Enhancing single camera calibration results using artificial bee colony optimisation within a virtual environment

Mojtaba A. Khanesar[1], Luke Todhunter[1], Vijay Pawar[2], Hannah Corcoran[2] Lindsay MacDonald[2], Stuart Robson[2], Samanta Piano[1]

[1]*Faculty of Engineering, University of Nottingham, NG8 1BB, Nottingham, UK*
[2]*Faculty of Engineering Science, University College London, WC1E 6BT UK*

Mojtaba.ahmadiehkhanesar@nottingham.ac.uk

Abstract

Close range photogrammetry is among the top candidates for non-contact metrology in high precision applications. It is frequently used within industrial environments for high precision measurement, automation, and control tasks. When using off-the-shelf cameras for such applications it is necessary first to understand how image content influences the image measurements made and in turn what effects this has on estimating imaging geometry. A virtual environment involving camera and digital objects may be used for testing the efficacy of machine learning algorithms. In this paper, enhancement of a single pose camera calibration process utilising a virtual environment and images taken with different lighting directions is investigated. The algorithm used for enhancing the calibration process is Artificial Bee Colony (ABC), a metaheuristic optimisation method. Multiple single image camera orientations are tested in this paper resulting in different extrinsic camera parameters. From multiple tests using different camera orientations, we observe that it is possible to enhance the calibration efficacy in terms of reprojection error using artificial bee colony when compared to an established two step Levenberg Marquardt (LM).

Vision sensor, artificial bee colony, central perspective camera, calibration, virtual environment, Blender, MATLAB, metrology

1. Introduction

Non-contact vision-based metrology is the process of performing measurements without physically contacting the object using imaging sensors [10, 11]. A range of metrology equipment within this category is routinely utilised within aerospace industries, automotive industries, and manufacturing environments [10, 11]. Photogrammetry [12, 13] and structured light metrology [12, 13] are typical techniques of non-contact vision-based metrology. However, the usage of cameras for measurements introduces systematic errors due to practical lens and sensor designs having significant non-linear distortions from the ideal central perspective projection [14]. Camera lens distortions and network solutions for multi-camera imaging geometries have been studied over many years, but it is only more recently that surface illumination and reflectance variations over the surfaces to be measured.

Use of a virtual environment to implement and test vision sensors and algorithms for manufacturing purposes is highly desirable [1-3]. In such an environment, designs for system configurations, algorithms, and sensors can be tested with high precision before practical implementation. A key task within a virtual manufacturing environment is evaluating optical metrology sensor placement under varying illumination conditions in order to estimate performance. In the work described in this paper, we are evaluating virtual environments able to simulate a multi-illumination dome imaging system. The system uses a single camera to take a sequence of images of a surface. Each image in the sequence is taken under point source illumination from different lighting directions. The resulting image stack is then used to estimate a pixel-by-pixel normal map based on knowledge of the sensing and illumination geometry.

Blender software with its ability to represent optical surface properties and light paths has been used for vision process simulation in medical environments [4, 5], for electrical component identification [6], in manufacturing environments [7, 8], and for educational purposes [9]. Drawing on Blender's OpenCV functionality and MATLAB integration, camera geometry including position and orientation, and internal optical parameters representing optical distortions can be modelled as an integral part of the virtual environment.

In this paper, images of a checkerboard calibration pattern, often used with OpenCV, and MATLAB, and representing a planar surface to be measured are synthesised. The image geometry is based on a perspective camera model with point source illumination provided by each of 64 different lighting orientations. Different lighting conditions result in different images of the checkerboard squares. The geometric effect of these variations can be usefully investigated as part of the least squares network adjustment process to estimate discrepancies between ideal and measured image coordinate reprojection residuals.

In the MATLAB environment [6], the camera was first calibrated using a two-step Levenberg Marquardt (LM) algorithm [15]. The calibrated parameters were then enhanced to determine if improvements were possible using an artificial bee colony (ABC) approach. The performance improvement of the proposed approach over two-step LM method is investigated under different orientations for the camera. It is shown that using the ABC algorithm results in enhancement of the camera calibration performance in terms of reprojection error.

The paper is organised as follows. Camera parameters including intrinsic and extrinsic parameters are explained in Section II. The basic camera calibration procedure using a checkerboard calibration board is discussed in Section III. ABC algorithm for enhancing camera calibration is summarised in Section IV. Overall multi-illumination dome system setup is presented in Section V. Simulation results, and conclusion remarks are presented in Sections V, and VI, respectively. Acknowledgements are given in Section VII.

2. Camera model: intrinsic and extrinsic parameters

2D imaging is the projection process of the real world in 3D onto a 2D image. The position of the features within the 2D image depend upon their real-world position and also on the camera parameters [17, 18]. Within OpenCV, and MATLAB machine vision toolbox, the overall projection process is formulated as follows,

$$
\begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} = \underbrace{\underbrace{\begin{bmatrix} f_x & \theta & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{intrinsic \\ matrix}} \underbrace{\begin{bmatrix} R_{xx} & R_{xy} & R_{xz} & T_x \\ R_{yx} & R_{yy} & R_{yz} & T_y \\ R_{zx} & R_{zy} & R_{zz} & T_z \end{bmatrix}_w}_{\substack{extrinsic \\ matrix}}}_{\substack{projection \\ matrix}} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}_w \quad (1)
$$

where (p_x, p_y, p_z) are the position coordinates of the point in real world, (p_u, p_v) are the corresponding coordinates of the point in the image in pixels, f_x and f_y are the camera intrinsic parameters such that $f_x = FS_x$, and $f_y = FS_y$, and $S_x(\frac{pixel}{mm})$, and $S_y(\frac{pixel}{mm})$ are the sensor's scale factors within x-coordinate, and y-coordinate, respectively. The parameter θ is the skew. The parameters $R_{ij}, i, j = x, y, z$ form the rotation matrix and $T_i, i = x, y, z$ form the translation matrix.

The radial distortion can be modelled as a polynomial function of the radius from the lens centre. The mathematical model of the distortion is as follows [19]:

$$
x_u = p_u(1 + k_2 r_d^2 + k_4 r_d^4 + \cdots)
$$
$$
y_u = p_v(1 + k_2 r_d^2 + k_4 r_d^4 + \cdots) \quad (2)
$$

where (x_u, y_u) are the coordinates of the points within the image as distorted by radial distortion function.

3. Camera calibration algorithms

The camera calibration application within MATLAB software [20] is used in this paper for calibration of the camera within Blender. The algorithm is summarised in this section. The assumptions are that:
1. the field of view for camera is no more than 95°.
2. the camera is modelled by a pinhole model.
To perform camera calibration identification, the following matrix equation is used within MATLAB:

$$
w \begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} = K[R \quad T] \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}_w \quad (3)
$$

where w is an arbitrary scaling factor, the matrix K is the camera intrinsic matrix, R presents the rotation matrix, and T presents

the translation matrix. In the first step of the algorithm, the corners of squares within the calibrating checkerboard are identified using image processing methods. Using the identified corners of squares and realistic corners from real-world, intrinsic and extrinsic parameters are estimated assuming that there exists no distortion using the algorithm given in [15]. The algorithm uses the LM Algorithm [21] as implemented in `Minpack` to find the projection matrix as well as intrinsic and extrinsic matrixes [22]. In the next step, considering the initial values for the distortion parameters as equal to zero and the distortion model of (2), a LM algorithm is applied on the overall camera projection process including the distortion parameters to estimate the overall parameters [23, 24].

4. Enhancement using artificial bee colony

To enhance the calibration results in terms of reprojection error, a meta-heuristic optimisation algorithm of artificial bee colony (ABC) is used. The objective is to find the optimal deviation of camera parameters from the calculated values from 2-step LM algorithm [15]. To perform such an optimisation, a cost function is defined as the difference between the projected real-world points to the image coordinate and the detected points on the image. The ABC algorithm is used to find the optimal deviation of the parameters with respect to the calculated values from LM algorithm optimisation method over all calibration images. This procedure is performed by optimising the cost function associated with the reprojection error.

This ABC algorithm employs swarm based optimisation which imitates the honeybee foraging behavior[25]. The optimal solution to this algorithm is the high dimensional position of the food source. Pseudocode of the ABC algorithm is given as follows [25]:
1. randomly initialise population $x_i, i = 1, \dots, SN$,
2. evaluate the population against the cost function,
3. the next few steps are repeated until convergence:
   a. a new set of calibration parameters is generated using the employed bee as $v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}), k \in 1, \dots, SN, j \in 1, \dots, D$, where $\varphi_{ij} \epsilon [0,1]$ is a uniform random number,
   b. the fitness function is evaluated for each solution $f(x_i)$ as the reprojection error between reprojection points from (3) using estimated camera parameters with the real points within the image.
   c. for each solution $f(x_i)$, its probability is given by:

$$
p_i = \frac{f(x_i)}{\sum_{i=1}^{SN} f(x_i)} \quad (4)
$$

   d. new solutions v_i for the onlooker bees are generated and evaluated from the solutions x_i selected depending on p_i,
   e. A greedy selection procedure is applied to select onlookers which ensures a new candidate is chosen based on its fitness compared to the current ones.
   f. Replace possible abandoned food sources with new food source using $x_{ij} = x_{i,min} + r_j(x_{i,max} - x_{i,min})$. The random parameter $r_j \epsilon [0,1]$ is a uniform random number.
   g. The overall best solution is updated by comparing the best solution in the iteration with the overall best solution.
   h. Terminate the algorithm if termination criteria are met, otherwise iterate the algorithm.

5. System Setup, image generation and processing

The imaging setup considered in this paper involves a fixed camera on the top of a dome, and 64 light sources located on the inside of the dome at fixed locations providing illumination geometry to take multiple images of the static object under the dome [26]. Figure 1 shows the CAD model of the system. Within this image, the position of the light sources is indicated by yellow, camera is positioned on top of the dome, and the dome with 1m radius is rendered in grey.
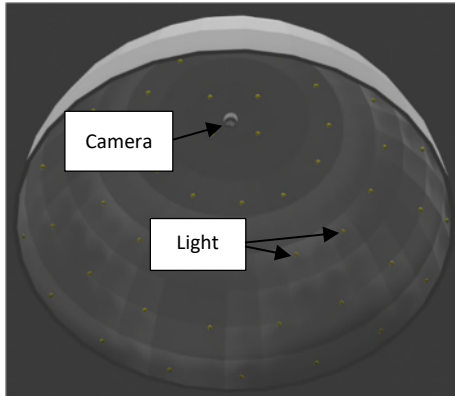

Figure 1. The CAD model corresponding to the dome, light sources, and camera on top.

To perform calibration, a checkerboard (Figure 2) is designed and imported into the Blender environment, centred in x-y plane. A camera is positioned 1 metre up the z-axis directly above the board with its Euler rotation angles being equal to (180, 0, -90). A total of 64 different light sources with beam angle of 15 degrees are used within this simulation environment, each facing towards the centre of the checkerboard. The light sources are turned on sequentially and synchronised with the camera shutter, resulting in 64 different images. Groups of images and real-world checkerboard corners are processed first with the 2-step LM method and then utilising the output extrinsic and intrinsic parameters as starting values with the ABC method.

6. Simulation results

A sample result from the calibration when the camera Euler rotation angle around axis is [-179.9141, 0.1432, -118.0724] is depicted in Figure 3. As can be seen, the reprojected points are very close to the originally detected square corners within the image. Reprojection errors in terms of pixel error values are shown in Figure 4. It is observed that the reprojection errors for the enhanced calibration using ABC algorithm are an order of magnitude smaller than those for the 2-step LM method. Table 1 presents the calibration results for 20 different camera orientations around the perpendicular downward direction. For each camera calibration, statistical results are given for 2-step LM algorithm and its enhanced version using ABC in terms of reprojection errors. As can be seen from this table, by using ABC algorithm the mean reprojection error decreases by 66.1% in comparison with 2-step LM method. Figure 5 presents the reprojection error versus orientation of the light sources in polar coordinates when camera Euler angle is equal to [-179.9141, 0.1432, -118.0724]. These figures clearly demonstrate that the calibration error using ABC algorithm is enhanced for images taken with light sources in most angles. Finally, the convergence graph for the ABC optimisation algorithm for the camera Euler angle is equal to [-179.9141, 0.1432, -118.0724] is illustrated in Fig. 6. This graph shows that the optimisation algorithm converges after more than 80 iterations.

Table 1 Reprojection error values

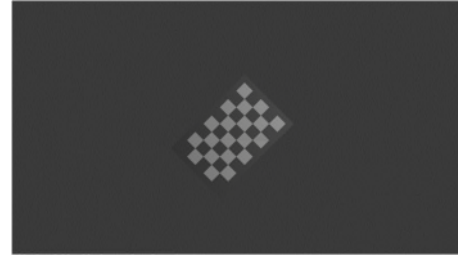|	Mean reprojection error (pixels)	Standard deviation of reprojection error (pixels)
Reprojection error values for checkerboard (pixels)	0.046	0.0712
Reprojection error values after ABC enhancement (pixels)	0.0156	0. 0178

Figure 2. Sample checkerboard calibration image.

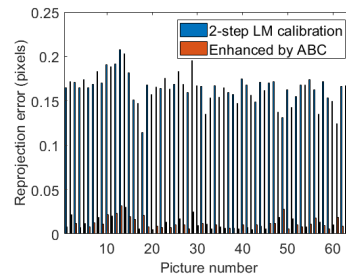**Figure 3.** Detected points versus reprojected points from enhanced calibration using ABC.

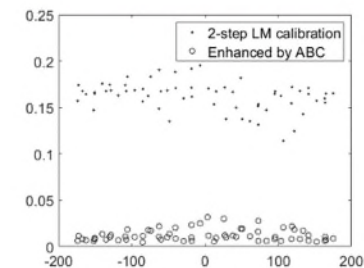**Figure 4.** Reprojection error obtained from checkerboard calibration.

Figure 5. reprojection error values as a function of θ.
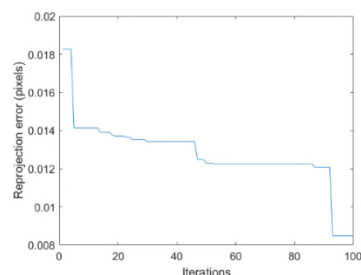

Figure 6. The convergence graph for ABC optimization algorithm.

7. Conclusions

In this paper, the Blender environment is used to make some initial vision simulation tests for an experimental imaging system. The work is part of an initial study to evaluate the metrology potential for simulating light sources, light surface interaction and camera geometry within a wider environment which might contain multiple objects with their reflections and occlusions. The basic simulation created in this software includes a camera mounted on top of a checkerboard and 64 light sources with 15 degrees beam angle facing towards the centre of the checkerboard. These light sources are turned on sequentially resulting in 64 different images of the checkerboard from a single camera pose. A cost function based on the mean reprojection error for the images is defined to optimise the camera parameters. ABC enhancement is shown to be an effective enhancement to a two-step LM algorithm calibration method. To show the effectiveness of this optimisation method, 20 different random camera orientations are used for calibration purposes. Comparison between the reprojected image from the image to the real world with and without enhancement are presented. Whilst these simulation results are limited to a single camera view, it is concluded that the ABC optimisation algorithm is worth pursing further to evaluate its effectiveness in both virtual and practical camera systems designed for non-contact metrology tasks.

Exploration of the camera pose and calibration determination within the virtual Blender environment has highlighted the value of further exploration for surface metrology applications. Further work will seek to better understand how such environments can support decisions on camera selection, machine learning method selection and implementation across different optical surfaces and geometries.

References

[1] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp,* vol. 61, pp. 335-340, 2017.

[2] M. Ericsson, X. Zhang, and A.-K. Christiansson, "Virtual commissioning of machine vision applications in aero engine manufacturing," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018: IEEE, pp. 1947-1952.

[3] J. Nilsson, M. Ericsson, and F. Danielsson, "Virtual machine vision in computer aided robotics," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*, 2009: IEEE, pp. 1-8.

[4] J. Cartucho, S. Tukra, Y. Li, D. S. Elson, and S. Giannarou, "VisionBlender: a tool to efficiently generate computer vision datasets for robotic surgery," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization,* vol. 9, no. 4, pp. 331-338, 2021.

[5] A. Munawar *et al.*, "Virtual reality for synergistic surgical training and data generation," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization,* vol. 10, no. 4, pp. 366-374, 2022.

[6] C. Wu *et al.*, "MotorFactory: A Blender Add-on for Large Dataset Generation of Small Electric Motors," *Procedia CIRP,* vol. 106, pp. 138-143, 2022.

[7] H. B. Lisboa, L. A. R. de Oliveira Santos, E. R. Miyashiro, K. J. Sugawara, P. E. Miyagi, and F. Junqueira, "3D virtual environments for manufacturing automation," in *22nd International Congress of Mechanical Engineering (COBEM 2013), University of São Paulo, Brazil November*, 2013, pp. 3-7.

[8] W. S. Harrison III and F. Proctor, "Virtual fusion: state of the art in component simulation/emulation for manufacturing," *Procedia Manufacturing,* vol. 1, pp. 110-121, 2015.

[9] M. Matys, M. Krajčovič, G. Gabajová, B. Furmannová, and P. Bubeník, "Integration of Virtual Reality into Teaching Process for Design of Manufacturing Systems," in *INTED2021 Proceedings*, 2021: IATED, pp. 190-198.

[10] M. A. Khanesar, M. Yan, M. Isa, S. Piano, and D. T. Branson, "Precision Denavit–Hartenberg Parameter Calibration for Industrial Robots Using a Laser Tracker System and Intelligent Optimization Approaches," *Sensors,* vol. 23, no. 12, p. 5368, 2023.

[11] S. Catalucci, A. Thompson, S. Piano, D. T. Branson III, and R. Leach, "Optical metrology for digital manufacturing: a review," *The International Journal of Advanced Manufacturing Technology,* vol. 120, no. 7-8, pp. 4271-4290, 2022.

[12] S. Catalucci, N. Senin, D. Sims-Waterhouse, S. Ziegelmeier, S. Piano, and R. Leach, "Measurement of complex freeform additively manufactured parts by structured light and photogrammetry," *Measurement,* vol. 164, p. 108081, 2020.

[13] A. Thompson *et al.*, "A single-click automated metrology demonstrator."

[14] R. Swaninathan, M. D. Grossberg, and S. K. Nayar, "A perspective on distortions," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2003, vol. 2: IEEE, pp. II-594.

[15] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 22, no. 11, pp. 1330-1334, 2000.

[16] F. Itami and T. Yamazaki, "An improved method for the calibration of a 2-D LiDAR with respect to a camera by using a checkerboard target," *IEEE Sensors Journal,* vol. 20, no. 14, pp. 7906-7917, 2020.

[17] F. Bottalico, C. Niezrecki, K. Jerath, Y. Luo, and A. Sabato, "Sensor-Based Calibration of Camera's Extrinsic Parameters for Stereophotogrammetry," *IEEE Sensors Journal,* vol. 23, no. 7, pp. 7776-7785, 2023.

[18] P. Sturm, "Pinhole Camera Model," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 610-613.

[19] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access,* vol. 9, pp. 113143-113155, 2021.

[20] (25/1/2024). *Camera calibration - Matlab R2023* [Online]. Available: https://uk.mathworks.com/help/vision/camera-calibration.html.

[21] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Levenberg marquardt algorithm for the training of type-2 fuzzy neuro systems with a novel type-2 fuzzy membership function," in *2011 IEEE symposium on advances in type-2 fuzzy logic systems (T2FUZZ)*, 2011: IEEE, pp. 88-93.

[22] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*, 2006: Springer, pp. 105-116.

[23] J. Heikkila and O. Silvén, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, 1997: IEEE, pp. 1106-1112.

[24] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[25] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review,* vol. 42, no. 1, pp. 21-57, 2014.

[26] L. W. MacDonald, A. H. Ahmadabadian, and S. Robson, "Determining the coordinates of lamps in an illumination dome," in *Videometrics, Range Imaging, and Applications XIII*, 2015, vol. 9528: SPIE, pp. 156-167.