

Online Active Calibration for a Multi-LRF System

Guoyang Xie^{‡*}, Tao Xu^{*}, Carsten Isert^{*}, Micheal Aeberhard[†], Shaohua Li[‡], Ming Liu[‡]

^{*}BMW China Services Ltd.

{guoyang.xie@partner.bmw.com, tomas.xu@bmw.com, carsten.isert@bmw.de}

[†]BMW Group Research and Technology

{micheal.aeberhard@bmw.de}

[‡]Hong Kong University of Science and Technology

{gxieaa, slias, eelium}@ust.hk

Abstract—Multi-LRF(Laser Range Finder) systems have been broadly utilized in sensor fusion for automobile. In order to convert multiple LRF data into a unified coordinate system, we have to obtain the rigid transformation among multi-LRF. In this paper, we propose a new algorithm for online extrinsic calibration of multi-LRFs by observing a planar checkerboard pattern and solving for transformation between the views of a planar checkerboard from a camera and multi-LRF. Existing LRF calibration is achieved by freely moving a checkerboard pattern and conducting much offline optimization. Compared with traditional algorithm, the advantages of our approach are twofold. Firstly, adopting the noise of images and LRF depth readings, we can exactly calculate the exact position and pose of the checkerboard that can largely reduce the transformation error. Secondly, the complete calibration process is online, which means the exact position and pose of the checkerboard can be obtained in real-time and manipulated by robotic arm. In the end, our calibration approach is validated through real experiments that show the superiority with respect to the state-of-art methods.

I. INTRODUCTION

With development of autonomous driving technology, multi-LRF play an important role in object classification [1], pedestrian detection [2], city construction [3]. In all these cases, the fusion of multi-LRF requires the knowledge of the relative pose for projecting the depth readings into the same coordinate system. Our paper addresses this extrinsic calibration problem.

A. Related Work

One of the most common techniques to solve the extrinsic calibration problem is to use a checkerboard pattern and a monocular camera to find the corresponding point-cloud among multi-LRF [4]–[6]. Therefore, the alignment between camera and LRF has to be accurately calibrated first. A number of published works for the extrinsic calibration of camera and LRF are to directly find the corresponding features between images and point-cloud. The basic idea of Li [4] and Wasielewski's [5] approach is to use constraints obtained from intensity edges on the image and point-cloud lying at the depth edges to optimize the transformation matrix between camera and LRF. The features are abstracted from the point-cloud that is extracted by the depth edges or 3D corners and intensity edges for the calibration target. The extrinsic calibration parameters can be obtained by minimizing the distances between the projected features. However, it is so

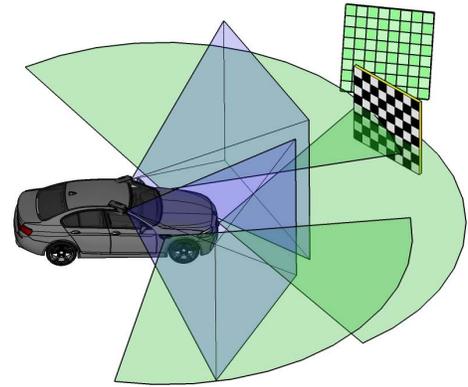


Fig. 1: Geometric model with multi-LRF and camera(Blue and green area represent camera viewing angle and LRF scanning range. The pose and position of green checkerboard armed with robotic manipulator represent the result of our proposed algorithm)

difficult to find such correspondences, because the LRF range is typically outside the visual spectrum. Moreover, certain types of features, like intensity edges, can only be detected in one sensor modality.

B. Contribution

Our approach is similar to the work of Zhang and Pless [6]. Zhang's method was achieved by freely moving a checkerboard in front of camera and LRF and computed the pose of checkerboard from plane-to-image homographies. Given the pose of a calibration checkerboard, the extrinsic calibration parameters are estimated by solving a nonlinear optimization problem which is established with a set of linear constraints among camera coordinates of the checkerboard and its related depth readings. We implemented Zhang and Pless's algorithm and found that the algorithm cannot provide sufficiently accurate results. Zhang's approach is not accurate due to Levenberg-Marquardt optimization method [7]–[9] where the accuracy heavily relies on the initial guess of transformation matrix Φ and the relative position of a camera with respect to the LRF reference frame Δ . Even though initial guess Φ and Δ

for each pose of the checkerboard can be solved by standard linear least square optimization, it is difficult to distinguish which position and pose of checkerboard can lead to the global minimum. Moreover, since all of raw data is achieved by freely moving a checkerboard on different kind of pose and position, it naturally suffers from intensity variations, the narrow fields of view, and the low-accuracy depth information. The proposed algorithm outperforms previous methods in three ways. Firstly, we design the cost function that is based on the position of LRF point-cloud with respect to the camera's coordinate of checkerboard. The error of cost function results from the noise of image and LRF depth-readings for each pose and position of checkerboard. Secondly, according to the distribution of noise, we calculate the exact position and pose of checkerboard for next step, which can largely reduce the cost and transformation error. Finally, the complete procedure is online. Fig. 1 provides a general setup of this calibration method.

C. Organization

The paper is organized as follows: Section II introduces the basic equation associated with geometric constraint and proposed estimation model on the rigid transformation from a camera coordinate system to LRF coordinate system. Section III gives extrinsic calibration parameter estimation method. Firstly, we use a Monte Carlo method to calculate noise distribution under the proposed cost function. Secondly, we apply a Gaussian Mixture Model to cluster the noise and Principal Component Analysis the noisy cluster. Ultimately, the Policy Gradient Descent Learning optimization method is proposed to obtain refined pose and position of checkerboard. Section IV we draw a conclusion by giving the experimental result.

II. BASIC GEOMETRIC CONCEPT

For sensor fusion between a multi-LRF and a camera, both the extrinsic and intrinsic parameters have to be estimated. The intrinsic parameter of a camera (e.g. focal length, optical center, pixel aspect ratio and skew parameters) can be estimated using the Matlab Camera Calibration Toolbox [10]. A projection from the world coordinates $P = [X, Y, Z]^T$ to the image coordinates $p = [u, v]^T$ can be represented as follows [11]:

$$p \sim K(RP + t) \quad (1)$$

K is the camera intrinsic matrix. R is a 3×3 orthonormal matrix representing the camera's orientation. t is a 3-vector representing camera coordinates under world coordinate system.

According to the derivation of Zhang's method [6], an LRF coordinate system can be defined with an origin at the LRF, and the LRF scan plane is the plane $Y = 0$ in the LRF coordinate system. X-Y-Z axes in LRF coordinate system are defined as forward, downward and leftward respectively, while X-Y-Z axes in camera coordinate system are defined as upward, rightward and backward respectively. The checkerboard is the

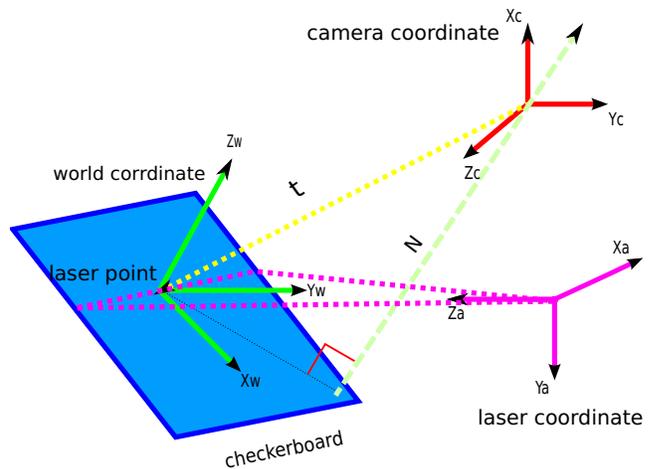


Fig. 2: Geometric interpretation of the camera coordinate, LRF coordinate, and checkerboard plane

plane $Z = 0$ in the world coordinate system. In the camera coordinate system, the checkerboard can be represented by a 3-vector N which is parallel to the normal of the checkerboard. N 's magnitude can be calculated by the distance from camera to the calibration plane.

$$N = -R_3(R_3^T \cdot t) \quad (2)$$

where R_3 is the 3rd column of rotation matrix R .

Given a LRF point P^l , we can determine its coordinate P^c in the camera coordinate system as :

$$P^l = \Phi P^c + \Delta \quad (3)$$

where Φ is a 3×3 orthonormal matrix representing the camera's orientation relative to the LRF and Δ is a 3-vector corresponding to camera position under the LRF coordinate system.

Because P^c is determined by N in the calibration plane, we can derive that:

$$N \cdot P^c = \|N\|^2 \quad (4)$$

Using (3) and (4), the geometric constraint of extrinsic calibration parameters between camera and LRF is described as Fig. 2 and can be concluded:

$$N \cdot \Phi^{-1}(P^l - \Delta) = \|N\|^2 \quad (5)$$

where the transformation matrix Φ can be parametrized as a 3-vector parameter k according to the Rodrigues formula [12]:

$$\Phi = I + (\sin \theta)K + (1 - \cos \theta)K^2 \quad (6)$$

where K is defined by:

$$K = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad (7)$$

III. EXTRINSIC CALIBRATION ALGORITHM

A. Monte Carlo algorithm with Gaussian Mixture Model

Given every LRF point-cloud j from data-set i and the pose of checkerboard N_i , we can define the error function $G(\Phi, \Delta)$ by using (5):

$$G(\Phi, \Delta) = \sum_i \sum_j (N_i \cdot (\Phi^{-1}(P_{ij}^l - \Delta)) - \|N_i\|^2)^2 \quad (8)$$

In general, noises exist in the output of camera and LRF, which have a great effect on the accuracy of extrinsic calibration parameters. We develop an online robust method to simulate the noise distribution. Firstly, there are six degrees of freedom in Φ and Δ from our predefined error function (6). Assuming these six variables are initialized uniform distributed, we practically assign different intervals (a_i, b_i) for different degrees of freedom with respect to different sensor setups. Suppose x_i is one degree of freedom, therefore:

$$x_i \sim \mathcal{U}(a_i, b_i) \quad (9)$$

We use Gaussian Mixture Model [13] to cluster the values of error functions and generate a new probability density function after the first Monte Carlo integration [14] $G(x_i)$. To begin with, the following errors are computed.

$$q(x) = \sum_{m=1}^M \pi_m(N)(x|\mu_m, \Sigma_m) \quad (10)$$

where π_m is the parameter of distribution of observation associated with cluster m . μ_m is the mean value of cluster m . Σ_m is the covariance matrix of cluster m . Given the value of error function, we initialize μ_m , Σ_m , π_m . After that, we evaluate the responsibilities by using the current parameter values:

$$\gamma(z_{im}) = \frac{\pi_m(N)(x_i|\mu_m, \Sigma_m)}{\sum_{j=1}^M \pi_j(N)(x_i|\mu_j, \Sigma_j)} \quad (11)$$

where $\gamma(z_{im})$ can be viewed as the responsibility that cluster m takes for explaining the observation x_i . The next step is to use current responsibilities to re-estimate the parameters:

$$\mu_m^{new} = \frac{1}{Q_m} \sum_{i=1}^6 \gamma_{z_{im}} x_i \quad (12)$$

$$\Sigma_m^{new} = \frac{1}{Q_m} \sum_{i=1}^6 \gamma_{z_{im}} (x_i - \mu_m^{new})(x_i - \mu_m^{new})^T \quad (13)$$

$$\pi_m^{new} = \frac{Q_m}{Q} \quad (14)$$

where

$$Q_m = \sum_{i=1}^6 \gamma(z_{im}) \quad (15)$$

Lastly, we evaluate the log likelihood and check for the convergence.

$$\ln q(x|\mu, \Sigma, \pi) = \sum_{i=1}^6 \ln \left\{ \sum_{m=1}^M \pi_m(N)(x_i|\mu_m, \Sigma_m) \right\} \quad (16)$$

When the log likelihood meets the convergence criterion, we use the new probability density function $q(x)$ to draw the samples. This method is referred as Importance Sampling [15]. Suppose probability density function of previous Monte Carlo Integration is $p(x)$. Hence:

$$\begin{aligned} E[G(x_i)] &= \int_{b_i}^{a_i} G(x_i)p(x_i)dx \\ &= \int_{b_i}^{a_i} \frac{G(x_i)p(x_i)q(x_i)}{q(x_i)}dx \\ &= E \left[\frac{G(x_i)p(x_i)}{q(x_i)} \right] \end{aligned} \quad (17)$$

By generating n samples $x_i \sim q(x)$, the new Monte Carlo estimator becomes:

$$\begin{aligned} I &= \frac{1}{n} \sum_{j=1}^n \frac{G(x_{ij})p(x_{ij})}{q(x_{ij})} \\ &= \frac{1}{n} \sum_{j=1}^n W(x_{ij})G(x_{ij}) \end{aligned} \quad (18)$$

where $W(x_{ij}) = \frac{p(x_{ij})}{q(x_{ij})}$ are the importance weights. We manually set up two criterion. Firstly, when the importance weight $W_{ij} < \delta$, our algorithm reject the sample value. Secondly, when variance of new Monte Carlo Integration is under one certain value ξ , we end up the algorithm and use GMM to pick up the most noisy cluster. A summary of Monte Carlo algorithm with Gaussian Mixture Model (MC-GMM) is as follows.

Algorithm 1 Monte Carlo algorithm with Gaussian Mixture Model (MC-GMM)

- 1: **Initialize** $x_i \sim \mathcal{U}(a_i, b_i)$
 - 2: **while** $Var_q[G(x_{ij})] \leq \xi$ **do**
 - 3: GMM cluster until convergence ▷ new pdf $q(x)$
 - 4: Importance Sampling ▷ using $q(x)$ for re-sampling
 - 5: **if** $W_{ij} \leq \delta$ **then**
 - 6: reject sampling value
 - 7: **end if**
 - 8: **end while**
-

B. Policy Gradient Descent Learning with Principal Component Analysis

In order to speed up for finding the minimum value of our predefined error function, we need to analyze the noise distribution. To eliminate the noise from Φ and Δ (totally 6 degree of freedom), we apply Principal Component Analysis to find out the first principle component of the noisy cluster. First of all, according to Monte Carlo simulation result with a Gaussian Mixture Model, we can pick up the most noisy cluster and apply PCA [16] to calculate the largest eigenvalue. Besides, we use Policy Gradient Descent [17] to find out which pose and position of the checkerboard can largely reduce the first component of the noisy cluster. There are also six degrees of freedom for checkerboard. Suppose $B_x, B_y, B_z, B_\alpha, B_\beta,$

B_γ are the poses and positions of checkerboard in the world frame. Assuming the policy parametrization is varied S times by small increments ΔB_{is} , our goal of the policy optimization is to optimize ΔB_{is} . Hence, our optimization cost function J is :

$$J(B_{is} + \Delta B_{is}) = \lambda(B_{is} + \Delta B_{is}) \quad (19)$$

where B_{is} represent i_{th} degree of freedom of checkerboard in the s iteration time, λ is the first component of corresponding noise cluster.

The policy parameter evolution, i.e. $B_{is} + \Delta B_{is}$, is performed, such that:

$$\Delta J = J(B_{is} + \Delta B_{is}) - J_{ref} \quad (20)$$

$$J_{ref} = J(B_{is} - \Delta B_{is}) \quad (21)$$

Using (20) and (21), we can derive the gradient as :

$$grad_B = (\Delta B_s^T \Delta B_s)^{-1} \Delta B_s^T \Delta J \quad (22)$$

In the last, update the pose of checkerboard by using gradient:

$$B_{i(s+1)} = B_{is} - \epsilon grad_B \quad (23)$$

where ϵ is the value of step size. When the optimization function is converged (e.g ΔJ is under one certain value ϵ), we can acquire final result for the pose and position of checkerboard and the rotation matrix between camera and LRF Φ , camera position w.r.t LRF coordinate system Δ . The estimation matrix Φ can find the nearest orthonormal matrix Φ_R to represent the rotation matrix, which is subject to $\Phi_R^T \Phi_R = I$. Φ_R can be calculated by using Frobenium norm [18]. The final rotation matrix can be derived as :

$$\Phi_R = \Phi(\Phi^T \Phi)^{-\frac{1}{2}} \quad (24)$$

Suppose there are two LRFs and one camera in our sensor setup. Φ_1 and Φ_2 represent the corresponding rotation matrices between the corresponding LRF and camera. So the rotation matrix between these two LRF is :

$$\Phi_1^2 = \Phi_2 \Phi_1^{-1} \quad (25)$$

Summary of Policy Gradient Descent with Principal Component Analysis is described as following:

Algorithm 2 Policy Gradient Descent Learning with Principal Component Analysis(PGDL-PCA)

- 1: **Initialize** policy parametrization B_i
 - 2: **while** $\Delta J \leq \epsilon$ **do**
 - 3: generate policy variation ΔB_i
 - 4: generate most noisy cluster by MC-GMM($B_i + \Delta B_i$)
 - 5: estimate $J_i = J(B_i + \Delta B_i)$ by PCA
 - 6: generate most noisy cluster by MC-GMM($B_i - \Delta B_i$)
 - 7: estimate $J_{ref} = J(B_i - \Delta B_i)$ by PCA
 - 8: compute $\Delta J_i = J_i - J_{ref}$
 - 9: update parameter B_i
 - 10: **end while**
 - 11: **return** B_i, Δ, Φ
-

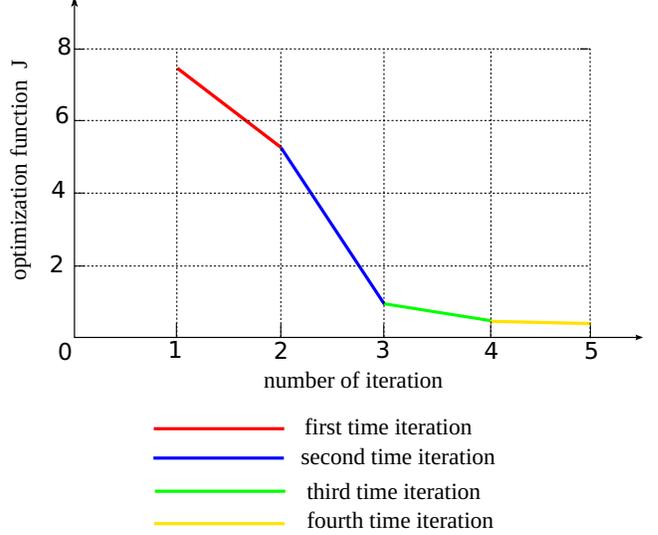


Fig. 3: Variation of color represents number of iterative optimization step

iteration	B_x	B_y	B_z	B_α	B_β	B_γ	$J(B + \Delta B)$
1	1.7m	1.2m	0.3m	0.51rad	0.46rad	0.42rad	7.2
2	1.8m	1.3m	0.4m	0.61rad	0.56rad	0.52rad	5.6
3	2.8m	2.3m	1.4m	1.61rad	1.56rad	1.52rad	0.98
4	3.2m	2.7m	1.8m	2.01rad	1.96rad	1.92rad	0.2

TABLE I: Value of optimized function J results from number of iterative optimization step

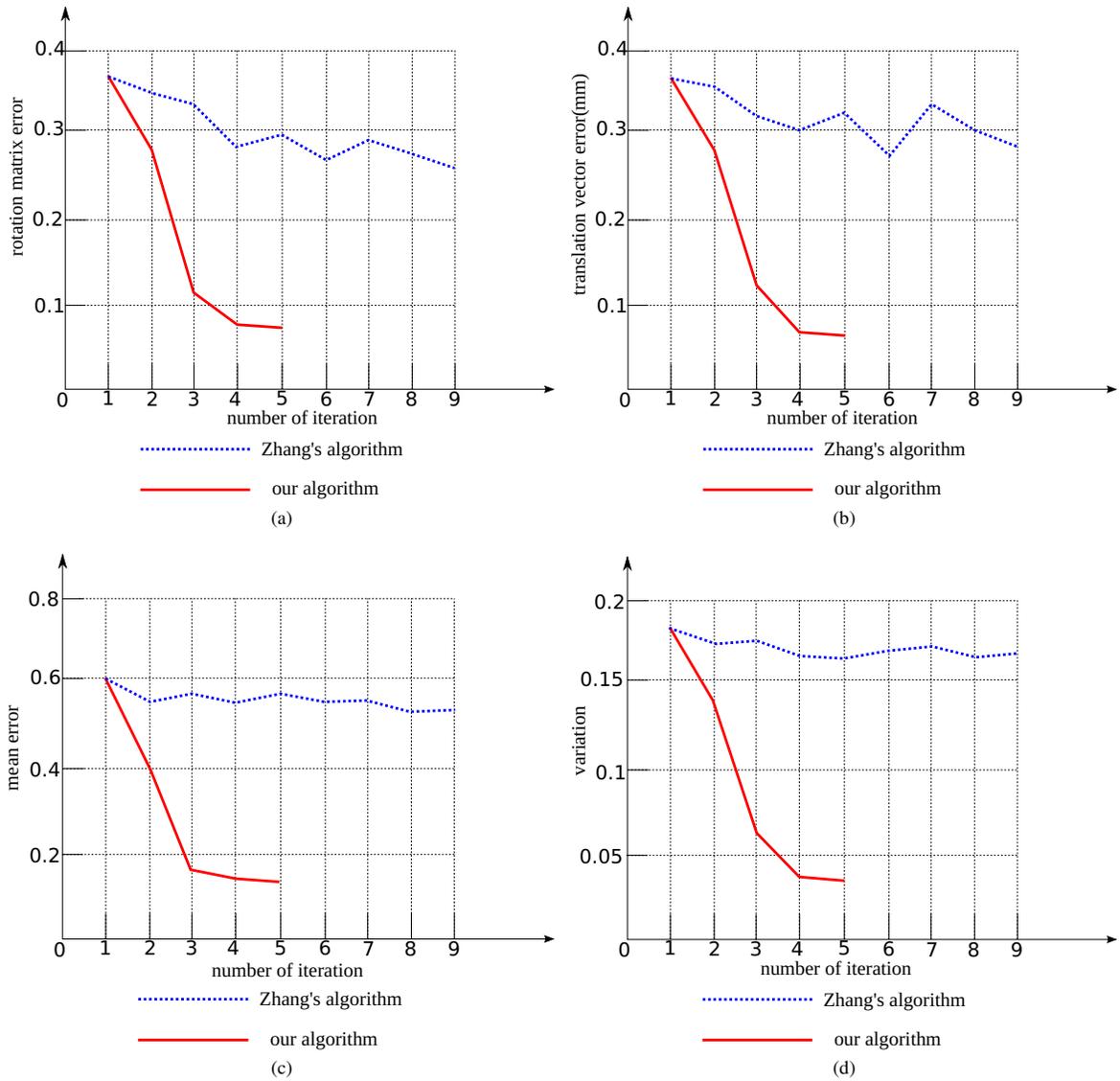
IV. EXPERIMENTAL RESULTS

A. Experiments with Simulated data

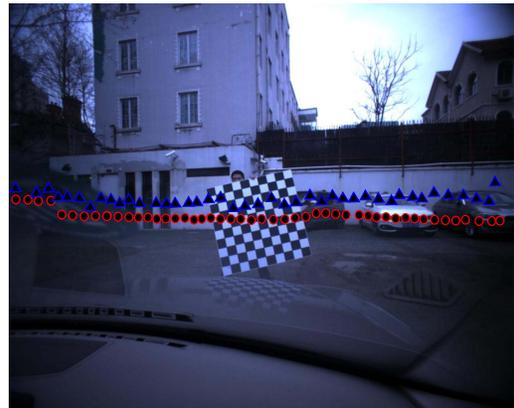
Simulated data provide us series of scenarios with controlled environment to study effects of noise on the estimation model. The estimated parameters are compared with the ground truth in order to measure the estimation errors. The simulations of our experimental setup are conducted to analyze the estimated parameter errors of our proposed method and that of Zhang's method [6].

In this experiment, we set up two Ibeo LRFs, two monocular cameras and conduct the simulation in Gazebo. The resolution of the IP camera is 1280×960 . There are 4 LRF layers for an Ibeo LRF. The separation angle of each layer is 0.08° . The angle increment rad for each layer is 0.05° . From our test, we take the 4th layer LRF point cloud as input. Our calibration pattern plane is a checkerboard defined by 10×10 grids. The size of the pattern square is $150mm \times 150mm$. The LRF and the camera are stationary, such that there is always a significant overlap between the field-of-views of two Ibeo LRFs. The checkerboard placement guarantees the intersection with multiple LRF scan plane.

The estimated extrinsic parameters Φ and Δ are compared with the ground truth Φ_{GT} and Δ_{GT} . The accuracy of rotation matrix is quantified by the angular magnitude of the residual rotation $\Phi^T \Phi_{GT}$ and by the relative translation error $\|\Delta - \Delta_{GT}\| / \|\Delta_{GT}\|$.



(e)



(f)

Fig. 4: (a) and (b) represent the estimation error of the rotation matrices Φ and Δ , with respect to the number of iteration. (c) and (d) represent the mean error and variance of the reprojection of LRF point-cloud into image coordinate system, with respect to the number of iteration. (e) and (f) represent the projection of right and left LRF into the images using the extrinsic calibration results obtained from 5 samples with our algorithm(circles) and Zhang's algorithm(triangles)

Fig. 3 and Table I show that the first component of the noisy cluster is largely reduced within 4 iteration and meet the convergence criterion. The aim of our algorithm is to find out the exact pose and position of the checkerboard which can largely reduce the estimation error. The accuracy of extrinsic parameters can be improved at each iteration, unlike that with free motion of the checkerboard.

B. Comparison with Zhang's method

For the sake of fairness, Zhang's method is practically implemented by using a hypothesize-and-test framework which is described as [6]. The experiment considers number of calibration planes and runs 100 calibration trials and compares the error distribution. Gaussian noise with mean 0 and standard deviation 0.5 pixel is added to the projected image.

From analysis of Fig. 4(a) and Fig. 4(b), our algorithm significantly decreases the chances of divergence during the iterative optimization step. Our calibration results has converged at the 4th iteration, while Zhang's method converge very slowly, e.g. per our test, Zhang's result cannot converge at the 9th iteration. Moreover, since the checkerboard motion is unconstrained, Zhang's method discards the existence of fewer images with LRF readings which can not fulfill the geometric constraint. For example, when checkerboard is holding in certain pose that is parallel to the LRF scan plane or cannot receive LRF depth readings, this one is an invalid pose. We have to manually construct a current valid pose set for Zhang's method, while there is no offline labor work for our solution in terms of valid pose set construction. For the nine-iteration test case, an exhaustive search of Zhang's solution space requires 88 trials. Further improvement of our algorithm's accuracy is marginal by further iterative refinement.

C. Experiments with Real Data

Our proposed method was tested on a robotic platform whose sensor setup is the same as those in the simulation environment. Although we do not have the ground truth of the extrinsic parameters Φ and Δ , the results from real data show that our proposed method is well performed and yield reliable calibration parameters. According to the characteristic of Ibeo LRF, one certain layer of LRF point-cloud should fit on one line. We use linear regression [19] to fit a straight line through the set of 175 points(the maximum number of point-cloud in one certain layer of Ibeo LRF) in such a way that makes the mean error and variance as small as possible. Suppose there are 175 point-cloud for each certain pose, e.g. per our test, $[X_i, Y_i]^T$ represents the projection of i th point-cloud into image coordinate system. \hat{Y}_i represent the prediction value estimated by linear regression with the corresponding X_i . The accuracy can be quantified by mean error $\frac{1}{175}(\hat{Y}_i - Y_i)$ and variance $\frac{\sum_{i=1}^{175}(Y_i - \hat{Y}_i)^2}{175-1}$. From analysis of Fig. 4(c) and Fig. 4(d), our algorithm significantly decrease the mean error and variance during the iterative optimization step, while Zhang's algorithm hardly decrease the mean error and variation even after 9 iteration. The calibration results are visualized as Fig. 4(e) and Fig. 4(f).

V. CONCLUSION

In this paper, we presented an online active calibration algorithm for multi-LRF systems. In view of the noise distribution of the input from camera and LRF, we can quickly determine the exact pose and position of a checkerboard to obtain a high accuracy of extrinsic calibration. The complete calibration procedure is online which is of great convenience. Experiments with real data proved the stability of the algorithm. The proposed algorithm outperforms state-of-the-art algorithms in terms of the less required number of checkerboard poses and lower time-cost.

VI. ACKNOWLEDGMENT

This work was partially supported by the HKUST Project IGN13EG03; partially sponsored by the Research Grant Council of Hong Kong SAR Government, China, under project No. 16206014 and National Natural Science Foundation of China No.6140021318

REFERENCES

- [1] B. Douillard, D. Fox, and F. Ramos, "Laser and vision based outdoor object mapping," in *Robotics: Science and Systems*, 2008.
- [2] C. Premevida, O. Ludwig, and U. Nunes, "Lidar and vision-based pedestrian detection system," *Journal of Field Robotics*, vol. 26, pp. 696–711, September 2009.
- [3] C. Früh and A. Zakhor, "An automated method for large-scale, ground-based city model acquisition," *International Journal of Computer Vision*, vol. 60, pp. 5–24, 2004.
- [4] G. Li, Y. Liu, L. Dong, X. Cai, and D. Zhou, "An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features," in *Intelligent Robots and Systems (IROS), 2007. IEEE/RSJ International Conference on*, 2007, pp. 3854–3859.
- [5] S. Wasielewski and S. O., "Calibration of a multi-sensor system laser rangefinder/camera," in *Intelligent Vehicles 1995 Symposium, Proceedings of the*, 1995, pp. 472–477.
- [6] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *Intelligent Robots and Systems (IROS), 2004. IEEE/RSJ International Conference on*, vol. 3, 2004, pp. 2301–2306.
- [7] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly Applied Math*, vol. 2, pp. 164–168, 1944.
- [8] D. Marquardt, "An algorithm for least squares estimation of nonlinear parameters," *SIAM Journal of Applied Math*, vol. 11, pp. 431–441, 1963.
- [9] J. Mor., "The levenberg-marquardt algorithm: Implementation and theory," *Lecture Notes in Mathematics*, vol. 630, pp. 105–116, 1977.
- [10] J. Bouguet, "Camera calibration toolbox for matlab," 2003.
- [11] R. I. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [12] J. E. Mebius, "Derivation of the euler-rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations," in *arXiv General Mathematics*, 2007.
- [13] C. Priebe and D. Marchette, "Adaptive mixture density estimation," *Pattern Recognition*, vol. 26, no. 5, pp. 771–785, 1993.
- [14] R. Caflisch, "Monte carlo and quasi-monte carlo methods," *Acta Numerica*, vol. 7, pp. 1–49, 1998.
- [15] M. Ferrari and S. Bellini, "Importance sampling simulation of turbo product codes," in *IEEE International Conference on Communications*, vol. 9, 2001, pp. 2773–2777.
- [16] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine Learning*, vol. 56, pp. 9–33, 2004.
- [17] P. Glynn, "Likelihood ratio gradient estimation for stochastic systems," *Communications of the ACM*, vol. 33, no. 10, pp. 75–84, 1990.
- [18] B. Horn, "Closed form solution of absolute orientation using unit quaternions," *Journal of the Optical Society*, vol. 4, no. 4, pp. 629–642, 1987.
- [19] D. A. Freedman, *Statistical Models: Theory and Practice*. Cambridge University Press, 2009.