Proceedings of the 2018 IEEE
International Conference on Robotics and Biomimetics
December 12-15, 2018, Kuala Lumpur, Malaysia

Proposing Camera Calibration Method using PPO (Proximal Policy Optimization) for Improving Camera Pose Estimations

Haitham Al-Jabri and Takafumi Matsumaru, Member, IEEE

Abstract— **This paper highlights camera orientation estimation accuracy and precision, as well as proposing a new camera calibration technique using a reinforcement learning method named PPO (Proximal Policy Optimization) in offline mode. The offline mode is used just for extracting the camera geometry parameters that are used for improving accuracy in real-time camera pose estimation techniques. We experiment and compare two popular techniques using 2D vision feedbacks and evaluate their accuracy beside other considerations related to real applications such as disturbance cases from surrounding environment and pose data stability. First, we use feature points detection ORB (Oriented FAST and Rotated BRIEF) and BF (Brute-Force) matcher to detect and match points in different frames, respectively. Second, we use FAST (Features from Accelerated Segment Test) corners and LK (Lucas–Kanade) optical flow methods to detect corners and track their flow in different frames. Those points and corners are then used for the pose estimation through optimization process with the: (a) calibration method of Zhang using chessboard pattern and (b) our proposed method using PPO. The results using our proposed calibration method show significant accuracy improvements and easier deployment for end-user compared to the pre-used methods.**

I. INTRODUCTION

The exponential growth of vision technology advancements in recent years contributes in widely spreading the usage of computer vision. Computer vision has become very significant in supporting the interaction of technologies with the real-world and therefore gains lots of researchers' attentions. There are many existing methods that analyze the vision feedbacks to extract useful data such as: objects/portions identification and pose estimations.

Camera pose estimation is a method that tracks the 3D orientation and position of a camera in the real-world. Lots of applications use different camera pose estimation techniques as a basement to implement different tasks. There are mainly two approaches to recover 3D pose from a single 2D camera frames: SfM (Structure from Motion) [1] and artificial markers [2]. The artificial markers approach has the problem of the need to distribute markers and distribution calibration in the system. SfM has the problem of rough pose estimation and lack of scaling. However, there is still progress in enhancing the quality of the SfM approach by improving features tracking, pose corrections using reprojection techniques and/or improving the camera calibration.

* Oman Government Ministry of Higher Education and Mitsubishi for Gas Corporation support this research

H. Al-Jabri and T. Matsumaru with Bio-Robotics & Human-Mechatronics Lab, Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Fukuoka, Japan(e-mail: h.aljabri@ fuji.waseda.jp).

Camera calibration is an essential process for the pose estimation using vision feedbacks to get the exact geometry relation between the world and images. Precisely, this relation is unique for every camera due to hardware exact dimensions and assembly discrepancies. There are several methods used to calibrate the camera and get geometry parameters [3][4][5].

The vision process techniques are widely used for many applications to extract useful data but sometimes with limitations such as processing time. Computational power has seen good recent improvements that make vision tasks more effective and reliable. The other motivation is the need for cost-effective, precise, and accurate techniques for positioning task feedbacks in robot applications. The overall targeted problem is overcoming accumulated errors and achieving higher robot positioning resolution with cost-effective components.

This paper highlights camera orientation estimation accuracy and precision and proposes new camera calibration technique using reinforcement learning PPO (Proximal Policy Optimization) [6] in offline mode. The offline mode is used just for extracting the camera geometry parameters and then, we use those parameters for improving accuracy in real-time camera pose estimation techniques. We discuss the differences between two different techniques of camera pose estimation using (a) calibration method of Zhang with chessboard pattern and (b) our proposed method.

II. RELATED WORKS

A. An Improved Method of Real-time Camera Pose Estimation Based on Descriptor Tracking

Zong et al. [7] proposed a method to improve real-time camera pose accuracy by using descriptor tracking. The main idea in their approach is to use LK optical-flow instead of matching technique to minimize errors. The results using this approach show stable readings in static environment. However, the limitation of this approach is that it might be disturbed by surrounding moving objects. Moreover, after changing the keyframe in which new feature points are tracked, the possibility of error caused by changing group of points appears. This leads to an increase in accumulated errors. We experiment and discuss this approach in this paper.

978-1-7281-0377-8/18/$31.00 © 2018 IEEE

B. *Feature Tracking and Synchronous Scene Generation with a Single Camera*

Chai and Matsumaru [8] proposed a system to generate a virtual scene in real-time. They use the features tracking technique (ORB-Optical flow) to estimate the camera pose. Their proposed system was to capture the video under the limitation of no other objects are moving in front of the camera. Their used camera is calibrated using Zhang [9] method.

Zhang proposed a camera calibration method using a known printed pattern and taking snapshots for the pattern by the camera from different view angles. OpenCV library [10] supports three types of patterns for calibration: (1) Classical black-white chessboard (2) Symmetrical circle pattern (3) Asymmetrical circle pattern. The pattern dimensions are measured and then inserted in the program to compute camera geometry parameters.

C. *Deep EndoVO*

Turan et al. [11] proposed a RCNN (Recurrent Convolutional Neural Network) based on visual-odometry approach for endoscopic capsule robots named as Deep EndoVO. Deep EndoVO has high rotational and translation accuracy compared to state-of-art SLAMs (Simultaneously Localization and Mapping), based on experimental results in real pig stomach dataset as shown in [11]. The process in Deep EndoVO does not require camera calibration which the system can model sequential frames from endoscopic video and solve pose estimation problem based on pre-request input of RGB images dataset. However, dataset requirement might be a limitation when applied on other environments.

D. *A Perceptual Measure for Deep Single Image Camera Calibration*

Hold-Geoffroy et al. [12] proposed a perceptual measure for camera calibration based on training a network that uses automatic generated samples from a large-scale panorama-dataset. The strength of this approach is that it relies on a large-scale human perception study where the results might be meaningful for the 3D scene reconstruction but might not be the case for the real-world camera pose estimation.

III. Theoretical Background

A. *Camera Pose Estimation*

1) *3D-2D Transformation*

Cameras can provide us with 2D images of 3D world and we still as human can estimate the 3D environment from 2D image. Using vision techniques to compute 3D data; the 3D-2D transformation need to be mathematically modeled [8]. This model is shown in Eq. (1), which relates the projected 3D points in 2D image.

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{K} \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (1)$$

$$\boldsymbol{K} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

the matrix $\boldsymbol{K}$ consists of the camera parameters which is called intrinsic matrix. The matrix $\boldsymbol{R}$ (the rotation matrix (3x3)) and $\boldsymbol{t}$ (the translation vector (3x1)) form what is called extrinsic matrix. The variables u and v are pixel points in the image and λ represents the scale of the points in real world. The constants f_x and f_y are focal lengths and u_0 and v_0 are the images actual origin which is called the principle point. The variables X_w, Y_w and Z_w are the world-coordinate points.

When using 2D images where there is no depth information, homogeneous coordinate system is used to be able to solve the pose estimation problem. Eq. (3) shows Eq. (1) in the homogeneous coordinate. This basically means that the variable x and y are not the same variables as u and v defined in Eq. (1). The solutions of these variables should then be divided by third row element to get corresponding u and v. Let scale $\lambda = 1$:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \boldsymbol{K} \begin{pmatrix} \boldsymbol{r^1} & \boldsymbol{r^2} & \boldsymbol{r^3} & \boldsymbol{t} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (3)$$

$$u = \frac{x}{\lambda} \ , \ v = \frac{y}{\lambda} \quad (4)$$

We can now define the local world point coordinate in which $Z_w = 0$ to simplify the problem. The vector $\boldsymbol{r^3}$ will then be multiplied by 0 and thus, it is omitted in Eq. (5).

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \boldsymbol{K} \begin{pmatrix} \boldsymbol{r^1} & \boldsymbol{r^2} & \boldsymbol{t} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix} \quad (5)$$

Here, we can define a 3x3 matrix which is called the homography matrix and it is basically defined as in Eq. (6). This matrix is widely used in image processing as the points transformation matrix between different frames.

$$\boldsymbol{H} = \boldsymbol{K} \begin{pmatrix} \boldsymbol{r^1} & \boldsymbol{r^2} & \boldsymbol{t} \end{pmatrix} \quad (6)$$

Now, if we consider two frames and we track a point in both frames as shown in Fig. 1, we can express the relation as in Eq. (7).

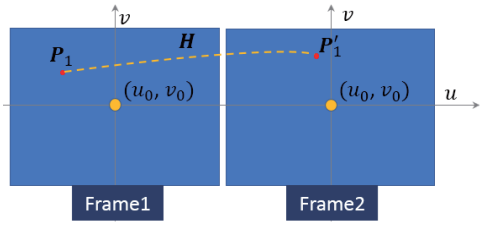


Figure 1: A single point relationship between two frames

$$\begin{pmatrix} x_1' \\ y_1' \\ \lambda_1' \end{pmatrix} = \boldsymbol{H} \begin{pmatrix} x_1 \\ y_1 \\ \lambda_1 \end{pmatrix} \quad (7)$$

where x_1', y_1' and λ_1' are parameters of a point in Frame2. The parameters x_1, y_1 and λ_1 are parameters of the same point in Frame1.

The matrix $\boldsymbol{H}$ is the homography matrix of the point between the two frames. So, expressing the matrix into equations is shown in Eqs. (8), (9) and (10).

$$x_1' = h_{11}x_1 + h_{12}y_1 + h_{13}\lambda_1 \quad (8)$$
$$y_1' = h_{21}x_1 + h_{22}y_1 + h_{23}\lambda_1 \quad (9)$$
$$\lambda_1' = h_{31}x_1 + h_{32}y_1 + h_{33}\lambda_1 \quad (10)$$

Substitute Eqs. (8), (9) and (10) into Eq. (4) and let $\lambda_1 = 1$:

$$u_1' = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (11)$$

$$v_1' = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (12)$$

Equations (11) and (12) represent one-point homography relations, and so, we have a group of points that are not moving in real-world but only the camera moves. The homography matrices of those points represent one homography matrix of the camera movement. Equation (13) shows listing of points' pairs equations shown in Eqs. (11) and (12) (at least 4 pairs) to solve for the homography elements using SVD (Singular Value Decomposition).

$$\begin{pmatrix} \cdots & & \\ \vdots & \ddots & \vdots \\ \cdots & & \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \mathbf{0} \quad (13)$$

After we get the homography matrix; $\boldsymbol{r^1}$, $\boldsymbol{r^2}$ and $\boldsymbol{t}$ can be decomposed if we know the camera parameters. The camera parameters are usually previously-known by a calibration process. The vector $\boldsymbol{r^3}$ as it is perpendicular to $\boldsymbol{r^1}$ and $\boldsymbol{r^2}$ is equal to the cross product of $\boldsymbol{r^1}$ and $\boldsymbol{r^2}$ as shown in Eq. (14).

$$\boldsymbol{r^3} = \boldsymbol{r^1} \times \boldsymbol{r^2} \quad (14)$$

1) Features Detection

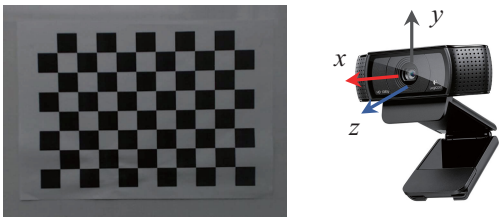
To get the pose estimation, some distinctive points in different viewpoints should be tracked. There are three main features that are commonly used in image processing pose estimation: Edges, Corners and Blobs.

Chai and Matsumaru [8] compared between five existing corners detection techniques: 1) HARRIS, 2) FAST, 3) SIFT, 4) SURE 5) ORB. They show that FAST technique has the minimum time cost among the others, but it is not capable for rotation and scale variances. The second fastest technique is ORB [13] and is better in brightness, rotation, and scale invariances. In our current paper, we use both FAST and ORB in different experiments to detect the feature points.

IV. Configuration

A. Experiment Components

We use in our experiments the Logitech c920r webcam and we firstly calibrate it using Zhang method with chessboard pattern shown in Fig. 2(a). The chessboard pattern has 70 corners and the size of each square is measured to be 22.9mm.



(a) (b)

Figure 2: Used a) Chessboard b) Logitech camera c920r

B. Number of Points Used and their Selection Strategy

To estimate the pose of the camera, the minimum requirement is to have 4-points' pairs. However, in this paper, we use 10- and 400-points' pairs in ORB-Matching technique (explained in detail in the following section). We use those number of points to show the difference in increasing or decreasing the number of points used for the pose estimation. The points are selected based on the best matching score. As the ORB points are described in a string and this string is matched with another point's string. We then select the best fit.

V. Pose Estimation Techniques

A. ORB-Matching

This technique is basically getting the feature points from two frames and matching them. Then, we use those matched features for the pose estimation. ORB is an improved version of FAST corners detection in which it can tackle with rotation and scale. The strength of this technique compared to FAST optical flow [14] is the ability to get the pose even with surrounding motions disturbances as shown in Fig. 3. However, the weakest point is that the pose results is not stable; there are fluctuations because it uses different group of points in every estimation. This problem is addressed in Fig. 8 in Results section which shows stationary orientation estimation using 10 of ORB points.

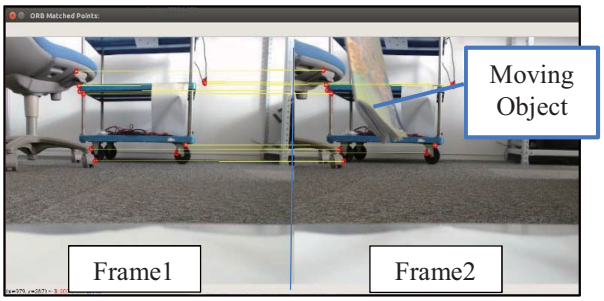


Figure 3: ORB-matching technique can obtain points even with surrounding disturbance motions

B. FAST-Optical Flow

This technique is basically calculating the flow of feature points. The key strength of this technique is that same group of points in each keyframe is used to estimate the pose in which the results are stable with offset errors than actual pose.

However, the points can be displaced by surrounding moving objects as shown in Fig. 4.

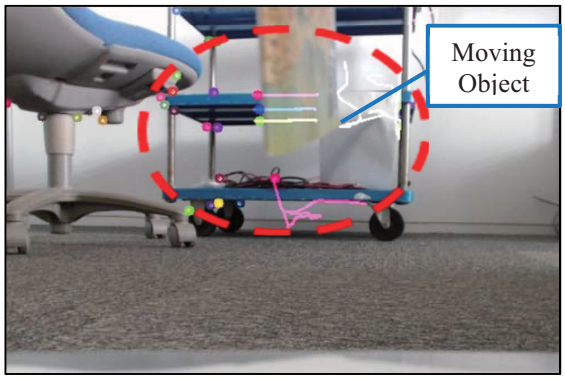


Figure 4: FAST-Optical Flow tracking technique is disturbed by surrounding motions

VI. Proposed Calibration Method

In this section, we explain our proposed method of the camera calibration. The idea came up after analyzing the pose data and figuring out what are the factors can be applied to improve camera pose accuracy. We found out even if the camera is not moving and the points are matched perfectly, the pose is still having some errors fluctuating for different group of points. Therefore, we first considered how to select the best points that lead to accurate results. We ended-up with some difficulties because of unlimited environment structures. Then, we made a trial in changing slightly the camera parameters. The results in this trial show big impact in the pose estimation. This fact led us to think of a way that can get the exact parameters of our used camera.

A. Solving Mechanisms

Our proposed calibration method is to use one image taken from the camera and then run PPO training to get the camera parameters using ORB-matching method. The image represents the two frames in the pose estimation. Thus, the expected pose results are a set of zeros. The objective function of the training is modeled in Eq. (15) and process flowchart is shown in Fig. 5 which is to get 10 pose estimations, calculate the error, then update the camera parameters.

Objective function:

$$Minimize\ \boldsymbol{Z} = \sum_{i=1}^{k}(|x_i| + |y_i| + |z_i|) \qquad (15)$$

where, i is the number of poses. The variables x, y, z are the 3D angles while k is the number of pose samples in each intrinsic matrix update. We used 10 samples in the training phase.

PPO is an effective optimizing process presented in [6] that contains three main key factors: (a) *state* (b) *reward* (c) *actions*. In our case, the *state* is the 10 poses accumulated errors for the 3D angles, the *reward* is formed as the score of Eq. (15), and *actions* are the resulted coefficients the we used for updating the camera parameters.

The process input is a highly structured image, meaning it has more feature points to use for the pose estimation. The pose estimation uses a group of selected points' pairs which are changing randomly in each iteration. The PPO learning-algorithms takes care of providing coefficients for updating camera parameters that converge to the stated objective function.

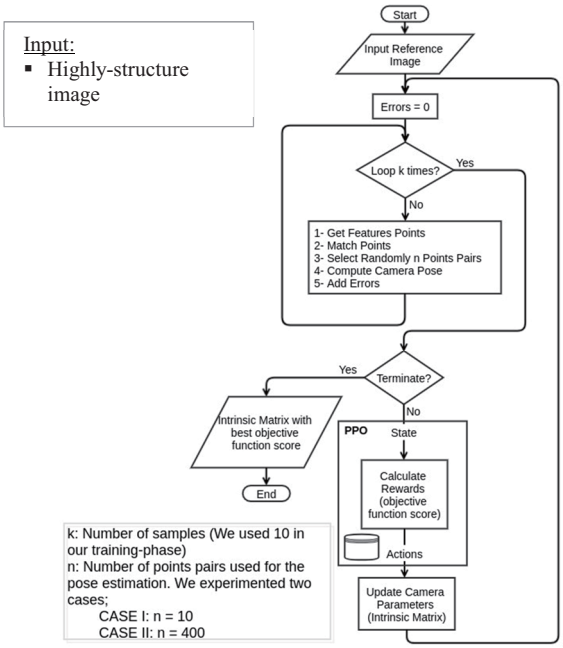


Figure 5: Flowchart of our proposed calibration method

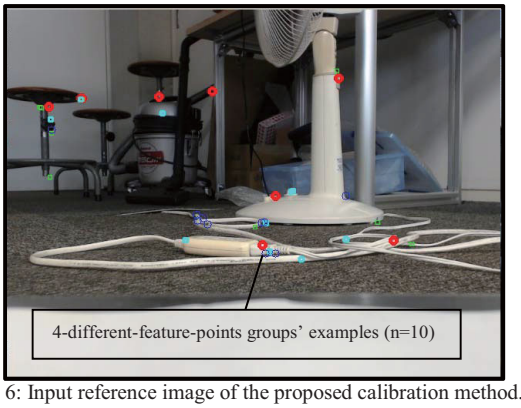


Figure 6: Input reference image of the proposed calibration method. It is highly structed (i.e. sufficient feature points)

B. Summary

The criteria to apply our proposed calibration method is relatively simple: We need to take only one highly structured image from the camera that needs to be calibrated. Fig. 6 shows a reference image that we used in our training phase experiments.

The summary of the proposed calibration method can be expressed as follow:

1- Input: A highly structured image as a reference image
2- Run the proposed method learning program (using reinforcement learning, PPO)
3- Terminate when satisfy with the best-found objective function score (Eq. (15)).

VII. Results And Discussions

A. Results

This section shows the results of our experiments with Zhang and our proposed methods to get the camera intrinsic matrix. We validate the results with direct stability experiment and re-projection errors evaluation. The used images size is 640x480.

1) Stability Experiment

The intrinsic matrix is used at stationary pose where the expected orientation to be (0,0,0). We first calibrate using Zhang method with 18 reference images and then we use our proposed method with a single image shown in Fig. 6 in previous section. We consider two cases in our proposed method: *Case I*, we use 400 ORB points while *Case II,* we use 10 ORB points. Table 1 shows the results of our used webcam intrinsic matrix (rounded to 1 decimal point) in different calibration methods, as well as the PPO objective function scores at the time we terminate the training. Intrinsic results were then initialized in live capturing process of the pose estimation using ORB-matching technique at stationary pose. 100 estimations of the three angles for each method are shown in Figs. 8, 9, and 10. The training phase in the present experiments is terminated within about 3-4 hours. More time for training might lead to better results.

Table 1: Calibration results of our used webcam (Logicool c920r)

	f_x	f_y	u_0	v_0	Z
Zhang	606.4	605.6	327.3	234.6	-
Case I	609.6	609.6	322.4	239.0	0.00162
Case II	601.7	601.7	338.5	257.9	0.01966

2) Re-projection Errors Evaluation

We carry out a re-projection error evaluation using pattern chessboard with *solvePnP* and *projectPoints* (*Opencv* built-in functions) in two different scenarios (15 images each) by using: (a) same reference images that we used in Zhang (b) different reference images than we used in Zhang. The reprojection error is calculated using Eq. (16).

$$Error = \frac{\sqrt{\sum_i (imgP_i - projP_i)^2}}{\sqrt{\sum_i (projP_i)^2}} \qquad (16)$$

i is the number of corners, $imgP_i$ are detected corners and $imgP_i$ are the points projection results of the pose estimation calculation using different intrinsic matrices parameters presented in Table 1.

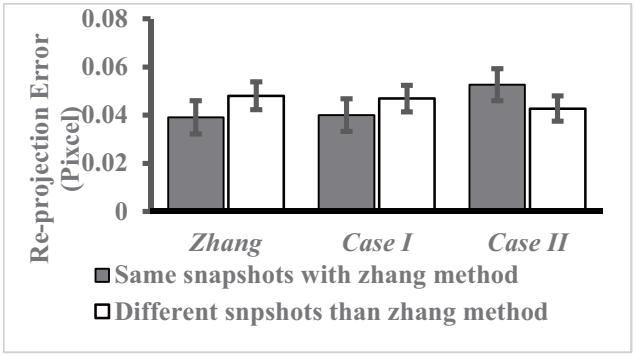


Figure 7: Comparing Re-Projection Errors

The re-projection evaluation experiment shows small errors for all trials. This may be attributed to the use of Lavenberg optimization inside *SolvPnP* function of which its main function is to minimize the reprojection errors.

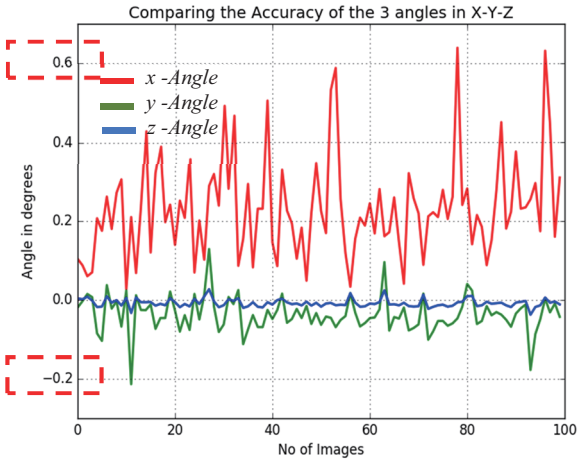


Figure 8: Camera calibration results using Zhang method and 10 ORB-matching points to estimate the 3D angles

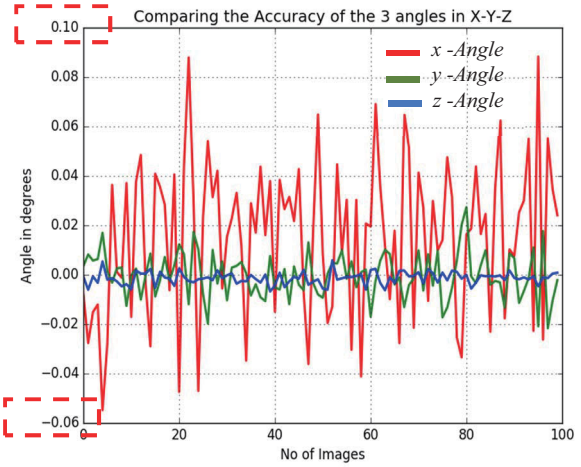


Figure 9: *Case I:* Proposed camera calibration results using 400 ORB-matching PPO training points to estimate the 3D angles

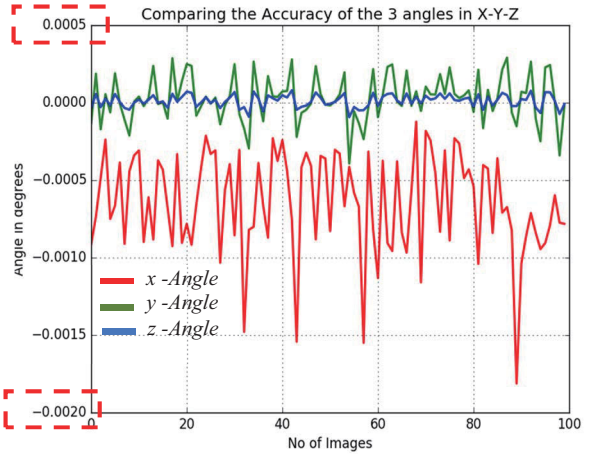


Figure 10: *Case II:* Proposed camera calibration results using 10 ORB-matching PPO training points to estimate the 3D angles

B. *Discussions*

In this section, we discuss the results presented in the previous section. The focal lengths (f_x, f_y) and principal point (u_0, v_0) are slightly different in all presented methods. This fact may be attributed to the fact that the focal lengths might not be ideally same for all points in the camera.

Using Zhang calibration method, the orientation estimation results shown in Fig. 8 fluctuates between -0.20° and 0.63°. The x-angle has the highest fluctuations range which is about 0° to 0.6°.

Using our proposed method, we can see good improvements specially when using 10 ORB points in the training phase. This may be because that the selected points' groups have more chances to be different than others. By using PPO algorithms, we try to approach the exact camera intrinsic matrix elements that meet the requirement of the stated objective function (Eq. (16)). *Case I* result in Fig. 9 shows fluctuations between -0.06° and 0.09°. These fluctuations mainly happen in x-angle data. *Case II* result in Fig. 10 shows fluctuations between -0.002° to 0.0003°. This shows significant improvements in the accuracy of our orientation angles estimations compared to Zhang method.

We test *Case II* results using V-Rep simulator to monitor camera orientation estimations in real time. Fig. 11 shows real-time experiment results to ensure that the intrinsic matrix of *Case II* results can also tackle other angles not only stationary as it is the case in training phase.

Figure 11: V-Rep Simulation of Real-time Camera 3D Angles Estimations using 10 ORB Points and Case II Intrinsic Matrix

VIII. Conclusion

The overall goal of this paper to improve camera pose estimation was successfully achieved. We introduce a new calibration method which only needs one highly structured reference image to get the camera intrinsic matrix compared to Zhang method that needs more than 10 reference pattern's images in different angles. Moreover, Zhang method accuracy depends on different factors like: quality of printed pattern, accuracy of pattern measurements, flatness of pattern surface and number of reference images used. Our proposed method only depends on structures in the image and learning objective function score at the time of terminating learning program. The stability of orientation results in ORB-Matching technique is improved by getting new parameters for the camera focal lengths (f_x, f_y) and principle point (u_0, v_0). The number of points used for the pose estimation in the training phase has different results; the case of 10 ORB points shows better results compared to 400 ORB points. This may be a result of the program has higher degree of possibility to change the groups of points used for the pose estimations in training phase.

However, Zhang method can also extract the distortion parameters that are not addressed in this paper because the used web camera image quality is good enough for our purpose. Thus, distortions can be neglected. This might not be the case for other type of cameras.

Good camera calibration leads to better pose estimation because the geometry of the camera is basically representing the intermediate relation between image and real-world coordinates.

Acknowledgment

Oman Government Ministry of Higher Education and Mitsubishi for Gas Corporation support this research, to which we would like to express our sincere gratitude.

References

[1] L. Jianxin, Q. Hangping, and W. Bo, "Survey of Structure from Motion," no. Cctot, pp. 72–76, 2014.
[2] R. Yu, T. Yang, J. Zheng, and X. Zhang, "Real-Time Camera Pose Estimation Based on Multiple Planar Markers," 2009.
[3] L. Tan, Y. Wang, H. Yu, and J. Zhu, "Automatic Camera Calibration Using Active Displays of a Virtual Pattern," pp. 1–13, 2017.
[4] T. Detection, "MATE : Machine Learning for Adaptive Calibration Template Detection," 2016.
[5] R. A. Boby and S. K. Saha, "Single Image based Camera Calibration and Pose Estimation of the End - effector of a Robot," pp. 2435–2440, 2016.
[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12.
[7] L. Zong, H. Wang, B. Wang, Q. Fu, and X. Sun, "An Improved Method of Real-time Camera Pose Estimation Based on Descriptor Tracking," pp. 903–908, 2017.
[8] I. J. Image, Z. Chai, and T. Matsumaru, "Feature Tracking and Synchronous Scene Generation with a Single Camera," no. June, pp. 1–12, 2016.
[9] Z. Zhang, "A Flexible New Technique for Camera Calibration," vol. 1998, 2008.
[10] opencv dev team, "Camera calibration With OpenCV," www.opencv.org..
[11] M. Turan, Y. Almalioglu, H. Araujo, and E. Konukoglu, "Deep EndoVO : A Recurrent Convolutional Neural Network (RCNN) based Visual Odometry Approach for Endoscopic Capsule Robots arXiv : 1708 . 06822v1 [cs . CV] 22 Aug 2017."
[12] Y. Hold-Geoffroy et al., "A Perceptual Measure for Deep Single Image Camera Calibration," 2017.
[13] E. Rublee and G. Bradski, "ORB : an efficient alternative to SIFT or SURF."
[14] H. Kim, J. Y. Lee, J. H. Kim, J. B. Kim, and W. Y. Han, "Object Recognition and Pose Estimation Using KLT," pp. 214–217, 2012.