A Comparative Study of the Application of Swarm Intelligence in Kruppa-Based Camera Auto-Calibration

Ahmad Fariz Hasan, Ali Abuassal, Mutaz Khairalla, Amar Faiz Zainal Abidin, Mohd Fairus Ahmad, Shamsul Amir Abdul Rais, Nor Hidayah Saad, Nur Anis Nordin, Ismail Mohd Khairuddin, Asrani Hj Lit, Hazriq Izzuan Jaafar

Abstract— This paper presented a comparative study of the application of two Swarm Intelligence algorithms: Particle Swarm Optimization and Firefly Algorithm in automatic camera calibration problem. The fitness function used in the camera calibration problem is based on the Kruppa's equation. A case study from a dataset provided by Le2i Universite de Bourgoune is taken for benchmarking the performance of both algorithms. The result is compared with previous literatures. Result obtained from these algorithms indicates there is potential for further improvement.

Index Terms— auto camera calibration; computational intelligence; firefly algorithm; particle swarm optimization; swarm intelligence; kruppa equation; fundamental matrix.

1 Introduction

In camera auto calibration problem, provided essential matrix and fundamental matrix, intrinsic parameters can be found by minimizing the cost function. The intrinsic parameters are the aspect ratio and skew represents the principal point. This paper employed two Swarm Intelligence algorithms: Particle Swarm Optimization and Firefly Algorithm in optimizing the intrinsic parameters by minimizing the cost function. The cost function chosen is the Kruppa's equation.

2 State Of The Art

Auto camera calibration problem required the algorithm to find optimal combination of camera calibration's parameters which keep the cost function at a minimal value. This problem is not a new problem where there are numerous literatures had attempted to solve the problem.

Y. Zhang and Q. Ji proposed the application of GA for camera calibration in year 2001 [6]. Genetic Algorithm is use to optimize interior and exterior camera parameters. The experimental result use synthetic and real images. The author claimed that GA produced an excellent performance in term of convergence, accuracy and robustness.

In year 2008, J. Z. Tao *et al.* proposed the application of Particle Swarm Optimization (PSO) in camera auto-calibration process [1]. In the paper, the author decided to find only theaspect ratio of the intrinsic parameters which consist of f_u and f_v (focal length in pixels along the axes of the image). The skew, γ is let to 0. While the other two intrinsic parameters: u_0 and v_0 are ignored based on recommendation by [2]. The fitness or cost function use in optimizing the parameters are as recommended by [3].

In the same year, K. Bilal and J. Qureshi investigate the application of several nature inspired optimization algorithms in tuning the parameters in camera auto-calibration. The algorithms use are Simulated Anneling (SA), Genetic Algorithm (GA), and PSO [4]. The main objective is to benchmark the performance of the algorithms based on several criteria: algorithm efficiency, algorithm accuracy, algorithm reliability, and calibration error (at different noise level). The authors concluded that if the application requires reliability, GA would be more suitable. While if the application requires precision, SA or PSO can be use.

On the following year, X. Song *et al.* proposed another implementation of PSO for single camera calibration [5]. A detailed experimental setup are explained in the paper. The result indicates that PSO provides satisfying calibration accuracy.

J. Li *et al.* proposed a hybrid of GA & PSO in order to improve the accuracy of the camera auto-calibration [7]. The simplified Kruppa's equation is use as cost function. The result indicates that the proposed approach produced 100% success rate compared to PSO (97%) and GA (98%).

- *Ahmad Fariz Hasan is an alumni of Universiti Teknologi Malaysia and currently a lecturer at University Malaysia Perlis, Malaysia. Email: farizhasan@unimap.edu.my*
- *Ali Abuassal is an alumni of Universiti Teknologi Malaysia.*
- *Mutaz Khairalla is an alumni of Universiti Teknologi Malaysia.*
- *Amar Faiz Zainal Abidin is currently a tutor at Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Malaysia. Email: amarfaiz@fke.utm.my.*
- *Mohd Fairus Ahmad is currently a lecturer at Universiti Malaysia Perlis, Malaysia. Email: fairusahmad@unimap.edu.my.*
- *Shamsul Amir Abdul Rais is currently a lecturer at Universiti Malaysia Perlis, Malaysia. Email: shamsulamir@unimap.edu.my.*
- *Nor Hidayah Saad is currently a lecturer at Universiti Malaysia Perlis, Malaysia. Email: hidayahsaad@unimap.edu.my..*
- *Nur Anis Nordin is currently a student of Universiti Teknologi MARA.*
- *Ismail Mohd Khairuddin is currently a lecturer of Universiti Malaysia Pahang.*
- *Asrani Hj Lit is currently a lecturer of Universiti Malaysia Sarawak.*
- *Hazriq Izzuan Jaafar is a lecturer in Universiti Teknikal Melaka, Malaysia.*

3 Motivation

Motivation to use Swarm Intelligence (SI) algorithms to optimize parameters in camera auto-calibration problem had been mentioned concisely in [1]. J. Z. Tao *et al.* mentioned that SI better than the traditional optimization algorithms:

Traditional optimization algorithms required initial value and sensitive to it. Most SI algorithms which based on Swarm Intelligence (SI) does not require initial value as the agent are randomly assigned in the search space at the initialization phase. Traditional optimization algorithms have higher chance to trap in local minima. Swarm Intelligence algorithms have less chance to trap in local minima due to their stochastic nature. Traditional optimization algorithms required extensive knowledge on the mathematical foundation behind the optimization strategies. All SI algorithms are nature inspired which make them easier to relate to. The algorithms also do not contain many mathematical equations.

Traditional optimization algorithms are more suitable to solve linear problem while SI algorithms can tackle both (linear and non-linear) problems quite well.

In this paper, the implementation of SI algorithms has been extended to five parameters compared to three in [1]. Implementation of Firefly Algorithm (FA) on camera auto-calibration also never been done before.

4 Camera Auto-Calibration

The main objective of camera calibration is to obtain the intrinsic camera parameters for a given number of images. In auto-calibration, as the word suggested, the camera calibration does not require supervision by the user. The intrinsic matrix is as shown in Equation (1).

$$K = \begin{bmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $[u_0, v_0]$ is the skew ratio, f_u is the product of focal length and magnification factor, ε. The magnification factor, ε is defined by Equation (2).

$$f_v = \varepsilon \times f_u \tag{2}$$

As suggested by previous authors [1, 3], a point, p is on the absolute conic case, vector $\boldsymbol{x} = (x, y, z)^{\boldsymbol{T}}$ satisfy Equation (3).

$$\boldsymbol{x}^T\boldsymbol{x} = 0 \tag{3}$$

Based on (3) we can then extend the work of [3], where now each point, p must satisfy Equation (4) or Equation (5).

$$p^T K^{-T} K^{-1} p = 0 \tag{4}$$

$$p^T \omega\, p = 0 \tag{5}$$

While the dual absolute conic of ω, ω^* is as Equation (6).

$$\omega^* = KK^T \tag{6}$$

R. I. Hartley in [3] simplified the Kruppa's equation to Equation (7) where values of r and s come from diagonal matrix, D which is described as Equation (8). Column vector, U are u_1, u_2, and u_3. Column vector, V are v_1, v_2, and v_3.

$$\frac{r^2 {v_1}^T \omega^* v_1}{{u_2}^T \omega^* u_2} = \frac{rs{v_1}^T \omega^* v_2}{-{u_2}^T \omega^* u_1} = \frac{s^2 {v_2}^T \omega^* v_2}{{u_1}^T \omega^* u_1} \tag{7}$$

$$D = \begin{bmatrix} r & & \\ & s & \\ & & 1 \end{bmatrix} \tag{8}$$

By assigning each part of Equation (7) as J_1, J_2, and J_3, we can obtained Equation (9) to Equation (11).

$$J_{i1} = \frac{r^2 {v_1}^T \omega^* v_1}{{u_2}^T \omega^* u_2} \tag{9}$$

$$J_{i2} = \frac{rs{v_1}^T \omega^* v_2}{-{u_2}^T \omega^* u_1} \tag{10}$$

$$J_{i3} = \frac{s^2 {v_2}^T \omega^* v_2}{{u_1}^T \omega^* u_1} \tag{11}$$

The optimized value of the parameters of intrinsic matrix can be obtained by finding value of parameters that minimize Equation (12).

$$Error = \sum_{z=1}^{im-1} \sqrt{{J_{i1}}^2 + {J_{i2}}^2 + {J_{i3}}^2} \tag{12}$$

where im is number of images. Note that Equation (12) is use as fitness function for SI algorithms.

5 Swarm Intelligence

Swarm Intelligence is an emerging field in Computational Intelligence (CI) where all the algorithms are inspired by the cooperative knowledge of nature. All SI algorithms consists of three main components: initial random position in the search space, fitness comparison between the agents and agent trying to improve their fitness by learning from other agents. Most of the algorithms differ on the third component where each algorithm has different learning methods based on the nature it inspired from.

One of the earliest SI introduced is PSO by J. Kennedy and R. Russell [8]. The movement of the flocking birds inspires the algorithm. The main gist of PSO is that the entire population will try to replicate their historical success and in the same try to follow the success of the best agent in the population. The movement of the agents in the search space dimension shows this attempt.

Firefly Algorithm was introduced by Xin-She Yang in 2007 which fundamentally based on the mating behavior of fireflies [9]. Instead the entire population trying to replicate the best agent success, agents in FA tries to compare its fitness with its neighbors. It will try to improve its fitness by learning from all agents that has better fitness than it.

For this experiment, both algorithms can be modeled using the same model. The proposed model suggests that the relationship of agent's position, $\boldsymbol{s}_I$ with intrinsic matrix parameters can be generalized as Equation (13).

$$\boldsymbol{s}_m = [f_u \; \beta \; u_0 \; f_v v_0]^T \quad (13)$$

Example, $\boldsymbol{s}_2 = [800\ 0\ 256\ 900\ 256]^T$ means that the 2nd agent suggesting that the parameters of the intrinsic matrix should be as follows: $f_u = 800, \beta = 0, u_0 = 256, f_v = 800, v_0 = 256$. Another point worth mentioning here is the selection of the fitness function. Fitness function is the function that the agents use to benchmark their proposed solution. This function must have a numerical value. As mentioned earlier, the fitness function of the SI algorithms is as shown in Equation (12) which can be rewritten as Equation (14).

$$f(\boldsymbol{s}_I) = \sum_{z=1}^{im-1} \sqrt{{J_{i1}}^2 + {J_{i2}}^2 + {J_{i3}}^2} \quad (14)$$

6 Modeling Camera Auto-Calibration Problem Using Firefly Algorithm

The algorithm starts by generating initial population of agent, randomly. Here the agent is the firefly. The fireflies' positions are evaluated using the fitness function in Equation (14). Light intensity, I_m is formulated to be equal to the inverse value of the firefly's fitness function as shown in Equation (15).

$$I_m = \frac{1}{f(s_m)} \quad (15)$$

From here on the algorithm will start looping until stopping criteria are fulfilled. For this study, maximum iteration, z is chosen as stopping criteria where the algorithm will stop when the iteration, t reached maximum iteration, z.

For each iteration, each agent will move toward to other agent with greater light intensity. The movement of this agent is bounded by Equation (16).

$$\boldsymbol{s}_m = \boldsymbol{s}_m + \beta_0 e^{-\gamma r_{mu}^2}(\boldsymbol{s}_m - \boldsymbol{s}_u) + \alpha \boldsymbol{e}_m \quad (16)$$

where r is the distance between two agents in Euclidean distance. Given agent m and agent n, the Euclidean distance can be calculated using Equation (17).

$$r_{mu} = \|\boldsymbol{s}_m - \boldsymbol{s}_u\| \quad (17)$$

β_0 is the agent's attractiveness at $r = 0$. γ is absorption coefficient. α is randomization parameter which in range [0,1]. $\boldsymbol{e}_m$ is a vector random number taken from uniform distribution.

Algorithm 1: Firefly Algorithm for camera auto-calibration

Set fitness function, $f(\boldsymbol{s}_m)$according to Equation (13) where $\boldsymbol{s}_m = [s_{m1}, s_{m2}, .., s_{mn}]^T$
Generate randomly initial population of agent, $\boldsymbol{s}_m$where $m = 1,2, .., q$
Find agent's light intensity, I_m at $\boldsymbol{S}_m$using Equation (15)
Define light absorption coefficient, γ
while $z < t$
  for m = 1 to q
    for u = 1 to q
      if $I_m < I_u$
        Move agent m towards u using Equation (16)
        Evaluate new solution using Equation (14), update $I_m$using Equation (15) and global best if necessary
      end if
    end for u
  end for m
end while
Post process results and visualization

The fitness of the new agent's position is evaluated and the light intensity is updated. If the fitness obtained smaller than the global best record, the new fitness will become the new global best and the agent's position is kept as the best solution found so far.

7 Modeling Camera Auto-Calibration Problem Using Particle Swarm Optimization

Similar to FA where PSO starts by randomly assigned the particle position based on Equation (13). Then the particle fitness is calculated using Equation (14).

Algorithm 2: Particle Swarm Optimization for camera auto-calibration

Initialize all particle by randomizing position based on Equation (13)
while$z < t$
  for m = 1 to q
    Calculate fitness for particle using Equation (14)
    if the particle fitness is better than previous $\boldsymbol{pbest}$then
      Set the particle fitness value as new $\boldsymbol{pbest}$
      if the $\boldsymbol{pbest}$is better than previous $\boldsymbol{gbest}$
      Set $\boldsymbol{pbest}$ as new$\boldsymbol{gbest}$
      end if
    end if
  end for m
  for m = 1 to q do
    Calculate particle velocity according to Equation (18)
    Update the particle position according to Equation (19)
  end for
end while
Post process results and visualization

Then the $\boldsymbol{pbest}$ and $\boldsymbol{gbest}$ will be updated if the particle has a better fitness value compared to the current $\boldsymbol{pbest}$ and $\boldsymbol{gbest}$ values. Then, the particle velocity, $\boldsymbol{v}_m^{z+1}$ is updated using Equation (18).

$$\boldsymbol{v}_m^{z+1} = \omega \boldsymbol{v}_m^z + r_1 c_1(\boldsymbol{v}_m^z - \boldsymbol{pbest}_m) + r_2 c_2(\boldsymbol{v}_m^z - \boldsymbol{gbest}) \quad (18)$$

where r_1 and r_2 are random values [0,1], c_1 is cognitive component and c_2 is social component. After that, the particle position is updated using Equation (19).

$$\boldsymbol{s}_m^{z+1} = \boldsymbol{s}_m^{z} + \boldsymbol{v}_m^{z+1} \tag{19}$$

The process is repeated until reaching the maximum iteration. The $\boldsymbol{gbest}$ is taken as the best found solution.

8 Implementation And Simulation Result

To compare the performance between the algorithms, the algorithms are tested using a dataset provided by Le2i Universite de Bourgoune [10]. The algorithm is written in MATLAB environment and the simulation is performed 10 times on a laptop equipped with 1.80GHz Intel Pentium Core 2 Duo processor with 2GB RAM.

Table 1 listed out the parameters values of SI algorithms used throughout this simulation.Both SI algorithms used same values for common parameters for benchmarking purposed. Table 2 listed out the fitness value obtained from the simulation done.

Table 1: Parameters of PSO and FA

	PSO	FA
Common Parameters		
Number of agents, i	100	100
Number of iterations, t	1000	1000
Number of computations	10	10
PSO Parameters		
Inertia weight, ω	0.9	Not applicable
Cognitive component, c_1	1.42	Not applicable
Social component, c_2	1.42	Not applicable
$r_1$and r_2	Random [0,1]	Not applicable
FA Parameters		
α	Not applicable	0.01
β	Not applicable	0.1
γ		0.001

Table 2: Fitness value obtained of PSO, FA and [11]

	PSO	FA	Levenberg-Marquart
Best	1.8481×10^{-7}	2.4394×10^{-4}	1.6051×10^{-13}
Worst	1.3582×10^{-4}	2.7532×10^{-4}	Not available
Mean	1.404×10^{-5}	2.669×10^{-4}	Not available
Standard Deviation	4.2787×10^{-5}	9.500×10^{-6}	Not available

In [11], the author implements self-calibration method proposed by [12]. The optimization strategy use is Levenberg-Marguart Algorithm (LMA) which is suitable for non linear system. Result indicates that LMA produces the best result, PSO at the second place, and FA in last place. The disadvantage of LMA is that it requires initial/rough estimation value of the intrinsic parameters while SI algorithms does not. From Table 2, it can be also seen that PSO is more accurate while FA is more precise. Table 3 listed the best and worst found parameters value of PSO, FA and LMA. The PSO and FA values are round up to one decimal point.

From Table 3, one can notice that if the proposed value of the best found PSO being round up to nearest integer, it will produces the same result like LMA. In general, SI algorithms are really high precision-value algorithm. To solve this problem, the user usually defines the number of precision of the parameters values.

Table 3: Parameters values proposed by PSO, FA and [11]

	PSO	FA	Levenberg-Marquart
Best	$\begin{bmatrix} 799.9 & 0.0 & 255.9 \\ 0 & 799.9 & 255.9 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 799.4 & 19.4 & 277.5 \\ 0 & 751.7 & 261.5 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 800 & 0 & 256 \\ 0 & 800 & 256 \\ 0 & 0 & 1 \end{bmatrix}$
Worst	$\begin{bmatrix} 989.9 & 0.0 & 304.8 \\ 0 & 1000.0 & 203.0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 716.2 & 58.6 & 270.8 \\ 0 & 670.0 & 254.9 \\ 0 & 0 & 1 \end{bmatrix}$	Not applicable

9 Conclusion

This chapter introduces reader to the application of SI to find optimized values of the intrinsic matrix's parameters for pinhole camera. The methodology of the proposed approach is explained in great details. The result indicates that there is potential for further study due to the good result obtained. Further study can be extended using different optimization strategies and fitness functions.

References

[1] J. Z. Tao, W. Wenhuan, W. Min, "Camera Autocalibration from Kruppa's Equations Using Particle Swarm Optimization," Proceeding of International Conference on Computer Science and Software Engineering, pp. 1032-1034, 2008.

[2] S. Bougnoux, "From Projective to Euclidean Space Under Any Practical Situation, A Criticism of Self-Calibration," Proceeding of 6th International Conference on Computer Vision, pp. 790-796.

[3] R. I. Hartley, "Kruppa's Equations Derived from the Fundamental Matrix," IEEE Transactions on Pattern Analysis and Machine Learning, vol. 19, no. 2, pp. 133-135, 2007.

[4] K. Bilal, J. Qureshi, "Nature Inspired Optimization Techniques for Camera Calibration," Proceeding of International Conference on Emerging Technologies, pp. 27-31, 2008.

[5] X. Song, B. Yang, Z. Feng, T. Xu, D. Zhu, Y. Jiang, "Camera Calibration Based on Particle Swarm Optimization," Proceeding of International Congress on Image and Signal Processing, pp. 1-5, 2009.

[6] Q. Ji, Y. Zhang, "Camera Calibration with Genetic Algorithms," IEEE Transactions on Systems, Man, and Cybernatics, Part A: Systems and Humans, vol. 31, no. 2, pp. 120-130, 2001.

[7] J. Li, Y. Yang, G. Fu, "Camera Self-Calibration Method Based on GA-PSO Algorithm," Proceeding of IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 149-152, 2011.

[8] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," Proceeding of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.

[9] X. S. Yang, "Firefly Algorithms for Multimodal Optimization," Lecture Notes in Computer Science, vol. 5792, pp. 169-178, 2009.
[10] University of Burgundy, Camera Autocalibration Dataset, URL: http://goo.gl/qxerh [Date Accesed: 19th January 2012].
[11] G. Lemaitre, "3D Digitization: Camera Autocalibration," URL: http://goo.gl/s8CTT [Date Accesed: 19th January 2012].