551	0.00549258
6	0.04373461	<b>0.00525359</b>	905.799	<b>0.0112368</b>
	GT as initial value		0.00672434	<b>0.00525461</b>

### c. Consistency check on the map joining result

A large scale simulation with 2 loops has been designed to check both BA and map joining proposed in this paper. The environment has features uniformly distributed in cuboid  $[-500\sim 500; -500\sim 500; -5\sim 5]$  with resolution  $[40; 40; 3]$ . The trajectory is similar with the real dataset with one square-shape loop and one rectangle-shape loop. There are 501 poses and the step distance of camera is 5m multiple a random scales from 0.8 to 1.2 to simulate the relative scales between poses. The simulation environment, trajectory and observations are shown in Figure 4.

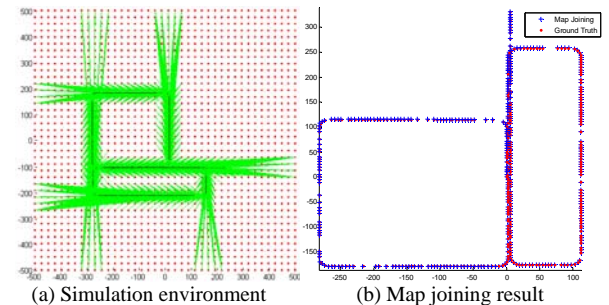


Fig. 4. Large scale simulations and map joining result.

The 501 images are divided into 25 groups and BA with parallax angle parametrization is used to build 25 local maps. Then the scale optimized map joining algorithm is used to build the global map.

TABLE II  
CONSISTENCY OF MAP JOINING RESULT BY NEES CHECK

Run	1	2	3	4	5
Dimensions	1214	1229	1220	1193	1211
Gate (95%)	1296.2	1331.7	1302.4	1274.5	1293.1
NEES	1174.1	1192.6	1290.4	1113.6	1151.4

Five simulations are run each with different random seeds for the observation noises. The kept pose translations in the final global map are used to check the consistency of map joining algorithm. The result is shown in Table II. We can see that all the results are consistent. Because the anchors of features may be different in different runs, the numbers of kept translations are not the same.

## B. Results using real experimental datasets

For the experimental results, we use the public available Málaga 2009 Robotic Dataset Collection [17]. This dataset was collected using an electric car equipped with laser scanners, cameras, IMU, and GPS receivers. As described in [17], a centimeter-level ground truth is provided which makes the dataset an ideal test bed for SLAM.

In this experiment we use the images captured by the right camera. The camera calibration parameters can be found in the dataset. SIFT descriptor [18] is used in both feature detection matching and loop closure detection. A Multi-level RANSAC [19] with thresholds as 2, 0.5, 0.1 and 0.05 has been used to remove the outliers.

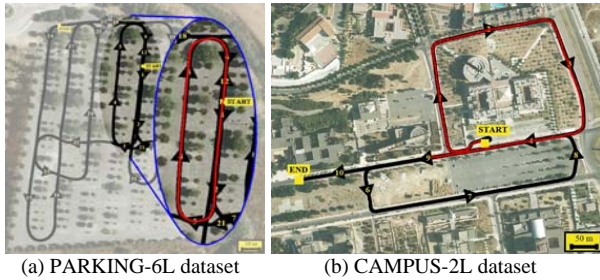


Fig. 5. The trajectories of the using two real datasets.

### a. Small loop of PARKING-6L dataset

We select one sequence of images collected from a 250m close loop trajectory. The original framerate is 7.5Hz. In this experiment we use 2.5 Hz as the framerate to select 170 from the 510 images. It is the same dataset used in [8].

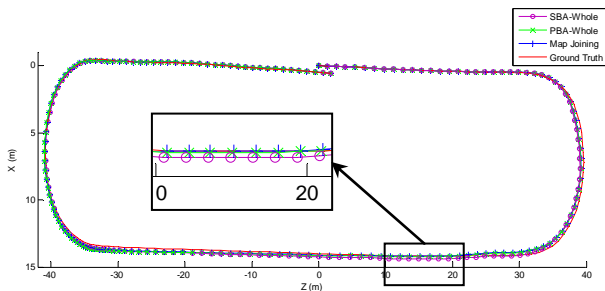


Fig. 6. SBA, BA using parallax angle parametrization and Map Joining results of PARKING-6L dataset.

There are 170 poses and 58404 features in total. The whole BA using parallax angle parametrization takes 14121.63 seconds and SBA takes 14931.69 seconds. The results of BA using parallax angle parametrization and SBA are almost the same. The final mean square errors of the proposed BA and SBA are 0.1092088 and 0.14144, respectively.

Then the 170 images are divided into 10 groups to build local maps. Each local map contains about 6K features. The BA of each local map costs about 20s to 50s. Deleting features, poses and computing information matrix of each local map takes about 3s to 10s. The map joining takes 872.22 seconds to build the global map including 159 poses or translations and 4436 features. The total time of local map building plus the map joining is 1316.32 seconds.

The ground truth, the results of SBA, BA using parallax angle parametrization, and the map joining are all shown in

Figure 6. All the three results are very close to the ground truth. Especially, results of map joining and BA using parallax angle are almost identical.

### b. Large loop of CAMPUS-2L dataset

A larger dataset has been used to further test the proposed algorithms. There are two loops in the whole CAMPUS-2L dataset. Here we only use the first loop (nearly square one) because there is no ground truth for the second one. The loop is about 1km long. The original framerate is also 7.5Hz and we use 1.5 Hz as the framerate to select 501 from 2505 images. There are 501 poses and 98668 features in total after SIFT matching and RANSAC.

Ten local maps are built by BA using parallax angle parametrization and then joined together using the proposed map joining algorithm. Each local map contains about 10k features. The BA of each local map costs about 27s to 100s. Map joining takes 730.26 seconds to build the global map containing 279 poses or translations and 4214 features. The total time of local map building plus the map joining is 1326.76 second. The available ground truth and the kept poses in the map joining result are shown in Figure 7.

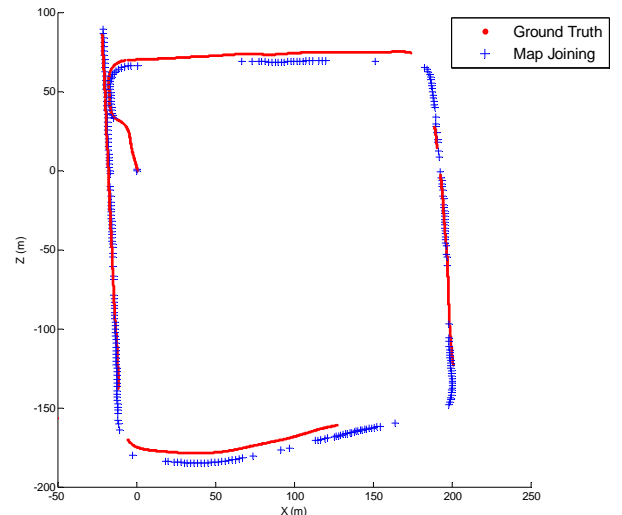


Fig. 7. Map joining result of CAMPUS-2L dataset.

## VIII. CONCLUSION

In this paper we have presented a unified feature parametrization for monocular SLAM. It uses the natural parallax angle as a parameter for 3D features and provides a theoretically rigorous presentation that suits for both nearby and distant features and can always provide accurate feature initialization no matter how the camera is moved. Since monocular SLAM can be formulated as an optimization problem, we believe this feature parameterization will have significant impact in the visual SLAM community since it can always guarantee accurate initial values.

Both bundle adjustment (BA) and local map joining algorithms have been developed using the proposed new feature parametrization. The BA using the proposed parametrization can converge to global minimum more easily as compared with BA using other parameterization. The

algorithms can optimize the camera poses, feature positions and the relative scales in one go. Simulation and experimental results demonstrate the high quality of the algorithms as compared with the ground truth.

The computational cost of the proposed map joining algorithm can be reduced further by reducing the number of features and the number of poses involved. However, the tradeoff between speed and performance (especially the ability to optimize the relative scales) needs further investigation.

This work makes use of SIFT for feature matching and RANSAC for outlier removal, a more efficient feature detection and matching algorithm especially for loop closing will also significantly reduce the overall computational cost.

## APPENDIX

In this appendix, we describe the formula of  $\vec{X}_{F_j}^l$  and the derivation of the second equation in observation function (1) in Section IV.

We first use the sine law to compute the depth of feature  $F_j$  from the main anchor  $Pose_i$ . Then compute the vector from  $Pose_i$  to feature  $F_j$  (see Figure 8).

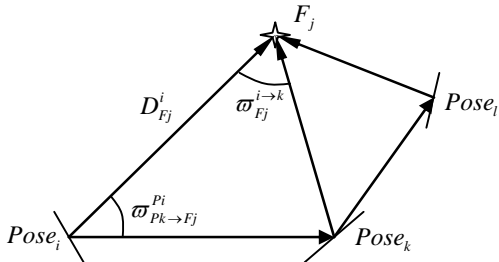


Fig. 8. Computation of vector from different pose to feature.

### A. Computation of $\varpi_{pk \to F_j}^{pi}$

The vector from the main anchor  $Pose_i$  to the associate anchor  $Pose_k$  can be computed as the relative translation  $T_{pk} - T_{pi}$ . So the angle  $\varpi_{pk \to F_j}^{pi}$  from the vector  $T_{pk} - T_{pi}$  to the vector  $\vec{X}_{F_j}^i Unit$  can be computed from the dot product between the two vectors  $T_{pk} - T_{pi}$  and  $\vec{X}_{F_j}^i Unit$  by

$$\varpi_{pk \to F_j}^{pi} = \arccos \left( \vec{X}_{F_j}^i Unit \cdot \frac{T_{pk} - T_{pi}}{\|T_{pk} - T_{pi}\|} \right)$$

### B. Computation of $D_{F_j}^i$

Suppose  $D_{F_j}^i$  is the depth of feature  $F_j$  from the main anchor  $Pose_i$ . It can be computed by using the sine law in the triangle structured by  $Pose_i$ ,  $Pose_k$  and  $F_j$ .

$$D_{F_j}^i = \frac{\sin(\varpi_{F_j}^{i \to k} + \varpi_{pk \to F_j}^{pi})}{\sin \varpi_{F_j}^{i \to k}} \|T_{pk} - T_{pi}\|$$

### C. Computation of the vector from $Pose_i$ to feature $F_j$

The vector  $\vec{X}_{F_j}^l$  from  $Pose_i$  to feature  $F_j$  can be computed by using vector computation.

$$\vec{X}_{F_j}^l = D_{F_j}^i \vec{X}_{F_j}^i Unit - (T_{pi} - T_{pi})$$

### D. Computation of the homogenous presentation of the projective of feature $F_j$

We can directly use the vector  $\vec{X}_{F_j}^l$  in the projective function. However, to avoid the numerical error caused by  $\sin \varpi_{F_j}^{i \to k}$  on the denominator in  $D_{F_j}^i$ , and because different depth of vector from pose to feature will not change the projective values of the feature in the images, we multiply the vector  $\vec{X}_{F_j}^l$  by  $\sin \varpi_{F_j}^{i \to k}$ . So the projective function can be written as

$$\begin{bmatrix} x_j^l \\ y_j^l \\ t_j^l \end{bmatrix} = K R_{pi} \sin \varpi_{F_j}^{i \to k} \vec{X}_{F_j}^l$$

This is the second formula in equation (1).

## REFERENCES

- [1] J. Civera, A.J. Davison, J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, Iss. 5, pp. 932-945, October 2008.
- [2] J. Sola, "Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations," In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, pp. 3513-3518, May 2010.
- [3] N. Trawny, S.I. Roumeliotis, "A Unified Framework for Nearby and Distant Landmarks in Bearing-Only SLAM," In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, USA, pp. 1923-1929, May 2006.
- [4] T. Bailey, "Constrained Initialisation for Bearing-Only SLAM," In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, pp. 1966-1971, November, 2003.
- [5] H. Strasdat, J. Montiel and A. J. Davison, "Real-time Monocular SLAM: Why Filter?" In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, pp. 2657-2664, May 2010.
- [6] K. Konolige, M. Agrawal, "FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping," *IEEE Transactions on Robotics*, vol. 24, Iss. 5, pp. 1066-1077, October 2008.
- [7] P. Pinies, J. D. Tardos, "Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision," *IEEE Transactions on Robotics*, vol. 24, Iss. 5, pp. 1094-1106, October 2008.
- [8] S. Huang, Z. Wang, G. Dissanayake, "Sparse Local Submap Joining Filter for Building Large-Scale Maps," *IEEE Transactions on Robotics*, vol. 24, Iss. 5, pp. 1121-1130, October 2008.
- [9] G. Hu, S. Huang, and G. Dissanayake, "3D I-SLSJF: A Consistent Sparse Local Submap Joining Algorithm for Building Large-Scale 3D Maps," In *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, pp. 6040 - 6045, 2009.
- [10] H. Strasdat, J. Montiel and A. J. Davison, "Scale Drift-Aware Large Scale Monocular SLAM," In *Proceedings of the Robotics: Science and Systems Conference (RSS)*, 2010.
- [11] M. I. A. Lourakis, and A. A. Argyros, SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Softw.* 36, 1, Article 2 (March 2009), 30 pages. DOI =10.1145/1486525.1486527 <http://doi.acm.org/10.1145/1486525.1486527>, 2009.

- [12] S. Huang and G. Dissanayake, "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM," *IEEE Transactions on Robotics*, vol. 23, Iss. 5, pp. 1036-1049, 2007.
- [13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, 2nd ed.*, Cambridge University Press, 2003, pp. 237-323.
- [14] J. L. Blanco, F. A. Moreno, J. Gonzalez, "A Collection of Outdoor Robotic Datasets with Centimeter-Accuracy Ground Truth," *Autonomous Robots*, vol. 27, pp. 327-351, August 2009.
- [15] D. G. Lowe, "Distinctive Image Features from Scale invariant Keypoints," *International Journal of Computer Vision*, vol. 60(2), pp. 91-110, 2004.
- [16] M. Pischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Common Assoc. Comp. Mach.*, vol. 24, pp. 381-95, 1981.